



Boru Bükme İşlemi Sonuçlarının Makine Öğrenmesi Algoritmaları ile Tahmin Edilmesi

Makine Mühendisliği Ana Bilim Dalı

Yüksek Lisans Tezi

Volkan Görüş

ORCID 0000-0001-8547-1968

Tez Danışmanı: Prof. Dr. Mehmet Çevik

Aralık 2021

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Volkan Görüş** tarafından hazırlanan **Boru Bükme İşlemi Sonuçlarının Makine Öğrenmesi Algoritmaları ile Tahmin Edilmesi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

ONAYLAYANLAR:

Tez Danışmanı: **Prof. Dr. Mehmet Çevik**
İzmir Kâtip Çelebi Üniversitesi

Jüri Üyeleri:

Dr. Öğr. Üyesi Çağlar Uyulan
İzmir Kâtip Çelebi Üniversitesi

Doç. Dr. Güzde Sarı
Manisa Celal Bayar Üniversitesi

Savunma Tarihi: 13.01.2022

Yazarlık Beyanı

Ben, **Volkan Görüş**, başlığı **Boru Bükme İşlemi Sonuçlarının Makine Öğrenmesi Algoritmaları ile Tahmin Edilmesi** olan bu tezimin ve tezin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 10.01.2022

Boru Bükme İşlemi Sonuçlarının Makine Öğrenmesi Algoritmaları ile Tahmin Edilmesi

Özet

Boru bükme işlemi sanayide, uzay-havacılık, otomotiv, inşaat, denizcilik, vb. birçok sektörde yoğun olarak kullanılmaktadır. Boru bükme işlemi manuel ve CNC kontrollü tezgâhlarda yapılmaktadır. Kullanılan malzemeye ve müşterinin isteği veya kullanım yerinin hassasiyetine göre çeşitli boru bükme yöntemleri kullanılmaktadır. Sanayide en çok kullanılan boru bükme yöntemleri; çekerek boru bükme yöntemi, sararak boru bükme yöntemi, basarak boru bükme yöntemi ve üç makaralı boru bükme yöntemidir. Boru bükme işleminde kullanılan borunun malzemesine, borunun dış çapına, et kalınlığına, bükme derecesine, bükme yarıçapına bağlı olarak farklı problemler oluşturmaktadır. Bu nedenlere bağlı olarak en çok karşılaşılan problemleri, borunun iç kısmında oluşan deformasyon (marullanma), borunun dış kısmında oluşan deformasyon (potlaşma), boru üzerinde oluşan izler, borunun çatlaması, borunun kırılması olarak sıralayabiliriz.

Bu çalışmada boru büküm işlemleri sonucunda ortaya çıkan problemleri önceden tahmin edebilmek için makine öğrenme (yapay zekâ) algoritmaları kullanılmıştır. İlk önce sabit ve değişken değerler tanımlanmıştır. Sabitler malzemenin cinsi ve büküm yarıçapı olarak; değişkenler ise borunun dış çapı, borunun et kalınlığı, büküm açısı, malafanın büküm eksenine uzaklığı ve malafa çeşidi olarak belirlenmiştir. Sabit ve değişken değerlere ait 150 adet deneyi içeren deneysel veri tablosu hazırlanmıştır. Deneysel veri tablosunda sonuçları verilen deneyler Mutlu Metal A.Ş.'deki CNC boru bükme makinesinde gerçekleştirilmiştir. Deneysel veri tablosundaki testler yapıldıktan sonra sonuçları listelenmiştir. Bu sonuçlar makine öğrenme algoritmasının ham verileri olarak kullanılmıştır. Altı farklı makine öğrenme algoritması kullanılarak sonuçlar tahmin edilmeye çalışılmıştır.

Bu çalışma sayesinde boru bükme işleminde ilk seferde doğru üretim yapılması sağlanarak, ön seri üretim maliyeti önemli ölçüde azaltılabilecektir.

Anahtar Sözcükler: Boru bükme, yapay zekâ, makine öğrenmesi, malafa, CNC, kaşık

Predicting the Results of the Pipe Bending Process with Machine Learning Algorithms

Abstract

Pipe bending is extensively used in aerospace, automotive, construction, marine, and in many other industries. Pipe bending is done manually and on CNC controlled workbenches. Various pipe bending methods are used according to the material used and the customer's request or the sensitivity of the place of use. The most commonly used pipe bending methods in the industry are; tube bending by pulling method, tube bending by wrapping method, tube bending by pressing method and three-roller tube bending method. Pipe bending creates different problems depending on the material of the pipe used, the outer diameter of the pipe, the wall thickness, the degree of bending, and the bending radius. Depending on these reasons, we can list the most common problems as deformation (surface warping) in the inside of the pipe, deformation on the outside of the pipe, traces on the pipe, cracking of the pipe, breakage of the pipe.

In this study, machine learning (artificial intelligence) algorithms are used to predict the problems that arise as a result of pipe bending operations. First, constant and variable values are defined. Constants values are material type and bend radius; variable values are the outer diameter of the pipe, the wall thickness of the pipe, the bending angle, the distance of the mandrel from the bending axis and the mandrel type. An experimental data table containing 150 experiments was prepared based on constant and variable values. The experiments, the results of which are given in the experimental data table, were carried out on the CNC pipe bending machine of Mutlu Metal A.Ş. The results are listed after the tests in the experimental data table are performed. These results were used as the raw data of the machine learning algorithm. The results were tried to be estimated using six different machine learning algorithms.

By means of this study, it will be possible to reduce the pre-series production cost significantly by ensuring the correct production at the first time in the pipe bending process.

Keywords: Tube bending, artificial intelligence, machine learning, mandrel, CNC, wiper

Teşekkür

Çalışmalarımnda her konuda bana yardımcı olan danışman hocam Sayın Prof. Dr. Mehmet ÇEVİK'e, bu süreçte yanımda olan ve desteğini esirgemeyen aileme ve makine öğrenmesi algoritmaları konusunda bilgi ve tecrübesini benimle paylaşan Sayın Dr. Mustafa BAHŞİ hocama teşekkür ediyorum. Ayrıca çalışmamın oluşmasında yardımını esirgemeyen, deneysel çalışmalara fırsat tanıyan Mutlu Metal San. ve Tic. A.Ş. ve Mutlu Marine San. ve Tic. A.Ş.'ye teşekkürlerimi sunarım.

İçindekiler

Yazarlık Beyanı	ii
Özet	iii
Abstract	v
Teşekkür	vii
Şekiller Listesi.....	xi
Tablolar Listesi.....	xiii
Kısaltmalar Listesi	xiv
Semboller Listesi.....	xv
1 Giriş	1
1.1 Genel Bilgiler	1
1.2 Boru Bükme İşleminde Kullanılan Ekipmanlar.....	2
1.2.1 Malafa	2
1.2.2 Bükme Kalıbı	2
1.2.3 Sıkma Kalıbı	3
1.2.4 Baskı Kalıbı	4
1.2.5 Kaşık Kalıbı	5
1.3 Boru Bükme Yöntemleri	5
1.3.1 Çekerek Boru Bükme Yöntemi	5
1.3.2 Sararak Boru Bükme Yöntemi	7
1.3.3 İterek Boru Bükme Yöntemi	7
1.3.4 Üç Makaralı Boru Bükme Yöntemi.....	8
1.4 Boru Bükme Süreçlerinde Karşılaşılan Problemler	10

1.4.1	Et Kalınlığının Homojen Olarak Kalmaması	10
1.4.2	Borunun İç Yüzeyinde Deformasyon Oluşması	11
1.4.3	Borunun Çıkış Ucunda Kambur Oluşması	11
1.4.4	Sıkma Kalıbı Bölgesinde Çiziklerin Oluşması	11
1.4.5	Büküm Alanında Malafanın Ucundaki Bilyelerin İzlerinin Oluşması	12
1.4.6	Büküm Alanının İç tarafında Çiziklerin Olması	12
1.4.7	Orta Çizgide İzlerin Oluşması	12
1.4.8	Geri Tepme	12
1.4.9	Çökme	13
1.4.10	Yırtılma	13
1.5	CNC Boru Bükme Makineleri	14
1.5.1	CNC Boru Bükme Makinelerinin Tarihçesi	14
1.5.2	CNC Boru Bükme Makinelerinin Çalışma Prensibi	15
2	CNC Boru Bükme İşlemi	16
3	Yapay Zekâ ve Makine Öğrenmesi	19
3.1	Yapay Zekâ	19
3.1.1	Yapay Zekânın Tarihçesi.....	19
3.1.2	Yazılım ve Programlama	20
3.1.3	Yapay Zekânın Tanımı ve Çeşitleri.....	21
3.2	Makine Öğrenmesi	23
3.2.1	Makine Öğrenmesinin Tanımı.....	23
3.2.2	Makine Öğrenmesinin Tarihçesi	25
3.2.3	Makine Öğrenmesinin Türleri	27
3.2.4	Makine Öğrenmesi Algoritmaları.....	29
3.2.4.1	Doğrusal Regresyon	29
3.2.4.2	Lojistik Regresyon	31
3.2.4.3	K-En Yakın Komşu	34

3.2.4.4	Destek Vektör Makinesi	35
3.2.4.5	Naive Bayes.....	37
3.2.4.6	Karar Ağacı	38
3.2.4.7	Rassal Orman	39
3.2.4.8	Bagging ve Boosting	40
3.2.4.9	Doğrusal Ayırt Edici Analizi	41
3.2.4.10	Boyut İndirgeme	41
3.2.4.11	Çapraz Doğrulama	42
4	DeneySEL Verilerin Elde Edilmesi	44
4.1	DeneySEL Veri Setinin Oluşturulması.....	44
4.2	DeneySEL Çalışma	45
5	Makine Öğrenmesinin DeneySEL Verilere Uygulanması	49
5.1	Makine Öğrenmesi Algoritmalarının Seçilmesi	49
5.2	Veri Ön İşleme	50
5.3	Lojistik Regresyon Metodunun Veri Kümesine Uygulanması.....	51
5.4	Karar Ağacı Metodunun Veri Kümesine Uygulanması.....	53
5.5	En Yakın Komşu Metodunun Veri Kümesine Uygulanması.....	55
5.6	Rassal Orman Metodunun Veri Kümesine Uygulanması.....	59
5.7	Naive Bayes Metodunun Veri Kümesine Uygulanması	61
5.8	Destek Vektör Makinesi Metodunun Veri Kümesine Uygulanması	63
6	Sonuç ve Değerlendirme	65
	Kaynaklar	67
	Ekler	70
	Özgeçmiş	76

Şekiller Listesi

Şekil 1.1	Malafa çeşitleri	2
Şekil 1.2	Bükme kalıbı	3
Şekil 1.3	Sıkma kalıbı.....	4
Şekil 1.4	Baskı kalıbı.....	4
Şekil 1.5	Kaşık kalıbı	5
Şekil 1.6	Çekerek boru bükme makinesi	6
Şekil 1.7	Sararak boru bükme yöntemi	7
Şekil 1.8	İterek boru bükme makinesi	8
Şekil 1.9	Üç makaralı boru bükme makinesi.....	9
Şekil 1.10	Et kalınlığındaki değişim	10
Şekil 1.11	Marullanma	11
Şekil 1.12	Geri tepme	13
Şekil 1.13	Yırtılma	14
Şekil 1.14	CNC boru bükme makinesi	15
Şekil 2.1	Boru bükme 3B çizimi	17
Şekil 2.2	Boru bükme işlemi 2B teknik çizimi	17
Şekil 2.3	CNC boru bükme makinesi program yazma modülü	18
Şekil 2.4	CNC boru bükme operasyonu	18
Şekil 3.1	Bir kaynak kodun bilgisayar tarafından işlenmesi	21
Şekil 3.2	Makine öğrenmesi süreci	23
Şekil 3.3	XOR sınıflandırması	26
Şekil 3.4	Makine öğrenmesi türleri	27
Şekil 3.5	Makine öğrenmesi türleri genel şeması.....	28
Şekil 3.6	Sigmoid fonksiyonu grafiği 1.....	32
Şekil 3.7	Sigmoid fonksiyonu grafiği 2.....	33
Şekil 3.8	En yakın komşu grafiği	34

Şekil 3.9	K-En yakın komşu grafiği.....	35
Şekil 3.10	Destek vektör makinesi algoritması	37
Şekil 3.11	Karar ağacı yapısı.....	38
Şekil 3.12	Rassal orman algoritmasının çalışma prensibi	40
Şekil 3.13	Bagging ve boosting arasındaki fark	41
Şekil 3.14	Çapraz doğrulama çalışma prensibi akışı	43
Şekil 4.1	Deney numunesi kodlama örneği.....	46
Şekil 4.2	Boru bükme işlemi sonucunun ölçüm yapılarak kontrolü	47
Şekil 4.3	Boru bükme işlemi sonrası görsel kusur	48
Şekil 5.1	Lojistik regresyon algoritması ROC grafiği.....	53
Şekil 5.2	Karar ağacı algoritması ROC grafiği	55
Şekil 5.3	En yakın komşu algoritması ROC grafiği (komşuluk derecesi = 5)	57
Şekil 5.4	En yakın komşu algoritması ROC grafiği (komşuluk derecesi = 2)	58
Şekil 5.5	Rassal orman algoritması ROC grafiği	60
Şekil 5.6	Naive Bayes algoritması ROC grafiği.....	62
Şekil 5.7	Destek vektör makinesi algoritması ROC grafiği	64

Tablolar Listesi

Tablo 3.1	Karışıklık matrisi (confusion matrix).....	24
Tablo 4.1	Deneysel çalışma parametre değerleri.....	45
Tablo 4.2	Deneysel veri kodlama tablosu	46
Tablo 5.1	Lojistik regresyon metodu ile elde edilen sonuçların karışıklık matrisi	52
Tablo 5.2	Lojistik regresyon metodu ile elde edilen sonuçların istatistiksel skorları	52
Tablo 5.3	Olasılık eşik değerlerinin değişimine göre lojistik regresyon metodu ile elde edilen modelin başarısı	52
Tablo 5.4	Karar ağacı metodu ile elde edilen sonuçların karışıklık matrisi	54
Tablo 5.5	Karar ağacı metodu ile elde edilen sonuçların istatistiksel skorları	54
Tablo 5.6	En yakın komşu metodu ile elde edilen sonuçların karışıklık matrisi (komşuluk derecesi = 5)	56
Tablo 5.7	En yakın komşu metodu ile elde edilen sonuçların istatistiksel skorları (komşuluk derecesi = 5)	56
Tablo 5.8	En yakın komşu metodu ile elde edilen sonuçların karışıklık matrisi (komşuluk derecesi = 2).....	57
Tablo 5.9	En yakın komşu metodu ile elde edilen sonuçların istatistiksel skorları (komşuluk derecesi = 2)	58
Tablo 5.10	Rassal orman metodu ile elde edilen sonuçların karışıklık matrisi	59
Tablo 5.11	Rassal orman metodu ile elde edilen sonuçların istatistiksel skorları ...	60
Tablo 5.12	Naive Bayes metodu ile elde edilen sonuçların karışıklık matrisi	61
Tablo 5.13	Naive Bayes metodu ile elde edilen sonuçların istatistiksel skorları	62
Tablo 5.14	Destek vektör makinesi metodu ile elde edilen sonuçların karışıklık matrisi	63
Tablo 5.15	Destek vektör makinesi metodu ile elde edilen sonuçların istatistiksel skorları	64

Kısaltmalar Listesi

AI	Yapay Zekâ (Artificial Intelligence)
CAM	Bilgisayar Destekli Üretim (Computer Aided Manufacturing)
CNC	Bilgisayar Sayısal Kontrol (Computer Numerical Control)
ML	Makine Öğrenmesi (Machine Learning)
ROC	Receiver Operating Characteristic
RSS	Residual Sum of Squares (Artık Kareler Toplamı)
TSS	Total Sum of Squares (Toplam Kareler Toplamı)

Semboller Listesi

β_0	bilinmeyen 1
β_1	bilinmeyen 2
ε	artık hata
x	Giriş değeri (Input)
\bar{x}	Ortalama giriş değeri
x_i	Herhangi bir giriş değeri
y	Sonuç değeri (Output)
\bar{y}	Ortalama sonuç değeri
\hat{y}_i	Herhangi bir i noktasındaki tahmini sonuç değeri

Bölüm 1

Giriş

1.1 Genel Bilgiler

Sanayide birçok alanda kullanılan boru bükme işlemi; belirli bir şekle ve tasarıma sahip boruların üzerine kuvvet uygulayarak boruya kalıcı şekil verme işlemidir. Bu işlem metal, alaşım, alüminyum, titanyum ve çelik gibi malzemeler için geçerlidir. Boru bükme ile ilgili işlemler önce tamamen insan gücüne dayalı el aparatları yardımıyla yapılırdı. Daha sonraları sırasıyla hidrolik sistemli boru bükme makineleri ve Bilgisayar Sayısal Kontrollü (CNC) boru bükme makineleri kullanılmaya başlandı. CNC boru bükme tezgâhları günümüzde artık yaygın olarak kullanılmaktadır. CNC tezgâhlarının sanayide kullanılmasının avantajlarını, daha düşük toleranstaki ürünlerin yapılması, kontrol edilebilme olanağının yüksek olması, ayar işlemi tamamlandıktan sonra hızlı ve seri üretim yapılması (maliyet ve zaman avantajı) olarak sıralayabiliriz. CNC boru bükme tezgâhında imalatını yaptığımız bir ürünün üretim zamanı ile insan gücüne dayalı mekanik metotlarla yapılan boru bükme işleminin üretim zamanı karşılaştırıldığında çok büyük bir fark olduğu görülecektir. Bu nedenle insan gücüne dayalı mekanik metotlarla yapılan boru bükme işlemlerinin kapsam alanları her geçen gün daha da daralmaktadır. CNC boru bükme tezgâhları ile yapılan bir parçanın işlem süresi önceden tayin edilebilmekte, her defasında aynı kalite ve özellikte (ölçülerde) parça üretimi yapılabilmektedir. Malzemenin çeşidine ve kullanım yerinin zorluk derecesine göre çeşitli boru bükme yöntemleri kullanılmaktadır. Bu yöntemlerden en yaygın kullanılanları çekerek bükme ve üç makaralı bükmedir.

1.2. Boru Bükme İşleminde Kullanılan Ekipmanlar

1.2.1 Malafa

İstenilen kalitede boruyu bükmek için kullanılan malafa (mandrel), borunun iç kısmında kalarak bükme bölgesinin çökmesini veya taşmasını yani form değişikliğine uğramasını engellemek için kullanılmaktadır [1-3]. Şekil 1.1'de görüldüğü gibi borunun dış çapı ve et kalınlığına göre kullanılan farklı malafa çeşitleri vardır. Bunlar topsuz malafa, tek toplu malafa, iki toplu malafa, üç toplu malafa ve dört toplu malafadır. Daha önce borunun bükme bölgesinde form bozuklukları olmaması için ıslak kum, plastik kablolar kullanılmaktaydı. Günümüzde kullanılan malafalar Ampco bronz veya çelik malzemelerden uygun ölçülerde yapılmaktadır. Genellikle borunun iç çapından 0,3mm küçük tasarlanarak boruya rahat girmesi sağlanır.



Şekil 1.1: Malafa çeşitleri

1.2.2 Bükme Kalıbı

Borunun bükme esnasında içine konumlandığı, boru bükme elemanıdır [2,3]. Soğuk şekillendirme işlemi yapan hemen hemen bütün kalıplar gibi bükme kalıpları da ısıtılma işlemi uygulanarak sertleştirilir. Şekil 1.2'de bükme kalıbı resmi verilmiştir. Bükme kalıpları 60-62 Rockwell C sertliğe sahiptirler. Çok düşük toleransta ve hassasiyette

büküm istenildiği takdirde bükme kalıbındaki boru yuvaları istenilen toleransa ve hassasiyete göre taşlama işlemine tabi tutulur.



Şekil 1.2: Bükme kalıbı

1.2.3 Sıkma Kalıbı

Sıkma kalıbı (Şekil 1.3) bükme işlemi yapılacak olan boruya uygulanan kuvveti dengeleyen ve daha önceden tasarlanmış borunun bükme sonrası belirlenen ölçüyü kaçırmamasını engelleyen ekipmandır [3]. Kalıp üzerinde açılan boru yuvaları, boruya uygulanan kuvvet esnasında kayma yapmaması için talaşlı imalatta tırtıl işlenir veya gerekli yüzeylere ince kum zımpara kâğıdı yapıştırılır.



Şekil 1.3: Sıkma kalıbı

1.2.4 Baskı Kalıbı

Şekil 1.4’de verilen baskı kalıbının görevi ise boruyu bükmek için kuvvet uygulandığında üretilen radyal stresi alarak borunun bükme ekseninden çıkmamasını sağlamaktır [3].



Şekil 1.4: Baskı kalıbı

1.2.5 Kaşık Kalıbı

Kaşık (wiper) kalıbı genellikle et kalınlığı düşük olan borularda kullanılır. Bu kalıp daha iyi bükme yapılması ve borunun formunun bozulmaması için sıyırıcı görevi üstlenir [2,3]. Şekil 1.5’te kaşık kalıbı verilmiştir.



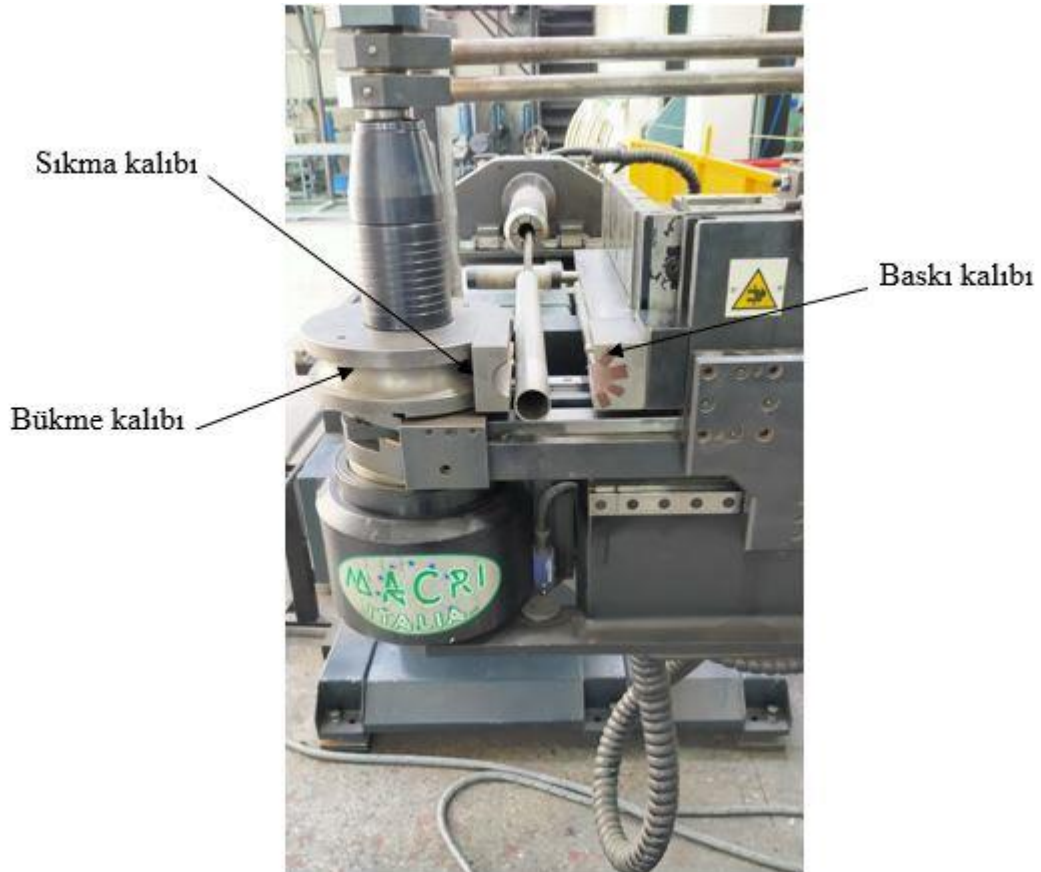
Şekil 1.5: Kaşık kalıbı

1.3 Boru Bükme Yöntemleri

1.3.1 Çekerek Boru Bükme Yöntemi

İlk uygulamaları 1800’lü yılların başına dayanan çekerek boru bükme yöntemi günümüzde sanayide en çok kullanılan boru bükme yöntemidir. Diğer yöntemlere göre daha düşük toleranslı olması ve boru bükme işlemi sırasında kontrollü ilerlemesi bu yöntemin tercih edilmesinin en önemli nedenidir. Şekil 1.6’da çekerek boru bükme makinesi ve makine elemanları sunulmuştur. Çekerek boru bükme yöntemi ile imal edilen parçalar günümüzde otomotiv sektöründe, mobilya sektöründe, tarım araç ve gereçlerinde, eğitim materyallerinde, spor araç ve gereçlerinde başta olmak üzere neredeyse yaşamın her alanında kullanılmaktadır. Çekerek boru bükme yönteminde;

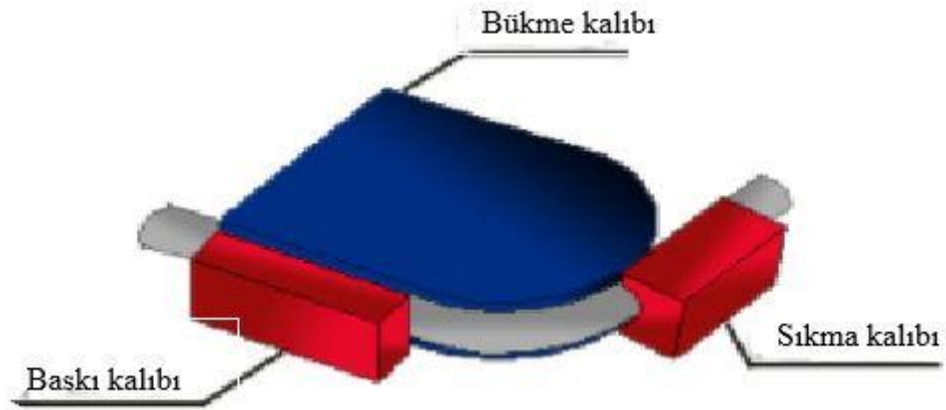
ön hazırlıkları yapılmış borunun (istenilen boyda kesilmiş, çapak temizleme işlemleri yapılmış ve borunun uç tarafları yağlanmış) içine uygun malafa yerleştirilir, bükme kalıbı bükme eksenini etrafında döndürülerek boruyu sarar ve borunun bükme işlemi gerçekleştirilir [4]. Bu boru bükme yönteminde malafa ve kaşık en önemli boru bükme ekipmanlarıdır. Bükme esnasında borunun içinde kalan malafa bükmede oluşacak problemlerin önüne geçer, diğer önemli boru bükme elemanı olan kaşık ise boruya uygulanan kuvvet esnasında iç bükme yüzeyinde oluşabilecek deformasyonları sıyrarak önler. Her bükme için kaşık gerekli değildir. Fakat ihtiyaç duyulduğu birçok alan vardır ve sorunu engellemek için çok önemli bir ekipmandır. Çekerek bükme yönteminin avantajları; işlem hassasiyeti, dar bükme yarıçapları ile çalışma imkânı, iki bükme arasındaki mesafeyi kontrol edebilmedir. Dezavantajı ise, kalıp maliyetleri yüksek olması nedeniyle bükme yöntemleri içinde en maliyetli olmasıdır.



Şekil 1.6: Çekerek boru bükme makinesi

1.3.2 Sararak Boru Bükme Yöntemi

Sararak boru bükme yöntemi çekerek boru bükme yöntemine oldukça benzerdir. Şekil 1.7'de şeması sunulan bu yöntemde, boru kaygan sıkma kalıbı ile sabit bükme kalıbı arasına sıkıştırılır. Bükme yarıçapı bloğu etrafında dönen kaygan sıkma kalıbı boruyu bükme kalıbının yarıçapı kadar bükür [4]. Günümüzde bu teknik bükme tezgâhlarında sıklıkla kullanılmaktadır. Bu tezgâhlarda çekerek boru bükme tezgâhında bulunan malafa mevcut değildir. Malafanın bu tezgâhta olmaması gerçekleştirilecek işin oluş şekline göre avantajlı veya dezavantajlı olabilmektedir. Bu bükme metodu genel olarak, borunun et kalınlığının fazla ve bükme yarıçapının boru yarıçapının üç katına kadar olduğu durumlar için uygundur. Çıkan ürünler müşterinin beklentilerini karşılıyor ise, sararak bükme metodu ekonomik açıdan oldukça uygun bir boru bükme metodudur. Bu bükme yönteminin avantajları, malafasız bükme için ekonomik olması ve kolay taşınabilir olması pratik çözümler üretmesidir. Dezavantajları ise, bükme kalitesinin düşük olması ve aynı parça üzerinde birden fazla bükme yapmanın oldukça zor olmasıdır.



Şekil 1.7: Sararak boru bükme yöntemi

1.3.3 İterek Boru Bükme Yöntemi

İterek boru bükme yöntemi sanayide mekanik ve hidrolik aparatlarla kullanılmaktadır. Bu boru bükme yönteminin çalışma prensibi şudur: uygun çapraz makaraların arasına yerleştirilen boru itici bir kuvvet yardımıyla makaraların çevresinde bükülür [4]. Bu

boru bükme metodunda da malafa desteği olmadığından dolayı et kalınlığı fazla ve bükme yarıçapı büyük olan boruların bükme işleminde kullanılmaktadır. Şekil 1.8’de gösterilen iterek boru bükme yönteminin avantajları basit ve ucuz olması, kelepçelerin pozisyonları değiştirilerek uygulanabilmesi, çeşitli araçlar ile geniş kullanım yelpazesi olmasıdır. Dezavantajları ise, standart ve devamlılığı olan bükümlerde dereceleri aynı çıkarmaması, genellikle bakır gibi yumuşak malzemelerin bükümleri için kullanılabilmesi, sadece tek bükümlü işler yapmak için uygun olması, bükümlerde içten ve dıştan deformasyonun oldukça belirgin olmasıdır.



Şekil 1.8: İterek boru bükme makinesi

1.3.4 Üç Makaralı Boru Bükme Yöntemi

Üç makaralı bükme yöntemi (Şekil 1.9), mekanik veya hidrolik sistemler ile sanayide en çok kullanılan bükme yöntemlerinden bir tanesidir. Boru, profil, lama, köşebent, şekilli profiller, dolu malzemeler bu alanda başlıca kullanılan bükme malzemeleridir. Bu boru bükme yöntemi, büyük yarıçaplı boruların bükme işlemleri için çok uygundur. Ayrıca boruya dışardan ısı vermeden soğuk deformasyon yöntemi ile 250, 350, 450 mm çapındaki büyük çaplı boruların bükme işlemini sağlayan tek yöntemdir. Bu yöntem, küçük çaplı borularda olduğu gibi uygun makaralar tezgâha bağlandığı takdirde büyük çaplı boruların bükme işlemlerinde de kullanılır. Bu yöntemde

tezgâhın üzerinde sabit ve hareketli makaralar vardır ve boru makaralar arasında yerleştirilir. İki makara altta sabit ve bir makara üste hareketli olarak tasarlanmıştır. Üstteki hareketli makara dikey yönde aşağı-yukarı hareket ettirilerek istenilen bükme yarıçapı oluşturulur. Hareketli olan üst makaranın aşağı yönde hareketi, aşağıdaki sabit olan iki makaraya yaklaşması bükme yarıçapını küçültür. Bunun tam tersi yönde hareketi ise bükme yarıçapını büyültür. Bükümü gerçekleştirecek olan borunun sabit ve hareketli makaraların arasından geçmesini gerçekleştirmek için makaraların birkaçı veya hepsine güç iletilir. Bu nedenle bu yöntemde bükme işlemi için ilave bir kuvvete ihtiyaç yoktur. Bükme yarıçapı makine olanaklarına, ölçülerine ve makaraların montaj yerlerine bağlıdır, fakat genel olarak bu bükme yöntemi büyük bükme yarıçaplı ve büyük çapta borular için daha uygun olmaktadır [4-6]. Sanayide mekanik veya hidrolik sistemler çok yaygın kullanılmaktadır. Fakat mekanik ve hidrolik sistemlerde borunun beslemesini ve ilerleme boyunun kontrol edilmesi zor olduğundan toleransları dar olan hassas bükme için CNC sistemleri kullanılmaktadır. Bu yöntemin avantajı ekonomik olması, aynı kalıplarla farklı bükme yarıçapları oluşturulabilmesi ve büyük yarıçaplarda fire ve hurda vermeden kullanım olanağı sağlamasıdır. Bu yöntemin avantajlı olmayan durumları ise; küçük bükme yarıçaplarında kullanılamaması, parça üzerinde farklı bükme yarıçaplarının kontrol edilememesi ve iki bükme arasındaki mesafenin kontrol edilme zorluğudur.



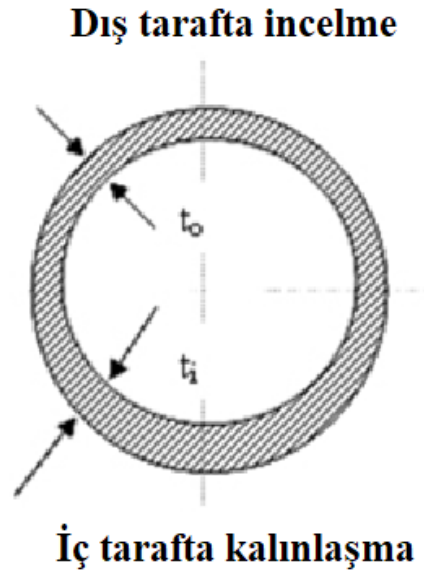
Şekil 1.9: Üç makaralı boru bükme makinesi

1.4 Boru Bükme Süreçlerinde Karşılaşılan Problemler

Bu bölümde boru bükme süreçlerinde karşılaşılan problemler ele alınacaktır. En sık karşılaşılan problemler, et kalınlığının homojen olarak kalmaması, borunun iç yüzeyinde deformasyon olması (marullanma), borunun çıkış ucunda kambur oluşması, sıkma kalıbı bölgesinde çiziklerin oluşması, büküm alanında malafanın ucundaki bilyelerin izlerinin oluşması, büküm alanının iç tarafında çiziklerin oluşması, orta çizgide izlerin oluşması, geri tepme, çökme ve yırtılmadır [7-11].

1.4.1 Et Kalınlığının Homojen Olarak Kalmaması

Boru bükme işlemi sırasında boruyu bükmek için uygulanan kuvvet karşısında boruda oluşan tepki kuvveti nedeniyle borunun et kalınlığında dış taraftan incelme iç taraftan ise kalınlaşma olabilmektedir. Şekil 1.10'da boru et kalınlığındaki değişimin şematik gösterimi sunulmuştur. Bu problemi gidermek için çekerek bükme yönteminde uygun malafa kullanılır. Diğer boru bükme yöntemlerinde ise genellikle yaşanan bir problemdir [11].



Şekil 1.10: Et kalınlığındaki değişim

1.4.2 Borunun İç Yüzeyinde Deformasyon Oluşması

Bükme yarıçapının küçük ve et kalınlık faktörünün (boru dış çapının et kalınlığına oranı) [1] düşük olduğu durumlarda marullanma adı verilen problemle karşılaşılır. Şekil 1.11’de borunun iç yüzeyinde oluşmuş deformasyon (marullanma) görülmektedir. Marullanma sorununun üstesinden gelmek için uygun kaşık kalıbı kullanılmalıdır [11].



Şekil 1.11: Marullanma

1.4.3 Borunun Çıkış Ucunda Kambur Oluşması

Boru bükme işlemi sırasında bükme kalıbı ile sıkma kalıbı arasında belirli bir mesafe olmalıdır. Bu mesafe genellikle 0,5–1mm arasındadır. Bu mesafenin bırakılmasının amacı, bükme işlemi yapmak için boru yuvasına oturturulmuş borunun daha iyi kavramasını ve parçanın kuvvet altında kaymasını önlemektir. Aksi halde boruda kambur adı verilen deformasyon oluşur [11].

1.4.4 Sıkma Kalıbı Bölgesinde Çiziklerin Oluşması

Bükme kalıbına yerleştirilen borunun bükülmesi esnasında, sıkma kalıbı bükülecek parçayı istenilen oranın dışında kavırsa, parça üzerinde uygulanan kuvvetin etkisi ile

parçanın sıkma kalıbına denk gelen bölgelerinde çizikler oluşur. Bunu önlemek için sıkma kalıbı tasarlanırken istenilen değerlere dikkat edilmelidir [11].

1.4.5 Büküm Alanında Malafanın Ucundaki Bilyelerin İzlerinin Oluşması

Çekerek boru bükme yönteminde kullanılan malafanın toplu olması durumunda bu problemle karşılaşılır. Malafanın boyutları ve malzemesi uygun seçilmediğinde borunun büküm bölgesinde malafanın ucunda bilyeye benzeyen malzemenin izlerinin oluşmasına neden olur [11].

1.4.6 Büküm Alanının İç Tarafında Çiziklerin Olması

Çekerek boru bükme yönteminde yaşanan problemdir. Kullanılan malafa istenilen ölçülerde ve uygun malzemede yapılmamışsa borunun büküm alanının iç tarafında çizikler oluşmasına sebep olur. Bu problemi ortadan kaldırmak için malafanın uygun ölçülerde ve uygun malzemedan yapılması gerekmektedir [11].

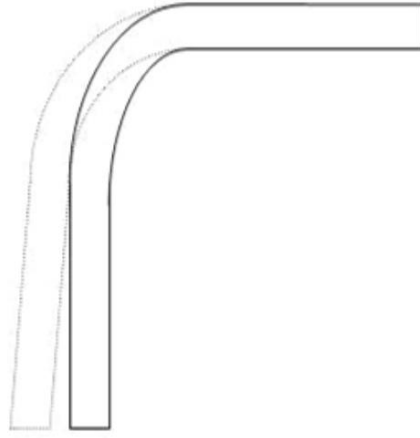
1.4.7 Orta Çizgide İzlerin Oluşması

Borunun dış yüzeyinde herhangi bir deformasyon olmaması için baskı kalıbı kullanılmaktadır. Baskı kalıbı borunun merkezinden dış duvarlara doğru oluşan gerilimi alarak borunun sırtı diye tabir edilen borunun dış yüzeyinde deformasyon olmasını engeller. Baskı kalıbı boru bükme işleminde bu görevi üstlenirken uygulanan kuvvetten dolayı boru baskı kalıbının içine doğru sürüklenir. Baskı kalıbı için, bükülecek boruya göre uygun malzeme seçimi yapılmadıysa ve uygun toleranslarla tasarlanmadıysa borunun baskı kalıbına temas ettiği orta noktalarda izler oluşur. Baskı kalıbı yapılırken uygun malzeme (örneğin, küresel tanecikli dökme demir) seçildiği ve uygun toleranslarla tasarlandığı takdirde bu problem çözülebilir [11].

1.4.8 Geri Tepme

Geri tepme, borunun bükme işlemi bitmek üzereyken, boru üzerinde kalan artık gerilmeler nedeniyle bükülen borunun istenilen ölçüye geldikten sonra uygulanan

bükme kuvvetinin ortadan kaldırılmasıyla, borunun kendi kendine büküm derecesinin tersine doğru hareket etmesiyle oluşan problemdir. Şekil 1.12’de gösterilen bu problemi ortadan kaldırmak için CNC tezgahlarında simülasyon programları kullanılır, NC veya mekanik boru bükme tezgahlarında ise ilk ayar hurdasında istenilen ölçü verildikten sonra çıkan ürünün üzerinden ölçü alınarak geri tepme sonucu uygun olmayan ölçü değeri ile istenilen ölçü değeri arasındaki fark kadar istenilen ölçüye ilave edilerek veya çıkarılarak problem çözülmüş olur [11,12].



Şekil 1.12: Geri tepme

1.4.9 Çökme

Et kalınlık faktörünün (borunun dış çapının et kalınlığına oranı) düşük olduğu borularda uygun malafa seçimi yapılmadığında boruya uygulanan dış kuvvetin etkisiyle borunun dıştaki kısmında çöküntüler oluşur. Bu problemi gidermek için borunun et kalınlık faktörüne dikkat edilerek uygun malafa seçimi yapılmalıdır [11].

1.4.10 Yırtılma

Bu problem türü, küçük bükme yarıçaplı boru bükümlerinde yaşanmaktadır. Bükme işlemini gerçekleştirmek için uygulanan dış kuvvet nedeniyle borunun büküm bölgesinin dış kısmında oluşan gerilime, boru malzemesi dayanamadığında yırtılma olur. Şekil 1.13’te borunun bükümü esnasında yırtılma oluşmuş bir numune

görülmektedir. Tasarım yapılırken bükme yarıçaplarına dikkat edilerek tasarım yapıldığı takdirde bu sorun ortadan kalkacaktır [11].

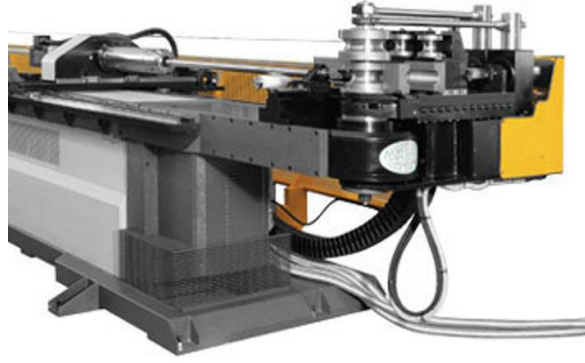


Şekil 1.13: Yırtılma

1.5 CNC Boru Bükme Makineleri

1.5.1 CNC Boru Bükme Makinelerinin Tarihçesi

Boru bükme operasyonları mekanik ve hidrolik makinelerde yapılırken teknolojinin ilerlemesi ve ilerleyen teknolojinin sanayide kullanılmasıyla, imalat makinesinde olduğu gibi boru bükme makinelerinde de CNC makinelere geçiş yapılmıştır. CNC boru bükme makineleri (Şekil 1.14) sayesinde firmaların işçilik süreleri kısaltılmış, işlem hızları artmış, aynı kalitede yüksek adetlerde parçalar üretilmeye başlanmıştır. Bu da firmalarda rekabet gücünü arttırmıştır.



Şekil 1.14: CNC boru bükme makinesi

1.5.2 CNC Boru Bükme Makinelerinin Çalışma Prensibi

CNC tezgâhların çalışma prensibi, makinelere bağlanmış bilgisayarlarla ve bu makinelere uygun programlama dili sayesinde iş parçasının hızlı bir şekilde ve hep aynı kalitede üretilmesini sağlamaktır. İşlem adımları, makineyi kullanan ve yetkinliği olan operatör yardımıyla CNC makinesi üzerindeki bilgisayar yardımıyla uygun programlama dili ile manuel yazabilir. Alternatif olarak, kişisel bir bilgisayarda bir CAD programı ile hazırlanan tasarım, ilgili programlama diline çevrilip istenilen formatta kayıt altına alınır ve USB bellek yardımıyla CNC makinesinde bulunan monitörün ilgili kısmına atılır, böylece CNC tezgâh programlanmış olur. Daha sonra mekanik aksamda yer alan her ekseninde bulunan servo/step motorlara gönderilen sinyaller sayesinde tezgâh hareket kabiliyetine ulaşır. Servo/step motorlarında enkoder bulunmaktadır. Bu enkoderler motorların gücüne göre seçilip montajları gerçekleştirilir. Uygun enkoderler CNC tezgâhındaki hareketin sağlanması için gereken eksene vidalı mil yardımı ile bağlanır. Ardından, kaplin yardımı ile de motora montajları gerçekleştirilir. Motora enerji verildikten sonra motorun dönme hareketi ilgili eksene iletilmiş olur.

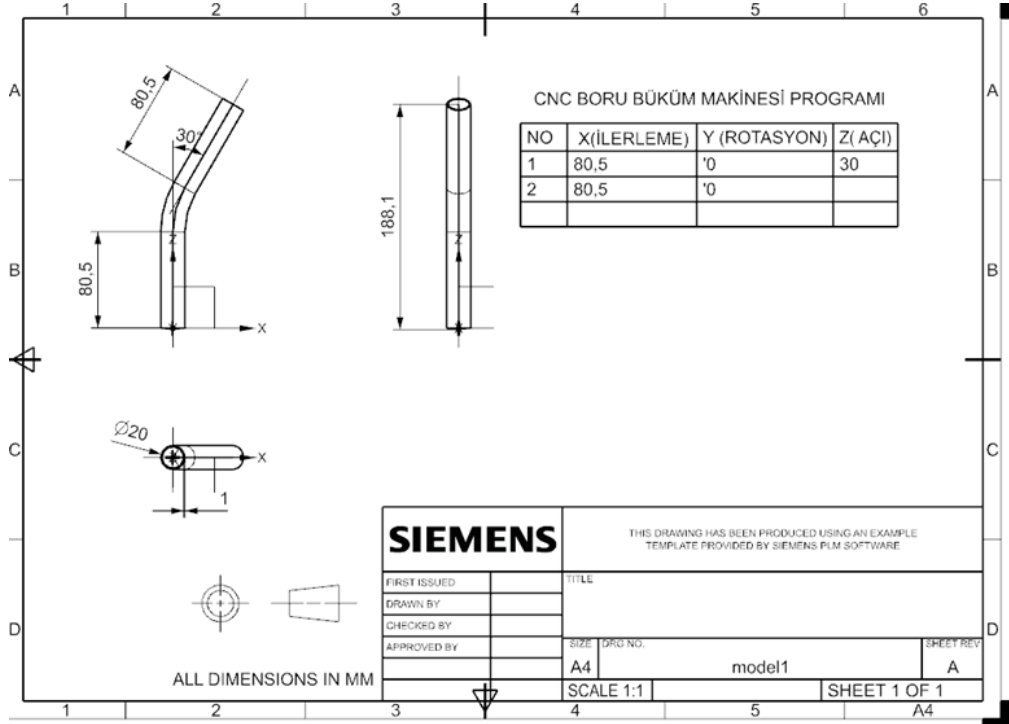
Bölüm 2

CNC Boru Bükme İşlemi

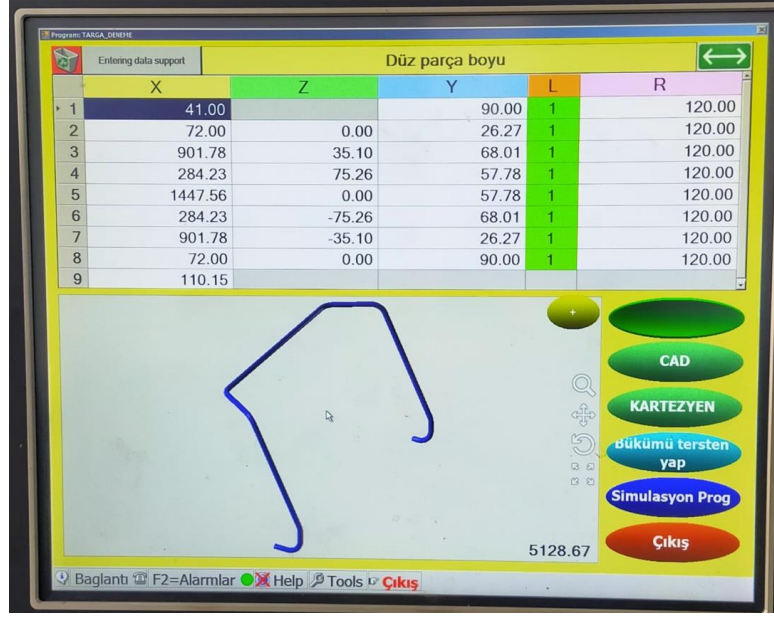
CNC boru bükme makinelerinde, ilgili iş parçasının problemsiz bükme işlemini gerçekleştirebilmek için bükme kalıbı, baskı kalıbı, sıkma kalıbı ve et kalınlığı faktörü dikkate alınarak kaşık kalıbı kullanılır. CNC boru bükme makinelerinde, çekerek boru bükme yöntemine göre genellikle daha uygun bir çalışma sistemi vardır. Boru iç çapına uygun malafa tasarlanmalıdır. Ayrıca borunun tarafsız (asal) eksenini temel alınarak program yazılabilmesi için 3B (Şekil 2.1) ve 2D (Şekil 2.2) çizimlerinin uygun bir çizim programında yapılması gerekmektedir. Tasarımı yapılan işlemin tasarım şartları göz önünde bulundurularak bükme yarıçapı, borunun dış çapı, borunun iç çapı ve borunun malzemesine göre CNC tezgâhına uygun kalıplar bağlanarak tezgâh ayarlanır. Tezgâha hareket emri verebilmemiz için program yazmamız gerekmektedir. İki çeşit program yazma yöntemi vardır. Birinci yöntemde boruya uygulanacak bükme yarıçapı, borunun bükülmeye başladığı noktaya kadar olan düz boru boyu 2B veya 3B çizimlerden alınarak Şekil 2.3'te de görüldüğü gibi tezgâhın üzerindeki veri giriş panelinde ilgili bölüme yazılır daha sonra bükme derecesi girilir, bir sonraki bükmede aynı şekilde ilerleyerek program sonlandırılır. İkinci yöntem ise kartezyen yöntemidir. Bükümü gerçekleştirilecek olan borunun asal ekseninin kesişim noktasının koordinatları (X, Y, Z) tezgâhın panelinde ilgili bölümlere girerek program yazılmış olur. Daha sonra yazılan program tezgâh üzerindeki simülasyon programında kontrol edilerek uygunluğuna bakılır. Herhangi bir problem yok ise Şekil 2.4'teki gibi boru bükme operasyonuna başlanır. İlgili programdaki değerlere göre tezgâh hareketlerini tamamlayarak bükme işlemi tamamlanmış olur.



Şekil 2.1: Boru bükme 3B çizimi



Şekil 2.2: Boru bükme işlemi 2B teknik çizimi



Şekil 2.3: CNC boru bükme makinesi program yazma modülü



Şekil 2.4: CNC boru bükme operasyonu

Bölüm 3

Yapay Zekâ ve Makine Öğrenmesi

3.1 Yapay Zekâ

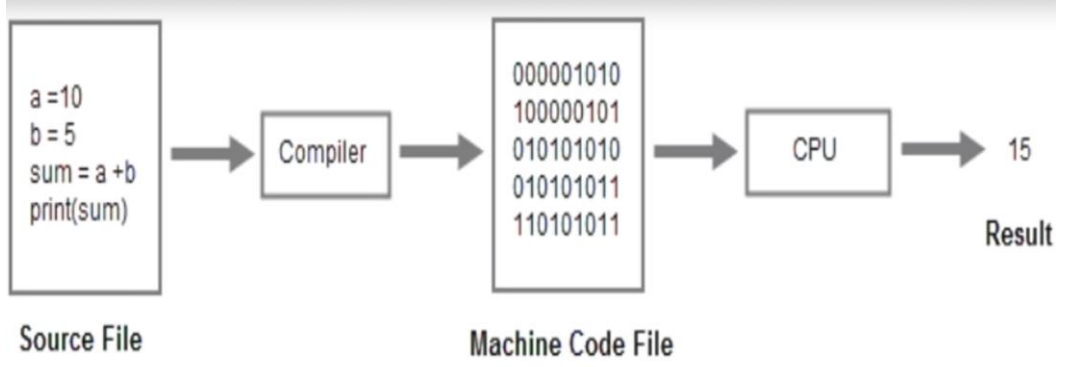
3.1.1 Yapay Zekânın Tarihçesi

Yapay zekâ günümüzde birçok insanın hafızasında yer edinen ve oldukça sık karşılaşılan ve kullanılmaya başlanan bir kavram haline gelmiştir. II. Dünya Savaşı sırasında Naziler tarafından gönderilecek gizli mesajlarının şifrelenmesi ve şifreli mesajlarının tekrar çözülmesi (kripto analiz) için kullanılan Enigma cihazı ile kripto analizi yaparak şifreli mesajları insanların anlayabileceği ifadelerle çeviren Turing makineleri ile yapay zeka kavramı hayatımıza dahil olmuştur [13]. Bir makinenin düşünebileceğini söylemenin mantıksal olarak mümkün olup olmadığını test etmek için bu konuda araştırmalar yapan İngiliz asıllı matematikçi ve bilgisayar bilimci olan Alan Turing, ilk olarak 1950 yılında Turing testi adı verilen kavramı ortaya atmıştır. Turing testi şu soruya yanıt aramaktadır: Birbirini tanımayan insanlar bilgisayar üzerinden birbirleriyle ve yapay zekâ diyalog yazılımı ile yazışarak haberleşiyorlar ve buna göre test yapan kişi yazdığı ve aldığı sorulardan acaba hangi bilgisayarın arkasında gerçek bir insan, hangisinin arkasında bir yazılım olduğunu bulabilecek midir? Sorulabilecek sorularda bir sınır bulunmamakta ve her türlü soru yöneltilebilmektedir [13]. 2014 yılına kadar yapılan testlerde başarıya ulaşamamıştır. Rus ve Ukraynalı bilgisayar programcıları tarafından geliştirilen Eugene Goostman adındaki yapay zekâ, 2014 yılında Turing testini geçmeyi başarmıştır. İşin enteresan tarafı yapay zekânın temeli sayılan Turing testini başka bir yapay zekânın geçmiş olmasıdır. Bu da yapay zekânın gelişimini gözler önüne sermektedir. Yapay zekâ akıllı makineler oluşturmanın genel disiplini. Çevremizde bulunan her maddenin kaynağı

atom olduğu gibi yapay zekânın kaynağı da verilerdir. Verilerden çıkarımlar yapmamızı sağlar. Yapay zekâ makine öğrenmesi ve derin öğrenme alt başlıklarına sahiptir. Kısaca bahsetmek gerekirse makine öğrenmesi, makineye sağlanan verilerle mantıklı, rasyonel sonuçlar üretmeyi sağlayan algoritmalarıdır. Örnek olarak, süper markette alışveriş fişleri ile oluşturulan algoritma sayesinde pasta alan müşterinin aynı zamanda yüksek ihtimalle içecek de alacağı sonucuna varılabilir. Derin öğrenmede buna ek olarak yapay sinir ağları kullanılır. Yapay sinir ağları insan beyninde kullanılan sinir ağlarının bir bakıma bilgisayar bilimine uyarlanmış halidir. Makine öğrenmesi ve derin öğrenme arasındaki fark da tam anlamıyla budur. Yani veriler arası farklar makine öğrenmesinde belirlenir ve sisteme tanıtımı gerçekleştirilir, derin öğrenmede ise bu farklılıklar yapay sinir ağları sayesinde otomatik olarak algılanır. Yapay zekâ teknolojisi her gün yeni bir alanda kullanılmaya başlanmakta ve birçok alanda yenilikler getirmektedir. Ülkeler stratejik bir alan olarak gördükleri yapay zekâ yatırımları için yıllık milyar dolar bütçeler ayırmaktadır. Yapay zekânın yaygın olarak kullanıldığı birkaç sektöre örnek olarak; endüstri, otomotiv, finans, uzay, sağlık bilimleri, tarım, servis hizmetleri sayılabilir.

3.1.2. Yazılım ve Programlama

Programlama veya diğer adıyla yazılım; bilgisayarlarda 1 (bir) ve 0 (sıfır)'lardan oluşan makine kodudur. Makine kodları, insanlar tarafından anlaşılabilirliği için ilk önce programlama dillerine çevrilirler. Şekil 3.1'deki örnekte belirtildiği gibi program dilini şöyle açıklayabiliriz. **a** değerine 10, **b** değerine 5 sayısı ataması yapılmış, **a+b** toplamının değeri talep edilmiş ve sonuç değerinin yazdırılması (print(sum)) amaçlanmıştır. Program dillerinde yazılan bu dosyaya kaynak dosyası adı verilmektedir. Kaynak dosyası (source file) önce derleme (compile) yapılır yani programlama dillerinde yazılan kod makine dillerine (makinenin anlayacağı koda) çevrilir; yani, 0 (sıfır) veya 1 (bir)'lere çevrilir. Daha sonra program çalıştırılır ve CPU yardımıyla talep edilen sonuca ulaşılır. Programlama dilinin temel mantığı bu şekildedir. Yazılan program önce derlenir sonra CPU tarafından çalıştırılarak sonuca ulaşılması sağlanır.



Şekil 3.1: Bir kaynak kodun bilgisayar tarafından işlenmesi

Günümüzde birçok programlama dili kullanılmaktadır. Her bir programlama dilinin özelleştiği farklı alanlar vardır. Örneğin, Python programla dili yapay zekâ algoritmalarının geliştirilmesinde, JavaScript programlama dili web yazılımlarının geliştirilmesinde, R programlama dili istatistiksel analiz ve veri bilimi çalışmalarında sıklıkla kullanılmaktadır. Dünya genelinde son yıllarda Python programlama dilinin en yaygın kullanılan programlama dili olduğu göze çarpmaktadır. Bunun nedeni ise python programlama dilinin kolay yazıla bilirliği, açık kaynaklı olması ve zengin kütüphaneye sahip olmasıdır.

3.1.3. Yapay Zekânın Tanımı ve Çeşitleri

Birçok tanımı olmasına rağmen genel olarak insan zekâsını taklit eden yazılımlar, yapay zekâ olarak adlandırılır. Bir yazılımın insan gibi düşünebilmesi ve insan gibi davranabilmesi ve ayrıca bunlara ek olarak rasyonel düşünüp, rasyonel şekilde hareket edebilmesine yapay zekâ denir. Yapay zekânın dört tane ana unsuru vardır: insan gibi düşünme (thinking like human), insan gibi hareket etme (acting like human), rasyonel düşünme (thinking rationally) ve rasyonel hareket etmedir (acting rationally). Aşağıda yapay zekâ ile ilgili temel bazı kavramlardan bahsedilmiştir [13].

- **İnsan gibi düşünme:** İnsan gibi düşünmeyi amaçlar. İnsan çok boyutlu bir varlıktır. Bundan dolayı felsefe, mantık, davranış bilimi, fizik gibi birçok kavramı hayatında barındırır. İnsan bir iş yaparken nasıl düşünürse, bu düşünme şeklinin yazılıma aktarılmasına insan gibi düşünme denir [13].

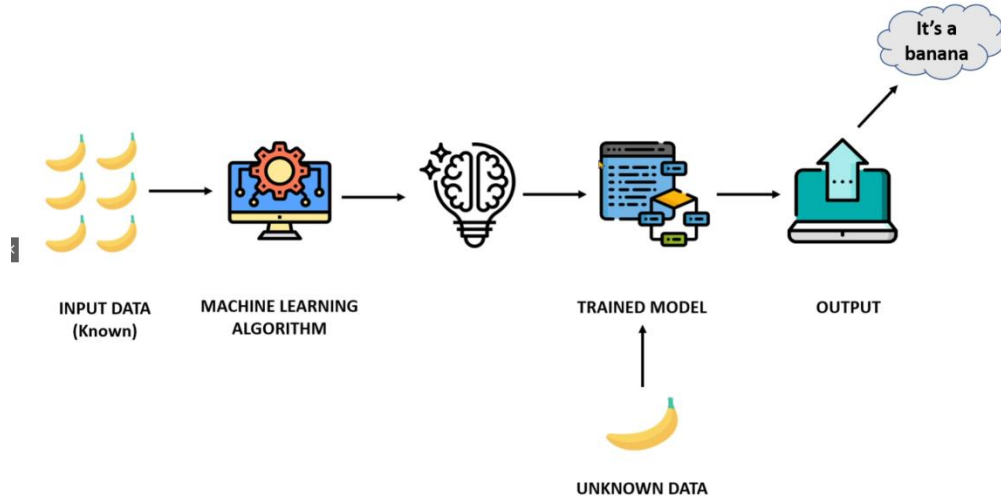
- **İnsan gibi hareket etme:** Bu kavram Alan Turing tarafından Turing testi ile açıklanmıştır. Buradaki amaç yapay zekâ sisteminde sorulan sorulara insan gibi cevap verebilmektir [13].
- **Rasyonel düşünme:** Temel amacı zihinsel işlemlerin, hesaplamalı yani somut modeller kullanılarak incelenmesidir. Doğru düşünerek ve her soruya doğru cevap vererek doğru karar almayı temel amaç edinir [13].
- **Rasyonel hareket:** Her sistemde olduğu gibi bu sistemin de bir iç ve bir dış yapısı vardır. Yani, dış ortam ile bir iletişimi vardır. Düşünme bizim içyapımızdır. Davranış ise dış ortama bağımlı olan kısımdır. Örnek verecek olursak; otonom araçlar çevrelerinde bulunan sensörler sayesinde dış ortamdaki verileri toplarlar ve karar vermek için iç tarafta bulunan komuta merkezindeki algoritmaları kullanırlar. İç ortam ve dış ortam arasında bilgi alış verişi yapılması ve alınan bu bilgiler ile bir karara varılabilmesi için yapay zekâda ajan kullanılmaktadır [13].
- **Ajan (rational agent):** Türkçeye kelime karşılığı ajan olarak çevrilmiştir fakat yapay zekâda bu kelimenin karşılığı insan zekâsı anlamına gelmektedir. İşlerin doğruluk dereceleri, harcanan süre (ne kadar sürede yapılabildiği), kullanılan elektrik miktarı, işlem yapılırken çıkan gürültü miktarı ile değerlendirilir. Algılama (percept), sensörler aracılığı ile alınan verilerdir. Ajanın algılama geçmişi, algılanan her verinin listelenmiş haline denir. Rasyonellik kavramı dört etkene bağlıdır. Bunlar başarı kriterlerini tanımlayan performans ölçüsü, ajanın çevre hakkında önceden bilgisi, ajanın gerçekleştirebileceği işler, ajanın bu güne kadar algılama sırasındadır. Ajan algıladığı bilgileri farklı çevrelerden elde eder. Bu çevre türleri ise tamamen gözlemlenebilir (fully observable) ve tamamen gözlemlenemeyen (partially observable) ortamlar olmak üzere iki çeşittir [13].
- **Tamamen gözlemlenebilir (fully observable) ortam:** Ajan sensörler ile çevresini tamamen taradığında ve tarama işlemi zamanının her saniyesi mümkün oluyorsa tamamen gözlemlenebilir ortam denir. Yani çevre hakkında bütün değerlere sahip olur [13].

- **Tamamen gözlemlenemez (partially observable) ortam:** Zamanın bütün saniyelerinde gözleme yapılamaz. Belli başlı zamanlarda gözleme yapılır. Çevre hakkında kısmi bilgilere sahip olunur [13].

3.2 Makine Öğrenmesi

3.2.1 Makine Öğrenmesinin Tanımı

Bilgisayar programlarının kendi deneyimleri aracılığıyla otomatik olarak öğrenmelerini ve gelişmelerini sağlayan algoritmaların bütününe makine öğrenmesi (machine learning) denilmektedir [14]. Makine öğrenmesi sürecinde, Şekil 3.2’de gösterildiği gibi ham veriler (input) mevcuttur. Mevcut olan veriler uygun algoritmalarla işlenip programlanır ve sonrasında eğitilir, bu eğitilmiş model bir sonuç oluşturur. Daha önce tanıtılmamış yeni bir veri sisteme dâhil olduğunda eğitilmiş geçmiş verilere istinaden sisteme yeni giren tanımadığımız veri ile ilgili sonucun tahmin edilmesine makine öğrenmesi süreci denir.



Şekil 3.2: Makine öğrenmesi süreci

Makine öğrenmesinde bir sınıflandırma algoritmasının performansını özetlemek için karışıklık matrisi (confusion matrix) adı verilen bir teknik kullanılır. Karışıklık matrisi; deneysel çalışma sonunda ortaya çıkan sonuçlar ile sınıflandırma ile tahmin edilen sonuçların karşılaştırması yapar ve performansını ifade eder [15]. Her sınıfta eşit

olmayan sayıda gözleminiz varsa veya veri kümenizde ikiden fazla sınıfınız varsa, sınıflandırma doğruluğu tek başına yanıltıcı olabilir. Bir karışıklık matrisi hesaplamak, sınıflandırma modelinizin neyi doğru yaptığı ve ne tür hatalar yaptığı konusunda size daha iyi bir fikir verebilir. Tablo 3.1’de görüldüğü gibi, karışıklık matrisi dört değere sahiptir [15,16].

Tablo 3.1: Karışıklık matrisi (confusion matrix)

	Tahmin 1:	Tahmin 0:	Toplam
Deney 1:	TP: Doğru Pozitif	FN: Yanlış Negatif	TP+FN
Deney 0:	FP: Yanlış Pozitif	TN: Doğru Negatif	FP+TN

Doğru Pozitif (TP: True Positive): Bir koşulun veya özelliğin varlığını doğru olarak gösteren bir test sonucu.

Yanlış Negatif (FN: False Negative): Belirli bir koşulun veya özelliğin bulunmadığını yanlış bir şekilde gösteren bir test sonucu.

Yanlış Pozitif (FP: False Positive): Belirli bir koşulun veya özelliğin mevcut olduğunu yanlış bir şekilde gösteren bir test sonucu.

Doğru Negatif (TN: True Negative): Bir koşulun veya özelliğin olmadığını doğru bir şekilde gösteren bir test sonucu.

Karışıklık matrisinde bulunan veriler sayesinde makine öğrenme algoritmalarının performansı belirlenebilmektedir. En çok kullanılan performans metrikleri; doğruluk, duyarlılık, hassasiyet ve f1 ölçütü dur.

Doğruluk (Accuracy): Gerçek deneysel veriler ile karşılaştırıldığında doğru tahmin yapılmış verilerin toplam verilere oranına denir [17,18].

$$Doğruluk = \frac{TP + TN}{TP + FN + FP + TN}$$

Duyarlılık (Precision): Gerçek deneysel veriler ile karşılaştırıldığında doğru tahmin edilen pozitif değerlerin doğru tahmin edilen pozitif ve negatif değerlere oranına denir [17,18].

$$Duyarlılık = \frac{TP}{TP + FP}$$

Hassasiyet (Recall): Gerçek deneysel veriler ile karşılaştırıldığında doğru tahmin edilen pozitif değerlerin sınıftaki tahminlere oranına denir [17,18].

$$Hassasiyet = \frac{TP}{TP + FN}$$

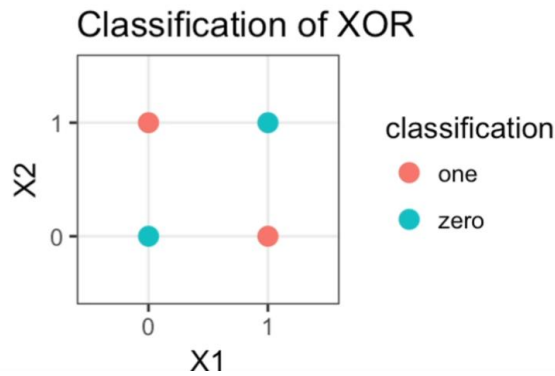
F1 ölçütü (F1 score): Hassasiyet ve duyarlılığın harmonik ortalamasıdır [17,18].

$$F1 \text{ ölçütü} = \frac{2 \times Duyarlılık \times Hassasiyet}{Duyarlılık + Hassasiyet}$$

3.2.2 Makine Öğrenmesinin Tarihçesi

Makine öğrenmesi ilk kez 1956 yılında Cornell Üniversitesinde akademisyen olan Frank Rosenblatt tarafından ortaya atıldı [19]. Frank Rosenblatt perceptron adında bir algoritma oluşturdu ve makalesinde bu algoritmadan bahsederek makine öğrenmesi ile ilgili başlangıç gerçekleştirmiş oldu. Perceptron günümüzde kullandığımız makine öğrenmesi algoritmaların çekirdeği olarak kabul edilir ve halen de kullanılmaktadır. Frank Rosenblatt, makalesinde kendi fikri olma yeteneğine sahip ilk makineden bahsetmiştir ve bu çok büyük bir heyecan uyandırmıştır. 1969 yılında Marvin Lee Minsky ve Seymour A. Papert adında iki akademisyen “Perceptrons: An Introduction to Computational Geometry” adında bir kitap yazdılar [20]. Bu kitabın içeriği 1957 yılında Frank Rosenblatt’ın oluşturduğu perceptron algoritmasının yeterli olmadığı yönündeydi. Bu konu ile ilgili XOR sınıflandırması (classification of XOR) deneyi yapılmıştı. Bir XOR (exclusive OR gate), yalnızca her iki girişi de birbirinden farklı olduğunda çıkış olarak 1 veren, aksi durumlarda 0 veren dijital bir mantık kapısıdır. Deneyde Şekil 3.3’te görüldüğü gibi kırmızı ve mavi renkli noktaların çapraz olarak konumlandığı bir diyagramda sınıflandırma yapılması istenmektedir. Yani, mavi renkli noktaların bir tarafa, kırmızı renkli noktaların da diğer tarafa toplanması istenilmektedir. İnsan beyni için çok kolay olan bu istenen durum için perceptrons

algoritması tarafından başarılı bir sonuç elde edilemedi. Perceptron algoritması bu diyagram için ortaya bir çizgi çizerek üst tarafta mavi renkli noktaları, alt tarafta kırmızı renkli noktaları gösteren bir yapı bulamadığından başarılı olamadı. İnsan beyninin kolaylıkla yapacağını, perceptron algoritmasının yapamadığı bu deneyle ispatlanmış oldu. O yıllarda doğru olan bu deney aslında yanlıştı. Çünkü perceptron algoritmasında bu noktaları ayırma yöntemi vardı. Çok katmanlı perceptronlarla bu ayırım çok kolay yapılabilirdi. Fakat o yıllarda bu yöntem bilinmediğinden yapay zekâ çalışmaları askıya alındı. Bu dönem yaklaşık 20 yıl kadar sürdü. Bu döneme yapay zekâ kışı (AI winter) denildi. Bu dönem içerisinde yapay zekâ çalışmaları için hiçbir üniversite ve sanayi kuruluşu fon ayırmadı.



Şekil 3.3: XOR sınıflandırması

1990 yıllarında yapay zeka tekrar gündeme geldi ama adı yapay zeka olarak değil de makine öğrenmesi (machine learning) olarak değiştirilmişti. Bunun sebebi geçmişteki olumsuzluğu gidermek ve araştırmalar için yeniden fon bulabilmektir. Makine öğrenmesi ile ilgilenen akademisyenler insan odaklı gitmek yerine bilgisayar odaklı gitmeyi tercih ettiler. O yıllarda yapay zeka üzerine çalışan bilim adamları algoritmalarını insan zihni ve psikolojisi üzerine kurmuşlardı, makine öğrenmesi üzerine çalışan bilim insanları ise algoritmalarını daha çok bilgisayar odaklı, istatistik ve optimizasyon üzerine kurmuşlardı. Aslında yapay zeka ile makine öğrenmesi arasındaki temel fark da tam anlamıyla budur. 1994 yılında IBM'in Thomas J. Watson Araştırma Merkezinde Gerald Tesauro adında bir araştırmacı tavla oyunu için makine öğrenmesi algoritması geliştirdi. Adını da TD-Gammon olarak belirledi. Bu algoritma

kendi kendisine karşı 100.000 defa oynadı. Tesauro algoritmaya sadece oyunun kurallarını öğretti. Algoritma kendi kendisiyle oynayarak tavla oynamayı iyileştirdi. Dünya tavla şampiyonu ile bir maç yaparak dünya tavla şampiyonunu yendi. Bu olay sonrası yapay zeka tekrar gündeme geldi. Tavla oyununa yapılan uygulama satranç oyunu için de yapıldı. 1997 yılında IBM'in deep blue adını verdiği algoritma ortaya çıkmış ve bu algoritma satrançta dünya şampiyonu olan Garry Kasparov'u yenmiştir.

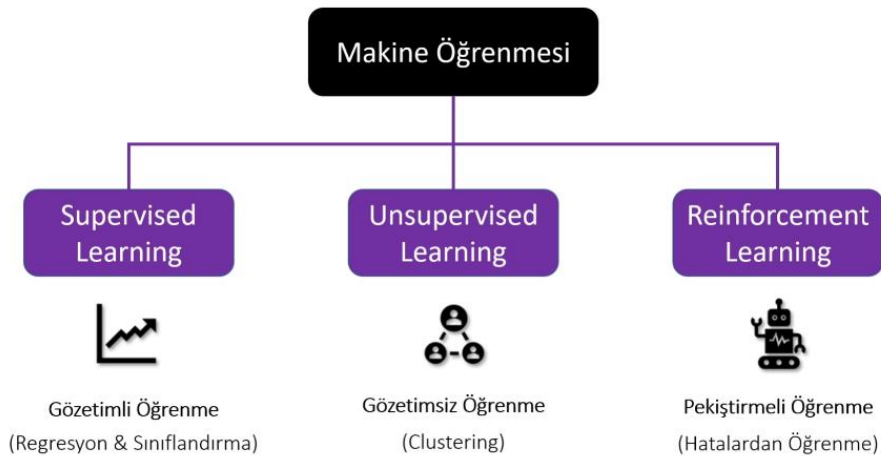
2011 yılında Amerika'da Jeopardy adlı bilgi yarışmasında IBM şirketinin üretmiş olduğu ve Watson adını verdiği yapay zeka programı, bilgi yarışmasında uzun yıllardır en başarılı olmuş iki yarışmacıya karşı yarışmış diğer yarışmacılarla arasında ciddi bir fark oluşturarak yarışmayı kazanmıştır.

2016 yılında DeepMind adında yeni kurulmuş bir şirket, makine öğrenmesi üzerine AlphaGo adlı bir yazılım geliştirmiştir. Çinlilerin milli oyunu olan Go oyununu defalarca kendi kendisiyle oynayarak öğrenme sürecini tamamladıktan sonra dünya Go oyunu şampiyonu Lee Sedol ile oynamış ve Sedol'u yenmiştir.

2019 yılında yine Deep Mind şirketinin geliştirdiği Alpha Star isimli bilgisayar programı bilinen en iyi starcraft oyuncularını ile bu oyunu oynamış ve hepsini yenmiştir.

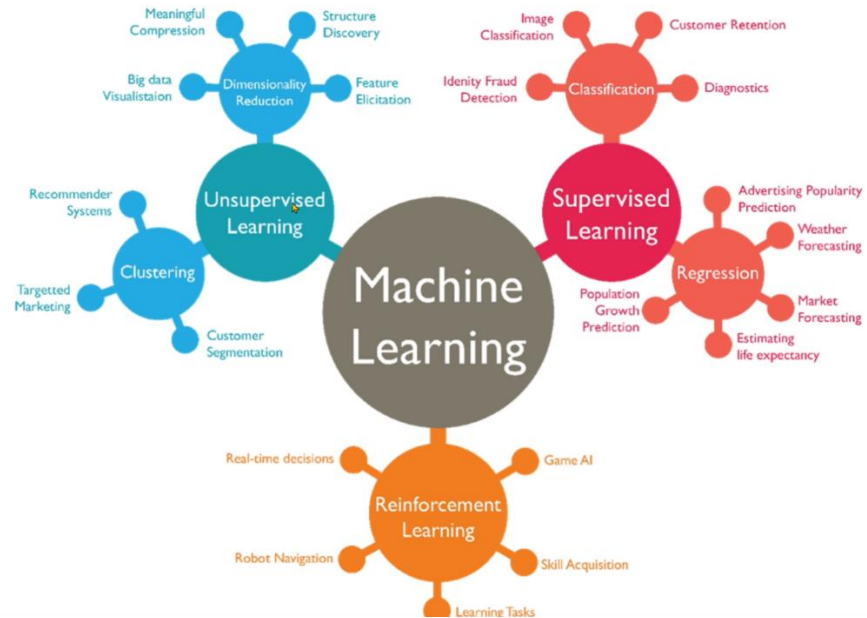
3.2.3 Makine Öğrenmesinin Türleri

Makine öğrenmesi türleri Şekil 3.4'de gösterildiği gibi genelde üç ana gruba ayrılmaktadır [21].



Şekil 3.4: Makine öğrenmesi türleri [21]

- **Gözetimli Öğrenme (Supervised learning):** Genelde makine öğrenmesinin büyük bir kısmı bu makine öğrenme türünü kullanmaktadır. Sonucu tanımlanmış veri kümeleri üzerinden(ham veri listesi) bilinmeyenleri tahmin etmeye çalışan makine öğrenmesi türüdür. Elimizde fiilen testi gerçekleşmiş verilerin sonuçlarına istinaden öğrenme sürecini tamamlayıp, test listesinde olmayan verinin sonucunun tahmin edilme sürecidir. (Şekil 3.5) [21]
- **Gözetimsiz Öğrenme (Unsupervised learning):** Bu makine öğrenmesi türünde elimizde veri kümeleri mevcut fakat sonuç bilinmemektedir. Veriler arasındaki ilişkiler üzerinden kümeleme (clustering) oluşturulur. Doğru bir cevap aranmamaktadır. Genelde biyolojide gen araştırmalarında, genlerin hangi diğer genlerle ilişkili olduğu çalışmalar bu şekildedir. Diğer bir örnek vermek gerekirse bu makine öğrenmesi türüne; Google news'in haberleri otomatik olarak gruplandırmasıdır. (Şekil 3.5) [21]
- **Pekiştirmeli Öğrenme (Reinforcement learning):** Sonucu tahmin ettikten sonra bu sonucun gerçekten doğru olup olmadığını bize söyleyen bir geri bildirim (feedback) mekanizmasıdır. Bu makine türüne örnek verecek olursak robotun yürümeyi öğrenmesi ve oyun oynamasıdır. (Şekil 3.5) [21]



Şekil 3.5: Makine öğrenmesi türleri genel şeması [21]

3.2.4 Makine Öğrenmesi Algoritmaları

Günümüzde en çok kullanılan makine öğrenmesi algoritmalarını aşağıdaki gibi sıralayabiliriz [21-24].

- Doğrusal Regresyon (Linear Regression)
- Lojistik Regresyon (Logistic Regression)
- K- En Yakın Komşu (K-NN: K-Nearest Neighbor)
- Destek Vektör Makineleri (Support Vector Machines)
- Naive Bayes
- Karar Ağaçları (Decision Trees)
- Rassal Orman (Random Forest)
- Bagging ve Boosting
- Doğrusal Ayırt Edici Analizi (Linear Discriminant Analysis)
- Boyut İndirgeme (Dimensionality Reduction)
- Çapraz Doğrulama (Cross Validation)

3.2.4.1 Doğrusal Regresyon

Doğrusal regresyon, x ve y arasında doğrusal bir ilişki olduğunu kabul eden en eski algoritmalarından bir tanesidir. Bu algoritma ile ilgili matematiksel hesaplamalar aşağıdaki formüllerle açıklanmıştır [25].

Gerçek ilişki:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

y = Sonuç değeri (Output)

β_0 = bilinmeyen 1

β_1 = bilinmeyen 2

x = Giriş değeri (Input)

ε = artık hata

Tahmini (herhangi bir x_i değerinde):

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

\hat{y}_i = Herhangi bir i noktasındaki tahmini sonuç değeri

\bar{y} = Ortalama sonuç değeri

\bar{x} = Ortalama giriş değeri

Burada x 'ler, yani girdiler, bilindiğine göre, tahmin etmeye çalıştığımız değişkenler β_0 ve β_1 'dir.

Hesap, en küçük kareler yöntemi ile; yani toplam hata karelerini hesaplayarak yapılır.

ε , bilinen değer ile tahmini değer arasındaki fark hatayı vermektedir.

$$\varepsilon_i = y_i - \hat{y}_i$$

RSS: Residual Sum of Squares (Artık kareler toplamı)

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_n^2$$

$$RSS = (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \dots + (y_n - \hat{y}_n)^2$$

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \dots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

En Küçük Kareler Yöntemi, RSS'yi minimize ederek, en iyi β_0 ve β_1 değerlerini tahmin eder.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \hat{x})(y_i - \hat{y})}{\sum_{i=1}^n (x_i - \hat{x})^2}$$

$$\hat{\beta}_0 = \hat{y} - \hat{\beta}_1 \hat{x}$$

Burada:

$$\bar{y} = \frac{1}{n} \sum_{i=0}^n y_i$$

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i$$

Modelin doğruluğunu ölçmek için,

TSS: Total Sum of Squares (Toplam Kareler Toplamı)

En kötü tahmin \bar{y} (düz çizgi)

Toplam hata kareleri: en kötü tahminin hatası

$$TSS = \sum_{i=1}^n (y_i - \bar{y})^2$$

R^2 : Tahmin katsayılarımızın ne kadar doğru sonuç verdiğini, yani çizdiğimiz regresyon doğrusunun gerçek veriye ne kadar iyi uyduğunu R^2 ile ölçeriz.

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

Yani R^2 size verideki değişimi ne kadar karşıladığınızı söyler. R^2 ne kadar büyükse (1'e ne kadar yakınsa) tahminimiz o kadar başarılı demektir. R^2 'ye tahmin katsayısı (coefficient of determination) denir.

3.2.4.2 Lojistik Regresyon

Her ne kadar adında regresyon kelimesi geçse de, lojistik regresyon bir regresyon uygulaması değildir. Lojistik regresyon aslında bir sınıflandırma (classification) uygulamasıdır. Algoritmalar bilindiği üzere temelde ikiye ayrılır. Regresyon ve sınıflandırmadır. Lojistik regresyon, sınıflandırma problemlerine genelde ilk uygulanan sınıflandırma algoritmasıdır. Amacı, farklı sınıflar (kategoriler) içinden tahmin yapmaktır [14,21,25]. Bu algoritma ile ilgili matematiksel hesaplamalar aşağıdaki formüllerle açıklanmıştır.

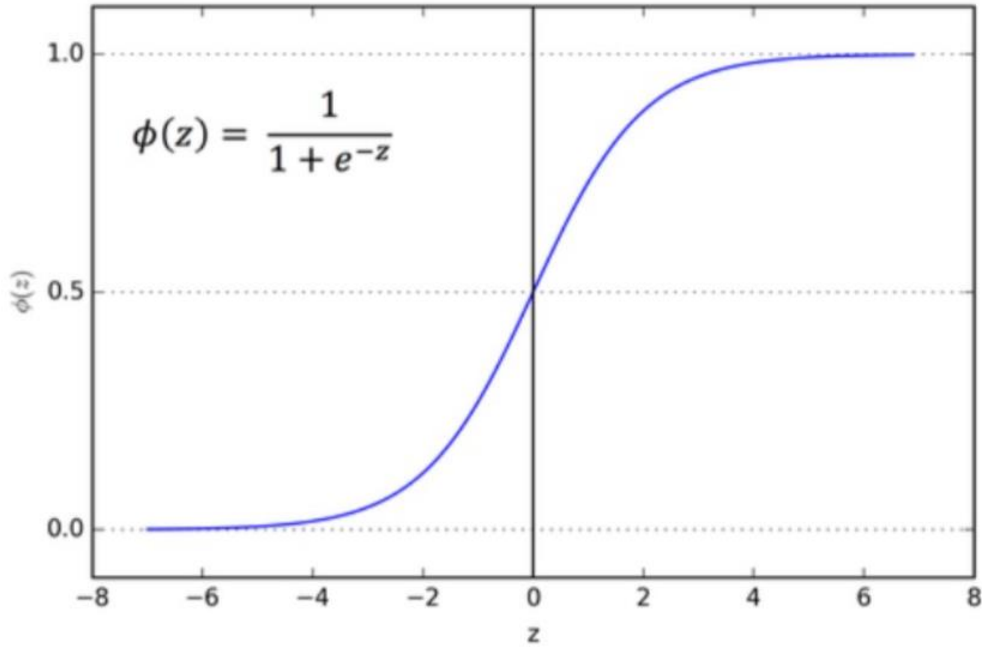
$$z = \beta_0 + \beta_1 x$$

Eğer birden çok girdi değişkeni olsaydı elimizde;

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Yukarıdaki lineer (doğrusal) eşitlik sonucu bulduğumuz z değeri (tahmin değeri) şu anda normal bir sayısal büyüklüktür. Ama bizim amacımız bir olasılık hesaplamak, o zaman z 'yi 0 ile 1 arasında bir olasılığa çevirmemiz gereklidir. İşte bu çevirme işini yapmak için sigmoid fonksiyonunu kullanacağız. Sigmoid fonksiyonu (ϕ), bir reel sayıyı bir olasılığa dönüştüren özel bir fonksiyondur.

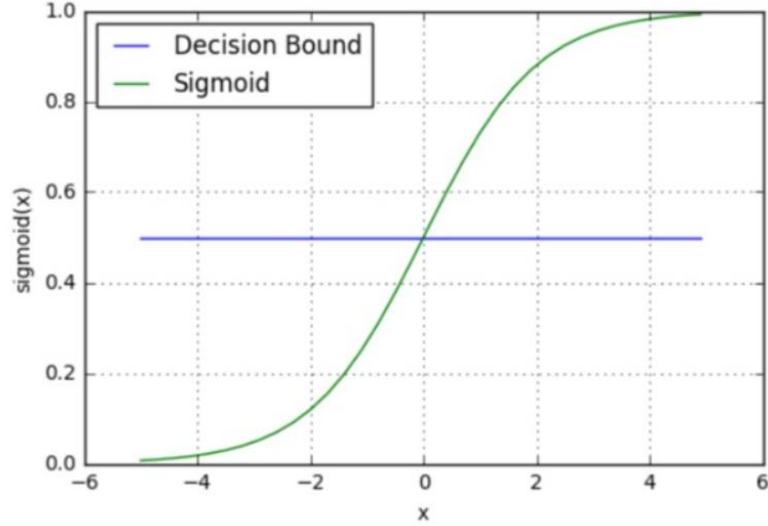
$$\phi(z) = \frac{1}{1 + e^{-z}}$$



Şekil 3.6: Sigmoid fonksiyonu grafiği 1 [25]

Şekil 6'da da görüldüğü gibi, Sigmoid fonksiyonu bir reel sayı olan z 'yi alır ve onu 0 – 1 arasında bir olasılığa götürür. Lojistik regresyonun sınıflandırmaya dönmesi ise; lojistik regresyon 0 ile 1 arasında bir sonuç getireceği için tam ortadan, yani 0.5'ten bir çizgi çizersek: çizginin altında kalanları 0 (yani class 0) çizginin üstünde kalanları 1 (yani class 1) olarak sınıflandırabiliriz. Şekil 3.7'de de gösterildiği gibi buradaki 0,5

değerine karar sınırı (decision boundary) denir. Bu değer mutlaka 0,5 olmak zorunda değildir. Bu değer 0,8 veya 0,4'de seçilebilir ama genellikle 0,5 değeri çok tercih edilmektedir.



Şekil 3.7: Sigmoid fonksiyonu grafiği 2 [26]

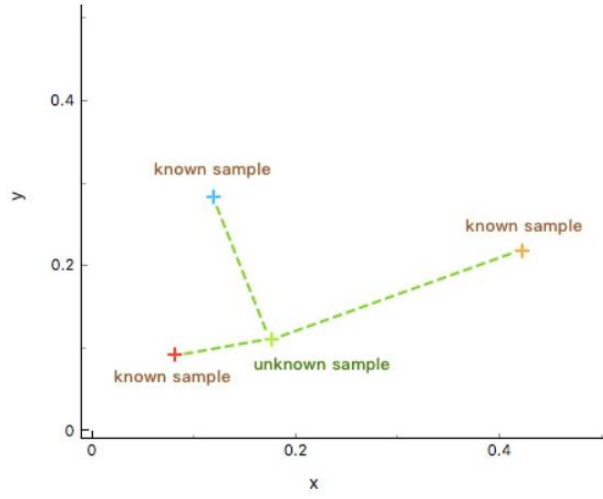
Lojistik regresyon varsayımlarını (assumptions) aşağıdaki gibi sıralayabiliriz; [27]

- Sonuç değişkeni (label, class) kategoriktir.
- Gözlemler (veriler) birbirinden bağımsızdır.
- Girdiler (değişkenler) arasında ilgileşim bulunmaz.
- Doğrusal bir ilişki ile z değeri hesaplandığı için x ile z arasında doğrusal bir ilişki olduğunu varsayar.
- Lojistik regresyonun başarılı olması için yüksek miktarda veri lazımdır. Yani çok fazla veriye ihtiyaç duyar. Doğrusal ilişki olduğu için az veri ile çok büyük oranda yanılma riski vardır.

Lojistik regresyon çıktıları ise; 0 (sıfır) ve 1 (bir), A ve B, Evet ve Hayır, Kabul ve Rettir.

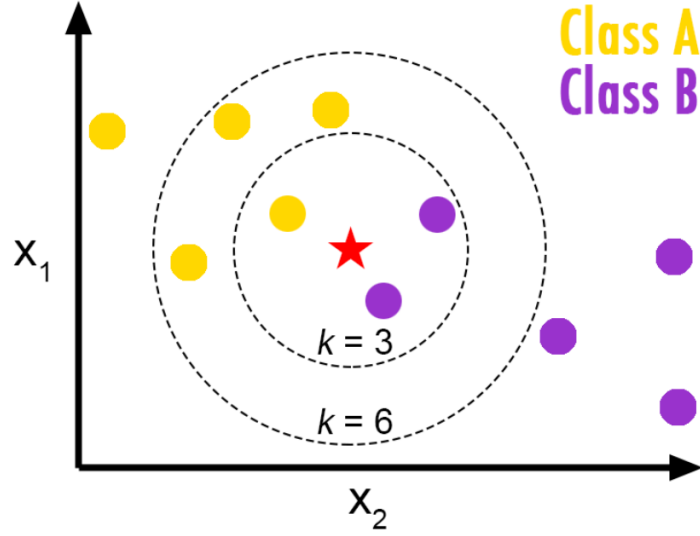
3.2.4.3 K-En Yakın Komşu

K- En Yakın Komşu (K-NN: K-Nearest Neighbor) sınıflandırma problemlerini çözmek için kullanılan basit ama etkili bir yöntemdir. K-NN'e geçmeden önce aslında, En Yakın Komşu (NN: Nearest Neighbor) algoritmasını anlamamız lazım. NN'de amaç bilinmeyen bir veri için, kendisine en yakın komşuyu bulmaktır. Bu en yakın verinin sonucu ne ise bilinmeyen için de o sonucu tahmin eder [27].



Şekil 3.8: En yakın komşu grafiği [27]

Şekil 3.8'de görüldüğü gibi, bilinmeyen noktaya en yakın nokta kırmızı renkli olduğu için, bilinmeyen noktanın rengi kırmızı olarak kabul edilir. K-NN algoritması aslında NN'nin genelleştirilmiş halidir. Burada artık tek (en yakın olan) komşuya değil de K adet komşuya bakıp karar verilir [14,21].



Şekil 3.9: K-En yakın komşu grafiği [14,21]

Şekil 3.9’da K adet komşu içinde var olan sınıfların oranlarını alırız ve en yüksek orana sahip olan sınıfı seçeriz.

- $K = 3$ için seçilecek sınıf $Mor \geq 2/3$
- $K = 6$ için seçilecek sınıf $Sarı \geq 4/6$

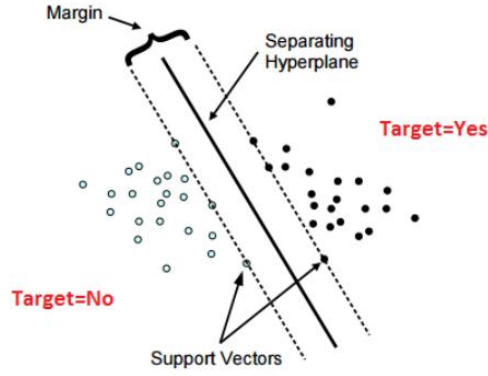
K-NN uygulaması: ilk önce veriler hazırlanır, hazırlanan veriler arası mesafeler hesaplanır, mesafeler hesaplandıktan sonra nasıl bir yöntemle karar verileceği belirlenir. İki karar verme yöntemi vardır, bunlardan birincisi çoğunluk yöntemi (sayıca en fazla olan sınıf kazanır) genelde bu yöntem kullanılır, diğeri ise ağırlıklandırma yöntemi (sınıflara ağırlık verilerek oranlar hesaplanır), yani veriler arası komşuların bulunması, bu komşulara göre sınıflandırma yapılması olarak tanımlanır [14,21].

3.2.4.4 Destek Vektör Makinesi

Destek Vektör Makinesi, en önemli ve en güçlü gözetimli öğrenme tekniklerinden biridir ve çok yaygın kullanılır. Destek Vektör Makinesi, hem sınıflandırma hem de regresyon için kullanılabilir ama yaygın olarak sınıflandırma için kullanılır [14]. Destek Vektör Makinesi ilk defa 1963'te Vladimir N. Vapnik ve Alexey Ya. Chervonenkis tarafından geliştirilmiştir. İlk zamanlar sadece doğrusal sınıflandırma

için kullanılmıştır, yani sadece doğrusal hiperdüzlemler ile çalışıyordu. 1993'te Corinna Cortes ve Vapnik tarafından “kernel trick” kullanılarak artık hem doğrusal hem de doğrusal olmayan sınıflandırma yapabilir hale getirildi ve bir anda çok popüler oldu [14,21]. Veri seti içindeki farklı sınıfları birbirinden ayıran bir ayırma çizgisi vardır. Bu ayırma çizgisine karar çizgisi (decision boundary) ismi verilmiştir. Şekil 3.10'daki gibi Destek Vektör Makinesi algoritması, noktaların bu hiperdüzleme olan uzaklıklarına göre onları sınıflandırır. Bu hiperdüzleme maximum margin hiperdüzlem de denir ve bu şekilde bir hiperdüzlem kullanan algoritmalara da “maximum margin classifier” adı verilir [14]. Destek Vektör Makinesi de bir “maximum margin classifier”. Burada hiperdüzleme en yakın olan veri noktalarına “destek vektörleri” adı verilir. Bu noktaların destek vektörleri ile uzaklık hesaplanır ve noktanın sınıfına karar verilir. Hiperdüzlemin iki yanındaki sınıflarla arasındaki mesafeye margin denir. Başka bir deyişle margin, iki sınıf arasındaki uzaklıktır. Sınıfların çizgileri destek vektörleri üzerinden geçer. Destek Vektör Makinesinin amacı bu margini maximize etmek ve veri seti içindeki en yüksek margin'e sahip hiperdüzlemi bulmaktır. Bu hiperdüzlem öyle olmalıdır ki farklı sınıflara ait destek vektörleri arasındaki mesafe maximum olsun; yani, aslında sınıfları en iyi ayıracak düzlem ya da eğrilemi (eğrisel düzlem) arar [14,21]. Bu arama işini şu iki adımda yapar,

- Sınıfları ayıran olası tüm hiperdüzlemleri hesaplar. (Sınıfları ayıran birden çok hiperdüzlem olabilir) Bunlar içinden margini en büyük olanı alır.
- Dolayısıyla seçtiği hiperdüzlem, kendisi ile destek vektörleri arasındaki mesafenin en büyük olduğu hiperdüzlemdir. (Eğer böyle bir hiperdüzlem varsa)



Şekil 3.10: Destek vektör makinesi algoritması

3.2.4.5 Naive Bayes

Naive Bayes, Bayes teoremi üzerine kurulu bir tahminleme yöntemidir. En basit gözetimli öğrenme metodlarından biridir. Basitliğine rağmen, hem çok hızlı hem de bazı problem türlerinde yüksek doğruluk oranına sahiptir. Naive Bayes, girdi içindeki değişkenlerin (features) bağımsız olduğunu varsayar. ‘Naïve’ (saf) kelimesi de buradan gelir [14,21].

Bayes teoremi:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Naive Bayes formülasyonu;

$$P(y|x_1, x_2, \dots, x_p) = \frac{P(x_1, x_2, \dots, x_p|y) * P(y)}{P(x_1, x_2, \dots, x_p)}$$

Bulmaya çalıştığımız olasılık aslında x_1, x_2, \dots, x_p 'ler verilmiş iken y 'nin olasılığıdır. Bunun için tersten gidiyoruz: elimizde y varken, tek tek x_p 'lerin olasılıklarını kullanıyoruz. x_p 'ler bağımsız olduğuna göre ve bağımsız olasılıklar çarpıldığına göre:

$$P(y|x_1, x_2, \dots, x_p) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1, x_2, \dots, x_p)}$$

$P(x_1, x_2, \dots, x_p)$ değerleri veri içinde olduğu ve sabit olduğu için:

$$P(y|x_1, x_2, \dots, x_p) \propto P(y) \prod_{i=1}^n P(x_i|y)$$

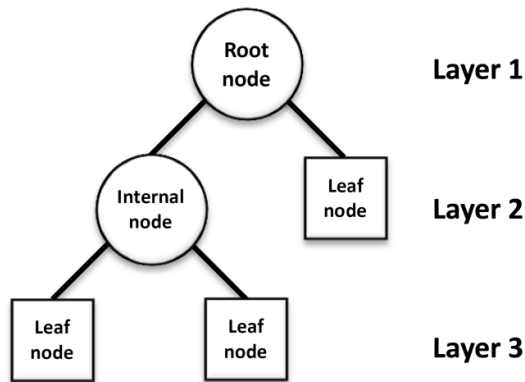
$$\Downarrow$$

$$\hat{y} = \arg \max P(y) \prod_{i=1}^n P(x_i|y)$$

Bu eşitlikte en büyük tahmin değerini, \hat{y} , veren sınıf olasılığına “maximum Aposteriori” (MAP) denir. Sonuç olarak Naive Bayes bilinmeyen bir verinin hangi sınıfa ait olduğunu tahmin etmeye çalışır (\hat{y}). Bunu yaparken, tüm sınıflar üzerinde eldeki verinin koşullu olasılığını hesaplar ve en büyük olasılık değerine ait sınıfı tahmin sınıfı olarak seçer [14,21,25].

3.2.4.6 Karar Ağacı

Karar Ağacı en önemli ve yaygın makine öğrenmesi uygulamaları arasında yer alır [14,21,25]. Ağaca benzer bir yapısı olduğu ve değişkenlerin olası kombinasyonu ile optimizasyon yaptığı için adı karar ağacı olarak geçer. Hem sınıflandırma hem de regresyon için kullanılır. Karar ağacı içinde, Şekil 3.11’de görüldüğü gibi, kök düğüm (root node), dallar (branch) ve yaprak düğümler (leaf nodes) bulundurur. Aradaki düğümlere iç düğüm (internal node) adı verilir ve her bir ara node düğüm bir durumu kontrol eder. Son düğümlere, yani artık bir alt düğümü olmayan düğümlere yaprak düğüm adı verilir.



Şekil 3.11: Karar ağacı yapısı

Karar ağaçlarının çalışma prensibi şöyledir; veri seti içindeki her bir değişken (attribute) için bir düğüm oluşturulur. En önemli değişken kök düğüme yerleştirilir. Kök düğümden başlanarak ayırma işlemi yapılır. Her ayırmada seçim kriterine bakılır. Süreç bu şekilde devam eder; bir yaprak düğüme gelip artık ayırma yapılamadığı anda süreç tamamlanır. Karar ağaçlarının en önemli kararlarından biri ayırma işlemi sırasında hangi kritere göre seçim yapılacağıdır. Genel olarak kullanılan seçim kriterleri iki tanedir. Birincisi, kategorik değişkenler için kullanılan “information gain”, diğeri ise sayısal değişkenler için Gini index’tir. Information gain kriterini kullanarak, her bir değişkenin içerdiği bilgi (information) miktarını tahmin etmeye çalışırız. Gini index ise rasgele seçilen bir veri noktasının yanlış tahmin edilmiş olma olasılığını hesaplamaya çalışır. 0 (sıfır) ve 1 (bir) arasında bir değer alır [14,21,25].

3.2.4.7 Rassal Orman

Rassal orman, yaygın olarak kullanılan bir gözetimli makine öğrenmesi algoritmasıdır. Hem sınıflandırma hem de regresyon için kullanılan türevleri mevcuttur. Çok kolay ve son derece esnektir ama bir o kadar da iyi sonuç verir. Temel olarak, verilen veri üzerinde rasgele karar ağaçları oluşturularak çalışır. Bu karar ağaçlarının her biri üzerinden bir tahmin yapar ve oylama yöntemi ile en iyi sonuca ulaşır. Rassal orman algoritması, birden çok karar ağacını birleştirir ve böylece ortaya bir orman çıkar. Rassal orman ismi de buradan gelir. Rassal ormanda ağaç sayısı arttıkça kesinlik (accuracy) değeri de artar. Rassal orman algoritması Şekil 3.12’de gösterildiği gibi iki aşamada gerçekleşir: İlk aşamada, **p** adet toplam değişken (feature) içinden **k** adedi rasgele olarak seçilir (**k**<**p**) ve bu **k** adet ile bir karar ağaçları oluşturulur [14,21].

n: toplam veri sayısı (satır sayısı)

p: toplam değişken (feature) sayısı (sütun sayısı)

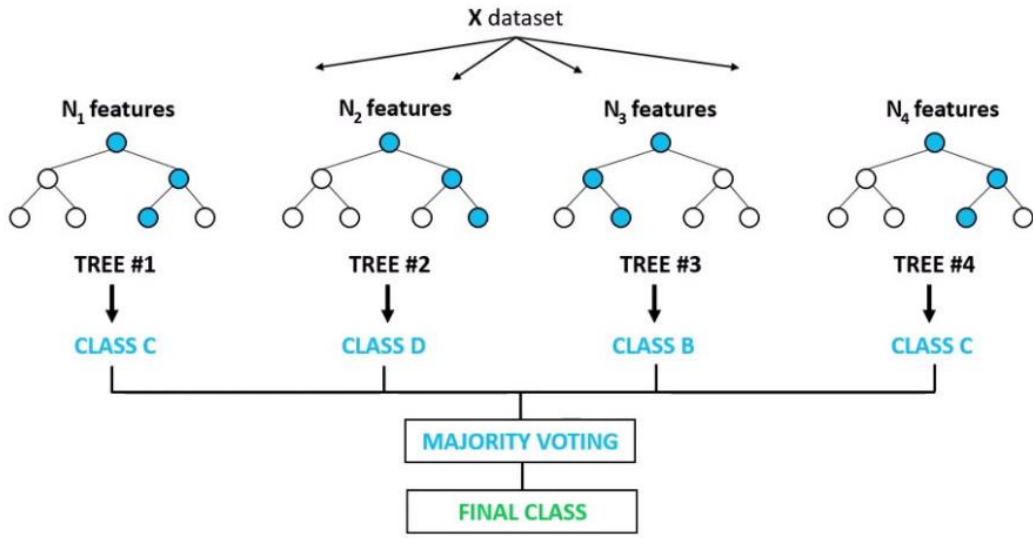
k: rasgele seçilen feature sayısı

l: bir karar ağacı içindeki maximum toplam node sayısı

m: Rassal Orman içindeki toplam karar ağacı sayısı (orman içindeki ağaç sayısı)

Toplam **p** adet değişken içinden rasgele olarak **k** adedi seçilir. Bu **k** değişken içinden en iyi ayrımı sağlayacak kök düğüm seçilir. Bu kök düğüm, alt düğümlere, onlar da

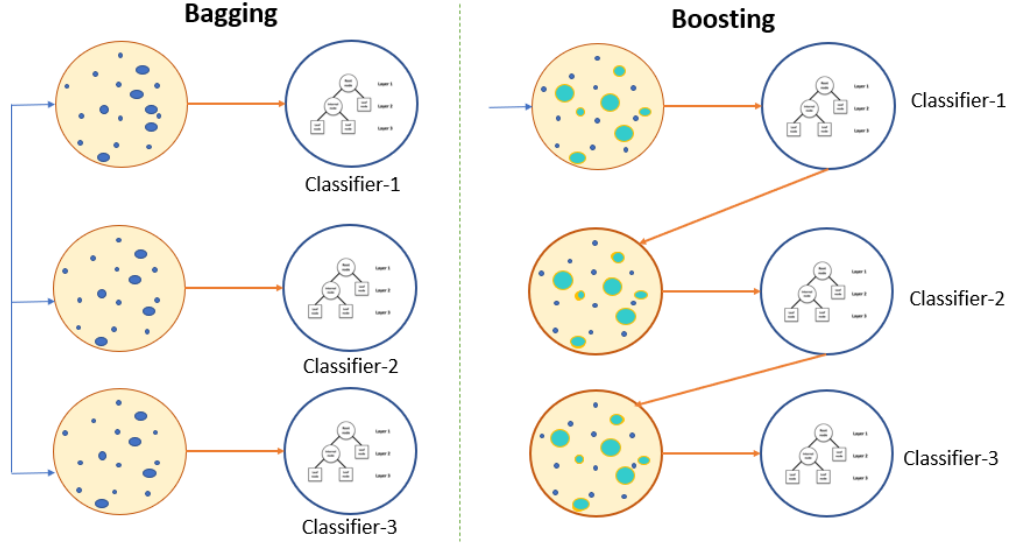
daha alt düğümlere ayırım yapılarak devam eder. Toplam bir adet düğüm elde edilene kadar adımlar tekrarlanır. Adımlar m kere tekrarlanarak, m adet karar ağacı elde edilir. İkinci aşamada ise, en iyi sonucu vermiş olan rassal orman parametreleri test veri üzerinde çalıştırılır. Sonra her bir karar ağacının vermiş olduğu sonuçlar arasında oylama yapılır. En yüksek sonucu veren aday nihai model olur.



Şekil 3.12: Rassal orman algoritmasının çalışma prensibi [25]

3.2.4.8 Bagging ve Boosting

Geleneksel olarak makine öğrenmesi uygulamaları şöyle çalışır: Lojistik regresyon, karar ağacı, destek vektör makinesi gibi bir tane algoritma seçilir, bu algoritma veri ile beslenir ve eğitilir. Eğitilen bu algoritma ile tahmin yapılır. Fakat bagging ve boosting yöntemlerinde durum bu şekilde işlemez. Bagging ve boosting yöntemlerinde bir değil, birden fazla algoritma alınır, onları güçlü algoritmalar haline getiren teknikleridir. Bagging ve boosting arasındaki fark Şekil 3.13'te gösterildiği gibidir; bagging zayıf algoritmaları paralel olarak eğitir, boosting ise zayıf algoritmaları seri olarak eğitir [21].



Şekil 3.13: Bagging ve boosting arasındaki fark [21]

Her bir model (zayıf algoritma) bir önceki modelin hatalı sınıflandırdığı veri noktalarını alır, onlara daha çok önem (katsayı) verir ve bu noktaları doğru sınıflandırmaya çalışır. Böylece her bir model bir önceki modelin hatalarını telafi eder. (hatalarından öğrenir de denilebilir). Bu şekilde model sayısı arttıkça toplam hata da azalır.

3.2.4.9 Doğrusal Ayırt Edici Analizi

Hazır verinin bileşenlerini parçalanması çok uygun olmadığında veriyi daha kolay parçalanabilir hale getirmekte kullanılır. Bu yöntemin başarılı olması için kovaryans matrisinden (covariance matrix) faydalanılır. Esasen tam anlamıyla bir sınıflandırma algoritması denilemez. Sınıflandırma işlemi öncesi, özellik çıkarımının ardından sınıflandırmak için tam anlamıyla ayırt edilmediği gözlemlendiğinde bir ön işlem gibi kullanılır. Sınıfları birbirinden ayırmak için, sınıfların dağılımını inceler ve ortalama değerleri arasındaki farklılığı kullanır [21].

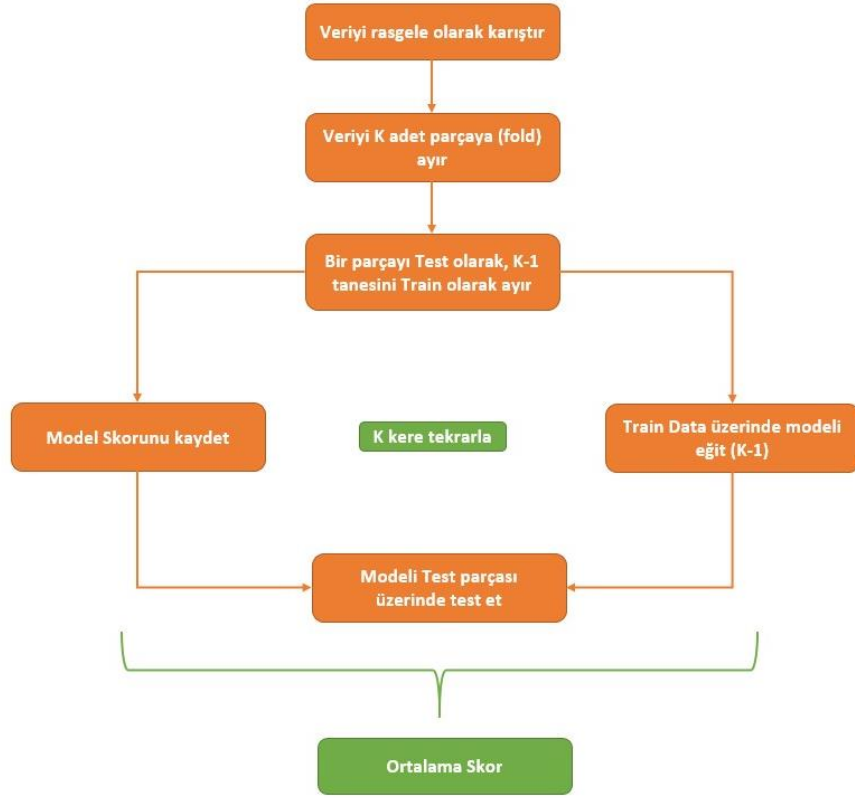
3.2.4.10 Boyut İndirgeme

PCA (Principal Component Analysis) en yaygın kullanılan boyut indirgeme tekniklerinden biridir. Temel olarak veriyi ortogonal alt uzaylara (orthogonal subspaces) yansıtarak veri içindeki en önemli değişkenlerin bulunmasını sağlar.

Böylece önemsiz değişkenler kolayca elenebilir. Bu eleme işi verilerdeki boyut sayısını azaltacağı için bu işleme literatürde boyut indirgeme denilmektedir. Veriyi yüksek boyutlu uzaydan alıp daha düşük boyutlu uzaya yansıtır. Bu yansıtma veri içindeki en yüksek varyansı sağlayan değişkenleri kullanır. En yüksek varyansı veren değişkenler veriyi en iyi temsil eden değişkenlerdir. Seçilen en yüksek varyansa sahip değişkenlere ana bileşenler (principal components) denir. İlk ana bileşen, en yüksek varyansa sahip olandır [14,21].

3.2.4.11 Çapraz Doğrulama

Çapraz doğrulama, tek bir deneme yapmak yerine bunu defalarca yaparak, mevcut problemi çözmeye çalışır. Her seferinde yeni bir deneme test verisi olacağı için, gerçek veriyi temsil yeteneği artmış olur ve iş daha az şansa bırakılmış olur. Birbirinden farklı çapraz doğrulama yöntemleri vardır. En fazla kullanılan K-fold çapraz doğrulama ve stratified K-fold çapraz doğrulama yöntemleridir. Çapraz doğrulama algoritmasının çalışma yöntemi Şekil 3.14'te de gösterildiği gibi veri setini rasgele olarak karıştırır ve karıştırılan veriyi K adet parçaya böler. Bir adet parçayı test için ayırır kalan diğer parçaları öğrenme için kullanır. “K-1” adet parça ile model eğitilir. Daha sonra başka bir parça alınıp döngü tekrarlanır. Böylece her iterasyonda farklı öğrenme-test verisi alınmış olur. K adet parçanın her biri için bu işlem yapıldığında K adet kayıp (loss) sonucu oluşmuş olur. Bu sonuçların ortalaması alınır ve ortalama kayıp sonucu bulunmuş olur [14,21].



Şekil 3.14: Çapraz doğrulama çalışma prensibi akışı [21]

Bölüm 4

Deneysel Verilerin Elde Edilmesi

4.1 Deneysel Veri Setinin Oluşturulması

Makine öğrenmesi algoritmalarının eğitilebilmesi ve test edilmesi için gerekli olan verilerin elde edilmesi için temel parametreler seçilerek 150 farklı boru bükme deneyi yapılması planlandı. Planlanan çalışmada AISI 316 kalite ostenitik paslanmaz çelik malzeme kullanıldı. Bükme yarıçapı 72,9 mm olarak belirlendi. Tablo 4.1’de belirtildiği gibi bükme parametreleri; borunun dış çapı, boru et kalınlığı, bükme açısı, kullanılacak olan malafanın bükme eksenine uzaklığı ve malafa çeşidi olarak seçildi. Bu parametrelerin değerleri ise şu şekilde belirlendi:

Borunun dış çapı: Ø20mm / Ø22mm / Ø25mm / Ø28mm / Ø30mm / Ø32mm / Ø38mm

Borunun et kalınlığı: 1mm / 1,5mm / 2mm / 2,5mm / 3mm

Bükme açısı: 30° / 45° / 60° / 75° / 90°

Malafanın bükme eksenine uzaklığı: 0mm / 3mm / 5mm

Malafa çeşidi: Topsuz düz malafa (0) / tek toplu malafa (1).

150 adet deney için belirlenen parametre değerleri bir Excel dosyasında Tablo 4.1’deki gibi listelendi. Deney sonuçlarını içeren tablo Ek A’da belirtilmiştir.

Tablo 4.1: Deneysel çalışma parametre değerleri

Deney		Malzeme Özellikleri			Uygulama Yöntemi		Sonuçlar		
Deney Adı	Deney Sıra No	Boru Dış Çapı	Boru Et Kalınlığı	Bükme Açısı	Malafanın Bükme Eksenine Uzaklığı	Malafa Çeşidi	Deney Sonucu	Gözlem	Notlar
...	1	20	1	30	0	0
...	2	20	1	30	3	1
...	3	20	1	45	5	0
...	4	20	1	45	0	1
...	5	20	1,5	45	3	0
...	6	20	1,5	60	5	1
...	7	20	1,5	60	0	0
...	8	20	1,5	60	3	1
...	9	20	2	75	5	0
...	10	20	2	75	0	1
...
...	148	38	3	90	0	1
...	149	38	3	90	3	0
...	150	38	3	90	5	1

4.2 Deneysel Çalışma

Deneysel çalışma Mutlu Metal A.Ş. de Marine bölümünde bulunan İtalyan Macri marka 6 eksenli Ø18mm'den Ø60mm'ye kadar boru bükme kapasiteli CNC boru bükme makinesinde gerçekleştirildi. Deneysel çalışma parametre değerleri listesindeki borular Siemens NX çizim programında 3D olarak tasarlandı. Daha sonra perspektif teknik çizimleri alınarak CNC boru bükme makinesinde boru bükme operasyonlarını gerçekleştirmek için boru bükme programı yazıldı ve CNC boru bükme makinesinin kontrol ünitesinde ilgili bölüme manuel olarak giriş yapıldı. Deneysel çalışma esnasında ve sonrasında herhangi bir karışıklık olmaması için numunelerin üzerine

deneysel çalışma parametre değerleri Tablo 4.2 'de gösterildiği gibi kodlama yapılmıştır. Kodlanmış bir deney numunesi Şekil 4.1'de örnek olarak gösterilmiştir.

Tablo 4.2: Deneysel veri kodlama tablosu

25	-	2	-	60	-	0	-	0	-	1
Borunun dış çapı		Et kalınlığı		Bükme derecesi		Malafanın bükme eksenine uzaklığı		Malafa çeşidi Topsuz: 0 Tek toplu: 1		Başarısız: 0 Başarılı: 1



Şekil 4.1: Deney numunesi kodlama örneği

Deneysel veri listesindeki 150 adet deneysel çalışmayı yapmak için belirlenen özellikteki ve kriterdeki borulara uygun kalıp setleri (bükme kalıbı, baskı kalıbı, sıkma kalıbı ve uygun malafa çeşidi) CNC boru bükme tezgâhına sırasıyla bağlandı; ardından bükme öncesi son ayarlar yapıldı. Burada özellikle belirtilen kriterdeki malafanın bükme eksenine uzaklığının ayarlanmasına hassasiyet gösterildi. Yapılacak her bir deney için Siemens NX programında deneysel veri listesinde belirlenen özelliklere uygun üç boyutlu tasarım gerçekleştirildi. Üç boyutlu teknik resimden imalat resimleri oluşturuldu. Oluşturulan üç boyutlu veriden CNC boru bükme makinesinde program yazmak için tasarımı yapılan borunun asal eksenlerinin kesişim koordinatları listelenip CNC boru bükme makinesinin program yazma modülüne manuel giriş yapıldı ve simülasyon yapılarak iş parçasının tezgaha çarpma riski olup olmadığına bakıldı.

Güvenli büküm işlemi yapılabileceği anlaşıldıktan sonra tezgâh büküm işlemine hazır hale getirildi. Büküm işlemi gerçekleştirilecek borunun imalat teknik resmindeki açınım ölçüsüne göre şerit testerede kesildi. Kesilen borunun kesim yüzeylerinde oluşan metal çapakları, çapak alma makinesinde uygun zımpara kullanılarak alındı. Boru bükme işlemine başlamadan önce borunun ağız taraflarına ve CNC boru bükme makinesinin malafasına malafa yağı sürülerek bükme işleminin daha kolay yapılması sağlandı. Daha sonra bükümü gerçekleştirilecek boru CNC boru bükme makinesinde bağlı olan malafaya geçirilerek tezgâha başlama emri verildi ve bükme işlemi başladı. Bükme bittiğinde tezgâh emniyetli pozisyona getirildi ve bükme işlemi gerçekleştirilmiş olan boru tezgâh üzerinden alındı. Şekil 4.2’de gösterildiği gibi bükümü gerçekleştirmiş olan borunun imalat teknik çizimine uygun olup olmadığı kalite kontrol görevlisi tarafından kontrol edildi. Bükümü gerçekleştirilmiş numunelerin kalite kontrolünde ölçüler açısından genel olarak bir problem olmadığı fakat boru bükme işlemi sonucunda istenmeyen görsel kusurlar olduğu tespit edilmiştir.



Şekil 4.2: Boru bükme işlemi sonucunun ölçüm yapılarak kontrolü

Şekil 4.3'te görsel kusuru olan bükülmüş bir numune görülmektedir. Bükme işlemi sonrasında yapılan kalite kontrol sonuçlarına göre ölçü hatası ve görsel kusuru olmayan numuneler için bükme başarılı, ölçü hatası veya görsel kusuru olan numuneler için bükme başarısız olarak kabul edilmiştir. Bükme işlemi başarılı olan deney numunelerinin üzerine çıkmaz kalemle 1 (bir), başarısız olan deney numunelerinin üzerine 0 (sıfır) değeri yazılmış ve sonuçlar ayrıca deneysel veri listesinde ilgili kısmına kaydedilmiştir. Ölçüm sonuçlarında oluşan hataların çeşitleri deneysel veri listesinin açıklama kısmında da belirtilmiştir. Aynı şekilde, deneysel veri listesinde belirtilen kriterlere göre (farklı dış çaplara ve malafanın çeşidine göre) tezgâha yeni kalıplar bağlanarak deneysel çalışmalara deney veri listesindeki sayı kadar devam edilmiştir. Böylece, deneysel çalışma tamamlandıktan sonra tüm sonuçlar deneysel veri listesine kaydedilmiştir.



Şekil 4.3: Boru bükme işlemi sonrası görsel kusur

Bölüm 5

Makine Öğrenmesinin Deneysel Verilere Uygulanması

5.1 Makine Öğrenmesi Algoritmalarının Seçilmesi

Makine öğrenmesi algoritmalarını oluşturmak için Python programlama dili kullanılmıştır. Python programlama dili diğer programlama dillerinden ayıran en önemli özellikler söz dizimi, anlam bilimi ve gelişmiş, ücretsiz kullanılan kütüphanesidir. Diğer programlama dilleri gibi derleyiciye ihtiyacı olmadığı için konuşma diline yakın bir programa dilidir. Bu nedenle öğrenilmesi kolay ve hızlıdır [28,29]. Bu programdaki numpy, pandas, matplotlib ve scikit-learn gibi hazır kütüphaneleri bu çalışmada faydalanılmıştır.

Çalışmalar “Colaboratory” ortamında yapılmıştır. Colaboratory veya kısaltılmış ismi ile “colab” bir web sayfası değildir. Tarayıcıda python programını yazmanızı ve çalıştırmanızı sağlar [28,29].

Deney sonucunda elde ettiğimiz veriler eğitim ve test verileri olarak rassal olarak ayrılmıştır. Eğitim için ayrılan veriler kullanılarak deney sonucunun tahmin edilmesinde aşağıda listelenen altı farklı sınıflandırma yöntemi kullanılmış, makine öğrenmesi algoritmaları elde edilmiştir.

1. Lojistik Regresyon
2. Karar Ağaçları
3. K-En Yakın Komşu Algoritması (KNN: K-Nearest Neighbors)

4. Rassal Orman
5. Naive Bayes
6. Destek Vektör Makinesi

Modellerin eğitilmesi için kullanılmayan test verileri ile her bir sınıflandırma algoritması ayrı ayrı test edilmiş ve test sonuçları fiziksel deney sonuçları ile kıyaslanarak yöntemin etkinliği incelenmiştir. Tüm makine öğrenmesi yöntemleri ile geliştirilen makine öğrenmesi algoritmaları için geçerli sonuçlar elde edilmiştir. Elde edilen sonuçlar doğruluk tabloları, doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan Alıcı Çalışma Özellikleri (ROC: Receiver Operating Characteristics) eğrileri ile sunulmuştur.

ROC eğrisi istatistiksel olarak bir olasılık eğrisidir. Makine öğrenme algoritmalarının performansını grafiksel olarak göstermek için kullanılır. ROC grafiğinde y eksenini pozitif doğru oranını (hassasiyet), x eksenini ise pozitif yanlış oranını belirtmektedir. Grafiğinin sol üst köşeye yaklaşması testin başarılı olduğunu yani testteki doğru oranının arttığını açıklar [30,31].

5.2 Veri Ön İşleme

Makine öğrenmesi algoritmalarının elde edilmesi için deney parametreleri; boru dış çapı, boru et kalınlığı, bükme açısı, malafanın bükme eksenine mesafesi ve malafanın çeşidi etiketleri bağımsız değişken olarak belirlenmiştir. Deneyin sonucu; deney sonucu etiketi ile bağımlı değişken olarak seçilmiştir. Bunlar Excel dosyasına kaydedilmiş ve veriler veri ön işlemesine hazır hale getirilmiştir. İki malzeme özelliği (Boru dış çapı, boru et kalınlığı) ve üçü üretim yöntemi (bükme açısı, malafanın bükme eksenine mesafesi, malafa çeşidi) olan parametreler veri kümesinin bağımsız değişkenleri olarak atanmıştır. Fiziksel deneyin ölçü açısından başarılı olması ve görsel olarak kusursuz olması durumu başarılı, diğer tüm durumlar ise başarısız olarak belirlenmiştir. Excel dosyasına kaydedilen tüm veriler pandas kütüphanesi ile data frame değişkeni tanımlanmıştır. Bağımsız değişkenler x ve bağımlı değişken y değişkenine atanmıştır. x ve y değişkenleri sklearn kütüphanesinden import edilen “train_test_split” metodu ile test büyüklüğü %33

olacak şekilde eğitim ve test verileri rassal olarak ikiye ayrılmıştır. Böylelikle 150 adet deney verisinden 100'ü modellerin eğitilmesinde, geriye kalan yani model eğitiminde kullanılmayan 50'si ise modellerin test edilmesinde kullanılmıştır. Bağımsız değişkenin eğitim verileri ve test verileri sırasıyla x_{train} ve x_{test} değişkenlerine, bağımlı değişkenin eğitim verileri ve test verileri sırasıyla y_{train} ve y_{test} değişkenlerine atanmıştır. Bağımsız değişkenin test ve eğitim değişkenleri sklearn kütüphanesinden import edilen “StandardScaler” metodu ile standart normal dağılıma göre ölçeklendirilerek x_{train} değişkeni X_{train} değişkenine ve x_{test} değişkeni X_{test} test değişkenine dönüştürülerek veri ön işleme adımı tamamlanmıştır. Böylelikle ölçeklendirilmiş test bağımsız değişkeni ve test bağımlı değişkeni kullanılarak sınıflandırma yöntemleri yardımıyla makine öğrenmesi algoritmaları geliştirilmiştir.

5.3 Lojistik Regresyon Metodunun Veri Kümesine Uygulanması

Lojistik regresyon sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden *LogisticRegression* metodu import edildi [32]. Sklearn makine öğrenmesi modelleri oluşturmak için kullanılan Python temelli bir kütüphanedir. Regresyon, kümeleme ve sınıflandırma için kullanılan pek çok öğrenme algoritmasına sahiptir. X_{train} ölçeklendirilmiş eğitim bağımsız değişkeni ve y_{train} eğitim bağımlı değişkeni ile model, `random_state=0` seçilerek ve metodun varsayılan parametreleri kullanılarak eğitildi. Model elde edildikten sonra X_{test} ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve y_{pred} değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek lojistik regresyon tahmin değerleri elde edilmiş oldu. Lojistik regresyon metodu ile elde edilen bu değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için hazırlanan karışıklık matrisi Tablo 5.1'de verilmiştir.

Tablo 5.1'de sunulan lojistik regresyon metodu ile elde edilen sonuçların karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 3'ünü yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %94 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 33'ünü doğru, 15 başarılı fiziksel deneyin 14'ünü doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.2'de sunulmuştur.

Tablo 5.1: Lojistik regresyon metodu ile elde edilen sonuçların karışıklık matrisi

	Tahmin: Başarısız	Tahmin: Başarısız	Toplam
Deney: Başarısız	33	2	35
Deney: Başarılı	1	14	15

Tablo 5.2: Lojistik regresyon metodu ile elde edilen sonuçların istatistiksel skorları

	Tahmin	Recall	F1-score
Başarısız	%97	%94	%96
Başarılı	%88	%93	%90
Makro Ortalama	%92	%94	%93
Ağırlıklı Ortalama	%94	%94	%94

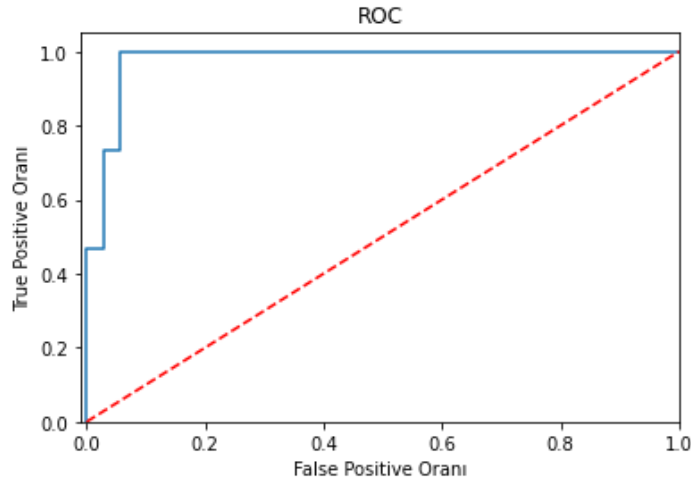
Lojistik regresyon modeli ile elde ettiğimiz her bir tahmin değeri 0 ile 1 arasındaki bir olasılık değeridir. Tahmin değeri varsayılan değer olan 0.5 den büyük olduğunda deney sonucu 1, küçük ya da eşit olduğunda 0 olarak atanmaktadır. Her bir tahmin olasılığının olasılık eşik değeri sırasıyla 0.6, 0.7, 0.8 ve 0.9 alınarak yöntemin doğruluk değeri tekrar hesaplanmış ve tahmin değerleri Tablo 5.3'deki gibi elde edilmiştir.

Tablo 5.3: Olasılık eşik değerlerinin değişimine göre lojistik regresyon metodu ile elde edilen modelin başarıları

	Olasılık Eşik Değeri				
	0.5	0.6	0.7	0.8	0.9
Model Başarı Yüzdesi	% 94	% 88	% 84	% 80	% 78

Tablo 5.3'de açıkça görüldüğü gibi olasılık eşik değeri büyüdükçe model başarı yüzdesi düşmektedir.

Şekil 5.1’de doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur. Lojistik regresyon modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı ya da başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması lojistik regresyon yönteminin başarısını ortaya koymaktadır.



Şekil 5.1: Lojistik regresyon algoritması ROC grafiği

5.4 Karar Ağacı Metodunun Veri Kümesine Uygulanması

Karar ağacı sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden *Decision Tree* metodu import edildi ve sınıflandırma yaparken entropy fonksiyonu kullanıldı. X_{train} ölçeklendirilmiş eğitim bağımsız değişkeni ve y_{train} eğitim bağımlı değişkeni ile model, metodun varsayılan parametreleri kullanılarak eğitildi. Model elde edildikten sonra X_{test} ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve y_{pred} değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek karar ağacı tahmin değerleri elde edilmiş oldu. Karar ağacı metodu

ile elde edilen bu değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için hazırlanan karışıklık matrisi Tablo 5.4'te verilmiştir.

Tablo 5.4: Karar ağacı metodu ile elde edilen sonuçların karışıklık matrisi

	Tahmin: Başarısız	Tahmin: Başarısız	
Deney: Başarısız	29	6	35
Deney: Başarılı	3	12	15

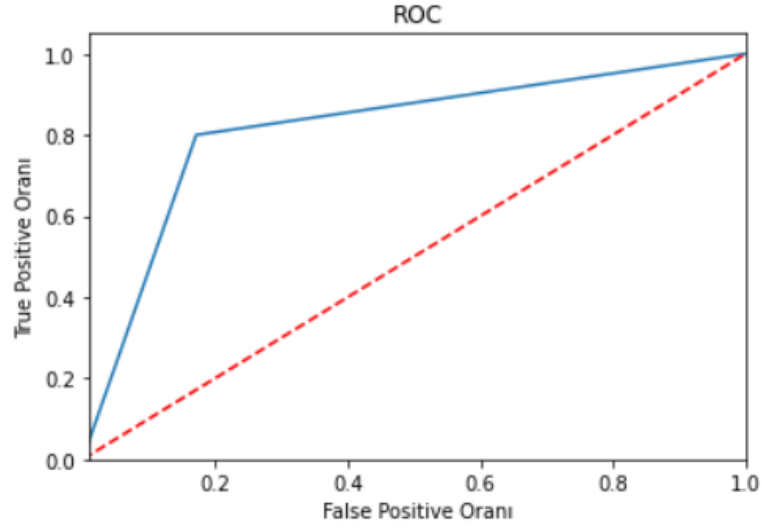
Tablo 5.4'te sunulan karar ağacı metodu ile elde edilen sonuçların karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 9'unu yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %82 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 29'unu doğru, 15 başarılı fiziksel deneyin 12'sini doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.5'te sunulmuştur.

Tablo 5.5: Karar ağacı metodu ile elde edilen sonuçların istatistiksel skorları

	Tahmin	Recall	F1-score
Başarısız	%91	%83	%87
Başarılı	%67	%80	%73
Makro Ortalama	%79	%81	%80
Ağırlıklı Ortalama	%83	%82	%82

Şekil 5.2'de doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur. Karar ağacı modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı yada başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması,

lojistik regresyon yöntemi kadar olmasa da, karar ağacı yönteminin de başarısını ortaya koymaktadır.



Şekil 5.2: Karar ağacı algoritması ROC grafiği

5.5 En Yakın Komşu Metodunun Veri Kümesine Uygulanması

En yakın komşu sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden *En yakın komşu* metodu import edildi. Komşuluk metriği *Minkowski* ve komşuluk derecesi 5 olarak tanımlandı. Model, x_{train} ölçeklendirilmiş eğitim bağımsız değişkeni ve y_{train} eğitim bağımlı değişkeni seçilerek ve metodun varsayılan parametreleri kullanılarak eğitildi. Model elde edildikten sonra x_{test} ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve y_{pred} değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek en yakın komşu algoritması tahmin değerleri elde edilmiş oldu. En yakın komşu sınıflandırma yöntemi ile elde edilen bu değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için hazırlanan karışıklık matrisi Tablo 5.6'da verilmiştir.

Tablo 5.6: En yakın komşu metodu ile elde edilen sonuçların karışıklık matrisi (komşuluk derecesi = 5)

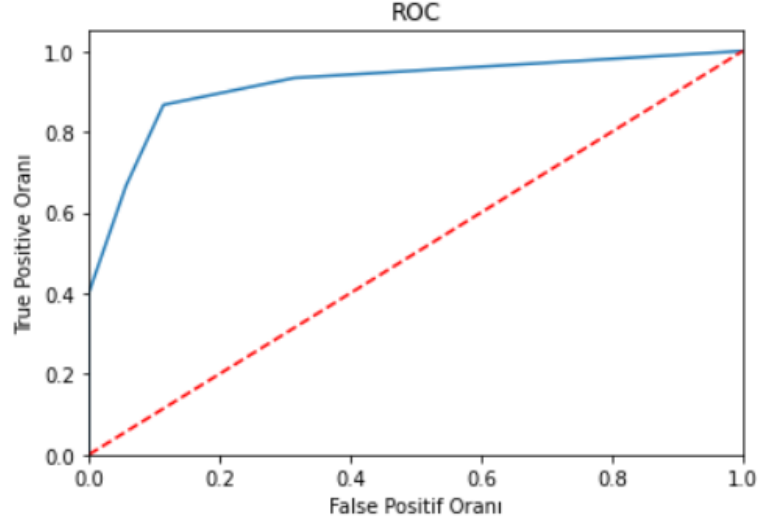
	Tahmin: Başarısız	Tahmin: Başarılı	
Deney: Başarısız	33	2	35
Deney: Başarılı	5	10	15

Tablo 5.6’da sunulan en yakın komşu sınıflandırma metodu ile elde edilen karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 7’sini yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %86 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 33’ünü doğru, 15 başarılı fiziksel deneyin 10’unu doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.7’de sunulmuştur.

Tablo 5.7: En yakın komşu metodu ile elde edilen sonuçların istatistiksel skorları (komşuluk derecesi = 5)

	Tahmin	Recall	F1-score
Başarısız	%87	%94	%90
Başarılı	%83	%67	%74
Makro Ortalama	%85	%80	%82
Ağırlıklı Ortalama	%86	%86	%86

Şekil 5.3’te doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi (komşuluk derecesi = 5 için) sunulmuştur. En yakın komşu modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı ya da başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması en yakın komşu sınıflandırma yönteminin başarısını ortaya koymaktadır.



Şekil 5.3: En yakın komşu algoritması ROC grafiği (komşuluk derecesi = 5)

Sadece komşuluk derecesi = 2 olarak değiştirerek algoritma tekrar eğitildi. Çıkan değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için karışıklık matrisi oluşturuldu. Komşuluk derecesi = 2 olarak değiştirilen verilerden elde edilen karışıklık matrisi Tablo 5.8’de verilmiştir.

Tablo 5.8: En yakın komşu metodu ile elde edilen sonuçların karışıklık matrisi (komşuluk derecesi = 2)

	Tahmin: Başarısız	Tahmin: Başarısız	
Deney: Başarısız	34	1	35
Deney: Başarılı	6	9	15

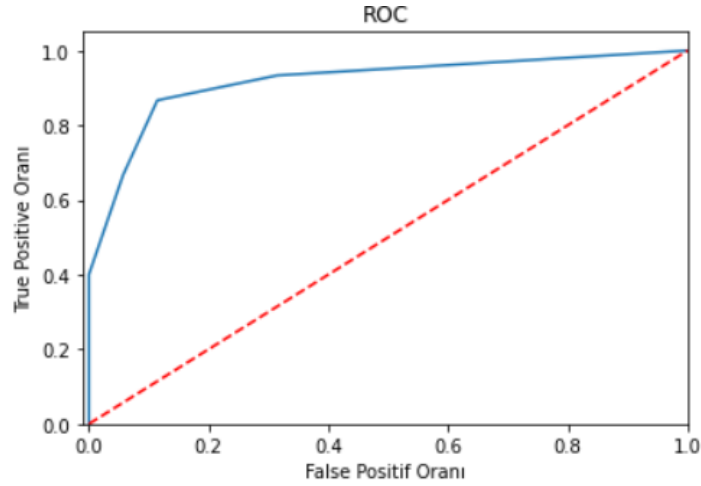
Tablo 5.8’de sunulan ve komşuluk derecesi = 2 olarak alınan en yakın komşu sınıflandırma metodu ile elde edilen karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 7 si yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %86 olduğunu göstermektedir. Ayrıca test verisi içerisindeki 35 başarısız fiziksel deneyin 34 ünü doğru, 15 başarılı fiziksel deneyin 9 ünü doğru

tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.9’da sunulmuştur.

Tablo 5.9: En yakın komşu metodu ile elde edilen sonuçların istatistiksel skorları (komşuluk derecesi = 2)

	Tahmin	Recall	F1-score
Başarısız	%85	%97	%91
Başarılı	%90	%60	%72
Makro Ortalama	%88	%79	%81
Ağırlıklı Ortalama	%86	%86	%85

Şekil 5.4’te, komşuluk derecesi = 2 olarak alınan en yakın komşu metodunda doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur.



Şekil 5.4: En yakın komşu algoritması ROC grafiği (komşuluk derecesi = 2)

5.6 Rassal Orman Metodunun Veri Kümesine Uygulanması

Rassal orman sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden Rassal orman metodu import edildi ve sınıflandırma yaparken entropy fonksiyonunu kullanıldı. X_train ölçeklendirilmiş eğitim bağımsız değişkeni ve y_train eğitim bağımlı değişkeni seçilerek ve metodun varsayılan parametreleri kullanılarak eğitildi. Model elde edildikten sonra X_test ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve y_pred değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek rassal orman metodu tahmin değerleri elde edilmiş oldu.

Tablo 5.10: Rassal orman metodu ile elde edilen sonuçların karışıklık matrisi

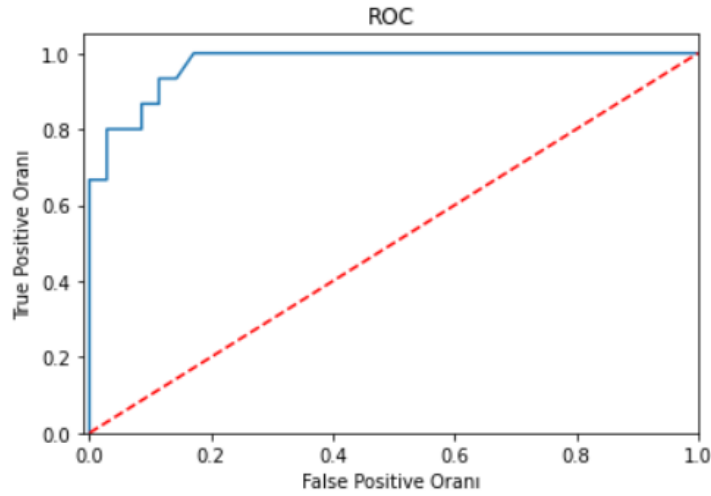
	Tahmin: Başarısız	Tahmin: Başarısız	
Deney: Başarısız	34	1	35
Deney: Başarılı	3	12	15

Tablo 5.10’da sunulan rassal orman sınıflandırma metodu ile elde edilen sonuçların karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 4’ünü yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %92 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 34’ünü doğru, 15 başarılı fiziksel deneyin 12’sini doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.11’de sunulmuştur.

Tablo 5.11: Rassal orman metodu ile elde edilen sonuçların istatistiksel skorları

	Tahmin	Recall	F1-score
Başarısız	%92	%97	%94
Başarılı	%92	%80	%86
Makro Ortalama	%92	%89	%90
Ağırlıklı Ortalama	%92	%92	%92

Şekil 5.5’de doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur. Rassal orman modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı ya da başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması rassal orman yönteminin başarısının oldukça yüksek olduğunu ortaya koymaktadır.



Şekil 5.5: Rassal orman algoritması ROC grafiği

5.7 Naive Bayes Metodunun Veri Kümesine Uygulanması

Naive Bayes sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden *Naive Bayes* metodu import edildi ve *gaussianNb* yaklaşımı kullanıldı. *X_train* ölçeklendirilmiş eğitim bağımsız değişkeni ve *y_train* eğitim bağımlı değişkeni ile varsayılan parametreleri ile eğitildi. Model elde edildikten sonra *X_test* ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve *y_pred* değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek Naive Bayes tahmin değerleri elde edilmiş oldu. Naive Bayes metodu ile elde edilen bu değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için hazırlanan karışıklık matrisi Tablo 5.12’de verilmiştir.

Tablo 5.12: Naive Bayes metodu ile elde edilen sonuçların karışıklık matrisi

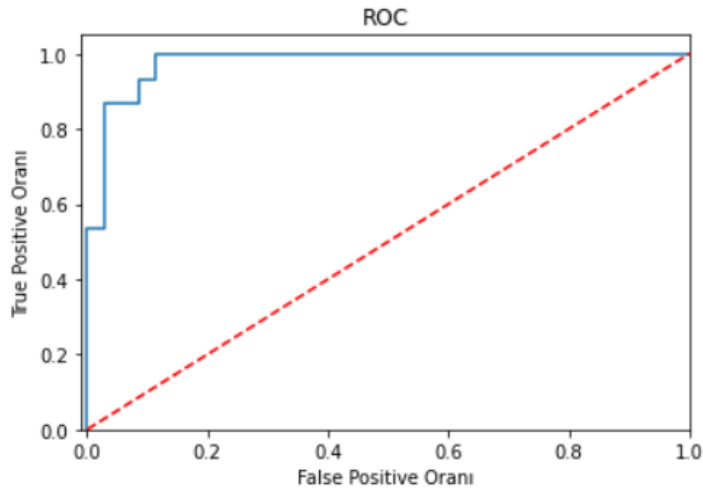
	Tahmin: Başarısız	Tahmin: Başarısız	
Deney: Başarısız	34	1	35
Deney: Başarılı	2	13	15

Tablo 5.12’de sunulan Naive Bayes sınıflandırma metodu ile elde edilen karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 3’ünü yanlış tahmin etmiştir. Bu da yöntemin doğruluk skorunun %94 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 34’ünü doğru, 15 başarılı fiziksel deneyin 13’ünü doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.13’te sunulmuştur.

Tablo 5.13: Naive Bayes metodu ile elde edilen sonuçların istatistiksel skorları

	Tahmin	Recall	F1-score
Başarısız	%94	%97	%96
Başarılı	%93	%87	%90
Makro Ortalama	%94	%92	%93
Ağırlıklı Ortalama	%94	%94	%94

Şekil 5.6’da doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur. Navie bayes modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı ya da başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması Naive Bayes yönteminin başarısını ortaya koymaktadır.



Şekil 5.6: Naive Bayes algoritması ROC grafiği

5.8 Destek Vektör Makinesi Metodunun Veri Kümesine Uygulanması

Destek Vektör Makinesi sınıflandırma yöntemi ile algoritmayı eğitmek için Python sklearn kütüphanesinden *Destek Vektör Makinesi* metodu import edildi. X_{train} ölçeklendirilmiş eğitim bağımsız değişkeni ve y_{train} eğitim bağımlı değişkeni ile metodun varsayılan parametreleri kullanılarak eğitildi. Model elde edildikten sonra X_{test} ölçeklendirilmiş test bağımsız değişkenlerinin tahmin değerleri elde edildi ve y_{pred} değişkenine atandı. Böylelikle eğitim için kullanılmayıp test verisi olarak sakladığımız 50 fiziksel deneyin verileri modele girilerek Destek Vektör Makinesi tahmin değerleri elde edilmiş oldu.

Destek Vektör Makinesi metodu ile elde edilen bu değerler ile fiziksel deneyin sonuçlarını karşılaştırmak için hazırlanan karışıklık matrisi Tablo 5.14'te verilmiştir.

Tablo 5.14: Destek vektör makinesi metodu ile elde edilen sonuçların karışıklık matrisi

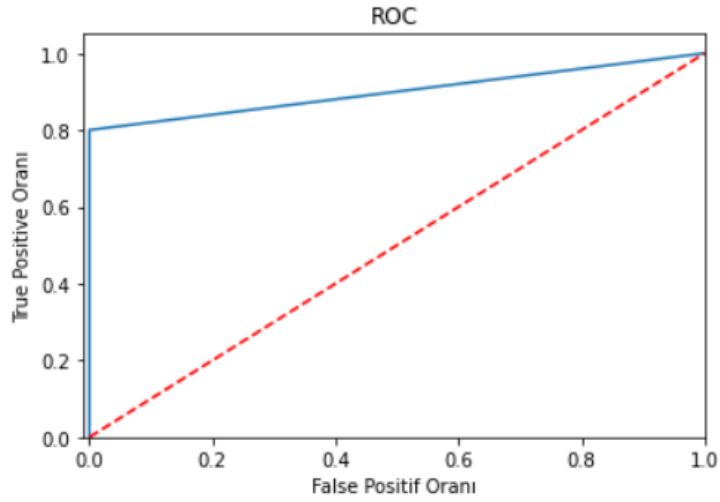
	Tahmin: Başarısız	Tahmin: Başarısız	
Deney: Başarısız	35	0	35
Deney: Başarılı	3	12	15

Tablo 5.14'de sunulan Destek Vektör Makinesi sınıflandırma metodu ile elde edilen sonuçların karışıklık matrisine göre makine öğrenmesi algoritmamız 50 test verisinden sadece 3'ünü yanlış tahmin etmiştir. Bu da Destek Vektör Makinesi yönteminin doğruluk skorunun %94 olduğunu göstermektedir. Test veri seti içerisindeki 35 başarısız fiziksel deneyin 35'ünü –yani, tamamını– doğru, 15 başarılı fiziksel deneyin 12'sini doğru tahmin etmiştir. Algoritma başarısının ayrıntılı istatistiksel skorları Tablo 5.15'te sunulmuştur.

Tablo 5.15: Destek vektör makinesi metodu ile elde edilen sonuçların istatistiksel skorları

	Tahmin	Recall	F1-score
Başarısız	%92	%100	%96
Başarılı	%100	%80	%89
Makro Ortalama	%96	%90	%92
Ağırlıklı Ortalama	%94	%94	%94

Şekil 5.7’de doğruluk test sonuçları ve hassasiyetin kesinliğe oranını belirlemek için geliştirilmiş olan ROC eğrisi sunulmuştur. Destek Vektör Makinesi modeli ile elde edilmiş olan makine öğrenmesi algoritması mavi ile, rassal tahmin değeri (hiçbir yöntem kullanmadan test verileri rasgele başarılı ya da başarısız olarak tahmin edilseydi) kırmızı ile gösterilmiştir. Mavi eğrinin kırmızı eğriden uzak olarak konumlanması Destek Vektör Makinesi yönteminin başarısını ortaya koymaktadır.



Şekil 5.7: Destek vektör makinesi algoritması ROC grafiği

Bölüm 6

Sonuç ve Değerlendirme

Bu çalışmada CNC boru bükme makinesinde bükümü gerçekleştirilecek boruların, makine öğrenmesi algoritmaları kullanılarak ilk seferde doğru olarak bükülmesi hedeflenmiştir. Bunun için CNC boru bükme makinesinde boru bükme işlemini etkileyen parametreler tanımlanmıştır. Bu parametreler esas alınarak 150 adet deneysel veri seti tablosu oluşturulmuş ve belirtilen 150 adet deneysel çalışma, yani bükme işlemi yapılmıştır. Deneysel çalışma sonuçları makine öğrenmesi algoritmalarının girdisi olarak kullanılmıştır. En yaygın olarak bilinen 6 adet makine öğrenmesi algoritması kullanılmıştır. Kullanılan algoritmalarından elde edilen sonuçların başarı oranları aşağıda verilmiştir:

1. Lojistik regresyon algoritması: %94 başarılı
2. Karar ağacı algoritması: %82 başarı
3. En yakın komşu algoritması: %86 başarı
4. Rassal orman algoritması: %92 başarı
5. Naive Bayes algoritması: %92 başarı
6. Destek vektör makinesi algoritması: %94 başarılıdır.

Bu sonuçlar değerlendirildiğinde, en düşük başarının %82 ile karar ağacı algoritmasında elde edildiği görülmektedir. En yüksek başarı ise %94 ile lojistik regresyon ve destek vektör makinesi algoritmalarında elde edilmiştir. %94'lük başarı, sektör açısından oldukça iyi bir sonuçtur.

Boru bükme işlemlerinde makine öğrenmesi algoritmaları kullanılarak zaman, işçilik maliyeti, hammadde maliyeti düşürülerek maddi kazanç sağlanacaktır.

Öte yandan, boru bükme deney sayısı artırıldığı ve bulunan sonuçlara göre bir optimizasyon yapıldığı takdirde başarının daha da yüksek olması, hatta %100'e ulaşması beklenebilir.

Bundan sonraki çalışmalarda hatasız ve kusursuz olarak bükme işlemi gerçekleştirilemeyen boruların makine öğrenmesi algoritmaları kullanılarak nasıl bükülebilecekleri konusunda çalışma yapılabilir. Bükümü gerçekleştirilemeyen boruların neden bükülemediği, bükme işleminin yapılabilmesi için hangi parametrelerin değişmesi gerektiğini ve ilave ekipman desteğine ihtiyacı olup olmadığı konusunda deneysel çalışmalar yapılabilir ve bunlar makine öğrenmesine eklenebilir.

Kaynaklar

1. Sorine, M. Formability of advanced high strength steel tubes in tube bending and hydroforming (master thesis). Waterloo: University of Waterloo; 2007.
2. Kestek, L. Sonlu elemanlar yöntemi ile boru bükme benzetimi (yüksek lisans tezi). İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Malzeme ve İmalat Anabilim Dalı; 2008. <https://tez.yok.gov.tr/>
3. Miller GG. Tube Forming Processes: A Comprehensive Guide; 2003.
4. Yavuz M. Boru bükme kalıp tasarımı ve üretim parametrelerinin incelenmesi (yüksek lisans tezi). Bursa: Uludağ Üniversitesi; 2011. <https://tez.yok.gov.tr/>
5. Fang J, Lu S, Wang K, Yao Z. Three-dimensional finite element model of high strength 21-6-9 stainless steel tube in rotary draw bending and its application. Indian Journal of Engineering & Material Sciences 2015; 22(2): 141-152.
6. Viorel P, Tabib A-K, Maier C. Bendability limits in three-roll tube bending. The Annals of “Dunarea De Jos” University of Galati, Technologies in Machine Building 2015; Fascicle V: 45-50.
7. Zahoor A. Compound cracks in pipes under combined bending and tension. International Journal of Pressure Vessels and Piping 1992; 52(3): 439-452.
8. Shr Sh-G. Bending of tubes for hydroforming: a state-of-the-art review and analysis (master thesis). Ohio: Ohio State University; 1998.
9. Pan K, Stelson KA. On the plastic deformation of a tube during bending. ASME Journal of Engineering for Industry 1995; 117(4): 494-500.
10. Yang H, Li H, Zhang Z, Zhan M, Liu J, Li G. Advances and trends on tube bending forming technologies. Chinese Journal of Aeronautics 2012; 25(1): 1-12.
11. Sandıkçı E. Boru bükümünde karşılaşılan problemler ve çözümleri (yüksek lisans tezi). Edirne: Trakya Üniversitesi; 2012. <https://tez.yok.gov.tr/>
12. Sözen L. Boru bükme operasyonu sonucunda meydana gelen geri yaylanma miktarının öngörülmesi (yüksek lisans tezi). Ankara: TOBB Ekonomi ve Teknoloji Üniversitesi; 2011. <https://tez.yok.gov.tr/>

13. Çınar A. Veri madenciliğinde sınıflandırma algoritmalarının performans değerlendirmesi ve R dili ile bir uygulama. *Öneri Dergisi* 2019; 14(51): 90-111.
14. Bishop CM. *Pattern recognition and machine learning*. Springer; 2006.
15. Tharwat A. Classification assessment methods. *Applied Computing and Informatics* 2020; 17(1): 168-192.
16. Şahin E. Makine öğrenme yöntemleri ve kümesi tekniği ile istenmeyen e-posta /e-posta sınıflaması (yüksek lisans tezi). Ankara: Hacettepe Üniversitesi; 2018. <https://tez.yok.gov.tr/>
17. Davis J, Goadrich M. The relationship between precision-recall and roc curves, in: *Proceedings of the 23rd International Conference on Machine Learning*, ACM, 2006; 233–240.
18. Powers DMW. Evaluation: From precision, recall and f-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2011; 2(1): 37–63.
19. Russel S, Norvig P. *Artificial Intelligence: A Modern Approach*, Third Edition. Pearson Series in Artificial Intelligence; 2016.
20. Rosenblatt F. *The design of intelligence automaton*. Research Trends, Cornell Aeronautical Laboratory 1958; VI(2).
21. Alpaydın E. *Machine Learning: The New AI*. The MIT Press; 2016.
22. Burgansky-Eliash Z, Wollstein G, Chu T, Ramsey JD, Glymour C, et al. Optical coherence tomography machine learning classifiers for glaucoma detection: a preliminary study. *Investigative Ophthalmology & Visual Science* 2005; 46(11): 4147-4152.
23. Öztemel E. *Yapay Sinir Ağları*. Papatya Yayıncılık, İstanbul; 2003.
24. Torkul O, Gülseçen S, Uyaroğlu Y, Çağıl G, Uçar MK. , *Mühendislikte yapay zeka ve uygulamaları*. Sakarya Üniversitesi Yayınları No:184; 2017.
25. Minsky LM, Papert SA. *Perceptron- An introduction to computational geometry*; 1998.
26. Zweig MH, Campbell G. Receiver-operating characteristics (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry* 1993; 39(4): 561-577.
27. Mitchell TM. *Machine Learning*, 1st ed. McGraw-Hill; 1997.

28. Cihan P, Coşkun H. Performance comparison of machine learning models for diabetes prediction. 29th Signal Processing and Communications Applications Conference; 2021.
29. Hacıfendioğlu Ş. Makine öğrenmesi yöntemleri ile glokom hastalığının teşhisi (yüksek lisans tezi). Konya: Selçuk Üniversitesi; 2012. <https://tez.yok.gov.tr/>
30. Ayvaz U, Çoban A, Gürüler H, Peker M. Python dilinin özellikleri, programlama eğitiminde ve yazılım dünyasındaki yeri. Muğla Sıtkı Koçman Üniversitesi, Bilişim Sistemleri Mühendisliği Bölümü, Muğla; 2016.
31. Brown CD, Davis HT. Receiver operating characteristics curves and related decision measures: A tutorial. Chemometrics and Intelligent Laboratory Systems 2006; 80(1): 24-38.
32. Altun M, Bal N, Bütüner R, Şahin SK, Koçali M. Herkes için Python programlama dili. Öğretmen Yetiştirme ve Geliştirme Genel Müdürlüğü; 2020.

Ekler

Ek A

Deney Sonuçları Listesi

Deney		Malzeme Özellikleri			Uygulama Yöntemi		Sonuçlar		
Deney Adı	Deney Sıra No	Boru Dış Çapı	Boru Et Kalınlığı	Bükme Açısı	Malafanın Bükme Eksenine Uzaklığı	Malafa Çeşidi	Deney Sonucu	Gözlem	Notlar
	1	20	1	30	0	0	0		
	2	20	1	30	3	1	1		
	3	20	1	45	5	0	1		
	4	20	1	45	0	1	0		
	5	20	1,5	45	3	0	0		
	6	20	1,5	60	5	1	1		
	7	20	1,5	60	0	0	0		
	8	20	1,5	60	3	1	1		
	9	20	2	75	5	0	1		
	10	20	2	75	0	1	0		
	11	20	2	75	3	0	0		
	12	20	2	90	5	1	1		
	13	20	2,5	90	0	0	0		
	14	20	2,5	90	3	1	1		
	15	20	2,5	90	5	0	1		
	16	20	2,5	30	0	1	0		
	17	20	3	30	3	0	0		
	18	20	3	45	5	1	1		
	19	20	3	45	0	0	0		

20	20	3	45	3	1	1
21	22	1	60	5	0	0
22	22	1	60	0	1	0
23	22	1	60	3	0	0
24	22	1	75	5	1	1
25	22	1,5	75	0	0	0
26	22	1,5	75	3	1	1
27	0	1,5	90	5	0	1
28	22	1,5	90	0	1	0
29	22	2	90	3	0	0
30	22	2	90	5	1	1
31	22	2	30	0	0	0
32	22	2	30	3	1	1
33	22	2,5	45	5	0	1
34	22	2,5	45	0	1	0
35	22	2,5	45	3	0	0
36	22	2,5	60	5	1	1
37	22	3	60	0	0	0
38	22	3	60	3	1	1
39	22	3	75	5	0	1
40	22	3	75	0	1	0
41	25	1	75	3	0	0
42	25	1	90	5	1	1
43	25	1	90	0	0	0
44	25	1	90	3	1	0
45	25	1,5	90	5	0	1
46	25	1,5	30	0	1	0
47	25	1,5	30	0	1	0
48	25	1,5	30	3	0	0
49	25	1,5	45	5	1	1
50	25	2	45	0	0	0
51	25	2	45	3	1	1
52	25	2	60	5	0	1
53	25	2	60	0	1	0
54	25	2,5	60	3	0	0

55	25	2,5	75	5	1	1
56	25	2,5	75	0	0	0
57	25	2,5	75	3	1	1
58	25	3	90	5	0	1
59	25	3	90	0	1	0
60	25	3	90	3	0	0
61	25	3	90	5	1	1
62	28	1	30	0	0	0
63	28	1	30	3	1	0
64	28	1	45	5	0	0
65	28	1	45	0	1	0
66	28	1,5	45	3	0	0
67	28	1,5	60	5	1	1
68	28	1,5	60	0	0	0
69	28	1,5	60	3	1	0
70	28	2	75	5	0	1
71	28	2	75	0	1	0
72	28	2	75	3	0	0
73	28	2	90	5	1	1
74	28	2,5	90	0	0	0
75	28	2,5	90	3	1	1
76	28	2,5	90	5	0	1
77	28	2,5	30	0	1	0
78	28	3	30	3	0	0
79	28	3	45	5	1	1
80	28	3	45	0	0	0
81	28	3	45	3	1	1
82	30	1	60	5	0	0
83	30	1	60	0	1	0
84	30	1	60	3	0	0
85	30	1	75	5	1	0
86	30	1,5	75	0	0	0
87	30	1,5	75	3	1	0
88	30	1,5	90	5	0	0
89	30	1,5	90	0	1	0

90	30	2	90	3	0	0
91	30	2	90	5	1	1
92	30	2	30	0	0	0
93	30	2	30	3	1	1
94	30	2,5	45	5	0	1
95	30	2,5	45	0	1	0
96	30	2,5	45	3	0	0
97	30	2,5	60	5	1	1
98	30	3	60	0	0	0
99	30	3	60	3	1	1
100	30	3	75	5	0	1
101	32	3	75	0	1	0
102	32	1	75	3	0	0
103	32	1	90	5	1	0
104	32	1	90	0	0	0
105	32	1	90	3	1	0
106	32	1,5	90	5	0	0
107	32	1,5	30	0	1	0
108	32	1,5	30	3	0	0
109	32	1,5	45	5	1	1
110	32	2	45	0	0	0
111	32	2	45	3	1	0
112	32	2	60	5	0	1
113	32	2	60	0	1	0
114	32	2,5	60	3	0	0
115	32	2,5	75	5	1	1
116	32	2,5	75	0	0	0
117	32	2,5	75	3	1	1
118	32	3	90	5	0	1
119	32	3	90	0	1	1
120	32	3	90	3	0	0
121	32	3	90	5	1	1
122	38	1	30	0	0	0
123	38	1	30	3	1	0
124	38	1	45	5	0	0

125	38	1	45	0	1	0
126	38	1	45	3	0	0
127	38	1	60	5	1	0
128	38	1,5	60	0	0	0
129	38	1,5	60	3	1	0
130	38	1,5	75	5	0	0
131	38	1,5	75	0	1	0
132	38	1,5	75	3	0	0
133	38	1,5	90	5	1	0
134	38	2	90	0	0	0
135	38	2	90	3	1	0
136	38	2	90	5	0	0
137	38	2	30	0	1	0
138	38	2	30	3	0	0
139	38	2	45	5	1	1
140	38	2,5	45	0	0	0
141	38	2,5	45	3	1	0
142	38	2,5	60	5	0	0
143	38	2,5	60	0	1	0
144	38	2,5	60	3	0	0
145	38	2,5	75	5	1	1
146	38	3	75	0	0	0
147	38	3	75	3	1	0
148	38	3	90	5	0	0
149	38	3	90	0	1	0
150	38	3	90	3	0	0

Özgeçmiş

Adı Soyadı: Volkan GÖRÜŞ

Eğitim:

2013–2016 Manisa Celal Bayar Üniversitesi, Metalürji ve Malzeme Müh.
Bölümü

İş Deneyimi:

2019 – 2021 Mutlu Metal A.Ş. – Marine Üretim Müdürü

2018 – 2019 Volkan İtfaiye A.Ş. – Ar-ge Merkezi Sorumlusu

2013 – 2016 Renkler Makine A.Ş. – Üretim Müdürü