



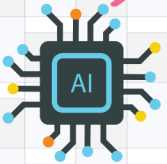
SAÜ MEZUNLAR DERNEĞİ

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 2

Sakarya
2018



YAPAY ZEKA



MAKİNE ÖĞRENMESİ



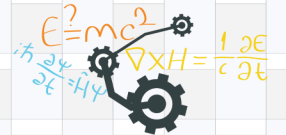
ROBOTİK



YAPAY SINIR
AĞLARI



SİBERNETİK



PROBLEM ÇÖZME

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 2

Editörler

Prof. Dr. Sevinç GÜLSEÇEN

Prof. Dr. Alpaslan FIĞLALI

Prof. Dr. Orhan TORKUL

Doç. Dr. İhsan Hakan SELVİ

Dr. Öğr. Üyesi Gültekin ÇAĞIL

Arş. Gör. Dr. Muhammed Kürşad UÇAR

SAKARYA ÜNİVERSİTESİ, MÜHENDİSLİK FAKÜLTESİ / [WWW.MF.SAKARYA.EDU.TR/](http://www.mf.sakarya.edu.tr/)

SAKARYA ÜNİVERSİTESİ YAPAY ZEKA SİSTEMLERİ UYGULAMA VE ARAŞTIRMA MERKEZİ /
WWW.YAZSUM.SAKARYA.EDU.TR

İSTANBUL ÜNİVERSİTESİ / WWW.ISTANBUL.EDU.TR

KOCAELİ ÜNİVERSİTESİ / WWW.KOCAELI.EDU.TR

Bu kitap ücretsiz dağıtılmak üzere ülkemizin geleceği için yapılmış bir hizmettir.

1. Baskı, Aralık 2018, SAKARYA

Dizgi, Arş.Gör.Dr. Muhammed Kürşad UÇAR - Sakarya Üniversitesi

Kapak, Nukeloveer Studio

ISBN, 978-605-2238-08-0

Sakarya Üniversitesi Yayınları No: 193

*Gece destan yazan, sabahında devleti için hizmet eden kahraman
milletimize ithaf olunur.*

İçindekiler

1	Parametre Optimizasyonu	11
1.1	Giriş	11
1.2	Metasezgisel Yöntem Parametreleri	14
1.3	Parametre Optimizasyonu	15
1.4	F-Race algoritması	16
1.5	Kaynakça	18
2	Kurumsal Kaynak Planlaması	25
2.1	Giriş	25
2.2	Kurumsal Kaynak Planlama Sistemleri	25
2.3	İş Süreçlerinin Modellenmesi	27
2.4	Petri-net Modelleme	29
2.5	BPMN ile Modelleme	30
2.6	BPEL ile Modelleme	32
2.7	EPC ile Modelleme	32
2.8	UML Faaliyet (Activity) Diyagramları	35
2.9	Yapay Zekâ Uygulamalarının İşletmelerde Kullanımı	36
2.10	Yapay Zekâ Uygulamalarının Modellenmesi	39
2.11	Sonuç	40
2.12	Kaynakça	41

2.13	Yazarlar Hakkında	44
3	Veri Görselleştirme	45
3.1	Giriş	45
3.2	Veri Görselleştirme	46
3.3	Görsel Analitik	48
3.4	İş Zekâsı ile Veri Görselleştirme İlişkisi	50
3.5	Görselleştirme Yazılımları	52
3.6	Tableau ile Örnek Gösterge Paneli Tasarımı	53
3.7	Sonuç	59
3.8	Kaynakça	59
4	PANDAS ile Veri Analizi	61
4.1	Giriş	61
4.2	Python Pandas Kütüphanesi Uygulamaları	64
4.2.1	Seriler	64
4.2.2	DataFrame	69
4.2.3	Veri Görselleştirme (Matplotlib Kütüphanesi)	79
4.3	Kaynakça	81
5	Tensorflow	83
5.1	Introduction	83
5.2	Background	84
5.2.1	What is Deep Learning?	84
5.3	Experiment	85
5.3.1	Tensorflow	85
5.3.2	Environment set-up	85
5.3.3	Use-case	86
5.3.4	Re-training the model	86
5.3.5	Using the Retrained Model	88
5.4	Conclusion	90
	Acknowledgement	91
5.5	Kaynakça	91
5.6	About the authors	91
6	R ile Kaba Kümeleme ve Uygulamaları	93
6.1	Giriş	93

6.2	Kaba Kümelemede Temel Kavramlar	94
6.3	Bulanık Kaba Küme Teorisi	102
6.4	'Roughsets' Yazılım Paketi	103
6.4.1	Temel Kavramların Uygulanması (BC)	105
6.4.2	Eksik Veri Tamamlama Uygulamaları (MV)	105
6.4.3	Ayrıklaştırma Uygulamaları (D)	106
6.4.4	Öznitelik Seçimi Uygulamaları (FS)	106
6.4.5	Örnek (Veri) Seçimi Uygulamaları (IS)	107
6.4.6	Kural Çıkarma Uygulamaları (RI)	107
6.4.7	Tahmin/Sınıflandırma Uygulamaları	108
6.5	Uygulama	108
6.5.1	Göğüs Kanseri Teşhisi Uygulaması	108
6.5.2	Kaba Kümeleme İle Kural Tabanlı Sınıflandırıcı	113
6.5.3	Bulanık Kaba Kümeleme İle Kural Tabanlı Sınıflandırıcı	115
6.6	Sonuçlar	118
6.7	Kaynakça	118
6.8	Yazarlar Hakkında	122
7	Lojistik ve Tedarik Zinciri Yönetimi	125
7.1	Giriş	126
7.2	Lojistik ve Tedarik Zinciri Yönetimi	126
7.3	Endüstri 4.0	127
7.4	Literatür Analizi	129
7.5	Sonuç ve Öneriler	133
7.6	Kaynakça	133
8	TWO Meta-Sezgisel Algoritma	139
8.1	Giriş	139
8.1.1	Fonksiyonun türevinin tanımsız veya çok modlu olması durumu	141
8.1.2	Karar değişkenlerinin çok olması durumu	141
8.2	Meta-Sezgisel Algoritmaların Çalışma Mantığı	142
8.2.1	Yörünge esaslı algoritmalar	142
8.2.2	Sürü esaslı algoritmalar	142
8.2.3	Gelişim esaslı algoritmalar	143
8.2.4	Meta-Sezgisel algoritmaların çözüm yaklaşımı	143
8.3	TWO Algoritması	144
8.4	TWO Algoritmasının Uygulaması	147
8.5	Sonuç	147

8.6	Kaynakça	149
9	Derin Öğrenme Uygulaması	151
9.1	Giriş	151
9.2	Derin Öğrenme Mimarileri	152
9.2.1	Kıvrımlı (Konvolüsyonel) Sinir Ağları	152
9.2.2	Derin İnanç Ağları	152
9.2.3	Derin Oto-Kodlayıcılar	153
9.3	Derin Öğrenme Uygulama Yapısı	154
9.3.1	Kıvrım (Convolution)	155
9.3.2	Bütünleştirme (Pooling)	155
9.3.3	Erken Durdurma (Early Stopping)	156
9.3.4	Düğüm Silme (Dropout)	156
9.4	Kalite Kontrol Uygulaması	156
9.4.1	Derin Öğrenme Modeli	157
9.4.2	Eğitim Süreci	162
9.5	Sonuç	162
9.6	Kaynakça	164
10	Python ile Görüntü İşlem	165
10.1	Giriş	165
10.1.1	Temel Tanımlar	166
10.2	Python ile Görüntü İşleme	166
10.2.1	Python Nedir?	166
10.2.2	OpenCV	167
10.3	İmge Bölütleme (Image Segmentation) Yöntemleri	167
10.3.1	Eşikleme (Thresholding)	168
10.4	Uygulama	173
10.5	Sonuç	186
10.6	Kaynakça	186

ÖNCE SÖZ

İşimiz vaktimizden çok deyip, bu devlete hizmette en önde gidenlerden olmak umuduyla.

Geçtiğimiz yıl vermiş olduğumuz söze kaldığımız yerden devam ediyoruz. Attığımız her adımı takip eden adımlarla hedeflerimize yavaş yavaş yaklaşıyoruz her bir birey ve devlet nezdinde. Bu yıl "Mühendislikte Yapay Zekâ ve Uygulamaları 2" kitabı ile bir seriye başlamak istiyoruz. Umarız ki bu tür hizmetler yetiştirdiğimiz öğrencilerimiz için faydalı olur ve her yıl bu kitabın devamını çıkarabiliriz.

Yapay zekâ her geçen gün önem kazanmaktadır. Rusya Devlet Başkanı Vladimir Putin, Rusya'da açılışı yapılan bir okulda, yapay zekâ teknolojisini en çok ilerleten devletin en büyük söz sahibi olacağını belirtti. Görüyoruz ki devlet bazında bir farkındalık var. Bu durum Türkiye'de yankı bulmakta ve devlet bazında çalışmalar başlamıştır. Bu yıl, 27-28 Mart 2018 tarihlerinde Ankara'da, Türk Silahlı Kuvvetleri bünyesinde "Büyük Veri, Yapay Zekâ, Otonom, Robotik, Karar Destek Sistemleri Çalıştayı" gerçekleştirildi. Görüyoruz ki ordumuzun da bu konuda çalışmaları başlamış ve Zırhlı ve Uzaktan Kontrol edilebilen Ejder Yalçın aracı, dronlar, İHA ve SİHA'lar gibi pek çok ürün görücüye çıkmaya başlamıştır. Bu çalışmaların teknolojileri üniversitelerde akademik olarak hazırlanmaktadır. Bu yüzden üniversitelerde geliştirilen teknolojilerin bir an evvel sanayiye aktarılabilmesi için çaba sarf edilmesi gerekir.

1900'lü yıllarda akademik olarak geliştirilen ürünlerin ticarileşmesi onlarca yıl sürmekteydi. Günümüzde ise ticarileşme birkaç yıl içinde gerçekleşip 2-3 yıl sonra yerine yeni ürünler geliştirilmektedir. Ticarileşme ve kullanım döngüsü giderek kısalmaktadır. Bu anlamda yapay zekâ odaklı çalışmaların ticarileşmesi ve pratikte kullanılabilme oranını Türkiye'de arttırmamız gerekir.

Elimizdeki bu kitap gerek teorik gerekse pratik uygulamalarla size yeni bir yol gösterici olmasını umuyoruz. Yapay zekâ oldukça geniş bir konudur. Zifiri karanlıkta her tarafı aydınlatamakta önümüzü görece kadar kendimize ve çevremize ışık tutmayı umuyoruz.

İşığımızın hiç kaybolmaması dileğiyle.

Editörler
Aralık 2018

Optimization

1. Parametre Optimizasyonu

Metasezgisel Yöntemlerde Parametre Optimizasyonu

Ahmet CİHAN¹, Alpaslan FIĞLALI²

¹Düzce Üniversitesi, Endüstri Mühendisliği Bölümü, Düzce

²Kocaeli Üniversitesi, Endüstri Mühendisliği Bölümü, Kocaeli

Özet

Metasezgisel optimizasyonda kullanılan yöntemler genel olarak çözüm uzayında hem global hem de yerel arama yapacak mekanizmalar içermektedir. Optimum veya optimuma yakın bir çözüm değeri elde edilebilmek için bu mekanizmaların uygun şekilde kullanılması gereklidir. Söz konusu mekanizmaların işleyişi metasezgiselle ve çözülecek problemin türüne bağlı olarak değişiklik gösteren bir takım parametre değerleri ile ilişkili olduğundan, iyi çözümlere ulaşabilmek için uygun parametre kümeleri ile çalışma zorunluluğu vardır. Bu çalışmada öncelikle optimizasyon problemlerinin çözümünde kullanılan yöntemler sınıflandırılacak ve temel bir kombinatoriyal optimizasyon problemi olan Gezgin Satıcı Problemi üzerinden çözüm yöntemlerine örnekler verilecektir. Daha sonra metasezgisel tanımı, özellikleri, sınıflandırmasından bahsedilerek; bir metasezgisel örneği olarak genetik algoritmanın çalışma prensipleri özetlenecektir. Ardından, çeşitli metasezgisellerin kullandıkları parametreler tanımlanacak ve parametre optimizasyonu için kullanılan yöntemler hakkında bilgi verilecektir.

Anahtar Kelimeler: Parametre Optimizasyonu, Metasezgisel Yöntemler

1.1 Giriş

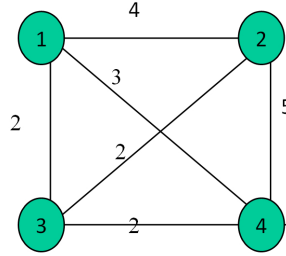
Sürekli veya kesikli optimizasyon problemlerinin çözümü için kullanılan yöntemler genel olarak aşağıdaki gibi sınıflandırılmaktadır:

1. Kesin Çözüm Yöntemleri
2. Sezgisel Çözüm Yöntemleri

3. Metasezgisel Çözüm Yöntemleri

Çözüm yöntemleri arasındaki farkları vurgulamak amacıyla söz konusu yöntemlerin en temel kombinatoriyal optimizasyon problemlerinden birisi olan Gezgin Satıcı Probleminin çözümü için nasıl uygulandığından bahsetmek yararlı olacaktır.

Gezgin satıcı problemi, Belirli bir başlangıç düğümünden çıkan, tüm diğer düğümlere uğrayan ve başlangıç noktasına dönen en kısa uzunluklu turun bulunması problemi olarak tanımlanabilir. Bağ uzunluklarının simetrik olmadığı durumlarda tüm çözüm seçeneklerinin listelenmesi (complete enumeration) istenirse $(n-1)!$ alternatif söz konusu olacaktır. Simetrik ağlarda ise $(n-1)!/2$ çözüm seçeneği söz konusu olmaktadır. Dolayısıyla Şekil 1.1'de verilen 4 düğüm ve 8 bağdan oluşan şebekede elde edilebilecek tüm yol seçeneklerinin sayısı 6'dır. Şekilde bağların üzerinde görülen değerler yol uzunluklarını ifade etmektedir. Örneğin 1 numaralı düğümünden başlamak koşuluyla elde edilen seçeneklerin listesi, yol uzunlukları ile birlikte Tablo 1.1'de verilmektedir.



Şekil 1.1: Dört düğümlü bir şebeke örneği

Tablo 1.1: Çözüm seçeneklerinin tam listesi

12341	11
12431	13
13241	12
13421	13
14231	12
14321	11

Çözüm seçeneklerinin sayısı $(n-1)!$ ile ifade edildiğinden düğüm satışının artışı ile birlikte çözüm seçeneklerinin sayısının artışı doğrusal olmamakta; üstel bir artış göstermektedir. Bu tür problemler bu nedenle zor problemler sınıfında yer almakta ve tam listeme işlemi belirli bir büyüklüğün üzerindeki problemlerde makul sürelerde gerçekleştirilemediğinden imkansız hale gelmektedir.

Gezgin Satıcı Problemi bir 0-1 Tamsayı Doğrusal Programlama Modeli olarak modellenmek ve çözülmek istendiğinde Şekil 1.1'de verilen şebeke için aşağıdaki denklem takımı elde edilecektir:

$$Z_{min} = 4X_{12} + 2X_{13} + 3X_{14} + 4X_{21} + 2X_{23} + 5X_{24} + 2X_{31} + 2X_{32} + 2X_{34} + 3X_{41} + 5X_{42} + 2X_{43}$$

$$\begin{array}{ll}
1 & X_{21} + X_{31} + X_{41} = 1 \quad X_{12} + X_{13} + X_{14} = 1 \\
2 & X_{12} + X_{32} + X_{42} = 1 \quad X_{21} + X_{23} + X_{24} = 1 \\
3 & X_{13} + X_{23} + X_{43} = 1 \quad X_{31} + X_{32} + X_{34} = 1 \\
4 & X_{14} + X_{24} + X_{34} = 1 \quad X_{41} + X_{42} + X_{43} = 1
\end{array}$$

$$X_{12}, X_{13}, X_{14}, X_{21}, X_{23}, X_{24}, X_{31}, X_{32}, X_{34}, X_{41}, X_{42}, X_{43} \in 0 \text{ veya } 1$$

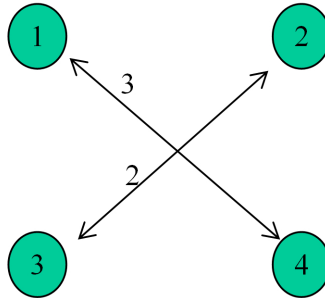
Modelde alt tur engelleme kısıtları toplu olarak verilmemiştir, çözüm aşamalarında gerekli olduğunda modele eklenerek çözüm süreci sürdürülmektedir.

Verilen modelin çözümünden elde edilen sonuçlar aşağıda verilmektedir ve minimizasyon olarak ifade edilen amaç fonksiyonu için oldukça iyi bir değer elde edilmiştir ancak çözümün şebeke üzerindeki ifadesini gösteren Şekil 1.2'den de görülebileceği gibi, oluşan ikili alt turlar nedeniyle tam bir tur elde edilememiştir.

$$\begin{array}{l}
X_{14} = 1 \\
X_{41} = 1 \\
X_{23} = 1 \\
X_{32} = 1 \\
Z = 5
\end{array}$$

Bu nedenle aşağıda verilen, gerekli alt tur engelleme kısıtlarını ekleyerek modeli yeniden çözmek gerekmektedir:

$$\begin{array}{l}
X_{14} + X_{41} < 2 \\
X_{23} + X_{32} < 2
\end{array}$$



Şekil 1.2: Alt tur içeren çözüm örneği

Benzer süreç alt tur içermeyen çözüm elde edilene kadar sürdürülmelidir. Görüldüğü gibi örnekte verilen çok küçük ölçekli problemde dahi zaman alıcı bu sürecin büyük ölçekli problemlerin çözümünde gerçekleştirilme şansı görülmemektedir. Bir diğer seçenek olan tüm alt tur engelleme kısıtlarının modele başlangıç aşamasında dahil edilerek çözüm aranmasında ise kısıt ve tamsayı değişken sayısının çok büyük olması nedeniyle belirli problem büyüklüklerinin üzerinde matematiksel programlama yöntemleriyle çözüm elde edilememekte ve sezgisel veya metasezgisel yöntemlere başvurulmaktadır.

Gezgin Satıcı Probleminin çözümü için en bilinen sezgiseller arasında Ey Yakın Komşu Yöntemi (Nearest Neighbour), Araya Ekleme Yöntemi (Closest Insertion) ve Geometrik Yöntem sayılabilir. Sezgisel yöntemlerle çok büyük boyutlu problemler için dahi çok kısa sürelerde çözüm elde edilmekte ancak elde edilen çözümün kalitesi garanti edilememektedir.

Zor problemlerin çözümü için kullanılan bir diğer seçenek ise metasezgisel yöntemlerin kullanılmasıdır. En bilinen metasezgiseller arasında Genetik Algoritmalar, Karınca Kolonisi Optimizasyonu, Parçacık Sürü Optimizasyonu, Diferansiyel Gelişim Algoritması, Tavlama Benzetimi ve Tabu Arama sayılabilir. Bu sayılanların dışında son dönemlerde Yapay Arı, Yarasa, Örümcek, Kör Fare, Ateş Böceği Algoritmaları gibi pek çok metasezgisel yöntem de önerilmiştir.

1.2 Metasezgisel Yöntem Parametreleri

Metasezgiseller, yüksek kaliteli çözümleri etkin bir şekilde üretmek için arama uzayında zeki arama stratejileri kullanan genel amaçlı sezgiseller olarak tanımlanabilir. Metasezgisellerin en önemli özelliği; çeşitlendirme (diversification) ve yoğunlaşma (intensification) arasındaki dinamik dengeyi oluşturmasıdır. Metasezgisellerde, çözüm uzayında hem global hem de yerel aramayı dengeli biçimde yaparak, yerel optimumlara takılmaksızın problemin optimum çözümüne yakınsamak üzere iteratif bir süreç işletilmektedir. Sözü edilen iteratif sürecin gerçekleştirilmesi bir takım yerel ve global arama mekanizmaları ile sağlanmaktadır. Bu mekanizmalara örnek olarak; Genetik Algoritmalarda optimum çözüme yakınsayabilmek için uygun üreme, çaprazlama (pozisyona dayalı, sıraya dayalı, dairesel, doğrusal, vb.) ve mutasyon (ters mutasyon, komşu iki noktanın değişimi, keyfi iki noktanın değişimi, vb.) operatörlerinin seçimi ve uygun sıklık ve oranların kullanılması gereklidir. Çözüm kalitesine etki eden diğer parametreler arasında kodlama yapısı, başlangıç popülasyonunun büyüklüğü ve iterasyon sayısı sayılabilir. Özel modellerde ise elitizm seçeneğinin kullanılıp kullanılmayacağı, paralel algoritmaların niteliği gibi ek parametreler de yer alabilir.

Tavlama Benzetimi Algoritmasının parametreleri arasında Başlangıç sıcaklığı, Her sıcaklıkta üretilecek çözüm sayısı, Sıcaklık azaltma fonksiyonu ve Algoritmayı durdurma şartı sayılabilir.

Karınca Sistemi parametreleri arasında, ize (feromon) verilen bağıl önem derecesi, görünürlük değerine verilen önem derecesi, t ile $t+n$ süreleri arasındaki buharlaşan iz (feromon) oranı, karınca sayısı ve tur sayısı sayılabilir.

Bir başka metasezgisel yöntem olan Parçacık Sürü Optimizasyonunun parametreleri ise Parçacık sayısı, Parçacık boyutu, V_{max} (bir iterasyonda, bir parçacıkta meydana gelebilecek maksimum değişiklik), Parçacık aralığı, Öğrenme faktörü ve Sonlandırma kriteri olarak sıralanabilir.

Parametrelerin uygun düzeylerinde seçilmesiyle, çeşitlendirme (diversification) ve yoğunlaşma (intensification) arasındaki uygun denge oluşturularak optimuma yakınsama sağlanabilir. Genel olarak metasezgisellerin rassal süreçler barındırmaları nedeniyle aynı problem için aynı parametre kümesi ve aynı sayıdaki iterasyon sonunda farklı çözüm değerleri elde edilebilmektedir. Ancak her metasezgisel için probleme özgü en uygun parametre kümesi ile çalışmak zorunludur. Tablo 1.2'de bir Gezgin Satıcı Probleminin Karınca Sistemi Algoritması için 3 farklı parametrenin 1. ve 2. düzeyleri kullanılarak oluşturulan çeşitli kombinasyonlarla yapılan 30 tekrarlı çözümün her birinden elde edilen değerler listelenmektedir. Görüldüğü gibi elde edilen en iyi çözüm değeri kullanılan parametre kümesine bağlı olarak değişmektedir. Ayrıca aynı parametre kümesi kullanılarak yapılan 30 farklı çözüm denemesinde elde edilen en iyi çözüm değerleri de farklılaşmaktadır.

1.3 Parametre Optimizasyonu

Önceki bölümde bahsedilen nedenlerle optimum çözüm değerine yakınsayabilmek için kullanılacak metasezgisel yöntemin en uygun parametre değerleri ile çalıştırılması gereklidir. Parametrelerin uygun düzeylerinin belirlenmesi için literatürde yer alan yöntemler aşağıdaki gibi listelenebilir:

1. Literatürden Yararlanmak: Belirli bir problem sınıfının çözümü için kullanılacak metasezgisel yöntemin parametre değerlerine, daha önceki çalışmalarda kullanılan ve iyi performans gösterdiği bilinen parametre değerlerinin kullanımı oldukça tercih edilen bir yöntem olarak görülmektedir.

2. Deneme Yanılma Yöntemi: Belirli bir problem sınıfının çözümü için daha önceki çalışmalarda kullanılmış olan parametre değerleri civarında farklı değerlerle parametrelerin ince ayarının yapıldığı çalışmalar da literatürde yer almaktadır. Bu durum özellikle belirli bir problem türü için parametrelerin literatürde bulunabildiği ancak çözümü aranan problem büyüklükleri ile daha önce çalışılmamış olması halinde sıklıkla kullanılmaktadır.

3. Her Adımda Bir Parametrenin Optimizasyonu: Metasezgiselin kullandığı parametre kümesinde yer alan parametrelerin uygun düzeylerine her adımda bir parametre için karar vererek tüm parametre düzeylerine karar verilene kadar süreç sürdürülmektedir. Bu yöntem daha öncekilerle karşılaştırıldığında daha sistematik bir yöntem olarak karşımıza çıkmakla birlikte, parametreler arası etkileşimin göz ardı edilmesi ve sonuç üzerinde en büyük etkiye sahip parametrenin belirlenememesi zayıf noktaları olarak karşımıza çıkmaktadır.

Aşağıda bir Gezgin Satıcı Probleminin çözümünde kullanılan Karınca Sistemi için en uygun parametre düzeylerine, her adımda bir parametrenin optimizasyonu yapılarak karar verilmektedir. Karınca sisteminin kullandığı 5 parametre aşağıda listelenmektedir. Bu parametrelerden karınca sayısı (m) ve tur sayısı (t) büyüdükçe çözüm süresi artmakla birlikte, çözüm kalitesinin arttığı bilinmektedir. Bu nedenle m ve t 'nin uygun düzeylerine makul bir çözüm süresi üzerinden karar verilebileceğinden, her iki parametre değeri de başlangıç aşamasında belirli bir değer olarak kabul edilmiştir.

- α : Problemden ize (feromon) verilen bağıl önem derecesi parametresi
- β : Problemden görünürlük değerine verilen önem derecesi parametresi
- ρ : t ile $t + n$ süreleri arasındaki buharlaşan iz (feromon) oranı
- m : karınca sayısı, t : tur sayısı

Tablo 1.3'de m ve t değerlerinin belirli olduğu durum için, α ve β değerleri sabit tutularak ρ 'nun farklı düzeyleri için elde edilen çözüm değerlerinin ortalamaları ve standart sapmaları görülmektedir. Parametrelerin etkilerini ayrıntılı olarak gözlemlemek amacıyla farklı büyüklükteki 5 problem üzerinde çalışılmıştır. Süreçlerin rassallığından kaynaklanan değişkenliklerin etkisini minimize etmek amacıyla her problem 30 tekrarlı olarak çözülmüş; tüm çözümlerin aritmetik ortalamaları ve standart sapmaları hesaplanarak verilmiştir. 30 çözüm içerisinde elde edilen en iyi değer de farklı bir karar seçeneği olarak kullanılabilir.

Minimizasyon amaçlandığından tüm problemlerin çözümü için $\rho = 1.00$ değerinin en iyi ortalama değerleri verdiği görülmektedir ve bu aşamada $\rho = 1.00$ olarak belirlenmiştir. İkinci adımda m , t ve ρ parametreleri sabit tutularak çözüm kalitesinin α parametresine bağlı değişimi incelenmelidir. Aynı problemlerin yine 30 tekrarlı olarak çözümü sonucu elde edilen çözüm değerlerinin ortalamaları ile standart sapmaları hesaplanmış ve Tablo 1.4'te sunulmuştur. Beş problemin dördünde en düşük ortalama değer elde edilmesini sağlayan $\alpha = 1.30$ değerinin uygun olacağı söylenebilir.

Üçüncü adımda m , t , ρ ve α parametreleri sabit tutularak çözüm kalitesinin β parametresine bağlı değişimi incelenmelidir. Aynı problemlerin yine 30 tekrarlı olarak çözümü sonucu elde edilen çözüm değerlerinin ortalamaları ile standart sapmaları hesaplanmış ve Tablo 1.5'te sunulmuştur.

Beş problemin üçünde en düşük ortalama değerin elde edilmesini sağlayan $\beta=1.30$ değerinin uygun olacağı söylenebilir. Görüldüğü gibi seçim işlemi sezgisellik ve sübjektiflik içermektedir.

4. Deney Tasarımı: Bir tasarlanmış deney, bir süreç ya da sistemin girdi değişkenleri üzerinde anlamlı değişikliklerin yapıldığı, böylelikle yanıt çıktılarında meydana gelen değişikliklerin sebeplerini tanımlamamızı sağlayan bir test ya da testler dizisidir. Kullanılan deney yapıları genel olarak Tam Faktöriyel, Kısmi Faktöriyel, Taguchi, Yanıt Yüzeyi Yöntemi şeklinde gruplandırılabilir. Tam Faktöriyel deneyler sonucunda metasezgiselin kullandığı parametrelerden hangisinin çözüm kalitesi üzerinde daha büyük etkisinin olduğu anlaşılabilir gibi, parametreler arasındaki etkileşimler de görülebilmektedir ancak çok sayıda deney yapmak gerektiğinden süreç zaman alıcı olmaktadır. Parametreler arası etkileşimin ihmal edildiği durumlarda yine parametrelerin sonuç üzerindeki etkisini görmek mümkün olduğundan Kısmi Faktöriyel ya da Taguchi Yöntemleri kullanılabilir. Taguchi yöntemi, belirli parametre sayısı ve düzeyleri için Taguchi tarafından önerilen ortogonal dizi standartlarının bulunması nedeniyle parametre optimizasyonunda daha yaygın bir kullanım alanı bulmaktadır.

Aşağıda Karınca Sistemi parametrelerinin Taguchi yöntemiyle belirlenmesine yönelik süreç özetlenecektir. Metasezgiselin 3 parametresi ve her parametrenin de Tablo 1.6'da verilen 2 seviyeli incelemesi yapılacağından Taguchi'nin L4 ortogonal dizisinin uygun olacağı görülmektedir.

Gerçekleştirilen dört deneyden 30 tekrarlı olarak elde edilen çözüm değerleri Tablo 1.7'de verilmektedir. İlk sütunda her 3 parametrenin de 1. Düzeyleri kullanılarak elde edilen çözümler yer almaktadır. Çözümlerin aritmetik ortalamasının ilk parametrenin 1. düzeyde; diğer parametrelerin ise 2. düzeylerinde kullanıldığında elde edildiği (7168,8) gözlemlenmektedir. Tüm deneylerde elde edilen minimum çözüm değeri (6840) de yine aynı çözüm kümesi ile elde edilmiştir. Dört deneye ait toplu sonuçlar Tablo 1.8'de verilmektedir.

Toplu sonuçlar üzerinde yapılan ve Tablo 1.9'da verilen analizlerde parametrelerin 1. ve 2. düzeyleri kullanılarak elde edilen çözümlerin ortalama değerleri arasındaki farklardan çözüm üzerinde en büyük etkisi olan parametrenin ρ parametresi olduğu görülmektedir. Bu parametrenin 2. düzeyi yerine 1. düzeyi kullanıldığında elde edilen çözüm değerlerinin ortalaması yaklaşık 94 birim iyileşmektedir. Bu nedenle bu parametre için 1. düzeyin tercih edilmesi uygun olacaktır. Benzer değerlendirmeler diğer parametreler için de yapıldığında, onlar için ise 2. düzeylerin seçilmesinin yararlı olacağı anlaşılmaktadır. Çözüm kalitesine ikinci sırada etki eden parametrenin ise -28,567 birimlik fark değeri ile β parametresi olduğu görülmektedir.

1.4 F-Race algoritması

F-Race algoritması metasezgisel yöntem parametrelerin belirlenmesinde istatistiksel test kullanan bir algoritmadır. Algoritma, parametre uzayını ayırklaştırmakta ve sınırlı sayıda parametrenin oluşturduğu vektörler ile çalışmaktadır.

Algoritmanın tanımlanabilmesi için ilk olarak şu nesnelere tanımlanmalıdır:

- Θ : Aday parametre yapılandırılmalarının sonlu elemanlı kümesi.
- I : Aday parametre yapılandırılmalarının test edileceği problemlerin sonlu veya sonsuz elemanlı kümesi.
- P_i : i probleminin karşılaşıma olasılığı.
- t : Her bir test problemi için belirlenen zamanı ifade eden fonksiyon.
- $c(\theta, i) = c(\theta, i, t(i))$: Seçilen bir i probleminin seçilen bir θ aday parametresi ile test problemine karşılık gelen sürede çözülmesinin maliyetini ifade eden rastgele değişken.
- $C \subset R$, c rastgele değişkeninin değer kümesi.

- P_C , C kümesi üzerinde tanımlanan bir olasılık ölçütü. $P_C(c|\theta, i)$ ile θ yapılandırmasının i problemine $t(i)$ kadar zamanda elde edilen çözümün c maliyetine ait olasılığı ifade edilmektedir.
 - $C(\theta) = C(\theta|\Theta, I, P_I, P_C, t)$ değeri θ 'ya göre optimize edilmelidir.
- Yani problem

$$\theta^* = \operatorname{argmin}_{\theta} C(\theta)$$

Biçiminde ifade edilebilmektedir.

Ancak $C(\theta)$ kriteri için farklı seçenekler söz konusudur. F-Race algoritması temel olarak maliyet değişkeni olan c rastgele değişkeninin beklenen değerini ele almışlardır. Böylece çözülmesi gereken ifade

$$C(\theta) = E_{I,C}[c(\theta, i)] = \int \int c(\theta, i) dP_C(c|\theta, i) dP_I(i)$$

biçimini almaktadır. Bu entegralin analitik çözümü her zaman açıkça tanımlanamayacağından dolayı F-Race algoritması beklenen değer hesabı için monte-carlo benzetimi kullanmaktadır.

Deney tasarımı yöntemi ayrılaştırılan tüm parametreleri içeren uzayın her bir örnek problemde çalıştırılması ile elde edilen sonuçları değerlendirmekte ve beklenen değer kriterini en iyi yapan değeri seçmektedir. F-Race algoritması ise parametre uzayında bulunan parametreleri deneme sırasında istatistiksel testler kullanarak mevcut parametrelerin elde ettiği sonuçlara göre baskınlık-basılgnlık kontrolü yaparak parametrelerden hangilerinin çözüm olamayacağını önceden belirlemeye ve çözüm olamayacak bu parametreleri denemeyi bırakmayı hedeflemektedir.

Bu işlem için kullanılan testler sıralama istatistiklerine dayalı testlerdir. Bu sebeple algoritma, parametre vektörleri kullanılarak elde edilen çözüm değerlerinin sayısal değeri yerine parametre vektörleri ile elde edilen çözümün diğer parametre vektörlerine göre kaçınıcı sırada olduğunu kullanmaktadır.

Algoritma her bir adımda bir örnek problem seçmekte ve seçilen örnek problem için elde bulunan bütün parametre vektörleri ile birer çözüm değeri elde etmektedir. Elde edilen bu çözüm değerleri sıralanmakta ve her bir vektörün kaçınıcı olduğu belirlenmektedir.

İkinci aşamada ise daha önceden elde edilen sıralamalar ile birlikte bu çözüm vektörlerinin en iyi olanı(en çok birinci olanı veya en düşük sıralama toplamına sahip olanı) ile diğer parametre vektörleri arasında anlamlı bir farklılığın bulunup bulunmadığı test edilmektedir. Bu işlem için Friedman'ın test istatistiği kullanılmaktadır. Test istatistiği şu şekilde ifade edilmektedir:

$$T = \frac{(n-1) \sum_{j=1}^n (R_j - k(n+1)/2)^2}{\sum_{l=1}^k \sum_{j=1}^n R_{lj}^2 - \frac{kn(n+1)^2}{4}}$$

Hesaplanan test istatistiğinin yaklaşık olarak $n-1$ serbestlik dereceli χ^2 dağılımına uyması beklenmektedir. Test sonucu değerlendirilerek parametreler arasında anlamlı farklılığın olmadığı görülür ise bir sonraki örnek probleme geçilmektedir. Aksi durumda ise en iyi parametre vektörü ile diğer vektörlerden bir veya birkaçı arasında önemli farklılık bulunduğu sonucu ortaya çıkar. Bu durum, bazı parametre vektörlerini denemenin artık gereksiz olduğunu ve bu vektörleri elemenin

mümkün olduğunu belirtir. Hangi vektörlerin elenmesi gerektiğini belirlemek için en iyi parametre vektörü ile diğer parametre vektörleri arasında yine istatistiksel test yapılır. Bu defa yapılan test en iyi olduğu düşünülen vektör ile diğer vektörlerin tamamıdır. Kullanılan test ise sıralama verileri için Student's t testidir. Artık sadece ikili karşılaştırma yapılmaktadır, dolayısı ile en iyi parametre vektörü ile seçilen diğer parametre vektörü arasında yapılan karşılaştırma test istatistiği şu şekildedir:

$$\frac{|R_j - R_h|}{\sqrt{\frac{2k(1 - \frac{T}{k(n-1)}) (\sum_{l=1}^k \sum_{j=1}^n R_{lj}^2 - \frac{kn(n+1)^2}{4n+1})}{(k-1)(n-1)}}} = t$$

Elde edilen test sonucu eğer seçilen güven düzeyi için belirlenen $t_{(1 - \alpha/2)}$ değerinden büyük ise en iyi parametre vektörü ile diğer parametre vektörü arasında anlamlı fark vardır ve artık diğer parametre vektörünün test edilmesine gerek olmamaktadır. Dolayısı ile elenecek olan parametre vektörü Θ kümesinden çıkartılabilir (Birattari ve arkadaşları, 2009).

Algoritma çalışmaya başlar başlamaz en iyi parametre vektörünün belirlenmesi örneklem az olduğundan hataya yol açabileceğinden algoritmaya ısınma süresi verilmektedir. Bu ısınma süresi, kaç çözüm elde edildikten sonra algoritmanın istatistiksel testlere başlayacağını belirtmektedir.

Algoritma çalışması için şu parametrelere ihtiyaç duymaktadır:

- Isınma çözümleri sayısı
- İstatistiksel geçerlilik seviyesi
- Algoritmanın toplam çalışma zamanı

Algoritmanın geliştirilmiş olmasından sonra ortaya çıkan bazı problemlerin giderilmesi amacı ile i-race adlı bir yöntem önerilmiştir. Bu yöntem de temel olarak aday parametre vektörlerinin nasıl belirleneceğine dair bir geliştirmeyi barındırmaktadır.

Anlaşılabacağı üzere metasezgisel yöntemlerde parametre optimizasyonu başlı başına bir konudur ve metasezgisel yöntemlerin parametrelerini belirlemek performans üzerinde oldukça büyük etkiye sahiptir. Konunun önemi nedeniyle yukarıda özetlenen yöntemlerin dışında i-race, REVAC, bonesa ve paramILS gibi pek çok yöntem daha önerilmiştir (Montero, 2012).

1.5 Kaynakça

M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, A Racing Algorithm for Configuring Metaheuristics, GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, pages 11–18. Morgan Kaufmann Publishers, San Francisco, CA, 2002.

E. Montero, Parameter Setting in Metaheuristics, Universidad Tecnica Federico Santa Maria, 2012

Tablo 1.2: Farklı parametre kümeleri ile elde edilen tekrarlı çözüm değerleri

Deneme no	1,1,1	1,2,2	2,1,2	2,2,1
1	7195	7336	7355	7223
2	7114	6840	7362	7191
3	7053	7193	7191	7333
4	7026	7194	7256	7237
5	7114	7306	7133	7362
6	7391	7336	7386	7295
7	7095	7034	7422	7030
8	7348	7305	7244	7263
9	7173	7276	6969	7219
10	7114	7244	7191	7114
11	7259	7191	7393	7332
12	7053	7198	7191	6960
13	7026	7180	6969	7100
14	7384	7191	7467	7288
15	6964	7056	7191	7110
16	7348	7180	7247	7235
17	7121	6947	7191	7247
18	7335	7202	7152	7303
19	7237	7100	7252	7394
20	7413	6974	6942	7359
21	7211	7136	7101	6857
22	7170	7020	7357	7305
23	7263	7393	7257	6997
24	7114	7069	7020	7305
25	7236	7053	7395	7327
26	7143	7312	7236	7399
27	7252	7310	7082	7223
28	7172	7115	7379	7191
29	7114	7121	7191	7310
30	7288	7252	7191	7399
A.Ort.	7190,87	7168,8	7223,77	7230,27
S.sapma	119,672	130,665	140,413	135,806

Tablo 1.3: Çözüm kalitesinin ρ 'ya bağlı değişimi

Problem No		İz Buharlaşma Derecesi (ρ parametresi)				
		0	0.25	0.50	0.75	1.00
P1	A.ortalama	9023	9021	8855	9058	8399
	S.Sapma	164,5	234,9	97,8	121,16	210,66
P2	A.ortalama	10132	10000	10075	10088	9188
	S.Sapma	161,4	309,7	268,8	341,4	200,3
P3	A.ortalama	10169	10139	10159	10120	9302
	S.Sapma	229,2	260,1	183,6	267,1	172,8
P4	A.ortalama	7439	7454	7393	7482	6849
	S.Sapma	148,8	120,6	168,5	166,4	120,2
P5	A.ortalama	203	198	197	191	176
	S.Sapma	4,34	3,97	4,05	4,65	1,72

Tablo 1.4: Çözüm kalitesinin α ya bağlı değişimi

Problem No		İz Yoğunluğu Bağlı önem Derecesi(α)					
		0.00	1.00	Oca.30	Oca.80	2.00	5.00
P1	A. ortalama	8964	8399	8104	7967	7932	8548
	S. Sapma	233	210,6	119,4	105,6	276,6	256,2
P2	A. ortalama	10178	9188	8992	9008	9066	9298
	S. Sapma	207	200,3	138,9	173,1	145,5	159,3
P3	A. ortalama	10303	9302	8733	8732	8901	9285
	S. Sapma	135,8	172,8	69,6	246,3	185,1	235,6
P4	A. ortalama	7316	6849	6696	6825	6760	7107
	S. Sapma	146,6	120,2	66,53	200,3	91,63	190,1
P5	A. ortalama	201	176	175	177	176	194
	S. Sapma	3.Ağu	1,7	3,2	3,2	4,5	7,8

Tablo 1.5: Çözüm kalitesinin β 'ya bağlı değişimi

Problem No		Öncelik Kuralı önem Derecesi(b)					
		0.00	1.00	Oca.30	Oca.80	2.00	5.00
P1	A. ortalama	8520	8399	8476	8494	8527	8454
	S. Sapma	137	210,6	154,8	125,1	66,2	53,6
P2	A. ortalama	9669	9188	9122	9349	9439	9449
	S. Sapma	143,4	200,3	250,8	122,2	202,3	169,9
P3	A. ortalama	8805	9302	9243	9505	9516	10076
	S. Sapma	196,6	172,8	90,2	191,8	150,2	171,1
P4	A. ortalama	6987	6849	6879	6955	7114	7441
	S. Sapma	132,5	120,2	110	106,1	178,6	144,2
P5	A. ortalama	175	175	174	176	179	193
	S. Sapma	5,47	3,22	3	5,1	4,63	5

Tablo 1.6: Karınca Sistemi parametreleri ve parametre düzeyleri

Faktör kodu	Faktör	Seviye	
		1	2
ρ	İz buharlaşma derecesi	0,25	0,75
α	İz yoğunluğu önem derecesi	1,25	3,75
β	Öncelik kuralı önem derecesi	1,25	3,75

Tablo 1.7: Problemin 30 tekrarlı çözüm değerleri

Deneme no	1,1,1	1,2,2	2,1,2	2,2,1
1	7195	7336	7355	7223
2	7114	6840	7362	7191
3	7053	7193	7191	7333
4	7026	7194	7256	7237
5	7114	7306	7133	7362
6	7391	7336	7386	7295
7	7095	7034	7422	7030
8	7348	7305	7244	7263
9	7173	7276	6969	7219
10	7114	7244	7191	7114
11	7259	7191	7393	7332
12	7053	7198	7191	6960
13	7026	7180	6969	7100
14	7384	7191	7467	7288
15	6964	7056	7191	7110
16	7348	7180	7247	7235
17	7121	6947	7191	7247
18	7335	7202	7152	7303
19	7237	7100	7252	7394
20	7413	6974	6942	7359
21	7211	7136	7101	6857
22	7170	7020	7357	7305
23	7263	7393	7257	6997
24	7114	7069	7020	7305
25	7236	7053	7395	7327
26	7143	7312	7236	7399
27	7252	7310	7082	7223
28	7172	7115	7379	7191
29	7114	7121	7191	7310
30	7288	7252	7191	7399
A.Ort.	7190,867	7168,8	7223,77	7230,27
S.sapma	119,6719	130,665	140,413	135,806

Tablo 1.8: Deney Sonuçları

Deney No	En iyi Değer	Ortalama Değer	S.Sapma
1	6964	7190,87	119,672
2	6840	7168,8	130,665
3	6942	7223,77	140,413
4	6857	7230,27	135,806

Tablo 1.9: Parametrelerin önem düzeyleri ve tercih edilen düzeyler

L4

Deney No	Parametre No.			En iyi		
	ρ	α	β	ρ 2-1	α 2-1	β 2-1
1	1	1	1	-5	-209	-39
2	1	2	2		Ort.	
3	2	1	2	ρ 2-1	α 2-1	β 2-1
4	2	2	1	94,367	-15,567	-28,567
					3,75	3,75
				1	2	2



2. Kurumsal Kaynak Planlaması

Kurumsal Kaynak Planlaması Ve Yapay Zekâ Uygulamalarının Modellenmesi

Alptekin ERKOLLAR¹, Gülay EKREN²

¹Sakarya Üniversitesi, ²Sinop Üniversitesi

2.1 Giriş

Bu bölümün amacı, Kurumsal Kaynak Planlama (Enterprise Resource Planning, ERP) sistemlerine kısa bir giriş yapmak, ERP sistemlerinde ve iş süreçlerinde kritik öneme sahip modelleme yaklaşımlarını ve araçlarını tanıtmak ve işletmelerde yapay zeka uygulamalarının modellenmesi üzerine genel bir bakış sunmaktır. Bu bölümde öncelikle ERP sistemlerinin genel yapısı, çalışma şekli, geçmişten bugüne evrimi ve günümüz işletmelerinin ERP sistemlerine duyulan gereksinim tartışılmıştır. Daha sonra ERP sistemlerinin uygulama (implementasyon) ve iyileştirme süreçlerinde kritik öneme sahip iş süreçlerinin modellenmesi, işletmelerde yapay zekâ kullanımına yönelik artan ilgi ve işletmelerde yapay zekâ uygulamalarının modellenmesi gibi konulara değinilmiştir.

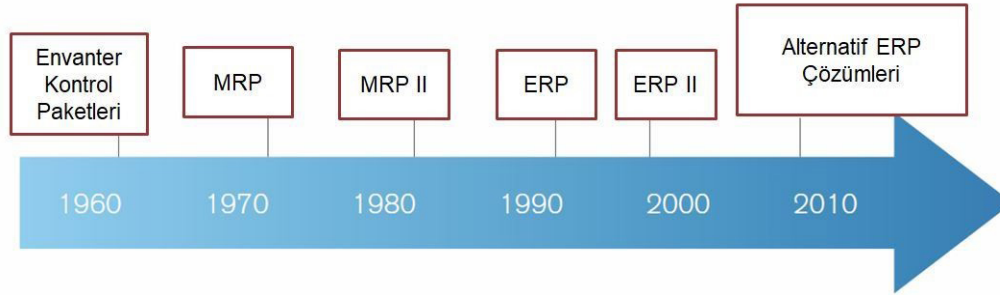
2.2 Kurumsal Kaynak Planlama Sistemleri

ERP sistemleri bir işletmenin temel süreçlerini ortak bir veritabanı ile birbirine bağlayan ve verilere gerçek zamanlı erişim sağlayan, yüksek maliyetli (kurulum, bakım, eğitim, danışmanlık hizmetleri maliyetleri vs.) karmaşık sistemlerdir. Bu sistemler finansal süreçlerin yanı sıra birbiriyle entegre tedarik, satış ve imalat gibi birçok sürecin izlenmesini sağlayan modüler yapıda sistemlerdir. ERP sistemleri, işletmelerin tüm iş gereksinimlerini, tek bir veritabanına bağlı entegre bir program aracılığı ile bir araya getirerek, tüm departmanların sahip oldukları bilgileri paylaşarak birbirleriyle etkileşim içinde çalışabilmelerine olanak sağlamaktadır.

ERP sistemleri 1960'lardan bu yana bir evrim sürecinin içinden geçmektedir (Şekil 2.1). 1960'larda işletmeler çoğunlukla envanter işleme üzerine odaklanmakta, yazılım paketlerinin geliştirilmesi ise daha çok malzemelerin tam zamanında teminine dayalı stoklama sistemlerinin oluşturulması

amacıyla kullanılmaktaydı. 1970'lerde oluşturulan MRP (Materials Requirement Planning, Malzeme Kaynak Planlaması) sistemleri, malzeme planlama sürecinde atılmış büyük bir adım olarak görülmüştür. Bu sistemle üretici artık, ürettiği her bir malın hangi tarihte ve ne miktarda temin edileceğini gösteren bir Ana Üretim Çizelgesine (Master Production Schedule, MPS) sahiptir. 1980'lerde MRP II (Manufacturing Resource Planning, İmalat Kaynak Planlaması) adı verilen yeni sistemler, üretim ve malzeme yönetiminin yanı sıra finansal muhasebe ve finansal yönetim sistemlerini içerecek şekilde geliştirilmiştir. 1990'larda ise bu sistemlerin kullanımı ilgili sektör dışında farklı sektörler tarafından da kullanılabilir şekilde tasarlanarak "ERP sistemleri" ortaya çıkmıştır. Bu sistemler, malzeme planlaması, ürün tasarımı, kapasite planlaması, insan kaynakları, bilgi ve iletişim teknolojileri, finans ve proje yönetimi gibi pek çok alan için çözümler içerecek şekilde geliştirilmiştir.

2000'li yıllarda internetin yaygınlaşması ile birlikte ERP sistemlerine, genişletilmiş ERP (Extended ERP) ve devamında ERP II gibi tanımlamalar yapılarak, daha büyük bir pazara hizmet verecek şekilde kapsamı genişletilmiştir. Bu sistemler bir işletmenin farklı şubeleri arasında veya farklı endüstriler arasında işletme fonksiyonlarının entegrasyonuna izin verecek şekilde; ileri planlama ve çizelgeleme ve müşteri ilişkileri yönetimi, tedarik zinciri yönetimi gibi e-iş çözümleri eklenerek yeniden yapılandırılmıştır (Rashid, Hossain ve Patrick, 2002). Bu gelişim süreci günümüzde halen bilgi ve iletişim teknolojilerindeki ve bulut teknolojilerindeki gelişmelere paralel olarak ilerlemektedir. 2010 yılından bu yana, mobil ve web tabanlı ERP sistemleri gibi yeni iş modelleri ve açık kaynak ERP (openERP, openBravo, Odoo vs), bulut tabanlı (on-demand) ERP gibi alternatif ERP çözümleri geliştirilmekte ve kullanılmaktadır (Koslowski ve Strüker, 2011; Pinckaers, Gardiner ve Van Vossel, 2011; Purohit, Jaiswal ve Pandey, 2012; Gelogo ve Kim, 2014).

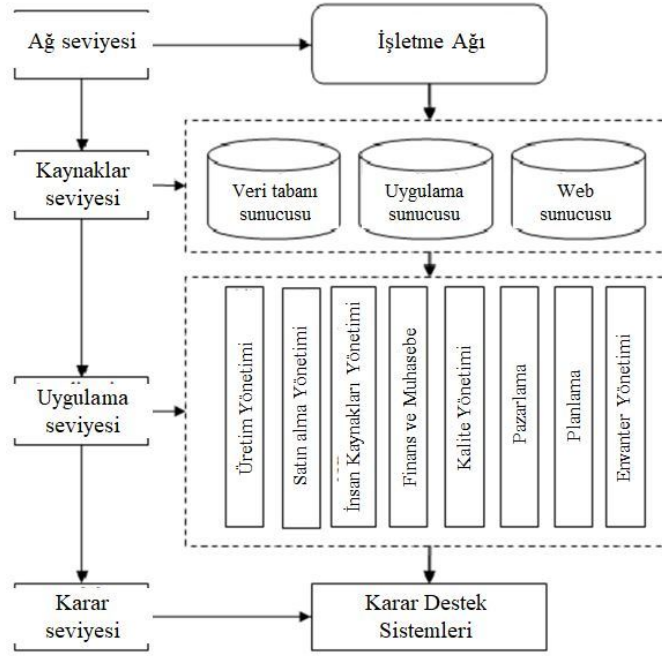


Şekil 2.1: ERP Sistemlerinin Gelişim Süreci

ERP sistemleri genel yapı itibarıyla ağ, kaynaklar, uygulama ve karar olmak üzere dört seviyeden oluşan bir mimariye sahiptir (Şekil 2.2). Ağ altyapısı sayesinde veriler, işletme içinde ve dışında sorunsuzca akmaktadır. Kaynaklar, ERP sisteminin ihtiyaç duyduğu donanım, yazılım ve verileri sağlamaktadır. Uygulamalar ise bir işletmenin farklı departmanlarındaki personel tarafından kullanılan alt sistemleri (modüller) kapsamaktadır. Bu modüller (üretim, satın alma, insan kaynakları, finans, muhasebe, kalite, pazarlama, planlama, envanter takibi vs.) ERP sisteminin görünen yüzünü oluşturmaktadır. Bir takım modeller veya metotlar kullanılarak alt sistemdeki veriler karar destek sistemleri aracılığıyla işlenmekte ve elde edilen raporlar üst yönetimin işletme hakkında vereceği kararlara destek olmaktadır.

Bu sistemlerin mimarisi, sistem içindeki modüllerin ve diğer öğelerin entegrasyonunu kolaylaştırmakta, bir işletme içindeki tüm fonksiyonel görevler arasında bilgi akışını tutarlı bir şekilde görünür kılmaktadır (Rashid ve ark., 2002).

ERP sistemleri genellikle ilişkisel bir veritabanlarını kullanmakta ve bu veritabanları SQL



Şekil 2.2: ERP Sistemlerinin Genel Yapısı

tipi sorguları gerektirmektedir. Ancak işletmeler ERP sistemlerinde genellikle işlevsellik ve hızı önemsemektedir. Sadece SQL kullanılan veritabanlarında sistemin ihtiyaç duyduğu işlevselliği ve hızı yakalamak mümkün değildir, çünkü bu veritabanlarında veriler dikey olarak büyüdüğü için yeni sunucularda performans sorunları ortaya çıkarabilmektedir, örneğin bazen bir sipariş yeri girmek sistemi kilitleyebilmektedir. Bu nedenle bu sistemlerde ilişkisel veritabanı dışında NoSQL veritabanları da kullanılabilir. NoSQL veritabanları büyük veriler için tasarlandığı için performans problemleri yaşanmamaktadır. Ayrıca, ilişkisel veritabanları sadece yapılandırılmış (structured) veriyi kullanabilirken, NoSQL veritabanları hem yapılandırılmış hem de yapılandırılmamış (unstructured) veya yarı yapılandırılmış (semi-structured) verileri kullanabilmektedir.

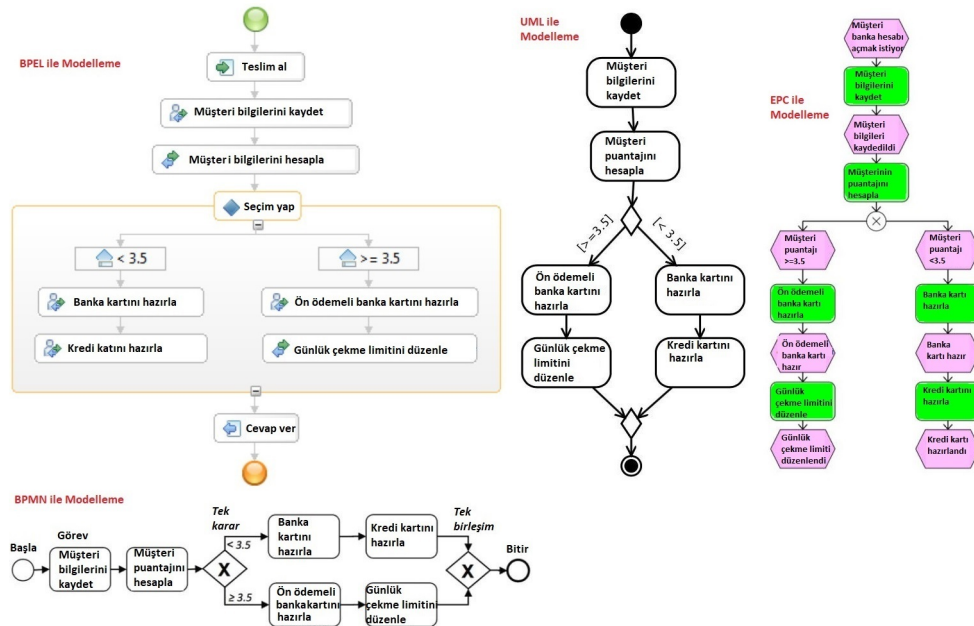
ERP sistemlerinde büyük veri kullanımına yönelik en güzel örneklerden biri 2014 yılında Brezilya'da gerçekleştirilen FIFA Dünya Kupası karşılaşmalarında yaşanmıştır. Bu turnuvanın şampiyonu Almanya olmuştur. Turnuvada SAP ve Almanya Futbol Federasyonu (DFB), kupadaki oyuncu performanslarını artırmak için büyük veriyi akıllı kararlara dönüştürecek bir sistem için işbirliği yapmışlardır. Futbolcuların antrenmanlarını, hazırlıklarını ve turnuvalarını analiz etmek ve bu analiz sonuçlarına göre oyuncu - takım performanslarını artırmak amacıyla SAP HANA ERP platformu üzerinde bir uygulama geliştirilmiştir. 10 dakikada 10 oyuncunun, 7 milyondan fazla veri noktasından sensörler aracılığı ile verileri toplanmıştır. Bu veriler, bir sonraki maçın antrenmanlarının ve hazırlığının yapılmasında kullanılmıştır. Bu uygulama Almanya'ya kupayı getirmede büyük rol oynamıştır (Doğan ve Arslantekin, 2016).

2.3 İş Süreçlerinin Modellenmesi

İş süreçlerinin modellenmesi (Business Process Modelling, BPM), işletmelerde, iş süreçlerini yeniden tasarlama, tekrarlanabilir hale getirme, mevcut riskleri belirleme, etkinleştirme, denetleme

ve süreç, otomasyonunu sağlama gibi amaçları gerçekleştirmek için kullanılan bir alandır. BPM genel olarak süreç sahibine, süreç analistlerine ve diğer tüm çalışanlara işletmede neler olup bittiğine dair genel bir fikir vermektedir. İş süreçlerini modellemenin asıl amacı işletmelerdeki iş süreç akışını tanımlamaktır. Bir işletmede bir iş için süreç modelleme yapıldıktan sonra şu soruların yanıtlanması beklenmektedir: İş sürecinin sonunda ne oldu? İş süreci içerisinde hangi faaliyetler gerçekleştirilir? Faaliyet sırası nedir? Faaliyetleri kimler gerçekleştirir? İşlem sırasında hangi ticari belgeler takas edilir? Süreç kusursuz mu? İyileştirme gerektiriyor mu, ileride nasıl güncellenebilir? gibi. Bu ve buna benzer sorulara cevap verilebilirse bir işletmede işlerin nasıl yürüdüğüne dair genel bir fikre sahip olunabilir. İşletmenin yapısal, örgütsel ve teknolojik olarak zayıf noktaları ve hatta süreçlerin tıkandığı noktalar tanımlanabilir ve sürecin iyileştirilmesine yönelik adımlar atılabilir.

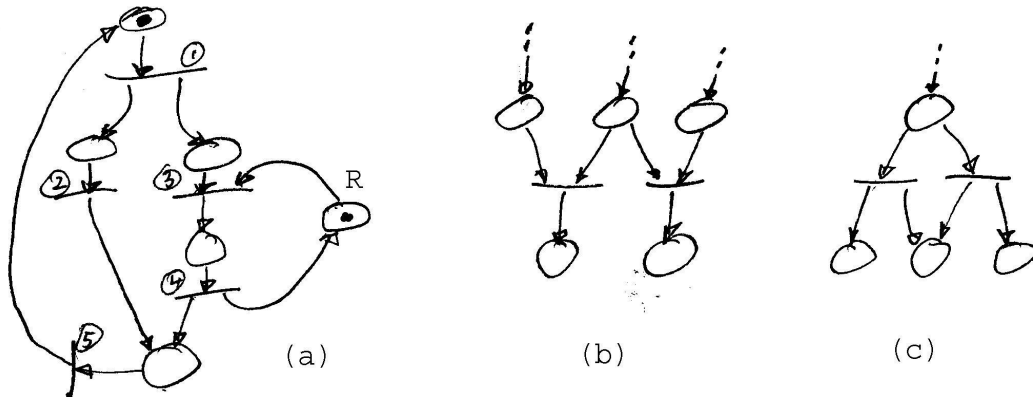
İşletmelerde iş süreçlerinin iyileştirilmesine yönelik önemli görülen diğer bir konu, iş süreçlerinin yeniden yapılandırılması (Business Process Re-engineering, BPR) konusudur. BPR, şirketlerin git-tikçe değişen pazarlarda rekabet edebilirliklerini ve kazançlarını artırmalarını sağlamak açısından önemli görülmektedir. Günümüz müşteri odaklı işletmelerin, temel süreçlerini yapılandırarak değer odaklı olmaları beklenmektedir (Williams, 2013). Diğer yandan, iş süreçlerini başarılı bir şekilde yeniden yapılandırılmanın önündeki en büyük engellerden biri karmaşık süreç mantığının zor yakalanmasıdır. Bu engeli aşmak için güçlü modelleme araçlarına ihtiyaç duyulmaktadır (Williams, 2013). Günümüzde işletmeler iş süreçlerini modellemek için ihtiyaçlarına ve sahip oldukları kaynaklara göre farklı modelleme araçları kullanmaktadır. Örneğin; BPMN, UML Aktivite Diyagramları, BPEL (Wohed ve ark., 2006), IDEF, ebXML, RosettaNet, EPC (Zur Muehlen ve Recker, 2013) gibi modelleme araçlarının birçoğu endüstriyel alanda yaygın olarak kullanılmaktadır (Şekil 2.3).



Şekil 2.3: Modelleme Araçları

2.4 Petri-net Modelleme

Petri-net modelleme süreç madenciliğinin temel yaklaşımıdır. Bu modelleme yaklaşımında bir jeton (token) ile sinyalin akışı modellenmektedir. Genel olarak bir petri ağında belirli sayıda yer (place) ve geçiş (transition) bulunmaktadır. Petri ağının (petri-net) durumu, farklı yerlerde bulunan jeton kümeleriyle tanımlanmaktadır. Tüm giriş yerleri en az bir jeton içerdiğinde bir geçiş etkinleştirilmektedir. Bir geçişin gerçekleştirdiği eylem, her girişte yerinden bir jeton almak ve her çıkışta yerine bir jeton eklemek şeklindedir (Bochman, 2008).



Şekil 2.4: Petri-net Modelleme Örneği (Bochman, 2008)

Şekil 2.4'deki petri-net ile modelleme örneğinde iki jeton bulunmaktadır. (a) örneğinde, R yerinde jeton olmadan 3 ve 4 numaralı geçişlerin olması mümkün değildir. R yerindeki jeton aslında 3 ve 4 geçişleri için gerekli bir kaynak olarak düşünülebilir. Bu kaynak bir araç, makine veya bu geçişleri yapan bir insan olabilir. (b) örneğinde, birbiri ile uyuşmayan iki geçiş görülmektedir. Her bir yerde sadece bir jeton olduğu varsayılırsa, geçişlerin her ikisinin çalışması mümkün olmayacaktır, bu durumda sadece bir geçişin seçilmesi zorunluluğu oluşacaktır. (c) örneğinde ise, sözde seçimi serbest bırakılmış iki geçiş görülmektedir. Eğer ağdaki tüm geçişler böyle olursa bu ağa serbest seçimli ağ denilmektedir (Bochman, 2008).

Bugüne kadar geliştirilmiş iş süreçlerini modelleme araçlarının temelinde petri ağı petri-net modelleme yaklaşımı yatmaktadır. İş süreçlerini modellemek için kullanılan standart yollardan bazıları; iş süreçleri modelleme yöntemlerinin uygulanması veya petri ağlarının kullanılmasıdır (Onggo, 2014; Béleccki ve Molnár, 2016).

Diğer yandan, işletmelerde süreçlerin otomatik olarak akışlarının çıkarılması ve simülasyonunun yapılması amacıyla farklı modelleme notasyonları kullanılmaktadır. Her süreç modelleme aracı farklı bir notasyon (gösterim) gerektirmektedir. Geçmişte süreçleri modellemek için daha çok akış diyagramları veya blok diyagramları kullanılmıştır. Günümüzde ise bu amaçla, Olay-güdümlü Süreç Zinciri (Event-driven Process Chain, EPC), Genişletilmiş Olay-güdümlü Süreç Zinciri (eEPC), Bileşik Modelleme Dili (The Unified Modeling Language, UML), XML Süreç Tanımlama Dili (The XML Process Definition Language XPD) ve IDEF0, IDEF3 (Integration DEFinition), İş Süreci Modelleme Notasyonu (Business Process Modelling Notation, BPMN) gibi daha karmaşık araçlar ve diller kullanılmaktadır.

2.5 BPMN ile Modelleme

Süreç, bir işletme veya kuruluş içinde gerçekleştirilen faaliyetler dizisidir. BPMN için süreç, bir dizi faaliyet ve faaliyetleri izleyen kontrollerin akış nesnelere aracılığı ile grafiksel olarak tasvir edilmesidir. Süreç kavramı özünde hiyerarşiktir. Süreçler, işletme veya kuruluş genelinde olabileceği gibi tek bir kişi tarafından gerçekleştirilen işlemler şeklinde farklı düzeylerde tanımlanabilir.

BPMN'in ortaya çıkış amacı, iş süreci tasarımı (business process design) ile süreç uygulaması (process implementation) arasında standart bir köprü kurmaktır. Diğer bir ifade ile temel amacı, işletme süreçlerinin ilk taslaklarını oluşturan iş analistleri, bu süreçleri gerçekleştirecek olan ve teknolojiyi uygulamaktan sorumlu teknik ekip ve son olarak bu süreçleri yönetecek ve izleyecek işletme yöneticilerinin kolay anlayacağı bir işaret dili oluşturmaktır. BPMN ayrıca, Web Servisleri için İş Süreçleri Yürütme Dili (Business Process Execution Language for Web Services, BPEL4WS) gibi iş süreçlerinin yürütülmesi için tasarlanan web servis tabanlı XML dillerinin, işletme odaklı bir şekilde görselleştirilmesini sağlamaktır. Özellikle hizmet tabanlı mimariye (service-oriented architecture, SAO) uygun iş süreçlerini modellemek için kullanılmaktadır (Pant, 2008). Şekil 2.5'te BPMN ile modellemede kullanılan temel notasyonlar görülmektedir.

BPMN elemanı	Notasyonu	BPMN elemanı	Notasyonu
Event		Lane	
Activity		Data object	
Gateway		Data input	
Sequence flow		Data output	
Message flow		Data store	
Association		Group	
Data association		Text annotation	
Pool			

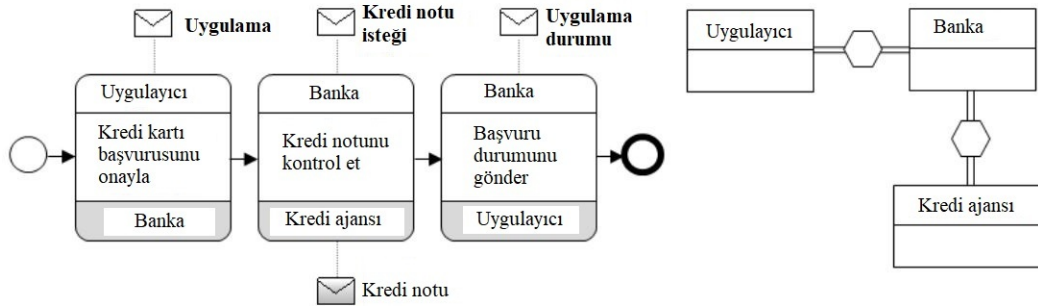
Şekil 2.5: BPMN Modellemede Temel Notasyonlar (OMG Document, 2010)

BPMN, iş süreçlerinde uygulanabilir olmayan yapıların; örneğin organizasyon yapıları ve bazı kaynakların; fonksiyonel arızalar, veri ve bilgi modelleri, strateji ve iş kuralları gibi modellenmesini desteklememektedir. Bunun yanı sıra, BPMN, her ne kadar veri (mesaj) akışını ve veri nesnelere (artifacts) faaliyetlerle ilişkisini gösterse de bir veri akış diyagramı olarak kabul edilmemektedir (Notation, 2006).

BPMN 2.0 notasyonunda, BPMN 1.0'da olmayan bazı eklemeler yapılmıştır. Mevcut süreç (process) ve işbirliği (collaboration) diyagramlarına ek olarak koreografi (choreography) ve konuşma (conversation) diyagramları eklenmiştir (Şekil 2.6). BPMN işbirliği diyagramları, BPMN süreç diyagramının tüm nesnelere içermektedir ve havuzlar (pool) arasındaki mesaj akışını desteklemektedir. Havuzlar arasında yüksek seviyedeki bir etkileşim ve kontrol akışının değişimi söz konusu olduğunda bu diyagramlar kullanılmaktadır (örneğin, bir katılımcı, diğer katılımcılarla bir mesaj akışı yoluyla ulaşana kadar beklemektedir). Genel olarak bir işbirliği diyagramında bir havuzda yalnızca bir işlemi göstermek, diğer havuza "kara kutu" olarak davranmak, ve bu havuz nesnesine karşılık gelen bir

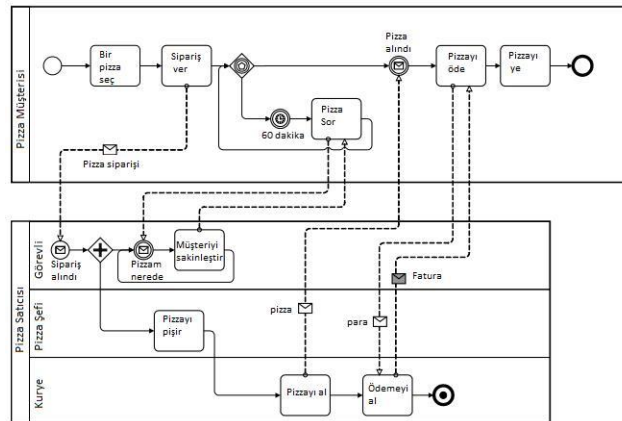
işbirliği diyagramı atamak iyi bir uygulamadır. Ancak süreç diyagramlarında katılımcı etkileşimi olmadığından (BPMN 2.0 ile) bunu göstermek için mesaj faaliyetlerini kullanmak gerekmektedir (Ronald, 2011; Onggo, 2014).

BPMN'e yeni eklenen koreografi diyagramının amacı katılımcılar arasındaki etkileşimi farklı bir formatta göstermektir. Bir süreçte bireysel görevlerin ayrıntıları yerine mesaj akışına yoğunlaşmaktadır. Dolayısıyla, yeni nesne tipleri olarak gönderen ve alıcıyı içermektedir (görevleri bir role bağlayarak veya swimlanes kullanarak) ve gönderilen/alınan mesajları bu nesneye bağlamaktadır. Ayrıca, süreç mantığını da (iş akışları ve ağ geçitleri kullanarak) göstermektedir. Konuşma diyagramı ise katılımcılar arasındaki konuşmaları göstermektedir ve bunları bir işbirliği diyagramı içindeki bir konuşma diyagramı ile tek tek mesaj akışını göstermeden ileri geri hareket edebilen konuşma nesneleri (altıgenler) olarak göstermektedir. Ayrıca süreç mantığı içermemektedir. Bu model türü, farklı kullanıcılar için bir sürecin farklı yönlerini göstermeye izin vermektedir (Ronald, 2011).



Şekil 2.6: Koreografi (solda) ve Konuşma (sağda) Diyagramları Örneği (Onggo, 2014)

Şekil 2.7'de işletme düzeyinde pizza siparişi verilmesi ve alınması sürecinin BPMN diyagramı modelleme örneği görülmektedir. Bu örnekte bir pizza müşterisi ile satıcısı arasındaki etkileşim (B2B düzeyde) işbirliği diyagramı ile modellenmiştir.



Şekil 2.7: B2B düzeyinde bir BPMN örneği (OMG Document, 2010)

İşbirliği diyagramları işletmelerde departmanlar, ekipler, tek başına çalışanlar veya yazılım sistemleri arasındaki etkileşimi modellemek amacıyla da kullanılabilir (OMG Document, 2010).

2.6 BPEL ile Modelleme

İş Süreci Yürütme Dili (Business Process Execution Language, BPEL) Web servislerinde kurumsal iş süreçlerini tanımlamak için kullanılan XML tabanlı bir dildir. Bu modelleme, iş süreçlerinde web servislerinin sırasıyla veya paralel olarak çalıştırıldığı sırayı belirlemektedir. BPEL ile koşullu davranışlar kolayca ifade edilebilmektedir. Örneğin bir web servisinin çağırılması diğer bir çağrının değerine bağlı olabilmektedir. Ayrıca döngüler oluşturmak, değişkenleri çağırmak, değerleri atamak ve kopyalamak, hata işleyicileri tanımlamak gibi farklı yapılar ve görevler birleştirilerek karmaşık iş süreçleri algoritmik bir şekilde tanımlanabilmektedir. Tipik bir BPEL senaryosunda, iş süreci bir istek almakta, bunu gerçekleştirmek için süreç ilgili Web servislerini çağırmakta ve ardından isteğe cevap vermektedir.

BPEL iş süreci “faaliyet (activity)” adı verilen adımlardan oluşmaktadır. İş süreçlerinin adımlarını belirleyen karmaşık algoritmalar tanımlamak için faaliyetler birleştirilebilir (Juric, 2004). BPEL iki farklı iş sürecini desteklemektedir; (1) bir iş etkileşimindeki katılımcının gerçek davranışını modellemek, (2) İçsel davranışları açığa vurmada protokolde yer alan her bir tarafın karşılıklı olarak görünen mesaj değişim davranışını (süreç tanımları) modellemek (<http://www.pnmssoft.com/resources/bpm-tutorial/bpel-and-bpmn-tutorial/>). Ancak BPEL, iş süreçlerini modellerken insanlar arasındaki etkileşimleri gösterememektedir, daha çok servis odaklı mimarilerin (Service-oriented Architecture, SOA) bir parçası olarak kullanılmaktadır. Şekil 2.8’de bazı BPEL faaliyetleri ve bu faaliyetleri temsil eden notasyonlar yer almaktadır.

receive		terminate		flow	
invoke		sequence		forEach	
reply		if		fault handler	
assign		while		event handler	
throw		repeatUntil		comp. handler	
wait		pick			

Şekil 2.8: BPEL faaliyetleri ve şekilleri (Baresi ve Guinea, 2011)

2.7 EPC ile Modelleme

Olay Güdümlü Proses Zinciri (Event-driven Process Chain, EPC), iş süreçlerini ve iş akışlarını tanımlamak için kullanılan diğer bir modelleme dilidir. Süreçlerin operasyon sırasını tanımlamak, süreçleri modellemek, analiz etmek ve iyileştirme potansiyellerini tanımlamak için kullanılabilir. Bu süreçler üretimden müşteri ilişkilerine kadar her türlü süreci kapsayabilir. EPC ile bir işletmenin yaptığı her iş, faaliyetlere ayrılmakta ve bu faaliyetler olaylar aracılığıyla etkileşime girmektedir. Örneğin "Müşteri siparişi gelir" şeklinde bir olay, farklı faaliyet türlerini tetikler ve örneğin "Sipariş teslim edildi" şeklinde başka bir faaliyetle sonlandırılır.

Günümüzde EPC, endüstriyel alanda iş süreçlerini modellemek amacıyla yaygın olarak kullanılmaktadır. Örneğin popüler bir ERP sağlayıcı firma olan SAP, SAP R/3 versiyonunda süreçleri belgelemek için EPC diyagramlarını kullanmıştır. Ayrıca, EPC'nin farklı modelleme araçları (örneğin ARIS, ADONIS, BIC, Bonapart, eDraw, MS Visio, bflow, Semtalk, SmartDraw, Visual Paradigm gibi) tarafından desteklenebiliyor olması yaygın kullanımında etkili olmaktadır. Diğer yandan bu modelleme dili temelde ARIS (ARchitecture of integrated Information Systems, Entegre Bilgi Sistemleri Mimarisi) metodunun bir parçasıdır. ARIS, 1990'larda Saarland Üniversitesinden Profesör August-Wilhelm Scheer ve arkadaşları tarafından başlatılan akademik bir araştırmanın sonucu olarak geliştirilmiştir (Tsironis ve ark., 2009). ARIS metodu, iş süreçlerinin modellenmesi için geliştirilmiş en popüler metodolojilerden biri olarak kabul edilmektedir ve birçok yazılım üreticisi tarafından benimsenmektedir. ARIS, hem bir BPM metodu, hem de kurumsal mimarileri tasarlamak için kullanılan bir çerçevedir. Süreç tasarımı, yönetimi, iş akışını ve uygulama işleyişini bütüncül bir bakış açısı ile analiz edecek yöntemler önermektedir. Asıl amacı iş gereksinimleri ve IT arasındaki boşluğu azaltmaktır. ARIS, kurumsal bir mimarideki karmaşıklığı azaltmak için iş süreçleri, süreçle ilgili ürün ve hizmetler, kuruluşun organizasyon şeması, işletme hedefleri ve stratejileri, bilgi akışı, IT mimarisi, kaynaklar, maliyetler ve benzeri bileşenlere odaklanan çeşitli görüşleri tanımlamaktadır (Pant, 2008). Bu bileşenler iş süreçleri hakkındaki tüm bilgilere yapılandırılmış bir görünüm sağlayan "ARIS Evi" kavramı altında toplanmaktadır. ARIS Evi, işletmelere beş görünüm sunmaktadır (Şekil 2.9):

1. *Süreç görünümü (veya Kontrol Görünümü)*; süreçlerin davranışını, süreçlerin ürün ve hizmetlerle, organizasyonla, işletme fonksiyonları ve verilerle ilişkisini gösteren merkezi bir görünümdür. Bu görünüm seçilen notasyondaki süreç modellerini, bilgi akışını, malzeme akışını, değer zincirlerini, iletişim diyagramlarını içermektedir.

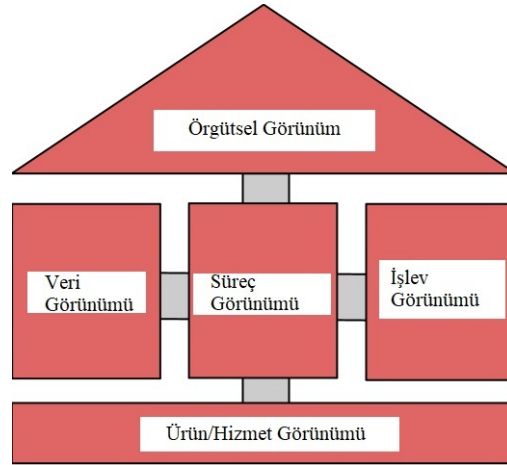
2. *Ürün ve hizmet görünümü*; ürünleri ve hizmetleri, onların yapılarını, ilişkilerini ve ürün/hizmet ağaçlarını göstermektedir.

3. *Örgütsel görünüm*; departmanlar, roller ve çalışanlar da dahil olmak üzere bir işletmenin organizasyon yapısını hiyerarşik organizasyon şemalarında göstermektedir. Organizasyon görünümü aynı zamanda teknik kaynaklar ve iletişim ağlarını da içermektedir.

4. *İşlev görünümü*; süreç görevlerini tanımlamakta, aynı zamanda iş hedeflerini, fonksiyonel hiyerarşileri ve uygulama yazılımını göstermektedir.

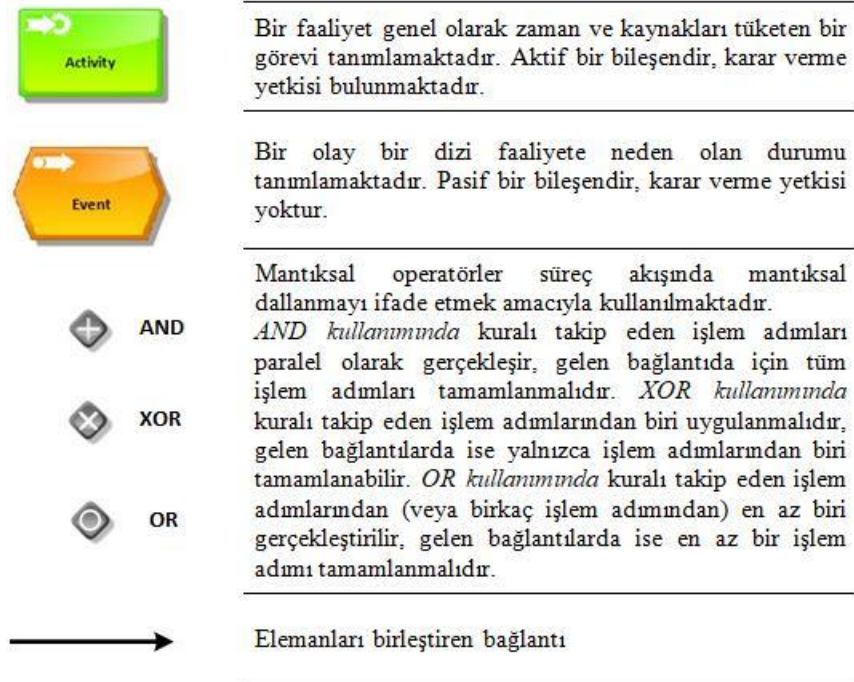
5. *Veri görünümü*; işletme verilerini ve bilgilerini göstermektedir. Bu görünüm ayrıca veri modelleri, veritabanı modelleri, bilgi haritaları ve bilgi yapılarını içermektedir.

ARIS açısından süreçler, işlevlerin mantıksal dizilişi olarak tanımlanabilir. ARIS'de bir veya birden fazla iş hedefini gerçekleştirmeye çalışan bir faaliyet herhangi bir olayın tetiklenmesi ile gerçekleşmektedir. Olaylar, veri görünümü ile işlev görünümünü birbirine bağlamaktadır. Dolayısıyla süreç görünümünü faaliyete geçirmektedirler. Olaylar ayrıca bir sürecin başlangıcında ve bitişinde yer almaktadır. Süreç görünümü bu mantığı kullandığı için bu modele "olay güdümlü süreç zinciri (EPC)" adı verilmektedir. Günümüzde EPC, SAP AG ile Almanya'da bulunan Saarland Üniversitesi işbirliği ile geliştirilerek eEPC adını almış, diğer bir ifade ile EPC başına "e" eki alarak "genişletilmiş olay-güdümlü süreç zinciri" olarak da tanımlanmaktadır. eEPC (extended EPC) içinde EPC'dan farklı olarak; uygulama sistemi, varlık tipi, olay, teknik terimler, işlevler, kariyer bilgisi, örgütsel birim, kişi, konum, girdiler, çıktılar gibi 400'den fazla bağlantı tipi tanımlanabilmektedir. EPC içinde ise sadece olaylar, faaliyetler ve ilişkiler (and, or, xor) tanımlanmaktadır (Seidlmeier, 2004). ARIS, eEPC modelini önermektedir. eEPC diyagramları oluşturmak için de ARIS yazılım platformu kullanılabilir (Tsironis et. al., 2009). Bunun yanı sıra ücretsiz ARIS Express sürümü, sadece EPC modellemesini desteklemektedir. Şekil 2.10'da ARIS Express'teki EPC notasyonları ve işlevleri



Şekil 2.9: ARIS evi

görülmemektedir (Ramona, 2010).



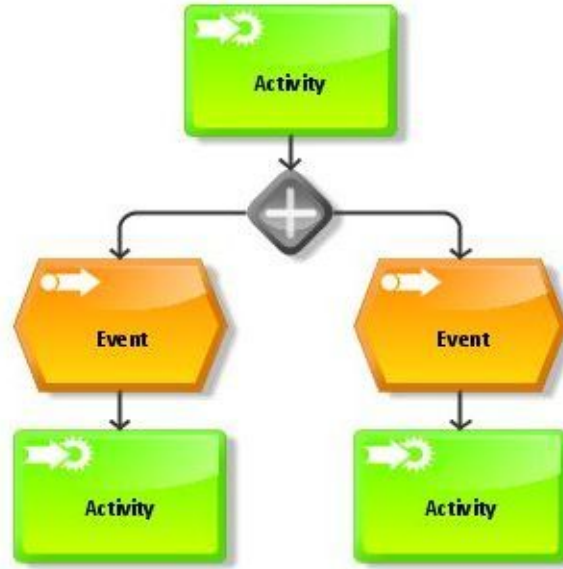
Şekil 2.10: EPC notasyonu örnekleri - ARIS Express (Ramona, 2010)

EPC ile iş süreçlerinin modellenmesi bazı temel kurallar çerçevesinde yürütülmektedir. Bu kurallar şu şekilde özetlenebilir:

- Bir EPC daima bir faaliyet (activity/function) ile başlar ve bir faaliyetle sonlandırılır. Bu kuralı yerine getirdiğimizde belirli faaliyetleri EPC ile modellememiz mümkündür. Buna hiyerarşik modelleme denilmektedir (Şekil 2.11).
- Olaylar (events) ve faaliyetler değişkendir. İki faaliyetin veya iki olayın doğrudan birbirini

takip etmesi mümkün değildir.

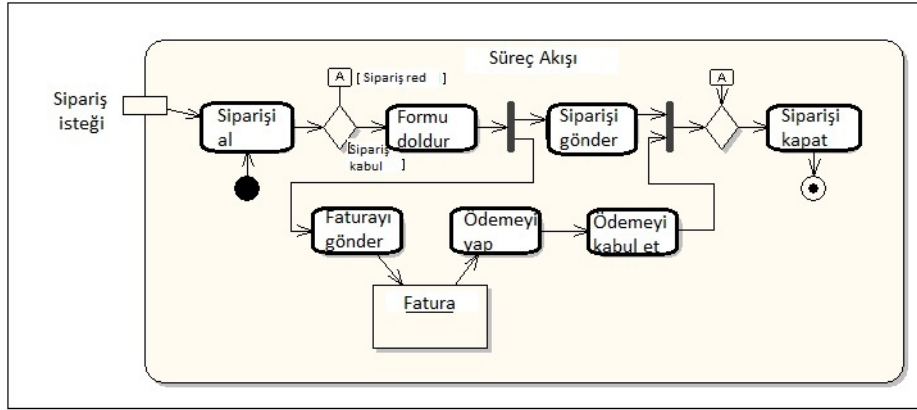
- Sadece bağlantı operatörleri olayları ve faaliyetleri birbirine bağlamaktadır.
- Olaylar faaliyetleri etkinleştirir, faaliyetler olayları oluşturur.
- Bir yolun (path) her dallanması, dallanmayı başlatan aynı türden bir bağlantı operatörü aracılığıyla yeniden birleştirilmelidir.
- OR veya XOR operatörlerinden önce bir olay gerçekleşemez. Sadece işlevlerin karar verme yetkisi bulunmaktadır. Bu nedenle alternatif bir yolun seçimi sadece bir işlev tarafından yapılabilmektedir.
- İşlevlerden ve veri elemanlarından sorumlu organizasyon birimleri, bir işlevin girişinde veya çıkışında olabilir.
- Olaylar bir EPC'yi başlatır veya EPC'nin bir işlevi tarafından tetiklenir. Statik bir durumu tanımlarlar. Organizasyon birimleri veya veri elemanları gibi unsurlarla bağlantı sadece işlevler tarafından gerçekleştirilir. Bunlar sürecin dinamikliğini tanımlamaktadır.



Şekil 2.11: Basit bir EPC Modelleme Örneği (ARIS Express tutorials, 2018)

2.8 UML Faaliyet (Activity) Diyagramları

UML faaliyet diyagramları, iş süreci modellemesi amacıyla kullanılmaktadır. Örneğin; tek kullanımlık bir durum veya use case (kullanım senaryosu) tarafından yakalanan mantığın modellenmesi veya bir iş kuralının ayrıntılı mantığının modellenmesi gibi. UML faaliyet diyagramları, karmaşık bir operasyonun iç mantığını modelleme potansiyeline sahiptir, ancak bir faaliyet şemasına ihtiyaç duymayacak kadar basit bir operasyonun yeniden yazılması daha iyi bir seçenektir. Bu diyagramlar genel olarak akış şemaları (flow charts) ve veri akış diyagramlarının (data flow diagrams) nesne yönelimli eşdeğeri olarak kabul edilebilir (<http://www.agilemodeling.com/>). Bir sipariş isteği için UML faaliyet diyagramı örneği Şekil 2.12'de görülmektedir (<https://sparxsystems.com>).



Şekil 2.12: UML Faaliyet Diyagramı Örneği

2.9 Yapay Zekâ Uygulamalarının İşletmelerde Kullanımı

Yapay zeka, görsel algı, konuşma tanıma, karar verme ve diller arası çeviri gibi normalde insan zekasını gerektiren görevleri gerçekleştirebilen bilgisayar sistemlerinin geliştirilmesi olarak tanımlanabilir (Oxford Dictionary). Ayrıca insan gibi davranan makine oluşturmak için tasarlanmış bir bilgisayar mühendisliği dalı olarak kabul edilmektedir. Günümüzde yapay zekanın iki farklı türü bulunmaktadır: birincisi tek bir görevi akıllıca yürütmek için tasarlanmış sistemleri içermektedir; örneğin; sürücüsüz bir aracı kullanmak gibi, diğeri bilinmeyen sorunlara çözüm bulmak için yeterli zekayı gerektiren sistemleri veya cihazları içermektedir. Bu sistemler ve cihazlar henüz gelişim aşamasındadır.

İlk ticari yapay zeka tabanlı uzman sistem Xcon, 1970'lerin sonunda geliştirilmiştir. Japon metro trenlerinde bulanık mantık (fuzzy logic) teknikleri 1980'lerin başında uygulanmaya başlanmış ve bir çimento üreticisi tarafından üretim uygulamalarında kullanılmıştır. İşletmelerin uzman sistemleri ve sinir ağlarını kullanmaya başlaması ve birçok büyük şirkette yapay zeka gruplarının oluşturulması 1980'lere dayanır. 1980'lerin sonunda, uzman sistemler endüstride giderek daha fazla ilgi görmeye başlar. 1990'larda, otomatik planlama yazılımları, bilgi yönetimi için yazılımlar, otomatik mortgage sigorta sistemleri ve otomatik yatırım karar vericileri gibi yapay zeka uygulamaları kullanılmıştır. Bunun yanı sıra, bir iş için günlük gelirlerin ve personel gereksinimlerinin tahminini iyileştiren yazılımlar, kredi sahtekârlığı tespit sistemleri ve karar destek sistemleri geliştirilir. Veri madenciliği araçları, e-posta filtreleri ve web arşivleme için tasarlanmış crawler gibi uygulamalar 2000'li yıllarda geliştirilmeye başlanır (Nordlander, 2001). 2010'lu yıllarda yapay zekanın yanı sıra makine öğrenmesi ve derin öğrenme, eğitimden sağlık sektörüne, evden sanayiye kadar bir çok alanda uygulama alanı bulmaktadır (Soni ve ark., 2018). Yapay zekanın işletmelerde uygulanabileceği alanlardan bazıları şu şekilde özetlenebilir (Global, 2017):

1. Uzman sistemler (Expert Systems): Bilgisayarların gerçek hayattaki durumlarda karar vermek için programlanabileceği sistemlerdir. Makinelerin, yazılımların ve belirli bilgilerin entegrasyonu ile sistemin son kullanıcıya sonuç, açıklama ve tavsiye vermesini sağlamaktadır.

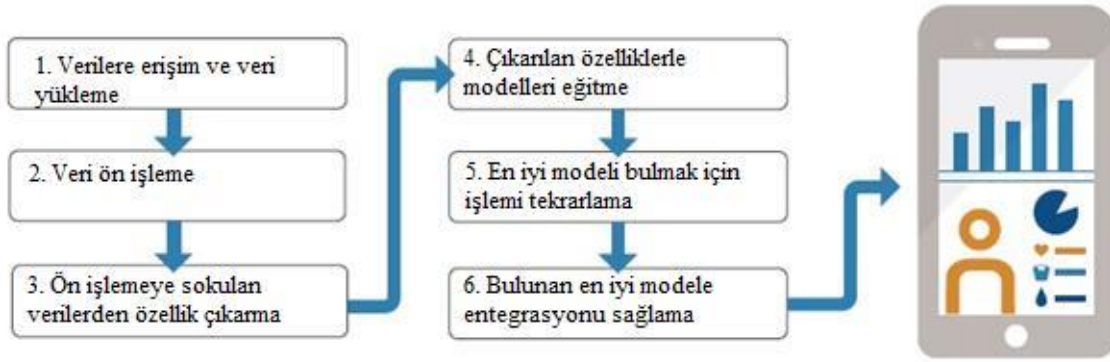
2. Doğal Dil (Natural Language): Chatbot adı verilen bilgisayar programlarının doğrudan bir kullanıcı veya müşteri ile iletişim kurduğunda doğal insan dilini tanıyabildiği yerlerdir.

3. Sinir Sistemleri (Neural Systems): İnsan beyninde meydana gelen fiziksel bağlantı türlerini yeniden üretmeye çalışarak zekayı taklit etmektedir. Örneğin, sinir sistemleri tarihsel verilere dayanarak gelecekteki olayları tahmin edebilir.

4. Robotik (Robotics): Işık, ısı, sıcaklık, ses ve basınç içeren sensör verilerini gören, duyan ve tepki veren bilgisayarlardır.

5. Oyun Sistemleri (Gaming Systems): Stratejik oyunlarda makine, bir insan gibi hareket ederek rakibe karşı etkili bir şekilde karşılık verebilir, bir dizi olası pozisyonu düşünebilir.

Modern yapay zeka uygulamalarının çoğu, 'makine öğrenmesi (Machine Learning)' olarak bilinen bir disiplin ile sağlanmaktadır. Makine öğrenmesi, yapay zekanın temel parçasıdır. Makinelerin verilerdeki kalıpları tespit edip programların davranışlarını bu kalıplara göre ayarlayabilmesi düşüncesine dayanmaktadır (Şekil 2.13). Makine öğrenmesi ile geliştirilebilecek yapay zeka uygulamalarına örnekler şu şekilde verilebilir; bir metni okuyarak müşterinin satın alma siparişi verip vermediğine karar vermek, satın alacağınız ürüne benzer diğer ürünleri sıralamak, görüntüleri tanımak ve içerdikleri öğelere göre sınıflandırmak, büyük hacimli metinleri gerçek zamanlı olarak çevirmek (Google Çeviri gibi), yüzlerin, konuşmaların kime ait olduğunu belirlemek, nesnelere tanımlamak.



Şekil 2.13: MATLAB'ta Makine Öğrenmesi İş Akışı- (<https://www.mathworks.com/>)

Diğer yandan derin öğrenme makine öğrenmesinin bir alt alanıdır. Derin öğrenme, makine öğrenmesinde GPU'ları (Graphical Processing Unit, Grafik İşlemci Ünitesi) kullanmaktadır. GPU, çok büyük miktarlarda eğitim verisi içermektedir, GPU yardımıyla güçlü ve verimli paralel hesaplama yapılabilmektedir. Derin öğrenme sayesinde sinir ağları teknik araçlarla taklit edilebilmektedir.

112 ülkeden ve 21 farklı sektörden 3.000'den fazla üst düzey yönetici, yönetici ve analist, ve 30'dan fazla teknoloji uzmanı ile yapılan görüşmelere dayanan küresel bir araştırmaya göre (Ransbotham ve ark., 2017), 20 işletmeden sadece biri, yapay zekayı iş süreçlerine kapsamlı bir şekilde dahil etmiştir. Diğer yandan tüm işletmelerin % 39'undan azının bir yapay zeka stratejisi bulunmaktadır. Bu araştırmaya göre, yapay zekanın fonksiyonel olarak en çok kullanıldığı sektörler; teknoloji, medya, telekom, tüketim, finansal hizmetler, serbest meslek hizmetleri, sağlık hizmetleri, endüstri, enerji ve kamu sektörüdür.

Günümüzde sağlık, eğitim, ulaştırma, enerji, işletme başta olmak üzere birçok sektördeki sistemlerin ve iş süreçlerinin kalitesini ve verimliliğini artırmak amacıyla yapay zeka kullanılmaktadır. Yapay zeka teknolojileri işletmelere, yeni pazarlara açılma fırsatı sunmakta ve rakiplerine karşı rekabetçi bir avantaj sağlamaktadır. Yapay zeka fiziksel, dijital ve biyolojik dünyayı birbirine bağlayan yeni teknolojilerin geliştirilmesinin (Endüstri 4.0) temel itici gücü olarak da kabul edilmektedir (Skilton ve Hovsepian, 2017). Diğer yandan yapay zeka teknolojilerinin günümüz teknolojileri ile çözemeyeceğimiz problemlere çözüm üretmesi beklenmektedir. Bu nedenle yapay zekanın önümüzdeki yıllarda iş, ekonomi ve toplum üzerinde artan bir etkiye sahip olması beklenmektedir (Skilton, 2017).

BPM, aynı zamanda iş süreçlerinin iyileştirilmesine odaklanan bir alandır. İş süreçlerinin daha verimli hale getirilmesi ve son kullanıcıların değişen ihtiyaçlarına yanıt vermesi gerekmektedir. Yapay zekanın BPM alanında kullanımı bu alandaki bir sonraki yıkıcı teknoloji olarak düşünülebilir. Çünkü günümüzde yapay zeka özellikle manuel olarak yapılan bir çok faaliyetin tam otomasyonunu mümkün kılmaktadır. İşletmelerde yapay zekanın kullanıldığı alanlar şu şekilde özetlenebilir:

1. Müşteri hizmetleri - Müşterilere gerçek zamanlı destek sağlamak için (örneğin, faturalandırma, siparişte yaşanan sıkıntıları çözüme gibi) sanal asistanlık programlarının kullanılması.

2. İş yüklerini azaltmak – Makineler üzerindeki akıllı sensör verileri yardımıyla servis/bakım isteklerini otomatik olarak yönlendirmek.

3. Lojistik optimizasyonu - Yapay zeka destekli görüntü tanıma araçları kullanarak lojistik altyapısını izlemek, optimize etmek, ulaşım rotalarını planlamak.

4. Üretim çıktılarını ve verimliliği arttırmak - Endüstriyel robotları iş akışına entegre ederek üretim hattını otomatikleştirmek.

5. Kesintileri önlemek - bir IT kesintisi gibi işinizi kesintiye uğratması muhtemel kalıpları tanımlamak için algılama teknikleri kullanmak. Özel bir yapay zeka yazılımı, sisteminize izinsiz girişleri tespit etmenize yardımcı olabilir.

6. Davranış tahmin etmek - Çevrimiçi davranış kalıplarını analiz etmek veya özel ürün teklifleri sunmak, kredi kartı sahtekarlığını tespit etmek, kullanıcı davranışlarına uygun reklamları hedeflemek.

IBM Watson (<https://www.ibm.com/watson/>), günümüz iş dünyasında kullanılan güçlü bir yapay zeka sistemidir. Bu sistem insan beyninin yapısından esinlenilerek tasarlanmıştır. Watson, insan beynine benzer şekilde zaman içinde veriler toplamakta ve bu verilerden elde edilen çıktılarına dayalı kararlar almaktadır. Bu sistem, toplanan verileri katmanlara ayırarak, verilerdeki kalıpları tanımlamakta ve önceki çıktılarına dayanan olası seçenekleri değerlendirmektedir. Watson bu şekilde bilinçli seçimler yapabilmektedir. Bu tür veri toplama ve analiz şekli, makine öğrenimi veya derin öğrenme olarak da adlandırılmaktadır. Apple'ın iPhone / iPad / iPod platformlarında kullanılan Siri uygulaması kişisel bir asistandır. Bulut ortamında yapay zekayı kullanarak doğal dili kullanan, temel soruları anlayabilen ve yanıtlayabilen bir uygulamadır. Siri, geliştirilen ilk yapay zeka ajanlarından biridir, ancak yetenekleri açısından gerçek yapay zeka sistemlerinden beklenenlerle karşılaştırıldığında temel düzeyde bir uygulamadır (Marinchak ve ark., 2018).

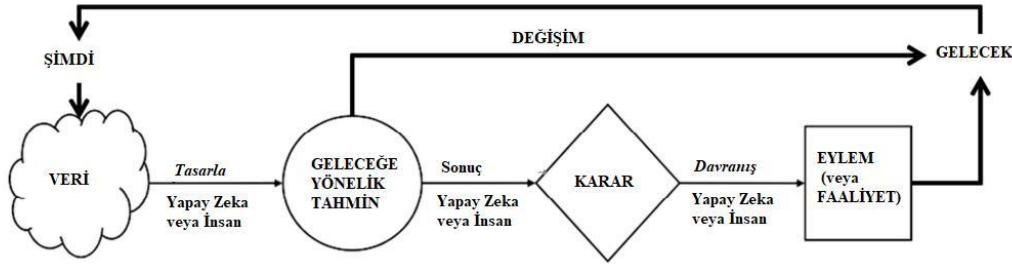
İşletmelerde farklı amaçlar için kullanılacak bir çok yapay zeka uygulaması bulunmaktadır. Bunlardan bazıları şu şekilde sıralanabilir (<https://60secondmarketer.com/>): (1) Sosyal medya kampanyalarını destekleyen uygulamalar; Rocco, Conversocial, Converseon, Insightpool, Sysomos, Unmetric, Crimson Hexagon, (2) Sizinle konuşmak için yapay zekayı kullanan chatbotlar; Botsify, Botsociety, Brainbot, Brainshop, Pandorabots, FlowAI, Flowxo, Gupshup, Botmother, Talla, (3) Daha iyi içerik üretmenize yardımcı olan yapay zeka platformları; Concured, Find the Ripple, Phrasee, BloomReach, Idio, Automated Insights Wordsmith, Twizoo, Narrativ, (4) Müşteri deneyiminizi geliştirmek için yapay zekayı kullanan CRM platformları; Persado, Tact AI, Getbase, Kluster, People AI, Growbots, Insidesales, Cognitivescale, (5) Yapay zekayı kullanan pazarlama ve reklam platformları; Aritic Pinpoint, Appier, GetResponce, Oracle Marketing Cloud, Kahuna, 4INFO, Drawbridge, (6) Yapay zeka içeren arama motoru optimizasyonu platformları; BrightEdge, MarketBrew, Acrolinx, Onspot, MatrixSEO, Can I rank, Wordlift, (7) Yapay zeka destekli web site tasarımı; b12, Adobe Sensei, Firedrop, Sentient Ascend, The Grid, Wix ADI, (8) Yapay zeka destekli imaj tanıma ve analiz platformları; Emotient, Amazon Rekognition, Google Cloud Vision API, Clarifai, Apple Vision, (9) Yapay zekayı kullanan iş zekası ve iş analitiği platformları; Crowdfower, Rapidminer, Tamr, Context Relevant, Datarobot, Paxata, Trifacta, Dataminr, Video Agnostic, Datorama, Lucy,

(10) Yapay zeka ile çalışan davranış analitiği platformları; Cogito, Afiniti, Altocloud, DSCVRY, Leadcrunch, Watson.

2.10 Yapay Zekâ Uygulamalarının Modellenmesi

İş süreçleri, belirli bir iş çıktısı almak, belirli bir hedefe ulaşmak veya bir müşteriye değer sağlamak için tasarlanmış birbirleriyle ilişkili faaliyetler kümesi olarak tanımlanabilir. Yaratıcı düşünme, öğrenme, ticaret, üretim gibi insanlar tarafından yürütülen herhangi bir faaliyet, iş süreci olarak modellenebilmekte veya analiz edilebilmektedir.

Bir iş süreci genellikle bir olayla başlatılmaktadır; örneğin banka müşterisinin bir ipotek kredisi talebinde bulunması gibi ve bir takım veriler kullanılmaktadır; örneğin müşterinin kredi geçmişi verileri gibi. Veriler iş sürecini hedefine doğru yönlendirmektedir; örneğin krediyi onaylamak veya reddetmek gibi. Günümüzün iş süreçleri birçok farklı veri kaynağını kullanabilmektedir. Bu veriler bir modelle eşleştirilerek, toplanır, analiz edilir, yorumlanır, değerlendirilir ve geleceğe dair tahminlerde bulunulur. Örneğin, ipotek onay sürecinde bir müşterinin borcunu ödeyebilme ihtimalini tahmin etmeye çalışır. Tahmine dayalı bir karar verilir ve bir veya birkaç eylem gerçekleştirilir. Örneğin, müşterinin kredi skoru doğru tahmin edilirse, kredi onaylanır, müşteriye bir hesap açılır ve para transfer edilir (Koehler, 2018). Şekil 2.14'teki modelde yapay zeka teknolojilerinin kullanıldığı bir iş sürecinde gerçekleştirilen ana faaliyetleri görülmektedir. Bu modelde bir süreç, geleceği tahmin etmek için verileri kullanmaktadır. Bir kararın verilmesinin temelinde tahminler bulunmaktadır, kararlar sonucu bazı eylemler gerçekleştirilerek bir insan veya yapay ajanın nasıl davranacağı belirlenmektedir. Bu arada süreç sonunda değişen veriler yeniden sürece dahil edilerek daha iyi kararlar alınması sağlanmaktadır.



Şekil 2.14: Yapay Zeka Teknolojileri ile Bir İş Süreci Tasarım Modeli (Koehler, 2018)

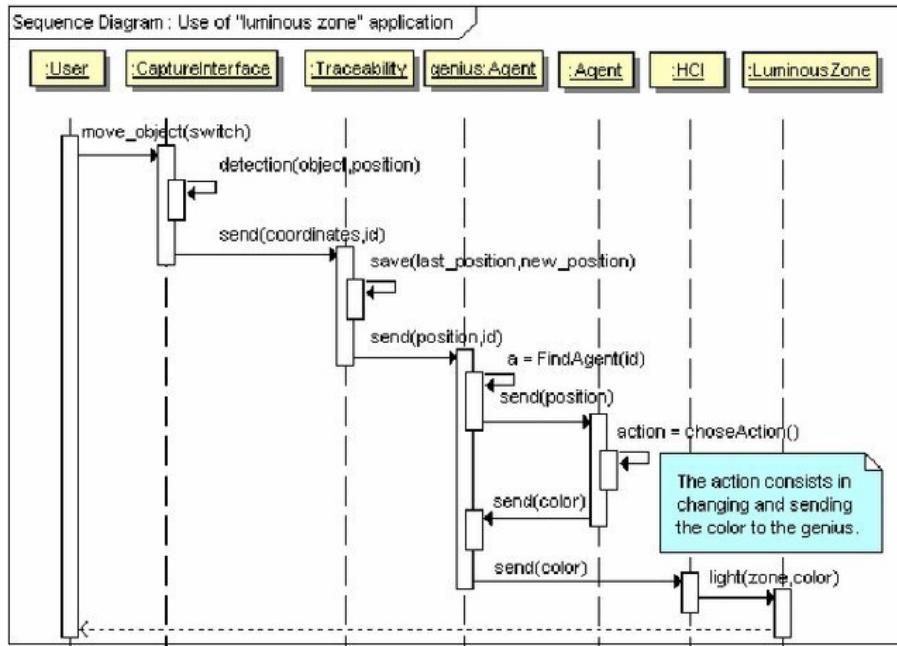
Sonuç olarak yapay zeka geleceğe yönelik tahmin, karar ve eylem döngüsünde hareket etmektedir. Belirli verilerin tasarımı geleceğe yönelik tahminlere, kararların alınması elde edilen sonuçlara ve bazı eylemlerin gerçekleşmesi davranışlara bağlıdır. Gelecekte bu eylemler karmaşık yollarla iç içe geçmektedir ancak bu eylemleri ayırmak ve onları daha iyi ayırt etmek, bunları destekleyen, otomatikleştiren teknolojileri geliştirmeyi gerektirmektedir. Örneğin, makine öğrenmesi, karar teorisi, fayda teorisi ve arama algoritmaları yapay zekayı destekleyen teknolojileridir. Şekil 2.14'deki yapay zeka ile iş süreci tasarım modelinde bu teknolojilere şu şekilde yer verilebilir (Koehler, 2018):

1. *Makine Öğrenmesi* - veriden tahmine: Makine öğrenmesi yöntemleri bir iş sürecinde geleceğe yönelik tahminlerde bulunmak için mevcut verilere uygulanabilir. Yapay zeka teknolojisi ile günümüzde, hazır kütüphanelerde mevcut olan sinir ağları veya karar ağaçları gibi birçok farklı makine öğrenme algoritması geliştirilmiştir.

2. *Karar teorisi ve Fayda Teorisi* – tahminden karara: Yapay zeka, bir ajanın fayda fonksiyonunu en üst düzeye çıkarmak ve gerçekleştirilmesi gereken eylemleri belirlemek için karar ve fayda teorisini incelemektedir. Karar teorisinin kökleri ekonomiye dayanır ve belirsizlik gerektiren durumları, riskleri incelemektedir çünkü eylemlerin sonuçları belirsizdir.

3. *Arama algoritmaları* – karardan eyleme: Arama algoritmaları, matematik ve yapay zekayı birleştirerek (özellikle yönelem araştırmaları), başarılı bir pazarlama kampanyasında hangi eylemlerin gerçekleştirileceği, planlama ve çizelgeleme problemlerine en uygun çözümleri bulmak, bir ürünü en iyi şekilde üretmek için bir dizi makinede belirli bir sayıda işin nasıl planlanacağı veya bir zamanlanmış siteler kümesinde malların en uygun şekilde gönderilmesi için bir turun nasıl planlanacağı gibi en iyi çözümü bulmaya odaklanmaktadır. Yapay zeka teknolojileri ile günümüzde güçlü optimizasyon ve problem çözme teknikleri dahil birçok farklı arama tekniği geliştirilmiştir.

İş süreçlerini modellemek amacıyla kullanılan araçlar, yapay zeka destekli iş uygulamalarını modellemek amacıyla da kullanılmaktadır. Örneğin Kubicki ve ark. (2009) tarafından geliştirilen Luminous Zone adlı bir uygulamada RFID etiketleriyle donatılmış masa etrafında birçok katılımcı fiziksel nesnelere (kalem, kitap gibi) kullanarak uygulamalarla (tasarım veya üretim görevleri, oyunlar, vb. gibi) etkileşimli olarak çalışabilmektedir. Bu uygulama ile birlikte masanın kullanımına yönelik bir senaryo Şekil 2.15'te bir UML diyagramı olan sequence diyagram ile modellenmiştir.

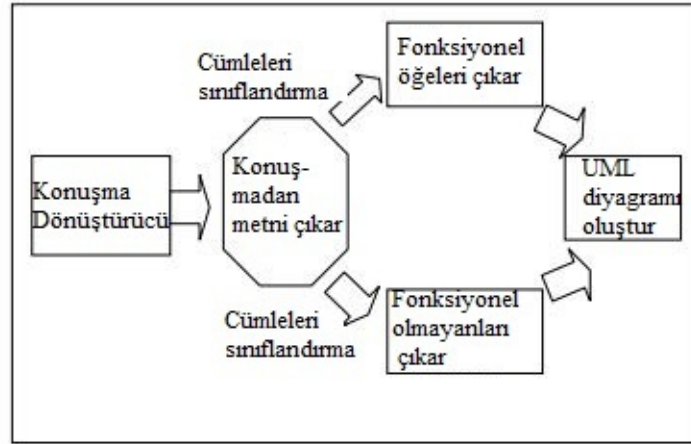


Şekil 2.15: Bir iş sürecinin UML ile modellenmesi örneği (Kubicki ve ark., 2009)

Diğer yandan Kumar (2009) tarafından geliştirilen metni konuşmaya dönüştürme uygulamasında kullanılan Şekil 2.16'daki mimari, UML diyagramlarına konuşma üreten bir sistemin modelini göstermektedir.

2.11 Sonuç

İşletmeler küresel dünyada rekabetçi avantaj sağlamak için iş süreçlerini yeniden yapılandırma ihtiyacı duymaktadır. Bu sürece destek olan ERP sistemleri; planlama, üretim, satış, pazarlama,



Şekil 2.16: UML diyagramlarına konuşma üreten bir sistem modeli örneği (Kumar, 2009)

dağıtım, muhasebe, finans, insan kaynakları yönetimi, proje yönetimi, envanter yönetimi, servis ve bakım, ulaşım gibi işletmelerin fonksiyonel alanlarını destekleyen modülleri kapsayan yazılım sistemleridir. Bu sistemler, işletmelerin çoğunlukla fonksiyonel anlamda birbiriyle uyumsuz ve dağınık olan eski bilgi sistemlerinin tek bir entegre sistem altında toplanması yoluyla oluşturulmaktadır. İşletmelerdeki iş süreçlerinin yeniden yapılandırılma süreci, işletmeler için olduğu kadar yazılım geliştiriciler için de zorlu bir süreçtir. Yazılım geliştiriciler, işletmeler için oluşturacakları sistemin, tasarım ve geliştirme sürecinde işletmelerin mevcut iş süreçlerini nasıl yürüttüklerini bilmek ve buna göre yeniden yapılandırmak durumundadırlar. İş süreci modelleme yaklaşımları, yeniden yapılandırma sürecinde yazılım geliştiricilere destek olmaktadır.

Diğer yandan işletmelerde yapay zeka uygulamalarının kullanımı son yıllarda gittikçe artmaktadır. Ancak bir işletmede bir yapay zeka uygulamasının geliştirilmesi büyük bir çaba (iş süreçlerine entegrasyon zorlukları ve son teknoloji sistemlerin anlaşılması, mevcut kullanılan sistem ve platformlarla kullanılabilirlik ve birlikte çalışabilirlik), yüksek bir maliyet (akıllı teknolojilerin oluşturulması, karmaşık doğası, güncellenmesi, bakımı) ve zaman (uzun implementasyon süreleri) gerektirmektedir. Bu çabanın, maliyetin ve zamanın asgari düzeyde optimizasyonu için iş süreçlerinin etkin bir şekilde modellenmesine ihtiyaç duyulmaktadır. Yapay zeka uygulamalarının iş süreçlerine dahil edilebilmesi için bir işletmedeki süreç yapısının iyi analiz edilmesi gerekmektedir. İş süreçlerinin modellenmesi sayesinde bir işletmedeki iş analistleri, yazılım geliştiriciler ve işletme yöneticisi arasında ortak bir dil geliştirilmektedir. Bu bölümde bu dilin temel kuralları hakkında temel nitelikte bilgi sunulmuştur. Bu bölümün üniversitemizin bilgisayar mühendisliği, yazılım mühendisliği, endüstri mühendisliği, yönetim bilişim sistemleri bölümlerinde okuyan öğrencilere, ders veren akademisyenlere temel nitelikte kaynak oluşturması açısından yararlı olması beklenmektedir.

2.12 Kaynakça

ARIS Express tutorials. Erişim adresi: <http://www.ariscommunity.com/aris-express/tutorials> (04.10.2018)

Baresi, L., & Guinea, S. (2011). Self-supervising bpel processes. *IEEE Transactions on Software*

Engineering, 37(2), 247-263.

Basics for Planning of Logical Systems. Erişim adresi: https://gc21.giz.de/ibt/en/opt/site/ilt/ibt/regionalportale/sadc/inhalt/logistics/module_03/rules_for_modelling_epcs.html (04.10.2018)

Béleczi, A., & Molnár, B. (2016). Modeling framework for designing and analyzing document-centric information systems based on HypergraphDB. In: E. Vatai (ed.), Proceedings of the 11th Joint Conference on Mathematics and Computer Science, Eger, Hungary, 20th – 22nd of May, 2016, published at <http://ceur-ws.org>

Bochman, G. (2008). Course Notes- Introduction to Petri Nets. Erişim adresi: <http://www.site.uottawa.ca/~bochmann/ELG7187C/CourseNotes/BehaviorModeling/Petri-nets/index.html> (04.10.2018)

Doğan, K., & Arslantekin, S. (2016). Büyük Veri: Önemi, Yapısı ve Günümüzdeki Durum. DTCF Dergisi, 56(1).

Event-driven process chain (EPC). Erişim adresi: <https://www.ariscommunity.com/event-driven-process-chain> (04.10.2018)

Gelogo, Y. E., & Kim, H. K. (2014). Mobile integrated enterprise resource planning system architecture. International Journal of Control and Automation, 7(3), 379-388.

Global, I. (2017). Artificial Intelligence (AI): What is it and how does it work? Erişim adresi: <https://www.lexology.com/library/detail.aspx?g=5424a424-c590-45f0-9e2a-ab05daff032d> (13.10.2018)

Pinckaers, F., Gardiner, G., & Van Vossel, E. (2011). Open ERP, a modern approach to integrated business management. Open Source management Solution.

Purohit, G. N., Jaiswal, M. P., & Pandey, S. (2012). Challenges involved in implementation of ERP on demand solution: Cloud computing. International Journal of Computer Science Issues (IJCSI), 9(4), 481.

Rashid, M. A., Hossain, L., & Patrick, J. D. (2002). The evolution of ERP systems: A historical perspective. In Enterprise Resource Planning: Solutions and Management (pp. 35-50). IGI Global.

Juric, M. B. (2004). A Hands-on Introduction to BPEL. Erişim adresi: <https://www.oracle.com/technetwork/articles/matjaz-bpel1-090575.html> (04.10.2018)

Koehler, J. (2018). Business Process Innovation with Artificial Intelligence: Levering Benefits and Controlling Operational Risks, European Business & Management, 4(2), 55-66. doi: 10.11648/j.ebm.20180402.12

Koslowski, T., & Strüker, J. (2011). ERP on demand platform. Business & Information Systems Engineering, 3(6), 359-367.

Kubicki, S., Lepreux, S., Lebrun, Y., Dos Santos, P., Kolski, C., & Caelen, J. (2009, July). New human-computer interactions using tangible objects: Application on a digital tabletop with RFID technology. In International Conference on Human-Computer Interaction (pp. 446-455). Springer, Berlin, Heidelberg.

Kumar, B. V. D. (2009). Speech to UML: An Intelligent Modeling Tool for Software Engineering. Computer Science Letter, 1(2), 1-7.

Marinchak, C. M., Forrest, E. Hoanca, B. (2018). The Impact of Artificial Intelligence and Virtual Personal Assistants on Marketing. In Mehdi Khosrow-Pour, D. B. A. (Eds.) Encyclopedia of Information Science and Technology, Fourth Edition. IGI Global: USA.

Nordlander, T. E. (2001). AI Surveying: Artificial Intelligence in Business. Department Of Management Science And Statistics De Montfort University.

Notation, B. P. M. (2006). Specification, Final Adopted Specification. Technical report, Object Management Group (OMG).

OMG Document (2010). BPMN 2.0 by Example, Version 1.0 (non-normative). Object Management Group, Inc. Erişim adresi: <https://www.omg.org/cgi-bin/doc?dtc/10-06-02> (28.09.2018)

Onggo, B. S. (2014). Agent-based simulation model representation using BPMN. In Formal languages for computer simulation: Transdisciplinary models and applications (pp. 378-400). IGI Global.

Pant, K. (2008). Business Process Modeling. Erişim adresi: <https://www.packtpub.com/books/content/business-process-modeling> (04.10.2018)

Ransbotham, S., Kiron, D., Gerbert, P., & Reeves, M. (2017). Reshaping Business With Artificial Intelligence: Closing the Gap Between Ambition and Action. MIT Sloan Management Review, 59(1).

Ramona, B. (2010). Basic rules of EPC modelling. Erişim adresi: <https://www.ariscommunity.com/users/rbaureis/2010-03-22-basic-rules-epc-modelling> (04.10.2018)

Ronald W. (2011). Learning BPMN 2 - Which models are available in BPMN?. Erişim adresi: <http://www.ariscommunity.com/users/roland-woldt/2011-01-28-learning-bpmn-2-which-models-are-available-bpmn> (04.10.2018)

Seidlmeier, H. (2004). Process Modeling with ARIS: A Practical Introduction. Springer Science & Business Media.

Satell, G. (2017). Mapping innovation: a playbook for navigating a disruptive age. McGraw-Hill Education.

Skilton, M., & Hovsepian, F. (2017). The 4th Industrial Revolution: Responding to the Impact of Artificial Intelligence on Business. Springer.

Skilton, M. (2017). The Impact of Artificial Intelligence on Business. Warwick Social Sciences Policy Briefing. Erişim adresi: https://warwick.ac.uk/fac/soc/impact/policybriefings/impact_of_ai_on_business_policybrief_final.pdf (13.10.2018)

Soni, N., Sharma, E., Singh, N., Kapoor, A. (2018). Impact of Artificial Intelligence on Business. International conference on Digital Innovations, Transformation, and Society Conference 2018 (Digits 2018), New Delhi, India.

Tsironis, L., Anastasiou, K., & Moustakis, V. (2009). A framework for BPML assessment and improvement: a case study using IDEF0 and eEPC. Business Process Management Journal, 15(3), 430-461.

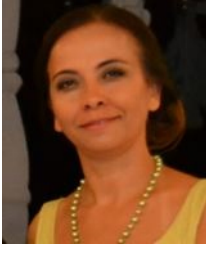
UML 2 Tutorial - Activity Diagram. Erişim adresi: https://sparxsystems.com/resources/uml2_tutorial/uml2_activitydiagram.html (04.10.2018)

Williams, T. (2013). Specification, D. Workflow Management within the ARIS Framework. Erişim adresi: http://www.pera.net/Methodologies/ARIS/ARIS_Paper_by_Ted_Williams.html (04.10.2018)

Wohed, P., van der Aalst, W. M., Dumas, M., ter Hofstede, A. H., & Russell, N. (2006, September). On the suitability of BPMN for business process modelling. In International conference on business process management (pp. 161-176). Springer, Berlin, Heidelberg.

Zur Muehlen, M., & Recker, J. (2013). How much language is enough? Theoretical and practical use of the business process modeling notation. In Seminal Contributions to Information Systems Engineering (pp. 429-443). Springer, Berlin, Heidelberg.

2.13 Yazarlar Hakkında



Öğr. Gör. Gülay EKREN lisans eğitimini 2003 yılında Ege Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Öğretmenliği bölümünde tamamladı. 2003-2008 yılları arasında Milli Eğitim Bakanlığına bağlı ilk ve orta dereceli okullarda öğretmenlik yaptı. 2014 yılında Gazi Üniversitesi Bilişim Enstitüsü Yönetim Bilişim Sistemleri yüksek lisans ve ardından 2016 yılında Anadolu Üniversitesi Sosyal Bilimler Enstitüsü Uzaktan Öğretim (Online) tezsiz yüksek lisans programlarını bitirdi. Şu anda Sakarya Üniversitesi Sosyal Bilimler Enstitüsü Yönetim Bilişim Sistemleri Anabilim dalında doktora eğitimine devam etmektedir. Aynı zamanda 2009 yılından bu yana Sinop

Üniversitesi Ayancık Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümünde öğretim görevlisi olarak çalışmaktadır. İlgili alanları, açık ve uzaktan eğitimin yönetim, teknoloji ve öğrenme alanları, öğrenme yönetim sistemleri, veritabanı yönetim sistemleri, eğitimde ve endüstride Arttırılmış Gerçeklik uygulamaları, Endüstri 4.0, iş süreçlerinin modellenmesi, ERP sistemleri olan Ekren, bu konularda araştırmalar yapmaktadır.

Posta adresi: Sinop Üniversitesi Ayancık Meslek Yüksekokulu, Ayancık, Sinop, Türkiye 57400
Tel (İş): +90 368 6133436 / Dahili: 6915

E-posta: gekren@sinop.edu.tr



Prof. Dr. Erkollar, yönetim bilişim sistemleri alanı profesörüdür. Uzmanlık alanları; yönetim bilişim sistemleri, Endüstri 4.0 ve akıllı fabrikalar, üretim yönetimi, teknoloji yönetimi, iş sistemlerinin modellenmesi, Veri yönetimi ve analizi, İş zekası ve simülasyon gibi konuları içermektedir. Son yayınlarından bazıları 'Flextrans 4.0 - Akıllı Şehirler için Akıllı Lojistik', 'Sanayi 4.0: Büyük Veri Devrimleri Akıllı Teknolojiler Gerekliyor' ve 'Sürdürülebilir Şehirlere Akıllı Ulaşım Gerekliyor: Endüstri 4.0 Taşımacılık Matrisi'. Prof. Erkollar, birçok kitabın yazar ve editörlüğünü, 200'den fazla hakemli makalenin ve

13 uluslararası derginin yayın kurulu üyeliğini yapmıştır. Yakın zamanda editörlüğünü üstlendiği "Enterprise and Business Management - Industry 4.0" adlı kitabı, uluslararası EBM Serisinde 2018 de yayınlanacaktır.

Posta adresi: Sakarya Üniversitesi, İşletme Fakültesi, Yönetim Bilişim Sistemleri, Serdivan, Sakarya, Türkiye 54050

E-posta: erkollar@sakarya.edu.tr

3. Veri Görselleştirme

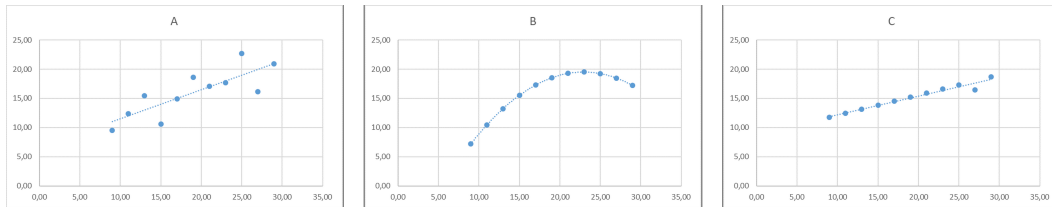
Veri Görselleştirme Ve Görsel Analitik Tableau ile Örnek Bir Uygulama

Halil İbrahim CEBECİ

3.1 Giriş

Forbes tarafından yayınlanan bir makalede her gün dünyada 2500 katrilyon byte veri üretilmektedir. Bu akıl almaz değer yıllar içerisinde süregelen bilişim alanındaki hızlı gelişimin bir sonucudur. Özellikle nesnelere interneti teknolojisinin ve açık veri inisiyatiflerinin etkisiyle bu veri üretim hızını çok daha fazla arttıracak da muhakkaktır. Bu verilerin bütünleştirilmesi, özetlenmesi ve kavramsallaştırılması için klasik tablolar teknikleri yerine veri görselleştirme biliminin tercih edilmesi, özellikle kullanıcıların teknik ve istatistiksel bilgi seviyelerine bağlı olarak son yıllarda yükselen bir eğilim olarak karşımıza çıkmaktadır.

Sayıları yorumlamak için harcanan çaba ve zaman düşünüldüğünde çoğu zaman uzun özetlenmiş tablolar yerine basit bir görsel çok daha iyi sonuçlar verebilir (Yuk and Diamond, 2014). Aşağıdaki Tablo 1 de üç farklı değişken arasındaki ilişkiyi göstermektedir (Wexler, Shaffer & Cotgreave, 2017). Bu ilişkinin sadece tablolara bakılarak kolayca anlaşılacağı açıktır.



Şekil 3.1: Bağımlı-Bağımsız Değişken Arasındaki İlişkinin Anlaşılması

Tablo 1 de herhangi bir trendin varlığının anlaşılması çokta kolay değilken, şekil 1 de doğrusal

ve parabolik trendler görseller yardımıyla çok açık bir şekilde ortaya konmuştur (Kirk, 2012).

Tablo 3.1: Bağımlı-Bağımsız Değişken Arasındaki İlişkinin Anlaşılması

A		B		C	
x	y	x	y	x	y
21.00	17.08	21.00	19.28	21.00	15.92
17.00	14.90	17.00	17.28	17.00	14.54
27.00	16.16	27.00	18.48	27.00	16.48
19.00	18.62	19.00	18.54	19.00	15.22
23.00	17.66	23.00	19.52	23.00	16.62
29.00	20.92	29.00	17.20	29.00	18.68
13.00	15.48	13.00	13.26	13.00	13.16
9.00	9.52	9.00	7.20	9.00	11.78
25.00	22.68	25.00	19.26	25.00	17.30
15.00	10.64	15.00	15.52	15.00	13.84
11.00	12.36	11.00	10.48	11.00	12.46

Veri görselleştirmenin önemi ortada iken hazırlanan bu çalışmada veri görselleştirme ve görsel analitik kavramlarının tanıtılması ve iş zekâsı – yapay zekâ eksenindeki konumun belirlenmesi amacıyla hazırlanan bu bölümde, Tableau iş zekâsı yazılımı ile bir gösterge paneli uygulaması konuyu desteklemek adına gerçekleştirilecektir.

3.2 Veri Görselleştirme

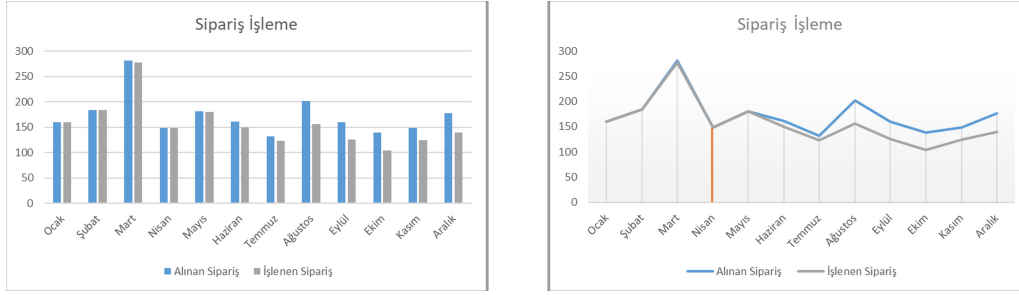
Veri görselleştirmenin en temel tanımı yaratıcı ve sanatsal bir bakış açısı ile büyük miktardaki verinin kolaylıkla anlaşılabilir görsellere çevrilmesi bilimidir (Yuk and Diamond, 2014). Tanımdan hareketle, veri görselleştirme biliminin öncelikli görevi yığın halindeki verilen kullanıcılara en uygun şekilde özetlenmesidir. Fakat son yıllarda özellikle iş zekâsı alanındaki gelişmeler ve büyük veri çalışmaları, toplanan yüksek boyuttaki verinin analitik amaçlar için ön işlenmesi süreçlerinin önemini arttırmaktadır. Veriyi anlamakla başlayan bu ön işleme süreçlerinde veri görselleştirme bilimi karşılaştırma, yorumlama ve analiz gibi bilişsel süreçlerde önemli rol oynamaktadır.

Veri görselleştirme kullanmanın arkasında aşağıdaki motivasyonların olduğu söylenebilir (Miller, 2017; Simon, 2014).

- Analiz sonuçlarını daha hızlı ve kolay bir şekilde okuyup yorumlanabilmesi,
- Vurgulamalar ve yönlendirmelere ile kullanıcılara sonuçların daha iyi anlatılabilmesi,
- Belirli bir problem alanına uygun çözüm üretebilme,
- Klasik yöntemler ile gözden kaçabilecek görünmez (aykırı) değerlere vurgu yapılabilmesi,
- Görsel analitik yaklaşımı ile verini sonraki zaman dilimlerindeki halinin tahmin edilebilmesi
- Tabular tekniklerle ortaya konamayacak farklı bakış açıları kazandırabilmesi

Verileri görselleştirirken her zaman için sofistike ve teknik grafikler en uygun çözüm olmayabilir. Görselleri sadece yazılım çıktısı olarak düşünmeyip, sürecin gereksinim analizi, amaç belirleme,

kullanıcı gruplarının tanımlanması, tasarım ve geri besleme adımlarını da içerdiği göz önüne alınmalıdır. Bazı durumlarda basit bir statik grafik ilgilenilen problem alanı ile ilgili yeterli ölçekteki bilgiyi efektif bir şekilde sunarken, karmaşık ve dinamik grafikler odak noktadan uzaklaştırarak amaca hizmet etmeyebilir (Kirk, 2012; Yuk and Diamond, 2014). Şekil 3.2 de hatalı tasarlanmış bir görsel ile bu görselin doğru tasarımı birlikte sunulmuştur (Knaffic, 2015). İlk tasarımda veri setindeki değişim net olarak anlaşılmaz iken, ikinci tasarımda nisan ayından itibaren yaşanan sorun gözler önüne serilmiştir.

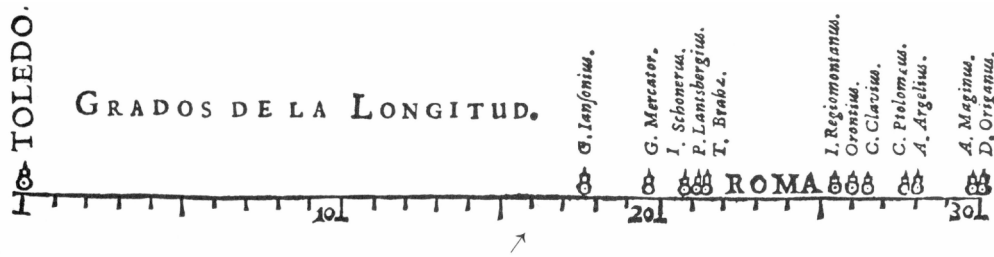


Şekil 3.2: Hatalı-doğru grafik

Şekil 3.2 den de anlaşılacağı üzere veri setini anlamadan hazırlanan her bir grafik amaca hizmet etmediği gibi çoğu zaman çelişkili sonuçlar ortaya koyarak kullanıcıları yanlış yönlendirebilir. Bu açıdan bakıldığında veri görselleştirme çalışmalarının veriyi anlama konusunda bir ön gereksinimi olduğu da söylenebilir. Çünkü verinin türü, boyutu, çeşitliliği grafik türünü tamamen değiştirebilir.

Veri görselleştirme bilimi, veri yönetimi ve iş analitiği alanındaki gelişmelere binaen yıllar içerisinde evrim göstermiştir. Son yıllarda statik grafikler yerine dinamik gösterge panelleri ve yönetim kokpiti uygulamaları ön plana çıkmaya başlamıştır.

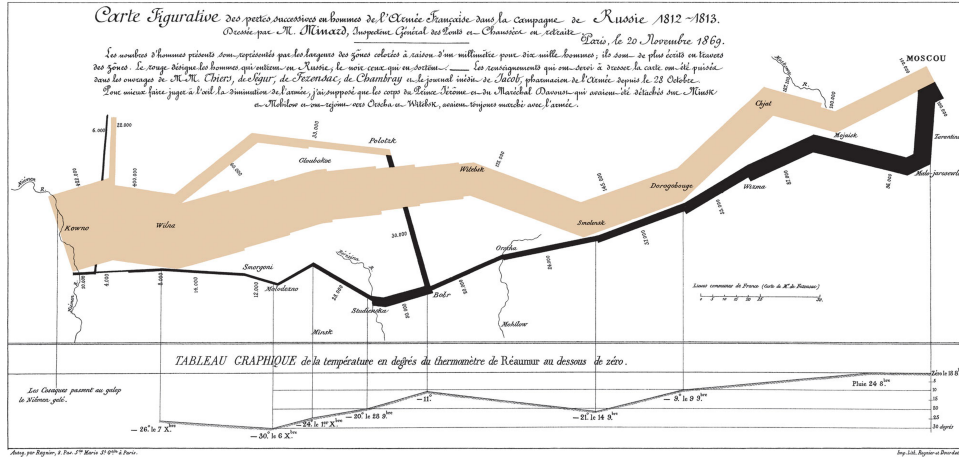
Her ne kadar son yıllarda görselleştirme alanına önemli bir eğilim olsa da, Veri görselleştirmeyi yeni bir kavram olarak algılamak doğru olmaz. Bu alandaki ilk çalışmalar insanlığın ilk yıllarına kadar dayanmaktadır (Riche et. al., 2018). Bununla birlikte İstatistiksel veriden elde edilen ilk grafik Astronot Michael Florent Van Langern tarafından çizilmiştir (Simon, 2014).



Şekil 3.3: İstatistik Verilerden Elde Edilen Bilinen İlk Grafik (Datavis.ca sitesinden alınmıştır.)

Veri görselleştirme tarihindeki önemli bir başka dönüm noktası da Jacques Bertin tarafından 1967 yazılan "Sémiologie Graphique" kitabıdır (Sharda, Delen & Turban, 2014). Bu kitapla birlikte bilginin görsellere dönüşüm süreci için yol haritaları sunulmuştur. Son yıllarda ise büyük veri kavramının öneminin artması ile birlikte gösterge panelleri ve görsel analitik çalışmaları artmış ve görselleştirme süreçleri dinamik ve çok boyutlu olarak evrimleşmiştir. Her ne kadar son yıllarda çok boyutlu grafikler ön planda olsa da, bilinen en önemli çok boyutlu grafik Şekil 3.4 te gösterilen

Charles Joseph Minard tarafından çizilen ve 1812 Rusya savaşı sırasındaki Napolyon ordusunun hareketlerini betimleyen grafikdir (Sharda, Delen & Turban, 2014).



Şekil 3.4: Charles Joseph Minard Tarafından Çizilen 1812 Rusya Savaşını Anlatan Grafik

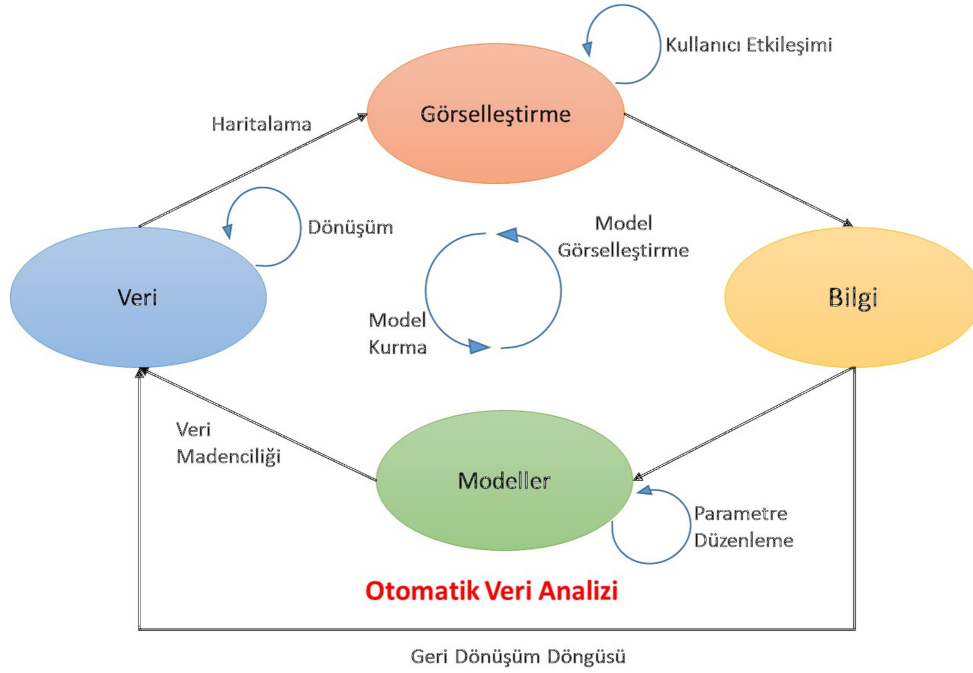
Grafikte açık kahverengi alanlar Rusya'ya giren asker sayısını (her milimetre başına 10.000 asker), siyah renkli kısımlar ise Rusya'dan ayrılan asker sayısını vermektedir. Renkli kısımlar hareket yönünü belirtirken, sıcaklık, enlem ve boylam bilgileri de grafik üzerinde sunulmuştur.

Son yıllarda çok boyutlu grafikler dinamik olarak iş zekâsı alanında gösterge panelleri ve yönetim kokpiti uygulamaları sıklıkla tercih edilirken, statik olarak ta infografikler ön plana çıkmıştır. Infografiklerin kullanımı ile birlikte grafiklerin belli bir hikâye kapsamında sunulması da tercih edilen yöntemlerdendir (Yuk and Diamond, 2014). Görselleştirme alanındaki çalışmaların son ulaştığı nokta ise görsel analitik olarak adlandırılan içerisinde analitik bileşenleri de barındıran grafiklerdir. Bu yaklaşımda hikâyeler ile birlikte neden sorgulayan ve mevcut durum yerine gelecekteki bir anı betimleyen kestirimci yaklaşımlar da veri görselleştirme süreci içerisine dâhil edilmiştir.

Daha önce veri görselleştirme çalışmalarının veriyi anlamakla başladığını belirtmiştik. Bazı durumlarda ise veriyi anlayabilmek için direkt olarak grafiklerden faydalanabiliriz. Bu açıdan bakıldığında Veri görselleştirme bilimi ile yapay zekâ ve veri madenciliği gibi komplike analitik alanlar arasında da yakın bir ilişki söz konusudur. Özellikle tecrübenin önemli bir unsur olduğu yapay zekâ uygulamalarında verinin temizlenmesi, bütünleştirilmesi, indirgenmesi ve dönüştürülmesi sürecinde veri görselleştirme en ez istatistik bilimi kadar önemli bir rol oynamaktadır.

3.3 Görsel Analitik

Görsel analitik 2000 yılların başında ortaya çıkan yeni kavramdır. Thomas Cook (2005) "Illmuniating the Path: Research and Development Agenda of Visual Analytics" kitabında ilk defa görsel analitik kavramını tanımlamıştır. Yapılan tanıma göre görsel analitik "İnteraktif görsel arayüzler ile gerçekleştirilen analitik çıkarsama bilimi" olarak ifade edilmiştir (Cook & Thomas, 2005). Yıllar içerisinde özellikle büyük veri ve iş zekâsı alanındaki gelişmeler dikkate alındığında bu tanımlı biraz daha genişletmekte fayda vardır. Bu bağlamda görsel analitik çok geniş ve karmaşık veri setlerini efektif bir şekilde anlayabilmek için otomatik analiz teknikleri ile görsel interaktif bileşenlerinin bir kombinasyonudur. Aşağıda şekil 6 da görsel analitik süreci betimlenmiştir (Keim, Kohlhammer & Ellis, 2010).



Şekil 3.5: Görsel Analitik Süreci

Görsel analitik grafikleri klasik grafiklerin aksine sadece “ne oluyor” sorusuna cevap vermez. Ayrıca “Ne olacak” sorusuna cevap verirken “Nasıl? ve Neden?” soruları için de ipuçları sunar (Mathew, 2018, Sharda, Delen & Turban, 2014). Aşağıdaki görselde görsel analitik ile veri görselleştirmenin karşılaştırılması gösterilmiştir (Sharda, Delen & Turban, 2014).

Şekil 3.6 dan da anlaşılacağı üzere görsel analitik, çıkarımsal istatistiksel yaklaşımları kullanabildiği gibi veri madenciliği ve yapay zekâ/makine öğrenmesi tekniklerinden de yararlanabilir (Riche et. al., 2018). Bazı başka kaynaklarda ise bu grafikler kullanıcılar tarafından manipüle edilebilen, erken uyarı sistemleri sayesinde çevik, işbirlikçi, basit, sezgisel ve analitik kalıpları vurgulayabilen yapıda olması gerekliliği ifade edilmiştir (Yuk and Diamond, 2014).

Yapay zekâ tekniklerinin görsel analitik sürecine analitik yetenekleri ile önemli katkı sağladığı muhakkaktır. Son yıllardaki bazı görsel analitik uygulamalarında öğrenme temelli yapay zekâ teknikleri bulut tabanlı kullanılan görsel analitik uygulamalarında kullanıcı geri bildirimlerinde öğrenerek görselleri zenginleştirme yolunda kullanılabilir (Keim, et. al., 2008).

Görsel analitik sürecinin kullanıcılara sağladığı faydaların belki de en önemlisi, karar destek sistemi ile iş analitiği arasında bir köprü vazifesi görmesidir. Diğer bazı faydalar ise aşağıdaki şekilde özetlenmiştir (Keim, et. al., 2008, Keim, Kohlhammer & Ellis, 2010).

- Bilgiyi sentezleyerek çok büyük boyutta, karmaşık, muğlak ve çoğu zaman birbiri ile çelişen yapıdaki veri kaynaklarının anlaşılabilir olmasını sağlamak
- Beklenen durumları tespit etmek ve beklenmeyen durumları keşfetmek
- Zamanında ve anlaşılabilir değerlendirmeler sunmak
- Eylem için efektif değerlendirme imkânı sunmak

Bu kadar önemli faydaları olmasına rağmen, bu süreç önünde bazı engellerinde olduğu muhakkaktır. Bu bağlamda görsel analitik sürecinde karşılaşılabilecek örnekler aşağıda sıralanmıştır (Keim, et. al., 2008).



Şekil 3.6: Görsel Analitik Sürecinin Veri Görselleştirmeden Farkı

- Veri boyutları ve hacmi dikkate alındığında olası ölçeklendirme problemleri
- Görsellerin ve veri kaynağının kalitesi
- Detaylandırma gereksinimlerinin tam olarak karşılanamaması
- Kullanıcı ara yüzlerinin işlevsel olmaması
- Görüntüleme cihazları ile ilgili problemler yaşanması
- Bazı problem alanlarında değerlendirme süreçlerinde yetersiz kalması
- Bilişim sistemi altyapı olanaklarının yetersizliği

Veri görselleştirme ve görsel analitik uygulamaları son yıllarda özellikle iş zekâsı alanında çok ön plana çıkmaktadır. Bu bağlamda iş zekâsı ile alanın ilişkisinin açıklanmasında fayda vardır.

3.4 İş Zekâsı ile Veri Görselleştirme İlişkisi

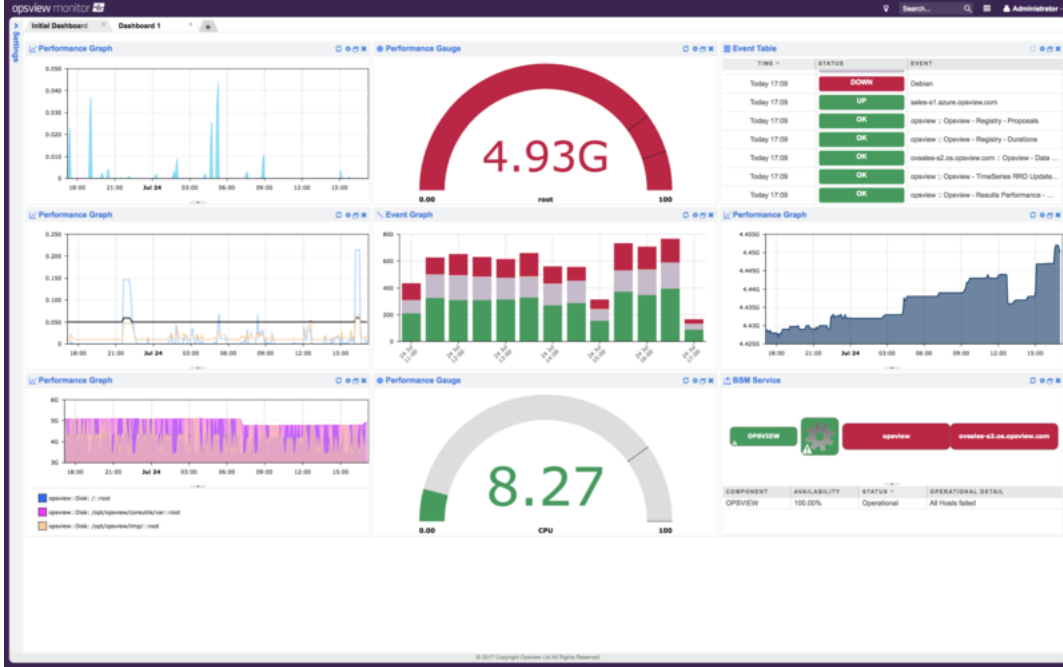
1989 yılında Gartner danışmanlık şirketinde çalışan analizci Howard Dresner İş zekâsını aşağıdaki şekilde tanımlamıştır.

“İş zekâsı temelde mimarileri, araçları, veritabanlarını, analitik yöntemleri, uygulamaları ve metodolojileri birleştiren bir "Şemsiye" terimidir”

Bu tanım yıllar içerisinde iş zekâsı alanında en kabul gören tanım olmuştur. Bu tanımdaki şemsiye terim ifadesi detaylandırıldığında herhangi bir iş zekâsı sisteminin 4 temel bileşeni olduğu söylenebilir: Veri Ambarları (Altyapı), İş Analitiği (Yöntem), İşletme Performans Yönetimi ve Ara yüz (Sharda, Delen & Turban, 2014). Bu kapsamda, tanımı daha derinlemesine incelediğimizde çok büyük boyuttaki veri setleri, modern analiz yöntemleri (Madencilik ve Yapay Zekâ yaklaşımları) ile incelenirken, analiz sonuçları kullanıcılara sunulurken performans yönetimi içerisinde yer alana anahtar performans göstergelerine bağlanması gerekmektedir. Kullanıcılara sunulması aşaması ise

çoğu zaman iş zekâsının en ön planda olduğu kısımdır ve bu aşamada kullanılan temel araç ise performans gösterge panelleridir.

Gösterge panelleri işletme performansını gerçek zamanlı (veya gerçek zaman yakın olarak) analiz edilmesini sağlayan, arka planda işletme performans yönetimi ile yoğun bir ilişkisi olan, kullanıcı etkileşimine olanak sağlayan dinamik bir veri görselleştirme tekniğidir (Stirrup, 2016). Bu görsel paneller belli bir amaç doğrultusunda hazırlanırken, bütün bilgiler kullanıcılara tek bir ekranda sunulur (Sharda, Delen & Turban, 2014). Şekil 3.7 de örnek bir gösterge paneli uygulaması görülmektedir.



Şekil 3.7: Örnek Dashboard (WikiMedia Commons sitesinden alınmıştır)

Yukarıdaki görselin bir gösterge paneli olarak sayılabilmesi için görsel analitik bileşenlerine de sahip olması beklenir. Yani panel üzerinden kullanıcılar gelecekteki bir an için tahminleri alabilirken, navigasyon yetenekleri ile olası aykırı durumlar için neden araştırması yapabilecek şekilde dinamik yapıda da olmalıdır (Goldmeier & Duggirala, 2015; Sharda, Delen & Turban, 2014). Bu açıdan bakıldığında gösterge paneli tasarımının sadece renkli görseller analiz sonuçlarını süslemek olmadığı da açıktır.

Gösterge paneli tasarımı sürecinde aşağıda yer alan 3 soruya cevap aranır (Gemignani, 2009).

- Gösterge panelini kullanacak kişiler kimlerdir?
 - Bu panellerle hangi değerler eklenecektir?
 - Hangi tür grafikler kullanılmalıdır? (Yukarıdaki iki soruya bağlı olarak) yetersizliği
- Bütün bu anlatılanlar ışığında iyi bir gösterge paneli tasarımının aşağıdaki yer alan özellikleri taşıması beklenir (Sharda, Delen & Turban, 2014).
- Görseller yardımıyla, özellikle istisnalar odaklanarak, veriler vurgulanmalı
 - Kullanıcı için minimal eğitim gereksinimi olmalı, yani sadece profesyoneller için değil herkes için olmalı
 - Farklı kaynaklardan tek bir ekrana özetlenmiş veri aktarımı

- Detaylardan özetlere, özetlerden detaylara geçiş sağlayan interaktif yapıda olmalı
- Gerçek zamanlı ve dinamik verileri sunmalı
- Uygulamak, yaymak veya sürdürmek için mümkün olan en az kodlamaya ihtiyaç duymalı

İş zekâsı kavramının her ne kadar çıkış noktası üst yönetici destek sistemleri olsa da, herhangi bir iş zekâsı uygulamasının her aşamaya destek sağlaması beklenir. İş zekâsı başka bir tanımda her tip kullanıcıya destek verebilen bir karar destek sistemi uygulamasıdır (Sharda, Delen & Turban, 2014). Bu anlamda gösterge panelleri her seviye kullanıcı için kolayca anlaşılabilir ve yönetilebilir şekilde tasarlanırlar. Son yıllarda ise özellikle üst yöneticilere karar verme süreçlerinde destek olacak şekilde daha gelişmiş, çoğu zaman birçok gösterge panelinin birden fazla cihaz üzerinde sunulduğu, SAP tarafında geliştirilen “Yönetim Kokpiti” kavramı ön plana çıkmıştır. Yönetim kokpitleri, yönetim kurulu toplantılarına başarı faktörleri, finansal göstergeler, pazar performansı, süreç ve çalışan performansı ve stratejik projelerin farklı duvarlarda, farklı gösterge panelleri ile sunulduğu “Savaş Odaları” yaklaşımı ile destek olurlar. Aşağıdaki resimde (Şekil 3.8) bir “Yönetim Kokpiti Savaş Odası” gösterilmiştir (Rainer, et. al., 2013).



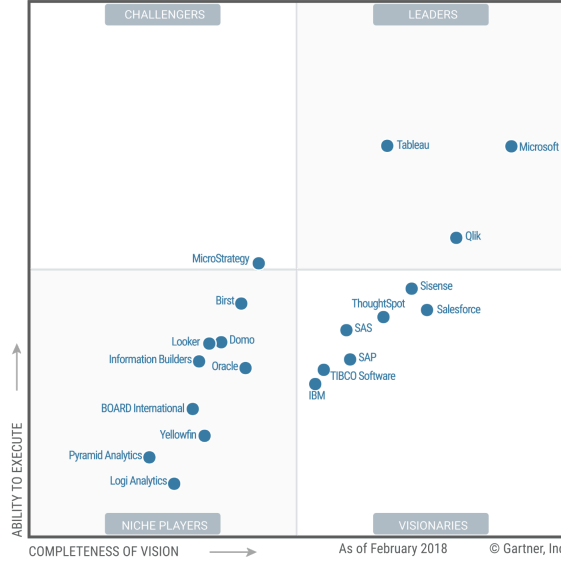
Şekil 3.8: Yönetim Kokpiti Savaş Odası

3.5 Görselleştirme Yazılımları

Son yıllarda veri görselleştirme alanına olan talep direkt olarak bu alandaki yazılımların sayısını da arttırmıştır. Tableau, Spotfire ve QlikView gibi tamamen görselleştirme alanına odaklanan yazılımların yanı sıra, büyük yazılım paketleri içerisinde yeni modüller de (SAP-Visual Intelligence, SAS-Visual Analytics, Microsoft-PowerBI, Oracle-Endeca) çıkmaya başlamıştır (Sharda, Delen & Turban, 2014). Diğer taraftan java tabanlı D3 ve açık kaynak kodlu R yazılımları da bu alanda özellikle esnek çalışmak isteyen yazılımcılar tarafında tercih edilmektedir (Kirk, 2012).

İş zekâsı alanında önemli çalışmaları olan Gartner danışmanlık şirketi tarafından her yıl yayınlanan piyasadaki öncü bilişim sistemlerini kullanım amaçlarına göre değerlendiren “Magic Quadrant”

grafğine göre 2018 yılı verilerine göre iş zekası alanındaki lider şirketler Tableau ve Microsoft PowerBI olarak görülmektedir. Şekil 3.9 de iş zekâsı alanındaki yazılımların değerlendirildiği grafiği görebilirsiniz.



Şekil 3.9: Gartner İş Zekâsı Alanındaki Magic Quadrant Grafiği (Şubat 2018 tarihli rapordan alınmıştır)

Bizde çalışmamızın devamında lider şirketlerden biri olan Tableau ile örnek uygulama hazırlamaya çalışacağız. Bu yazılımı diğer birçok görselleştirme yazılımından ayıran temel özellik ise kodlama gerektirmeyen oldukça kullanımı kolay bir ara yüze sahip olmasıdır. Ayrıca program kullanıcıları veri türleri ve içerikleri bağlamında yönlendirmeler ile destek sağlamaktadır.

3.6 Tableau ile Örnek Gösterge Paneli Tasarımı

Tableau yazılımı önceden de değindiğimiz gibi iş zekâsı alanında özellikle kolay kullanımı ve gelişmiş veri görselleştirme yetenekleri ile uzun yıllardır ön plandadır. Öncelikle tableau ekranları ile ilgili kısa bilgiler sunarak uygulamaya başlanacaktır.

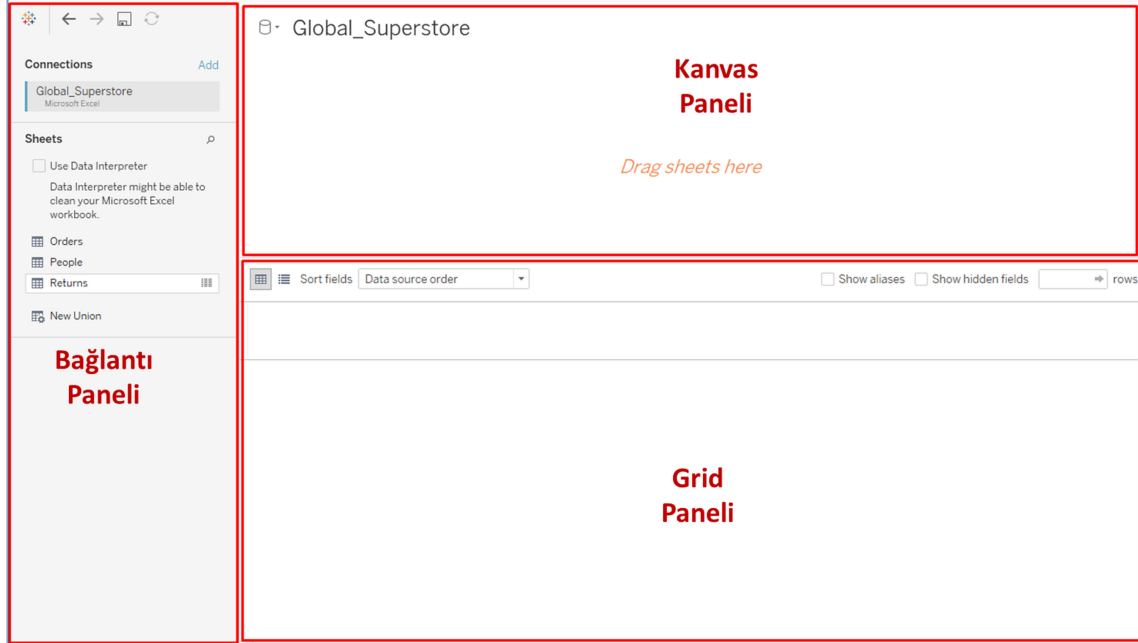
Yazılımın açılış ekranının veri setleri eklemesi gerçekleştirilebilirken, daha önce çalışılan çalışma kitapları da açılabilir. Tableau yazılımının diğer iş zekâsı yazılımlarına göre güçlü yanlarından biri de dosyadan veya sunucudan veri çekme kabiliyetlerinin çok üst düzey olmasıdır. Sadece Connect panelinde ilgili dosya türü ve sunucu türü seçilerek basitçe işlemler gerçekleştirilebilir. Son sürümünde yazılım 20 ye yakın dosya türünü ve 50 civarı sunucu bağlantısını tanımaktadır.

Herhangi bir veri kaynağı eklendiğinde yazılım veri kaynağı sayfasını açar (Şekil 3.10).

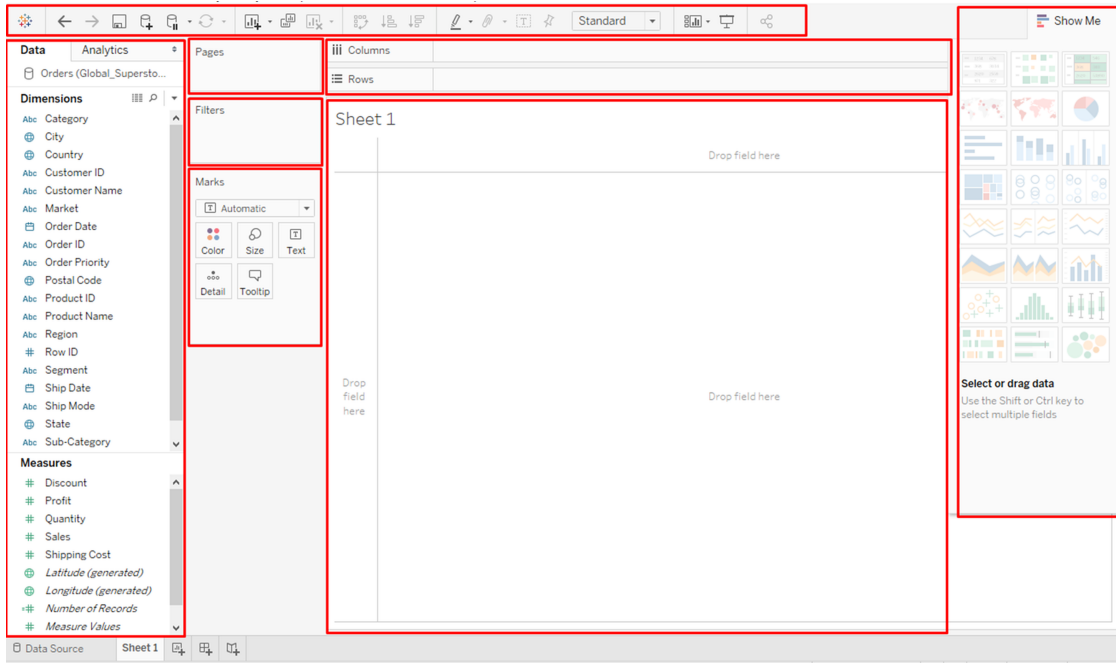
Bu alanda bağlantı panelinde dosya ve sunucu üzerindeki tablolar üzerinde işlemler gerçekleştirilirken, farklı tabloların "Join" kavramı ile bağlandığı panele ise "Kanvas Paneli" adı verilir. Grid panelinde ise veri ile ilgili ön işleme süreçleri gerçekleştirilir.

Tableau daki asıl çalışma alanı "Sheet" (çalışma sayfası) olarak adlandırılır ve Veri ve İşaretçiler (Marks) panellerini içerir (Şekil 3.11).

Yazılım ile ilgili bilinmesi gereken son konu ise bütün değişkenleri Dimensions (boyut) ve Measures (ölçek) olarak adlandırılmasıdır. Boyutlar kesikli ve kategorik değerleri içerirken, ölçekler



Şekil 3.10: Tableau Veri Kaynağı Sayfası



Şekil 3.11: Tableau Çalışma Sayfası Ekranı Görüntüsü

sürekli ve sayısal değerlere karşılık gelir. İşaretçiler alanında ise görseller ile ilgili filtreleme, renklendirme gibi biçimlendirme özellikleri tasarlanır. Üst kısımda yer alan şeritte ise sayfanın geneli ile ilgili biçimlendirme işlemleri gerçekleştirilir.

Yazılım bünyesinden üç farklı çalışma sayfası tasarlanabilir.

- Çalışma Sayfası (Sheet): Tek bir grafik içerir.
- Gösterge Paneli (Dashboard): Daha önceden hazırlanmış çalışma sayfaları eklenerek hazırlanır. Tasarım alanındaki görsel ve nesne sayısı tamamen kullanıcı ihtiyaçlarına göre belirlenir. Bu anlamda bir gereksinim analizine gereksinim duyulabilir.
- Hikaye (Story): Belirli bir konuda bir grafik serisi hazırlamak için kullanılır. Tümdengelim yöntemi ile vurgulanmak istenen duruma doğru belirli bir çizelge takip edilerek daha önce hazırlanan grafikler ve gösterge panelleri hikayeleştirilir.

Yukarıda sunulan bilgiler ışığında bir gösterge paneli uygulaması hazırlanması düşünülmektedir. Gösterge panelinde kullanılacak olan veri IBM sitesi tarafından (Stacker IV, 2018) paylaşılan Satış ve pazarlama alanındaki verileri içeren bir excel dosyasının Türkçe'ye adapte edilmiş halidir. Veri seti outdoor sporları alanında toptan satış yapan bir mağazaya ait satış verilerini içermektedir. Veri seti içerisindeki değerler sadece Avrupa ülkelerine satışı içerecek şekilde ön işleme süreçlerinde filtrelenerek indirgenmiştir. Bu bağlamda veri setiyle aşağıdaki alanları içeren bir gösterge paneli tasarlamaya çalışılacaktır.

- Ortalama birim kara göre renklendirme yapılan bir Avrupa haritası
- Satış hacmine göre alt ürün grupları isimlerini içeren bir kelime bulutu
- Ortalama Maliyet, Ortalama Kar ve Karlılık Oranı değerlerini vurgulayan ayrı ayrı bilgi kutuları
- Ortalama Karı en yüksek olan ilk 10 ürünün gösterildiği çubuk grafiği
- Sipariş metoduna göre boyutları ayarlanmış bir treemap grafiği

Ayrıca bütün bu grafikler yıllara göre, ülkelere göre (harita üzerinden) ve ana ürün kategorilerine göre aynı anda filtrelenebilir şekilde tasarlanmalıdır.

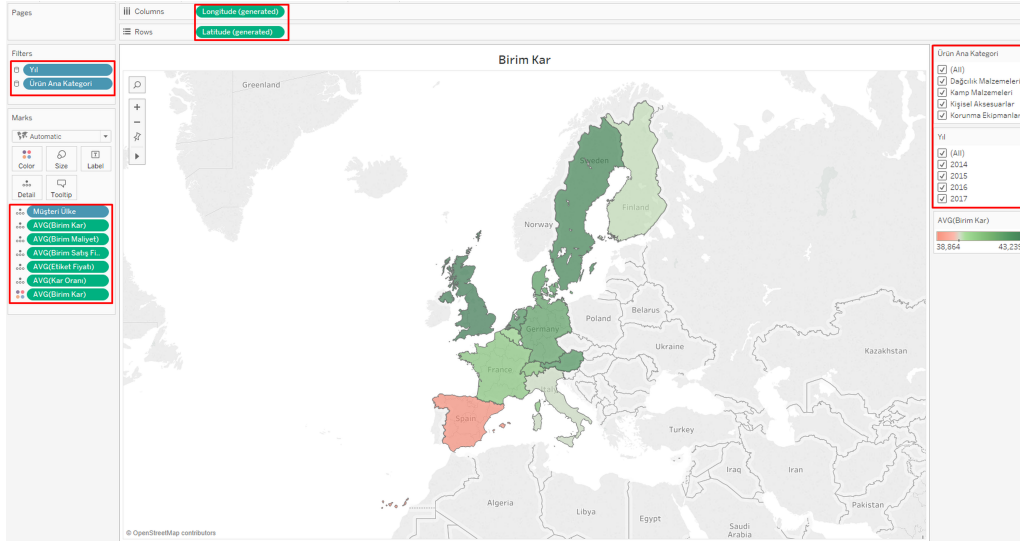
İlk olarak Avrupa haritasının olduğu grafiği tasarlanacaktır. Bunun için Müşteri ülke boyutunu ve Birim ölçөгünü seçerek, Tableau programının grafik öneri alanı olan "Show Me" düğmesinde "Maps" grafiği seçilir. Burada birim karı yazılımın otomatik olarak İşaretçiler kısmına attığına dikkat ediniz. Doğru bir ölçeklendirme için işaretçilerde yer alan birim kar değişkenin sağ tuş menüsünden toplamdan ortalama değerine değiştirilir. Eğer fare imleci ile gezerken ek bilgiler göstermesi talep ediliyorsa, bu durumda istenilen bütün değişkenler işaretçiler alanındaki "Detail" düğmesine taşınabilir. Birim kar grafiği Şekil 3.12 deki gibi görülecektir.

Sonradan gösterge paneli tasarımında kullanmak üzere yıl ve ana ürün kategorisi filtreleri mevcut grafik üzerine yerleştirilir. Bunun için her iki değişken de sürükleyip bırak yöntemi ile Filtreler alanına bırakılmalıdır.

Sıradaki grafik kelime bulutu grafiğidir. Burada yapılması gereken Alt ürün gruplarının işaretçiler kısmına "Text" düğmesine taşınmasıdır. Sonrasında Birim kar ölçөгü "Size" düğmesine bırakılır. Son olarak işaretçiler alanındaki grafik türünü text olarak değiştirirken ürün ana gruplarını da renklendirme için kullanılır. Kelime bulutu Şekil 3.13 deki gibi görülecektir.

Bilgi kutularının 3 farklı ekranda tamamen aynı biçimde hazırlanması gerekmektedir. Bunun için ölçekler işaretçiler alanına taşınıp sağ tuş menüsünden toplamdan ortalamaya döndürülmelidir. Aşağıdaki şekil 14 teki gibi görüntüler elde edilmelidir. Bunun için işaretçiler alanındaki "Size" ayarı yükseltilmelidir.

Şimdi en yüksek karı getiren ürün gruplarını betimleyen çubuk grafiği tasarımına geçilebilir. Bunun için ilk aşamada birim karı (ortalamaya çevirerek) sütun alanına ve Ürün boyutu satır alanına



Şekil 3.12: Birim Kar Harita Grafiği

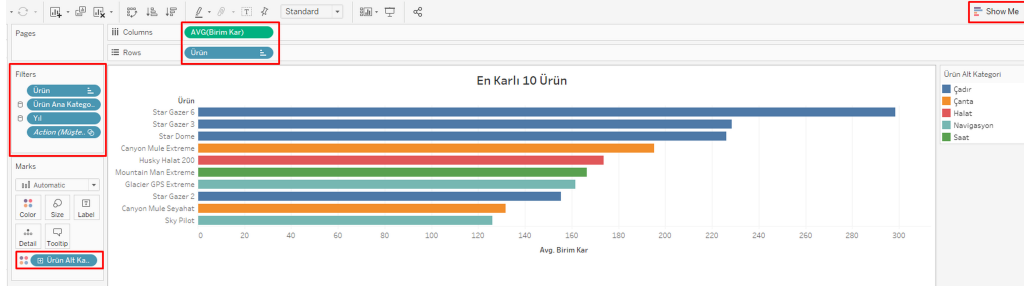


Şekil 3.13: Kelime Bulutu

Ortalama Maliyet	Ortalama Kar	Karlılık Oranı
65,86	42,14	0,4372

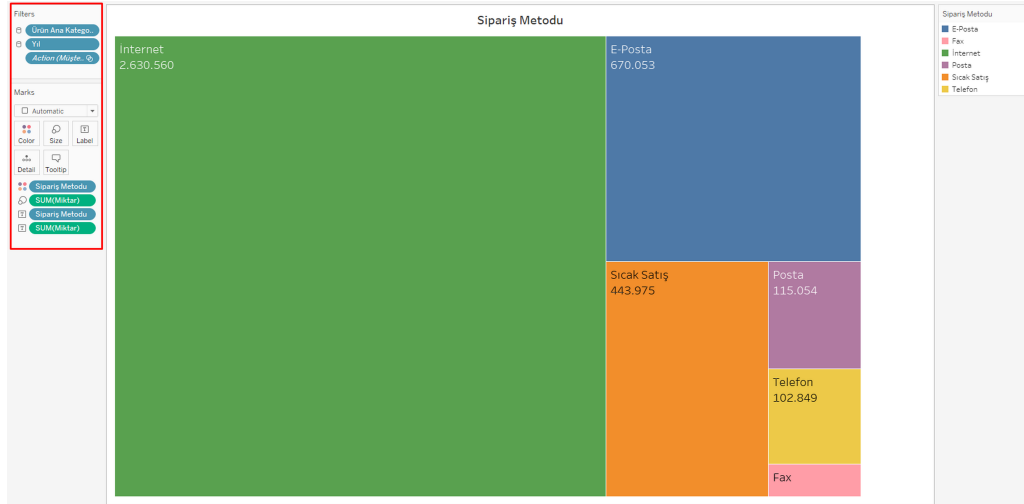
Şekil 3.14: Bilgi Kutucukları

taşınır. Sonrasında ürün boyutu filtreler taşınır ve çıkan ekranda “Top” tabına geçilerek, gelen ekranda alt kısımda yer alan menüden birim kar ve ilk 10 değerleri seçilir. Şekil 3.15 e benzemesi için gerekli son ayar, ürün ana gruplarının “Color” işaretçisine taşınması ve şeritte yer alan sıralama düğmesi ile sıralama yapılmasıdır.



Şekil 3.15: En Karlı Ürünler Çubuk Grafiği

Gösterge paneli hazırlamaya geçilmeden önce çizilmesi gereken son grafik ise sipariş metodlarının yer aldığı “Treemap” grafiğidir. Burada yapılması gereken sipariş metodu ve Birim kar değişkenlerini birlikte seçerek “Show Me” düğmesinden treemap grafiğine tıklamaktır. Renklendirmeyi değiştirmek için sipariş metodu “Color” düğmesine taşınır. Şekil 3.16 teki renkler için “Color” düğmesine bir kez daha tıklanarak “Edit” yapılır. Sipariş miktarını etiket olarak tasarlamak için “Label” sekmesine tıklayarak grafik tamamlanır.

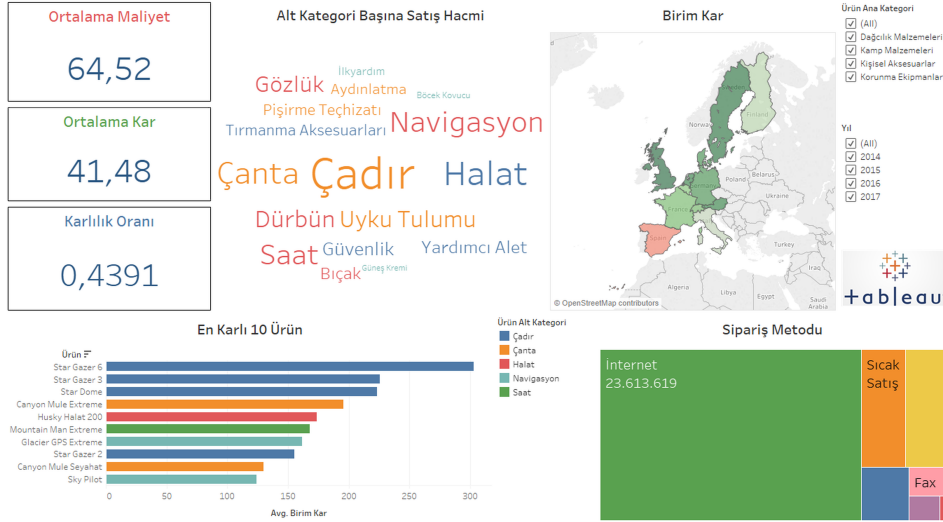


Şekil 3.16: Sipariş Metodu Treemap Grafiği

Bütün öncül grafikler hazırlandığı için artık gösterge paneli tasarımına geçilebilir. Gösterge paneli hazırlamak için açılan sayfada solda yer alan sayfalar isteğe bağlı olarak sağ tarafa taşınır ve yerleşimler yapılır. Yerleşim yapılırken “Floating” seçimi yapılarak nesnelerin birbiri üzerine taşınması tercih edilebilir. Eğer fazladan göstergeler grafiklerden gelecek ekranı kalabalıklaştırıyor ise fazla göstergeler kaldırılabilir.

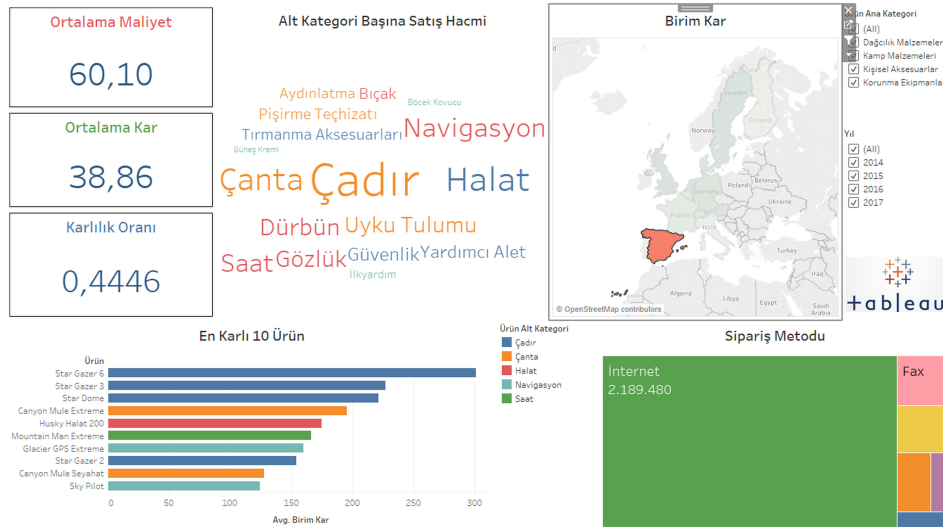
Tasarım tamamen kullanıcının hayal gücüne bağlıdır. Ama dikkat edilmesi gereken odak kaymasına neden olmayacak ve kolayca okunabilir tasarımlar hazırlamaktır. Aşağıdaki Şekil 3.17 da

yer alan tasarım sürükle bırak yöntemi ile oluşturulmuştur. Burada sadece fazladan Tableau logosu Object sekmesinde eklenmiştir.



Şekil 3.17: Gösterge Paneli

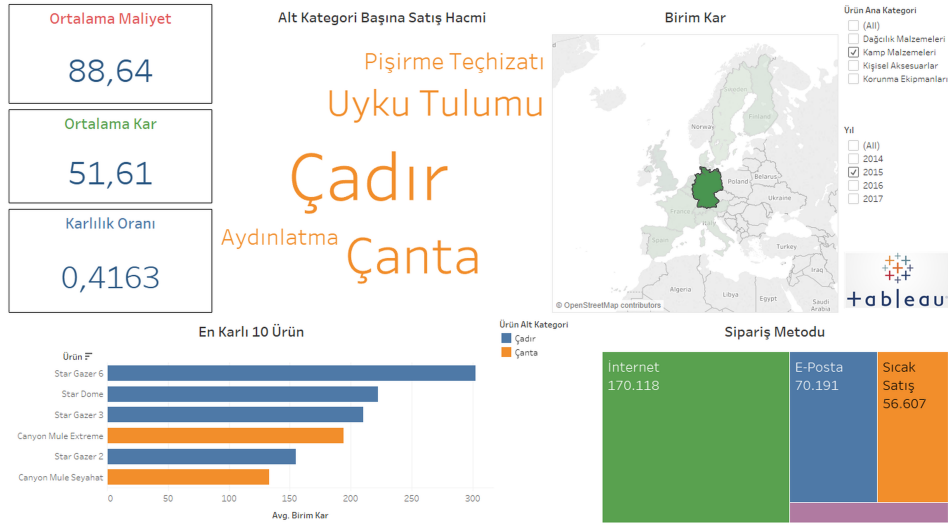
Bu tasarım üzerindeki filtrelerin düzgün çalışması için sağ panelde yer alan filtreleme alanlarına gidilerek sağ tuş menüsünde “Apply to Worksheets” düğmesinde bütün çalışma sayfaları seçilmelidir. Bu işlem her bir filtre alanı için ayrı ayrı tasarlanmalıdır. Ayrıca harita üzerindeki bir ülke seçildiğinde bütün gösterge panelinin filtrelenmesi isteniyorsa, harita grafiği üzerinde sağ tarafta çıkan filtre düğmesine basılmalıdır. Bu şekilde yapılan dinamik filtreleme sürecine örnek aşağıdaki Şekil 3.18 de görülmektedir. Düşük karlı görünen İspanya üzerine tıkladığında bütün gösterge panelindeki değerlerin değiştiğine dikkat ediniz.



Şekil 3.18: Filtrelenmiş Gösterge Paneli

Son olarak Almanya’da 2015 yılında yapılan Kamp Malzemelerini bulmaya çalışalım. Sadece 3

fare tıklaması ile bu bilgiye aşağıdaki Şekil 3.18 de olduğu gibi ulaşabiliriz.



Şekil 3.19: Almanya 2015 yılı Kamp Malzemeleri Ürün Grubu Filtresi Yapılmış Gösterge Paneli

Bütün değerlerin seçimle değiştiğine dikkat ediniz. Örneğin Sol üst köşede yer alan 88,64 değeri önceki bölümlerde gösterilen iki değerden (64,52 ve 60,10) farklıdır ve sadece Almanya'nın 2015 yılı kamp malzemeleri satışlarından hesaplanmıştır.

3.7 Sonuç

Son yıllarda bilişim sistemleri alanındaki gelişmelere paralel olarak veri analizi ve işleme alanında da büyük ilerlemeler olmuştur. Firmalar büyük boyuttaki verilerini daha anlaşılabilir formatta sunabilmek adına veri görselleştirme alanına yönelmeye başlamışlardır. Bu alandaki çalışmalar özellikle 2000 yılından sonra analitik bileşenleri de içeren dinamik grafikler alanında yoğunlaşmıştır. Bu alanda uzmanlaşmış yazılımlarının sayısının artması ile birlikte gösterge panelleri ve görsel analitik uygulamaları yavaş yavaş bir endüstri standardına dönüşmeye başlamıştır. Çalışmamızda bu tarihsel süreç özetlenirken, bu alanda çalışacak uygulamacılar için teorik bir altyapı oluşturulması amaçlanmıştır. Ayrıca bir gösterge paneli uygulaması da alandaki lider yazılımlardan bir tanesi yardımıyla gerçekleştirilerek, teorinin somutlaştırılmasına çalışılmıştır. Çalışma sonucu olarak veri görselleştirme kavramının öneminin artarak devam ettiği tartışılmış ve bu durumun özellikle yapay zekâ yaklaşımlarının etkisi ile yıllar içerisinde farklılaşarak ve evrilerek devam edeceği öngörülmüştür.

3.8 Kaynakça

Cook, K. A., & Thomas, J. J. (2005). Illuminating the path: The research and development agenda for visual analytics (No. PNNL-SA-45230). Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

Gemignani, Z. (2009). A Guide to Creating Dashboards People Love to Use. URL http://www.cpsc.org/assets/Data/guide_to_dashboard_design1.pdf. Erişim Tarihi: 12.10.2018

Goldmeier, J., & Duggirala, P. (2015). Dashboards for Excel. Apress. Gösterge Paneli Örneği, URL: <https://goo.gl/eQiMb8> Erişim Tarihi 13.10.2018

Howson, C., Sallam, R., Richardson, J., Tapadinhas, J., Idoine, C., & Woodward, A. "Magic Quadrant for Analytics and Business Intelligence Platforms", Şubat 2018, Url: <https://www.gartner.com/doc/reprints?id=1-4RXB56A&ct=180227&st=sb> Erişim Tarihi: 13.10.2018

Keim, D., Andrienko, G., Fekete, J. D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In Information visualization (pp. 154-175). Springer, Berlin, Heidelberg.

Keim, E. D., Kohlhammer, J., & Ellis, G. (2010). Mastering the Information Age: Solving Problems with Visual Analytics, Eurographics Association.

Kirk, A. (2012). Data Visualization: a successful design process. Packt Publishing Ltd.

Knaflic, C. N. (2015). Storytelling with data: A data visualization guide for business professionals. John Wiley & Sons.

Matthew N., "Why Visual Analytics", White Paper, Url: <https://www.tableau.com/learn/whitepapers/why-visual-analytics> Erişim Tarihi: 12.10.2018

Miller, J. D. (2017). Big Data Visualization. Packt Publishing Ltd.

Rainer, R. K., Cegielski, C. G., Splettstoesser-Hogeterp, I., & Sanchez-Rodriguez, C. (2013). Introduction to information systems. John Wiley & Sons.

Riche, N. H., Hurter, C., Diakopoulos, N., & Carpendale, S. (Eds.). (2018). Data-driven Storytelling. CRC Press.

Sharda, R., Delen, D., Turban, E., Aronson, J., & Liang, T. P. (2014). Business Intelligence and Analytics: Systems for Decision Support-(Required). London: Prentice Hall.

Simon, P. (2014). The visual organization: data visualization, Big Data, and the quest for better decisions. John Wiley & Sons.

Stacker VI,M. Retail Sales & Marketing – Profit & Cost, Örnek Veri Seti, URL: <https://www.ibm.com/communities/analytics/watson-analytics-blog/retail-sales-marketing-profit-cost/> Erişim Tarihi 13.10.2018

Stirrup, J. (2016). Tableau Dashboard Cookbook. Packt Publishing Ltd.

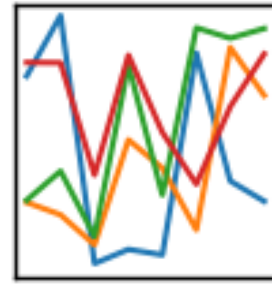
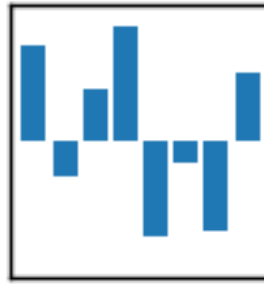
Van Langren, M. F. Url: <http://www.datavis.ca/gallery/langren/> Erişim Tarihi: 14.10.2018

Wexler, S., Shaffer, J., & Cotgreave, A. (2017). The Big Book of Dashboards: Visualizing Your Data Using Real-world Business Scenarios. John Wiley & Sons.

Yuk, M., & Diamond, S. (2014). Data visualization for dummies. John Wiley & Sons.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



4. PANDAS ile Veri Analizi

PANDAS ile Veri Analizi

Prof. Dr. İsmail Hakkı CEDİMOĞL, Doç. Dr. İ. Hakan SELVİ, Ezgi ÇAKMAK

4.1 Giriş

Günümüzde veri miktarının artması ile veri analizi oldukça önem kazanmıştır. Veri analizi, kaynak verilerin incelenip anlamlı bilgiler çıkartılması ve bu bilgilerin faydalı tahminler yapmak için kullanılmasını içeren bir süreçtir. Analiz yapmak için önce verilerin toplanması gerekmektedir. Toplanan veriler tablo veya veri çerçeveleri gibi ortamlarında saklanır. Verilerden eleme, sorgulama, seçme gibi yöntemler ile anlamlı bilgiler elde edilir. Bu bilgiler mevcut problemlerin çözümü, geleceğe yönelik tahmin yapma gibi birçok farklı sorunun çözümünde kullanılabilir. Veri analizi; birçok farklı alanda farklı endüstrilerde farklı sorulara çözüm aramak veya hipotezlerin doğruluğunu test etmek/reddetmek için kullanılır.

Python programlama dili kolay kullanımı ile veri bilimciler için çok tercih edilen bir seçenek olmuştur. Python özellik olarak içerdiği sayısal, matematiksel ve veri görselleştirme kütüphaneleri ile (NumPy, Matplotlib, Skikit...) veri analizini kolaylaştırmaktadır. Bu bölümde, veri analizi için büyük kolaylıklar sağlayan açık kaynak kodlu Pandas kütüphanesinin temel kullanımı anlatılmıştır. Örnekler, Jupyter Notebook ile hazırlanmıştır.

Python kütüphanelerinden birisi olan pandas, veri analizini hızlı ve kolay hale getirmek için tasarlanmıştır. pandas veri yapıları olarak, Seri ve DataFrame'i kapsamaktadır. Aynı zamanda pandas ile döngüsüz veri işlemi yapılabilmektedir.

Pandas ile erişilebilen dosyalar şu şekilde sıralanabilir:

- CSV dosyalarını okuma ve yazma
- JSON dosyalarını okuma ve yazma
- Excel dosyalarını okuma ve yazma
- XML dosyalarını okuma ve yazma
- HTML sayfalarından veri ve imaj alma
- Bir ilişkisel veritabanından veri alma ve depolama

- MongoDB'den veri alma ve depolama

Pandas temel veri yapılarının ikisi olan seri ve dataframe'dir.

Seri'lerin Özellikleri:

- Pandas veri yapılarından birisi olan Series, aynı veri türünü saklayabilen tek boyutlu dizilerdir.
- Series (seriler) bir pandas nesnesidir.
- Python'da kullanılan ve built-in olarak hazır olan list veri yapısına benzemektedir.
- Python list yapısından farkı ise, her bir eleman ile bağlantılı ve indeks olarak adlandırılan bir etiketi mevcuttur. pandas kütüphanesini kullanabilmek için, pandas kütüphanesi import etmek yeterli olmaktadır. "import pandas as pd" ifadesi ile pandas kütüphanesi hafızaya yüklenmiş olur. Pandas seri veri yapısı örnekleri aşağıda verilmiştir:

Örnek 1:

```
import pandas as pd
#seri = pd.Series([data], index=[index])
sayilar = pd.Series([0, 1, 4, 9, 16, 25, 36, 49],
name='Sayi_Kareleri')
print sayilar
```

Örnek 2:

```
#indeks olusturma
import pandas as pd
ERP_Market = pd.Series([28, 15, 12, 7, 5, 2, 2, 29],
index=['SAP', 'Oracle', 'Microsoft', 'Infor', 'Epicor', 'IFS',
'Lawson', 'Diger_ERP_Yazilimleri'])
print ERP_Market
```

Örnek 3:

```
#indeks olusturma ve dilimleme
import pandas as pd
ERP_Market = pd.Series([28, 15, 12, 7, 5, 2, 2, 29], index=['SAP',
'Oracle', 'Microsoft', 'Infor', 'Epicor', 'IFS', 'Lawson',
'Diger_ERP_Yazilimleri'])
print ERP_Market
print
print ERP_Market[3]
print
print ERP_Market[0:5] #dilimleme islemi
print
print ERP_Market['Epicor']
print
print ERP_Market['Oracle':'IFS'] #dilimleme islemi
```

Dictionary ile Seri Deklerasyonu

Örnek 4:

```
# Dictionary ile Series deklasyonu
import pandas as pd
ERP_Market = pd.Series({
    'SAP': 28,
    'Oracle': 15,
    'Microsoft': 12,
    'Infor': 7,
    'Epicor': 5,
    'IFS': 2,
    'Lawson': 2,
    'Diger_ERP_Yazilimleri': 29
})
print
print ERP_Market
```

Örnek 5:

```
# Dictionary ile Series deklasyonu
import pandas as pd

ERP_Market = pd.Series({
    'SAP': 28,
    'Oracle': 15,
    'Microsoft': 12,
    'Infor': 7,
    'Epicor': 5,
    'IFS': 2,
    'Lawson': 2,
    'Diger_ERP_Yazilimleri': 29
})
print
print ERP_Market
```

Örnek 6:

```
ERP_Fiyat = pd.Series({
    'SAP': 1000000,
    'Oracle': 800000,
    'Microsoft': 500000,
    'Infor': 300000,
    'Epicor': 400000,
    'IFS': 1000000,
```

```
'Lawson': 300000,
'Diger_ERP_Yazilimlari': 300000
})
print
print ERP_Fiyat
```

Pandas kütüphanesinde kullanılan veri yapılarından bir diğeri ise DataFrame'dir. DataFrame satır ve sütunlardan oluşan bir tablodur. DataFrame'deki her bir sütun Seri nesnesidir. Satırlarda ise Seri'nin elemanları(veriler) yer almaktadır. DataFrame, built-in Python Dictionary kullanılarak da oluşturulabilir. DataFrame nesnesinin 2 indeksi vardır, bunlardan bir tanesi sütun indeksi, diğeri ise satır indeksidir. Satır indeksi belirtilmemiş ise, pandas RangeIndex oluşturur. Range 0 ile N-1 arasındadır. N ise DataFrame'deki satır sayısıdır. Görsel olarak, bir pandas DataFrame'in görünümü, indeks, satır ve sütunlardan oluşmaktadır. DataFrame'in indeks, sütunlar ve veriler (değerler) olmak üzere 3 bileşeni vardır.

DataFrame oluşturulması:

Örnek 7:

```
ERP_Bilgi = pd.DataFrame({
'Pazar_Payi_(Yuzde)': ERP_Market ,
'Ortalama_Fiyat_(TL)': ERP_Fiyat
})
print
print (ERP_Bilgi)
```

Pandas önce, read_csv fonksiyonunu kullanarak diskten belleğe(memory) ve DataFrame'e veri okur. İndeks etiketi ve sütun adı, sırasıyla indeksin ve sütunların tek tek üyelerini ifade eder. İndeks terimi, tüm indeks etiketlerini, sütun terimi ise tüm sütun adlarını göstermektedir.

Örnek 8:

```
import pandas as pd
# csv dosyasından verilerin yuklenmesi
veri = pd.read_csv('filmVerileri.csv')
```

4.2 Python Pandas Kütüphanesi Uygulamaları

Seriler, tek boyutlu veri türü olarak tanımlanır. DataFrameler ise birden fazla boyutu olan daha tablo yapılarıdır. İki veri türü ile de kullanılan temel işlemler ilerleyen başlıklarda gösterilecektir.

4.2.1 Seriler

Seriler, indeks değeri bulunan tek boyutlu dizilerdir. Tablo 4.1'de Serilerin yapısı gösterilmiştir. Tabloda görüldüğü üzere, seride yer alan değerler indeks değerleri ile eşleştirilerek tutulurlar.: <http://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.html> (Erişim Tarihi:14/10/2018)

Tablo 4.1: Seriler

Seriler	
İndeks	Değer
0	5
1	-3
2	8
3	2

```
import pandas as pd
import numpy as np
```

```
seri = pd.Series([5,-3,8,2])
```

```
seri
```

```
0    5
1   -3
2    8
3    2
dtype: int64
```

Şekil 4.1: Seri Oluşturma

Seri Oluşturma ve Temel İşlemler

Seri oluşturmak için öncelikle Pandas kütüphanesinin çalışmamıza dahil edilmesi gerekmektedir. Seri oluştururken Series argümanına değerler dizi olarak tanımlanır. Şekil 4.1'de gösterilmiştir.

Seri tanımlarken, indeks değerleri belirlenmez ise 0'dan başlayarak otomatik olarak oluşturulur. Bazı durumlarda farklı indeks değerleri kullanmak gerekebilir. Bu durumda, Seri oluşturulurken her değer için bir indeks karşılığı olacak şekilde, indeks değerleri dizi olarak tanımlanmalıdır. Şekil 4.2'de indeks değerleri ile birlikte bir Seri oluşturma gösterilmiştir.

```
seri = pd.Series([1,2,3,4] , index=['d','c','b','a'])
```

```
seri
```

```
d    1
c    2
b    3
a    4
dtype: int64
```

Şekil 4.2: İndeks Değeri Tanımlama

Seri oluşturduktan sonra değerlere ulaşmak için nesnenin .values ve indekslere ulaşmak için ise

.index özelliği kullanılır.

Veri Seçme

Seriler ile işlem yapılırken, istenilen değer veya değerleri seçmek için indeks veya anahtar kullanılır. Şekil 4.3'te beş elemanlı, elemanları 0 ile 1 değerleri arasında rastgele oluşturulan bir Seri içinde istenilen değeri seçmek için indeks yöntemi kullanılmıştır. İndeksi 'B' olan eleman seçilmiştir.

```
s =pd.Series(np.random.rand(5),index=list("ABCDE"))
```

```
s
```

A	0.822718
B	0.199441
C	0.491712
D	0.096613
E	0.939354

```
dtype: float64
```

```
s['B']
```

```
0.19944105044882598
```

Şekil 4.3: İndeks ile Veri Seçme

Diğer bir yöntem ise değer dizindeki sırasına yani anahtarına göre seçmektir. Şekil 4.4'te anahtar kullanılarak seçme işlemi gerçekleştirilmiştir.

```
s
```

A	0.822718
B	0.199441
C	0.491712
D	0.096613
E	0.939354

```
dtype: float64
```

```
s[3]
```

```
0.09661278939674423
```

Şekil 4.4: Anahtar ile Veri Seçme

Örnekte (Şekil 4.5) yer alan ilk komut ile sırasıyla 0'dan 3'e kadar seçim yapılmış, ikinci komut ile ise indeksi A ve D olan değerler seçilmiştir. Seçme işlemi kalıcı olmamakla birlikte yalnızca istenilen değerlerin listelenmesini sağlamaktadır. Ancak yapılmak istenilen işleme göre seçilen değerlerden yeni bir Seri oluşturulabilir, değerler değiştirilebilir veya farklı matematiksel işlemler yapılabilir.


```
s[0:3]
A    0.822718
B    0.199441
C    0.491712
dtype: float64

s[['A', 'D']]
A    0.822718
D    0.096613
dtype: float64
```

Şekil 4.5: Birden Fazla Veri Seçmek

Şekil 4.6'da yer alan ilk örnekte indeksi 'A' ve 'C' olan değerlerden s2 adında yeni bir seri oluşturulmuş, ikinci örnekte ise s serisinin ikinci değeri 1 olarak değiştirilmiştir. Serideki veri tipi float olduğu için değer 1.0 olarak atanmıştır.

```
s2= s[['A', 'C']]

s2
A    0.822718
C    0.491712
dtype: float64

s[2] = 1

s
A    0.822718
B    0.199441
C    1.000000
D    0.096613
E    0.939354
dtype: float64
```

Şekil 4.6

Filtreleme ve Matematiksel Operatörler

NumPy kütüphanesi, matematiksel operatörler (<, >, <=, >=, =) ile filtreleme yapılmasına olanak sağlar. Örneğin, Seri içerisinde hangi elemanların 0.5'ten küçük olduğu öğrenilmek istenirse Şekil 4.8'deki gibi sorgulanır.

NumPy kütüphanesi kullanarak Seriler ile matematiksel işlemler (+, -, *, /) yapmak da mümkündür. Şekil 4.8'de yer alan örnekte seri ilk olarak 2 ile çarpılmış sonrasında ise karesi alınmıştır.

```
s[s<0.5]
B    0.199441
D    0.096613
dtype: float64
```

```
s[s>=1]
C    1.0
dtype: float64
```

Şekil 4.7

```
s = s*2
s
A    1.645436
B    0.398882
C    2.000000
D    0.193226
E    1.878708
dtype: float64
```

```
s = s**2
s
A    2.707460
B    0.159107
C    4.000000
D    0.037336
E    3.529544
dtype: float64
```

Şekil 4.8

```
def f(i):
    if i <1:
        return "küçük"
    elif i>=2 and i<=3:
        return "orta"
    else:
        return "büyük"
```

```
s.apply(f)
```

```
A    orta
B    küçük
C    büyük
D    küçük
E    büyük
dtype: object
```

Şekil 4.9

Matematiksel işlemlerin dışında, fonksiyonlar ile sınıflandırma veya filtreleme yapılabilir. Örneğin; hazırlanan fonksiyon ile 1'den küçük değerler 'küçük', 2 ve 3 arasında bulunanlar 'orta' ve 3'ten büyük değerler ise 'büyük' olarak sınıflandırılmaktadır. Fonksiyonu herhangi bir seriyi sınıflandırmak üzere kullanmak için Seri nesnesinin 'apply' özelliği kullanılır. (Şekil 4.9)

Sıralama

Sayısal değerler içeren bir Serinin değerlerinin büyükten küçüğe veya küçükten büyüğe sıralanması için `sort_values` fonksiyonu kullanılır. `sort_values` fonksiyonu kullanılırken eğer belirtilmez ise küçükten büyüğe sıralama yapar ve kalıcı değildir. Ascending parametresini False olarak değiştirerek sıralama büyükten küçüğe yapılır. Yapılan sıralamanın seriyi kalıcı olarak değiştirmesi isteniyorsa `inplace` parametresi True olarak ayarlanmalıdır. (Şekil 4.10)

Not A Number

NaN değeri, boş bir alanın ya da sayısal olarak tanımlanamayan bir değeri temsil eder ve bu değerler ile işlem yapılamaz. Bu sebeple, veri analizi yaparken NaN (Not a Number) değerleri problemlere yol açmaktadır. Dolayısıyla bu değerlerin belirlenmesi oldukça önemlidir. Pandas ile bir seri içinde hangi değerlerin NaN olduğunu görmek için `isnull()` veya NaN olmayan değerler için `notnull()` fonksiyonları kullanılır.

4.2.2 DataFrame

DataFrame, iki boyutlu- tablo şeklindeki veri türüdür. Serilerden farklı olarak, sütunlar yer alır ve her sütun farkı veri tipi taşıyabilir. Tablo 4.2'de DataFrame yapısı gösterilmiştir. Seriler de olduğu gibi DataFrame'lerde de indeks bulunur. 'Ad', 'Ders' ve 'Final' oluşturulan DataFrame'in sütunlarıdır.

DataFrame Oluşturma ve Temel İşlemler

DataFrame oluşturmak için birkaç yöntem kullanmak mümkündür. Bunlardan ilki dictionary (sözlük) türünden üretilen DataFrame'dir. Şekil 4.11'de sözlükten DataFrame oluşturma işlemi gösterilmiştir.

Diğer bir yöntem ise önceki bölümde üzerinde çalıştığımız serilerden DataFrame'in sütunlarını oluşturmaktır. Örnek 4.12 ilk olarak, değerleri rastgele 0 ile 1 arasındaki sayılardan oluşan beş

```
s.sort_values()
```

```
D    0.037336  
B    0.159107  
A    2.707460  
E    3.529544  
C    4.000000  
dtype: float64
```

```
s
```

```
A    2.707460  
B    0.159107  
C    4.000000  
D    0.037336  
E    3.529544  
dtype: float64
```

```
s.sort_values(ascending=False,inplace=True)
```

```
s
```

```
C    4.000000  
E    3.529544  
A    2.707460  
B    0.159107  
D    0.037336  
dtype: float64
```

Şekil 4.10

Tablo 4.2: DataFrame Yapısı

DataFrame			
İndeks	Ad	Vize	Final
0	Ali	84	70
1	Ayşe	65	83
2	Ahmet	75	90
3	Mehmet	90	85

```
dictionary = {'Ad' : ['Ali', 'Ayşe', 'Ahmet', 'Mehmet'],
              'Vize' : [84, 65, 75, 90],
              'Final' : [70, 83, 90, 85]}
```

```
df = pd.DataFrame(dictionary)
```

```
df
```

	Ad	Final	Vize
0	Ali	70	84
1	Ayşe	83	65
2	Ahmet	90	75
3	Mehmet	85	90

Şekil 4.11: Sözlükten DataFrame Oluşturmak

elemanlı bir seri oluşturulmuştur. Bu seri df2 adındaki dataframe'in ilk sütunu olarak tanımlanmış ve sütun '1' olarak adlandırılmıştır. df2'nin ikinci sütunu ise birinci sütununun iki ile çarpılması sonucu elde edilmiştir.

Eleman Seçme

DataFrame sütunlarına ulaşmak için columns, indekslerine ulaşmak için index ve elemanlarına ulaşmak için ise values özelliği kullanılır. Oluşturduğunuz frame adından sonra nokta ve ardından tab tuşuna basarak kullanmak isteyebileceğiniz özelliklere ulaşabilirsiniz. Bu işlem yapıldıktan sonra kullanılacak özellikler listelenir. Şekil 4.13'te gösterilmiştir.

df'nin elemanlarına ulaşmak için Şekil 4.4'te values özelliğini kullanılmıştır.

Aynı yöntem ile kolon isimlerine ulaştıktan sonra sadece seçmek istediğimiz kolona ait elemanları listeleyebiliriz. Şekil 4.15'te yer alan örnekte bunun için iki farklı yöntem kullanılmıştır:

Şekil 4.16 indeks değerlerine göre seçme işlemi yapmak için aşağıdaki komut takip edilmelidir.

Aynı anda birden fazla satır seçmek için Şekil 4.17'de yer alan yöntem kullanılır.

Eğer belirli bir aralıktaki satırların seçilmesi gerekiyorsa Şekil 4.18'de yapıldığı gibi aralık belirtmek için ':' işareti kullanılır.

```
sr = pd.Series(np.random.rand(5), index=list("abcde"))
```

```
sr
```

```
a    0.737907  
b    0.176606  
c    0.283075  
d    0.495037  
e    0.229827  
dtype: float64
```

```
df2 = pd.DataFrame(sr, columns=['1'])
```

```
df2
```

	1
a	0.737907
b	0.176606
c	0.283075
d	0.495037
e	0.229827

```
df2['2'] = df2['1']*2
```

```
df2
```

	1	2
a	0.737907	1.475814
b	0.176606	0.353212
c	0.283075	0.566150
d	0.495037	0.990074
e	0.229827	0.459654

Şekil 4.12: Serilerden DataFrame Oluşturmak

```
df. abs
df. add
df. add_prefix
df. add_suffix
df. agg
df. aggregate
df. align
df. all
df. any
df. append
df.
```

Şekil 4.13

```
df. values
array([[ 'Ali', 70, 84],
       [ 'Ayşe', 83, 65],
       [ 'Ahmet', 90, 75],
       [ 'Mehmet', 85, 90]], dtype=object)
```

Şekil 4.14

```
df. columns
Index([ 'Ad', 'Final', 'Vize'], dtype='object')

df. Final
0      84
1      65
2      75
3      90
Name: Final, dtype: int64

df[ 'Final' ]
0      84
1      65
2      75
3      90
Name: Final, dtype: int64
```

Şekil 4.15: Sütun Adına Göre Seçme

```
df.ix[1]
```

```
Ad      Ayşe  
Final   65  
Vize     83  
Name: 1, dtype: object
```

Şekil 4.16: İndekse Göre Seçme

```
df.ix[[1,3]]
```

	Ad	Final	Vize
1	Ayşe	65	83
3	Mehmet	90	85

Şekil 4.17

```
df.ix[1:2]
```

	Ad	Final	Vize
1	Ayşe	65	83
2	Ahmet	75	90

Şekil 4.18

Son olarak, eğer framedeki tek bir elemana ulaşılmak istenirse, bunun için önce kolon adı ve sonra indeks numarası yazılarak seçme işlemi gerçekleştirilir. (Şekil 4.19)

```
df['Ad'][3]
```

```
'Mehmet'
```

Şekil 4.19

Tek elemana ulaşmak için kullanılan bir yöntem ise 'loc' metodudur. 'loc' metodunun kullanımı Şekil 4.20'de gösterilmiştir.

```
df.loc[3, 'Ad']
```

```
'Mehmet'
```

Şekil 4.20

Satır/Sütun Ekleme, Değiştirme, Silme

Mevcut frame'de bir elemanı değiştirmek için yukarıda gördüğümüz seçme yöntemleri kullanılır. Seçilen elemana yeni değer atanarak değiştirme işlemi gerçekleştirilmiş olur. Bunun için kullanılacak en iyi yöntem loc metodudur. (Şekil 4.21) (Kaynak: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>)

```
df.loc[2, 'Final'] = 85
df
```

	Ad	Final	Vize
0	Ali	84	70
1	Ayşe	65	83
2	Ahmet	85	90
3	Mehmet	90	85

Şekil 4.21

Yeni bir sütun eklemek için yeni sütunun ismi ve değerleri atanmalıdır. Şekil 4.22'de yer alan örnekte Örnekte yeni değerler Final ve Vize değerlerinin ortalamasının alınması ile oluşturulmuştur. İstenilen değerlerin dizi şeklinde tanımlanmasıyla da yeni sütun oluşturulabilir.

Yeni bir satır eklemek için, Seçme işlemlerinde kullandığımız 'ix' metodu ile yeni indeks numarasını belirleyerek atama yapabiliriz. Örnekte, indeks numarası 4 olan yeni bir öğrenci eklenmiştir. (Şekil 4.21)

Satır ve sütunlar için yapılmak istenen değiştirme işlemleri de ekleme işleminde yapılan işlemlerin tekrarı ile oluşturulur.

```
df['Ortalama'] = (df.Final/2) + (df.Vize/2)
```

```
df
```

	Ad	Final	Vize	Ortalama
0	Ali	84	70	77.0
1	Ayşe	65	83	74.0
2	Ahmet	85	90	87.5
3	Mehmet	90	85	87.5

Şekil 4.22

```
df.ix[4] = ['Fatma', 80, 60, 70]
```

```
df
```

	Ad	Final	Vize	Ortalama
0	Ali	84	70	77.0
1	Ayşe	65	83	74.0
2	Ahmet	85	90	87.5
3	Mehmet	90	85	87.5
4	Fatma	80	60	70.0

Şekil 4.23

```
del df['Vize']
df
```

	Ad	Final	Ortalama
0	Ali	84	77.0
1	Ayşe	65	74.0
2	Ahmet	85	87.5
3	Mehmet	90	87.5
4	Fatma	80	70.0

Şekil 4.24

Tablodan herhangi bir sütunu silmek için 'del' komutu kullanılır. Bu komutun kullanımını Şekil 4.24'te yer almaktadır.

Tabloadan herhangi bir satır silmek için drop metodu kullanılır. Bu metodun kullanımını Şekil 4.25'te gösterilmiştir.

```
df.drop([2])
```

	Ad	Final	Ortalama
0	Ali	84	77.0
1	Ayşe	65	74.0
3	Mehmet	90	87.5
4	Fatma	80	70.0

Şekil 4.25

Dosya Okuma ve Yazma İşlemleri

DataFrameleri oluşturabileceğimiz gibi herhangi bir formatta bulunan veri setlerini de çalışmalarımızda kullanabiliriz. Bu bölümde bunlardan en sık kullanılan csv dosyası okuma ve yazma işlemlerinizi gerçekleştireceğiz. En çok kullanılan dosya okuma ve yazma metodları aşağıda tablo olarak verilmiştir. Farklı türlerdeki verileri (csv, JSON, excel, html vb.) Pandas kütüphanesini kullanarak kolaylıkla çalışmalarımızda kullanabiliriz.

Tablo 4.3: <https://pandas.pydata.org/pandas-docs/stable/io.html>

Format	Veri Türü	Okuma	Yazma
text	CSV	read_csv	to_csv
text	JSON	read_json	to_json
text	HTML	read_html	to_html
binary	MS Excel	read_excel	to_excel
binary	HDF5 Format	read_hdf	to_hdf
SQL	SQL	read_sql	to_sql
SQL	Google Big Query	read_gbq	to_gbq

Csv dosyaları (comma separated value) virgülle ayrılmış verilerden oluşan dosya türüdür. (Şekil 4.26) Herhangi bir dosya türünü çalışma sayfanızda okumak için dosyanın aynı dizinde bulunması gerekmektedir. Dosyayı yükledikten sonra read.csv() metoduna dosyanın ismi yazılarak ulaşılabilir.

Örnekte okutulan csv dosyasından bir frame oluşturulmuştur. Filmler tablosu; filmin puanı, ismi, türü, süresi ve oyuncu listesinden oluşmaktadır. Aşağıda, daha önce anlatılan seçme ve filtreleme yöntemleri kullanılarak bu tablo üzerinden örnekler hazırlanmıştır.

1. İlk örnekte, star_rating yani puanı 9 ve üzerinde olan filmleri listeleme işlemi yapılmıştır. (Şekil 4.27);

```
filmler = pd.read_csv('filmler.csv')
```

```
filmler
```

	star_rating	title	genre	duration	actors_list
0	9.3	The Shawshank Redemption	Crime	142	Tim Robbins, Morgan Freeman, Bob Gunton
1	9.2	The Godfather	Crime	175	Marlon Brando, Al Pacino, James Caan
2	9.1	The Godfather: Part II	Crime	200	Al Pacino, Robert De Niro, Robert Duvall
3	9.0	The Dark Knight	Action	152	Christian Bale, Heath Ledger, Aaron Eckhart
4	8.9	Pulp Fiction	Crime	154	John Travolta, Uma Thurman, Samuel L. Jackson
5	8.9	12 Angry Men	Drama	96	Henry Fonda, Lee J. Cobb, Martin Balsam
6	8.9	The Good, the Bad and the Ugly	Western	161	Clint Eastwood, Eli Wallach, Lee Van Cleef
7	8.9	The Lord of the Rings: The Return of the King	Adventure	201	Elijah Wood, Viggo Mortensen, Ian McKellen
8	8.9	Schindler's List	Biography	195	Liam Neeson, Ralph Fiennes, Ben Kingsley
9	8.9	Fight Club	Drama	139	Brad Pitt, Edward Norton, Helena Bonham Carter
10	8.8	The Lord of the Rings: The Fellowship of the Ring	Adventure	178	Elijah Wood, Ian McKellen, Orlando Bloom
11	8.8	Inception	Action	148	Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen...
12	8.8	Star Wars: Episode V - The Empire Strikes Back	Action	124	Mark Hamill, Harrison Ford, Carrie Fisher

Şekil 4.26

```
filmler.loc[(filmler.star_rating >=9)]
```

	star_rating	title	genre	duration	actors_list
0	9.3	The Shawshank Redemption	Crime	142	Tim Robbins, Morgan Freeman, Bob Gunton
1	9.2	The Godfather	Crime	175	Marlon Brando, Al Pacino, James Caan
2	9.1	The Godfather: Part II	Crime	200	Al Pacino, Robert De Niro, Robert Duvall
3	9.0	The Dark Knight	Action	152	Christian Bale, Heath Ledger, Aaron Eckhart

Şekil 4.27

2. Şekil 4.28’de bulunan, puanı 8.5 üzerinde olan ve türü komedi olan filmleri listelemek için iki sorgu ve (&) koşuluyla birbirine bağlanmıştır.

```
filmler.loc[(filmler.star_rating >= 8.5) & (filmler.genre == 'Comedy')]
```

	star_rating	title	genre	duration	actors_list
25	8.6	Life Is Beautiful	Comedy	116	Roberto Benigni, Nicoletta Braschi, Giorgio Ca...
29	8.6	City Lights	Comedy	87	Charles Chaplin, Virginia Cherrill, Florence Lee
35	8.6	Modern Times	Comedy	87	Charles Chaplin, Paulette Goddard, Henry Bergman
48	8.5	Dr. Strangelove or: How I Learned to Stop Worr...	Comedy	95	Peter Sellers, George C. Scott, Sterling Hayden
52	8.5	The Great Dictator	Comedy	125	Charles Chaplin, Paulette Goddard, Jack Oakie
60	8.5	3 Idiots	Comedy	170	Aamir Khan, Madhavan, Mona Singh

Şekil 4.28

3. Şekil 4.29’da türlere göre toplam film sayılarına ulaşmak için groupby() methodu kullanılmıştır.

```
mt = filmler.groupby('genre')[('title')].count().reset_index(name="total")
mt.head()
```

	genre	total
0	Action	136
1	Adventure	75
2	Animation	62
3	Biography	77
4	Comedy	156

Şekil 4.29

Örnekte, Groupby() metodu ile filmler türlerine göre sınıflandırılmış, Count() metodu ile de her türe ait film sayısı hesaplanmıştır. Oluşturulan bu yeni tablo mt adında yeni bir frame atanmış ve bu framedeki ilk 5 satır head() metodu ile listelenmiştir.

Son örnekte hazırlanan mt tablosunu bir csv dosyasına yazdırmak için to_csv() metodu kullanılarak oluşturulacak dosyanın adı tanımlanmıştır. (Şekil 4.30)

```
mt.to_csv('mt.csv')
```

Şekil 4.30

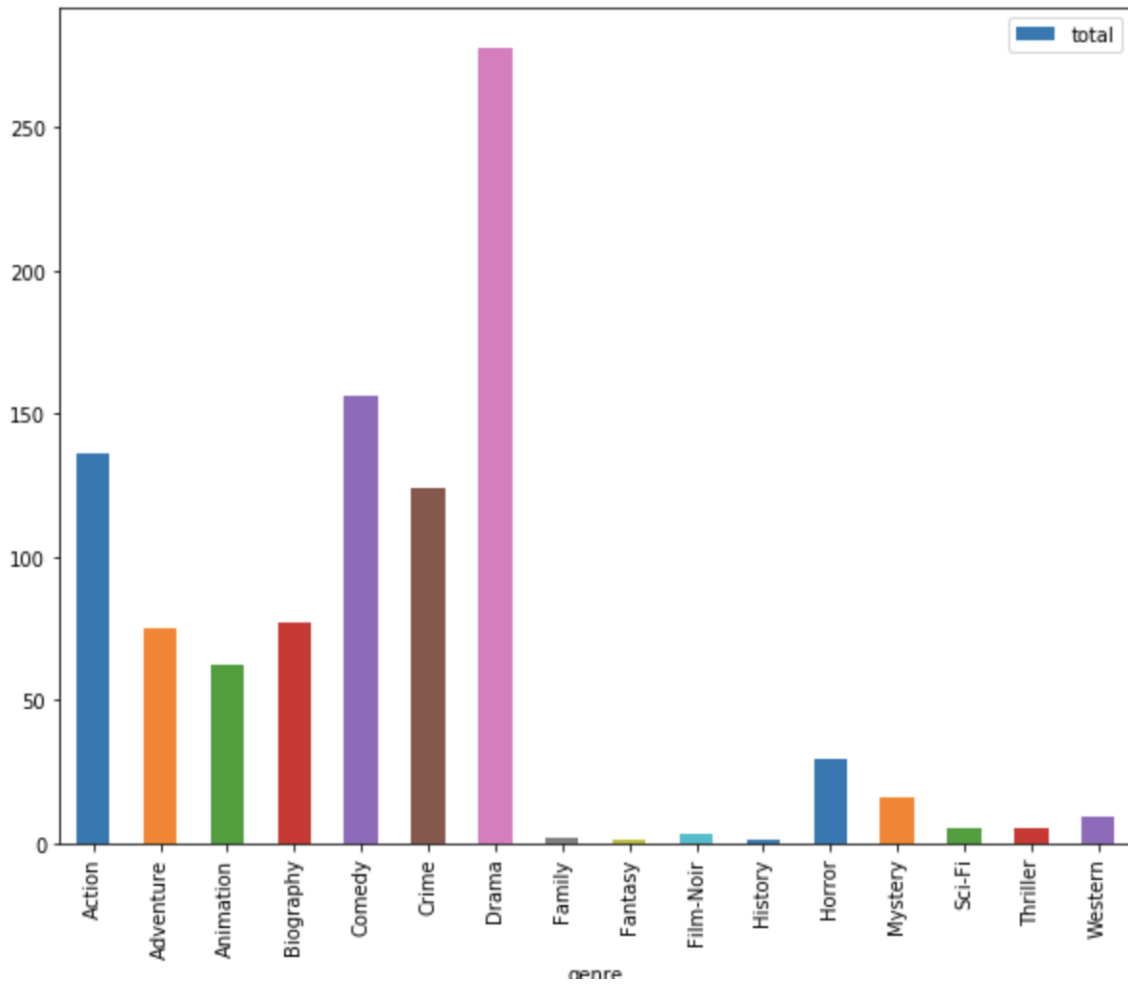
Bu dosyaya Home sekmesinden ulaşarak gerekli ise kaydedilir.

4.2.3 Veri Görselleştirme (Matplotlib Kütüphanesi)

Python ile veri görselleştirmek için kullanılan matplotlib kütüphanesi, Pandas ile yapılan çalışmalarda çok kullanılmaktadır. Bu kütüphane hazırladığımız tabloları kolaylıkla grafiğe dönüştürmemizi sağlamaktadır. Kütüphaneyi kullanabilmek için öncelikle import edilmesi gerekmektedir. Örnekte, mt tablosunun tür ve toplam film sayısına göre sütun grafiği çizdirilmiştir. (Şekil 4.31)

```
import matplotlib.pyplot as plt
%matplotlib inline
mt.plot.bar(x="genre",y="total",figsize=(10,8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1145cdf60>
```



Şekil 4.31

4.3 Kaynakça

[1] F. Nelli, Python Data Analytics: Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language,, Apress, 2015.

[2] J. P. Mueller, Beginning Programming with Python For Dummies, John Wiley and Sons,Inc., 2014.

[3] W. W. McKinney, Python for Data Analysis, O'Reilly Media,Inc., 2017.

[4] T. Petrou, Pandas Cookbook, Packt Publishing, 2017.

[5] A. Fandango, Python Data Analysis, Packt Publishing, 2017.

[6] M. Heydt, Learning Pandas, Packt Publishing, 2017.

[7] A. Bronshtein, "A Quick Introduction to the "Pandas" Python Library," 18 Nisan 2017. [Çevrimiçi]. Available: <https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673>. [Erişildi: 20 Kasım 2018].

[8] Compare Business Products, "The Top Ten ERP Vendors," 2018. [Çevrimiçi]. Available: https://www.enterpriseinnovation.net/files/whitepapers/top_10_erp_vendors.pdf. [Erişildi: 20 Kasım 2018].

[9] A. Kandpal, "An Introduction to Data analysis with Pandas," 5 Aralık 2017. [Çevrimiçi]. Available: <https://medium.com/@harrypotter0/an-introduction-to-data-analysis-with-pandas-27ecbce2853>. [Erişildi: 20 Kasım 2018].

[10] A. Khashtamov, "Data Analysis with Pandas," 13 Ağustos 2017. [Çevrimiçi]. Available: <https://khashtamov.com/en/pandas-data-analysis>. [Erişildi: 20 Kasım 2018].

[11] C. Moffitt, "Pandas Pivot Table Explained," 29 Aralık 2014. [Çevrimiçi]. Available: <http://pbpython.com/pandas-pivot-table-explained.html>. [Erişildi: 20 Kasım 2018].

[12] K. Oleinik, "Python for Data Analysis," 2017. [Çevrimiçi]. Available: <https://www.bu.edu/tech/files/2017/09/Python-for-Data-Analysis.pdf>. [Erişildi: 20 Kasım 2018].

[13] L. Sugandha, "Working with pandas DataFrames," 23 Şubat 2018. [Çevrimiçi]. Available: <https://hub.packtpub.com/working-with-pandas-dataframes>. [Erişildi: 20 Kasım 2018].

[14] C. Talha, "imdb-data-pandas-visualization," 2018. [Çevrimiçi]. Available: https://github.com/thechaudharysab/imdb-data-pandas-visualization/blob/master/data/imdb_1000.csv. [Erişildi: 20 Kasım 2018].

[15] J. VanderPlas, "Introducing Pandas Objects," 13 Temmuz 2015. [Çevrimiçi]. Available: <https://www.oreilly.com/learning/introducing-pandas-objects>. [Erişildi: 20 Kasım 2018].

[16] K. Willems, "Pandas Tutorial: Data Wes McKinneyFrames in Python," 23 Şubat 2018. [Çevrimiçi]. Available: <https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python>. [Erişildi: 20 Kasım 2018].

[17] W. McKinney; PyData Development Team, "pandas: powerful Python data analysis toolkit," 12 Haziran 2018. [Çevrimiçi]. Available: <http://pandas.pydata.org/pandas-docs/stable>. [Erişildi: 20 Kasım 2018].



5. Tensorflow

Turkish Food Recognition Using Deep Learning With Tensorflow

Malik Bennabi, Conrad Reuben Japhet

Abstract

This paper consists of a brief study of possible application of Deep Learning techniques on Turkish Food Recognition. The present work starts with a brief background section, followed with an experiment using Google's Tensorflow Deep Learning library was performed. This included setting-up the environment on Linux, Retraining the tensorflow-for-poets provided model, testing the retrained model and putting it into production on a mobile device.

5.1 Introduction

Creating intelligent machines has been the dream of inventors for a long time, and this dates back to civilizations older than Greece[3]. A mythical figure like Pygmalion may be interpreted as a legendary inventor, and Galatea may be regarded as artificial life. Today, artificial intelligence (AI) is a growing field with active research and many practical applications[3]. Speech or images recognition, diagnoses in medicine and even prediction of trends in the stock market are few examples of possible application fields of artificial intelligence (AI). Artificial intelligence has proved to be effective and efficient in solving complex problems, however, still encounters challenges in solving those problems that the human brain processes in an easy way and computers don't, such as image recognition and classification[3].

This use-case project is an overview of the solutions to these more intuitive problems. This kind of solutions are to allow computers to learn from experience and understand the world in terms of a hierarchy of concepts, with each concept defined through its relation to simpler concepts. Accumulating knowledge from experience, helps avoid the need for human operators to specify all the knowledge and steps that the computer needs to perform a given task. Ian Goodfellow, Yoshua

Bengio, and Aaron CourvilleThe wrote in their Deep Learning book's introduction: "hierarchy of concepts enables the computer to learn complicated concepts by building them out of simpler ones. If we draw a graph showing how these concepts are built on top of each other, the graph is deep, with many layers. For this reason, we call this approach to AI deep learning."[3]

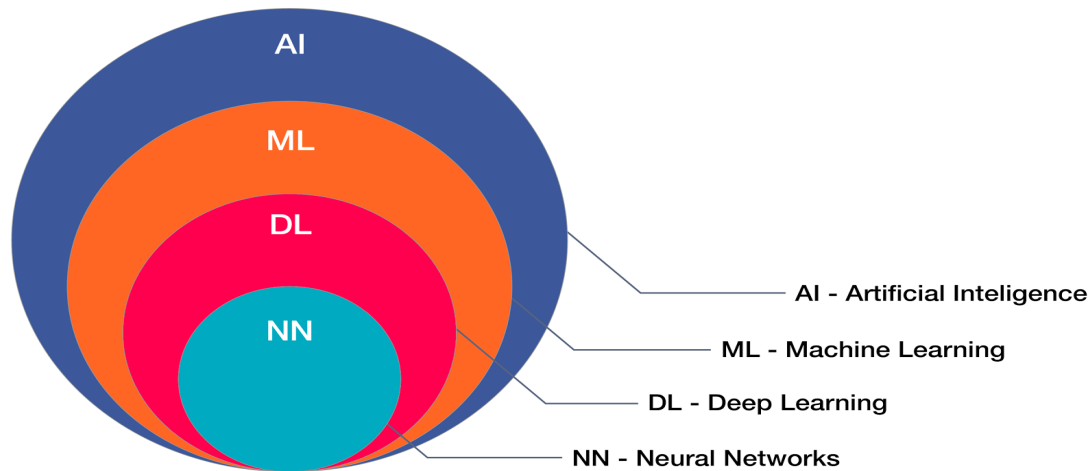
Artificial intelligence, Machine learning, and deep learning are hot topics. Inspiring use cases are being seen on a daily basis. A very prominent topic in Artificial Intelligence field, is Deep Learning, and has been in the focus for few years now. It is especially known for its breakthroughs in fields like Computer Vision and Computer Games.

5.2 Background

5.2.1 What is Deep Learning?

To understand what deep learning is, we first need to understand the relationship between deep learning, machine learning, neural networks, and artificial intelligence[9].

The best way to think of this relationship and visualize it is to represent them as concentric circles:



Şekil 5.1: Relationship between AI, ML, DL, and NN Source: Google images - "Unknown"

At the outermost ring you have artificial intelligence (using computers to reason). One layer inside of that is machine learning. One level closer to the center, artificial neural networks and deep learning take place. Generally speaking, deep learning is a closer name for artificial neural networks as the word "deep" refers to the number of layers the network has, thus, the depth of the network[9].)

Neural networks' inspiration came from the cerebral cortex. At the first (basic) level is the perceptron, the mathematical representation of a biological neuron. Similar to the cerebral cortex, several layers of interconnected perceptrons can be in between the two ends of the network.

The first layer is the input layer where each node takes an input, and passes its output as the input to each node in the next layer. There are generally no connections between nodes in the same layer and the last layer produces the outputs of the network. Middle parts of the network are called hidden layers. These neurons are triggered and activated by nodes in the previous layer.

5.3 Experiment

In this part our approach for testing tensorflow object detection and image classification as well as the doing of the testing will be explained.

5.3.1 Tensorflow

According to Google, Tensorflow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well[2]. TensorFlow provides multiple APIs. The lowest level API- -TensorFlow Core– provides complete programming control. The higher level APIs are built on top of TensorFlow Core. These higher level APIs are typically easier to learn and use than TensorFlow Core. In addition, the higher level APIs make repetitive tasks easier and more consistent between different users. A high-level API like `tf.estimator` helps manage data sets, estimators, training and inference. (Martín A., Paul B., Jianmin C., Zhifeng C., Andy D. et al. 2016)

MobileNets

MobileNets is a family of computer vision models for TensorFlow, designed to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application[1]. MobileNets are small, low-latency, low-power models built to meet the resource limitations of a variety of use cases, these constitute of 30 layers and the design of the network is quite straightforward[1]:

1. Convolutional layer with stride 2
2. Depthwise layer
3. Pointwise layer that doubles the number of channels
4. Depthwise layer with stride 2
5. Pointwise layer that doubles the number of channels
6. Depthwise layer
7. Pointwise layer
8. Depthwise layer with stride 2
9. Pointwise layer that doubles the number of channels

and so on. . .

5.3.2 Environment set-up

After reading about the possible environment setups available to work on Tensorflow on Linux mint 18.1, the following have been chosen for the better adaptability they provide with the used operating system.[4]

The first step was installing the programming language libraries on the operating system through the command line to make sure it sets the system into it, the chosen language is **Python 3** and was installed with:

```
$ sudo apt-get install python3-pip python3-dev
```

Then considering the hardware and computing power we had, **TensorFlow with CPU support only** tensorflow was installed on terminal, using the virtual environment **virtualenv**.

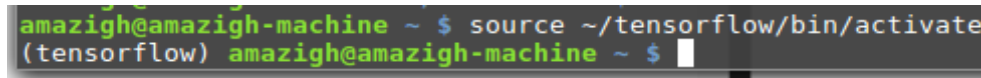
Virtualenv is a virtual Python environment isolated from other Python development, incapable of interfering with or being affected by other Python programs on the same machine.[7]

```
$ sudo apt-get install python3-pip python3-dev python-virtualenv
$ sudo apt install virtualenv
$ virtualenv --system-site-packages -p python3 ~/tensorflow
```

Activating tensorflow:

```
$ source ~/tensorflow/bin/activate
```

The prompt should change and be headed by (tensorflow) like: at the end of this process, an upgrade



```
amazigh@amazigh-machine ~ $ source ~/tensorflow/bin/activate
(tensorflow) amazigh@amazigh-machine ~ $
```

was run to insure the latest features of the library.

```
$ pip3 install --upgrade tensorflow
```

5.3.3 Use-case

In this use-case, transfer learning has been used, which means the test has been started with a model that has been already trained on another problem. Then has been retrained on a similar problem. Building a deep learning model from scratch and training it can take days, but transfer learning can be done in a dramatically shorter period.

The model in use, has been trained on the ImageNet Large Visual Recognition Challenge dataset. The model can be adapted to the needs of the wanted experiment and a choice of model architectures is available, so as to determine the right trade off between speed, size and accuracy for the problem. This use-case has been inspired and derived from a google' codelabs tensorflow for poets model and tutorials[6].

First we started by cloning the git repository of the code used to pre-train the model from tensorflow for poets.

```
$ git clone https://github.com/googlecodelabs/tensorflow-for-poets-2
```

5.3.4 Re-training the model

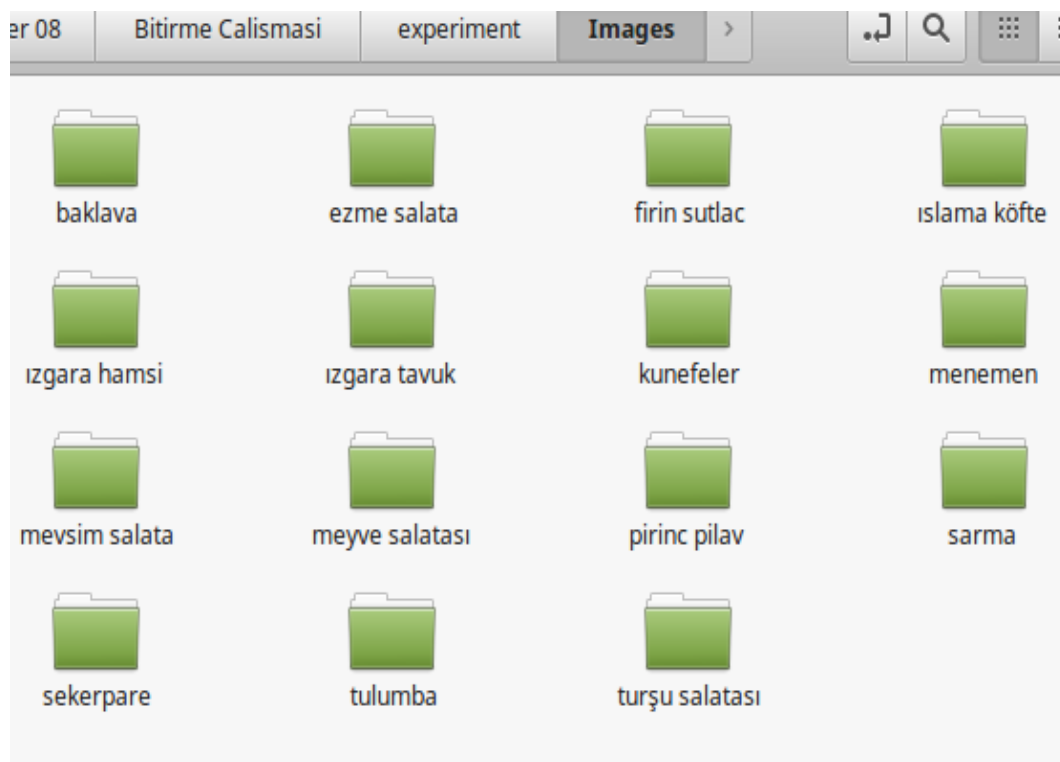
Preparing the training images: The goal was to train the model on 15 different types of food specifically Turkish food. 15 sets of images have been downloaded from Google and separated into 15 folders, each containing a given category. In our case the categories are:

- Baklava
- Sekerpare
- Firin Sutlaç
- Tulumba
- Kunefe
- Ezme Salatasi

- Mevsim Salatasi
- Meyve Salatasi
- Tursu Salatasi
- Islama Kofte
- Izgara Tavuk
- Izgara Hamsi
- Sarma
- Menemen
- Pirinç Pilavi

In addition to some more images in a separate folder to test the model after the training.

In total 4.563 images were used in the training phase, with an average of 300 images for each category.



Running the training , after setting-up the needed settings, the training was initiated using the following command [6]:

```
python -m scripts.retrain \
--bottleneck_dir=tf_files/bottlenecks \
--how_many_training_steps=1000000 \
--model_dir=tf_files/models/ \
--summaries_dir=tf_files/training_summaries/"${ARCHITECTURE}" \
--output_graph=tf_files/retrained_graph.pb \
--output_labels=tf_files/retrained_labels.txt \
--architecture="${ARCHITECTURE}" \
```

```
—image_dir=tf_files/images
```

```
(tensorflow) amazigh@amazigh-machine ~/Desktop/Semester 08/Bitirme Calismasi/experiment/code $ python -m scripts.retrain --bottleneck_dir=tf_files/bottlenecks --how_many_training_steps=1000000 --model_dir=tf_files/models/ --summaris_dir=tf_files/training_summaries/"${ARCHITECTURE}" --output_graph=tf_files/retrained_graph.pb --output_labels=tf_files/retrained_labels.txt --architecture="${ARCHITECTURE}" --image_dir=tf_files/images
>> Downloading mobilenet_v1_0.50_224_frozen.tgz 62.5%
```

Note: These models are networks with millions of parameters that can differentiate a large number of classes. Only the final layer of the network was trained, therefore, the training was supposed to end in a reasonable amount of time. However, due to the large number of requested iterations that was 1 million iteration for 4.563 images the used hardware (CPU) stopped due to overheating after about 6 hours of training and only reaching about 11000 iterations. As a consequence of the above mentioned hardware issue and limitation, the model was only trained over a bit more than 11000 iterations.

5.3.5 Using the Retrained Model

In this part, few images were used to test the retrained model. There images are new and the model has not been trained on them. The images were input to the model using the following script[6]:

```
python -m scripts.label_image \
—graph=tf_files/retrained_graph.pb \
—image=tf_files/name_of_the_image.jpg
```

Test of Sarma image recognition:

The model model predicted with 84% accuracy that the image is sarma. Considering the quality of the image and given that there are other food items in the picture, we can say that the accuracy of this output is quite high.

Test of Izgara Tavuk image recognition:

The model was 76% that the image contained an Izgara Tavuk. Considering that this food item does have many angles and does not come in a standard shape ,this type of accuracy is expected.

Setting the model into Android

The last step was optimizing and compressing the model and setting it up in an android App and then run it. To do so, a Tensorflow-Android converter, which is provided by Google along with the tensorflow-for-poets library, was used, facilitating the integration of the model into an android device [6].

Adding the model files to the android project The project is configured to search for a graph.pb, and a labels.txt files in the android/tfmobile/assets directory. Therefore, this is the perfect place to keep the new model[6].

```
cp tf_files/rounded_graph.pb android/tfmobile/assets/graph.pb
cp tf_files/retrained_labels.txt android/tfmobile/assets/labels.txt
```



(a) Sarma

```
Evaluation time (1-image): 0.161s
sarma 0.84352905
ezme salata 0.08194717
zgara tavuk 0.04536394
mevsim salata 0.018963708
baklava 0.0056189587
```

(b) Recognition Results

Şekil 5.2: Sarma Recognition



(a) Tavuk Izgara

```
Evaluation time (1-image): 0.154s
zgara tavuk 0.76119655
slama k fte 0.16231318
sekerpare 0.051555555
baklava 0.012797756
ezme salata 0.008074546
```

(b) Recognition Results

Şekil 5.3: Tavuk Izgara Recognition

Setting up the *ClassifierActivity.java* activity The TensorFlow Interface used by the app requires that the results are pulled by name[6]. The app is set up to read the output of the baseline MobileNet, named "*mobilenet_v1/predictions/softmax*". The output node for our model has to be



Şekil 5.4: Real time recognition of two plates

hard coded as *"final_result"*:

```
private static final String INPUT_NAME = "input";
private static final String OUTPUT_NAME = "final_result";
```

After saving these critical changes, the app can be installed on an android device and run with no errors.

5.4 Conclusion

Considering all the resources at our disposal the results we got from the trained model were, although expected, quite adequate. Although we used around 300 images for each picture category and at a total of 11000 iterations for each the model gave quite accurate results. It needs to be said that the model used (MobileNets) also had an effect on the overall accuracy. Considering also the array of deep learning frameworks available e.g. Caffe, Chainer, Gluon, Keras... etc, the Tensorflow library proved to be robust and fit this kind of work. Another important result that was concluded after experimenting training a neural network on a CPU (Central Processing Unit) is the very slow speed and hardware damage issue as the training progresses. For future better training, GPUs (Graphical Processing Unit) should be used and preferably GPUs with high cooling capabilities, where cloud based training resources come to be useful, however, the right and convenient environment should be carefully chosen to be able to use the training resulted models and as well as being able to put these model to production.

Acknowledgement

We would like to acknowledge the supervision of Prof.Dr. İsmail Hakkı CEDİMOĞLU, head of Information Systems Engineering department at Sakarya University.

5.5 Kaynakça

[1] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam, Google Inc.

[2] TensorFlow: Large-scale machine learning on heterogeneous systems, 2015: Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Software available from tensorflow.org

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning <http://www.deeplearningbook.org>

[4] Data Camp TensorFlow Tutorial <https://www.datacamp.com/community/tutorials/tensorflow-tutorial>

[5] Github Tensorflow repositories <https://github.com/tensorflow/tfjs-models/tree/master/mobilenet>

[6] Google Tensorflow for poets. <https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/>

[7] Deep Learning with TensorFlow and Python <https://pythonprogramming.net/tensorflow-introduction-machine-learning-tutorial/>

[8] Images. <https://images.google.com/>

[9] Charlie Crawford, Introduction to Deep Learning. <https://blog.algorithmia.com/introduction-to-deep-learning-2016/>

5.6 About the authors

Malik Bennabi is a senior student of Information Systems Engineering at the University of Sakarya (Turkey). He has been a student of Electrical and Electronic Engineering at the IEEE University of Boumerdès (Algeria). He has completed the Cisco CCNA in 2011. He has been an Erasmus+ scholar at the School of Computer Engineering, University of Oviedo (Spain) during 2016-2017. He has successfully completed several Entrepreneurship programs in different countries. He is a co-founder of I-sen, an emotional education startup in Asturias (Spain). He is a winner of several entrepreneurship and innovation competitions around the world and has been a mentor at several hackathons including the IoT Olympiads Boston 2014 (USA). He is a polyglot, speaking fluently 7 languages and comfortable with 2 and continually learning. He successfully completed the summer research program provided by the University of South Florida (USA) and the University of Oviedo (Spain). His research interests include Machine and Deep learning, Artificial intelligence, Blockchain, Entrepreneurship and Innovative learning. Malik.bennabi@ogr.sakarya.edu.tr

Conrad Reuben Japhet is a senior student of Information Systems Engineering at the University of Sakarya (Turkey). He has been part of the Erasmus + alumni at Lublin University of Technology in Poland. His research and fields of interest include machine learning, data analysis, information gathering, and database administration. He has dreams of making applications that change societal norms through integrating people like never before. Conrad.japhet@gmail.com



6. R ile Kaba Kümeleme ve Uygulamaları

R Programlama Dili ile Kaba Kümeleme Uygulamaları

Doç. Dr. Safiye TURGAY, Prof. Dr. Orhan TORKUL, Arş. Gör. Merve ŞİŞÇİ,
Doç. Dr. Tahsin TURGAY

6.1 Giriş

Kaba küme ve bulanık kaba küme kavramları ve bu kavramlara ait temel özellikler R programlama dili kullanılarak çeşitli örnekler üzerinde uygulamalı olarak gösterilmiştir. Kaba küme teorisi ve bulanık kaba küme teorisi; özellikleri, avantajları ve kullanım alanları ile birlikte tanımlanmıştır. Aynı zamanda, kaba kümelemede yer alan nitelik azaltımı ve kural elde edilmesi sürecine ilişkin algoritma ile kaba kümeleme ve bulanık kaba kümeleme için geliştirilen bulanık 'RoughSets' Paketinin temel kullanım yapısı, komutlar ve özellikleri uygulamalı olarak R programı üzerindeki fonksiyonlar ve işlem basamakları detaylı olarak gösterilmiştir. Yazılımda Göğüs Kanseri Teşhisi veri seti kullanılarak uygulama kapsamı tanıtılmıştır.

Kaba küme kavramı, mantık olarak belirsiz bilgiyi analiz eden kural çıkarımı için klasik küme teorisinden geliştirilmiş bir yaklaşımdır ve Pawlak (1982) tarafından tanımlanmıştır. Özellikle bilgisayar ortamında R programlama dili ile ilgili yazılım kodlarının uygulamalı olarak çalıştırılması oldukça önemlidir. Kaba kümeleme yaklaşımı, özellikle büyük verilerin analizi ve veritabanı yönetiminde gerekli olan sınıflandırma ve analiz işlemlerinde eksik ve belirsiz veri yapılarına rağmen etkin işlem ve analiz yapabilme yeteneğine sahiptir (Tsumoto, 1998). Kaba küme yaklaşımı, veri madenciliğinde sıklıkla uygulanan yöntemlerden birisidir (Ziarko, 1995; Turgay vd. 2017,2018). Kaba kümelemenin en büyük avantajı, bulanık kümeleme teorisindeki ve istatistikte yer alan olasılık veya üyelik derecesi gibi veriler hakkında herhangi bir ek bilgiye ihtiyaç duymamasıdır (Phuong vd. 2001). Kaba küme teorisi, belirsizlik ve bilgi belirsizliği ile başa çıkmak için kullanılan matematiksel bir araçtır (Pawlak vd. 1991; 1994;1995). Birçok pratik uygulamada çok etkili olduğu kanıtlanmıştır. Ancak, kaba küme teorisinde, hatanın tanımı için oluşturulan yapı oldukça basittir (Li ve Wang,

2004). Bu nedenle, kaba kümeleme tarafından oluşturulan kurallar genellikle kararsızdır ve düşük sınıflandırma doğruluğuna sahiptir.

Kaba kümeleme sadece veri tabanlarından bilgi madenciliği için öğrenme aktivitesini hızlandırmak ve anlaşılabilirliğini kolaylaştırmak için kullanılmak değil aynı zamanda, geliştirilen algoritma ile karar tablosunda yer alan verileri akıllıca indirgenmesi ve istenilen koşullarda filtrelemek için de kullanılabilir. Bu fonksiyonları yerine getirmek için çeşitli algoritmalar geliştirilmiştir. Bu çalışmada bütünleştirilen karar tablosundan verimli sınıflandırma kurallarını çıkarmak için Shen ve Chouchoulas, (2002) tarafından geliştirilen Quick-Reduct algoritması uygulama üzerinde gösterilmiştir. Bilgi (kural kümesi) azaltılmış karar tablosundan ilgili veriler çekilerek kaba kümeleme yaklaşımı kullanılarak veriler sınıflandırılmıştır.

Bu bölüm sırasıyla kaba kümelemede temel kavramlar ve özellikleri, bulanık kaba küme teorisi, RoughSets Yazılım paketi, komutlar ve özellikleri, uygulama ve sonuç kısımlarından oluşmaktadır.

6.2 Kaba Kümelemede Temel Kavramlar

Kaba küme yaklaşımı son yıllarda büyük verilerin analizi, içermiş oldukları eksik ya da belirsiz yapıdan dolayı tercih edilmekle birlikte veri madenciliği yöntemleri ile de büyük verilerin analizine yardımcı olan bir yaklaşımdır. Kaba kümeleme ile büyük verilerin sınıflandırılması, gelişen bilgisayar teknolojisi ile birlikte bulanık kaba kümeleme mantığının da geliştirilmesi ile birlikte daha belirsiz ve eksik olan verilerin hızlı bir şekilde analiz edilmesine olanak sağlayabilmektedir. Kaba küme yaklaşımının en önemli avantajlarından birisi, istatistikte yer alan olasılık veya üyelik derecesi gibi ek bilgileri kullanmaz. Bulanık kaba kümelemede ise eksik olan bilgiler 0-1 aralığında gruplandırılarak kümeleme sayesinde kuralların elde edilmesi sağlanabilmektedir. Kaba kümeleme ve bulanık kaba kümeleme, belirsizlik ve bilgi belirsizliği dezavantajını avantaja çeviren bir yaklaşımdır. Kaba kümelemenin en önemli dezavantajı ise düşük doğruluk değerine sahip sınıfa sahip olması ve beraberinde kararsız bir kural yapısına sahip olabilmesidir. Geliştirilen algoritmalar ve detay çalışmalarla bu eksikliğin etkisi azaltılmaya çalışılmaktadır.

Bu bölümde kaba küme kavramını oluşturan temel fonksiyonlara yer verilmiştir. Özellikle kaba küme kavramının fonksiyon yapısı ile birlikte sistem üzerinde temsil edilmesi Bilgi Sistemi adını verdiğimiz yapı ile mümkün olmaktadır. Bu yapı içerisinde problem ve çözüm uzayını kapsayan evren ile birlikte sonlu öznitelik kümesi ve öznitelik değerleri de yer almaktadır.

Tanım 1: Bilgi sistemi gösterimi

Bilgi sistemi $S = \langle U, A, V, f \rangle$ olduğuna göre

U evren olarak adlandırılan sonlu bir nesnel kümesi,

A sonlu bir öznitelik kümesi

V bilgi fonksiyonu

a özniteliği $V = U_a \in AV_a$

V_a Değerler kümesinde yer alan değerleri gösterir.

f özniteliği $U \times A \rightarrow V, \forall a \in A$ için, $f(x, a) \in V_a$ fonksiyonu ile ifade edilir.

Sınıflandırma problemlerinde; seri durum özelliği ve seri karar özelliğinin karar tablosu içerisindeki yapısı ve gösterimi aşağıdaki gibidir;

C seri durum özniteliği

D seri karar özelliği

karar tablosu, $A = C \cup D$ ve $C \cap D = \emptyset$ olduğu varsayılır.

Tanım 2: Ayrılmazlık ilişkisi

$S = \langle U, A, V, f \rangle$ bir bilgi sistemi içerisinde, her $[P \subseteq A, U$ 'deki $IND(P)$ ayrılmazlık ilişkisi ise (Li ve Wang, 2004) aşağıdaki gibidir:

$$IND(P) = \{(x, y) \in U \times U : f(x, a) = f(y, a) \forall a \in P\} \quad (6.1)$$

Tanım 3: Alt ve üst yaklaşımlar, sınır bölgesi tanımları (Riza vd. 2014b)

Alt yaklaşım: kümeye kesin olarak ait olan elemanlar

Üst yaklaşım: kümeye ait olma olasılığı olan elemanları içerir.

R_x bir x elemanın denklik sınıfını göstermek üzere yaklaşımları ve sınır bölgeleri şu şekilde ifade edilir;

$$X \text{ için } R \text{ -alt yaklaşımı: } R_*(x) = Y \{R(x) : R(x) \subseteq X\} \quad (6.2)$$

$$X \text{ için } R \text{ -üst yaklaşımı: } R^*(x) = Y \{R(x) : R(x) \cap X \neq \emptyset\} \quad (6.3)$$

$$X \text{ için } R \text{ -sınır yaklaşımı: } RN_R(X) = R^*(x) - R_*(x) \quad (6.4)$$

Tanım 4: Öznitelik azaltma

Öz nitelik azaltma (özellik seçimi), bazı kriterlere göre öznitelik alt kümesinin, verilerin sınıflandırma ilişkisini temsil etmesi yönünden, tüm özniteliklerin optimal bir alt kümesini bulma işlemidir. Bir sınıflandırıcıya sağlanan iyi bir öznitelik alt kümesi, sınıflandırma doğruluğunu artırabilir, hesaplama süresini koruyabilir ve sonuçlarını basitleştirebilir (Last vd. 2001). Öznitelik indirgemede silinen öznitelikler iki kategoride ele alınabilir;

1. herhangi bir sınıflandırma yeteneğine sahip olmayan alakasız ve yedek öznitelikler
2. gürültülü öznitelikler

Birinci kategori, alakasız bir özellik, sınıflandırmaya herhangi bir değer katmaz ve hiçbir şekilde etkilemez. Dolayısıyla gereksiz bir özellik, sınıflandırmaya yeni bir şey eklemeyi (Li ve Wang, 2004). İkinci kategori ise gürültülü öznitelikleri temsil eder ve bu özellikler bazı sınıflandırma kabiliyetlerini temsil eder, ancak bu özellik gürültünün etkisiyle gerçek sınıflandırma ilişkisinin değerini bozar (Swiniarski ve Skowron, 2003; Beynon, 2001).

Genel olarak, kaba küme teorisi, çok sayıda öznitelik içeren büyük bir veri tabanından alakasız ve gereksiz özniteliklerin azaltılmasını hedefler (Jelonek vd. 1995). Örnek olarak, bağımlılık (yakınsama kalitesi, sınıflandırma kalitesi) (Li ve Wang, 2004) ve entropi (Düntschi, 1998; Li vd. 2004) kaba küme teorisinde en yaygın iki önemli azaltma ölçütüdür.

Öznitelik alt kümelerini ölçmek için kullanılırken, yüksek öznitelik alt kümesi bazı gürültülü öznitelikler içerebilir ve sınıflandırıcının genelleştirmesini bozabilir ve gürültüyü aşırı tahmin edebilir ya da yararlı öznitelikleri silebilir (Chen vd. 2016). Bu bahsedilen tüm özellikler aşağıda yer alan örnek üzerinde gösterilmiştir.

Bir öznitelik, $S = \langle U, C \cup D, V, f \rangle$ karar tablosunda, $P (P \subseteq C)$ öznitelik seti, aşağıdaki koşulları sağlayan C niteliklerin indirgenmesi durumudur:

$$\gamma^P(D) = \gamma^C(D) \text{ ve } \gamma^P(D) \neq \gamma^{P'}(D) \forall P' \subset P \quad (6.5)$$

$$\gamma\{X, Y\}(\{d\}) = \frac{|P_{\{X, Y\}}(\{d\})|}{|U|} \quad (6.6)$$

Koşul özniteliklerinin (C) bir indirgemesi, karar sınıflarını C ile aynı ayırt etme yeteneği ile ayırt edebilen bir alt kümedir ve indirgemedeki özniteliklerin hiçbiri ayırt edici kabiliyetini azaltmadan ortadan kaldırılamaz.

$\gamma_P(Q) = \gamma_{P-\{a\}}(Q)$ durumunda Q'nun P'ye göre vazgeçilmez durumudur. Aynı şekilde P'de Q'ya göre vazgeçilmez durumdur.

Örnek 1: Bilgi fonksiyonunun gösterimi ve kaba kümeleme uygulaması

Bu örnek uygulamasında evren U ile temsil edilmiş ve a_1, a_2 ve a_3 niteliklerine sahip 10 adet veri yer almaktadır (Tablo 6.1). V_1, V_2 ve V_3 öznitelikleri, a_1, a_2 ve a_3 nitelikler kümesinde yer alan değerlerden oluşmaktadır.

Tablo 6.1: Örnek 1' e ait temel veriler (Hasegawa vd., 2009; Walczak ve Massart, 1999; Jensen vd., 2014)

U	a_1	a_2	a_3
x_1	4	1	5
x_2	5	4	1
x_3	4	1	5
x_4	4	4	5
x_5	1	1	6
x_6	1	1	4
x_7	5	4	1
x_8	1	1	6
x_9	4	1	5
x_{10}	5	4	1

V_1 1, 4, 5 veri kümesine ait $a_1 = 5, a_2 = 4$ ve $a_3 = 1$ koşulunu sağlayan veri kümesi ise $\{x_2, x_7, x_{10}\}$ 'dur. $a_1 = 4, a_2 = 1$ ve $a_3 = 5$ koşulunu sağlayan veri kümesi ise $\{x_1, x_3, x_9\}$ 'dur (Tablo 6.2).

$d = 1$ değeri V_1 şartını sağlayan $a_1 = 4, a_2 = 1$ ve $a_3 = 5$ değeri için atanırken, $d = 4$ değeri V_1 şartını sağlayan $a_1 = 5, a_2 = 4$ ve $a_3 = 1$ değeri için atanmış; $d = 5$ değeri ise $a_1 = 1, a_2 = 1$ değerleri için atanmıştır. Karar niteliği değerine göre oluşturulan sınıflar ise

$simf_1 = \{x_1, x_3, x_9\}$ ile $simf_2 = \{x_2, x_4, x_7, x_{10}\}$ ve $simf_3 = \{x_5, x_6, x_8\}$ değerlerinden oluşturulmuştur.

$U = \{x_1, x_2, x_3, \dots, x_{10}\}; A = \{a_1, a_2, a_3\}; V_1 = \{1, 4, 5\}; V_2 = \{1, 4\}; V_3 = \{1, 4, 5, 6\}$

Veri kümesindeki özdeş nesnelere $R(x_1) = R(x_3) = R(x_9) = \{x_1, x_3, x_9\};$

$R(x_2) = R(x_7) = R(x_{10}) = \{x_2, x_7, x_{10}\};$

$R(x_5) = R(x_8) = \{x_5, x_8\}; R(x_6) = \{x_6\}$

U'nun

$$X = \{x_1, x_3, x_4, x_5, x_9\}$$

Tablo 6.2: Koşul durumlarının gösterimi

U	a_1	a_2	a_3
x_1	4	1	5
x_2	5	4	1
x_3	4	1	5
x_4	4	4	5
x_5	1	1	6
x_6	1	1	4
x_7	5	4	1
x_8	1	1	6
x_9	4	1	5
x_{10}	5	4	1

Tablo 6.3: Veriler üzerinde öznitelik azaltımının uygulanması

U/A	a_1	a_2	a_3
$R1 = \{x_1, x_3, x_9\}$	4	1	5
$R2 = \{x_2, x_7, x_{10}\}$	5	4	1
$R3 = \{x_4\}$	4	4	5
$R4 = \{x_5, x_8\}$	1	1	6
$R5 = \{x_6\}$	1	1	4

Kaldırılan öznitelik değeri

Göz ardı edilen öznitelik değeri

alt kümesi için;

$R_*(x) = \{x_1, x_3, x_9\}$; $R^*(x) = \{x_1, x_3, x_4, x_5, x_8, x_9\}$; $RN^*(x) = R^*(x) - R_*(x) = \{x_5, x_8\}$ (Tablo 6.4)

Bu örnek ile 3 şart niteliği $\{a_1, a_2, a_3\}$ ve 1 karar niteliği $\{d\}$ 'nin gösterim biçimi $sinif1 = \{x_1, x_3, x_9\}$; $sinif2 = \{x_2, x_4, x_7, x_{10}\}$; $sinif3 = \{x_5, x_6, x_8\}$ $C = 0$ değerini sağlayan veri yapıları ise 2, 3, 5 ile birlikte ifade edilmiştir.

Örnek 2: Kaba Kümelemede Öznitelik Azaltımı ile ilgili uygulama

10 veriden oluşan temel veri kümesinde $\{a_2, a_3\}$ nitelikleri dikkate alınarak kural kümesi tanımlanmıştır (Tablo 6.5) (Hasegawa vd., 2009; Walczak ve Massart, 1999; Jensen vd, 2014).

$\{a_2, a_3\}$ niteliklerini birlikte içeren kural kümesinin tanımlanması esnasında, $a_3 \neq 0$ olduğunda, $P = \{a_2, a_3\}$ ve $Q = \{d\}$ koşullarının sağlanmasında aşağıdaki işlemler gerçekleştirilir.

$$P_{IND(P)}(Q) = U \{\emptyset, \{3, 6\}, \{4, 7\}\} = \{3, 4, 6, 7\}$$

$$\gamma_{\{a_2, a_3\}}(\{d\}) = \frac{|P_{\{a_2, a_3\}}(\{d\})|}{|U|} = \frac{|\{3, 4, 6, 7\}|}{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}} = \frac{4}{10}$$

Tanım 5: Kaba kümelemede değişken hassaslığı

Kaba kümelemede değişken hassaslığı, verilerin dağılım değerlerine bağlı olarak oluşturabile-

Tablo 6.4: Karar tablosu örneği (Hasegawa vd., 2009; Walczak ve Massart, 1999; Jensen vd, 2014)

U	a_1	a_2	a_3	d
x_1	4	1	5	1
x_2	3	4	1	4
x_3	4	1	5	1
x_4	4	4	5	4
x_5	1	1	6	5
x_6	1	1	4	5
x_7	3	4	1	4
x_8	1	1	6	5
x_9	4	1	5	1
x_{10}	3	4	1	4

Tablo 6.5: Örnek 2'ye ait temel veriler

$x \in U$	a_1	a_2	a_3	a_4	$\Rightarrow d$
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1
9	2	2	0	1	1
10	1	0	2	0	1

cekleri sınıfların kalite değerlerinin hesaplanmasına olanak sağlar. Şekil 6.1'de detaylı olarak gösterilmiştir.

Kaba kümelemede değişken hassaslığı Denklem (7-9) ve eşik aralık değeri Denklem (10) da verilmiştir;

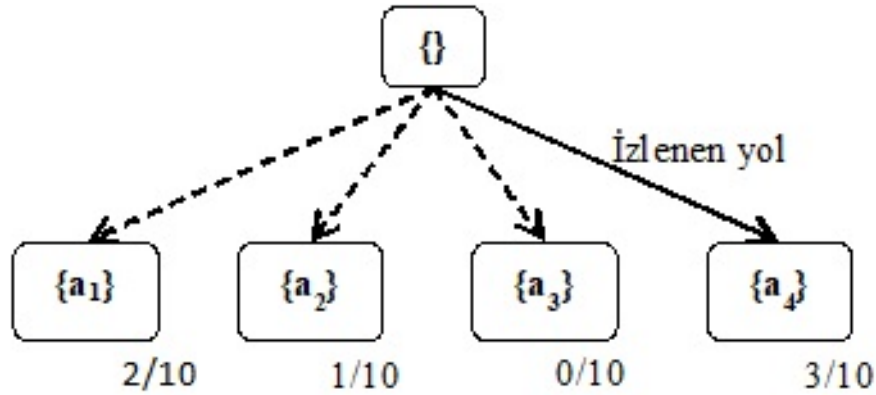
$$C(a_1, a_2) = 1 - \frac{|a_1 \cap a_2|}{|a_1|} \quad (6.7)$$

$$C(a_2, a_3) = 1 - \frac{|a_2 \cap a_3|}{|a_2|} \quad (6.8)$$

$$a_2 \subseteq_{\beta} a_3 \text{ iff } c(a_2, a_3) \leq \beta \quad (6.9)$$

$$0 \leq \beta \leq 0.5 \quad (6.10)$$

$$R_{min} = \{\{a_2, a_4\}, \{a_3, a_4\}\}$$



Şekil 6.1: Nitelik azaltımında seçilen alternatifin gösterimi

Tanım 6. Öznitelik azaltımında kullanılan QuickReduct Algoritması

Bu algoritma ilk olarak Shen ve Chouchoulas tarafından geliştirilmiş ve nitelik azaltımında etkin bir algoritmadır (Tablo 6.6). Aynı zamanda algoritmanın çalışması Şekil 6.2’de gösterilmiştir.

QuickReduct Algoritması, azaltım yapılacak olan niteliklerin bulunmasında en etkili algoritmalarından birisidir. Olası alt kümeleri üreterek, başlangıçta herhangi bir azaltım yapmadan hesaplar ve boş kümeleri öncelikli olarak elimine ederek, maksimum değeri hesaplayabileceği kümeyi seçer.

Tanım 7: Fark Edilebilirlik Matrisi Yaklaşımı

Veri ya da veriler kümesinin, evrenin belirli bir alt kümesinden nasıl fark edileceğini ve aynı zamanda belirli bir özelliğe sahip verilerin fark edilmesi sürecinde değişken özelliklerinin de dikkate alınarak verilerin özelliklerine göre toplam değerlerinin ele alındığı süreçtir. N adet veriden oluşan bir evrene ait bilgi kümesinde fark edilebilirlik matrisi c_{ij} elemanları $n \times n$ boyutlu bir matrisin x_i satır ve x_j sütun değerlerinden oluşur. Farklı özelliğe sahip x_i ve x_j verileri

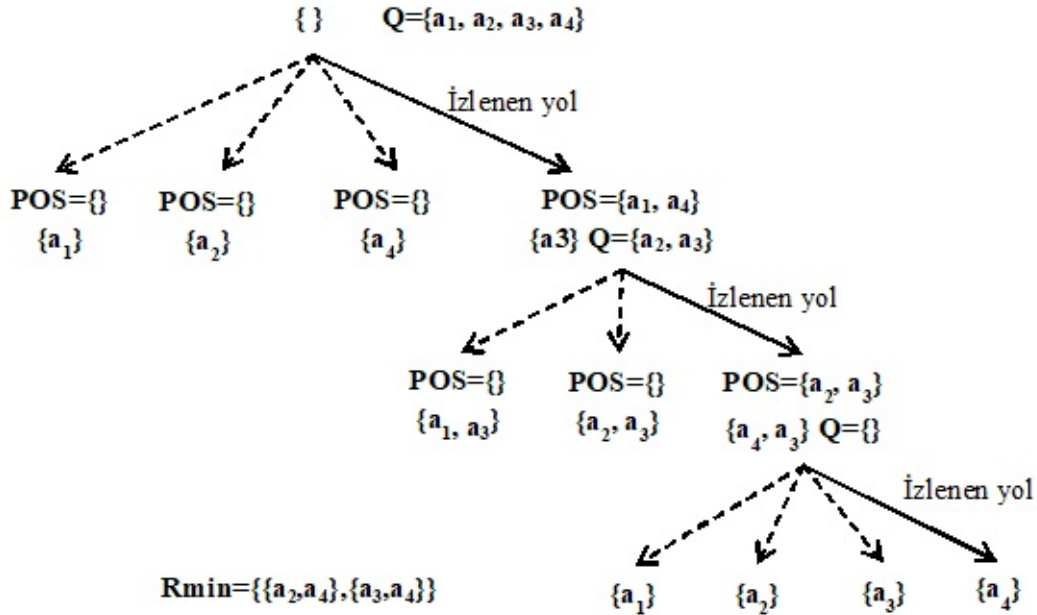
$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \quad i, j = 1, 2, \dots, n \quad (6.11)$$

Örnek 3: Fark Edilebilirlik Matrisi Yaklaşımı ile ilgili uygulama

Bu örnekte ise fark edilebilirlik matrisinin oluşturulması ve ilgili analiz işleminin nasıl gerçekleştirildiği gösterilmektedir (Hasegawa vd., 2009; Walczak ve Massart, 1999; Jensen vd., 2014; Komorowski, 2014). Örneğe ait veri kümesi Tablo 6.7’de verilmiştir. Tablo 6.8’de ise fark edilebilirlik matrisinin elde edilmiş biçimi gösterilmiştir.

Tablo 6.6: QuickReduct Algoritması (Shen ve Chouchoulas, 2002)

QUICKREDUCT Algoritması
(Hızlı Azaltım Algoritması)
QUICK REDUCT (a_i, D)
Girdi a_i – tam koşulsal nitelikler kümesi
D – karar nitelikleri kümesi
Çıktı R – Kaba nitelik kümesi
QUICK REDUCT Algoritması
(1) $R \leftarrow \{\}$
(2) while $R(D) \neq C(D)$
(3) $T \leftarrow R$
(4) Foreach $x \in (C - R)$
(5) if $RU\{x\}D > T(D)$
(6) $T \leftarrow RU\{x\}$
(7) $R \leftarrow T$
(8) until $R(D) \geq a_i(D)$
(9) return R



Şekil 6.2: Quick-Reduct algoritması ile nitelik azaltımı süreci

$$f_a(a_1, a_2, a_3, a_4) = \{a_1 \vee a_2 \vee a_3 \vee a_4\} \wedge \{a_1 \vee a_3 \vee a_4\} \wedge \{a_2 \vee a_3\} \wedge \{a_4\} \wedge \{a_1 \vee a_2 \vee a_3\} \wedge \{a_1 \vee a_2 \vee a_4\} \wedge \{a_2 \vee a_3 \vee a_4\} \wedge \{a_1 \vee a_4\}$$

Tablo 6.7: Fark edilebilirlik matrisi uygulaması için veri kümesi

$x \in U$	a_1	a_2	a_3	a_4	$\Rightarrow d$
1	1	0	2	2	0
2	0	1	1	1	2
3	2	0	0	1	1
4	1	1	0	2	2
5	1	0	2	0	1
6	2	2	0	1	1
7	2	1	1	1	2
8	0	1	1	0	1
9	2	2	0	1	1
10	1	0	2	0	1

Tablo 6.8: Fark edilebilirlik matrisinin veri kümesine uygulama biçimi

$\in U$	1	2	3	4	5	6	7	8	9	10
1										
2	a_1, a_2, a_3, a_4									
3	a_1, a_3, a_4	a_1, a_2, a_3								
4	a_1, a_4	a_1, a_3, a_4	a_1, a_2, a_4							
5	a_4	a_1, a_2, a_3, a_4	a_1, a_3, a_4	a_2, a_3, a_4						
6	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_2	a_1, a_2, a_3, a_4	a_1, a_2, a_3, a_4					
7	a_1, a_2, a_3, a_4	a_1	a_2, a_3	a_1, a_3, a_4	a_1, a_2, a_3, a_4	a_2, a_3				
8	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3, a_4	a_1, a_3, a_4	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_1, a_4			
9	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_2	a_1, a_2, a_3	a_1, a_2, a_3, a_4		a_2, a_3, a_4	a_1, a_2, a_3, a_4		
10	a_4	a_1, a_2, a_3, a_4	a_1, a_3, a_4	a_2, a_3, a_4		a_1, a_2, a_3, a_4	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1, a_2, a_3, a_4	

Kaldırılan kümeler

$$f_d(a_1, a_2, a_3, a_4) = \{a_2 \vee a_3\} \wedge \{a_4\} = \{3, 6\}, \{4, 7\}$$

Tanım 8: Dinamik Azaltımlar

Dinamik azaltım kavramı özellikle büyük ve sürekli değişkenlik arz eden verilerden kurallar elde edilmesi sürecinde kullanılan bir yaklaşımdır (Tablo 6.9). Bu yaklaşım ile birlikte büyük veriler daha kısa sürede ve etkin bir biçimde analiz edilebilmektedir. Denklem (12)'de dinamik azaltım formülü yer alırken Denklem(13)'de ise dinamik azaltımın etkililiği ölçülmüştür.

$$DR_\varepsilon(A, F) = \{a_3 \in Azaltım(A, a_4) : S_F(C) \geq \varepsilon\} \quad (6.12)$$

$$S_F(a_3) = \frac{|\{B \in F : a_3 \in Azaltım(B, a_4)\}|}{|F|} \quad (6.13)$$

$$A = \{\{a_2, a_4\}, \{a_3, a_4\}, \{a_1, a_2, a_4\}, \{a_1, a_3, a_4\}, \{a_2, a_3, a_4\}\}$$

$$R = \{\{a_2, a_4\}, \{a_2, a_3, a_4\}, \{a_1, a_2, a_4\}\}$$

Tablo 6.9: Dinamik Azaltımlar Algoritması (Yong vd. 2014)

- Dinamik Azaltım (A, ε , değer)
 A: orijinal karar tablosu
 ε : dinamik azaltım eşik değeri
 C: sütun
 deger: iterasyon sayısı
- (1) $R \leftarrow \{\}$
 - (2) $A \leftarrow \{A\}$ için tüm azaltımların hesaplanması
 - (3) *for* $j = 1 \dots \text{deger}$
 - (4) $A_i \leftarrow \{A\}$ için rassal satırları sil
 - (5) $R \leftarrow R \cup (A_i)$ için tüm azaltımların hesaplanması
 - (6) $\forall C \in A$
 - (7) *if* $S_F(C, R) \geq \varepsilon$
 - (8) *output* C

6.3 Bulanık Kaba Küme Teorisi

Bulanık küme teorisi ilk olarak 1990 yılında Dubois ve Prade tarafından önerilmiştir. Ayırt edilebilirlik ilişkilerini hesaplamak için bulanık tolerans ilişkilerini dikkate almıştır. A'nın herhangi bir B alt kümesi için

$$r_a = (x, y) = 1 - \frac{|a(x) - a(y)|}{a_{max} - a_{min}} \quad (6.14)$$

$$R_B(x, y) = T(r_a(x, y)) \quad a \in B \quad (6.15)$$

r_a bulanık benzerlik denklemini ifade ederken, T ise t-norm yapısını sembolize etmektedir. Alt yaklaşım için,

$$(R_B \downarrow x)(y) = \inf_{x \in U} I(R_B(x, y), X(x)) \quad (6.16)$$

Radzikowasoka ve Kerre tarafından önerilmiştir. I kapsayıcı yapıyı alt yaklaşım için ifade ederken, üst yaklaşım için ise T t-norm yapısını göstermektedir.

Üst yaklaşım için ise,

$$(R_B \uparrow x)(y) = \sup_{x \in U} T(R_B(x, y), X(x)) \quad (6.17)$$

t-norm (üçgen normlar) ile birlikte, Schweizer ve Sklar tarafından öne sürülmüştür. Herhangi bir $a \in [0, 1]$ aralığı için t-norm $T(a, 1) = a$ şeklinde tanımlanır ve Lukasiewicz'in t-normu dikkate alındığında, $T(a, 1) = \max(a + b - 1, 0)$ olarak ifade edilebilir (Derrac vd. 2012). Bulanık kaba kümelemede nitelik seçimi işleminde üç aşamalı bir süreç gerçekleştirilir (Huang vd. 2016). Bu süreçler ise;

- Değerlendirme: bulanık kaba alt seviye yaklaşımı uygulanır.

- Oluşturma: sezgisel algoritmalar yardımı ile alt ve üst yaklaşımların değerlendirilmesi süreci
 - Durdurma kriteri: eşik değeri temel alınarak sistemin değerlendirme sürecinde durdurma noktası da tespit edilmiş olur.

Bulanık tolerans sınıfları kullanılarak bulanık kurallar oluşturulur. Bulanık kuralların elde edilmesi sürecinde dikkate alınan algoritma ile kapsayıcı ve t-norm yapısı dikkate alınarak kuralların kapsama özellikleri ile birlikte ilgili kurallar üretilir (Tablo 6.10).

Tablo 6.10: Bulanık Kaba Kümeleme Algoritması (Derrac vd. 2013; Zeng vd. 2015, Jensen vd. 2009)

```

(1)  $B := \{\}, Kurallar := \{\}, Kaps := \{\}$ 
(2) do
(3)    $T := \{\}B,$ 
(4)   for each  $y \in (A \setminus B)$ 
(5)     for each  $y \in X \setminus Kapsama(Kaps)$ 
(6)       if  $POSB \cup \{a\} = POSA(y)$ 
(7)          $CHECK(B \cup \{a\}, RB \cup \{a\}, RA(y))$ 
(8)       if  $\gamma B \cup \{a\} > \gamma T$ 
(9)          $T := B \cup \{a\}$ 
(10)     $B := T$ 
(11) until  $\gamma B = \gamma A$ 
(12) return  $B, Kurallar$ 

```

Bulanık kaba kümelemede $r_{i,j}$ i ve j indis değerlerine a_i benzerlik derecesi, c_i i sınıfına ait değer, j indisine ait derece değeri ise alt yaklaşım ile ait olunan sınıf değeri;

$$\min \{ \max(1 - r_j, i, c_i) \quad i = 1, \dots, n \} \quad (6.18)$$

üst yaklaşım ile ait olunan sınıf değeri;

$$\max \{ \min(r_j, i, c_i) \quad i = 1, \dots, n \} \quad (6.19)$$

ile ifade edilir.

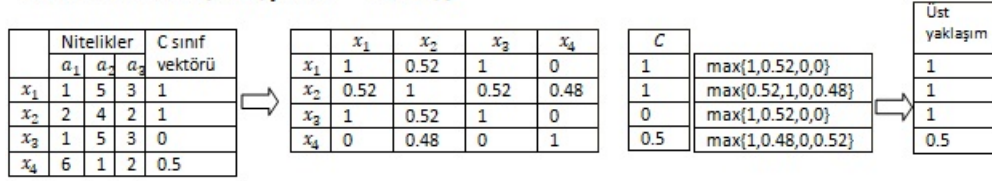
Örnek 4: Bulanık kaba kümelemeye ait alt ve üst yaklaşımları dikkate alan karar zincirinin oluşturulması (Teredesai, 2015).

Bu uygulamada 3 nitelik ve 4 veriden oluşan bir veri kümesi dikkate alınmış ve bulanık alt yaklaşım ve üst yaklaşım (Denklem 18-19) uygulanarak veriler analiz edilmiştir (Üst yaklaşım için Şekil 6.3, Alt yaklaşım için Şekil 6.4).

6.4 'Roughsets' Yazılım Paketi

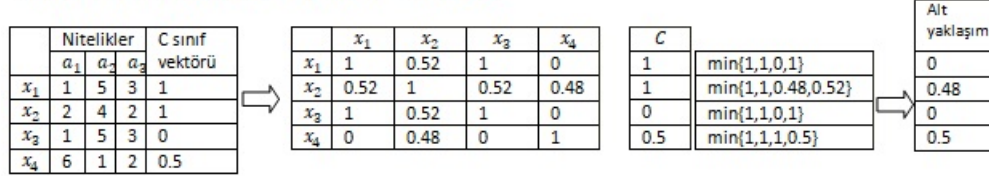
2015 yılında Rıza vd. tarafından geliştirilen R programı için yazılmış 'RoughSets' paketi kaba kümeleme ve bulanık kaba kümeleme teorisine dayanan algoritmaların uygulamalarını içeren bir yazılım paketidir. 'RoughSets' paketi ile ilgili temel bilgilere <http://cran.r-project.org/package=RoughSets> adresinden ulaşılabilir.

Üst yaklaşım: $\max\{\min(r_j, i, c_i | i = 1, \dots, n)\}$



Şekil 6.3: Üst yaklaşım ile bulanık kaba kümeleme karar sonuçları

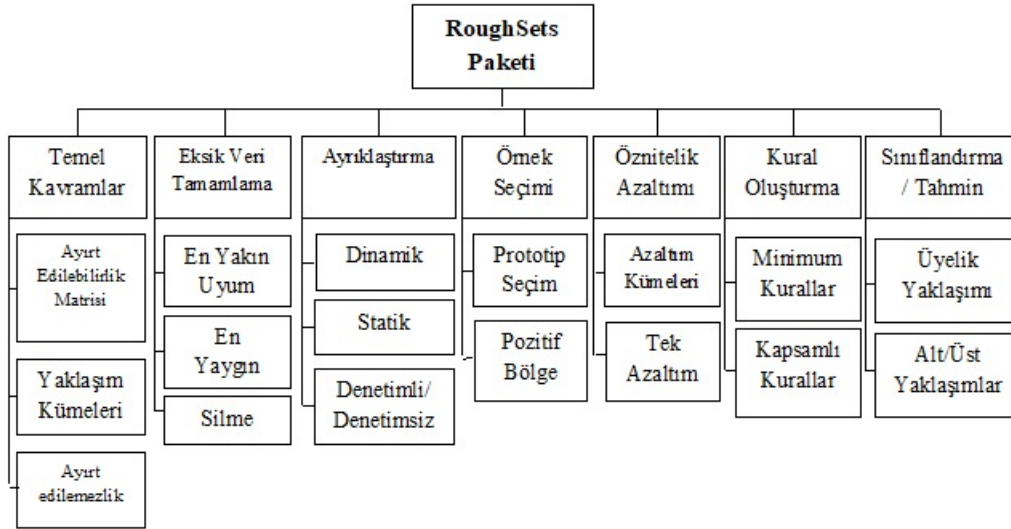
Alt yaklaşım: $\min\{\max(1 - r_j, i, c_i | i = 1, \dots, n)\}$



Şekil 6.4: Alt yaklaşım ile bulanık kaba kümeleme karar sonuçları

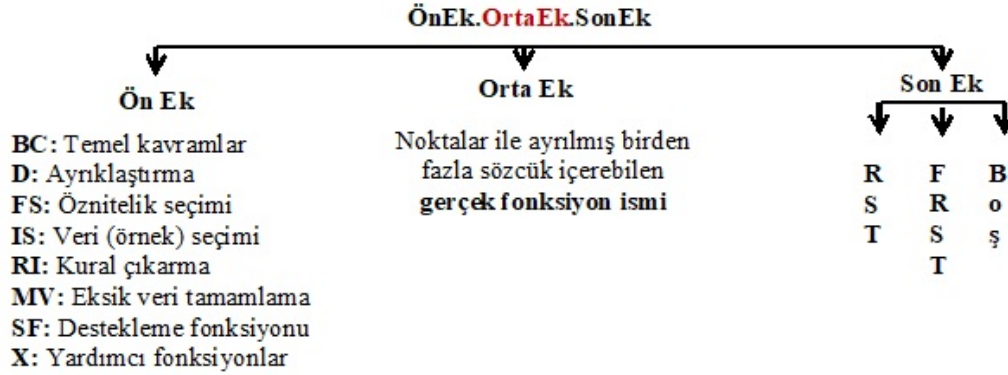
Verileri analiz etmek için Kaba Kümeleme Teorisi kullanıldığında, nesnelere / örnekler için ayırt edilemezlik ilişkisini kullandığından, ek parametrelere ihtiyaç duyulmaz. Kaba Kümeleme Teorisinin bir uzantısı olan Bulanık Kaba Kümeleme Teorisi, bulanık kümeler ve Kaba Kümeleme Teorisi ile ifade edilen belirsizlik ve ayırt edilemezlik kavramlarını birleştirir. Bu kavram, verilere ayırıştırma işlemi uygulamadan sürekli öznitelikleri analiz etmeyi sağlar.

'RoughSets' paketinin içerisindeki yöntemler işlevlerine göre, temel kavramlar, ayırıştırma, öznitelik seçimi, örnek (veri) seçimi, kural çıkarma ve tahmin/sınıflandırma olmak üzere gruplandırılabilir. Temel fonksiyonlarından bazıları Şekil 6.5'te özetlenmiştir.



Şekil 6.5: RoughSets' paketi tarafından sunulan temel fonksiyonlar (Abbas ve Burney, 2016)

Bu bölümde verilen fonksiyonların açıklamaları için RoughSets paketinin kitapçığından yararlanılmıştır (Rıza vd. 2015). Yazılım paketinde yer alan fonksiyonlar noktalar ile ayrılmış üç kısımdan oluşur (Şekil 6.6).



Şekil 6.6: 'RoughSets' paketinde bulunan fonksiyonların genel yapısı (Rıza vd. 2014b; Rıza vd. 2015)

6.4.1 Temel Kavramların Uygulanması (BC)

Bu bölümdeki fonksiyonlar kaba kümelemenin temel bileşenlerini oluşturmaktadır. Tanımlanan bu fonksiyonlar aynı zamanda uygulamada temel bileşenler olarak kullanılır. Temel kavramlarda sunulan ayırt edilemezlik ilişkileri iki nesnenin bazı özellikler tarafından ayırt edilemeyeceğini belirleyen bir ilişkidir. Alt ve üst yaklaşımlar, nesnelerin kesinlik ile sınıflandırılıp sınıflandırılmayacağını gösterir. Pozitif bölge ve bağımsızlık derecesi fonksiyonları, pozitif bölgede bulunan nesnelere ve bağımlılık derecesini belirlemek için kullanılır. Ayırt edilemezlik matrisi fonksiyonları ise her bir nesne çiftini ayırt eden özellikleri gösteren bir ayırt edilebilirlik matrisi oluşturmak için kullanılır (Rıza vd. 2015).

6.4.2 Eksik Veri Tamamlama Uygulamaları (MV)

Uygulamalarda, karar tabloları olarak sunulan girdi verileri eksik öznitelik ve karar değerlerine sahip olabilir yani karar tabloları eksik olarak belirtilebilir (Grzymala-Busse, 2005, 2008). Bir nitelik değerinin eksik olmasının iki ana nedeni vardır: değer kaybı (örn. silinmiştir) veya değer önemli olmaması durumudur. İlk durumda nitelik değeri kullanışlıdır, ancak mevcut durumda erişim imkanı yoktur. İkinci durumda ise, değer önemli değildir, bu nedenle bu gibi değerler de "dikkate alınmayan" durumlar olarak adlandırılır (Grzymala-Busse, 2005). Eksik nitelik dezavantaj durumunu ortadan kaldırmak için 'RoughSets' paketi birtakım eksik değer tamamlama fonksiyonları içermektedir (Rıza vd. 2015):

- MV.deletionCases
- MV.mostCommonValResConcept
- MV.mostCommonVal
- MV.globalClosestFit
- MV.conceptClosestFit

uygulanabileceği gibi, 'RoughSets' paketi tarafından, yöntemler için MV.missingValueCompletion() olan bir sarmalayıcı fonksiyon sağlanmıştır.

MV.missingValueCompletion(decision.table, type.method)

Burada, *decision.table* ile uygulamada kullanılacak karar tablosu, *type.method* ile kullanılacak yaklaşıma göre;

- "deletionCases"
 - "mostCommonValResConcept"
 - "mostCommonVal"
 - "globalClosestFit"
 - "conceptClosestFit"
- tanımlanır.

6.4.3 Ayırıklaştırma Uygulamaları (D)

Sayısal bir özellik, çok küçük nesnelere sınıfının ortaya çıkmasına neden olabilir. Bu da, önemi kabul edilebilir bir düzeyin altında olan kurallara yol açar. Sayısal değerleri değer aralıkları gibi sınıflara toplamak genellikle ayırıklaştırma veya yatay sıkıştırma olarak adlandırılır (Düntsche ve Gediga, 2000). Ayırıklaştırma süreci bir bilgi sisteminde, gerçek değerli nitelikleri nominal/sembolik olanlara dönüştürmek için kullanılır. Kaba kümeleme açısından, bu görev, nesnelere arasındaki fark edilebilirliği sürdürmeye çalışır. Ayırıklaştırma, kaba küme teorisinde genellikle uygulanan ilk adımlardan birisidir. 'RoughSets' paketinin ayırıklaştırma için sunduğu bazı fonksiyonlar (Rıza vd. 2015);

- D.max.discriminability.matrix.RST
- D.local.discriminability.matrix.RST
- D.global.discriminability.heuristic.RST
- D.discretize.quantiles.RST
- D.discretize.equal.intervals.RST

'RoughSets' paketi tarafından, ayırıklaştırma yöntemleri için D.discretization.RST() olan bir sarmalayıcı fonksiyon sağlanmıştır.

D.discretization.RST(decision.table, type.method,...)

type.method için;

- "global.discriminability"
- "local.discriminability"
- "unsupervised.intervals"
- "unsupervised.quantiles"

algoritmalarından biri seçilebilir.

6.4.4 Öznitelik Seçimi Uygulamaları (FS)

Öznitelik seçimi, değişken seçimi veya değişken alt kümesi seçimi, büyük veri kümesinden tam öznitelik kümesi ile aynı kaliteye sahip öznitelik alt kümesini bulma sürecidir. Başka bir deyişle, amaç önemli özellikleri seçmek ve gereksiz olanları ortadan kaldırmaktır. Çok fazla öznitelik sınıflandırma doğruluğunu etkileyebilir. Bu nedenle, veri madenciliğinde sınıflandırıcı doğruluğunu geliştirmek için öznitelik seçimi kullanılır (Singh vd., 2016). Kaba kümeleme yaklaşımı istenilen nitelikleri seçmek için, problem alanında mümkün olduğunca çok sayıda niteliği elemeye çalışır ve kabul edilebilir doğrulukla yararlı ve anlamlı sonuçlar elde eder. Az sayıda özelliğe sahip olmak, genellikle daha kolay yorumlanabilecek basit modeller oluşturulmasını sağlar (Tseng ve Huang, 2007). Kaba kümeleme ve bulanık kaba kümeleme açısından, öznitelik seçimi, süper-daraltmalar ve daraltmaların aranması anlamına gelir. 'RoughSets' paketinin öznitelik seçimi için sunduğu

fonksiyonlardan bazıları ise (Rıza vd. 2015);

- FS.quickreduct.RST
- FS.quickreduct.FRST
- FS.greedy.heuristic.superreduct.RST
- FS.nearOpt.fvprs.FRST
- FS.permutation.heuristic.reduct.RST

Fonksiyonların kullanım örneği şu şekildedir:

FS.reduct.computation(decision.table, method,...)

FS.feature.subset.computation(decision.table,method,...)

FS.all.reducts.computation(discernibilityMatrix)

6.4.5 Örnek (Veri) Seçimi Uygulamaları (IS)

Bu işlem, eğitim veri kümelerinden gürültülü, gereksiz veya tutarsız örnekleri kaldırmak, tutarlı olanları korumak için geliştirilmiştir. Sınıflandırmada pozitif katkı sağlamayan, doğruluk derecesi düşük olan örnekler kaldırılır (Rıza vd. 2014b). Kaba Kümeleme Teorisinde, sınır bölgesinde veya pozitif bölgede yer alan her bir nesne değerlendirilir. Sınır bölgesindeki ve negatif bölgede kalan nesnelere silinir. Bulanık Kaba Kümeleme Teorisindeki amaç, bulanık-kaba pozitif bölge tarafından belirlenen diğer örneklerle çakışmalara neden olan örnekleri kaldırmaktır. Bu örnekler kaldırılarak eğitim verilerinin kalitesi iyileştirilebilir ve sınıflandırıcı eğitim süresi azaltılabilir (Jensen ve Cornelis, 2010). 'RoughSets' paketinde bulunan örnek seçimi fonksiyonları ise (Rıza vd. 2015);

- IS.FRIS.FRST
- IS.FRPS.FRST

Örnek olarak IS.FRPS.FRST() fonksiyonu şu şekilde uygulanabilir:

IS.FRPS.FRST(decision.table, type.alpha = " FRPS.2 ")

Burada, *decision.table* ile eğitim veri setinden oluşturulan karar tablosu tanımlanırken, *type.alpha* ile eşiği belirlemek için, tüm örnekler üzerinde uygulanan çeşitli denklemler (FRPS.1, FRPS.2, FRPS.3, FRPS.4) den biri seçilerek belirtilir.

6.4.6 Kural Çıkarma Uygulamaları (RI)

Kural, bir veri kümesinde bulunan bilgiyi temsil eder. Kaba küme ve bulanık kaba küme kural çıkarma algoritmaları, derin bilgileri ortaya çıkarabilecek ve yeni anlayışlar sağlayabilecek IF - THEN karar kurallarını oluşturur. Bu elde edilen karar kuralları sistemin analiz edilmesi ve değerlendirilmesine yardımcı olur (Fan vd., 2009). Yaygın olarak, bu sürece makine öğrenmesinde öğrenme aşaması denir. Daha sonra, yeni veri kümeleri üzerinden tahmin edilen değerler, kurallar dikkate alınarak belirlenebilir. Paket tarafından sunulan bazı kural çıkarma fonksiyonları (Rıza vd. 2015);

- RI.indiscernibilityBasedRules.RST
- RI.hybridFS.FRST
- RI.GFRS.FRST

Fonksiyonların kullanımları şu şekildedir:

RI.indiscernibilityBasedRules.RST(decision.table, feature.set)

RI.hybridFS.FRST(decision.table, control = list())

RI.GFRS.FRST(decision.table, control= list())

6.4.7 Tahmin/Sınıflandırma Uygulamaları

Yeni bir veri kümesinin (test verileri) karar değerlerini tahmin etmek için Tahmin/Sınıflandırma fonksiyonları kullanılır. Bu fonksiyonun kullanımı genel olarak şu şekildedir (Rıza vd. 2015):

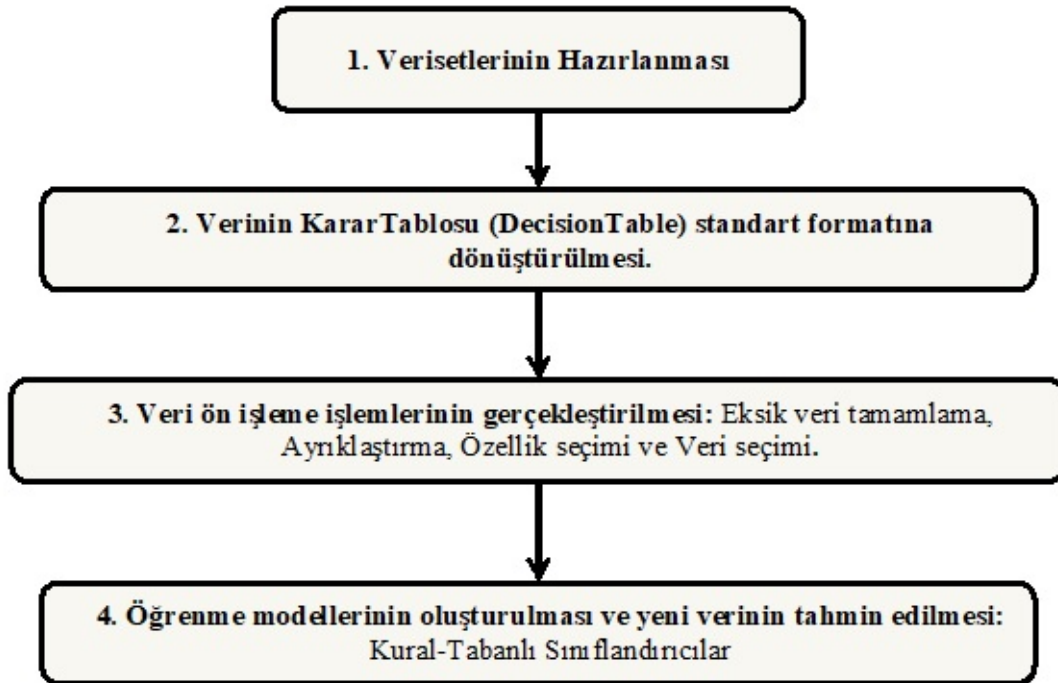
predict(object, newdata, ...)

Burada object "RuleSetRST" veya "RuleSetFRST" sınıfından kalan nesne Kaba KümeTeorisinden türeyen kural indüksiyon metodlarının uygulamaları ile üretilir. newdata ile, tahminlerin yapılacağı verileri temsil eden "DecisionTable" sınıfından kalan bir nesne ile tanımlanır.

'RoughSets' paketi veri analizi için fonksiyonlar sağlamanın yanı sıra, RoughSetData adı ile paket içerisinde gömülü veri analizinde kullanılacak karar tabloları da içermektedir. Aynı zamanda, fonksiyonların kullanımına örnek olması açısından, paket içerisinde her bir fonksiyon ile ilgili birçok demo uygulama da bulunmaktadır.

6.5 Uygulama

'RoughSets' yazılım paketi kullanılarak veri analizi gerçekleştirmek için temel olarak aşağıdaki adımlar izlenir (Şekil 6.7):



Şekil 6.7: 'RoughSets' paketi veri analizi genel adımları (Rıza vd. 2014)

6.5.1 Göğüs Kanseri Teşhisi Uygulaması

Uygulamada kullanılacak olan Breast Cancer (Wisconsin) Veri seti Uci Machine Learning Repository sitesinden alınmıştır. Veri kümesinin orijinal değerleri Wisconsin Üniversitesi Hastanelerinden Dr. William H. Wolberg tarafından elde edilmiştir. Veri setinde 16 eksik değere sahip örneklerle beraber toplam 699 adet örnek bulunmaktadır. Her bir veri kaydı 10 farklı nitelik değerine sahiptir. İlk 9

nitelik göğüs kitlesinin ince iğne aspirasyonu yardımıyla elde edilmiş resmi kullanılarak hesaplanan değerlerdir. 10. nitelik sınıf etiketidir (Takcı, 2016; Wolberg ve Mangasarian, 1990). 458 örneğin sınıfı iyi huylu kitle iken, 241 örneğin sınıfı kötü huylu göğüs kitlesidir. Veri setinde bulunan nitelikler ve değer aralıkları Tablo 6.11’de, veri setinden alınan 10 örnek ise Tablo 6.12’de verilmektedir.

Tablo 6.11: Göğüs kanseri veri setindeki öznitelikler ve değer aralıkları (Takcı, 2016)

	Öznitelik	Değer	Kısaltma
1	yığın kalınlığı	[1,10]	ykal
2	hücre boyutu homojenliği	[1,10]	hbh
3	hücre şekli homojenliği	[1,10]	hsh
4	marjinal yapışma	[1,10]	my
5	epitel hücre boyutu	[1,10]	ehb
6	çıplak çekirdekler	[1,10]	cc
7	yumuşak kromatin	[1,10]	ykroom
8	normal çekirdekçik	[1,10]	nc
9	mitoz	[1,10]	m
10	sınıf	2: iyi huylu (kansersiz olmayan) 4: kötü huylu (kansersiz)	sınıf

Tablo 6.12: Göğüs kanseri veri setinden örnek bir bölüm

Örnek No	ykal	hbh	hsh	my	ehb	cc	ykroom	nc	m	sınıf
1	5	1	1	1	2	1	3	1	1	2
2	5	4	4	5	7	10	3	2	1	2
3	3	1	1	1	2	2	3	1	1	2
4	6	8	8	1	3	4	3	7	1	2
5	4	1	1	3	2	1	3	1	1	2
6	8	10	10	8	7	10	9	7	1	4
7	1	1	1	1	2	10	3	1	1	2
8	2	1	2	1	2	1	3	1	1	2
9	2	1	1	1	2	1	1	1	5	2
10	4	2	1	1	2	1	2	1	1	2

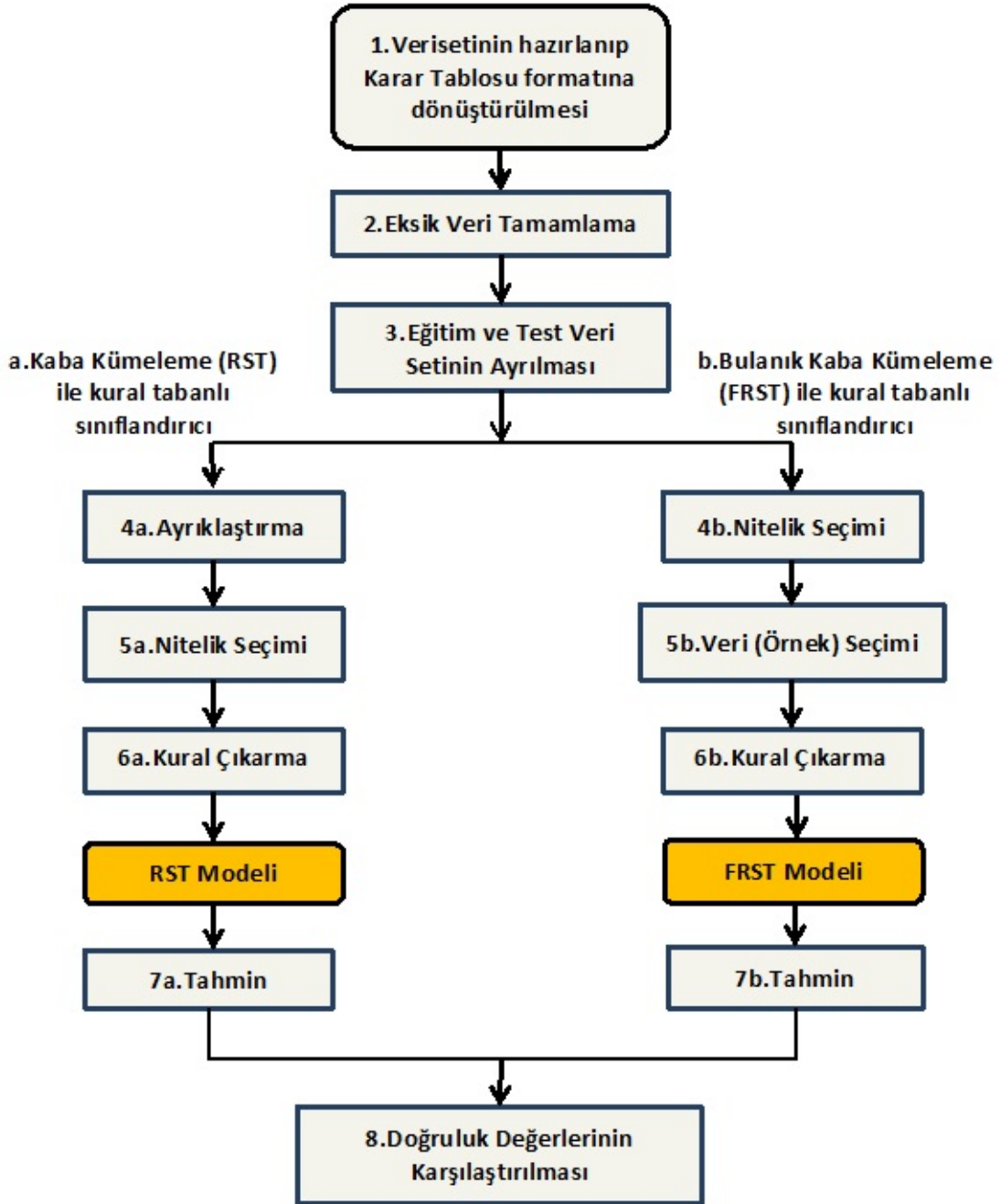
Göğüs kanseri teşhisi uygulaması iki kural oluşturma yöntemi kullanılarak gerçekleştirilecektir:

1. Kaba Kümeleme (RST) İle Kural Tabanlı Sınıflandırıcı
2. Bulanık Kaba Kümeleme (FRST) İle Kural Tabanlı Sınıflandırıcı

Her iki yöntem için uygulama adımları Şekil 6.8’de verilmektedir.

‘RoughSets’, R programlama dili için oluşturulmuş bir pakettir. Uygulamaların geliştirilebilmesi için;

1. <https://www.r-project.org> sayfasından R İstatistiksel Programlama Dilinin çalışmanın yapılacağı bilgisayarın işletim sistemine uygun yeni bir versiyonunun kurulması
2. <https://www.rstudio.com/products/RStudio/> sayfasına gidip “Desktop” ikonuna tıklanması ve ücretsiz RStudio Desktop versiyonunun indirilip kurulması

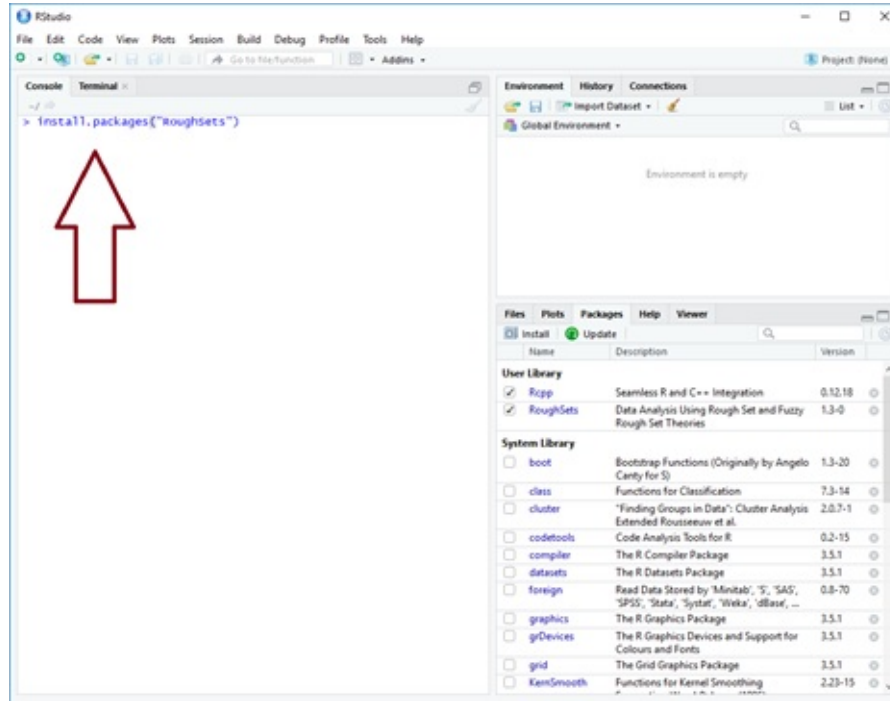


Şekil 6.8: Göğüs Kanseri Teşhisi uygulama adımları

3. Kurulumlar gerçekleştirildikten sonra, 'RoughSets' paketinin yüklenmesi gerekmektedir. `install.packages()` fonksiyonu R komut satırından istenilen paketin indirilmesi ve yüklenmesi için kullanılır. Şekil 6.9' da gösterildiği gibi, kurulan RStudio programı başlatıldığında, programın arayüzünde **Console** bölümüne;

`install.packages("RoughSets")`

kodu yazılıp çalıştırılarak 'RoughSets' paketi yüklenebilir.



Şekil 6.9: RStudio programı arayüzü

Veri analizine başlanmadan önce, yüklenen 'RoughSets' paketi `library()` fonksiyonu kullanılarak aktif hale getirilir:

`library(RoughSets)`

Adım 1. Veri Setinin Hazırlanması

Uygulamada kullanılacak veri seti CSV uzantılı Excel dosyasında düzenlenmiştir. Veri setinde eksik değerler Şekil 6.10'da gösterildiği gibi 'Not Available' (NA) ile tanımlanır.

Herhangi bir 'RoughSets' fonksiyonu çağrılmadan önce, verinin karar tablosu (DecisionTable) formatına dönüştürülmesi gerekir. DecisionTable bir karar sistemini ve bilgi sistemini ifade etmek için kullanılan bir sınıftır. `SF.read.DecisionTable()` fonksiyonu kullanılarak csv, txt, dat, Rdata, xls uzantılı bir dosyadan veri seti R ortamına aktararak karar tablosu formatına dönüştürülebilir.

`SF.read.DecisionTable(filename, decision.attr, indx.nominal,...)`

Burada; filename ile veri setinin bulunduğu dosyanın adı, decision.attr ile karar niteliğinin indeksi, indx.nominal ile veri setindeki nominal niteliklerin indeksleri tanımlanır. filename yerine file.choose() fonksiyonu kullanıldığında bir pencere açılır ve o pencereden ilgili dosya seçilerek R ortamına aktarılabilir. Uygulama için veri yükleme kod bloğu şu şekildedir:

`karar.tablosu <- SF.read.DecisionTable(file.choose(), decision.attr = 10, sep="," , indx.nominal = c(10))`

139	4	1	2	1	2	1	2	1	1	2
140	1	1	1	1	1	NA	2	1	1	2
141	3	1	1	1	2	1	1	1	1	2
142	2	1	1	1	2	1	1	1	1	2
143	9	5	5	4	4	5	4	3	3	4
144	1	1	1	1	2	5	1	1	1	2
145	2	1	1	1	2	1	2	1	1	2
146	1	1	3	1	2	NA	2	1	1	2
147	3	4	5	2	6	8	4	1	1	4
148	1	1	1	1	3	2	2	1	1	2
149	3	1	1	3	8	1	5	8	1	2

Showing 139 to 150 of 699 entries

Şekil 6.10: Eksik değerlere sahip veri setinin bir bölümü

Çalışmada kullanılan göğüs kanseri veri setinde 10. nitelik, sınıf niteliği yani karar niteliğidir. İlk 9 nitelik gerçek değerlerden oluşurken 10. nitelik, 2. ve 4. niteliklerin nominal değerlerinden oluşmaktadır. Bu yüzden, `decision.attr = 10` ve `indx.nominal = c(10)` olarak tanımlanmıştır. Bu durum aynı zamanda, `sep=","` ile verilerin Excel dosyasında virgül ile ayrılmış oldukları da belirtilir.

Karar tablosu formatına dönüştürülen veri setindeki nitelikler `colnames()` fonksiyonu ile isimlendirilebilir. Niteliklerin isimleri vektör formatında verilmelidir. Vektörler `combine` (birleştir) kelimesinin ilk harfini temsil eden `c` harfi ile oluşturulmaktadır.

```
colnames(karar.tablosu) <- c("ykal", "hbh", "hsh", "my", "ehb", "cc", "ykrom",
"nc", "m", "sınıf")
```

	ykal	hbh	hsh	my	ehb	cc	ykrom	nc	m	sınıf
139	4	1	2	1	2	1	2	1	1	2
140	1	1	1	1	1	NA	2	1	1	2
141	3	1	1	1	2	1	1	1	1	2
142	2	1	1	1	2	1	1	1	1	2
143	9	5	5	4	4	5	4	3	3	4
144	1	1	1	1	2	5	1	1	1	2
145	2	1	1	1	2	1	2	1	1	2
146	1	1	3	1	2	NA	2	1	1	2
147	3	4	5	2	6	8	4	1	1	4
148	1	1	1	1	3	2	2	1	1	2
149	3	1	1	3	8	1	5	8	1	2

Showing 139 to 150 of 699 entries

Şekil 6.11: Karar tablosunun özniteliklerinin isimlendirilmesi

Adım 2.Eksik Veri Tamamlama

Şekil 6.10 ve Şekil 6.11' de görüldüğü gibi eksik değerler (NA) içeren veri satırları bulunmaktadır. Bu uygulamada eksik değer tamamlama yöntemi olarak "globalClosestFit" seçilmiştir.

fonk.MV < – MV.missingValueCompletion(karar.tablosu, type.method = "globalClosestFit")

Fonksiyon çıktısı olarak (fonk.MV) eksik değerlerin bulunduğu indeksler ve değerleri içeren val.NA elde edildiği için, fonksiyon uygulandıktan sonra SF.applyDecTable() fonksiyonu uygulanarak yeni bir karar tablosu oluşturulur. SF.applyDecTable() fonksiyonu, belirli bir nesne / model uygulandıktan sonra yeni bir karar tablosu elde etmek için kullanılır. Fonksiyonu kullanmak için, eksik değer tamamlama, özellik seçimi, örnek seçimi veya ayırıklaştırma nesnelere olan modeller önceden hesaplanır. Genel kullanımı şu şekildedir:

SF.applyDecTable(decision.table, object, control = list())

Fonksiyon bu uygulama için şu şekilde uygulanır:

yeni.karTablo < – SF.applyDecTable(karar.tablosu, fonk.MV)

Adım 3.Eğitim Ve Test Veri Setinin Ayrılması

Uygulamada kullanılacak göğüs kanseri veri setinin %80'i modeli eğitmek için eğitim veri seti, %20'si ise eğitilen modelin doğruluğunu test etmek için test veri seti olmak üzere ikiye ayrılacaktır. Veri seti ayrılmadan önce veri satırlarının rassal olarak sıralanması sağlanacaktır.

Veri kümesinden n boyutunda rastgele bir örnek almak için sample() fonksiyonu kullanılır. nrow() fonksiyonu yeni.karTablo adındaki karar tablosunun satır sayısını vermektedir. sample() fonksiyonunun uygulanması sonucunda rassal sıralanmış satırlardan oluşan dt.Rassal karar tablosu oluşturulacaktır.

set.seed(5)

dt.Rassal < – yeni.karTablo [sample(nrow(yeni.karTablo)),]

Aşağıdaki kod ile dt.Rassal karar tablosunun veri satır sayısının %80'i hesaplanarak snr değişkenine atanmaktadır.

snr < – round(0.8 * nrow(dt.Rassal))

Aşağıdaki iki kod ile dt.Rassal karar tablosunun snr değişkeni kadar veri satırı eğitim veri seti için brst.tra karar tablosuna, snr değişkeninden sonraki veri satırları ise test veri seti olmak üzere brst.tst karar tablosuna atanır. Model eğitildikten sonra test veri setindeki sınıf etiketlerini model tahmin edeceği için, brst.tst karar tablosu oluşturulurken son nitelik olan sınıf niteliği tabloya dahil edilmez. Bu da *-ncol(dt.Rassal)* ile sağlanır.

brst.tra < – SF.asDecisionTable(dt.Rassal [1:snr,], decision.attr = 10, indx.nominal = c(10))

brst.tst < – SF.asDecisionTable(dt.Rassal [+ (snr+1):nrow(dt.Rassal), -ncol(dt.Rassal)])

Test veri seti için tahmin çalışmaları yapıldıktan sonra gerçek değerler ile karşılaştırmak üzere, test için ayrılmış veri setinin gerçek karar sınıflarının bulunduğu sütun aşağıdaki kod kullanılarak alınır.

gercek.deg < – dt.Rassal[(snr+1):nrow(dt.Rassal), ncol(dt.Rassal), drop = FALSE]

6.5.2 Kaba Kümeleme İle Kural Tabanlı Sınıflandırıcı

Adım 4a.Ayrıklaştırma

Gerçek değerlerin tüm veri madenciliği teknikleriyle işlenmesi mümkün olamamaktadır. Bilgi keşif ve analiz sistemlerinin çoğunda, sürekli değerlerin, bilgi çıkarma işleminin uygulanmasından

önce, ayrı aralıklara dönüştürülmesi gerekir (Rıza vd. 2015). Bu örnekte kuantillere dayalı Kaba Kümeleme tabanlı ayrıklaştırma yöntemi uygulanmıştır. Aralık (nOfIntervals) 4 olarak alınmıştır.

kes.deger < – **D.discretize.quantiles.RST(brst.tra, nOfIntervals = 4)**

D.discretize.quantiles.RST() fonksiyonunun çıktısı (kes.deger) kesim değerlerini içeren bir "Ayrıklaştırma" sınıfının nesnesidir. Yeni karar tablosunu oluşturmak için SF.applyDecTable() fonksiyonu kullanılmalıdır.

SF.applyDecTable() fonksiyonu eğitim ve test veri kümelerine ayrı ayrı uygulanır. Fonksiyon, eğitim veri kümesi için;

d.tra < – **SF.applyDecTable(brst.tra, kes.deger)**

Test veri kümesi için;

d.tst < – **SF.applyDecTable(brst.tst, kes.deger)**

şeklinde uygulanmıştır. Örnek olarak Şekil 6.12’de görüldüğü gibi ykal niteliğindeki veriler [-Inf, 2], (2,4], (4,6] ve (6,Inf] olmak üzere 4 aralığa ayrılmıştır. Sınıf niteliği hariç diğer nitelikteki veriler de fonksiyonun uygulanması sayesinde ayrıklaştırılmıştır.

	ykal	hbh	hsh	my	ehb	cc	ykrom	nc	m	sınıf
1	(2,4]	[-Inf,1]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
2	(4,6]	(1,4]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
3	(4,6]	[-Inf,1]	[-Inf,1]	(1,3]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
4	[-Inf,2]	[-Inf,1]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
5	[-Inf,2]	(1,4]	(1,5]	(1,3]	[-Inf,2]	[-Inf,1]	(4, Inf]	(1,4]	[-Inf,1]	2
6	(6, Inf]	(4, Inf]	(5, Inf]	(3, Inf]	(4, Inf]	(5, Inf]	(4, Inf]	[-Inf,1]	(1, Inf]	4
7	[-Inf,2]	[-Inf,1]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
8	(4,6]	[-Inf,1]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,1]	2
9	(2,4]	[-Inf,1]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	(2,3]	[-Inf,1]	[-Inf,1]	2
10	[-Inf,2]	[-Inf,1]	(1,5]	[-Inf,1]	[-Inf,2]	(1,5]	(3,4]	(1,4]	[-Inf,1]	2
11	(4,6]	(4, Inf]	(1,5]	(3, Inf]	(4, Inf]	(5, Inf]	(4, Inf]	(4, Inf]	[-Inf,1]	4

Showing 1 to 11 of 558 entries

Şekil 6.12: Kuantillere dayalı Kaba Kümeleme tabanlı ayrıklaştırma yöntemi ile ayrıklaştırılmış değerler

Adım 5a.Öznitelik Seçimi

Veri setinde herhangi bir sınıflandırma yeteneğine sahip olmayan veya gürültülü öznitelikler bulunabilmektedir. Veri analizinin kalitesinin ve hızının artırılması yollarından biri de öznitelik azaltmadır. Bu örnekte öznitelik azaltma yöntemlerinden kaba kümeleme tabanlı hızlı daraltma yöntemi (quickreduct.rst) kullanılmaktadır (Rıza vd. 2014b).

azalt1.rst < – **FS.feature.subset.computation(d.tra, method="quickreduct.rst")**

Yeni bir karar tablosu oluşturmak için SF.applyDecTable() fonksiyonu uygulanır.

fs.tra < – **SF.applyDecTable(d.tra, azalt1.rst)**

Şekil 6.13’de görüldüğü gibi öznitelik seçimi fonksiyonu uygulanmasıyla 10 nitelikten 8 nitelik elde edilmiştir.

	ykal	hbh	my	ehb	cc	ykrom	nc	sınıf
1	(2,4]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
2	(4,6]	(1,4]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
3	(4,6]	[-Inf,1]	(1,3]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
4	[-Inf,2]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
5	[-Inf,2]	(1,4]	(1,3]	[-Inf,2]	[-Inf,1]	(4, Inf]	(1,4]	2
6	(6, Inf]	(4, Inf]	(3, Inf]	(4, Inf]	(5, Inf]	(4, Inf]	[-Inf,1]	4
7	[-Inf,2]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
8	(4,6]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	[-Inf,2]	[-Inf,1]	2
9	(2,4]	[-Inf,1]	[-Inf,1]	[-Inf,2]	[-Inf,1]	(2,3]	[-Inf,1]	2
10	[-Inf,2]	[-Inf,1]	[-Inf,1]	[-Inf,2]	(1,5]	(3,4]	(1,4]	2
11	(4,6]	(4, Inf]	(3, Inf]	(4, Inf]	(5, Inf]	(4, Inf]	(4, Inf]	4

Showing 1 to 11 of 558 entries

Şekil 6.13: Kaba Kümeleme Tabanlı Hızlı Azaltım Yöntemi ile elde edilen öznitelikler

Adım 6a.Kural Çıkarma

Bu örnekte kaba kümeleme tabanlı ayırt edilemezlik sınıflarından kural oluşturma yöntemi (RI.indiscernibilityBasedRules.RST) kullanılarak verilen bir şart niteliği kümesinden IF-THEN kuralları oluşturulmuştur.

kurallar1 < – **RI.indiscernibilityBasedRules.RST(d.tra, azalt1.rst)**

Adım 7a.Test Veri Seti için Tahmin

Elde edilen kurallardan sonra, yeni nesnelerin (test veri setindeki) karar sınıflarını tahmin etmek için predict() fonksiyonu uygulanır.

predict(object, newdata, ...)

Burada, *object* RI.hybridFS.FRST() fonksiyonunun uygulanmasından sonra miras kalan sınıftır. *newdata* ise tahmin sürecine ilişkin bir dataframe veya veri matrisi (m x n) içeren bir "DecisionTable" sınıfıdır. M, örnek sayısı ve n giriş nitelikleri sayısıdır. Bu verilerin her bir öznitelik için eğitim veri setindeki öznitelikler ile aynı isimlere sahip olmaları gerekmektedir.

tahmin1 < – **predict(kurallar1, d.tst)**

6.5.3 Bulanık Kaba Kümeleme İle Kural Tabanlı Sınıflandırıcı

Adım 4b.Öznitelik Seçimi

Bulanık Kaba Kümeleme yaklaşımı ile ayırıklaştırma yapmadan veri analizi gerçekleştirilebilir. Bu uygulamada nitelik seçimi için bulanık kaba kümeleme yaklaşımı tabanlı hızlı daraltma yöntemi (quickreduct.frst) kullanılacaktır. Burada karar tablosu olarak, önceki uygulamada eksik değerleri elenmiş eğitim veri seti (brst.tra) kullanılacaktır.

azalt2 < – **FS.feature.subset.computation(brst.tra, method = "quickreduct.frst")**

Yeni karar tablolarını oluşturmak için eğitim ve test veri setine SF.applyDecTable() fonksiyonu uygulanır.

brst.tra.fs < – **SF.applyDecTable(brst.tra, azalt2)**

brst.tst.fs < – **SF.applyDecTable(brst.tst, azalt2)**

	ykal	hsh	my	ehb	cc	ykrom	nc	sınıf
141	3	1	1	2	1	1	1	2
479	5	1	1	2	1	1	1	2
640	5	1	3	2	1	1	1	2
199	1	1	1	2	1	1	1	2
73	1	3	2	2	1	7	2	2
488	10	10	10	6	10	8	1	4
366	2	1	1	2	1	2	1	2
560	5	1	1	2	1	2	1	2
662	4	1	1	2	1	3	1	2
76	1	2	1	2	2	4	2	2
189	5	4	10	5	8	9	10	4

Showing 1 to 11 of 558 entries

Şekil 6.14: Bulanık Kaba Kümeleme Tabanlı Hızlı Azaltım Yöntemi ile elde edilen öznelitlikler

Şekil 6.14'te görüldüğü gibi bulanık kaba kümeleme tabanlı hızlı daraltma yöntemi uygulanarak 10 niteliğe sahip karar tablosu 8 niteliğe düşürülmüştür.

Adım 5b. Veri (Örnek) Seçimi

Bu uygulamada, diğer kaba kümeleme uygulamasına ek olarak gürültülü, gereksiz veya tutarsız veri satırlarının eğitim veri setinden çıkarılması için bulanık kaba örnek seçimi algoritması kullanılmıştır.

```
orn.sec <- IS.FRIS.FRST(brst.tra.fs, control = list(threshold.tau = 0.2, alpha = 1))
```

Burada, *control* ile parametreler listesi tanımlanırken, *threshold.tau* bir nesnenin kaldırılıp kaldırılamayacağını belirleyen bir değeri almaktadır. Nesne, eşikten az ise veri kümesinden çıkarılabilir. *alpha* ise, bulanık benzerlik ölçüsünün derecesini belirleyen bir parametredir.

```
brst.tra.is <- SF.applyDecTable(brst.tra.fs, orn.sec)
```

Adım 6b. Kural Çıkarma

Bu uygulamada kural çıkarma işlemi için hibrit bulanık-kaba kural çıkarma ve özellik seçimine dayanan kuralların üretilmesini sağlayan *RI.hybridFS.FRST()* fonksiyonu kullanılacaktır. Fonksiyon şu şekilde uygulanır:

```
RI.hybridFS.FRST(decision.table, control = list())
```

Fonksiyonda *decision.table* karar tablosunu temsil eden bir sınıfı, *control* ise *type.aggregation*, *type.relation*, *t.implicator*'ı içeren diğer parametreler listesidir (Rıza vd. 2014).

```
kontrol <- list(type.aggregation = c("t.tnorm", "lukasiewicz"), type.relation = c("tolerance", "eq.3"), t.implicator = "kleene dienes")
```

```
kurallar2 <- RI.hybridFS.FRST(brst.tra.is, kontrol)
```

Kurallar *summary()* fonksiyonu ile Şekil 6.15'teki gibi görüntülenebilir. Toplam 172 adet kural oluşturulmuştur.

```
summary(kurallar2)
```

Adım 7b. Test Veri Seti için Tahmin

Kurallar elde edildikten sonra *predict()* fonksiyonu uygulanarak yeni değerler tahmin edilebilir.

```

The rules :
[[1]]
[1] "IF ykrom is around 10 and s10f is around 2 and ykrom is around 10 THEN s10f is 4"

[[2]]
[1] "IF hsh is around 1 and yka1 is around 1 THEN s10f is 2"

[[3]]
[1] "IF hsh is around 10 and yka1 is around 10 THEN s10f is 4"

[[4]]
[1] "IF hsh is around 10 and yka1 is around 5 THEN s10f is 4"

[[5]]
[1] "IF hsh is around 4 and yka1 is around 10 THEN s10f is 4"

[[6]]
[1] "IF hsh is around 9 and yka1 is around 8 THEN s10f is 4"

[[7]]
[1] "IF hsh is around 5 and yka1 is around 10 THEN s10f is 4"

[[8]]
[1] "IF hsh is around 10 and yka1 is around 8 THEN s10f is 4"

[[9]]
[1] "IF hsh is around 8 and yka1 is around 10 THEN s10f is 4"

[[10]]
[1] "IF hsh is around 2 and yka1 is around 10 THEN s10f is 4"

```

Şekil 6.15: Hızlı Kurallar (QuickRules) algoritması ile oluşturulmuş kuralların bir bölümü

tahmin2< – predict(kurallar2, brst.tst.fs)

Adım 8.Yöntemlerin Doğruluk Değerlerinin Karşılaştırılması

Kaba Kümeleme Tabanlı Sınıflandırma uygulaması ve Bulanık Kaba Kümeleme Tabanlı Sınıflandırma uygulamasından elde edilen tahmin sonuçları, sınıflandırma problemleri için performans ölçüm değeri olarak kullanılabilen Roc (Receiver Operating Characteric) analizindeki doğruluk (accuracy) değeri ile karşılaştırılacaktır (Dirican, 2001; Tomak, 2010). ROC eğrisi; testin sınıflandırma gücünün belirlenmesine, çeşitli testlerin performanslarının kıyaslanmasına, laboratuvar sonuçlarının kalitesinin, uygulayıcının gelişiminin izlenmesine ve farklı uygulayıcıların tanı etkinliklerinin karşılaştırılmasına imkan tanır (Dirican, 2001). Doğruluk değerinin formülü şu şekildedir (Arı vd. 2017; Dirican, 2001) Denklem(20-21):

$$Doğruluk = \frac{Doğru\ karar\ sayısı}{Tüm\ durumlar} \quad (6.20)$$

$$Doğruluk = \frac{DP + DN}{DP + YN + YP + DN} \quad (6.21)$$

Burada;

DP: Doğru Pozitif (Gerçek durum pozitif iken tahmin sonucu pozitif çıkan durum sayısı)

DN: Doğru Negatif (Gerçek durum negatif iken tahmin sonucu negatif çıkan durum sayısı)

YP: Yanlış Pozitif (Gerçek durum negatif iken tahmin sonucu pozitif çıkan durum sayısı)

YN: Yanlış Negatif (Gerçek durum pozitif iken tahmin sonucu negatif çıkan durum sayısı)

ifade etmektedir. Kaba Kümeleme Tabanlı Sınıflandırma uygulamasında tahmin1 adıyla oluşturulan karar sınıflarının doğruluk değeri şu kod bloğu ile hesaplanabilir:

dogruluk.1< – 100*sum(tahmin1== gercek.deg)/nrow(tahmin1)

Bulanık Kaba Kümeleme Tabanlı Sınıflandırma uygulamasında tahmin2 adıyla oluşturulan karar sınıflarının doğruluk değeri ise şu şekilde hesaplanabilir:

dogruluk.2 < - 100*sum(tahmin2 == gercek.deg)/nrow(tahmin2)

SONUÇ:

dogruluk.1 = % 85.61151

dogruluk.2 = % 95.68345

6.6 Sonuçlar

1982 yılında Pawlak tarafından sunulan kaba küme kavramı, belirsiz bilgiyi analiz ederek kural çıkarımı gerçekleştirmek üzere klasik küme teorisinden geliştirilmiştir. Kaba kümeleme son yıllarda büyük verilerin işlenmesi ve veri madenciliği uygulamaları ile her geçen gün daha fazla gündeme gelmektedir. Kaba kümelemede sadece veri tabanlarından bilgi madenciliği için öğrenme aktivitesini hızlandırmak ve anlaşılabilirliğini kolaylaştırmak hedeflenmemiş aynı zamanda, geliştirilen algoritma ile karar tablosunda yer alan verilerin akılcıca indirgenmesi ve istenilen koşullarda filtrelemek için kullanılması da amaçlanmıştır. Bu bölümde kaba küme fonksiyonları ve R dilinde uygulama örneklerine yer verilmiştir. Daha çok, kaba kümeleme ve bulanık kaba kümeleme için geliştirilen 'RoughSets' Paketinin yazılımı ve bu yazılımda yer alan kaba küme ve bulanık kaba küme analiz fonksiyonları kısaca tanıtılmıştır. Kaba kümeleme ve bulanık kaba kümelemede kural elde etme süreci ve nitelik azaltımları örnekler ve R yazılımında program kodları ve açıklamaları ile uygulamalı olarak gösterilmiştir. Göğüs Kanseri Teşhisi veri seti kullanılarak teorik olarak açıklanan tüm kavramlar yazılım ile birlikte uygulamalı olarak gösterilmiştir.

6.7 Kaynakça

Abbas, Z. & Burney, S.M.A. (2016). A Survey of Software Packages Used for Rough Set Analysis, *Journal of Computer and Communications* 4, 10-18.

Arı, A. & Berberler, M.E. (2017). Yapay Sinir Ağları İçin Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı, *Acta Infolica*, Sayı: 1 (2), s: 55-73.

Beynon, M. (2001). Reducts with in the variable precision roughsets model: A further investigation *European Journal of Operational Research*, 134, pp. 592-605.

Chen, J., Lin, Y., Li, J., Lin, G., Ma, Z., & Tan, A. (2016). A rough set method for the minimum vertexcover problem of graphs. *Applied Soft Computing*, 42(May), 360–367. doi:10.1016/j.asoc.2016.02.003.

CRAN: <http://cran.r-project.org/package=RoughSets>.

Derrac, J., Cornelis, C., Garcia, S., & Herrera, F. (2012). Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection'. *Information Sciences*, 186(1), 73–92. doi:10.1016/j.ins.2011.09.027.

Derrac, J., Verbiest, N., Garcia, S., Cornelis, C., Herrera, F. (2013) On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection, *Soft Comput*, 17, 223–238 DOI 10.1007/s00500-012-0888-3.

Dirican, A. (2001). Tanı testi performansının değerlendirilmesi ve kıyaslanması, *Cerrahpaşa Tıp Dergisi*. 32, 25-30.

Dubois, D. & Prade, H.(1990). Rough fuzzy sets and fuzzy roughsets, *International Journal of General Systems* 17, 191–208.

Důntsch, I. & Gediga, G. (1998). Uncertainty measures of rough set prediction Artificial Intelligence, 106 (1), pp. 109-137.

Důntsch, I. & Gediga, G. (2000). Rough set data analysis: A road to non-invasive knowledge discovery, Methodos Publishers, ISBN: 190328001X.

Fan, Y., Tseng, T., Chern, C., & Huang, C. (2009). Rule induction based on an incremental rough set, Expert Systems with Applications 36, 11439–11450.

Grzymala-Busse, J.W. (2005). Rough Set Strategies to Data with Missing Attribute Values. In: Young Lin T., Ohsuga S., Liau C.J., Hu X. (eds) Foundations and Novel Approaches in Data Mining. Studies in Computational Intelligence, 9, ISBN: 978-3-540-28315-7. 197-212.

Grzymala-Busse J.W. (2008) Three Approaches to Missing Attribute Values: A Rough Set Perspective. In: Lin T.Y., Xie Y., Wasilewska A., Liau C.J. (eds) Data Mining: Foundations and Practice. Studies in Computational Intelligence, 118., Springer, ISBN: 978-3-540-78487-6, 139-152.

Hasegawa, K., Koyama, M., Arakawa, M. & Funatsu, K. (2009). Application of datamining to quantitative structure-activity relationship using roughset theory, Chemometrics and Intelligent Laboratory Systems, 99, 66-70.

[https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+\(original\)](https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original))

<https://www.mimuw.edu.pl/son/datamining/RSDM/Intro.pdf>

<https://www.webpages.uncc.edu/ras/KDD-02/rough-sets.PPT>

Huang, C. C. Tseng (Bill), T., & Tang, C.Y. (2016). Feature extraction using rough set theory in service sector application from incremental perspective. Computers & Industrial Engineering, 91, 30-41.

Jelonek, J. , Krawiec, K. & Slowinski, R. (1995). Rough set reduction of attributes and their domains for neural Networks Computational Intelligence, 11 (2), pp. 339-347.

Jensen, R., Cornelis, C., Shen, Q. (2009) Hybrid Fuzzy-Rough Rule Induction and Feature Selection, FUZZ-IE August 20-29.

Jensen, R. & Cornelis, C. (2010). Fuzzy-Rough Instance Selection, IEEE World Congress on Computational Intelligence, 1776-1772.

Jensen, R., Tuson, A. & Shen, Q. (2014). Finding rough and fuzzy- rough set reducts with SAT, Information Sciences, 25 100-120.

Komorowski J. (2014). Learning Rule-Based Models - The Rough Set Approach. In: Brahme A. (Editor in Chief.) Comprehensive Biomedical Physics, vol. 6, 19-39. Amsterdam: Elsevier.

Last, M., Kandel, A. & Maimon, O. (2001). Information-theoretic algorithm for feature selection Pattern Recognition Letters, 22 , pp. 799-811

Li, R. & Wang, Z. (2004). Mining classification rules using rough sets and neural networks, European Journal of Operational Research, 157, 2, 439-448.

Lukasiewicz, J. (1967) ‘On 3-valued Logic’, 1920, in S. McCall (ed.), Polish Logic, Oxford University Press

Pawlak, Z. (1982). “Rough Sets”, International Journal of Computer and Information Sciences, Vol.11, 341-356.

Pawlak, Z., Grzymala-Busse, J., Slowinski, R. & Ziarko, W. (1995). Roughsets, Communications of the ACM, 38 (11), pp. 88-95

Pawlak, Z. (1991). RoughSets, Theoretical Aspects of Reasoning about Data. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Pawlak, Z., & Skowron, A. (1994). Rough Set Rudiments. The International Workshop on Rough Sets and Soft Computing, 72.

Pawlak, J., Grzymala-Busse, Z., Slowinski, R., & Ziarko, W. (1995). Roughsets. *Communications of the ACM*, 38(11), 89–95. doi:10.1145/219717.219791

Radzikowska, A. M. & Kerre, E. E. (2002). A comparative study of fuzzy roughsets. *Fuzzy Sets and Systems*, vol. 126, no. 2, pp. 137–155.

Riza, L.S., Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Slezak, D. & Benitez, J.M. (2014a). “RoughSets”: A classification tool based on rough sets and fuzzy rough sets in R. 15 Ekim 2018 tarihinde <https://www.youtube.com/watch?v=cbV2ECwRPA8> sayfasından erişildi.

Riza, L.S., Janusz, A., Bergmeir, C., Cornelis, C., Herrera, F., Slezak, D. & Benitez, J.M. (2014b). Implementing algorithms of rough set theory and fuzzy rough set theory in the R package “RoughSets”, *Information Sciences* 287, 68-89.

Riza, L.S., Janusz, A., Slezak, D., Cornelis, C., Herrera, F., Benitez, J.M., Bergmeir, C. & Stawicki, S. (2015). Package ‘RoughSets’.

Schweizer, B. & Sklar, A. (1960). Statistical metric spaces, *Pacific Journal of Mathematics*, 10, 215- 229.

Shen, Q. & Chouchoulas, A. (2002). A rough-fuzzy approach for generating classification rules, *Pattern Recognition* 35, 2425 – 2438.

Singh, A.A.G., Leavline, E.J., Priyanka, E. & Sumathi, C. (2016). Feature Selection Using Rough Set For Improving the Performance of the Supervised Learner, *International Journal of Advanced Science and Technology* 87, 1-8.

Swiniarski R.W. & Skowron, A. (2003). Rough set methods in feature selection and recognition, *Pattern Recognition Letters*, 24 , pp. 833-849.

Takcı, H. (2016). Centroid Simflayıcılar Yardımıyla Meme Kanseri Teşhisi, *Gazi Üniversitesi*, 31(2), 323-330.

Teredesai, A. (2015). Product Decisions Through Big Data. 15 Ekim 2018 tarihinde <https://www.slideshare.net/webdatascience/uw-center-for-data-science> sayfasından erişildi.

Tomak, L. & Bek, Y. (2010). İşlem Karakteristik Eğrisi Analizi Ve Eğri Altında Kalan Alanların Karşılaştırılması, *Journal of Experimental and Clinical Medicine*, Vol:27, no:2, s:58-65.

Tseng, B., & Huang, C. C. (2007). Rough set-based approach to feature selection in customer relationship management. *Omega*, 35, 365–383.

Tsumoto, S. (1998). Automated extraction of medical expert system rules from clinical databases based on rough set theory. *Information Sciences*, 112(1-4), 67–84. doi:10.1016/S0020-0255(98)10021-X

Turgay, S. & Torkul, O. (2017). *Kaba Kümeleme, Mühendislikte Yapay Zeka Uygulamaları*, Sakarya Üniversitesi Kütüphanesi Yayınevi, ISBN, 978-605-4735-98-3.

Turgay, S., Torkul, O., Turgay, T. (2018). *Rough-Set-Based Decision Model for Incomplete Information Systems*, *Encyclopedia of Information Science and Technology*, ed. Mehdi Khosrow-Pour, Fourth, IGI Publishing.

Walczak, B. & Massart, D.L. (1999). Tutorial Roughsets theory, *Chemometrics and Intelligent Laboratory Systems* 47, 1-16.

Wolberg, W H. & Mangasarian, O.L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *Proceedings of the National Academy of Sciences, U.S.A.* 87, 9193-9196.

Yong, L., Wenliang, H., Yunliang, J. & Zhiyong, Z. (2014). Quick attribute reduct algorithm for neighborhood rough set model, *Information Sciences*, 271, 65-81.

Zeng, A., Li, T., Liu, D., Zhang, J., Chen, H., A. (2015) fuzzy rough set approach for incremental feature selection on hybrid information systems, *Fuzzy Sets and Systems*, 256, 1, 39-60.

Zhao, S.Y., Eric C. C., Tsang, D. & Chen, G. (2009). The model of fuzzy variable precision roughsets, *IEEE Transactions on Fuzzy Systems* 17, 451–467.

Ziarko W. (1995). Introduction to the special issue on roughsets and knowledge discovery, *Computational Intelligence*, 11 (2), pp. 223-226.

6.8 Yazarlar Hakkında



Doç. Dr. Safiye Turgay lisans derecesini, İstanbul Teknik Üniversitesi Endüstri Mühendisliği, yüksek lisans ve doktora derecesini Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Ana Bilim dalından almıştır. Abant İzzet Baysal Üniversitesi Bilgisayar Programcılığı, Bilgisayar ve Öğretim Teknolojileri ve Eğitimi, İşletme, Sakarya Üniversitesi Yönetim Bilişim Sistemleri bölümlerinde öğretim üyeliği yapan Safiye Turgay halen Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde öğretim üyesi olarak görev yapmaktadır. Çok etmenli sistemler, bulanık mantık, karar destek sistemleri, üretim sistemleri, çok kriterli karar verme teknikleri ve kaba kümeleme konularında çok sayıda yayını bulunmaktadır.



Prof. Dr. Orhan Torkul, Yalova Üniversitesi Rektör Yardımcılığı görevinde bulunmaktadır. Prof. Dr. Orhan Torkul Lisans derecesini, Sakarya Üniversitesi Sakarya Mühendislik Fakültesi Endüstri Mühendisliği bölümünden 1982 yılında aldıktan sonra yüksek lisans derecesini Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'ndan 1987 yılında, doktora derecesini İngiltere Cranfield Teknoloji Enstitüsü'nden 1993 yılında almıştır.

Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde 1993-1998 yılları arasında Yardımcı Doçent, 1998-2003 yılları arasında Doçent, 2003 yılından itibaren ise Profesör olarak görev yapmıştır. 1993-1995 yılları arasında Mühendislik Fakültesi Dekan Yardımcılığı, 1997-2011 yılları arasında Enformatik Bölüm Başkanlığı, 2005-2011 yılları arasında Uzaktan Eğitim Araştırma ve Uygulama Merkezi Müdürlüğü, 2011-2014 tarihleri arasında Endüstri Mühendisliği Bölüm Başkanı ve 2014-2018 yılları arasında ise Mühendislik Fakültesi Dekanı olarak görev yapmıştır.

İmalat Planlama ve Kontrol, Yönetim Bilişim Sistemleri, Uzaktan Eğitim, Yapay Zeka, Dijital Dönüşüm, Üretim planlama ve kontrol, hücresel imalat sistemleri ve Endüstri 4.0 alanlarında araştırmalar ve çalışmalar yapmaktadır. Çok sayıda uluslararası hakemli dergide makale, ulusal ve uluslararası konferanslarda bildiriler ve kitap bölümleri vardır. Birçok ulusal ve uluslararası konferans ve sempozyum düzenleme komitelerinde yer almış ve davetli konuşmacı olarak çok sayıda konferans, bilimsel toplantı ve çalıştaylara katılmıştır.



Arş. Gör. Merve Şişçi, lisans eğitimini 2013 yılında Sakarya Üniversitesi Endüstri Mühendisliği'nde tamamlamıştır. 2014 yılında Dumlupınar Üniversitesi Endüstri Mühendisliği'nde Araştırma Görevlisi olarak göreve başlamış ve 2017 yılında Dumlupınar Üniversitesi Endüstri Mühendisliği Anabilim dalında "Yapay Sinir Ağları İle Hisse Senedi Kapanış Fiyatlarının Tahmini ve Portföy Optimizasyonu" tez çalışması ile yüksek lisans derecesini almıştır. 2017 yılında Sakarya Üniversitesi Endüstri Mühendisliği Anabilim dalında doktora eğitimine başlamıştır. 2017 yılından itibaren Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nde geçici görevlendirme ile Araştırma

Görevlisi olarak görevine devam etmektedir. Yapay Sinir Ağları, Derin Öğrenme, Makine Öğrenmesi, Kaba Kümeleme, Üretim Planlama ve Kontrol ve Montaj Hatları alanlarında araştırmalar yapmaktadır.



Doç. Dr. Tahsin Turgay, DPÜ, Mühendislik Fakültesi, İnşaat Mühendisliği lisans eğitimini (1997), Sakarya Üniversitesi Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Yapı Anabilim dalında yüksek lisans (2000) ve YTÜ, Fen Bilimleri Enstitüsü, İnşaat Mühendisliği Yapı Anabilim dalında doktora derecesini (2007) tamamlamıştır. ODTÜ, Mühendislik Fakültesi, İnşaat Mühendisliği doktora sonrası araştırmacı olarak çalışmıştır(2012). Doç. Dr. Tahsin Turgay, Abant İzzet Baysal Üniversitesinde, Yıldız Teknik üniversitesinde, Ortadoğu Teknik Üniversitesinde çeşitli akademik görevlerde bulunmuştur. Sakarya Üniversitesi, Sanat Tasarım ve Mimarlık Fakültesi, Mimarlık Bölümü kurucu bölüm başkanıdır. Yazarın, yapı mühendisliği, yapı mekaniği, yapılarda bünyesel modelleme, yapıların performansı, mimaride taşıyıcı sistemler, tarihi yapıların taşıyıcı sistemleri, sürdürülebilir mimarlık alanlarında çeşitli araştırmaları mevcut olup çalışmaları devam etmektedir.



7. Lojistik ve Tedarik Zinciri Yönetimi

Lojistik ve Tedarik Zinciri Yönetiminde Endüstri 4.0 Uygulamaları

Melike Kübra EKİZ¹, Murat ÇOLAK¹, Alpaslan FIĞLALI¹

¹Kocaeli Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, Kocaeli / Türkiye

Özet

Günümüzde rekabet kavramı firmalar düzeyinde olmaktan çıkarak firmaların yer aldığı tedarik zincirleri seviyesinde gerçekleşmeye başlamıştır. Tedarik zincirlerinin rekabetçiliği ise zincirin her aşamasında yer alan stok miktarları ile ölçülmektedir. Tedarik zincirinin her bileşeninin sahip olduğu stok miktarının fazla olması maliyetlerin artmasına neden olacak ve tedarik zincirinin rekabetçiliğini azaltacaktır. Bu noktada, tedarikçiden müşteriye kadar zincir üzerindeki bütün bileşenlerin arasındaki iletişimin önemi ortaya çıkmaktadır. Dördüncü endüstri devrimi olarak adlandırılan Endüstri 4.0 ile ortaya çıkan yeni teknolojiler tedarik zinciri bileşenleri arasındaki etkin iletişimin sağlanmasına ve süreçlerin etkin yönetilmesine imkan sağlamaktadır. Bu teknolojiler ile lojistiğin temel süreçleri olan taşıma ve depolamanın daha etkin ve verimli yapılması mümkün olmuştur. Diğer taraftan, lojistik ve Endüstri 4.0 kavramlarının birlikte kullanılması Lojistik 4.0 kavramını doğurmuştur. Bu kavramın ortaya çıkmasıyla birlikte lojistik ve tedarik zinciri yönetimi ile ilgili Endüstri 4.0 çalışmaları literatürde yer almaya başlamıştır. Bu çalışmada, literatürde yer alan lojistik ve tedarik zinciri yönetimi ile ilgili Endüstri 4.0 uygulamaları incelenmiştir. İncelenen çalışmalar makale türü, yıl, ülke, dergi, temel Endüstri 4.0 çalışma alanı ve sektör gibi özelliklere göre sınıflandırılmış ve analiz edilmiştir. Böylece, literatürde bu konu ile ilgili yapılmış çalışmalar hakkında araştırmacılara yol göstermek ve literatürde yer alan boşluklar hakkında bilgi vermek hedeflenmiştir.

Anahtar Kelimeler: Endüstri 4.0, lojistik yönetimi, tedarik zinciri yönetimi, lojistik 4.0, nesnelerin interneti.

7.1 Giriş

Tedarik zinciri yönetimi, tedarikçiler, üreticiler, dağıtıcılar, perakendeciler ve müşterilerden oluşan tedarik zinciri bileşenlerine ait faaliyetlerin etkin ve verimli bir şekilde planlanması, uygulanması ve kontrolü olarak tanımlanabilir. Diğer taraftan lojistik, tedarik zinciri bileşenleri arasındaki ürün, hizmet, bilgi ve paranın akışını ve depolanmasını kapsayan tedarik zinciri aşamasıdır. Diğer bir ifadeyle, doğru ürünün, doğru miktarda, doğru biçimde, doğru zamanda, doğru kaynaktan, doğru yolla ve doğru fiyata sağlanmasıdır [1]. Bu bağlamda, tedarik (inbound) lojistiği kavramı hammaddenin üreticiye ulaştırılmasını ifade etmekte iken, dağıtım (outbound) lojistiği ürün veya hizmetin müşteriye ulaştırılmasını kapsamaktadır.

Endüstri 4.0 kavramı genel manada birbirleriyle haberleşen, sensörler yardımıyla ortamı algılayabilen ve veri analizi yapabilen robotların üretimi devraldığı, bununla birlikte daha az maliyetli, daha kaliteli ve daha verimli üretimin yapıldığı bir üretim ortamını ifade etmektedir. [2]. Bu kavram ilk olarak 2011 yılında Hannover fuarında ortaya atılmış ve bunun ardından 2013 yılında Kagerman, Helbig ve Wahlster tarafından hazırlanarak Alman hükümetine sunulan “Endüstri 4.0 Stratejik Girişiminin Uygulanması İçin Öneriler” raporunda yer almıştır. Diğer taraftan, Endüstri 4.0 18. yüzyılın sonlarından günümüze kadar gelen sırasıyla mekanik, elektrik, otomasyon ve internet kavramları ile ifade edebileceğimiz dört sanayi devriminin sonuncusunu ifade etmektedir [3].

Üretim sektöründe Endüstri 4.0 ile birlikte gerçekleşen ilerlemeler lojistik ve tedarik zinciri yönetimi süreçlerini de etkilemiş ve lojistik 4.0 kavramının ortaya çıkmasına neden olmuştur. Lojistik 4.0, lojistiğin temel fonksiyonları olan taşıma ve depolama faaliyetlerinde yeni teknolojilerin kullanılmasını ve bu süreçlerin daha etkin ve daha verimli gerçekleştirilmesini ifade etmektedir. Endüstri 4.0, lojistiğin 7 doğrusu olarak ifade edilen ürün, miktar, biçim, zaman, kaynak, yöntem ve fiyat kavramlarının doğruluğunun sağlanması süreçlerinin hepsini etkilemektedir. Sanayinin gelişimi incelendiğinde, lojistik faaliyetlerin de buna paralel olarak geliştiği görülmektedir. Lojistik 1.0 makineleşmiş taşımacılığı, lojistik 2.0 otomatik depolama ve sıralama sistemlerini, lojistik 3.0 depolama yönetim sistemleri ve taşımacılık yönetim sistemleri gibi bilgisayar programlarının kullanılmasını ifade ederken lojistik 4.0 ise RFID ve otonom araçlar gibi lojistik sistemlerin geliştirilmesini ifade etmektedir [3].

Bu çalışmada, literatürde yer alan lojistik ve tedarik zinciri yönetimi ile ilgili Endüstri 4.0 çalışmaları incelenmiştir. Akademik veri tabanlarında belirlenen anahtar kelimeler yardımıyla araştırma yapılmış ve analiz edilecek makaleler tespit edilmiştir. Belirlenen makaleler makale türü, yıl, ülke, dergi adı, temel Endüstri 4.0 konusu ve sektör gibi başlıklara göre sınıflandırılmış ve analiz edilmiştir. Böylece, araştırmacılara yol göstermek ve literatürde yer alan boşluklar hakkında bilgi sunmak hedeflenmiştir.

Çalışma 5 bölümden meydana gelmektedir. İkinci bölümde lojistik ve tedarik zinciri yönetimi kavramları tanımlanmıştır. Üçüncü bölümde Endüstri 4.0 kavramı ve sanayinin gelişim evreleri hakkında bilgi verilmiştir. Dördüncü bölümde literatür analiz sonuçları sunulurken beşinci bölümde sonuç ve önerilere yer verilerek çalışma sonlandırılmıştır.

7.2 Lojistik ve Tedarik Zinciri Yönetimi

Lojistik; üretilen ürünün veya hizmetin ilk üreticiden çıkışı ile başlayarak son tüketiciye ulaşmasına kadar geçen süreçleri lojistiğin 7 unsurunu içererek sağlamaktadır. Lojistiğin 7 unsuru; doğru ürünü veya hizmeti, rekabetçi fiyat ile istenilen kalitede ve miktarda, doğru yere, doğru zamanda, doğru şekilde sağlamak olarak tanımlanabilir. Tedarik Zinciri ise ilk üreticiden son tüketiciye kadar gerçekleştirilen süreçleri kapsayan lojistiği içine alan bir kavramdır. Bir ürünün üretilmesi için

gerekli olan hammaddeden başlayarak ürünün tüketiciye ulaşması ve tüketiciye ulaşmasından sonraki süreçleri de kapsamaktadır. Lojistik süreçlerine talep ve sipariş yönetimi, satın alma, planlama, stok (envanter) yönetimi, nakliye, depo yönetimi, paketleme/ambalajlama, üretim lojistiğindeki akışlar ve iadeler örnek olarak verilebilir.

Günümüzde lojistik ve tedarik zinciri sektörü sadece yerel rekabetin yaşandığı ortamdan çıkmakta ve küresel rekabet ortamına girmektedir. Firmaların birbirleri ile rekabet anlayışları ürettikleri ürünler ve bu ürünlerin kalitesi seviyesinde kalmamakta, lojistik faaliyetleri seviyesine sıçramaktadır. Bundan dolayı lojistik ve tedarik zincirini daha etkin ve verimli kullanabilmek adına lojistik sistemdeki birçok gelişmenin yanı sıra son dönemlerde yeşil tedarik zinciri ve tersine lojistik kavramları da ortaya çıkmıştır. Yeşil tedarik zinciri; bir ürün veya hizmetin üretilmesinde kullanılacak olan hammaddenin satın alınması, üretilmesi, dağıtımının yapılması ve ömrünü tamamlamış ürünlerin geri dönüşümünün yapılması süreçlerinde çevreyi gözetmesi olarak tanımlanabilir [4]. Bir başka ifade ile yeşil tedarik zinciri, kaynakları daha etkin ve verimli kullanarak, lojistik ve tedarik zinciri süreçlerinde çevreye ve geri dönüşüme önem vererek hareket etmek olarak da tanımlanabilir. Tersine lojistik ise iade, defolu veya kullanılmış ürünler ile palet vb. taşıma araçlarının tüketim noktasından başlangıç noktasına akışının sağlanmasıdır [5]. Bu ürünlerin geri dönüşümü veya farklı satış kanallarında pazarlanması sağlanabilir. Firmalar için ürünlerin tersine lojistik maliyeti önemli bir kalem oluşturmaktadır. Bundan dolayı tersine lojistik, lojistik süreçlerinin her bir aşamasında uygulanmalı ve kontrol edilmelidir.

Hayatımızın her aşamasında yer alan lojistik ve tedarik zinciri yönetimindeki gelişmeler endüstri devrimlerine paralel bir şekilde ilerlemektedir ve bundan dolayı endüstri devrimleri gibi dört evreden oluşmaktadır. Lojistik 1.0 olarak adlandırılan ilk evrede buhar makinesinin icadı ile insan veya hayvan gücüne dayalı olarak gerçekleştirilen üretim ve taşımacılığın yerini makineler ve araçlar almıştır. Genellikle lojistik faaliyetleri kara yolu ile gerçekleştirilirken bu evrede deniz ve demir yolu da lojistik faaliyetlerinde kullanılmıştır. Üretim lojistiğinde ise insan gücü kullanılarak taşımalar ve elleçleme işlemi gerçekleştirilerek biten ürün veya hammaddeler depolanmaktadır. Lojistik 2.0 olarak adlandırılan evrede ise elektriğin hatlarda ve taşımacılıkta kullanılması ile insan gücü kullanılarak gerçekleştirilen bir çok işlem, elektrikli araçlar ile gerçekleştirilmektedir. Örneğin; insan gücü ile çalışan taşıma araçlarının yerini forklift gibi motorla çalışan araçlar almıştır. Lojistik 3.0 olarak adlandırılan evrede ise üretimde esnekliği sağlayan CNC,DNC gibi tezgahlar görülürken hatlarda insan gücünün yerine kullanılan ve otomatik üretim yapmayı sağlayan endüstriyel robotlar görülmektedir [3].

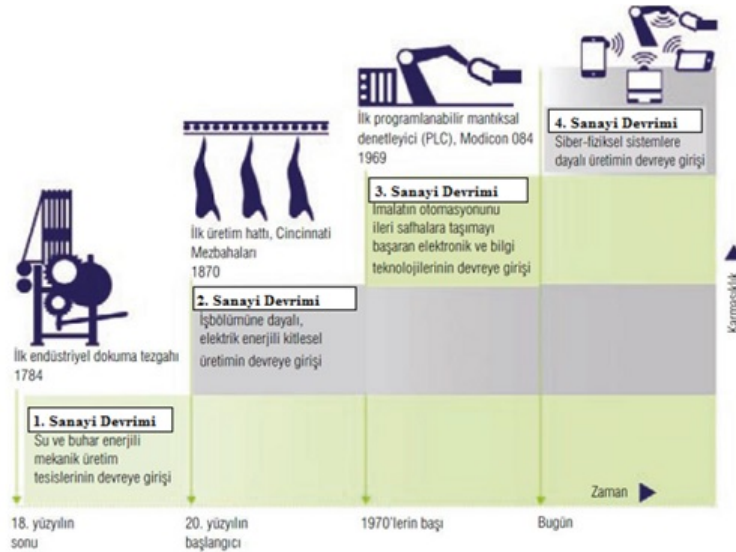
Endüstri 4.0'ın lojistik sektörüne uygulanmış hali Lojistik 4.0 olarak adlandırılmaktadır. Lojistik 4.0'ın başlıca konuları arasında Siber-Fiziksel Sistemler, Nesnelerin İnterneti, Bulut Bilişim, Simülasyon ve Büyük Veri sayılabilmektedir. Akıllı hizmetler ve ürünler ise Lojistik 4.0'ın önemli bileşenleridir [6]. Örneğin; en yaygın bileşenlerden biri olan RFID (Radyo Frekansı Tanımlama) etiketleri lojistik nesnelerin tanımlanması ve izlenmesinde kullanılmaktadır. Siber-Fiziksel Sistemler gerçek dünya ve sanal dünyayı birbirine bağlamakta, Nesnelerin İnterneti ise dinamik tedarik zincirini kontrol etmekte ve Bulut Bilişim ile depolanan lojistik süreçlerinde oluşmuş büyük veriyi analiz ederek bilgileri eş zamanlı paylaşabilmektedir. Başlıca konulardan biri olan Simülasyon yöntemi ile gidilecek rotalara ait çalışmalar yapılabilmekte ve oluşabilecek aksilikler engellenebilmektedir.

7.3 Endüstri 4.0

Sanayi devrimlerinin tarihsel gelişimi incelendiğinde Endüstri 1.0, Endüstri 2.0, Endüstri 3.0 ve Endüstri 4.0 olmak üzere dört sanayi devrimi karşımıza çıkmaktadır. Bu devrimler analiz edildiğinde

üretim anlayışında meydana gelen köklü değişimlerden isimlerini aldıkları görülmektedir. Bu tarihsel süreç ise 18. yüzyılın sonlarından günümüze kadar olan zaman dilimini ve bu zaman dilimi içerisinde üretim anlayışında ortaya çıkan değişimleri ifade etmektedir. Diğer taraftan, zaman geçtikçe üretim ortamının karmaşıklığının arttığı ve daha iyi çözümlere ihtiyaç duyulduğu görülmektedir.

1784 yılında buhar makinasının icadıyla ortaya atılan birinci sanayi devrimi (Endüstri 1.0) kavramı, su ve buhar gücü yardımıyla çalışan mekanik tezgahların insan gücünün yerine kullanılmasını ifade etmektedir. İkinci sanayi devrimi (Endüstri 2.0), 20. yüzyılın başlarından itibaren montaj hatlarında elektrik enerjisi kullanarak iş bölümüne dayalı kitlesel üretim yapılmasını ifade etmektedir. Üçüncü sanayi devrimi (Endüstri 3.0), 1970li yılların başlarında ortaya çıkmıştır. Bu devrim, üretimde otomasyona geçişi ve taşımada elektronik ve bilgi teknolojilerine dayalı sistemlerin devreye alınmasını kapsamaktadır. Sanayi devrimlerinin sonuncusu olan dördüncü sanayi devrimi (Endüstri 4.0) üretimin; nesnelerin interneti, hizmetlerin interneti ve siber fiziksel sistemler gibi kavramlara dayalı gerçekleştirilmesini içermektedir [7]. Endüstri devrimlerinin tarihsel gelişimi Şekil 7.1'de görülmektedir.



Şekil 7.1: Endüstri devrimlerinin tarihsel gelişimi [7]

Genel anlamda “internet” kelimesi ile ifade edilebilecek olan Endüstri 4.0; nesnelerin interneti, hizmetlerin interneti ve siber fiziksel sistemler olmak üzere 3 temel yapıdan oluşmaktadır. Bu üç temel yapının yanında Büyük Veri, Bulut Bilişim, Simülasyon, 3D Yazıcılar ve Artırılmış Gerçeklik gibi kavramlar Endüstri 4.0’ın diğer bileşenleri olarak karşımıza çıkmaktadır [3], [8].

Nesnelerin İnterneti: Farklı kaynaklardan veri toplanmasını, çoğaltılmasını ve organize edilmesini anlatan bu kavram fiziksel yaşamdaki nesnelerin ve bunların içinde yer alan sensörlerin kablosuz ya da kablolu olarak internet erişiminin olmasını ifade etmektedir. Diğer bir ifadeyle, nesnelerin birbirleriyle iletişim halinde olmaları ve işleri kendilerinin yönetmesidir.

Hizmetlerin İnterneti: Hizmet sektörü tarafından sunulan hizmetlerin çeşitlenmesi sonucunda bu sektörün bilgi teknolojileri ile entegre edilmesi ve faaliyetlerinin internet üzerinden yürütülmesi sağlanmış ve bu durum “hizmetlerin interneti” olarak adlandırılmıştır.

Siber Fiziksel Sistemler: Endüstri 4.0 kapsamında üretim ortamında gerçekleştirilen gözleme, koordinasyon ve kontrol gibi temel faaliyetlerin bir teknoloji tarafından yönetilmesini ve

böylece otomasyon yardımıyla müdahale gerekmeden çalışılmasını sağlayan sistemlerdir.

Büyük Veri: Klasik veri tabanı yazımlarının yakalaması, depolaması, yönetmesi ve analiz etmesi mümkün olmayan boyuttaki veri kümelerini ifade etmektedir. Teknolojinin ilerlemesiyle birlikte üretim ortamlarında da analiz edilmesi gereken büyük veriler ortaya çıkmaktadır.

Bulut Bilişim: Bilgilerin internet yardımıyla bilişim araçları arasında çevrim içi paylaşılması ve erişiminin kolay hale gelmesi olarak tanımlanmaktadır.

Simülasyon: Gerçek sistemin davranışlarının bir model yardımıyla taklit edilmesi, bu sistem üzerinde yapılacak değişikliklerin farklı senaryolar olarak analiz edilmesi ve en uygun senaryonun belirlenmesine imkan sağlayan bir karar destek sistemidir.

3D Yazıcılar: Bilgisayarda yer alan verileri 3 boyutlu elle tutulabilen cisimlere dönüştüren araçlardır. İlerleyen dönemlerde mekanik parçaların da bu yazıcılar tarafından basılması planlanmaktadır.

Artırılmış Gerçeklik: Fiziksel ortamın bilgisayar yardımıyla dinamik, canlı ve eş zamanlı olarak hissedilmesine ve karşılıklı iletişime imkan tanıyan üç boyutlu bir simülasyon modelidir.

7.4 Literatür Analizi

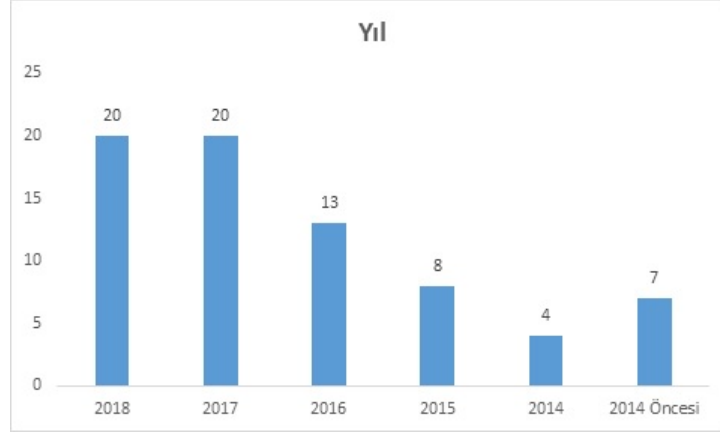
Çalışmanın bu bölümünde anahtar kelimeler yardımıyla akademik veri tabanlarında yapılan araştırmalar sonucunda ulaşılan ve Endüstri 4.0, lojistik ve tedarik zinciri kavramlarıyla ilgili olan makaleler analiz edilmiştir. Literatür araştırması “lojistik + Endüstri 4.0”, “tedarik zinciri yönetimi + Endüstri 4.0”, “lojistik + nesnelerin interneti”, “tedarik zinciri yönetimi + nesnelerin interneti”, “lojistik + büyük veri analizi”, “tedarik zinciri yönetimi + büyük veri analizi”, “lojistik + bulut bilişim”, “tedarik zinciri yönetimi + bulut bilişim”, “lojistik + siber fiziksel sistemler”, “tedarik zinciri yönetimi + siber fiziksel sistemler”, “lojistik + simülasyon”, “tedarik zinciri yönetimi + simülasyon”, “akıllı lojistik” anahtar kelimeleri kullanılarak gerçekleştirilmiştir. Yapılan elemeler sonucunda elde edilen 72 makale yıl, makale türü, ülke, dergi, temel Endüstri 4.0 konusu ve sektör gibi özelliklere göre analiz edilmiştir. Analiz sonuçları şekil ve tablolar yardımıyla aşağıda sunulmuştur:

Tablo 7.1: Yayınların yıllara göre sayıları ve kaynakları

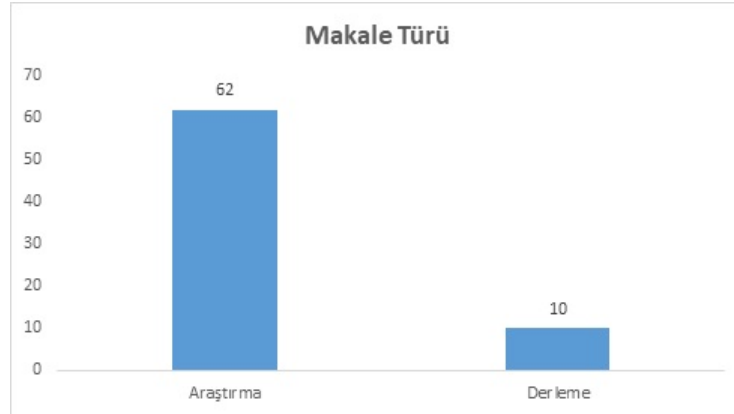
Yıllar	Yayın Sayısı	Kaynaklar
2018	20	[9-28]
2017	20	[29-48]
2016	13	[49-61]
2015	8	[62-69]
2014	4	[70-73]
2014 Öncesi	7	[74-80]

Tablo 7.1 ve Şekil 7.2 analiz edildiğinde bu alandaki yayın sayılarının yıllara göre giderek arttığı görülmektedir. Yayın sayılarının giderek artmasından hareketle konu ile ilgili endüstriyel uygulama alanının genişlemekte olduğu ve bu alanın çalışmaya değer olduğunu söylemek mümkündür.

Şekil 7.3'te makale türlerine göre yayın sayıları sunulmaktadır. Şekilde görüldüğü gibi, bu alanda araştırma makaleleri literatürde daha çok yer almaktadır. Diğer taraftan, araştırmacılara bu alanda durum analizi sunmak için hazırlanan derleme makaleler de literatürde yer almaktadır.



Şekil 7.2: Yayınların yıllara göre sayılarının grafiksel gösterimi

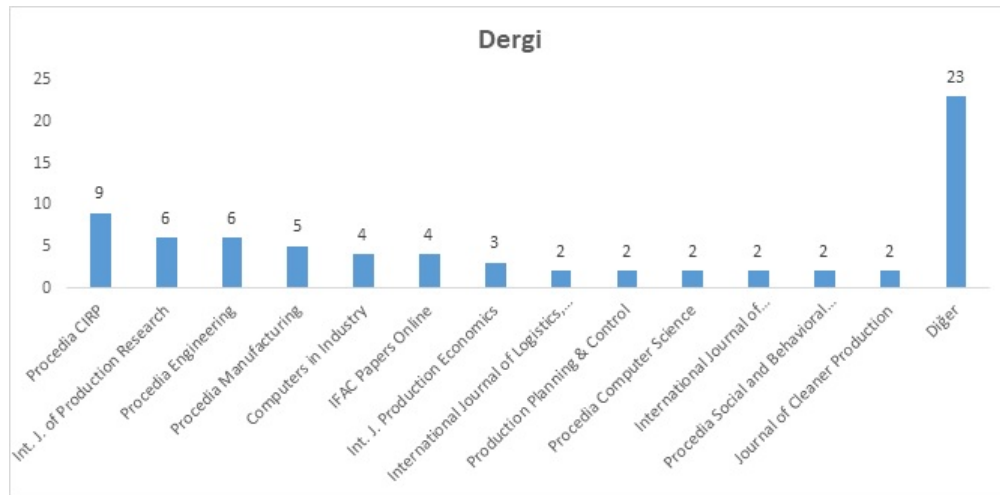


Şekil 7.3: Yayınların türlerine göre sayılarının grafiksel gösterimi

Tablo 7.2: Yayınların ülkelere göre sayıları

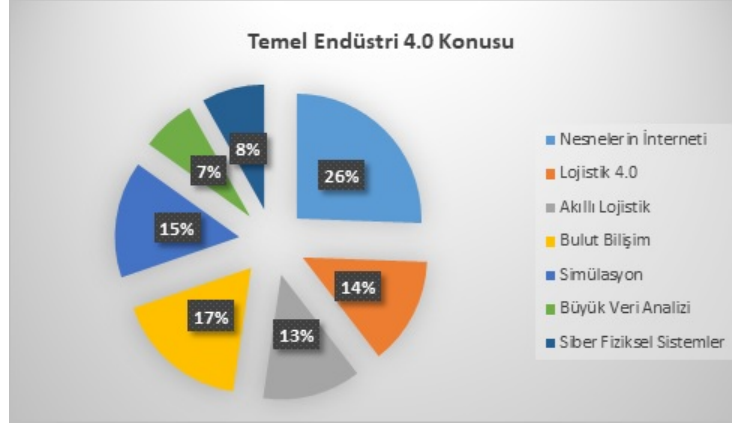
Ülkeler	Yayın Sayısı
Çin	9
Birleşik Krallık	6
Almanya	6
Hollanda	5
Brezilya	4
Hindistan	4
USA	4
Rusya	3
Avustralya	3
Polonya	3
Portekiz	3
İtalya	2
Türkiye	2
Fransa	2
Slovakya	2
Diğer	14

Yayınların ülkelere göre sayıları Tablo 7.2’de sunulmaktadır. Tablo incelendiğinde çok farklı ülkelerde bu alanda yayınların yapıldığı görülmektedir. Bu alanda en çok yayının yapıldığı ülke Çin iken, onu sırasıyla Birleşik Krallık ve Almanya takip etmektedir.



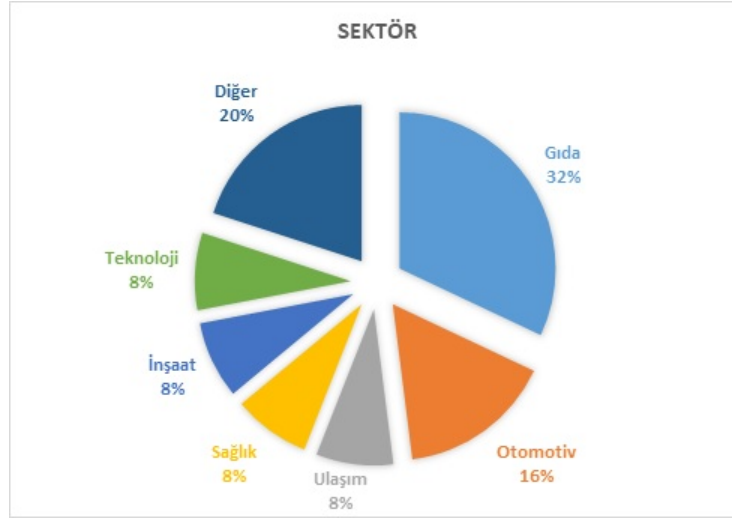
Şekil 7.4: Yayınların dergilere göre sayılarının grafiksel gösterim

Bu alanda yapılan yayınların dergilere göre sayıları Şekil 7.4'te verilmektedir. En çok yayının Procedia CIRP dergisinde yer aldığı görülmektedir. Bu dergiyi sırasıyla International Journal of Production Research ve Procedia Engineering dergileri takip etmektedir. Diğer taraftan, bu alanda konferans bildirilerinin de makaleye dönüştüğü görülmektedir.



Şekil 7.5: Yayınların temel Endüstri 4.0 konularına göre analizi

Yayınların temel Endüstri 4.0 başlıklarına göre analizi Şekil 7.5'te sunulmaktadır. Şekilde görüldüğü gibi bu alanda en çok nesnelerin interneti ve lojistik 4.0 kavramları uygulama alanı bulmuştur.



Şekil 7.6: Yayınların sektörlere göre analizi

Yayınların sektörlere göre analiz sonuçları Şekil 7.6'da verilmektedir. Şekilde görüldüğü gibi literatürde en çok gıda ve otomotiv sektörleri ile ilgili çalışmalar yapılmıştır. Ulaşım, sağlık, inşaat ve teknoloji sektörlerinin oranlarının birbirine eşit olduğu görülmektedir.

Bütün bu analizlere ek olarak, lojistik ve Endüstri 4.0 kavramları ile ilgili literatürde yer alan çalışmalarda çok kriterli karar verme yöntemleri, yapısal eşitlik modelleme, faktör analizi matematiksel modelleme, bulanık kümeler, metasezgisel yöntemler, yapay sinir ağları, simülasyon

ve SWOT analizi yöntemleri de çeşitli amaçlarla kullanılmaktadır. Ayrıca yayınların uygulama alanları incelendiğinde genel anlamda bütün lojistik ve tedarik zinciri yönetimi süreçleri (üretim planlama, talep yönetimi, satın alma, taşıma, depolama, tedarik zinciri optimizasyonu) ile ilgili Endüstri 4.0 uygulamalarının gerçekleştirildiği görülmektedir. Çalışmalar analiz edildiğinde taşıma ve depolama süreçlerinde otonom araçlar, RFID, ERP ve GPS/GPRS teknolojilerinin kullanıldığı da görülmektedir.

7.5 Sonuç ve Öneriler

Endüstri 4.0'ı oluşturan başlıca konular (Siber-Fiziksel Sistemler, Nesnelerin İnterneti, Bulut Bilişim, Simülasyon, Büyük Veri) tüm üretim sistemlerinde ve bu üretim sistemlerine ait unsurlarda köklü bir değişimin gerçekleşmesine sebep olmuştur. Üretim sistemleri ile direkt bağlantısı olan tedarik zinciri ve lojistik kavramlarında köklü değişimler gerçekleşmiş ve lojistik 4.0 evresine geçiş başlamıştır. Gerçekleştirdiğimiz çalışmada farklı veri tabanlarında belirlenen çeşitli anahtar kelimeler ile “Lojistik ve Tedarik Zinciri Yönetiminde Endüstri 4.0 Uygulamaları” konusunda tarama gerçekleştirilmiş ve elde edilen çalışmalar sistematik şekilde sınıflandırılmıştır. Analiz edilen 72 makale; türüne (araştırma, derleme), yıllara, ülkelere, dergilere, temel Endüstri 4.0 başlığına (Nesnelerin İnterneti, Lojistik 4.0, Akıllı Lojistik, Bulut Bilişim, Simülasyon, Büyük Veri Analizi) ve sektöre (gıda, otomotiv, ulaşım, sağlık, inşaat, teknoloji ve diğer) göre sınıflandırılmıştır.

Türüne göre ayrılan makaleler 62 araştırma ve 10 derleme makalesi olarak belirlenmiştir. Yıllara göre analizlerin sonucunda lojistik ve tedarik zinciri kavramları ile ilgili Endüstri 4.0 uygulamalarının giderek arttığı görülmüştür. Ülkelere göre analizlerde diğer dünya ülkelerine göre ülkemizdeki çalışmaların daha az olduğu belirlenmiştir. Dergi bazında bakıldığında en fazla yoğunluğun görüldüğü grup konferans makaleleridir. Lojistik 4.0 ve Akıllı Lojistik grubunda daha çok kavramsal çalışmalar yer alırken diğer guruplarda uygulama çalışmaları da yer almaktadır. Lojistiğin yeni konuları olan Yeşil Tedarik Zinciri ve Tersine Lojistik konuları ile ilgili uygulamaların az sayıda olduğu belirlenmiştir.

Son dönemlerde ülkelerin ve şirketlerin çevreye olan duyarlılıklarının ve karbon ayak izlerini azaltma çabalarının artması nedeniyle yeşil lojistik konusundaki çalışmalar artmaktadır. Endüstri 4.0 uygulamalarının yeşil lojistik konusundaki çabalara katkı sağlaması beklenmektedir. Tersine lojistik faaliyetleri ile iade ve hasarlı ürünlerin çıkış noktasına dönüşleri sağlandığından birim lojistik maliyetleri ileri yönlü hareketleri ile karşılaştırıldığında oransal olarak çok yüksek düzeylere ulaşabilmektedir. Endüstri 4.0 uygulamalarının söz konusu yüksek maliyetlerin düşürülmesine de önemli katkı sağlaması beklenmektedir.

7.6 Kaynakça

- [1] Öztemel, E., & Gürsev, S. (2018). Türkiye’de Lojistik Yönetiminde Endüstri 4.0 Etkileri ve Yatırım İmkanlarına Bakış Üzerine Anket Uygulaması, Marmara Fen Bilimleri Dergisi, 2, 157-168.
- [2] Yıldız, A. (2018). Endüstri 4.0 ve akıllı fabrikalar, Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22(2), 546-556.
- [3] Şekkeli, Z. H., & Bakan, İ. (2018). Endüstri 4.0’ın Etkisiyle Lojistik 4.0, Journal of Life Economics, 5(2), 17-36.
- [4] Atrek, B., & Özdağoğlu, A. (2014). Yeşil Tedarik Zinciri Uygulamaları: Alüminyum Doğrama Sektörü İzmir Örneği, Anadolu Üniversitesi Sosyal Bilimler Dergisi, 13-25.

- [5] Özdemir Akyıldırım, Ö., & Abdılđaev, M. (2016). Türkiye’de Tersine Lojistik Uygulamaları Üzerine Karşılaştırmalı Bir Sınıflandırma Çalışması, *Ege Akademik Bakış*, 16(2), 257-272.
- [6] Göçmen, E., & Erol, R. (2018). The Transition To Industry 4.0 In One Of The Turkish Logistics Company, *International Journal Of 3d Printing Technologies and Digital Industry*, 2, 76-85.
- [7] Kılıç, S. & Alkan, R. M. (2018). Dördüncü Sanayi Devrimi 4.0: Dünya ve Türkiye Değerlendirmeleri, *Girişimcilik İnovasyon ve Pazarlama Araştırmaları Dergisi*, 2(3), 29-49.
- [8] Soylu, A. (2018). Endüstri 4.0 ve Girişimcilikte Yeni Yaklaşımlar, *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 32, 43-57.
- [9] Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., & Lin, Y. (2018). Big data analytics in supply chain management: A state-of-the-art literature review, *Computers and Operations Research*, 98, 254-264.
- [10] Dev, N. K., Shankar, R., Gupta, R., & Dong, J. (2018). Multi-criteria evaluation of real-time key performance indicators of supply chain with consideration of big data architecture, *Computers & Industrial Engineering*, <https://doi.org/10.1016/j.cie.2018.04.012>.
- [11] Raman, S., Patwa, N., Niranjana, I., Ranjan, U., Moorthy, K., & Mehta, A. (2018). Impact of big data on supply chain management, *International Journal of Logistics Research and Applications*, DOI: 10.1080/13675567.2018.1459523.
- [12] Vazquez-Martinez, G. A., Gonzalez-Compean, J. L., Sosa-Sosa, V. J., Morales-Sandoval, & M., Perez, J. C. (2018). CloudChain: A novel distribution model for digital products based on supply chain principles, *International Journal of Information Management*, 39, 90-103.
- [13] Kochan, C. G., Nowicki, D. R., Sauser, B., & Randall, W. S. (2018). Impact of cloud-based information sharing on hospital supply chain performance: A system dynamics framework, *International Journal of Production Economics*, 195, 168-185.
- [14] Zhou, L., Zhang, L., & Ren, L. (2018). Modelling and simulation of logistics service selection in cloud manufacturing, *Procedia CIRP*, 72, 916-921.
- [15] Pires, M. C., Frazzon, E. M., Danielli, A. M. C., Kück, M., & Freitag, M. (2018). Towards a simulation-based optimization approach to integrate supply chain planning and control, *Procedia CIRP*, 72, 520-525.
- [16] Ding, B. (2018). Pharma Industry 4.0: literature review and research opportunities in sustainable pharmaceutical supply chains, *Process Safety and Environmental Protection*, 119, 115-130.
- [17] Abdel-Basset, M., Manogaran, G., & Mohamed, M. (2018). Internet of Things (IoT) and its impact on supply chain: A framework for building smart, secure and efficient systems, *Future Generation Computer Systems*, 86, 614–628.
- [18] Bake, J. S., Pessoa, M. V., & Becker, J. M. (2018). Service Chain Logistics Management for Increasing Equipment Uptime, *Procedia CIRP*, 73, 210–215.
- [19] Büyüközkan, G., & Göçer, F. (2018). Digital Supply Chain: Literature review and a proposed framework for future research, *Computers in Industry*, 97, 157–177.
- [20] Dallasega, P., Rauch, E., & Linder, C. (2018). Industry 4.0 as an enabler of proximity for construction supply chains: A systematic literature review, *Computers in Industry*, 99, 205–225.
- [21] Gruzauskas, V., Baskutis, S., & Navickas, V. (2018). Minimizing the trade-off between sustainability and cost effective performance by using autonomous vehicles, *Journal of Cleaner Production*, 184, 709-717.
- [22] Ivanov, D., Dolgui, A., & Sokolov, B. (2018). The impact of digital technology and Industry 4.0 on the ripple effect and supply chain risk analytics, *International Journal of Production Research*, 1-18.

- [23] Kayikci, Y. (2018). Sustainability impact of digitization in logistics, *Procedia Manufacturing*, 21, 782-789.
- [24] Ku, A. Y. (2018). Anticipating critical materials implications from the Internet of Things (IOT): Potential stress on future supply chains from emerging data storage technologies, *Sustainable Materials and Technologies*, 15, 27–32.
- [25] Liu, S., Zhang, G., & Wang, L. (2018). IoT-enabled Dynamic Optimisation for Sustainable Reverse Logistics, *Procedia CIRP*, 69, 662 – 667.
- [26] Luthra, S., & Mangla, S. K. (2018). Evaluating challenges to Industry 4.0 initiatives for supply chainsustainability in emerging economies, *Process Safety and Environmental Protection*, 117, 168–179.
- [27] Mehami, J., Nawari, M., & Zhang, R. Y. (2018). Smart automated guided vehicles for manufacturing in the context of Industry 4.0, *Procedia Manufacturing*, 26, 1077–1086.
- [28] Vass, T. d., Shee, H., & Miah, S. (2018). The effect of “Internet of Things” on supply chain integration and performance: An organisational capability perspective, *Australasian Journal of Information Systems*, 22, 1-29.
- [29] Barbosa, M. W., de la Calle Vivente, A., Ladeira, M. B., & de Oliveira, M. P. V. (2017). Managing supply chain resources with Big Data Analytics: a systematic review, *International Journal of Logistics Research and Applications*, DOI: 10.1080/13675567.2017.1369501.
- [30] Lamba, K., & Singh, S. P. (2017). Big data in operations and supply chain management: current trends and future perspectives, *Production Planning & Control*, 28(11-12), 877-890.
- [31] Musa, Z., & Vidyasankar, K. (2017). A Fog Computing Framework for Blackberry Supply Chain Management, *Procedia Computer Science*, 113, 178-185.
- [32] Yu, Y., Cao, R. Q., & Schniederjans, D. (2017). Cloud computing and its impact on service level: a multi-agent simulation model, *International Journal of Production Research*, 55(15), 4341-4353.
- [33] Subramanian, N., & Abdulrahman, M. D. (2017). Logistics and cloud computing service providers’ cooperation: a resilience perspective, *Production Planning & Control*, 28(11-12), 919-928.
- [34] Arsovski, S., Arsovski, Z., Stefanovic, M., Tadic, D., & Aleksic, A. (2017). Organisational resilience in a cloud-based enterprise in a supply chain: a challenge for innovative SMEs, *International Journal of Computer Integrated Manufacturing*, 30(4-5), 409-419.
- [35] Kovalsky, M., & Micieta, B. (2017). Support planning and optimization of intelligent logistics systems, *Procedia Engineering*, 192, 451-456.
- [36] Amirkolaii, K. N., Baboli, A., Shahzad, M. K., & Tonadre, R. (2017). Demand Forecasting for Irregular Demands in Business Aircraft Spare Parts Supply Chains by using Artificial Intelligence (AI), *IFAC PapersOnLine*, 50(1), 15221-15226.
- [37] Silva, N., Ferreira, L. M. D. F., Silva, C., Magalhaes, V., & Neto, P. (2017). Improving Supply Chain Visibility With Artificial Neural Networks, *Procedia Manufacturing*, 11, 2083-2090.
- [38] Accorsi, R., Bortolini, M., & Baruffaldi, G. (2017). Internet-of-Things paradigm in food supply chains control and management, *Procedia Manufacturing*, 11, 889 – 895.
- [39] Barreto, L., Amaral, A., & Pereira, T. (2017). Industry 4.0 implications in logistics: an overview, *Procedia Manufacturing*, 13, 1245-1252.
- [40] Ben-Daya, M., Hassini, E., & Bahroun, Z. (2017). Internet of things and supply chain management: a literature review, *International Journal of Production Research*, 1-24.
- [41] Emiliano, W., Telhada, J., & Carvalho, M. d. (2017). Home helath care logistics planning: areview and framework, *Procedia Manufacturing*, 13, 948–955.

- [42] Furmann, R., Furmannova, B., & Wiecek, D. (2017). Interactive design of reconfigurable logistics systems, *Procedia Engineering*, 192, 207 – 212.
- [43] Hofmann, E., & Rüsçh, M. (2017). Industry 4.0 and the current status as well as future prospects on logistics, *Computers in Industry*, 89, 23–34.
- [44] Lee, C., Lv, Y., Ng, K., Ho, W., & Choy, K. (2017). Design and application of Internet of things-based warehouse management system for smart logistics, *International Journal of Production Research*, 2753-2768.
- [45] Tjahjono, B., Esplugues, C., Ares, E., & Pelaez, G. (2017). What does Industry 4.0 mean to Supply Chain?, *Procedia Manufacturing*, 13, 1175–1182.
- [46] Trappey, A. J., Trappey, C. V., Fan, C.-Y., Hsu, A. P., Li, X.-K., & Lee, I. J. (2017). IoT patent roadmap for smart logistic service provision in the context of Industry 4.0, *Journal of the Chinese Institute of Engineers*, 593-602 .
- [47] Witkowski, K. (2017). Internet of Things, Big Data, Industry 4.0– Innovative Solutions in Logistics and Supply Chains Management, *Procedia Engineering*, 182, 763 – 769.
- [48] Yan, B., Wang, X., & Shi, P. (2017). Risk assessment and control of agricultural supply chains under Internet of Things, *Agricultural Economics Research, Policy and Practice in Southern Africa*, 56, 1-12.
- [49] Bruque-Camara, S., Moyano-Fuentes, J., & Maqueira-Marin, J. M. (2016). Supply chain integration through community cloud: Effects on operational performance, *Journal of Purchasing & Supply Management*, 22, 141-153.
- [50] Xing, K., Qian, W., & Zaman, A. U. (2016). Development of a cloud-based platform for footprint assessment in green supply chain management, *Journal of Cleaner Production*, 139, 191-203.
- [51] Ivanov, D., Sokolov, B., & Ivanova, M. (2016). Schedule Coordination in Cyber-Physical Supply Networks Industry 4.0, *IFAC-Papers Online*, 49(12), 839-844.
- [52] Schuhmacher, J., & Hummel, V. (2016). Decentralized control of logistic processes in cyber-physical production systems at the example of ESB Logistics Learning Factory, *Procedia CIRP*, 54, 19-24.
- [53] Blöchl, S. J., & Schneider, M. (2016). Simulation Game for Intelligent Production Logistics – The PuLL® Learning Factory, *Procedia CIRP*, 54, 130-135.
- [54] McFarlane, D., Giannakis, V., & Lu, W. (2016). Intelligent logistics: Involving the customer, *Computers in Industry*, 81, 105-115.
- [55] Wiecek, P. (2016). Intelligent Approach to Inventory Control in Logistics under Uncertainty Conditions, *Transportation Research Procedia*, 18, 164-171.
- [56] Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., & Ivanova, M. (2016). A dynamic model and an algorithm for shortterm supply chain scheduling in the smart factory industry 4.0, *International Journal of Production Research*, 386-402.
- [57] Karakostas, B., & Bessis, N. (2016). The application of Internet of Things Brokers in logistics, *International Journal of Advanced Logistics*, 5(2), 70-75.
- [58] Knoll, D., Prüglmeier, M., & Reinhart, G. (2016). Predicting Future Inbound Logistics Processes using Machine Learning, *Procedia CIRP*, 52, 145 – 150.
- [59] Qu, T., Thüerer, M., Wang, J., Wang, Z., Fu, H., Li, C., & Huang, G. Q. (2016). System dynamics analysis for an Internet-of-Things-enabled production logistics system, *International Journal of Production Research*, 2622-2649.
- [60] Shi, Q., Ding, X., Zuo, J., & Zillante, G. (2016). Mobile Internet based construction supply chain management: A critical review, *Automation in Construction*, 72, 143–154.

- [61] Verdouw, C., Wolfert, J., Beulens, A., & Rialland, A. (2016). Virtualization of food supply chains with the internet of things, *Journal of Food Engineering*, 176, 128-136.
- [62] Singh, A., Mishra, N., Ali, S. I., Shukla, N., & Shankar, R. (2015). Cloud computing technology: Reducing carbon footprint in beef supply chain, *Int. J. Production Economics*, 164, 462-471.
- [63] Becker, T., Weimer, D., & Pannek, J. (2015). Network structures and decentralized control in logistics: topology, interfaces, and dynamics, *International Journal of Advanced Logistics*, 4(1), 1-8.
- [64] Seitz, KF., & Nyhuis, P. (2015). Cyber-Physical Production Systems Combined with Logistic Models – A Learning Factory Concept for an Improved Production Planning and Control, *Procedia CIRP*, 32, 92-97.
- [65] Frazzon, E. M., Silva, L. S., & Hurtado, P. A. (2015). Synchronizing and Improving Supply Chains through the application of Cyber- Physical Systems, *IFAC-PapersOnline*, 48(3), 2059-2064.
- [66] Oliveira, R. R., Cardoso, I. M. G., Barbosa, J. L. V., da Costa, C. A., & Prado, M. P. (2015). An intelligent model for logistics management based on geofencing algorithms and RFID technology, *Expert Systems with Applications*, 42, 6082-6097.
- [67] Ng, I., Scharf, K., Pogrebna, G., & Maull, R. (2015). Contextual variety, Internet-of-Things and the choice of tailoring over platform: Mass customisation strategy in supply chain management, *Int. J. Production Economics*, 159, 76–87.
- [68] Verdouw, C., Robbmond, R., Verwaart, T., Wolfert, J., & Beulens, A. (2015). A reference architecture for IoT-based logistic information systems in agri-food supply chains, *Enterprise Information Systems*, 12, 755-779.
- [69] Z., D. R., Nait-Sidi-Moh, A., Durand, D., & Fortin, J. (2015). Using Internet of Things technologies for a collaborative supply chain: Application to tracking of pallets and containers, *Procedia Computer Science*, 56, 550 – 557.
- [70] Kassahun, A., Hartog, R. J. M., Sadowski, T., Scholten, H., Bartram, T., Wolfert, S., & Beulens, A. J. M. (2014). Enabling chain-wide transparency in meat supply chains based on the EPCIS global standard and cloud-based services, *Computers and Electronics in Agriculture*, 109, 179-190.
- [71] Cheng, M., Luo, H., Zhong, R. Y., Lan, S., & Huang, G. Q. (2014). Cloud-based Auction Tower for Perishable Supply Chain Trading, *Procedia CIRP*, 25, 329-336.
- [72] Ivanov, D., Sokolov, B., & Raguinia, E. A. D. (2014). Integrated dynamic scheduling of material flows and distributed information services in collaborative cyber-physical supply networks, *International Journal of Systems Science: Operations & Logistics*, 1(1), 18-26.
- [73] Geerts, G. L., & O’Leary, D. E. (2014). A supply chain of things: The EAGLET ontology for highly visible supply chains, *Decision Support Systems*, 63, 3–22.
- [74] Verdouw, C., Beulens, A., & Vorst, J. v. (2013). Virtualisation of floricultural supply chains: A review from an Internet of Things perspective, *Computers and Electronics in Agriculture*, 99, 160–175.
- [75] Eckhardt, J., & Rantala, J. (2012). The role of intelligent logistics centres in a multimodal and cost-effective transport system, *Procedia - Social and Behavioral Sciences*, 48, 612-621.
- [76] Guo, Z., Zhang, Z., & Li, W. (2012). Establishment of Intelligent Identification Management Platform in Railway Logistics System by Means of the Internet of Things, *Procedia Engineering*, 29, 726 – 730.
- [77] Sun, C. (2012). Application of RFID Technology for Logistics on Internet of Things, *AASRI Procedia*, 1, 106 – 111.

[78] Jun, C., & Wei, M. Y. (2011). The Research of Supply Chain Information Collaboration Based on Cloud Computing, *Procedia Environmental Sciences*, 10, 875-880.

[79] Adamski, A. (2011). Hierarchical Integrated Intelligent Logistics System Platform, *Procedia Social and Behavioral Sciences*, 20, 1004-1016.

[80] Wadhwa, P. S., & Madaan, J. (2006). Intelligent Agent Technology For Reverse Logistics, *IFAC Proceedings Volumes*, 39(3), 215-220.

8. TWO Meta-Sezgisel Algoritma

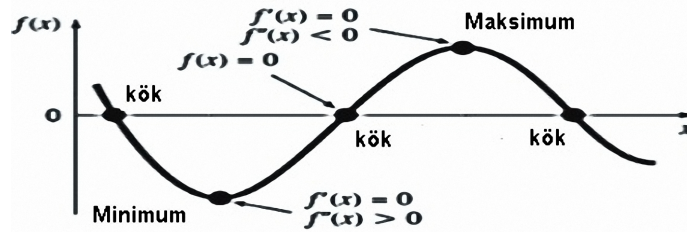
TWO Meta-Sezgisel Algoritması ve Uygulaması

Numan ÇELEBİ¹

¹Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilişim Sistemleri Mühendisliği Bölümü, Sakarya / Türkiye

8.1 Giriş

Optimizasyon, belirli şartlar altında herhangi bir problemi en iyi yapan sonuçları içeren çözüm(leri)ü bulma işlemidir. Başka bir tanımlama ile optimizasyon, verilen kısıtlara bağlı olarak amaç fonksiyonunu maksimum (karı arttırmayı) veya minimum (maliyeti azaltmayı) yapmayı amaçlar. Optimizasyon ve kök bulma birbirleriyle ilişkili olan iki kavramdır. Kök bulma, fonksiyonu sıfır yapan noktaları bulma işlemidir. Optimizasyon ise fonksiyon üzerindeki minimum ya da maksimum olan noktayı belirlemeye çalışır. Optimum, eğrinin düz olduğu noktadır. Bu nokta yani fonksiyonun birinci türevinin $f'(x)$ sıfır olduğu yerdir. İkinci türev $f''(x)$ optimum noktanın minimum mu yoksa maksimum mu olduğunu gösterir (Şekil 8.1). Eğer $f''(x) < 0$ ise bulunulan nokta fonksiyonun maksimum noktasıdır. Eğer $f''(x) > 0$ ise bulunulan nokta fonksiyonun minimum noktasıdır.

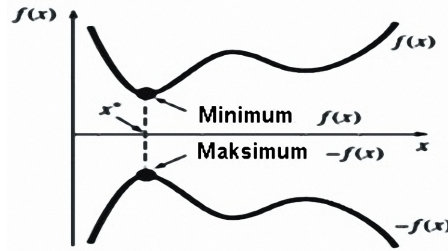


Şekil 8.1: Optimizasyon ve kök bulma arasındaki ilişki

Optimizasyon problemlerinin matematiksel olarak genel formu aşağıdaki biçimde yazılır.

$$\begin{aligned} \min f_i(x), \quad (i = 1, \dots, M) \\ \text{subject to } \phi_j(x) = 0 \quad (j = 1, \dots, J) \\ \varphi_k(x) < 0, \quad (k = 1, \dots, K) \end{aligned}$$

Bu ifadede $f_i(x)$, $\phi_j(x)$ ve $\varphi_k(x)$ tasarım fonksiyonlarıdır. x_i , tasarım ya da karar değişkenleri olarak adlandırılır. $f_i(x)$ fonksiyonuna amaç fonksiyonu denir. $M = 1$ ise fonksiyon tek değişkenli, tek amaçlı bir fonksiyona dönüşür. Amaç fonksiyonu lineer ya da non-lineer olabilir. $\phi_j(x)$ ve $\varphi_k(x)$ eşitsizlikleri kısıtlar olarak adlandırılır. Amaç fonksiyonunu maksimizasyon problemi olarak ta formülize edebiliriz. Yani $f_i(x)$ 'in minimizasyonu, $-f_i(x)$ 'in maksimizasyonuna eşittir (Şekil 8.2).



Şekil 8.2: Maksimum ve minimumluk arasındaki ilişki

Eğer $\phi_j(x)$ ve $\varphi_k(x)$ kısıtları ve amaç fonksiyonu lineer ise problem lineer programlama adını alır. Lineer programlama için en bilinen yöntem Simplex metodudur. Eğer $f_i(x)$, $\phi_j(x)$ ve $\varphi_k(x)$ non-lineer ise o zaman problem non-lineer optimizasyon problemi adını alır. Eğer $J = K = 0$ olursa bu durumda problem kısıtsız optimizasyon problemi adını alır. Bu durum çoğu zaman problemi kolaylaştırır. Örneğin, literatürde Rosenbrock Banana fonksiyonu olarak bilinen aşağıdaki fonksiyonun minimumu türev yöntemi ile bulunmuştur.

$$f(x, y) = (1 - x^2) + 100(y - x^2) \quad (8.1)$$

$$\frac{\delta y}{\delta x} = 2(1 - x) - 400(y - x^2)x = 0 \quad (8.2)$$

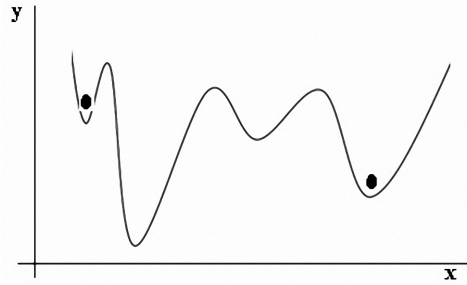
$$\frac{\delta y}{\delta x} = 200(y - x^2) = 0 \quad (8.3)$$

Üç numaralı denklem $y = x^2$ olduğunu ifade etmektedir. $2(1 - x) - 400(y - x^2)x = 0$ denkleminden $(1 - x) = 0$ olduğu görülmektedir. Bu durumda $f(x)_{\min} = 0$ değeri, $x = y = 1$ noktasında elde edilir.

Sonuç olarak bu gibi fonksiyonlar için gradient-based (Newton's Rapshon gibi) metotları fonksiyonun minimumunu bulmak için kullanabiliriz. Ancak çözümü aranan fonksiyonun gradient'ini bilmediğimiz bir fonksiyon ise ya da, ele alınan fonksiyondaki karar değişkenlerinin sayısı çok fazla ise ya da fonksiyon çok çukurlu bir yapıya sahip ise fonksiyonun minimum/maksimum değerini bulmada bazı sorunlar ortaya çıkar. Aşağıda verilen kısımlarda bu sorunlar kısaca açıklanmıştır.

8.1.1 Fonksiyonun türevinin tanımsız veya çok modlu olması durumu

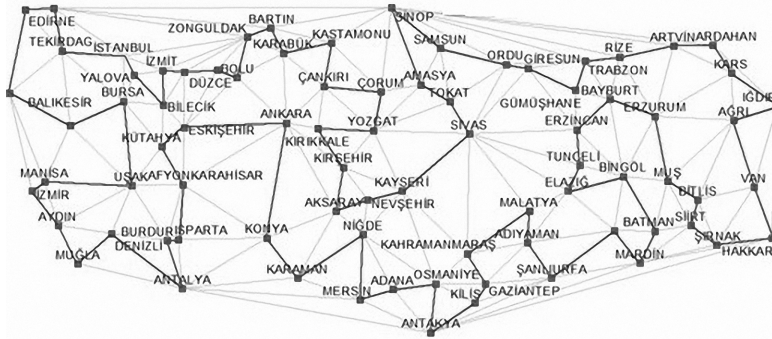
Türev işlemi için $f(x,y) = (|x| + |y|) \exp [+ \sin(x^2) + \sin(y^2)]$ fonksiyonunu göz önüne alalım. Bu fonksiyonunun $(0,0)$ 'daki türevi $|x| + |y|$ faktöründen dolayı tanımsızdır. Bu durumda gradient-based metotlarını kullanamayız. Bundan dolayı Nelder-Mead ve Simplex metot gibi gradient-free metotlara ihtiyaç vardır. Amaç fonksiyonu sadece bir çukura sahipse optimizasyon problemi unimodal olur. Bu durumda bu metotlarla çözümü bulmak mümkün olur. Tek çukurlu fonksiyonlarda bulunan yerel optimum çözüm, tek optimumdur (aynı zamanda bulunan nokta global optimumdur). Ancak amaç fonksiyonu multimodal olduğunda (sin fonksiyonu gibi) bu tür metotlar çözümü bulmakta yetersiz kalmaktadır. Bu metotlarda eğer aranan minimum nokta başlangıç noktasından çok uzak ise, algoritma genellikle yerel minimuma yakalanır ve global minimumu bulamaz. Örneğin aşağıdaki fonksiyonun minimum noktasını bulmak istersek başlangıç değerleri sağ ekseninde olduğunda farklı, sol ekseninde olduğunda farklı sonuç verebilir.



Şekil 8.3: Fonksiyonun çok modlu olması durumu

8.1.2 Karar değişkenlerinin çok olması durumu

Başka bir önemli problemde, karar değişkenlerinin artması ya da çok büyük olması durumunda ortaya çıkar ($n = 50000$ gibi). Örneğin iyi bilinen gezgin satıcı problemi (Şekil 8.4) N tane şehir ziyaret edecek en kısa yolu bulmayı amaçlar. Bu kombinatoriyel bir problemdir ve şehir sayısı n olması durumunda çözüm sayısı $n!$ olur. Eğer $n = 100$ olması durumunda çözüm sayısı 9.3×10^{157} gibi astronomik bir rakama erişir. Bu rakam tüm kombinasyonları araştırmanın/denemenin pratik olmadığı anlamına gelmektedir. Bu durumda başka alternatif metotlar kullanmak gerekir.



Şekil 8.4: Optimizasyon da karar değişkenlerinin çok olması durumu

Meta-Sezgisel metotlar yukarıda kısaca açıklanan bu tür problemlerin üstesinden gelmek için

tasarlanmış algoritmalarıdır. Bu algoritmaların çoğu tabiattan esinlenerek geliştirilmişlerdir. Tabiat-taki başarılı davranışları örnek alırlar. Meta-Sezgisel algoritmalar, deneme yanılmayı, öğrenmeyi ve adaptasyonu kullanarak problemleri çözerler. Bu algoritmalar her zaman en iyi çözümü garanti etmez ancak makul bir zaman içerisinde optimale yakın çözümler elde etmemizi sağlar.

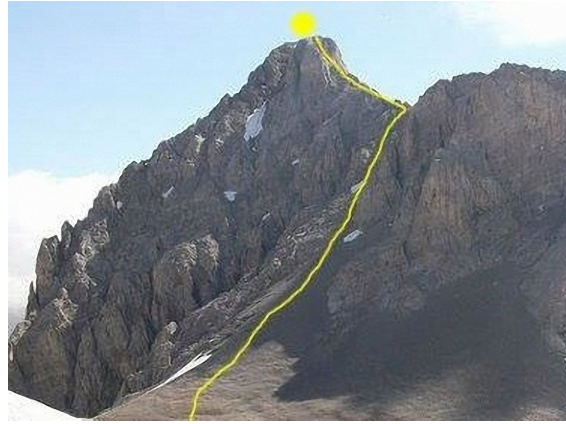
Sonuç olarak, eğer tasarım değişkenleri ayrık (discrete) ise o zaman optimizasyon problemi ayrık optimizasyon olarak adlandırılır. Ayrık optimizasyon kombinatoriyel problem olarak bilinir. Gezgin satıcı problemi bu sınıfa giren bir problemdir. Eğer belli bir aralıkta (interval) tüm değişkenler reel değişkenler olursa optimizasyon problemi sürekli (continuous) problemi adını alır.

8.2 Meta-Sezgisel Algoritmaların Çalışma Mantığı

Bu bölümde meta-sezgisel algoritmaların optimizasyon noktasını arama mantığı açıklanmaya çalışılacaktır. Bunun için yörünge esaslı, sürü esaslı ve gelişim esaslı algoritmalarından örnekler verilmiştir.

8.2.1 Yörünge esaslı algoritmalar

Optimal bir çözümü bulmak hazine araştırmasına benzetilir. Tepeli bir arazide gömülü bir hazineyi araştırdığımızı düşünelim. Elimizde herhangi bir kılavuz (guide) yok ise arama sadece random bir yapıya dönüşür. Dolayısıyla etkin bir yolumuz olmadığını gösterir. Bu durumda tüm noktaları araştırmak gerekir. Eğer hazinenin en yüksek tepede olduğu söylenirse, araştırma yapanlar direkt olarak en yüksek tepeye çıkacak yolları araştırmaya yönelir. Bu yaklaşım klasik Hill-Climbing tekniği olarak bilinir. Ancak tepeye çıkmak için uygun olan yolu belirten bir kılavuz olması durumunda işlem daha kolay hale dönüşür (Şekil 8.5). Çoğu durumda arama işlemi için elimizdeki kılavuza bağlı olarak bulunduğumuz/işaretleli noktalarda rastgele dolaşımalar yaparız. Yani arama işleminde işaret olmayan bölgelerde arama işlemi gerçekleştirilmez. Bu şekildeki bir hazine arama işlemi yörünge tabanlı bir yöntemle daha hızlı gerçekleştirilebilir. Örneğin Simulated Annealing sıcaklığı kullanan yörünge esaslı bir algoritmadır.

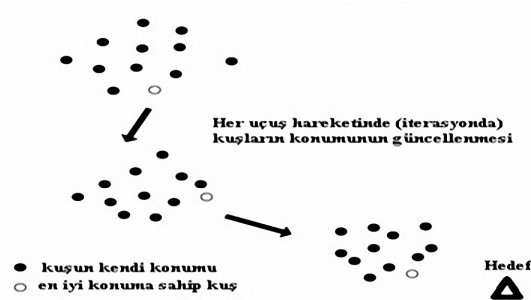


Şekil 8.5: Yörünge esaslı arama yaklaşımı

8.2.2 Sürü esaslı algoritmalar

Hazine arama işleminde elimizde bir yörünge yok ise bu durumda bir grup insana sorarak bilgiyi paylaşma yöntemi ile arama işlemini gerçekleştirebiliriz. Örneğin parçacık sürü optimizasyon (PSO)

algoritması bu tür bilgi paylaşımlı algoritmalar sınıfına girer. Kuşların uzayda yerini bilmedikleri yiyeceği (hazine araştırması) arama işlemi, bir problemin çözümünü aramaya benzetilir. Kuşlar yiyeceğe en yakın kuşu takip ederler ve onun konum bilgisini kendi konum bilgileri ile yer değiştirirler (Şekil 8.8). Kuşların ne yöne uçacağı sürü içindeki en iyi konumdaki kuşun (grup bilgisi) koordinatı ile o ana kadar ki kendi en iyi konum bilgisinin vektörel birleşimine göre belirlenir. Her kuşun ayrı bir en iyi değeri (personelBest) olurken, kuşlar içindeki en iyi personelBest değerine sahip kuşun değeri, tüm sürü için hedef koordinat olarak paylaşılır ve global en iyi değer (globalBest) olarak bilinir.



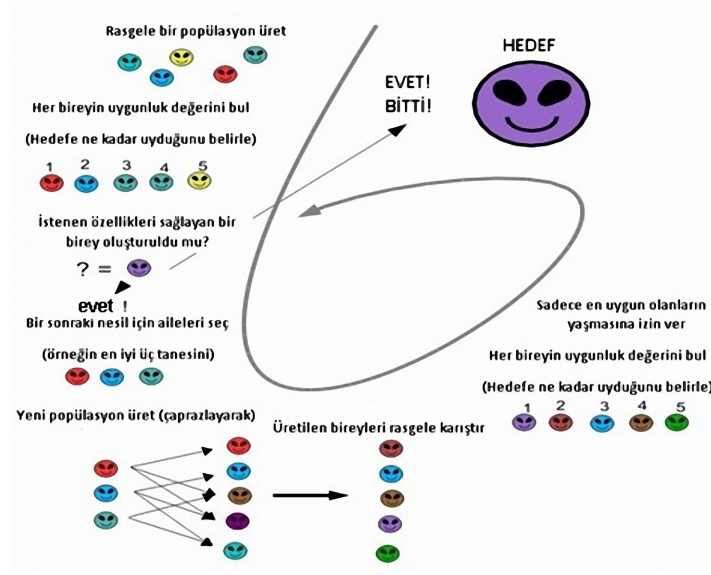
Şekil 8.6: Sürü esaslı arama yaklaşımı

8.2.3 Gelişim esaslı algoritmalar

Arama stratejimizi biraz daha geliştirebiliriz. İyi olan avcılarını tutarak arama işlemine başlayabiliriz. Evaluation algoritma bu tür metotlara örnektir. Değerlendirme işlemi bireylerin uygunluk (fitness) değerlerine göre yapılır. Tabiat binlerce yıldır çeşitli problemleri çözüyor ve sadece en iyi ve en güçlü çözümler hayatta kalıyor. Gelişim esaslı algoritmalar tabiatın bu davranışlarından esinlenerek geliştirilmiş algoritmalarlardır. Örneğin; genetik algoritma ve gelişim algoritması gibi. Başlangıçta üretilen popülasyon içerisindeki bir bireyin çözüme ne kadar yakın olduğu, çözülmesi istenen probleme bağlı bir amaç fonksiyonudur. Bu algoritmalar her adayın ne kadar güçlü olduğunu bu amaç fonksiyonuna göre hesaplar ve buna göre bir sonraki neslin olacak ebeveynlerini ya da olacak bireylerini belirler. Daha sonra makul bir nesil oluşturmak için ebeveynlere genetik arama işlemlerini (üreme, çaprazlama ve mutasyon) uygular. Bu döngü her defasında daha güçlü bireyler oluşturarak hedeflenen sonuç elde edilinceye kadar tekrarlanır (Şekil 8.7).

8.2.4 Meta-Sezgisel algoritmaların çözüm yaklaşımı

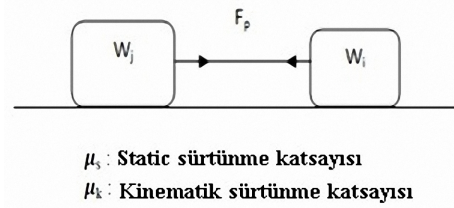
Meta-Sezgisel algoritmalar için iki temel eleman vardır. Birincisi en iyi fitness değerini seçme; bu işlem optimaliteye yakınsamayı (convergence) sağlar. İkincisi rasgelelik; elde edilen çözümlerin lokal optimuma takılmasını (yakalanmasını) engeller. Aynı zamanda çözümlerin çeşitliliğini artırır. Sonuç olarak Simulated Annealing (Tavlama Benzetimi Algoritması) yörünge Esaslı Algoritma; Particle Swarm Optimization (PSO)-Parça Sürü Optimizasyon Algoritması, grup bilgisi paylaşımlı algoritma; Evaluation Algorithm (EA), Gelişim algoritması, en iyinin yaşatılması esasına dayanan algoritmalarlardır.



Şekil 8.7: Gelişim esaslı arama yaklaşımı

8.3 TWO Algoritması

TWO (Tug Of War Optimization) algoritması fiziksel modellemesi ip çekme oyunundan esinlenerek geliştirilmiş sürü esaslı bir meta-sezgisel algoritmadır. İdealize edilmiş ip çekme oyununda ve iki takımın ağırlıkları olmak üzere düz bir yüzey üzerinde yatıyor oldukları göz önüne alınır.



Şekil 8.8: Gelişim esaslı arama yaklaşımı

Takımlar Newton'un üçüncü yasasına göre F_p gibi iki tane eşit ve karşıt kuvvete maruz kalır. (i) nesnesi için, çekme kuvveti maksimum statik sürtünme kuvvetinden küçük olduğu sürece, nesne (takım) yerinde kalır. Aksi takdirde, oluşan çekme kuvveti aşağıdaki formül ile hesaplanır.

$$F_r = F_p - W_i \mu_k \quad (8.4)$$

Bu kuvvetin sonucunda nesne/takım (i), Newton'un ikinci yasasına göre nesne/takım (j)'ye doğru aşağıdaki formüle göre ivmelenir.

$$a = \frac{F_r}{W_i/g} \quad (8.5)$$

Böylece (i) nesnesinin yeni pozisyonu aşağıdaki formüle göre hesaplanır.

$$X_i^{new} = \frac{1}{2}at^2 + X_i^{old} \quad (8.6)$$

Her bir aday çözüm $X_i = \{x_{i,j}\}$, ip çekme yarışmasında olan takımları temsil eder. Çekme kuvveti, ağırlıkla orantılı olduğu kabul edilir. Doğal olarak karşı takım yarışmayı kaybetmemek için ipi en azından aynı kuvvetle tutmak zorundadır. Hafif olan takım ağır olan takımın tarafına doğru ivmelenir (hızlanır). Bu durum TWO algoritmasının yakınsamasını (convergence) ifade eder. Aşağıda algoritmanın adımları verilmiştir.

Adım-1. Başlatma (Initalization)

N elemanlı bir başlangıç çözümü rasgele üretilir.

$$x_{ij}^0 = x_{j,\min} + rand(x_{j,\max} - x_{j,\min}) \quad j = 1, 2, \dots, N \quad (8.7)$$

Bu formülde x_{ij}^0 i . aday çözümün, j . değişkeninin başlangıç çözümüdür. $x_{j,\max}$ ve $x_{j,\min}$ j . değişken için izin verilen sınır değerleridir. rand uniform dağılıma göre [0,1] aralığında rasgele sayı üreticidir.

Adım-2. Aday çözümlerin değerlendirilmesi ve ağırlık atanması

Aday çözümler için amaç fonksiyonunun değerleri hesaplanır. Tüm çözümler leage (lig) olarak adlandırılarak hafızada saklanır. Her çözüm aşağıdaki ağırlık ile bir takım (team) olarak dikkate alınır.

$$W_i = \left(\frac{fit(i) - fit_{worst}}{fit_{best} - fit_{worst}} \right) + 1 \quad i = 1, 2, \dots, N \quad (8.8)$$

$fit(i)$ i .particle (bireyin) uygunluk (fitness) değeridir. fit_{best} takım içerisindeki en iyi fitness değeridir. fit_{worst} takım içerisindeki en kötü fitness değeridir.

Adım-3. Çekişme ve yer değiştirme

Takımdaki her bir birey, her iterasyonda yeni pozisyonuna hareket etmek için diğer takım bireylerinin tümüne karşı yarışır. Takım tarafından çıkarılan güç statik sürtünme katsayısına W_{μ_s} eşit olduğu kabul edilir. i ve j takımları arasındaki çekme gücü $F_{p,ij} \max\{W_i\mu_s, W_j\mu_s\}$ olarak belirlenir. Bu tanım, ağır olan (güçlü olan) takımın pozisyonu değişmemiş olarak ifade eder. k . iterasyondaki çekişme sonucunda j takımı ile i takımını etkileyen güç aşağıdaki formülle hesaplanır:

$$F_{r,ij}^k = F_{p,ij}^k - W_i^k \mu_k \quad (8.9)$$

Bu formülde, $F_{p,ij}^k$ k . iterasyonda i ve j takımları arasındaki çekme kuvvetidir ve μ_k de kinematik sürtünme katsayısıdır. Sonuç olarak i takımı j takımına doğru hızlanması aşağıdaki formüle göre gerçekleşir.

$$a_{ij}^k = \left(\frac{F_{r,ij}^k}{W_i^k \mu_k} \right) g_{ij}^k \quad (8.10)$$

a_{ij}^k itakımının k .iterasyonda j takımına doğru hızlanmasını ifade eder. g_{ij}^k yer çekimi ivmesidir ve şu şekilde ifade edilir:

$$g_{ij}^k = X_j^k - X_i^k \quad (8.11)$$

Bu fomüldeki X_j^k ve X_i^k k .iterasyondaki *ive* jaday çözümler için pozisyon vektörleridir. Sonuç olarak; itakımının j takımı ile çekişmesi sonucunda meydana gelen yer değişim miktarı (displacement) aşağıdaki formüle göre belirleni.

$$\Delta X_{ij}^k = \frac{1}{2} a_{ij}^k \Delta t^2 + \alpha^k \beta (X_{\max} - X_{\min}) \cdot randn(1, n) \quad (8.12)$$

α [0.9, 0.99] aralığında seçilen bir sabittir. Büyük olduğunda algoritmanın yakınsamasını yavaşlatır, arama uzayında daha fazla keşifler yapar. β parametresi araştırma uzayında hareket ederken aday çözümlerin adımlarını kontrol eder (0,1] aralığında seçilir. X_{\max} ve X_{\min} araştırma uzayının üst ve alt sınır değerlerdir. $randn(1, n)$ normal dağılıma göre üretilen random sayı vektörüdür. itakımının toplam yer değiştirmesi aşağıdaki formülle hesaplanır.

$$\Delta X_i^k = \sum_{j=1}^N \Delta X_{ij}^k \quad (8.13)$$

k .iterasyon sonunda itakımının yeni pozisyonu aşağıdaki formüle göre belirlenir.

$$X_i^{k+1} = X_i^k + \Delta X_i^k \quad (8.14)$$

Adım-4. Güncelleme

Her iterasyon (döngü) sonunda bir birleriyle rekabet eden takımlar güncellenmesi gerekir. Bu işlem bir önceki iterasyondaki çözüm değeri (fitness) ile bulunulan iterasyondaki çözüm değerinin karşılaştırılması ile gerçekleştirilir. Bulunulan iterasyondaki çözüm değeri (fitness) bir önceki çözüm değerinden iyi ise yer değiştirilir. Çözüm değeri iyi değilse var olan çözüm değeri ile bir sonraki iterasyona geçilir.

Adım-5. Sınır kontrolü

Yarışan takımların araştırma uzayı (alanı) dışına çıkması durumunda, takımların tekrar saha içine çekilmesi işlemidir. Bu işlem farklı meta-heuristik algoritalarda farklı biçimde yapılabilmektedir. Genellikle (flayback strategy) sınırı aşmadan önceki pozisyona döndürülme yöntemi kullanılır. TWO algoritmasında ise aşağıdaki strateji kullanılmaktadır.

$$X_{ij}^k = GB_j + \left(\frac{randn}{k} \right) (GB_j - X_{ij}^{k-1}) \quad (8.15)$$

Bu formülde GB_j şu ana kadar ki en iyi çözüm $randn$ ise standart normal dağılımı ifade etmektedir.

Adım-6. Sonlandırma

Adım-2 ile Adım-5 arasındaki işlemler, sonlanma kriterleri gerçekleşene kadar tekrar edilir. Adımları yukarıda açıklanan TWO algoritmasının sözde kodunu aşağıda verilmiştir.

Procedure Tug of War Optimization**Begin**

Parametrelere değer ataması yap;

N boyutlu rasgele bir başlangıç aday popülasyonu (çözümü) üret;

Tüm rasgele aday çözümleri kaydederek lig'i (müsabakayı) başlat;

While Sonlandırma şartı sağlanana kadar **do**

Aday çözümler için amaç fonksiyonunun değerini hesapla;

Yeni çözümleri sırala ve lig'i güncelle;

Ligdeki takımların W_i ağırlıklarını $fit(x_i)$ 'ye göre güncelle;

For Her takımı için

If $W_i > W_j$

Denklem (12)'u kullanarak takımını takımına doğru hareket ettir;

End if

Denklem (13)'u kullanarak takımının toplam yer değiştirmesini hesapla;

Denklem (14)'i kullanarak takımının son pozisyonunu hesapla;

Sınır kontrolünü yap;

End for

End while

End

8.4 TWO Algoritmasının Uygulaması

Bu bölümde sürekli optimizasyon probleminin TWO algoritması ile çözümünün gösterildiği örnek bir çalışma vermiştir. Bunun için literatürde iki değişkenli ve altı hörgüçlü deve sırtı fonksiyonu olarak bilinen aşağıdaki test fonksiyon göz önüne alınmıştır.

$$z = \left[4 - 2.1x^2 + \frac{x^4}{3} \right] x^2 + xy + [-4 + 4y^2]y^2$$

Bu fonksiyonda x değişkeninin sınırları ± 3 ve y değişkeninin sınırları ± 2 aralığında tanımlanmıştır. Global minimum değeri $x = -0.0898, y = 0.7126$ noktalarında $z = -1.0316$ olarak veriliyor. Bu fonksiyonun global minimum değerini TWO algoritmasını kodlayarak elde edilmesi istenmektedir. Bu çalışmada TWO algoritması JAVA programlama dili kullanılarak kodlanmıştır. Programın çalıştırılması ile elde edilen global minimum değeri ve bu değer gerçekleştiği x ve y değerleri aşağıda programın ekran çıktı görüntüsünde verilmiştir.

Ele alınan test fonksiyonuna göre TWO algoritmasının çözümü bulma zamanı yani çözüme yakınsama grafiği aşağıda verilmiştir.

Grafikten de gözüktüğü gibi TWO algoritması global minimum değerine 11. iterasyonda ulaşmıştır. Algoritmanın bu kadar kısa bir zamanda global minimuma erişmesi beklenen bir durumdur. Zira ele alınan test fonksiyonu eğitim amaçlı olup basit fonksiyonlar sınıfında yer almaktadır.

8.5 Sonuç

Bu çalışmada optimizasyon konusunda temel kavramlar hakkında kısa bir açıklama yapılmıştır. Klasik optimizasyon yöntemleri ile problemleri çözmeye karşılaşılan sorunlardan bahsedilmiştir. Bu tür problemleri aşmak için geliştirilmiş olan sezgisel algoritmalarından yörünge tabanlı, grup bilgisi

```

119         secici=1;
120     }
121     }
122 }
123 }
124 }
125 //sınır kontrolü yapıyoruz..
126 //use the side constraint handling technique to regenerate violating variables
127 }
128 for(int i=0; i<boyut; i++) {

```

```

<terminated> TWO_ALGORITMASI_METOTLU [Java Application] C:\Program Files\Java\jre-10.0.1\bin\javaw.exe (5 Eyl 2018 21:40:40)
fitness----> -1.0316020801967878 x-----> -0.092362123287936 y-----> 0.7132812489656892
fitness----> -1.0316020801967878 x-----> -0.092362123287936 y-----> 0.7132812489656892
fitness----> -1.0316020801967878 x-----> -0.092362123287936 y-----> 0.7132812489656892
fitness----> -1.0316085321321857 x-----> -0.09203829202485193 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.0920318204626079 y-----> 0.7123952547155487
fitness----> -1.0316086363959247 x-----> -0.0920318204626079 y-----> 0.7123952547155487
fitness----> -1.0316086363959194 x-----> -0.09203181866996275 y-----> 0.7123952545728923
fitness----> -1.0316086363959194 x-----> -0.09203181866996275 y-----> 0.7123952545728923
fitness----> -1.0316086363959247 x-----> -0.0920318186696565 y-----> 0.7123952545728852
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851
fitness----> -1.0316086363959247 x-----> -0.09203181866965128 y-----> 0.7123952545728851

```

Şekil 8.9: Ele alınan fonksiyonun TWO algoritması ile elde edilen çözüm ekranı



Şekil 8.10: Algoritmanın yakınsama grafiği

paylaşımli ve evrimsel esaslı algoritmaların çalışma mantığı ve çözüm yaklaşımları açıklanmıştır. Çalışmanın deneysel aşamasında iki yıllık geçmişe sahip yeni bir meta-sezgisel algoritma olan TWO algoritması açıklanarak, çalışma adımları verilmiştir. Algoritma Java programlama dili ile kodlanmış ve literatürde altı hörgüçlü deve fonksiyonu olarak bilinen problemin optimum değerleri kodlanan program kullanılarak elde edilmiştir. Deneysel sonuçlar kısmında programın ekran çıktı görüntüleri ve algoritmanın yakınsama grafiği verilmiştir.

8.6 Kaynakça

1. Chapra, Steven C. Numerical methods for engineers, McGraw-Hill International Edition, 2006.
2. Yang, Xin-She. Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley & Sons, Inc. 2010
3. Çelebi, N. Bir Afet Yardım Merkezinin Sezgisel Algoritmalar Yardımıyla Konumlandırılması, XI. Ulusal Üretim Araştırmaları Sempozyumu 23–24 Haziran 2011,
4. Kaveh, A. Zolghadr, A. Truss shape and size optimization with frequency constraints using Tug of War Optimization, Asian Journal of Civil Engineering, Vol. 18, No.2, 2017
5. Kaveh, A. Zolghadr, A. A novel Meta-Huristic Algorithm: Tug of War Optimization, International Journal of Optimization in Civil Engineering, 6(4):469-492, 2016
6. Kaveh, A. Shokohi, F A.Ahmadi, B. Optimal Analysis and Design of Water Distribution Systems Using Tug of War Optimization Algorithm, International Journal of Optimization in Civil Engineering, 6(4):191-208, 2017

9. Derin Öğrenme Uygulaması

Derin Öğrenme Tabanlı Kalite Kontrol Uygulama Tasarımı

Prof. Dr. Orhan TORKUL¹, Muhammet Raşit CESUR¹, Elif CESUR²

¹Sakarya Üniversitesi, ¹İstanbul Medeniyet Üniversitesi

9.1 Giriş

Bilgisayarların donanımsal gelişimi yapay zeka teknolojilerinin gelişimine daima olumlu katkı sağlamıştır. Daha çok işlem gerektiren karmaşık analizlerin yapılması ve sonuç elde edilmesi bilgisayarlardaki gelişmelere bağlı olarak mümkün hale gelmiştir. Bu sayede yapay zeka sistemleri daha kapsamlı analizler yapabilecek ve daha geniş problemleri çözebilecek hale gelmiştir. Bu değişim yapay sinir ağlarını etkileyerek bu alanda yeni bir teknolojik dönüşümün kapısını aralamıştır. Yapay sinir ağlarının dönüşümü bir bütün olarak çalışan sinir ağının dinamik çalışmasına olanak tanımıştır. Dinamik çalışan yapay sinir ağları, klasik yapay sinir ağlarının aksine problemin bütünü aynı modeli kullanarak değil, problemin alt kümelerine en çok uyacak modeller bütünü kullanarak problemi çözmektedirler.

Problemi küçük parçalara ayırarak detaylı alt çözümlerden sonuç üreten dinamik yapay sinir ağları, çalışma biçimlerine uygun olarak, derin sinir ağları olarak adlandırılmışlardır. Derin sinir ağlarında yapay sinir ağlarının aksine ağı düğümleri (nöronlar) problemin ayrı parçaları üzerinde uzmanlaşırlar. Yapay sinir ağlarında ise tüm nöronlar problemin bütününe bir çözüm getirmeye çalışmaktadırlar. Bundan dolayı derin ağlar daha küçük detayları yakalayarak daha doğru sonuç üretmektedirler. Fakat derin ağların detayları yakalayabilmesi için detayları içeren büyük miktarda veri ile eğitilmesi gerekmektedir. Büyük veri derin ağların eğitim sürecini oldukça uzatsa da, derin ağların yapay sinir ağları üzerinde kurduğu üstünlük karmaşık problemlerin çözümünde bu teknolojinin yaygın olarak kullanılmasını sağlamıştır.

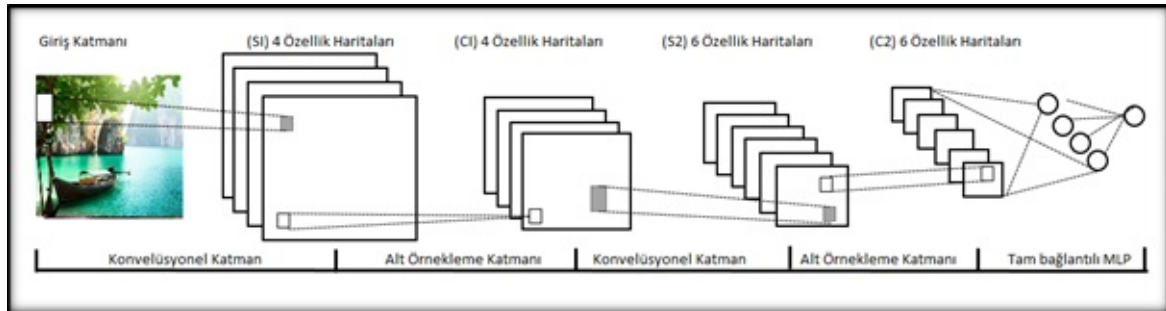
9.2 Derin Öğrenme Mimarileri

Düşünme ve problem çözme esnasında beynimiz biz farkında olmadan çok sayıda işlem gerçekleştirir. Bu işlemler genellikle sınıflandırma (anormallik tespiti, nesne tanıma, yüz tanıma, vb.), kümeleme, tahmin, optimizasyon, simülasyon, çıkarım işlemleridir. Bir yapay zeka sisteminin problemleri çözebilmesi için bahsedilen işlemleri gerçekleştirecek mekanizmalara sahip olması gerekmektedir. İhtiyaç duyulan yapının oluşturulabilmesi için derin sinir ağlarında çok sayıda katman tasarlanarak her katmanın belirli işlerde özelleşmesi sağlanabilmektedir. Her katman uzmanlaştığı konuda yaptığı çıkarımları daha ilerideki katmanlara iletir. Önündeki katmanlardan veri alan katmanlar da aldıkları girdileri kullanarak sonuç üretirler.

Örneğin ürün tanıyacak bir sistemde bir katman ürünün rengini tespit edebilir. Çünkü her rengin çok sayıda tonu bulunmaktadır ve kamera fotoğrafını çektiği ürünün rengini, ışıktandırmaya bağlı olarak yakın ama farklı tonlarda gösterecektir. Başka bir katman ürünün büyüklüğü hakkında, bir katman da ürünün şekli hakkında karar verebilir. Bu kararlar bir katmana iletilip, ilgili katmanda ürünün detay bilgileri değerlendirilerek, ürünün tespiti sağlanabilir. Derin öğrenme sistemlerinin çalışma sistemikleri yapay sinir ağlarının dinamikleştirilmesine dayansa da ihtiyaca ve probleme özgü olarak çeşitli derin öğrenme algoritmaları geliştirilmiştir. Bu bölümde konvolüsyonel sinir ağları, derin inanç ağları ve derin oto-kodlayıcılar incelenmiştir (Liu et al., 2017).

9.2.1 Kıvrımlı (Konvolüsyonel) Sinir Ağları

Kıvrımlı sinir ağları ilk kez (LeCun, Bottou, Bengio, & Haffner, 1998) tarafından geliştirilmiştir ve günümüzde birçok alanda uygulamaları görülmektedir. Kıvrımlı sinir ağları çok katmanlı ileri beslemeli yapay sinir ağlarının bir türüdür ve birçok kıvrımlı bütünleştirici katmandan oluşmaktadır. Kıvrımlı sinir ağları gizli katmanlara sahip olması, ağırlık atamaları, hata hesaplamaları gibi özellikleri açısından yapay sinir ağı ile ortak özellik göstermektedir. Aynı zamanda, kıvrımlı sinir ağları tam bağımlı katmanlara ihtiyaç duymazlar ve ayrıca gizli katmanlarda farklı görevler için özelleşebilirler (Deng, 2014).

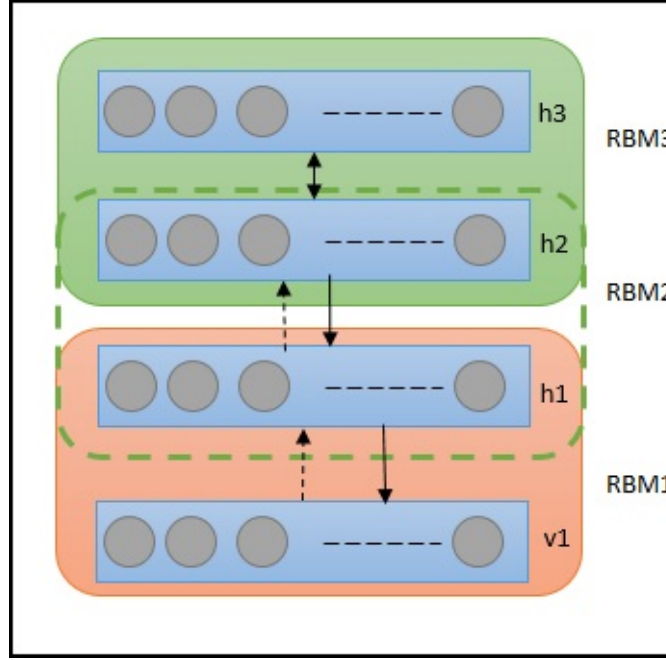


Şekil 9.1: Kıvrımlı sinir ağlarının iş akış şeması

9.2.2 Derin İnanç Ağları

Derin inanç ağları tek seferde bir gizli özellik katmanını öğrenmek amacıyla Kısıtlı Boltzmann Makinesi (RBM) kullanılarak eğitilmiş olasılıklı bir modeldir (Hinton, Osindero, & Teh, 2006). İlk katın eğitim süreci tamamlandıktan sonra, ikinci RBM'nin gizli birimleri katmanı eklenir. İstenen katman sayısına ulaşılan kadar katmanlar birbiri ardına eklenir. Birinci katın öğrenilmiş özellikleri ikinci kat için girdi olarak alınır ve ikinci tabaka aynı şekilde eğitilir.

Günümüzde, derin inanç ağları, denetlenmemiş veya üretken öğrenme algoritmalarına kıyasla, daha az sıklıkta kullanılmaktadır, ancak yine de derin öğrenme kapsamı içerisinde tanımlanmaktadır.



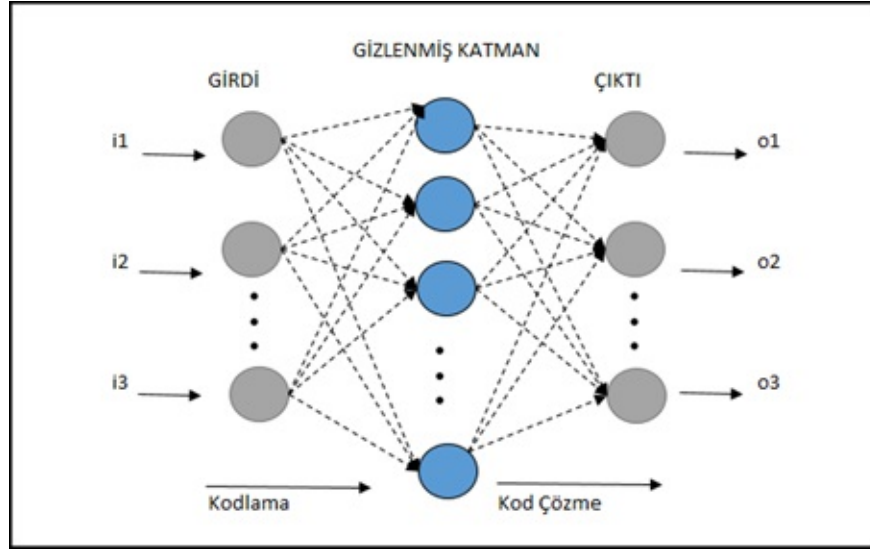
Şekil 9.2: Üç Sınırlı Boltzmann Makinasından (RBM) oluşan Derin İnanç Ağları

Şekil 9.2’de verilen geleneksel bir DBN mimarisini, anlatmak gerekirse, giriş verilerini ve birden çok gizli katmanı olan bir yapıdan oluşur. DBN’nin her bir tabakası, RBM’nin eğitim prosedürüne göre eğitilmiştir. RBM1 eğitimi tamamlandıktan sonra modele RBM2’nin gizli birimleri eklenir. RBM1’in gizli aktivasyonları RBM2’ye görünür bir RBM2 tabakası olarak beslenir ve bu işlem RBM3 için de tekrarlanır.

9.2.3 Derin Oto-Kodlayıcılar

Oto-kodlayıcılar denetimsiz öğrenim için kullanılabilir basit sinir ağları olarak tanımlanmıştır (Rumelhart, Hinton, & Williams, 2013). En önemli özelliği çıkış değerlerini giriş değerlerine eşit olarak ayarlamaya çalışmasıdır³. Çıkış nöronlarının sayısı girişlerle aynıdır, ancak gizli nöronların sayısı farklıdır. Böylece giriş verileri farklı sayıda birim ile ifade edilir. Bu gizli nöronların ağırlık değerleri, geri yayılma algoritması kullanılarak belirlenmektedir. Örnek bir oto kodlayıcı, Şekil 9.3’te gösterilmiştir.

Bir oto-kodlayıcı, aslında kendi girdilerini yeniden yapılandırmayı amaçlayan yapay bir sinir ağı modelidir. Bu nedenle, giriş etkili kodlama yeteneğine sahip olduğu zaman ya da gizli birimlerin sayısı, birtakım giriş birimlerinden daha az olduğunda, boyutsallığın azaltılması durumları için de kullanılmaktadır.



Şekil 9.3: Üç Sınırlı Boltzmann Makinasından (RBM) oluşan Derin İnanç Ağları

9.3 Derin Öğrenme Uygulama Yapısı

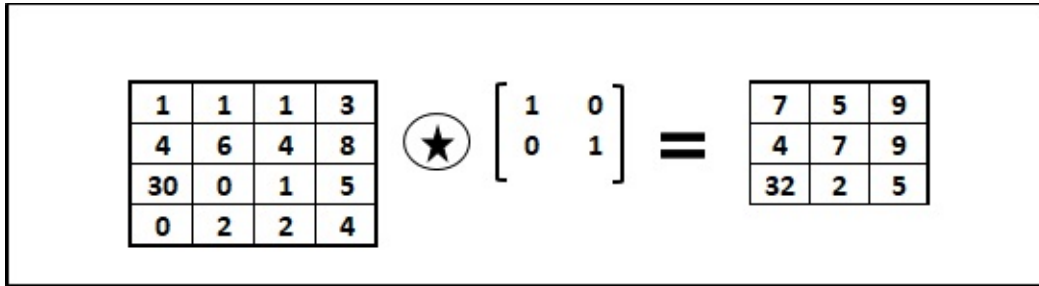
Derin sinir ağlarında ağın doğru çalışması ve yapay sinir ağlarında görülen ezberleme gibi istenmeyen durumlardan kaçınılması amacıyla yapay sinir ağlarında kullanılan tekniklerin yanında bazı yeni yöntemler geliştirilmiştir. Geliştirilen yöntemler derin ağların dinamik çalışmasını ve katmanların farklı işlere yoğunlaşmasını sağlamaktadır. Derin ağların bu dinamik yapısı her veriyi aynı model ile analiz etmeye çalışan yapay sinir ağlarından ayıran en belirgin özelliğidir. Bu sebeple, derin ağlara özel geliştirilen yöntemler ağın tüm katmanlarını aynı anda çalıştırmak yerine analiz edilecek veriye özgü katmanları ön plana çıkararak çalıştırmayı hedeflemektedir. Derin ağların uygulama yapısı bahsedilen dinamik yapı çerçevesinde şekillendirilmektedir.

Derin öğrenme sistemleri yapay sinir ağının özelleşmiş bir çeşidi olduğu için yapay sinir ağlarında görülen klasik bazı problemler bu sistemlerde de görülmektedir. Karşılaşılan en yaygın problemlerden biri ezberleme (overfitting) problemidir. Veri setinin eğitim süreci sırasında eğitim adımları ilerledikçe belirli bir aşamadan sonra hatanın sifıra yaklaşması sebebiyle ezberleme adı verilen bu durum ortaya çıkmaktadır. Karşılaşılan diğer bir sorun ise eksik öğrenme (underfitting) durumudur. Bu sorun yetersiz eğitim koşullarından kaynaklanmaktadır.

Derin sinir ağının tasarımı da yapay sinir ağı tasarımına benzemektedir. Yapay sinir ağında olduğu gibi gizli katmanlar, aktivasyon fonksiyonları, öğrenme algoritmaları, kayıp (loss) fonksiyonları, öğrenme katsayısı gibi üst değişkenler (hyperparameters) kullanılmaktadır. Bu üst değişkenlere bütünleştirme (pooling), kıvrım (convolution), otomatik kodlayıcı (auto-encoder) katmanları ve düğüm silme (dropout) işlemi gibi üst değişkenler eklenerek derin sinir ağını oluşturmaktadır. Derin ağların eğitiminde ve uygulamasında yeni üst değişkenler kullanılsa da derin ağlar ile yapay sinir ağlarının eğitim süreci benzerdir. Eğitim sırasında ezberlemenin ve eksik öğrenme sorunlarının önlenmesi için benzer çalışmalar yapılmaktadır. Bu bölümde eğitim sürecindeki çalışmalar ve üst değişkenler dört ana başlık altında incelenmiştir.

9.3.1 Kıvrım (Convolution)

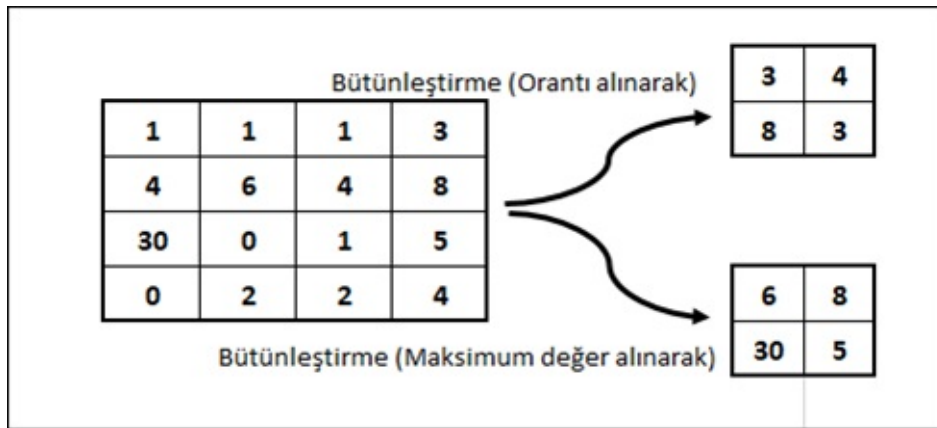
Kıvrım katmanı, özellik haritaları adı verilen yeni görüntüler oluşturur (Lecun, Bengio, & Hinton, 2015). Özellik haritası, orijinal görüntünün benzersiz özelliklerini vurgular. Kıvrım tabakası diğer sinir ağ katmanlarına göre çok farklı bir şekilde çalışmaktadır. Bu katman, bağlantı ağırlıklarını ya da herhangi bir ağırlıklı toplamı kullanmaz. Bunun yerine, görüntüleri dönüştüren filtreler içerir. Bu filtreler kıvrım filtreleri olarak adlandırılmaktadır. Görüntünün kıvrım tabakası aracılığıyla özellik haritası çıkarılır. Şekil 9.4'te kıvrım katmanının nasıl çalıştığı gösteren temsili bir matris işlemi anlatılmaktadır.



Şekil 9.4: Kıvrım katmanının nasıl çalıştığına dair bir örnek (Kim, 2017)

9.3.2 Bütünleştirme (Pooling)

Bütünleştirme işlemi, önemli bilgilerin korunarak girdinin uzamsal boyutlarını azaltmak için kullanılır. Boyutları azaltmak amacıyla kullanılan bu bütünleştirici katman, kıvrım katmanının çıkışında bulunmaktadır. Bütünleştirme katmanındaki işlemleri yürütmek için, görüntüden piksellerin nasıl seçileceğini ve temsil değerinin nasıl ayarlanacağını öncelikle belirlemeliyiz. Komşu pikseller genellikle kare matrisinden seçilir ve bütünleştirilen piksel sayısı probleme bağlı olarak değişir. Aşağıda verilen örnekte maksimum bütünleştirme ve ortalama bütünleştirme örnekleri verilmiştir.

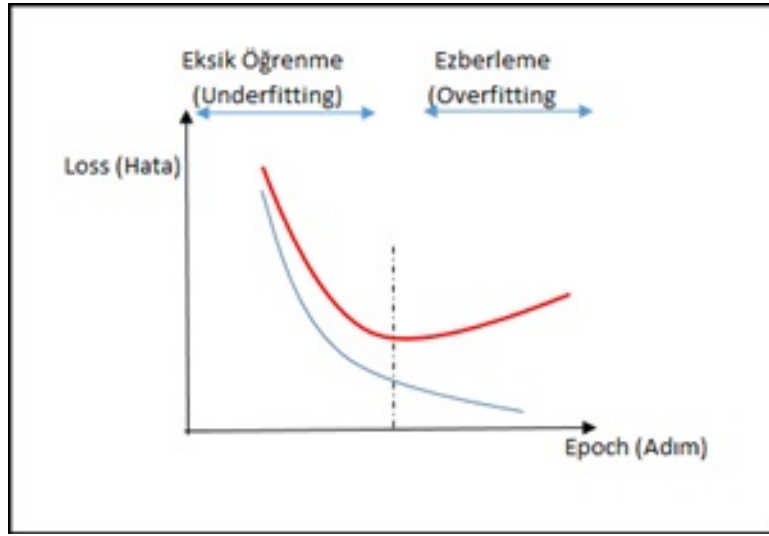


Şekil 9.5: İki farklı yöntem ile bütünleştirme işlemi (Kim, 2017)

9.3.3 Erken Durdurma (Early Stopping)

Veri setinin eğitimi sırasında genellikle eğitim hatasının zaman içinde sürekli olarak azaldığını gözlemlenmektedir, ancak doğrulama seti hatası tekrar yükselmeye başlar. Bu duruma ezberlenme denildiği yukarıda belirtilmiştir. Bu davranış Şekil 9.6'da gösterilmiştir.

Eğitim dönemlerinin çok az olması durumunda eğitim verilerinden öğrenilecek verilerin tam öğrenilmemesine sebep olurken, çok fazla dönem kullanılmasında ise gürültü verilerinin de eklendiği görülmektedir. Bu durumda erken durdurma yöntemi ile dönem sayısını belirlenerek ezberleme durumu engellenmiş olur. Erken durdurma tekniği derin öğrenmede en yaygın olarak kullanılan düzenleme şeklidir. Popülarlığı hem etkinliği hem de sadeliğinden kaynaklanmaktadır. (Goodfellow, Bengio, & Courville, 2017)



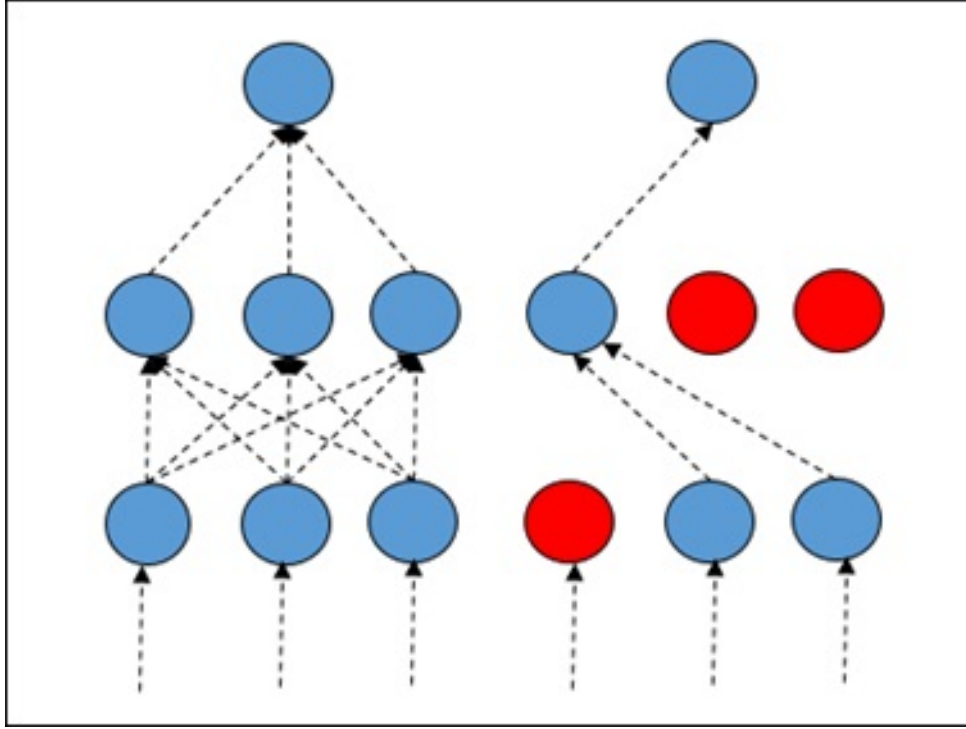
Şekil 9.6: Ezberleme ve eksik öğrenme durumu

9.3.4 Düğüm Silme (Dropout)

Düğüm silme yöntemi, derin sinir ağlarında ezberleme durumunu önlemek için en çok tercih edilen yöntemlerden biri haline gelmiş çok farklı bir yöntemdir. Düğüm silme katmanları, ağ düğümlerinin bir kısmını ve bu düğümlere ilgili giriş ve çıkış bağlantılarını dondurur. Bu sayede operasyon, onları ezberleme olma ihtimali olan herhangi bir düğüm veya özelliğe sahip olma olasılığını ortadan kaldırır. Eğitim sırasında bırakma, sadece p olasılıkla (bir hiperparametre) olan bir nöronun aktif tutulması ya da başka bir şekilde sıfırlanması ile gerçekleştirilir. Düğüm silme yöntemi ağı nöronların herhangi birine (veya herhangi bir küçük kombinasyonuna) çok bağımlı hale gelmesini önler. Bu yöntem ile birçok farklı sinir ağı mimarisinde verimli bir şekilde birleştirme yolu sağlanarak ezberleme sorununu önlenmektedir. Düğüm silme süreci Şekil 9.7'de resimli olarak ifade edilir.

9.4 Kalite Kontrol Uygulaması

Kalite kontrol işlemi ürünün standartlara uygunluğunun tespiti olduğu için bir sınıflandırma problemidir. Fakat kontrol işleminden önce ölçüm yapılması gerektiği için problem karmaşık hale gelmektedir. Çünkü kontrol edilecek üründe farklılığın tespit edilmesi, miktarının anlaşılması, farklılığın niteliğinin belirlenmesi ve elde edilen bulgulara göre karar verilmesi gerekmektedir. Bu durum



Şekil 9.7: Düğüm Silme seti (Buduma & Locascio, 2015)

kalite süreçlerinde yapay zeka ihtiyacını ortaya çıkarmaktadır. Yapay zeka hem ölçüm sürecinde hem de karar sürecinde rol alabileceği gibi çeşitli sensörlerden alınan ölçüm verilerindeki gürültünün giderilmesinde de kullanılmakta ve ölçüm sonuçlarının standartlaşmasını sağlamaktadır.

Örneğin çalışan bir motor titreşime neden olacaktır. Motor üzerindeki bir noktanın sıcaklığı ölçülmek isteniyorsa titreşim faktörünün ölçüm sürecinde ele alınması gerekebilir. Sıcaklık ölçümü kızılötesi bir sensörle yapıldığında, sürekli aynı noktayı ölçmek imkansız olacaktır. Temaslı bir sensörle yapıldığında da sensörün montaj noktalarında titreşim kaynaklı gevşemeler olabilir. Gevşeme sebebiyle thermocouple yüzeyi motora ideal bir temas sağlamayabilir. Eğer hassas ölçümlere ihtiyaç duyuluyorsa, ölçülen gerçek değer belirlenmesinde ortamdaki tüm şartları içeren bir model kurmak ya da yapay zeka kullanmak gerekmektedir. Örnekten anlaşılacağı üzere en basit ölçümler dahi ortam şartları hesaba katıldığında karmaşık hale gelebilmektedir. Bu nedenle kalite kontrol işlemlerinde yapay zeka kullanımı büyük avantajlar sağlamaktadır.

Yapay zeka sadece verilerdeki gürültülerin giderilmesinde değil, sınıflandırma işlemlerinde de oldukça başarılı sonuçlar üretmektedir. Bu yüzden yüzey kusurlarının tespitinde yapay zeka kullanımı hem kontrol sürecinin esnekliği (ürün çeşitliliğine adapte olmak) için, hem de karar verme sürecinin etkinliği açısından önemlidir.

9.4.1 Derin Öğrenme Modeli

Yapay sinir ağlarının bir çeşidi olan derin öğrenme sistemleri yapay sinir ağlarına benzer biçimde tasarlanmaktadır. Derin öğrenme sistemlerinde, sistemin dinamik ve daha doğru çalışmasını sağlayan yenilikler de bulunmaktadır. Kurgulanan kalite kontrol modeli de yapay sinir ağlarına ait bileşenleri ve derin öğrenme sistemlerine ait yenilikleri içermektedir. Kurgulanan modelde giriş kat-

manı, birden fazla gizli katman ve çıkış katmanları yer almaktadır. Kontrolü yapacak ürün görüntüsü sayısal hale getirilmeli ve normalize edilerek girdi katmanına iletilmektedir. Normalizasyon işlemi tüm sayısal değerlerin 1 ile 0 veya 1 ile -1 aralığına ölçeklendirilmesidir. Normalizasyon işleminin gerekliliği katmanların çalışması esnasında güçlü ilişkilerin ön plana çıkarılması için katmana gelen değerlere aktivasyon fonksiyonunun uygulanmasından kaynaklanmaktadır. Aktivasyon fonksiyonları katmana gelen sayısal değerleri güçlendiren veya zayıflayan, 1 ile 0 veya 1 ile -1 arasında değerler üreten fonksiyonlardır. Çeşitli aktivasyon fonksiyonları bulunmaktadır, fakat en yaygın kullanılanları sigmoid, tanjant hiperbolik, relu ve softmax fonksiyonlarıdır. Bu fonksiyonların ve katmanların çalışma prensibinin anlaşılması için Şekil 9.8'de verilen örnek görüntü verileri normalize edilerek aktivasyon fonksiyonları uygulanmıştır. Sayısallaştırma işlemi için resmin RGB (kırmızı, yeşil ve mavi) değerlerinin ortalaması hesaplanarak resim siyah-beyaz hale dönüştürülmüştür. Çünkü kırmızı değeri 0 ile 255 arasında bir değer, yeşil değeri 0 ile 255 arasında başka bir değer, mavi değeri de kırmızı ve yeşilden bağımsız 0 ile 255 arasında ayrı bir değerdir. Resim siyah-beyaz hale dönüştürülünce her piksel (nokta) için 0 ile 255 arasında bir değer elde edilmiştir. Bu değerlerin normalize edilmesi için tüm sayısal piksel değerleri 255'e bölünmüştür. Bu sayede değerler 0 ile 1 arasında ölçeklendirilmiştir.

60	45	22	89	37	0,24	0,18	0,09	0,35	0,15
25	3	46	75	21	0,1	0,01	0,18	0,29	0,08
8	92	36	7	1	0,03	0,36	0,14	0,03	0
66	88	13	20	12	0,26	0,35	0,05	0,08	0,05
15	40	10	6	11	0,06	0,16	0,04	0,02	0,04

Şekil 9.8: Girdi ve normalize edilmiş girdi değerleri

Girdi katmanında sigmoid aktivasyon fonksiyonu uygulandığında değerler Şekil 9.9'deki gibi olmaktadır. Sigmoid aktivasyon fonksiyonu -6 ile 6 aralığında çalışsa da mevcut girdiyi 0 ile 1 arasında ölçeklendirerek elde edilecek değerlerle, -6 ile 6 arasında ölçeklendirilerek elde edilecek değerlerin sapmaları birbirinden önemli düzeyde farklı olmayacaktır.

0,56	0,54	0,52	0,59	0,54
0,52	0,5	0,54	0,57	0,52
0,51	0,59	0,53	0,51	0,5
0,56	0,59	0,51	0,52	0,51
0,51	0,54	0,51	0,5	0,51

Şekil 9.9: Girdi değerlerine sigmoid aktivasyon fonksiyonunun uygulanması

Bu değerler kıvrım katmanında relu ya da softmax aktivasyon fonksiyonundan geçirilmektedir. Relu fonksiyonunda 0'dan küçük değerler 0'a eşitlenmekte, 0'dan büyük değerler ise aynen kullanılmaktadır. Şekil 9.9'da verilen girdi değerlerine relu uygulandığında sonuç olarak aynı matris elde edilecektir. Softmax fonksiyonu ise değerler arasındaki küçük farklılıkları giderek girdi değerlerini yakınlaştırmaktadır. Girdi değerlerine softmax fonksiyonu uygulandığında Şekil 9.10'da verilen

matris elde edilmektedir.

0,041	0,04	0,04	0,042	0,04
0,04	0,039	0,04	0,042	0,04
0,039	0,042	0,04	0,039	0,039
0,041	0,042	0,039	0,04	0,039
0,039	0,04	0,039	0,039	0,039

Şekil 9.10: Girdi değerlerine sigmoid aktivasyon fonksiyonunun uygulanması

Kıvrım katmanında elde edilen değerler bütünleştirme katmanında sadeleştirilecek ve başka bir kıvrım katmanında da sadeleştirilen değerler öğrenilecektir. Bütünleştirme işlemi gerçekleştirilirken bütünleştirme matrisine göre bütünleştirme stratejisi uygulanır. Bütünleştirme matrisi gelen girdi değerlerinden hangilerinin seçileceğini ve verinin ne kadar küçültüleceğini belirlemektedir. Şekil 11’de iki farklı 3x3’lük bütünleştirme matrisi yer almaktadır. Uygulamada genellikle tüm satır ve sütunları 1 olan matris kullanılırken probleme özgün farklı matrisler de tercih edilebilir. Matrislerin boyutları da genellikle 2x2 ile 5x5 arasında olmaktadır.

1	1	1
1	1	1
1	1	1

1	0	0
0	1	0
0	0	1

Şekil 9.11: Bütünleştirme matrisi

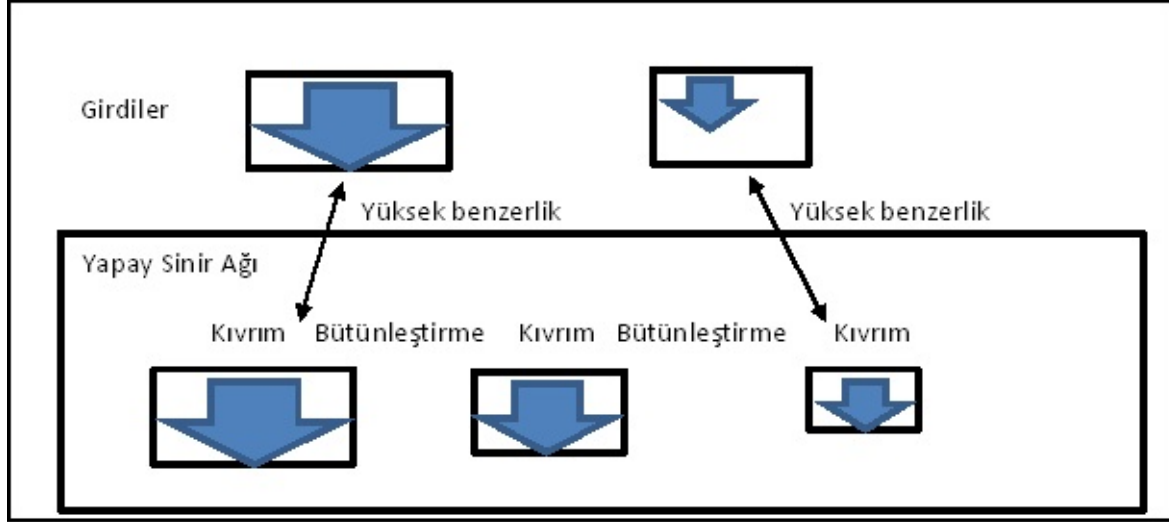
Bütünleştirme işlemi esnasında bütünleştirme matrisi girdi verisinin üzerinde birer satır ve birer sütun kaydırılarak gezdirilir. Girdi matrisinde, bütünleştirme matrisindeki 1 olan hücrelere karşılık gelen değerlere bütünleştirme stratejisi uygulanarak matriste küçülme sağlanır. Yaygın kullanılan bütünleştirme stratejileri max-pooling, min-pooling ve average-pooling stratejileridir. Max-pooling uygulandığında seçilen veriler içinden en büyük olanı, min-pooling uygulandığında seçilen verilerden en küçük olanı ve average-pooling uygulandığında seçilen değerlerin ortalaması sonuç matrisine yazılmaktadır. Şekil 9.10’da verilen girdilere Şekil 9.11’deki tüm değerleri 1 olan bütünleştirme matrisi average-pooling stratejisi ile uygulandığında Şekil 9.12’deki matris elde edilmektedir. Bütünleştirme işlemi esnasında girdi matrisinin ilk üç satır ve sütunundaki değerlerin ortalaması sonuç matrisin ilk satırı ve ilk sütununa yazılmıştır. Girdi matrisinin ikinci, üçüncü ve dördüncü sütunları ile ilk üç satırındaki değerlerinin ortalaması ise sonuç matrisinin ilk satırındaki ikinci sütuna yazılmıştır. Hesaplamalar bu şekilde devam ettirildiğinde 5x5’lik matris 3x3’lük matris haline gelmiştir.

0,04	0,04	0,04
0,04	0,04	0,04
0,04	0,04	0,039

Şekil 9.12: Bütünleştirme işleminin sonucu

Kıvrım ve bütünleştirme işlerinin sonucunda ağ resimdeki en büyük detaydan küçük detaylara kadar öğrenmekte ve kıyaslama yapabilmektedir. Kıvrım ve bütünleştirme işlemleri Şekil 9.13’teki

gibi ön plana çıkarılmak istenen özelliklerin bazen ilk kıvrım katmanında bazen de ilerideki kıvrım katmanlarında yakalanmasını sağlamaktadır. Bütünleştirme stratejisi katmanda ön plana çıkarılacak özelliklerin belirlenmesinde çok önemli rol oynamaktadır. Farklılıkların ortaya konması için genellikle average-pooling işlemi uygulanmaktadır. Fakat problemin içeriğine göre strateji belirlemek gerekmektedir. Örneğin sadece beyaz renkli bir ürün üzerindeki deformasyonlar anlaşılacak isteniyorsa, min-pooling iyi bir stratejidir. Çünkü beyaz rengin RGB değeri çok yüksektir. Min-pooling çizik ve lekelenme gibi renk değerlerini düşürecek durumların daha alt katmanlara aktarılmasını sağlayacak ve bir katmanda güçlü bir sinyal oluşmasına neden olacaktır.



Şekil 9.13: Kıvrım ve bütünleştirici katmanlarının çalışması

Geliştirilen uygulamada gizli katmanda başlangıç olarak üç adet kıvrım (konvolüsyon) ve bütünleştirici (pooling) katman kullanılmıştır. Kıvrımlı yapay sinir ağlarında başlangıç bu katmanlarla yapılmaktadır. Kıvrım ve bütünleştirici katmanları ürün resmi üzerindeki yüzey kusurlarının ön plana çıkarılıp ağa öğretilmesi için vazgeçilmez katmanlardır. Bu katmanların Tensorflow kütüphanesi kullanılarak oluşturulmasında kullanılan Python komutları Şekil 9.14’te verilmiştir.

```
import tensorflow as tf
girdi = tf.reshape(veri, [-1, en, boy, 1])
kivrim1 = tf.layers.conv2d(
    inputs= girdi, filters=fs, kernel_size=ks,
    padding="same", activation=tf.nn.relu)
butun1 = tf.layers.average_pooling2d(inputs=kivrim1, pool_size=[3, 3],
    strides=1)
```

Şekil 9.14: Kıvrım ve bütünleştirici katmanı oluşturma komutları

Komutların ilk satırında Tensorflow kütüphanesi tanıtılmaktadır. Sonraki satırda kullanılan “reshape” fonksiyonu ise veriyi en ve boy ölçülerinde birden fazla resim olarak ayırmayı sağlamaktadır. İlk şekil parametresinin -1 olması kullanılacak resim sayısının sistem tarafından algılanması istendiği içindir. Sonraki aşamada ise ilk kıvrım katmanı oluşturulmaktadır. “inputs” parametresi olarak girdi “reshape” ile düzenlenen veriler yer almaktadır. “filters” değişkeni relu ile kaç adet özellik çıkarılacağı belirlenmektedir. Eğer özellikler kare matris olarak ifade edilmişse özellik çıkarımında

relu uygulanacak matris boyutunu belirten “kernel” değişkeni tek değer olabilir. Aksi takdirde bir en, bir de boy verisini içeren iki elemanlı bir matris kullanılmalıdır. “padding” parametresinin “same” olması girdi veri boyutu ile çıktı veri boyutunun aynı olacağını gösterir. Varsayılan değeri ise “valid” olarak atanmıştır. Kırınım katmanı bitince bütünleştirme katmanı eklemek için max_pooling1d, max_pooling2d, max_pooling3d, average_pooling1d, average_pooling2d ve average_pooling3d fonksiyonları yer almaktadır. Tüm ağ tek boyutlu, iki boyutlu ya da üç boyutlu matrisler için çalıştırılabilmektedir.

Kırınım ve bütünleştirme katmanlarından sonra tam bağımlı katmanlarla ağın sonlandırılması gerekmektedir. Kırınım ve bütünleştirme işlemleri belirli konularda özelleşmiş katmanların oluşmasını sağlamaktadır. Bu katmanların çıktılarına göre düğüm silme (dropout), tahmin ve sınıflandırma yapacak “dense” katmanları eklenmelidir. Dense katmanları klasik sinir ağına benzer yapıda tek boyutlu olarak çalışmaktadırlar. Bundan dolayı “reshape” fonksiyonuyla veriler yeniden boyutlandırılmaktadır. “en_son” ve “boy_son” değişkenleri bütünleştirme işlemini sonucunda oluşacak küçük resmin eni ve boyu olmalıdır.

```
butunl_tekBoyut = tf.reshape(butunl, [-1, en_son * boy_son * fs])
dense = tf.layers.dense(inputs=butunl_tekBoyut, units=en*boy,
    activation=tf.nn.relu)
dugum_silme = tf.layers.dropout(inputs=dense, rate=0.1,
    training_mode=tf.estimator.ModeKeys.TRAIN)
cikti = tf.layers.dense(inputs=dugum_silme, units=cikti_sayisi)
```

Şekil 9.15: Dense katmanı oluşturma komutları

Düğüm silme ezberlemeyi engellemek için gerçekleştirilen bir işlemdir. Bu işlem ağın performansını artırmasına karşın, fazla düğüm silmek ağın hata yapmasına neden olacaktır. Bu yüzden düğüm silme oranı düşük tutulmalıdır. Düğüm silme işleminden sonra, elde edilen matris çıktı katmanına iletilecektir. Çıktı katmanı elde edilmek istenen sonucun detaylarına göre tasarlanmalıdır. Örneğin kalite kontrol işleminin sadece “Tamam” veya “Hatalı” olarak sonuçlanması isteniyorsa bir adet çıktı yeterli olacaktır. Farklı sonuçlar elde etmek için farklı sayıda çıktılar oluşturulmalıdır.

Katmanlar oluşturulduktan sonra Şekil 9.16’da verildiği gibi kayıp fonksiyonunu belirlemesi gerekmektedir. Kayıp fonksiyonunda yer alan “veri_sinifi” değişkeni sınıflandırma sonuçlarının yazılacağı tek boyutlu bir matristir. Veri sınıflandırma işlemlerinde kayıp fonksiyonu olarak “Softmax Cross Entropy” öğrenme algoritması olarak da “Gradient Descent Optimization” yaygın olarak kullanılan algoritmalarıdır. Modelin öğrenme katsayısı 0.05 olarak ayarlanmıştır. Bu değer düşürülmesi öğrenme süreci uzatır, artırılması ise yeterli öğrenememeye neden olmaktadır. Ağın eğitim modeli de belirlendikten sonra derin sinir ağı oluşturulmaktadır.

```
veri_sinifi = tf_y = tf.placeholder(tf.int32, [veri_sayisi])
kayip = tf.losses.sparse_softmax_cross_entropy(labels=veri_sinifi,
    logits=cikti)
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.05)
egitim_modeli = optimizer.minimize(loss=kayip,
    global_step=tf.train.get_global_step())
model = tf.estimator.EstimatorSpec(mode=mode, loss=kayip,
    train_op=egitim_modeli)
```

Şekil 9.16: Model eğitim komutları

9.4.2 Eğitim Süreci

Kalite kontrol için eğitim sürecinde hem doğru ürünün hem de Şekil 9.17’de görülen çeşitli deformasyon resimlerinin sisteme tanıtılması gerekmektedir. Öğrenmenin sağlıklı olması için çok sayıda veriye ihtiyaç vardır. Fakat resim içeriklerinin çevrilmesi veya simetriğinin oluşturulması verinin yeterli miktara ulaşmasını engellemektedir. Bundan dolayı doğru ürün ile deforme olmuş ürünler resim çıkarma (image subtraction) yöntemi ile karşılaştırılarak, deformasyonun olduğu noktalardaki değişim oranları belirlenmelidir. Daha sonra, belirlenen değişim oranları ürün resminin üzerinde rasgele noktalara uygulanabilir.



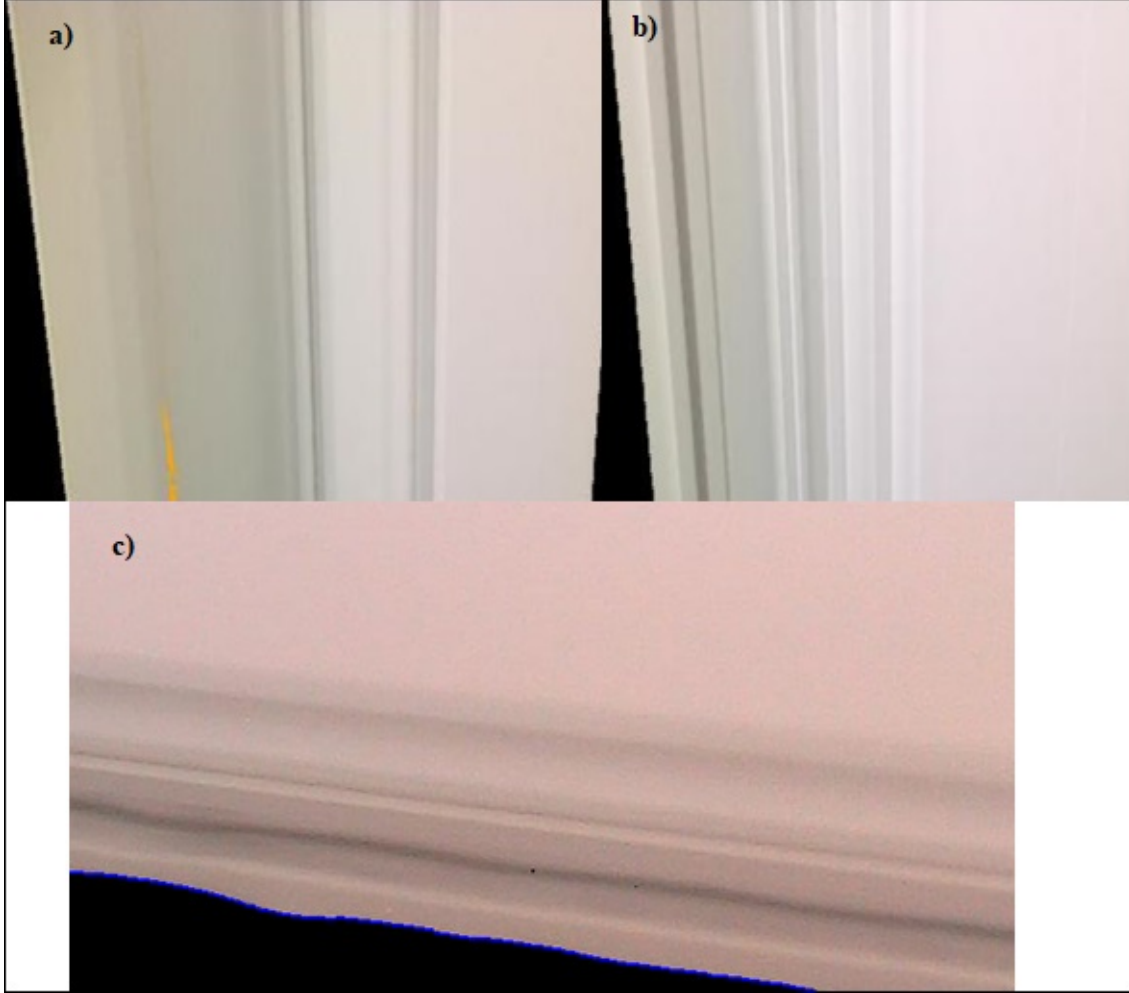
Şekil 9.17: Deformasyon resimleri

Sistem sağlam ve deforme olmuş ürünlerin resimleriyle eğitildikten sonra model “tf.train.Saver()” nesnesi ile daha sonra kullanılmak üzere kaydedilebilir. Eğitimi tamamlanan modelin hatalı olarak belirttiği pikseller Şekil 9.18’de görülmektedir.

9.5 Sonuç

Derin öğrenme, yapay sinir ağlarının dinamik çalışarak daha esnek ve karmaşık problemleri modellemede daha başarılı hale gelmesini sağlamıştır. Buna karşın derin öğrenme sistemlerinin eğitim sürecinde yüksek miktarda veriye ihtiyaç duyulmaktadır. Kalite kontrol süreci gibi bir süreçte sistemin ihtiyaç duyduğu miktarda deforme ürün verisi elde etmek kolay değildir. Çünkü ürünlere hasar vermenin maliyeti vardır. Üretim sisteminde oluşan hataların miktarı da bir derin öğrenme sistemini eğitmeye yetecek boyutta olmayabilir. Bu durumda veri üretme ihtiyacı ortaya çıkmaktadır. Fakat veri üretme işleminin rasgele olması yapay zekanın yanlış öğrenmesine ve hatalı çıktılar üretmesine neden olacaktır.

Piksel değerleri rastgele değiştirilemeyeceğinden, değiştirilecek piksellerdeki değişim miktarının deformasyona uygun olmasına çalışmak gerekmektedir. Bunun için öncelikle deforme olan noktaların belirlenmesi gerekmektedir. Belirlenen noktalardaki değişim oranı tespit edilip, resmin farklı noktalarına tespit edilen oranlar uygulanabilir. Tespitin oransal olarak yapılması gerekmektedir çünkü ışık dağılımı ve kamera açısı gibi parametreler resmin farklı noktalarında renk şiddetlerinin farklı olmasına neden olmaktadır. Oransal değişimin piksellere uygulanması deformasyon görüntüsünün daha doğru oluşmasını sağlayacaktır. Oluşturulan verilerle eğitilen derin sinir ağlarıyla yüzey problemleri tespit edilmiştir. Bu işlemin yapay zekayla gerçekleştirilmesi, ürüne bağlı parametrelerin ortadan kaldırılmasını sağlamıştır. Bu sayede, geliştirilen sistem her türlü ürünü öğrenip analiz edebilecek hale gelmiştir.



Şekil 9.18: a)Yüzey lekesi, b)Yüzey çizdiği, c)Kenar dalgalanması

9.6 Kaynakça

Buduma, N., & Locascio, N. (2015). Fundamentals of Deep Learning. Nature. <https://doi.org/10.1017/CB09781107415324.004>

Deng, L. (2014). Deep Learning: Methods and Applications. Foundations and Trends® in Signal Processing. <https://doi.org/10.1561/20000000039>

Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep Learning (Adaptive Computation and Machine Learning). The Mit Press. <https://doi.org/10.1520/JFS11687J>

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computation. <https://doi.org/10.1162/neco.2006.18.7.1527>

Kim, P. (2017). MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence. Library of Congress Control Number. <https://doi.org/10.1007/978-1-4842-2845-6>

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature. <https://doi.org/10.1038/nature14539>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE. <https://doi.org/10.1109/5.726791>

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234(November 2016), 11–26. <https://doi.org/10.1016/j.neucom.2016.12.038>

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2013). Learning Internal Representations by Error Propagation. In Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence. <https://doi.org/10.1016/B978-1-4832-1446-7.50035-2>



python powered

```
print("Hello, world!")
```

10. Python ile Görüntü İşlem

Python ile Görüntü İşleme ve Örnek Bir Uygulama

Gültekin ÇAĞIL¹, Recep Tayyip KARAYİĞİT¹

¹Sakarya Üniversitesi, Endüstri Mühendisliği

10.1 Giriş

1990'lı yıllardan günümüze bilişim dünyasında yaşanan hızlı gelişmeler dijital evrende bulunan veri miktarının hızla artmasına yol açmıştır. 2012 yılı rakamlarına göre dijital evrende 2.7 zetabyte(10^9 terrabyte) veri bulunmakta ve her geçen gün 2.5 exabyte(10^6 terrabyte) boyutunda veri bu evrene eklenmektedir[1]. EMC Bilgi Altyapısından sorumlu CEO David Goulden tarafından geçmiş yıllarda yapılan açıklamaya göre 2020 yılında tüm insanlığın ürettiği verinin toplam boyutunun 44 zetabyte değerine ulaşacağını öngörülmektedir[2]. Bu hızlı artış; bilgi yığını, bilgi kirliliği, istenilen bilginin doğrudan elde edilememesi gibi sorunları ortaya çıkarmış, ayrıca bu veri yığınlarındaki verilerin anlamlandırılarak bilgi haline dönüştürülmesi arayışlarını beraberinde getirmiştir. Bu konuda her geçen gün yeni çalışmalar yapılmakta ve bu çalışmalarda makine öğrenmesi teknikleri en çok kullanılan yöntemlerin başında gelmektedir.

Görüntü işleme, yapay zekânın endüstriyel alandaki ilk uygulama alanlarından biri olup dijital evrendeki verilerin bilgiye dönüştürme yollarından biridir. Tıbbi teşhis amacı ile radyolojik görüntülerin incelenmesi, basılı ya da el yazısı bir metnin bilgisayar ortamına aktarılması, bir görüntünün gerçekliğinin araştırılması, uydu görüntülerinin incelenmesi, görüntülerden nesne tanınması, görüntü ile olay ilişkilendirmesi gibi konular makine öğrenmesi teknikleriyle görüntü işleme üzerine çalışılan popüler başlıklardır.

Bu çalışmada görüntü işlemenin alt dallarından biri olan imge bölütleme'ye (Image Segmentation) ait birkaç teknikten bahsedilerek örnek bir uygulamaya yer verilecektir. Uygulamaya ait kodlama işlemleri, Python programlama dilinin 3.7 sürümü kullanılarak gerçekleştirilmiştir.

10.1.1 Temel Tanımlar

Bu bölümde makine öğrenmesi, görüntü işleme ve bilgisayar/makine görmesi başlıkları kısaca açıklanarak, bu başlıkların yakın ilişkileri nedeniyle ortaya çıkan anlam karışıklıkları giderilmeye çalışılacaktır.

Makine Öğrenmesi

Makine öğrenmesi; istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma, yapay zekâ, uyarlamalı denetim ve kuramsal bilgisayar bilimi gibi alanlarla yakın ilişki içerisinde olan, veri setlerinden öğrenmeyi ve anlamlı bilgiler edinmeyi mümkün kılan algoritmaların, tasarım ve geliştirme süreçlerini temel alan bir bilim dalıdır. Bu bilim dalının temel amacı bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır.

Görüntü İşleme

Görüntü işleme, cihazlardan elde edilen görüntülerin iyileştirilerek bilgisayarlar için uygun ve anlamlı hale getirildiği bir süreçtir. Bu konudaki çalışmalar 1960'lardan beri gündemde olmasına karşın görüntü işleme uygulamalarının güçlü bilgisayarlara ihtiyaç duyması araştırmaların hızlı ilerlemesine engel olmuştur. 50-60'lı yıllarda sadece ay'dan alınan görüntüler incelenebilmekteyken 90'lı yıllarda bilgisayar teknolojisinin de gelişmesiyle görüntü üzerindeki gürültü olarak adlandırılan lekelerden arındırma, ortamlar arasındaki farklılıkların algılatılması gibi temel işlemler yapılmaya başlanmıştır. Günümüzde ise bilgisayar sistemleri bu işlenen görüntülerden bir anlam çıkartıp, çıkardığı anlam doğrultusunda kararlar verebilmektedir.

Bu çalışmamızda görüntü kalitesi kötü olan, düşük ışıpta çekilmiş bir yazının görüntü işleme yöntemleriyle okunabilir bir hal alması sağlanmaya çalışılacaktır.

Bilgisayar/Makine Görmesi

Bilgisayar/Makine görmesi, bilgisayarların gerçek hayattaki nesnelere tanınması ve çevreden alınan verilerden elde edilen bilgilere göre aksiyon vermesini temel alan sistemlerdir. Bu sistemler günlük hayattaki insanın öngörebileceği durumları bilgisayar/makinenin öngörebilmesini amaçlar. Nesne tanımlama, metin okuma, araçlarda bulunan yaya/araç tanıma ve yol-şerit takibi, kalabalık ortamlar için tehdit algılama sistemleri bu alanda yapılan çalışmalara örnek olarak verilebilir.

Yukarıda açıklamış olduğumuz başlıkların birbirleriyle olan ilişkilerini özetlemek gerekirse: görüntü işleme işlemi, ham olarak alınan görüntünün makine öğrenmesi tekniklerinin uygulanabileceği, bilgisayar için anlamlı olabilecek bir görüntü formatına dönüştürülmesidir. Makine öğrenmesi teknikleri ile süreç anlaşılmasına çalışılır ve elde edilen çıktılar doğrultusunda bilgisayar/makine görmesi ile önceden kararlaştırılmış ya da sonradan öğrenilmiş aksiyon(lar) gerçekleştirilir. Örneğin; yolda ilerleyen bir araç ve karşıdan karşıya geçen bir yayayı düşündüğümüzde, aracın kamerası ile elde edilen görüntü çeşitli resim işleme teknikleriyle makine öğrenmesi algoritmalarının kullanıp anlam çıkarabileceği bir görüntü formatına getirilir. Makine öğrenmesi teknikleri sayesinde karşıdan geçen nesne tanımlanmaya çalışılır. Bilgisayar/makine görmesi ile araç ve yayanın bulunduğu pozisyonlar değerlendirilerek olası bir çarpışma ihtimali hesaplanır ve elde edilen sonuçlara göre gerekli olması durumunda aracın hızına müdahale edilir.

10.2 Python ile Görüntü İşleme

10.2.1 Python Nedir?

Python, Guido Van Rossum adlı Hollandalı bir programcı tarafından 90'lı yılların başında geliştirilmeye başlanan bir programlama dilidir. Guido Van Rossum bu programlama dilini, The Monty Python

adlı bir İngiliz komedi grubunun, Monty Python's Flying Circus adlı gösterisinden esinlenerek adlandırmıştır [3]. Her ne kadar ismini bir piton yılanından almamış olsa da, Python programlama dili, pek çok yerde bir yılan figürü ile temsil edildiği görülmektedir.

Python programlarının en büyük özelliklerinden birisi, C ve C++ gibi dillerin aksine, derlenmeye gerek olmadan çalıştırılabilirlerdir. Bu sayede program geliştirme süreci hızlanmaktadır. Ayrıca Python programlama dilinin basit ve temiz söz dizimi, onu pek çok programcı tarafından tercih edilen bir dil haline getirmiştir[4].

Python'ın diğer güçlü tarafı ise birçok farklı alanda açık kaynak kodlu olarak sunulan güçlü bir kütüphane desteğinin bulunmasıdır. Bu kütüphane desteği sayesinde tercih edilen yazılım dilleri arasında üst sıralarda yer almaktadır. Bu çalışmamızda Python'ın görüntü işleme alanında popüler olan kütüphanelerinden "OpenCV" ve grafik işlemleri için kullanılan "matplotlib" kütüphanelerinden faydalanılacaktır.

10.2.2 OpenCV

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında Intel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam ettirilmiştir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış, ilerleyen dönemlerde birçok algoritması C++ dili ile geliştirilmeye devam edilmiştir. Açık kaynak kodlu olan OpenCV kütüphanesinin, BSD (Berkeley Software Distribution) lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabilme olanağını sağlamaktadır. OpenCV kütüphanesi işletim sisteminden bağımsız olarak Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında sorunsuzca çalışabilmektedir. C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla da Visual Basic.Net, C# ve Visual C++ dilleri ile topluluklar tarafından geliştirilen farklı wrapper'lar aracılığıyla Perl ve Ruby programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir[5].

OpenCV kütüphanesi içerisinde görüntü işlemeye (Image Processing) ve makine öğrenmesine (Machine Learning) yönelik 2500'den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesnelere ayırt etme, insan hareketlerini tespit etme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapma, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir[6].

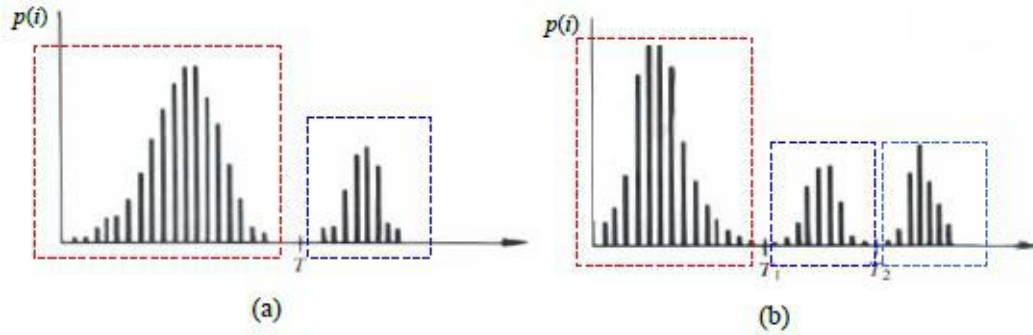
10.3 İmge Bölütleme (Image Segmentation) Yöntemleri

İmge bölütleme, bir görüntüden benzer nitelikteki pikselleri gruplara ayırma işlemidir. İmge bölütleme işlemini zor kılan tüm görüntülere uyabilecek genel bir yönteminin olmamasıdır. İstatistik biliminde kümeleme analizi olarak da bilinen imge bölütleme işlemi, geniş bir çalışma alanına sahip olup konuyla ilgili yüzlerce farklı algoritması bulunmaktadır[7]. Her görüntü farklı koşullarda kaydedildiği için algoritmaların başarı yüzdesi tespit edilmesi istenen nesneye ve fotoğrafın kaydedildiği çevresel koşullarına göre değişim göstermektedir. Bu nedenle mükemmel bir imge bölütleme yöntemi bulunmamaktadır.

Bu bölümde imge bölütleme yöntemlerinden biri olan Eşikleme (Thresholding) tekniklerinden bahsedilerek Python'un OpenCV kütüphanesine ait eşikleme metotlarından bazılarının nasıl kullanılacağı gösterilecektir.

10.3.1 Eşikleme (Thresholding)

Eşikleme, görüntü bölütleme amacı için kullanılan en önemli yaklaşımlardan birisidir. Eşikleme işleminde amaç, görüntü içerisindeki nesnelere görüntü arka planından ayırmaktır. Eşikleme işlemi için, görüntüdeki gri seviye dağılımlarını gösteren görüntü histogramından faydalanılır. Örneğin, koyu bir arka plan üzerinde açık renkli nesnelere oluşan $f(i, j)$ görüntüsüne ilişkin gri seviye histogramı Şekil 10.1 (a)'daki biçime sahip olacaktır. Bu histograma göre, nesnelere ve arka plana ait pikseller olmak üzere, görüntüyü iki ana grupta değerlendirmek mümkündür. Bu durumda nesnelere arka plandan ayırmak için en kolay yol, histogramdan göreceli olarak belirlenen bir T eşik değeri ile görüntüdeki piksel değerlerini karşılaştırmak olacaktır. Buna göre, görüntüdeki herhangi bir (i, j) pikseli için; $f(i, j) > T$ ise (i, j) pikseli nesneye ait bir nokta, $f(i, j) \leq T$ ise (i, j) pikseli arka plana ait bir nokta olacaktır. Diğer taraftan, görüntüye ilişkin histogram Şekil 10.1 (b)'deki gibi ikisi nesneye biri de arka plana ait olmak üzere üç gri seviye grubundan oluşabilir. Buna göre görüntüdeki herhangi bir (i, j) pikseli için; $T_1 < f(i, j) \leq T_2$ aralığındaki pikseller bir nesneye, $f(i, j) > T_2$ aralığındaki pikseller diğer bir nesneye ve $f(i, j) \leq T_1$ aralığındaki pikseller de görüntü arka planına karşı düşecektir.



Şekil 10.1: Tek bir eşik değeri ve birden çok (çoklu) eşik değeri ile bölmelenen gri seviye histogram biçimleri

Şekil 10.1 (b)'deki gibi verilen çoklu eşikleme işlemi genel olarak tek bir eşik değeri ile görüntüyü eşiklemekten daha az güvenilirdir. Bunun nedeni, birden fazla bölgeyi etkili bir biçimde bölütleyen eşik değerlerinin belirlenmesindeki zorluktur. Tipik olarak bu yapıdaki eşikleme problemleri, bölgesel olarak değişen eşik değerlerinin belirlenmesi ile çözülebilir. Bu anlamda en genel olarak eşik değeri, fonksiyon olarak

$$T = T[x, y, f(x, y), p(x, y)] \quad (10.1)$$

ile verilir. Burada $f(x, y)$, (x, y) noktasındaki gri seviye olup $p(x, y)$ ise bu noktanın bazı bölgesel özelliklerini belirtir – örneğin bu özellik (x, y) noktasının komşuluğundaki piksel gri seviye değerlerinin ortalaması olabilir. Eşiklenmiş bir $g(x, y)$ görüntüsü

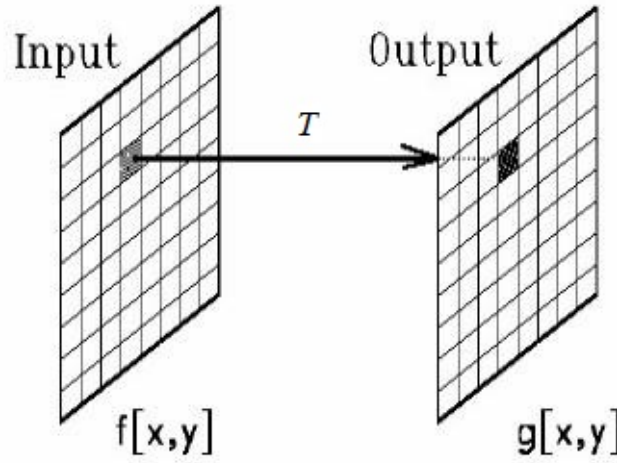
$$g(x, y) = \begin{cases} 1; & \text{eğer } f(x, y) > T \\ 0; & \text{eğer } f(x, y) < T \end{cases} \quad (10.2)$$

biçiminde tanımlanır. Buna göre, 2 numaralı fonksiyonun sonucunda 1 ile etiketlenen pikseller nesneye, 0 ile etiketlenenler ise arka plana karşı düşer. Eğer ki T sadece $f(x, y)$ 'ye bağlı ise bu

durumda 1 numaralı eşitlik ile belirlenen eşik değeri bütünsel (global) eşik olarak adlandırılır. Şekil 10.1 (a), böyle bir eşik değerine örnektir. Eğer T hem $f(x,y)$ hem de $p(x,y)$ 'ye bağlı ise bu durumda 1 numaralı eşiklik ile belirlenen eşik değeri bölgesel (local) eşik olarak adlandırılır. Buna ek olarak, Eğer T , x ve y uzaysal koordinatlarına bağlı ise bu durumda 1 numaralı eşitlik ile belirlenen eşik değeri dinamik eşik olarak adlandırılır[8].

Bütünsel (Global) Eşikleme

Bütünsel eşikleme, Şekil 10.1 (a)'da gösterildiği gibi tek bir eşik değeri kullanılarak nesnenin arka plandan ayrıştırılması işlemidir. Bu işlemin başarısı diğer bölütme yöntemlerinde olduğu gibi görüntünün kaydedildiği ortam koşullarına bağlı olarak gerçekleşir. Nesne ve arka plandaki ışık farkları gibi zıtlıklar piksellerin gruplandırılmasını kolaylaştıracağı için daha verimli gerçekleşmesini sağlayacaktır.



Şekil 10.2: Bir pikselin ($f(x,y)$) Eşik değerine göre yeni değer($g(x,y)$) atanma işlemi

Bütünsel eşikleme işlemi için Şekil 10.2'de görüldüğü gibi her bir pikselin değeri belirli bir sınır değerine göre yeniden hesaplanır. Teori kısmının ardından Python'un görüntü işleme kütüphanesi olan OpenCV kütüphanesinde bütünsel eşikleme işleminin nasıl gerçekleştirileceğini inceleyelim. OpenCV kütüphanesinde sıkça kullanılan threshold tipleri aşağıdaki gibidir:

- **THRESH_BINARY:** Kaynak görüntünün piksel değeri ($f(x,y)$) eşik değerinden (T) büyükse oluşturulacak olan hedef görüntüde karşılık gelen piksel değerine ($g(x,y)$) belirlenmiş olan maksimum değer (*max Value*), küçük ya da eşit olması durumunda sıfır atanır. THRESH_BINARY eşikleme tipinin kullanmış olduğu algoritma aşağıdaki gibidir:

$$\begin{aligned}
 & \text{If } f(x,y) > T : \\
 & \quad g(x,y) = \text{max Value} \\
 & \text{else :} \\
 & \quad g(x,y) = 0
 \end{aligned}
 \tag{10.3}$$

- **THRESH_BINARY_INV:** Kaynak görüntünün piksel değeri($f(x,y)$) sıfırdan büyükse oluşturulacak olan hedef görüntüde karşılık gelen piksel değerine($g(x,y)$) sıfır, küçük ya da eşit ol-

ması durumunda belirlenmiş olan maksimum değer(*max Value*) atanır. `THRESH_BINARY_INV` eşikleme tipinin kullanmış olduğu algoritma aşağıdaki gibidir:

$$\begin{aligned}
 & \text{If}(x,y) > 0 : \\
 & \quad g(x,y) = 0 \\
 & \text{else} : \\
 & \quad g(x,y) = \text{max Value}
 \end{aligned} \tag{10.4}$$

- **THRESH_TRUNC:** Kaynak görüntünün piksel değeri($f(x,y)$) sıfırdan büyükse oluşturulacak olan hedef görüntüde karşılık gelen piksel değerine($g(x,y)$) eşik değeri(T) atanırken, küçük ya da eşit olması durumunda kaynak görüntünün piksel değeri korunur. Python'da eşikleme metodu oluşturulurken maksimum değer parametresine bir değer girilir fakat bu değer bu eşikleme tipi için görüntü oluşturulmasında herhangi bir etkisi yoktur. `THRESH_TRUNC` eşikleme tipinin kullanmış olduğu algoritma aşağıdaki gibidir:

$$\begin{aligned}
 & \text{If}(x,y) > 0 : \\
 & \quad g(x,y) = T \\
 & \text{else} : \\
 & \quad g(x,y) = f(x,y)
 \end{aligned} \tag{10.5}$$

- **THRESH_TOZERO:** Kaynak görüntünün piksel değeri($f(x,y)$) sıfırdan büyükse oluşturulacak olan hedef görüntüde karşılık gelen piksel değerine($g(x,y)$) kaynak görüntünün piksel değeri, küçük ya da eşit olması durumunda sıfır atanır. Python'da eşikleme metodu oluşturulurken maksimum değer parametresine bir değer girilir fakat bu değer bu eşikleme tipi için görüntü oluşturulmasında herhangi bir etkisi yoktur. `THRESH_TOZERO` eşikleme tipinin kullanmış olduğu algoritma aşağıdaki gibidir:

$$\begin{aligned}
 & \text{If}(x,y) > 0 : \\
 & \quad g(x,y) = f(x,y) \\
 & \text{else} : \\
 & \quad g(x,y) = 0
 \end{aligned} \tag{10.6}$$

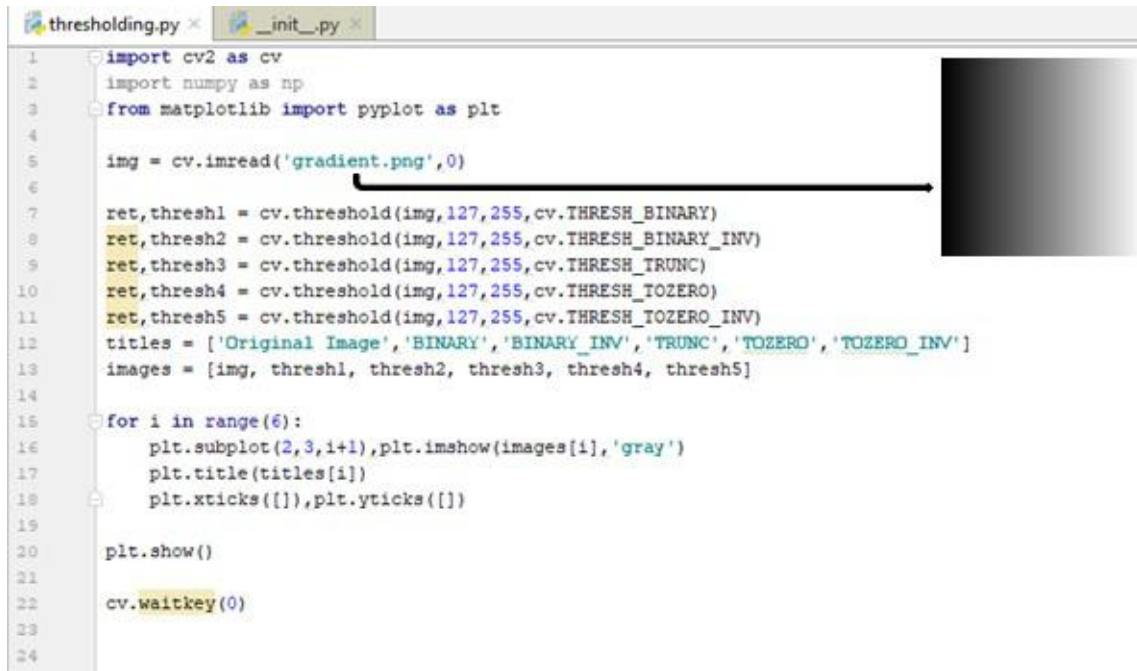
- **THRESH_TOZERO_INV:** Kaynak görüntünün piksel değeri ($f(x,y)$) sıfırdan büyükse oluşturulacak olan hedef görüntüde karşılık gelen piksel değerine ($g(x,y)$) sıfır, küçük ya da eşit olması durumunda kaynak görüntünün piksel değeri atanır. Python'da eşikleme metodu oluşturulurken maksimum değer parametresine bir değer girilir fakat bu değer bu eşikleme tipi için görüntü oluşturulmasında herhangi bir etkisi yoktur. `THRESH_TOZERO_INV` eşikleme tipinin kullanmış olduğu algoritma aşağıdaki gibidir:

$$\begin{aligned}
 & \text{If}(x,y) > 0 : \\
 & \quad g(x,y) = 0 \\
 & \text{else} : \\
 & \quad g(x,y) = f(x,y)
 \end{aligned} \tag{10.7}$$

Thresholding metotunun; kaynak görüntü, eşik değeri, maksimum değer ve threshold tipi olmak üzere 4 farklı giriş parametresi bulunmaktadır. Python programlama dilinde gerçekleştirilen örnek bir kod satırı aşağıda paylaşılmıştır.

$$\text{ret, thresh} = \text{cv2.threshold}(\text{img}, 127, 255, \text{cv2.THRESH_BINARY_INV}) \quad (10.8)$$

Eşikleme tiplerinin daha iyi anlaşılması için, 0-255 gri tonlama çizelgesini temsil eden bir görüntü (gradient.png) üzerinde farklı threshold tiplerinin uygulandığı örnek uygulama ait Python kodları (Şekil 10.3) ve ekran çıktısı (Şekil 10.4) aşağıda paylaşılmıştır. Kullanılan görüntüde en koyu (resmin en sol noktası) noktasının piksel değeri sıfır, orta noktasının 127 ve en parlak noktasının değeri 255'tir.



```

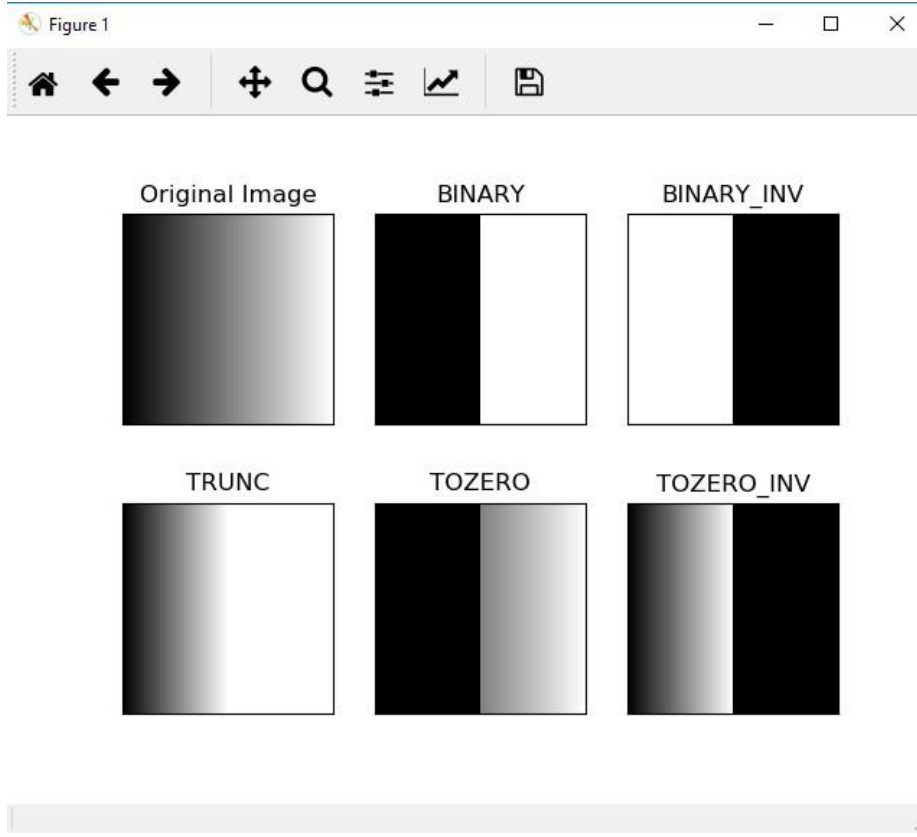
1  import cv2 as cv
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  img = cv.imread('gradient.png',0)
6
7  ret, thresh1 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)
8  ret, thresh2 = cv.threshold(img, 127, 255, cv.THRESH_BINARY_INV)
9  ret, thresh3 = cv.threshold(img, 127, 255, cv.THRESH_TRUNC)
10 ret, thresh4 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO)
11 ret, thresh5 = cv.threshold(img, 127, 255, cv.THRESH_TOZERO_INV)
12 titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
13 images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
14
15 for i in range(6):
16     plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
17     plt.title(titles[i])
18     plt.xticks([],plt.yticks([]))
19
20 plt.show()
21
22 cv.waitKey(0)
23
24

```

Şekil 10.3: Bütünsel eşikleme tiplerinin uygulandığı uygulamaya ait Python kodları

Bölgesel (Dinamik) Eşikleme

Bölgesel Eşikleme, görüntünün farklı bölgelerindeki eşik değerlerini belirli algoritmalar aracılığıyla hesaplayarak kaynak görüntü için en uygun eşik değerini bulmayı hedefleyen eşikleme türüdür. Bölgesel eşikleme işlemleri, OpenCV kütüphanesindeki adaptiveThreshold metodu çağırılarak gerçekleştirilebilir. Thresholding metotunun; kaynak görüntü, maksimum değer, adaptive (uyarlanmış) threshold tipi, komşu alan büyüklüğünü belirleyen blok boyutu ve ağırlıklı toplam ya da ortalama değerden çıkartılan bir sabit sayı olmak üzere toplamda 6 farklı giriş parametresi bulunmaktadır. Uygulama kısmında ADAPTIVE_THRESH_GAUSSIAN_C ve THRESH_MEAN_C Python programlama dilinde gerçekleştirilen örnek bir adaptiveThreshold metoduna ait kod satırı aşağıdaki



Şekil 10.4: Bütünsel eşikleme tiplerinin uygulandığı uygulamanın ekran çıktısı

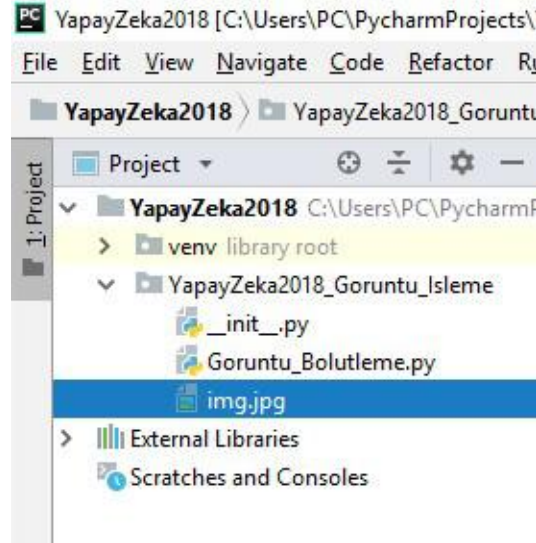
gibidir:

```
resim = cv.adaptiveThreshold(img,255,cv.ADAPTIVE_THRESH_GAUSSIAN_C,  
cv.THRESH_BINARY,11,2) (10.9)
```

Uygulama kısmında, görüntü eşik değerinin komşu alanların piksel değerlerinin ortalaması ve gaussian penceresinin ağırlık toplamlarına göre hesaplandığı THRESH_MEAN_C ve THRESH_GAUSSIAN_C uyarlanmış eşik tipleri incelenecektir.

10.4 Uygulama

Üzerinde çalışacağımız görüntü, kodlama işlemlerimizi gerçekleştirdiğimiz PyCharm editörünün aktif proje kısmında yüklü olmalıdır. Aksi durumda görüntü ile ilgili işlem yapılmak istendiğinde dosyanın - dış kaynak olarak adres - yolu işlem esnasında belirtilmelidir. PyCharm editöründe resim yükleme işlemi oldukça kolaydır. Çalışmak istediğiniz görüntü dosyası kopyaladıktan sonra ilgili Python Package dosyası (YapayZeka2018_Goruntu_Isleme) üzerindeyken sağ tıklama işleminin ardından yapıştır komutu ile kolayca gerçekleştirilebilir. Yapıştırma işleminin ardından görüntü dosyası Şekil 10.5'te olduğu gibi Python Package dosyasının (YapayZeka2018_Goruntu_Isleme) altında yer alacaktır.



Şekil 10.5: Python'da Projeye Resim Ekleme İşlemi

Görüntü işleme ve grafik çizme kütüphanelerini kullanabilmek için bu kütüphanelerin Python dosyasında aktif hale getirilmesi gerekmektedir. Kütüphanelerin aktif hale getirilmesi "import" komutuyla sağlanmaktadır. Bu kütüphaneler daha sonraki adımlarda "as" ifadesinin ardından belirtilen isimlerle çağırılacaktır. Aşağıda gerekli kütüphanelerin kullanılabilir hale getirilmesi için örnek kodlama işlemi gösterilmiştir.

```
import cv2 as cv  
from matplotlib import pyplot as plt (10.10)
```

Img.jpg ismindeki görüntü, OpenCV içerisindeki `.imread` metodundan faydalanarak img değişkenine atılmaktadır. Burada sonraki adımlarda resme thresholding işlemleri uygulanacağı için, görüntü metoda girilen ikinci parametresine sıfır değeri girilerek siyah-beyaz formata getirilir. `Imread` metodunun kullanımı aşağıdaki şekilde gerçekleştirilebilir.

```
img=cv.imread('img.jpg',0) (10.11)
```

Kullanılacak olan görüntü siyah-beyaz formata getirilmesinin ardından görüntüye uygulanacak olan eşikleme tipleri için eşik değerleri `T` ve `MaxValue` değişkenlerine atılır. `Threshold` metoduna ait örnek kodlama işlemi aşağıdaki olduğu gibi gerçekleştirilebilir. Uygulamamızda eşik değeri (`T`) 5 ve üst sınır değeri (`MaxValue`) 255 olarak belirlenmiştir.

```
T=5
MaxValue=255
ret,thresh1=cv.threshold(img,T,MaxValue,cv.THRESH_BINARY)
ret,thresh2=cv.threshold(img,T,MaxValue,cv.THRESH_BINARY_INV) (10.12)
ret,thresh3=cv.threshold(img,T,MaxValue,cv.THRESH_TRUNC)
ret,thresh4=cv.threshold(img,T,MaxValue,cv.THRESH_TOZERO)
ret,thresh5=cv.threshold(img,T,MaxValue,cv.THRESH_TOZERO_INV)
```

Uygulanan `threshold` tiplerinin tek bir tabloda görülebilmesi için döngü işlemlerinde kullanılması amacıyla aşağıdaki gibi başlık isimlerinin ve resimlerin tutulduğu listeler oluşturulur.

```
basliklar=['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images=[img,thresh1,thresh2,thresh3,thresh4,thresh5] (10.13)
```

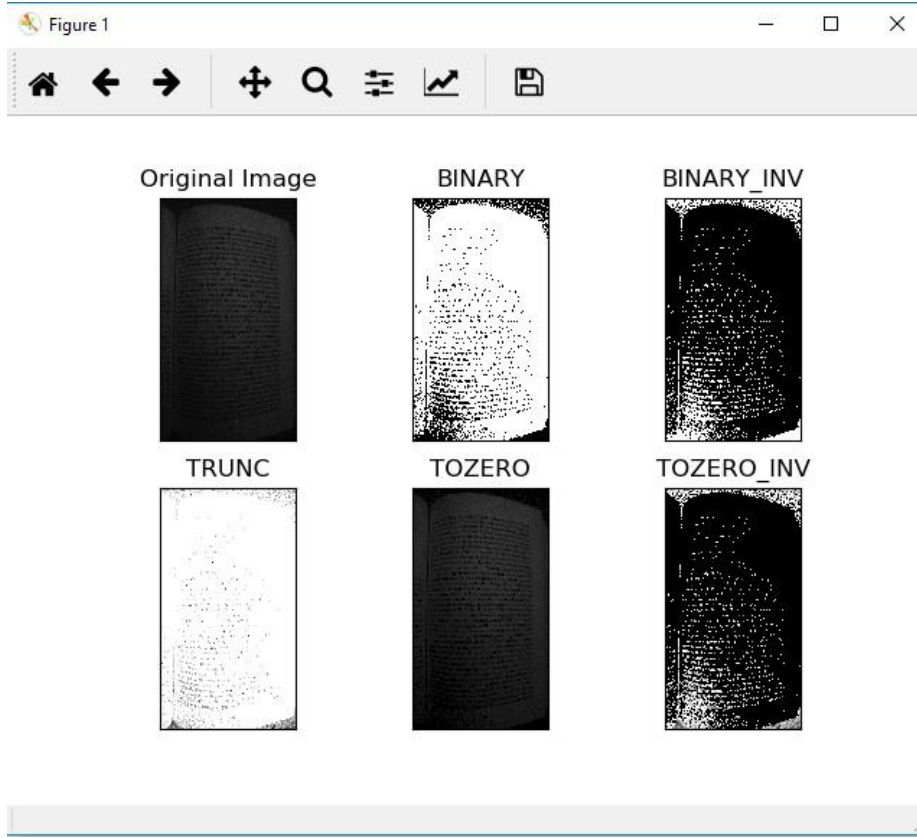
Bu listeler aşağıdaki gibi `for` döngü işlemlerinde kullanılarak tüm `thresh` tiplerinin bir arada gözlemlenebileceği bir tablo oluşturulur. Ardından `.show` metodu ile oluşturulmuş olan tablo ekrana yansıtılır.

```
for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],gray') (10.14)
    plt.title(basliklar[i])
    plt.xticks([],plt.yticks([]))
```

Buraya kadar yapılan işlemlerin çıktısı Şekil 10.6 daki gibidir:

Görüntüleri biraz daha yakından ve ayrı pencerelerden incelemek istersek `for` döngüsüne aşağıdaki gibi değiştirebilirsiniz.

```
for i in range(6):
    cv.imshow(basliklar[i],images[i]) (10.15)
```



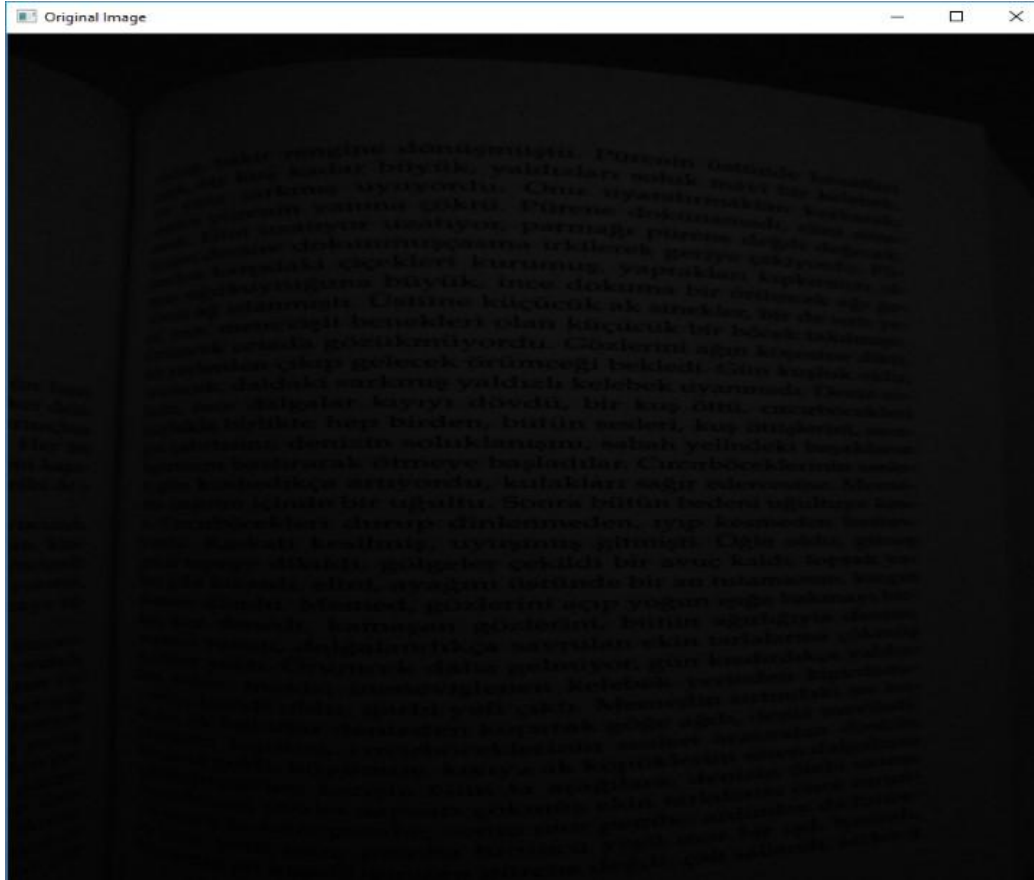
Şekil 10.6: $T = 5$ eşik seviyesinde farklı eşikleme metotlarının uygulama çıktısı

Elde edilen görüntülerin gözlemlenebilmesi için programı bir süreliğine bekletecek `waitKey` ve bekleme işlemin ardından açık pencereleri kapatarak Ram'de boş yer kaplamayı önleyen `destroyAllWindows` metodları çağırılmalıdır. Aksi halde kodlama işlemi hatasız bir şekilde çalışırken çıktıları görmemiz mümkün olmayacaktır. Bu iki metodun kullanımı aşağıdaki gibi gerçekleştirilebilir.

```
cv.waitKey(0)
cv.destroyAllWindows()
```

(10.16)

Uygulamanın ayrı pencerelerde oluşturulmuş sonuçları aşağıdaki gibidir:

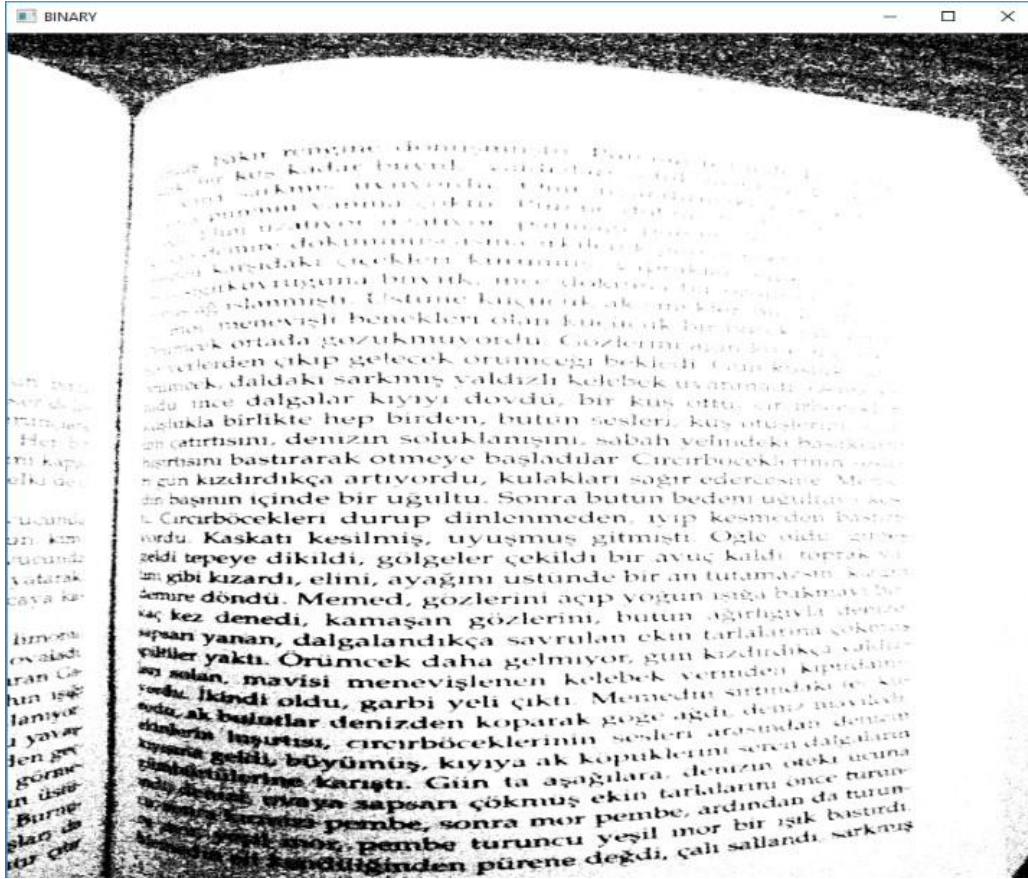


Şekil 10.7: Uygulamada kullanılan düşük ışıktaki kaydedilmiş görüntü (img.jpg)

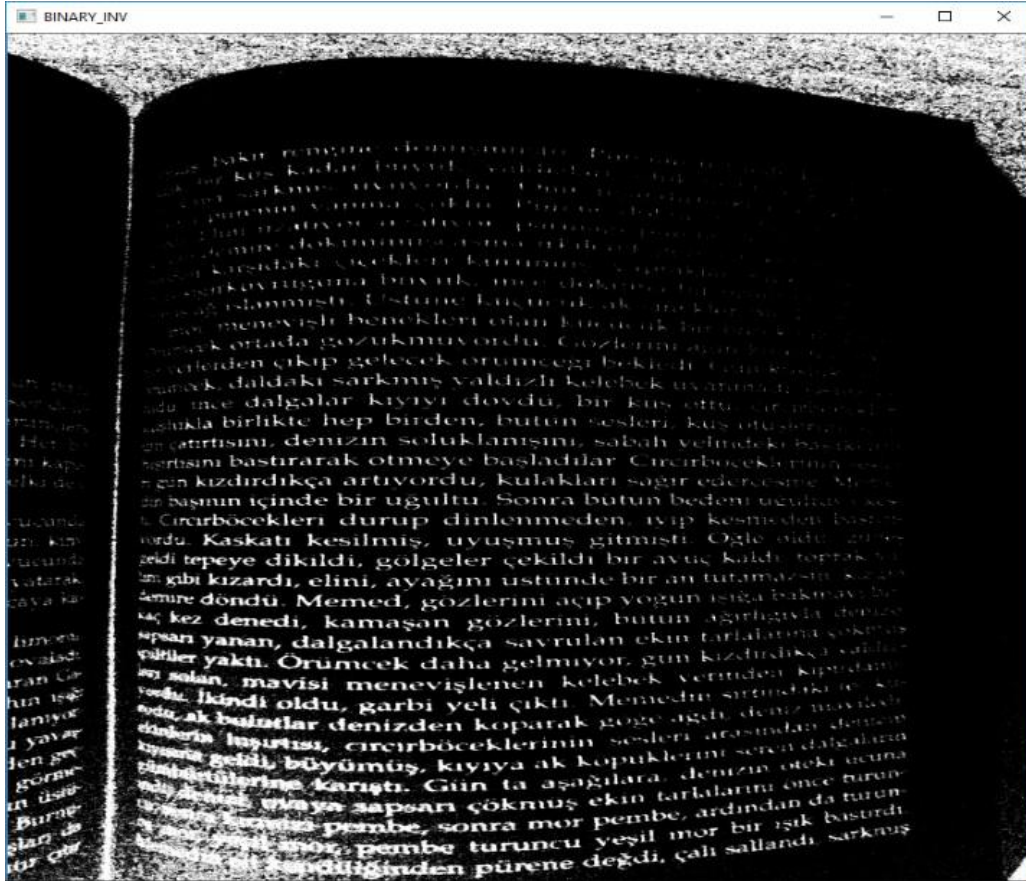
Elde edilen görseller arasında en iyi görüntünün `THRESH_BINARY` ile elde edildiği görülmektedir. Fakat bu durum daha önceden belirtmiş olduğumuz gibi üzerinde çalışmış görsellere, ışığın düştüğü yüzeye, görüntü kalitesine, ışık yoğunluğuna gibi birçok çevresel faktöre farklılık göstermektedir. Farklı koşullarda farklı threshold tipi daha iyi bir sonuç sağlayabilir.

Aşağıda `THRESH_BINARY` tipindeki $T = 5$ (Şekil 10.13) ve $T = 10$ (Şekil 10.14) eşik değerleri için uygulama çıktıları paylaşılmıştır.

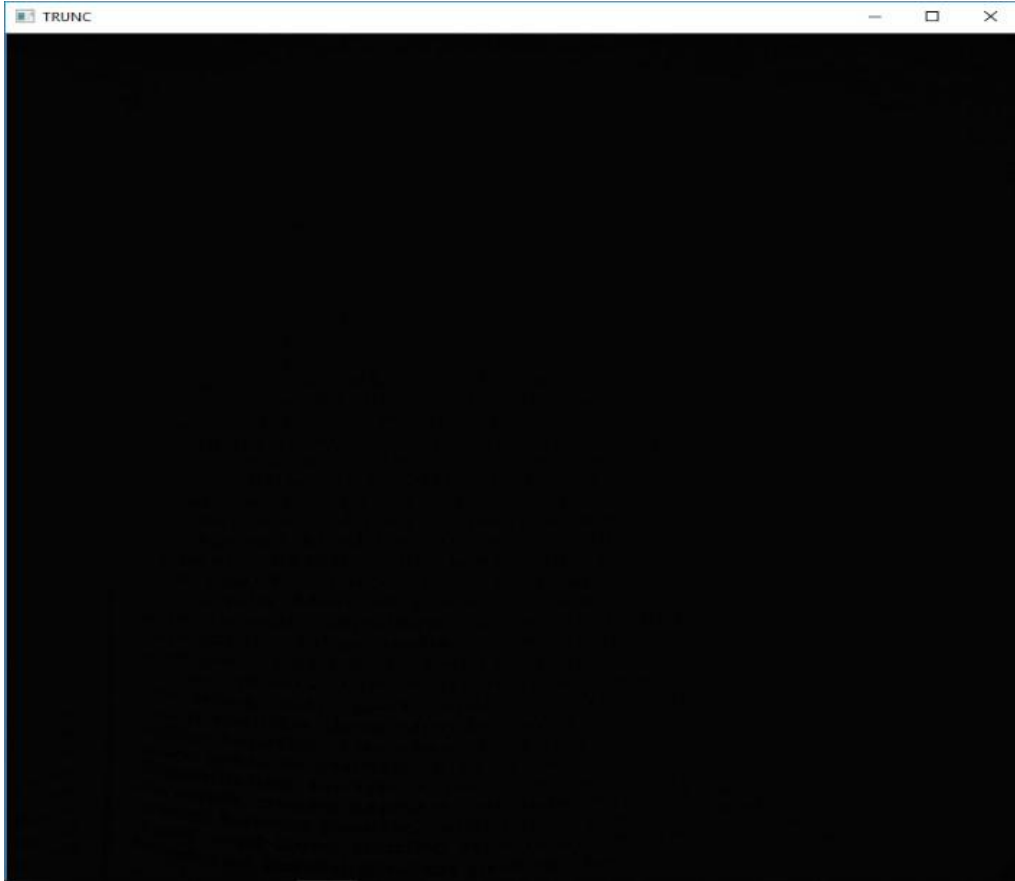
Her iki görselden de anlaşılabilir gibi bütünsel eşikleme yöntemleri farklı eşik değerleri farklı bölgeler için farklı sonuçlar oluşturmaktadır. Daha iyi bir sonuç etmek için görüntünün farklı ışık dağılımı olan bölgelerini tespit etmek ve her bir bölgeye özgü eşik değerleri uygulanmalıdır.



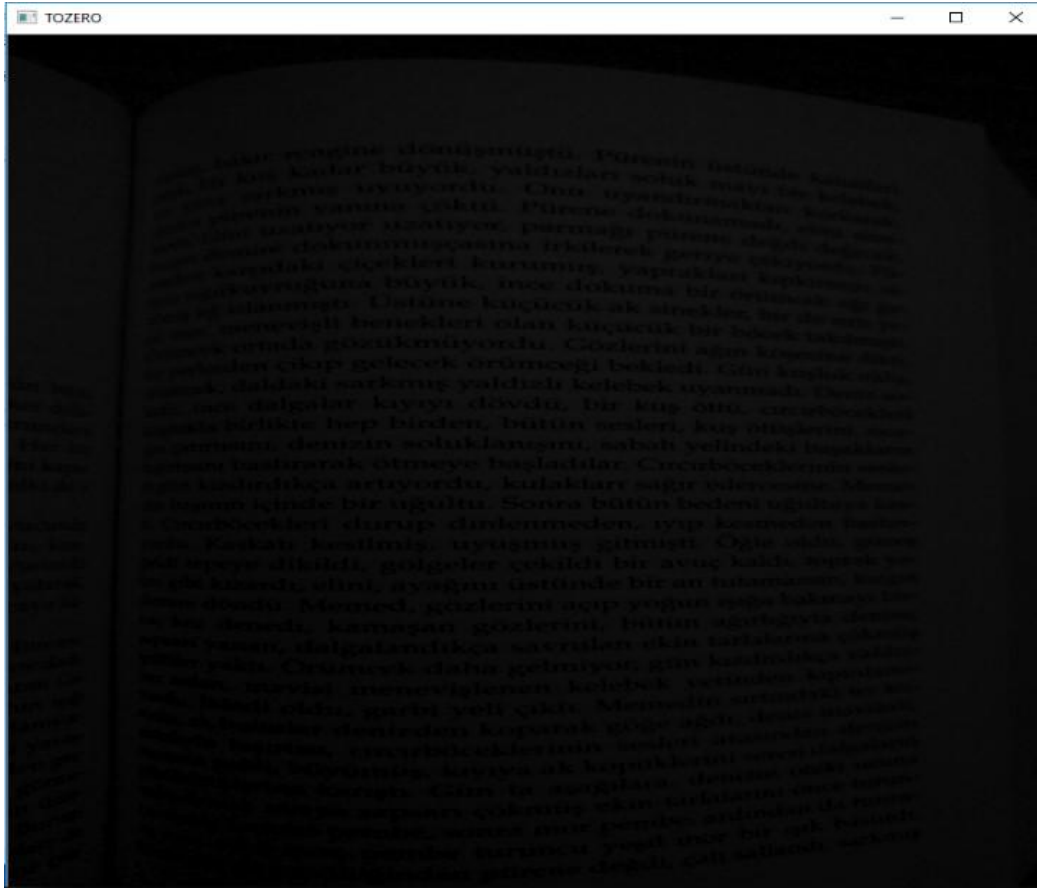
Şekil 10.8: $T = 5$ eşik seviyesinde THRESH_BINARY işleminin uygulanması sonucunda elde edilen görüntü



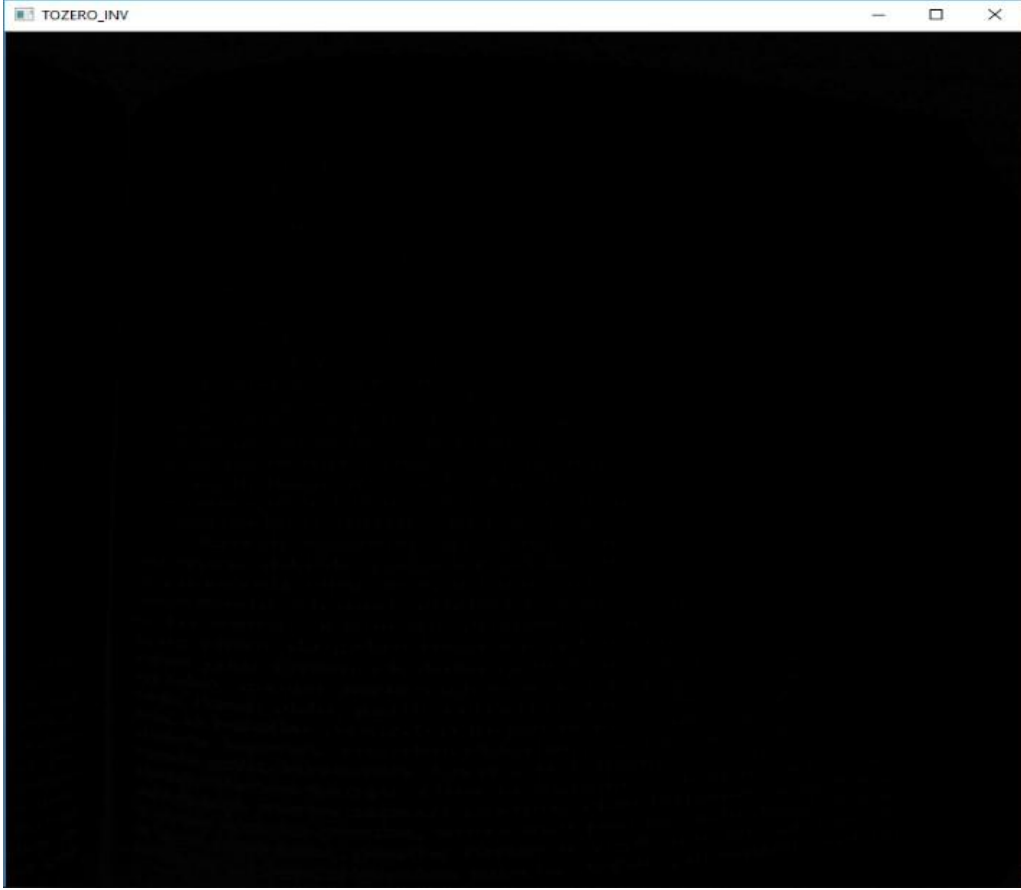
Şekil 10.9: $T = 5$ eşik seviyesinde THRESH_BINARY_INV işleminin uygulanması sonucunda elde edilen görüntü



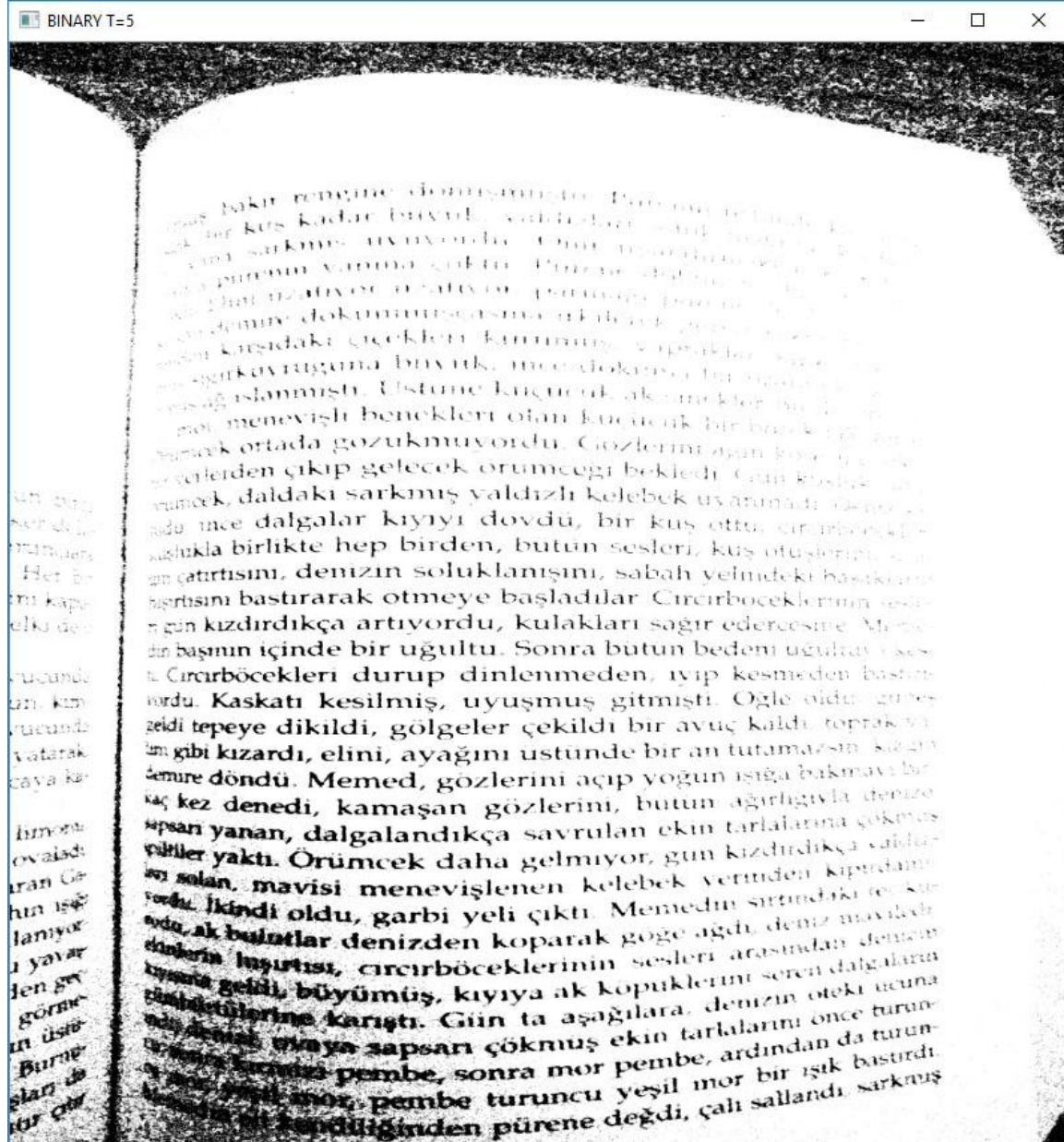
Şekil 10.10: $T = 5$ eşik seviyesinde THRESH_TRUNC işleminin uygulanması sonucunda elde edilen görüntü



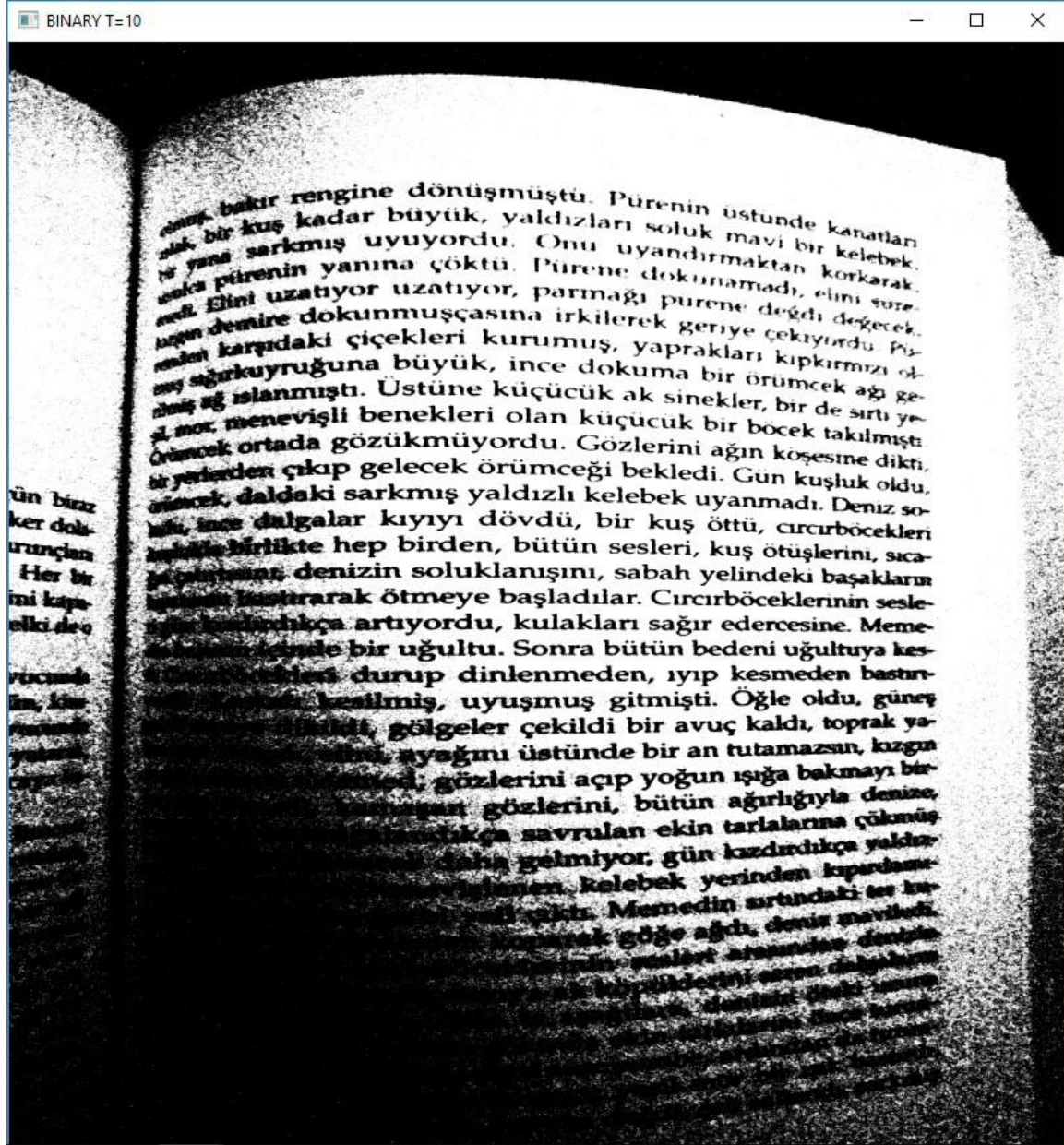
Şekil 10.11: $T = 5$ eşik seviyesinde THRESH_TOZERO işleminin uygulanması sonucunda elde edilen görüntü



Şekil 10.12: $T = 5$ eşik seviyesinde THRESH_TOZERO_INV işleminin uygulanması sonucunda elde edilen görüntü

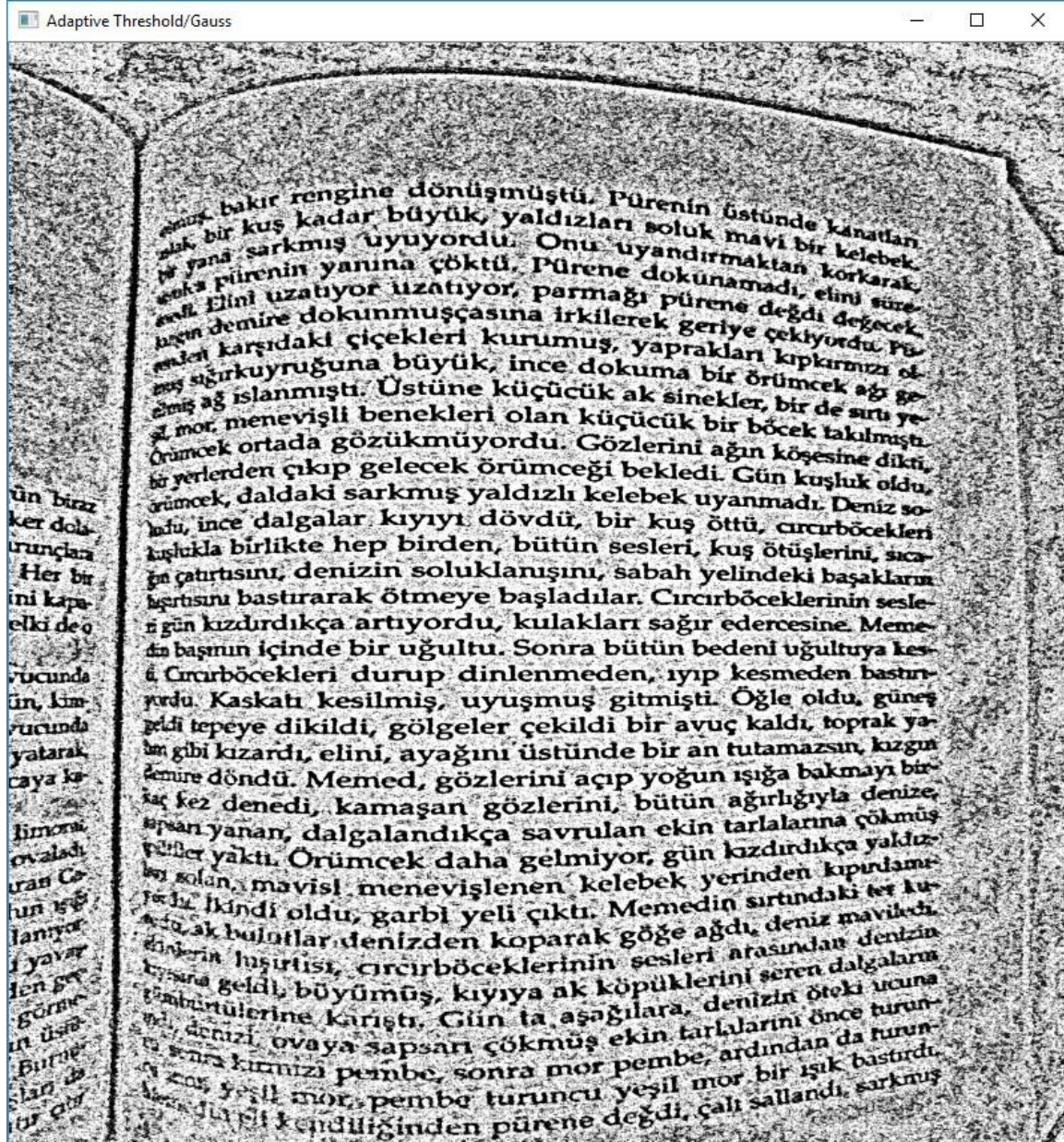


Şekil 10.13: $T = 5$ eşik seviyesinde THRESH_BINARY işleminin uygulanması sonucunda elde edilen görüntü

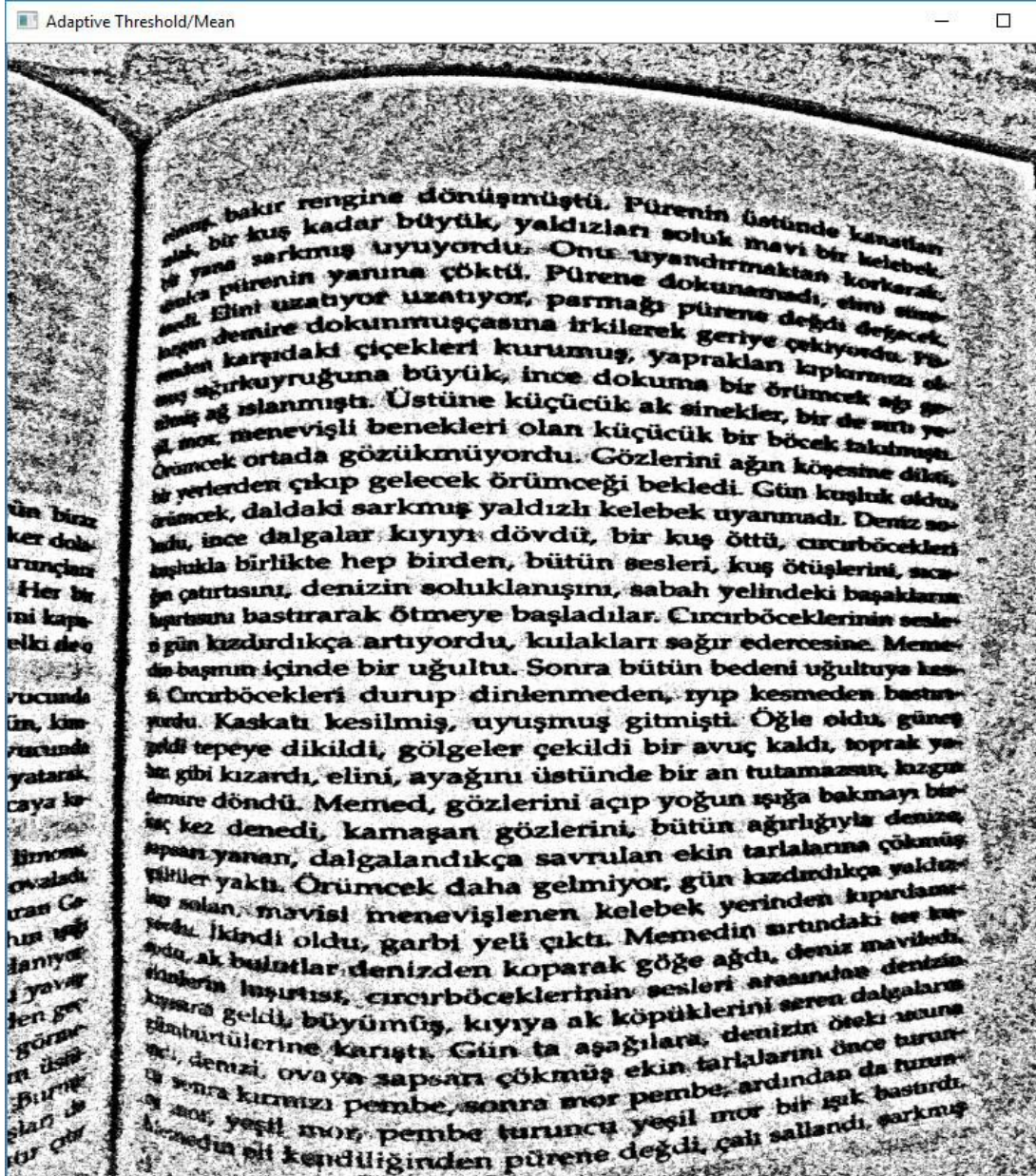


Şekil 10.14: $T = 10$ eşik seviyesinde THRESH_BINARY işleminin uygulanması sonucunda elde edilen görüntü

Bu durum daha önce bahsettiğimiz bölgesel eşikleme yöntemleriyle mümkündür. Bölgesel eşikleme yöntemi OpenCV kütüphanesindeki `adaptiveThreshold` metodu ile gerçekleştirilmektedir. Bu metodun iki farklı tekniğe göre uygulama ve çıktıları aşağıdaki gibidir.



Şekil 10.15: `.adaptiveThreshold` metodunun `THRESH_MEAN_C` tekniğinin sonucunda elde edilen görüntü



Şekil 10.16: .adaptiveThreshold metodunun ve THRESH_GAUSSIAN_C tekniğinin sonucunda elde edilen görüntü

```

th_gaussian=cv.adaptiveThreshold(img,MaxValue,cv.ADAPTIVE_THRESH_GAUSSIAN_C,cv.
THRESH_BINARY,115,1)
cv.imshow('Adaptive Threshold/Gauss',th_gaussian)

th_mean=cv.adaptiveThreshold(img,MaxValue,cv.ADAPTIVE_THRESH_MEAN_C,cv.
THRESH_BINARY,115,1)
cv.imshow('Adaptive Threshold/Mean',th_mean)

cv.waitKey(0)
cv.destroyAllWindows()

```

(10.17)

10.5 Sonuç

Gerçekleştirilmiş olan uygulamada farklı bütünsel ve bölgesel eşikleme yöntemleri kullanılmıştır. Kullanılan bütünsel eşikleme tiplerinden THRESH_BINARY ile diğer threshold tiplerine göre daha iyi bir sonuç elde edilmiştir. Bu sonuç, diğer threshold tiplerinin daha kötü olduğunu değil içerisinde bulunduğumuz şartlar için uygun olmadığını gösterir. Daha önce vurguladığımız üzere bir şarta uygun mükemmel bir bütünsel eşikleme yöntemi bulunmamaktadır.

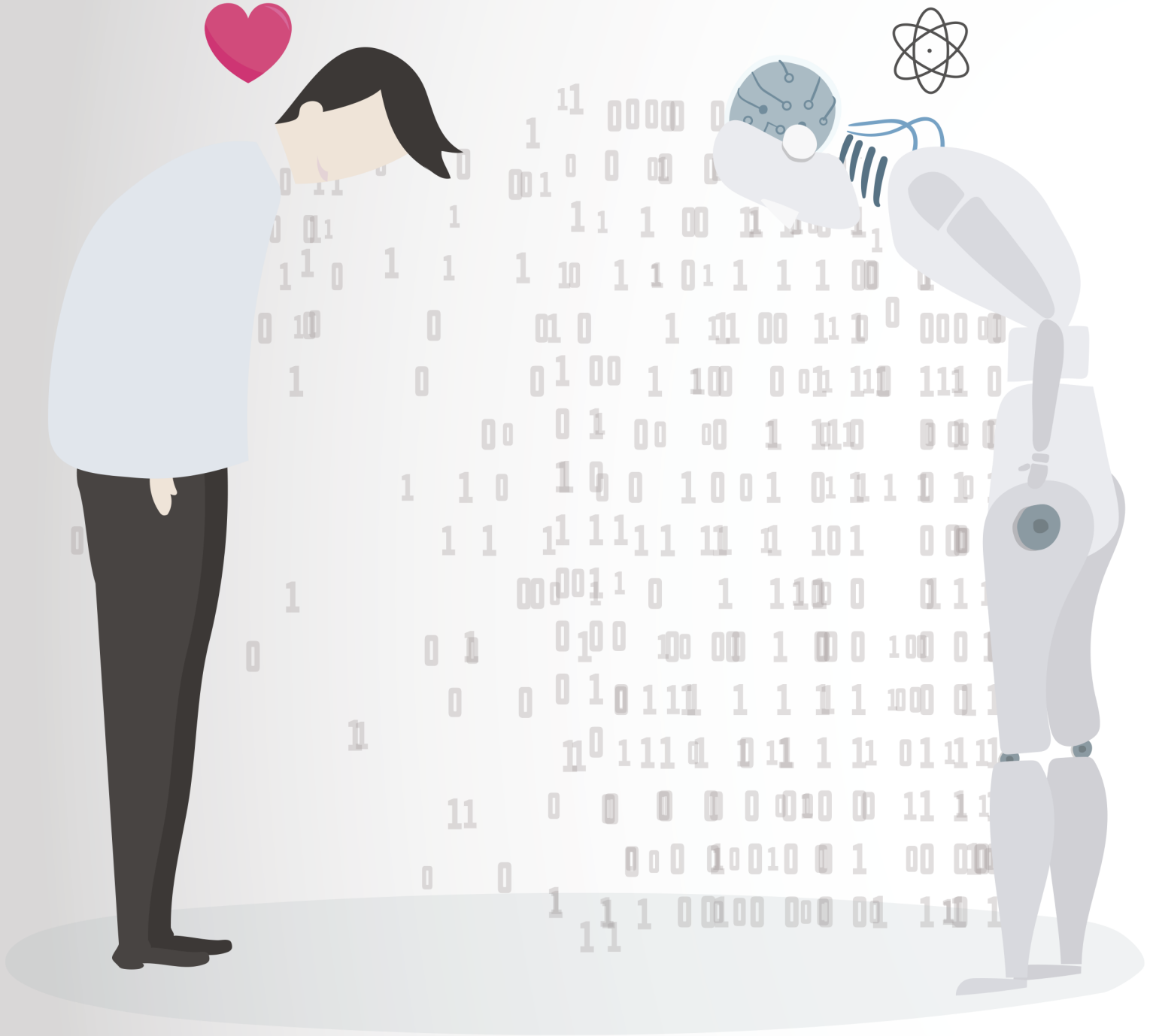
Bütünsel eşikleme yöntemlerinde belirlenen eşik değeri tüm görüntü için geçerli olduğundan, görüntü içerisinde bölgesel farklılıklar olumsuz sonuçların oluşmasına neden olur. Böyle durumlarda belirlenen eşik değeri bir bölge için iyi bir sonuç verirken diğeri için elde edilen sonucun kötüleşmesine neden olacaktır. Uygulamamızdaki örnekte eşik değerinin (T) 5'ten 10'a artışı görüntünün sağ-üst ve sol-alt bölümlerinde ters orantılı bir sonuç vermiştir. $T = 5$ minimum eşik değeri için görüntünün alt-orta kısmında okunaklı bir görüntü elde edilebilirken, görüntünün üst ve kenar kısımları net olarak okunamamaktadır. $T = 10$ minimum eşik değeri için ise görüntünün sağ üst kısmında net bir görüntü elde edilebilirken görüntünün alt kısımlarında iyi bir sonuç yakalanamamıştır. Bu sonuç bizlere kaynak görüntümüzün sağ üst tarafındaki piksel değerleri bu iki eşik değeri arasında yer alırken, görüntünün sol alt köşesindeki piksel yoğunluğunun bu sınırlardan küçük olduğunu göstermektedir. Bu bütünsel eşikleme yönteminin en büyük dezavantajlarından biridir.

Kaynak görüntünün farklı bölgesel özelliklerinin olması durumunda bu bölgelere özel eşik değerlerinin hesaplanıp uygulandığı bölgesel eşikleme yöntemleri kullanılır. Uygulamamızda bölgesel eşikleme yöntemi OpenCV kütüphanesindeki .adaptivethreshold() metodu ile gerçekleştirilmektedir. Ortalama (Mean) ve Gaussian teknikleriyle gerçekleştirilen adaptive threshold işlemleri sonuçlarının oldukça yakın olmasıyla beraber Gaussian tekniğinde daha temiz bir görüntü elde edildiği görülmektedir. Bu durum bütünsel eşikleme için de belirttiğimiz gibi sadece kullandığımız görüntünün özellikleri için Gaussian tekniğinin daha başarılı olduğunu gösterir. Farklı bir senaryo ya da çevresel koşullar için farklı tekniklerin daha iyi sonuç vermesi muhtemeldir.

10.6 Kaynakça

- [1] G. Oles, "Revolution or evolution of traditional Business Intelligence concept," 2013

- [2] 2020'de dünyanın toplam verisi 44 zettabyte'a ulaşacak!, Sabah Gazetesi , [Online]. Available: <https://www.sabah.com.tr/teknoloji/2015/05/06/2020de-dunyanin-toplam-verisi-44-zettabytea-ulasacak> , [Accessed: 10.10.2018]
- [3] Fırat Özgül, Python 3 için Türkçe Kılavuz Sürüm 3 , [Online]. Available: <https://belgeler.yazbel.com/python-istihza.pdf> , [Accessed: 10.10.2018]
- [4] Fırat Özgül, Python 3 için Türkçe Kılavuz Sürüm 3 , [Online]. Available: <https://belgeler.yazbel.com/python-istihza.pdf> , [Accessed: 10.10.2018]
- [5] MESUT PİŞKİN, OPENCV İLE GÖRÜNTÜ İŞLEME, [Online]. Available: <http://mesutpiskin.com/blog/wp-content/uploads/2017/01/0penCV%20Kitap.pdf> , [Accessed: 10.10.2018]
- [6] MESUT PİŞKİN, OPENCV İLE GÖRÜNTÜ İŞLEME, [Online]. Available: <http://mesutpiskin.com/blog/wp-content/uploads/2017/01/0penCV%20Kitap.pdf> , [Accessed: 10.10.2018]
- [7] Richard Szeliski, "Computer Vision: Algorithms and Applications", 2010
- [8] Dr. Aydın KIZILKAYA, "Görüntü Bölütleme", [Online] Available: http://akizilkaya.pamukkale.edu.tr/B%C3%B6l%C3%BCm4_goruntu_isleme.pdf [Accessed: 10.10.2018]
- [9] Fundamentals of Digital Image Processing Roger L. Easton, Jr.22 November 2010
- [10] Image Thresholding , [Online]. Available: https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html , [Accessed: 10.10.2018]



SAKARYA
ÜNİVERSİTESİ



www.yazsum.sakarya.edu.tr
SAÜ Mezunlar Derneği

nukeloveer
STUDIO

www.nukeloveerstudio.com