YILDIZ TEKNİK ÜNİVERSİTESİ

KİMYA-METALÜRJİ FAKÜLTESİ

MATEMATİK MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ

EKSİK VERİLERİN TAMAMLANMASI İÇİN KULLANILAN ALGORİTMALAR VE YÖNTEMLER

Tez Yöneticisi : Yrd. Doç. Dr. ARZU TURAN DİNÇEL

12053050 TUĞÇE KİRAZ

İstanbul, 2017

YILDIZ TEKNİK ÜNİVERSİTESİ

KİMYA-METALÜRJİ FAKÜLTESİ

MATEMATİK MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME TEZİ

EKSİK VERİLERİN TAMAMLANMASI İÇİN KULLANILAN ALGORİTMALAR VE YÖNTEMLER

Tez Yöneticisi : Yrd. Doç. Dr. ARZU TURAN DİNÇEL

12053050 TUĞÇE KİRAZ

İstanbul,2017

İÇİNDEKİLER **Sayfa**

SEMBOL LİSTESİ

Y	Regresyon Analizinde Bağımlı Değişken
X	Regresyon Analizinde Bağımsız Değişken
α	X=0 için Y'nin Sabit Değeri
β	Regresyon Katsayısı
ε	Varyansı 2 Olan Hata
W	Benzerlik Oranı
I	Nöron Değeri

KISALTMA LİSTESİ

EYK	En Yakın Komşu Algoritması
NB	Naive Bayes Algoritması
NBI-OI	Order Irrelevant Strategy
NBI-OR	Order Relevant Strategy
NBI-Hm	Hybrid Strategy
GA	Genetik Algoritma
DVR	Destek Vektör Regresyonu
YSA	Yapay Sinir Ağları
BCO	Bulanık c- Ortalama Metodu

ŞEKİL LİSTESİ **Sayfa**

TABLO LİSTESİ

ÖNSÖZ

Bu tez çalışmasında Veri Madenciliği alanında kullanılan büyük datalarda olan kayıp veri sorunu ele alınmıştır. Bu sorunun giderilmesi için kullanılan istatistiksel yöntemler ve algoritmalar hakkında bilgi verilmiştir.

Öncelikle tez konusunu seçerken isteklerimi göz önünde bulundurup bana yardımcı olan tez danışmanım Yrd. Doç. Dr. Arzu Turan Dinçel'e teşekkürlerimi sunarım. Tez esnasında yanımda olan arkadaşlarım ve tüm eğitim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan sevgili aileme teşekkürlerimi bir borç bilirim.

TUĞÇE KİRAZ

ÖZET

Veri kümeleri; veri madenciliği, makine öğrenmesi veya yapay zeka gibi disiplinlerin uygulanabilmesi için gereklidir. Veri kümelerindeki verinin kalitesi, doğru araştırma sonuçları elde edebilmek adına önemli bir konudur. Veri kümelerinde çeşitli nedenlerle veri kalitesini azaltan değeri olmayan nitelikler bulunabilmektedir. Değeri olmayan bu eksik değerler yapılmak istenen çalışmaya ait sonuçların güvenirliğini riske atabilmektedir. Bu nedenle veri kalitesini artırmaya yönelik yöntemler ile veri kümelerindeki eksik değer probleminin giderilmesi gerekmektedir. Bu tez çalışmasında eksik değer hesaplamasında kullanılan klasik yöntemlerden bahsedilerek alternatif gelişmiş yöntemler önerilmiştir.En basit olarak kayıp veri olan kayıtı yok saymak bulunabilecek çözümlerden birisidir. Ancak kayıp verilerin çok olduğu bir sette bu yöntem hata oranını oldukça yükseltir. Bu yöntemin yerine kayıp veriyi; Regresyon ile belirleme, Hot/Cold Deck ile Belirleme, Beklenti Maksimizasyonu, Son Gözlemi İleri Taşıma, Çoklu Atama, Karar Ağacı, Naive Bayes gibi yöntemler kullanılabilir.
Bu çalışmada bu yöntemlerin Avantaj ve Dezavantajları karşılaştırılmıştır.

ABSTRACT

Research data values when creating a two-dimensional data sets is very important. Sometimes this can greatly affect the research data sets have missing values. To reduce this problem, a lot of fault-tolerance methods have been developed that may arise. Most simply ignore the missing data record, which is one of the solutions can be found. However, there are a lot of lost data error rate of this method is quite raises a set. Instead of this method is that the data loss, the regression and determination, Hot / Cold Deck and Identification, Expectation Maximization, Last Observation Onward Transfer, Multiple Assignment, such as the decision tree methods can be used. Advantages and Disadvantages of these methods are compared in this study.

1.GİRİŞ

Kayıp veri terminolojisi ilk kez Little ve Rubin tarafından kullanılmıştır [3].Bu çalışmada kayıp verileri - varsayımsal açıdan -oluşum nedenlerine göre 3 ana sınıfta değerlendirmişlerdir.

1. Tümüyle Raslantısal Kayıp: Verileri setlerini oluştururken tamamen istek dışı oluşan veri kayıplarıdır. Örneğin bir anket çalışmasında soruyu görmeyip cevaplamama veya verilerden bazılarının kaybolması gibi.

2. Raslantısal Kayıp: Bir anket çalışmasında ; , örneklem bireylerini oluşturan grubun sorulan sorulara bilerek atlaması veya yanlış cevaplar vermesidir.

3. Raslantısal Olmayan Kayıp/ Gözardı Edilemez Kayıp: Anketteki bir sorunun yanlış sorulmasından dolayı doğru cevabın çözülemediği sorular bu varsayıma örnektir.

Eksik değerler veri madenciliği, makine öğrenmesi ve diğer bilgi sistemlerinde istenmeyen bir durumdur. Son yıllarda eksik değerlerin ya da diğer adıyla kayıp verilerin hesaplanması ve tahmin edilmesi, araştırmacıların kaliteli veriye ulaşma isteğinden dolayı popüler bir konu haline gelmiştir. Verinin bilgiye dönüşme sürecine bilgi keşfi denmektedir. Veri kalitesi makine öğrenmesi, veri madenciliği ve bilgi keşfi için büyük öneme sahiptir. Veri temizleme veri önişleme aşamalarından biridir. Veri temizleme sürecinin amacı kaliteli veri üretmektir. Verilerden yola çıkılarak elde edilecek olan bilginin keşfi sürecinde var olan veriler eksik değerler içerebilmektedir. Bu eksik değerlerin giderilmesi bilgi keşfi sürecinde göz önünde bulundurulması gereken bir adımdır. Çoğu bilgisayar bilimleri yöntemi, yapay sinir ağları, destek vektör makineleri ve karar ağaçları gibi kestirim yapan yaklaşımlar daha önce görülmüş veriyi girişte eğitim verisi olarak alarak çıkışta bir sınıflama yapmaktadır. Bu gibi kestirim modelleri girişte bir veya birden çok veride eksik değer olması durumunda ya çalışmaz ya da hatalı tahmin üretmektedir. Sonuç olarak eğer giriş veri nitelikleri tam değil ise bu durumda karar verme amacıyla kullanılamazlar. Veri kümelerindeki eksik değerlerle başa çıkabilmek için eksik veriyi göz ardı etmek, ilgili kaydı silmek, sıfır ile doldurmak, sık geçen ifade ile doldurmak veya ilgili kayıt ya da niteliğin satır, sütun ortalaması ile doldurulması gibi basit klasik yöntemler karmaşık hesaplama yöntemlerinin yerine kullanılmaktadır.

Fakat bu gibi basit klasik yöntemler eksik gözlemleri yok sayarak verimi düşürmekte aynı zamanda var olan veriyi yanlılaştırarak sistematik anlamda kalitesizleştirmektedir.

2.VERİ MADENCİLİĞİ

Bu bölümde veri madenciliği, veri ambarı ve veri madenciliği yöntemleri anlatılacaktır.

2.1 Veri Madenciliği Nedir?

Büyük miktardaki veriler içerisinden önemli olanlarını bulup çıkarmaya Veri Madenciliği denir. Veriler üzerinde çözümlemeler yapmak amacıyla ve veriyi çözümleyip bilgiye ulaşabilmek için veri madenciliği yöntemi ortaya çıkmıştır. Veri madenciliği bir sorgulama işlemi veya istatistik programlarıyla yapılmış bir çalışma değildir. Veri madenciliği milyarlarca veri ve çok fazla değişken ile ilgilenir. Veri madenciliği uygulamalarında alt yapı gereksinimi veri ambarı sayesinde sağlanır.

2.2 Veri Ambarı Nedir?

Veri ambarı, 1991 yılında ilk kez William H. Inmon tarafından ortaya atılan veri ambarı, yönetimin kararlarını desteklemek amacı ile çeşitli kaynaklardan elde ettikleri bilgileri zaman değişkeni kullanarak veri toplama olarak tanımlanmaktadır. Kısaca birçok veri tabanından alınarak birleştirilen verilerin toplandığı depolardır. Veri ambarlarının özelliği kullanıcılara farklı detay düzeyleri sağlayabilmesidir. Detayın en alt düzeyi arşivlenen kayıtların kendisi ile ilgili iken, daha üst düzeyler zaman gibi daha fazla bilginin toplanması ile ilgilidir. Veri ambarları ciddi yatırımlar gerektirmekte ve uygulanması bir yıl veya daha uzun zaman almaktadır.

Veri ambarları, verilerin üzerine yazmaya ve verilerde değişiklik yapmak için değil sadece okumaya yönelik olarak oluşturulmaktadır. Bu nedenle veri ambarında veriler, analiz yapmayı kolaylaştıran bir formatta tutulmaktadır. Burada analiz; sorgular, raporlar, karar destek sistemleri veya istatistiki hesapları kapsamaktadır. Birbiriyle bütünleşik olmayan uygulamaların bütünleştirilmesine olanak sağlar. Veri Ambarları, sağlık sektöründen bilişim sistemlerine, işletmelerin pazarlama bölümünden üretime, geleceğe dönük tahminler yapmada, sonuçlar çıkarmada ve işletmelerin yönetim stratejilerini belirlemede kullanılmakta olan bir sistemdir. Pahalı bir yatırım maliyeti olsa bile sonuç olarak getirisi (yararı) bu maliyeti kat kat aşmaktadır.

İş organizasyonlarında bilgi akış mimarisinde veri ambarları iki amaçla oluşturulmaktadır:

• Hareketsel ve organizasyonel görevler arasındaki depo ve analitik stratejik verilerin birikimini sağlar. Bu veriler daha sonra yeniden kullanılmak üzere arşivlenir. Veri ambarları verilerin sorgulanabildiği ve analiz yapılabilen bir depodur.

• Veri Ambarlarının pazarda yeni fırsatlar bulmaya, rekabete katkı, yoğun proje çevirimi, iş, envanter, ürün maliyetlerinin azalmasının yanında farklı işlere ait verilerin ilişkilendirilmesi, karar destek ve alınan bilgiye hızlı cevap verebilme gibi birçok katkısı vardır. Karar verme sürecinde yöneticilere destek vermek amacıyla hazırlanmış; konuya yönelik bütünleşik, zaman boyutu olan ve sadece okunabilen veri topluluğudur. Bir işletmenin sahip olduğu verinin, eskileri de dahil olmak üzere, karar destek amacıyla kullanılmasına olanak sağlar.

2.3 Veri Ambarının Temel Özellikleri

• İşlemsel çevrede yer alan veri bir süzme işlemi sonucunda veri ambarı çevresine aktarılır.

• Zaman yelpazesi her iki sistemde farklılık gösterir. İşlemsel ortamdaki veri çok taze, veri ambarındaki eskidir.

• Veri ambarı özet bilgileri içerebilir. İşlemsel veri ise içermez.

• Bütünleştirmeyi sağlamak için verinin önemli bir kısmı belirli bir dönüşümden sonra veri ambarına aktarılır.

2.4 Veri Ambarının İçerdiği Veriler

2.4.1 Meta Data

• Doğrudan işlemsel çevreden gelen veriyi içermez.

• Karar Destek Sistemleri analizlerine yardım etmek üzere yaratılan bir dizindir.

• İşlemsel çevreden veri ambarına dönüştürülen verilerin konumları hakkında bilgi verir.

• İşlemsel çevreden alınan verinin hangi algoritmaya göre düşük yada yüksek seviyede özetlendiği hakkında bilgi verir.

2.4.2 Ayrıntı Veri

Bu veri en son olayları içermektedir ve henüz işlenmediği için diğerlerine oranla daha büyük hacimlidir.

2.4.3 Eski ayrıntı Veri

Ayrıntı verinin dışında kalan verilerdir. Daha eski tarihe aitlerdir.

2.4.4 Düşük Düzeyde(seviyede) Özetlenmiş Veri

Ayrıntı veriden süzülerek elde edilen düşük seviyede özetlenmiş veridir.

2.4.5 Yüksek Seviyede Özetlenmiş Veri

Ayrıntı veri daha yüksek düzeyde özetlenerek, kolayca erişilebilir hale getirebilir.

2.5 Veri Ambarının Kullanım Amaçları

- Müşterilerin gizli kalmış satın alma eğilimlerini tespit etmek
- Satış analizi ve trendler üzerine odaklanmak,
- Finansal analiz(Maliyetlerin azaltılması dolayısıyla rekabet avantajının sağlanması)
- Stratejik Analiz (Bir Karar Destek Sistemi olmasından dolayı)
- İşler arasında ilişkilerin belirlenebilmesi
- Müşteri ihtiyaçlarına çabuk cevap verebilme Veriyi yönetmek için “veri ambarı”, verileri çözümleyip bilgiye ulaşılabilmesi için “veri madenciliği” yöntemleri ortaya çıkmıştır.

2.6 Veri Madenciliği Süreci

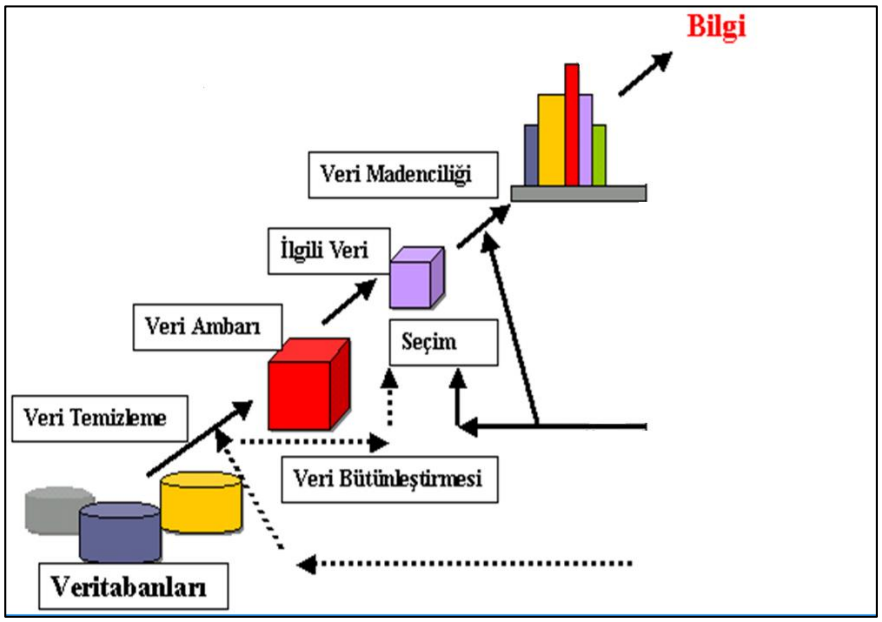
Veri madenciliği süreci:

1. Veri temizleme
2. Veri bütünleştirme
3. Veri indirgeme

4. Veri dönüştürme

5. Veri madenciliği algoritmasını uygulama

6. Sonuçları sunum ve değerlendirme



Şekil 2.1 Veri Madenciliği Süreci

2.6.1 Veri Temizleme: Veri tabanında yer alan tutarsız ve hatalı verilere gürültü denir. Verilerdeki gürültüyü temizlemek için; eksik değer içeren kayıtlar atılabilir, kayıp değerlerin yerine sabit bir değer atanabilir, diğer verilerin ortalaması hesaplanarak kayıp veriler yerine bu değer yazılabilir, verilere uygun bir tahmin (karar ağacı, regresyon) yapılarak eksik veri yerine kullanılabilir.

2.6.2 Veri Bütünleştirme: Farklı veri tabanlarından ya da veri kaynaklarından elde edilen verilerin birlikte değerlendirmeye alınabilmesi için farklı türdeki verilerin tek türe dönüştürülmesi işlemidir. Bunun en yaygın örneği cinsiyette görülmektedir.

Çok fazla tipte tutulabilen bir veri olup, bir veri tabanında 0/1 olarak tutulurken diğer veri tabanında E/K veya Erkek/Kadın şeklinde tutulabilir. Bilginin keşfinde başarı verinin uyumuna da bağlı olmaktadır.

2.6.3 Veri İndirgeme: Veri madenciliği uygulamalarında çözümlemeden elde edilecek sonucun değişmeyeceğine inanılıyorsa veri sayısı ya da değişkenlerin sayısı azaltılabilir. Veri indirgeme yöntemleri; veri sıkıştırma, örnekleme, genelleme, birleştirme veya veri küpü, boyut indirgeme.

2.6.4 Veri Dönüştürme: Verinin kullanılacak modele göre içeriğini koruyarak şeklinin dönüştürülmesi işlemidir. Dönüştürme işlemi kullanılacak modele uygun biçimde yapılmalıdır. Çünkü verinin gösterilmesinde kullanılacak model ve algoritma önemli bir rol oynamaktadır. Değişkenlerin ortalama ve varyansları birbirlerinden önemli ölçüde farklı olduğu taktirde büyük ortalama ve varyansa sahip değişkenlerin diğerleri üzerindeki baskısı daha fazla olur ve onların rollerini önemli ölçüde azaltır. Bu yüzden veri üzerinde normalizasyon işlemi yapılmalıdır.

2.6.5 Veri Madenciliği Algoritmasını Uygulama: Veri hazır hale getirildikten sonra konuyla ilgili veri madenciliği algoritmaları uygulanır.

2.6.6 Sonuçları Sunum ve Değerlendirme: Algoritmalar uygulandıktan sonra, sonuçlar düzenlenerek ilgili yerlere sunulur. Örneğin hiyerarşik kümeleme yöntemi uygulanmış ise sonuçlar dendrogram grafiği sunulur.

2.7 Veri Madenciliği Yöntemleri

1. Sınıflandırma

2. Kümeleme

3. Birliktelik Kuralı

2.7.1 Sınıflandırma

Sınıflandırma veri madenciliğinin en çok kullanıldığı alandır. Var olan veri tabanının bir kısmı eğitim olarak kullanılarak sınıflandırma kuralları oluşturulur. Bu kurallar yardımıyla yeni bir durum ortaya çıktığında nasıl karar verileceği belirlenir.

Veri madenciliğinin sınıflandırma grubu içerisinde en sık kullandığı teknik karar ağaçlarıdır. Aynı zamanda lojistik regresyon, diskriminant analizi, sinir ağları ve fuzzy setleri de kullanılmaktadır.

İnsanlar verileri daima sınıflandırdıkları, kategorize ettikleri ve derecelendirdikleri için sınıflandırma, hem veri madenciliğinin temeli olarak hem de veri hazırlama aracı olarak da kullanılabilmektedir.

Sınıflandırma Süreci: Verilerin sınıflandırılma süreci iki adımdan oluşur.

1-Veri kümelerine uygun bir model ortaya konur. Söz konusu model veri tabanındaki alan isimleri kullanılarak gerçekleştirilir. Sınıflandırma modelinin elde edilmesi için veritabanından bir kısım eğitim verileri olarak kullanılır. Bu veriler veritabanından rastgele seçilir.

Tablo 2.1 Sınıflandırma Model Kurma Süreci

Eğitim Verisi

Müşteri	Borç	Gelir	Risk
Ali	Yüksek	Yüksek	Kötü
Ayşe	Yüksek	Yüksek	Kötü
Fatma	Yüksek	Düşük	Kötü
Fuat	Düşük	Yüksek	İyi
Ece	Düşük	Düşük	Kötü
Ayla	Düşük	Yüksek	İyi

⇩

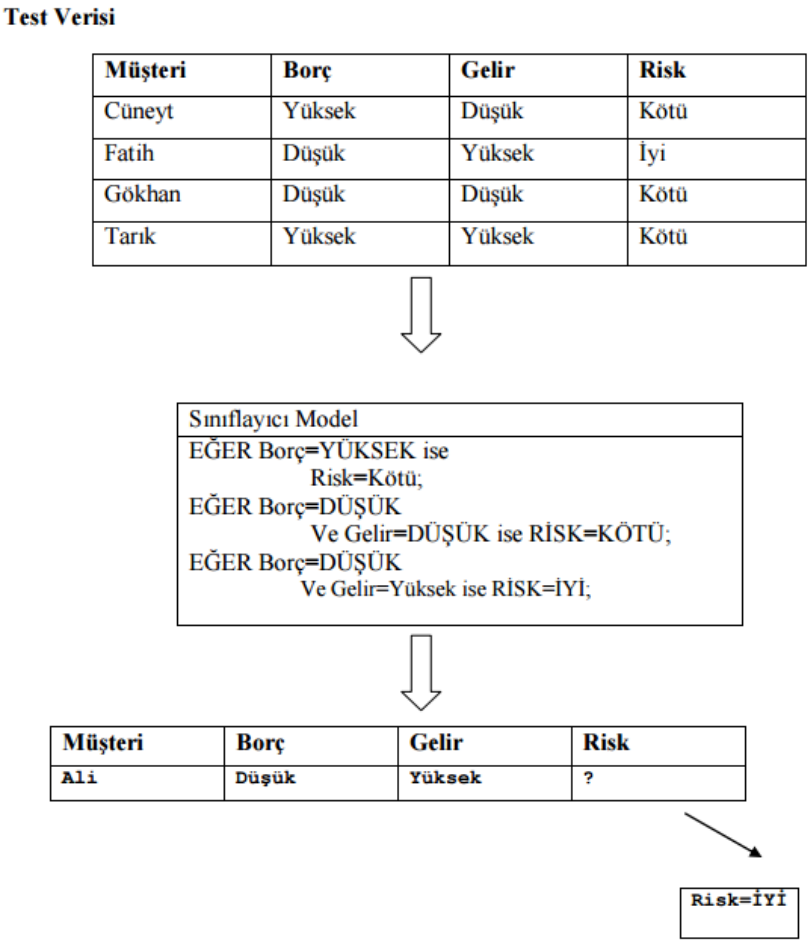
Sınıflandırma algoritması

⇩

Sınıflayıcı Model
EĞER Borç=YÜKSEK ise Risk=Kötü; EĞER Borç=DÜŞÜK Ve Gelir=DÜŞÜK ise RİSK=KÖTÜ; EĞER Borç=DÜŞÜK Ve Gelir=Yüksek ise RİSK=İYİ;

2- Test verileri üzerinde sınıflandırma kuralları belirlenir. Ardından söz konusu kurallar bu kez test verilerine dayanarak sınanır. Örneğin Ali adlı yeni bir banka müşterisinin kredi talebinde bulunduğunu varsayalım. Bu müşterinin risk durumunu belirlemek için örnek verilerden

elde edilen karar kuralı doğrudan uygulanır. Bu müşteri için Borç=Düşük, Gelir=Yüksek olduğu biliniyorsa risk durumunun Risk=İYİ olduğu hemen anlaşılır.



Şekil 2.2 Test Verisi Üzerinde Sınıflandırma Kuralları Belirleme

Yukarıdaki test sonucunda elde edilen modelin doğru olduğu kabul edilecek olursa, bu model diğer veriler üzerinde de uygulanır. Elde edilen sonuç model mevcut ya da olası müşterilerin gelecekteki kredi talep risklerini belirlemede kullanılır

3- Karar ağaçları akış şemalarına benzeyen yapılardır. Her bir nitelik bir düğüm tarafından temsil edilir. Dallar ve yapraklar ağaç yapısının elemanlarıdır. En son yapı ”yaprak”, en üst yapı “kök” ve bunların arasında kalan yapı ise “dal” olarak adlandırılır. (Quinlan,1993).

Karar ağaçları sınıflama algoritmasını uygulayabilmek için uygun bir alt yapı sağlamaktadır. Karar ağacı oluşturmak için birçok yöntem geliştirilmiştir. Bunlar temel olarak:

1. Entropiye dayalı algoritmalar
2. Sınıflandırma ve Resresyon araçları
3. Bellek tabanlı sınıflandırma modelleri

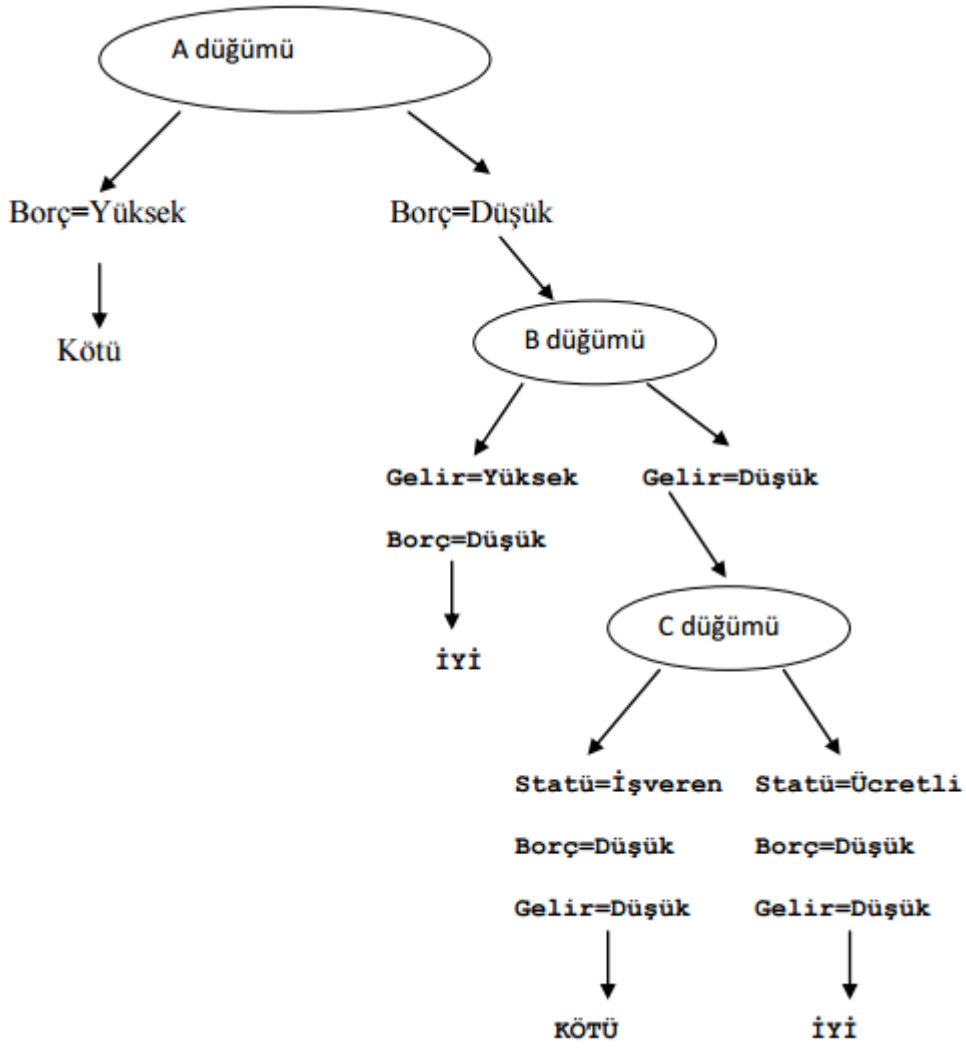
Örnek:

Tablo 2.2 Örnek Tablo

Borç	**Gelir**	**Statü**	**Risk**
Yüksek	Yüksek	İşveren	Kötü
Yüksek	Yüksek	Ücretli	Kötü
Yüksek	Düşük	Ücretli	Kötü
Düşük	Düşük	Ücretli	İyi
Düşük	Düşük	İşveren	Kötü
Düşük	Yüksek	İşveren	İyi
Düşük	Yüksek	Ücretli	İyi
Düşük	Düşük	Ücretli	İyi
Düşük	Düşük	İşveren	Kötü
Düşük	Yüksek	İşveren	İyi

Tablodan yararlanılarak karar ağacı oluşturulur. Karar ağacı oluşturulduktan sonra karar kuralları oluşturulur.

Tablo 2.3 Karar Koşulları



Kurallar: Kural.1:Borç Yüksek ise Risk Kötü

Kural.2:Borç Düşük ve Gelir=Yüksek ise Risk=İyi

Kural.3:Borç Düşük ve Gelir=Düşük ve Statü=İşveren ise Risk=Kötü

Kural.4:Borç Düşük ve Gelir=Düşük ve Statü=İşveren ise Risk=Kötü

2.7.2 Kümeleme

Verilerin kendi aralarındaki benzerliklerin göz önüne alınarak gruplandırılması işlemidir ve kümeleme yöntemlerinin çoğu veri arasındaki uzaklıkları kullanır. Hiyerarşik Kümeleme yöntemleri en yakın komşu algoritması ve en uzak komşu algoritmasıdır. Hiyerarşik olmayan kümeleme yöntemleri arasında k-ortalamalar yöntemi sayılabilir.

Uygulamada çok sayıda kümeleme yöntemi kullanılmaktadır. Bu yöntemler, değişkenler arasındaki benzerliklerden ya da farklılıklardan yararlanarak bir kümeyi alt kümelere ayırmakta kullanılmaktadır.

Hangi tekniğin kullanılacağı küme sayısına bağlı olmakla birlikte her iki tekniğin beraber kullanılması çok daha yararlıdır. Böylece hem sonuçları hem de iki tekniğin hangisinin daha uygun sonuçlar verdiğini karşılaştırmak mümkün olmaktadır.

Kümeleme analizinin amacı, gruplanmamış verileri benzerliklerine göre sınıflandırmak ve araştırmacıya özetleyici bilgiler elde etmede yardımcı olmaktır. Kümeleme analizinin uygulanabilmesi için verilerin normal dağılımlı olması varsayımı olmakla birlikte, bu varsayım teoride kalmakta ve uygulamalarda göz ardı edilmektedir. Sadece uzaklık değerlerinin normal dağılıma uygunluğu ile yetinilmektedir. Bu varsayımın sağlanması durumunda kümeleme analizinde Kovaryans matrisi için farklı bir varsayım gerekmemektedir.

‘Küme, birbirine yakın (benzer) nesnelerin çok boyutlu uzayda oluşturdukları bulutlar benzetmesi’ seklinde tanımlanabilir.(Hüseyin Tatlıdil, Uygulamalı Çok Değişkenli istatistiksel Analiz, Ankara: Ziraat Matbaacılık, , 2002,s.330.) Kümeleme analizi ise; bu kümeleri oluşturma işlemidir.

Örnek:

Tablo 2.4 Örnek Tablo

Gözlem	X1	X2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7

Bu tabloya en yakın komşu algoritması uygulandığında;

Tablo 2.5 Sonuç Tablosu

Kümeler
Küme 1=1,2
Küme 2=4,5
Küme 3=3,4,5
Küme 4=1,2,3,4,5

1984 yılında Londra'da kolera salgını baş göstermiş. Çok ciddi ölümler kaydedilmiş(10675 kişi) John Snow bir harita üzerinde ölen kişilerin yerlerini işaretlediğinde kayıpların bazı bölgelerde yoğunlaştığını fark ediyor. O bölgede su pompalarına bakılıp atık su tesisindeki problem tespit edilerek koleradan meydana gelen ölümler engellenmiş. Ana sokaklardan birindeki su pompasının sapını çıkarmak kolera salgınının sonlanması için yeterli olmuştur. Bu veri madenciliğinde kümeleme yönteminin ilk kez yapıldığı kağıt kalemle analizdir. Veri miktarı az olduğu için kağıt kalemle yapmakta bir sıkıntı yok ama günümüzde bu pek mümkün değil.

2.7.3 Birliktelik Kuralları

Veri tabanı içinde yer alan kayıtların birbiriyle olan ilişkilerini inceleyerek, hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan veri madenciliği yöntemleridir. Özellikle pazarlama alanında uygulanmaktadır (Pazar sepet analizleri). Bu yöntemler birlikte olma kurallarını belirli olasılıklarla ortaya koyar.

Birliktelik çözümlemelerinin en yaygın uygulaması perakende satışlarda müşterilerin satın alma eğilimlerini belirlemek amacıyla yapılmaktadır. Müşterilerin bir anda satın aldığı tüm ürünleri ele alarak satın alma eğilimini ortaya koyan uygulamalara ”Pazar sepet çözümlemesi” denilmektedir.

Örneğin; bir mağazadan parfüm alan müşterilerin %60’ının aynı alışverişte parfüm satın aldıklarını söylemek, bu birlikte gerçekleşen olaylara örnek olarak verilebilir.

2.7.3.1 Apriori Algoritması: Birliktelik kurallarının üretilmesi için kullanılan en yaygın yöntemdir. Aşamaları:

• Destek ve güven ölçütlerini karşılaştırmak üzere eşik değerler belirlenir. Uygulamadan elde edilen sonuçların bu eşik değere eşit ya da büyük olması beklenir.

• Destek sayıları hesaplanır. Bu destek sayıları eşik destek sayısı ile karşılaştırılır. Eşik destek sayısından küçük değerlere sahip satırlar çözümlemeden çıkarılır ve koşula uygun kayırlar göz önüne alınır.

• Bu şeçilen ürünler bu kez ikişerli gruplandırılarak bu grupların tekrar sayıları elde edilir. Bu sayılar eşik destek sayıları ile karşılaştırılır. Eşik değerden küçük değerlere sahip satırlar çözümlemeden çıkarılır.

• Bu kez üçerli, dörderli vb. gruplandırmalar yapılarak bu grupların destek sayıları elde edilir ve eşik değeri ile karılaştırılır, eşik değere uygun olduğu sürece işlemlere devam edilir.

• Ürün grubu belirlendikten sonra kural destek ölçütüne bakılarak birliktelik kuralları türetilir ve bu kuralların her birisiyle ilgili olarak güven ölçütleri hesaplanır.

Tablo 2.6 Örnek Tablo

Müşteri	Aldığı ürünler
1	Makarna, yağ, meyve suyu, peynir
2	Makarna, ketçap
3	Ketçap, yağ, meyve suyu, bira
4	Makarna, ketçap, yağ, meyve suyu
5	Makarna, ketçap, yağ, bira

Apriori algoritması uygulandığında şu sonuçlar elde edilir:

- {ketçap, meyve suyu} - {yağ} (s=0,4 c=1.0)
- {ketçap, yağ} - {meyve suyu} (s=0,4 c=0.67)
- {yağ, meyve suyu} - {ketçap} (s=0,4 c=0.67)
- {meyve suyu} - {ketçap, yağ} (s=0,4 c=0.67)
- {yağ} - {ketçap, meyve suyu} (s=0,4 c=0.5)
- {ketçap} - {yağ, meyve suyu} (s=0,4 c=0.5)

3.VERİ MADENCİLİĞİNDE EKSİK VERİ

Araştırmalarda iki boyutlu veri setleri oluşturulurken veri değerleri çok önem arz etmektedir. Bazen bu veri setlerinde eksik değerler olması araştırmayı oldukça etkileyebilir. Bu sorundan doğabilecek hata toleransını azaltabilmek için birçok yöntem geliştirilmiştir. En basit olarak kayıp veri olan kayıtı yok saymak bulunabilecek çözümlerden birisidir. Ancak kayıp verilerin çok olduğu bir sette bu yöntem hata oranını oldukça yükseltir. Bu yöntemin yerine kayıp veriler için kullanılan yöntemler bu bölümde ele alınacaktır.

3.1 Metodoloji

Eksik verilerin tamamlanmasında temel istatistikleri kullanan basit yöntemlerden çeşitli analizleri temel alan algoritmalara kadar farklı uygulama ve yöntemler geliştirilmiştir. Bu ihtiyacın temel sebebi incelenen verilerin dağılım özelliklerinin bozulmamasını sağlamaktır. Eksik verilerin ele alınması ile ilgili temel yaklaşımlar hiçbir verinin eksik olmadığı gözlemlerin kullanılması ve korelasyona dayalı yöntemlerde eksik verisi bulunmayan değişken çiftlerinin kullanılması şeklinde ortaya çıkmışlardır. Model tabanlı olmayan eksik veri tamamlama yaklaşımları olarak ortalama ve kesikli verilerde medyan değerinin kullanılması yaygındır. Bu yöntemler dağılımı homojenliği arttırıcı şekilde bozmakta ve gözlemlerin küme özelliklerini dikkate almamaktadırlar. Verilerin belirli kümeler veya sıralamalara göre girilmesi durumunda “en yakın komşu” gözlemlerin ortalama veya medyan değerinin kullanılması tercih edilmektedir. Yakın zamanda gelişen bu yaklaşım “hot deck imputation” olarak tanımlanmakta; diğer veri tabanlarının veya merkezi eğilim ölçülerinin kullanılmasında ise “cold deck imputation” adını almaktadır. Model tabanlı eksik veri tamamlama yöntemlerinde ise zaman serilerinde eksik verinin komşularına göre yarı ortalamalar yönteminin kullanılması veya trend denklemi yardımı ile interpolasyon yapılması mümkün olmaktadır.

Gözlemlenebilen/Ölçülebilen bütün sistemler de belirli oranlarda - az ya da çok - kayıp veri varlığı kaçınılmazdır. Veri kaybı belirlenen ya da belirlenemeyen pek çok nedenden kaynaklanabilir. Doğru ve güvenilir analizler için veri kümesinin eksiksiz olması oldukça önemlidir.

Kayıp verilerin değerlendirilmesindeki ilk yöntem kayıp veri olan kayıtı yok saymaktır. Ancak kayıp verinin çok olduğu yada az kayıta sahip testlerde bu çözüm sonucu yanlış değerlere saptırmaktadır. Bu yüzden kayıp verilerin yerine bir değer ataması yapmak çok daha iyi bir sonuç olduğu ortaya çıkmıştır. Bu çalışmada ele alınacak olan yöntemler;

- Regresyon Analizi
  -En Küçük Kareler Metodu
- Hot Deck Algoritması
  -En Yakın Komşu Algoritması
- Naive Bayes ile Değer Atama Metodu
  -Naive Bayes Algoritması
- Karar Ağaçları
  -C4.5 Karar Ağacı Algoritması
- Yapay Sinir Ağları
- Genetik Algoritmalar
- Bulanık C-ortalama Yöntemi
- Beklenti Maksimizasyonu Algoritması
  -EM Algoritması
- Markoz Zinciri ile Çoklu Atama Metodu
  -Monte Carlo Yöntemi
- Yerine Ortalama Koyma Yöntemi

Bu tez çalışmasında tüm algoritma ve istatistik yöntemler hakkında bilgi veriliyor olup, uygulama kısmında sadece regresyon analizi, en yakın komşu algoritması ve yerine koyma metodu yöntemleri ele alınmıştır.

3.2 Regresyon Analizi

Bu yöntem, değerler arasındaki ilişkileri tahmin etmek için kullanılan istatistiksel bir tekniktir. Aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi belirlemek ve bu ilişkiyi kullanarak o konu ile ilgili tahminler (estimation) ya da kestirimler (prediction) yapabilmek amacıyla yapılır.

Bu analiz tekniğinde iki (basit regresyon) veya daha fazla değişken (çoklu regresyon) arasındaki ilişki açıklamak için matematiksel bir model kullanılır ve bu model regresyon modeli olarak adlandırılır. Aşağıda basit regresyon analizi anlatılmıştır;

Basit regresyon modeli;

$$Y=\alpha+\beta X+\varepsilon \qquad (3.1)$$

Şeklinde bir bağımlı ve bir de bağımsız değişken içeren bir modeldir. Burada:

Y; bağımlı (sonuç) değişken olup belli bir hataya sahip olduğu varsayılır.

X; bağımsız (sebep) değişkeni olup hatasız ölçüldüğü varsayılır.

α; sabit olup X=0 olduğunda Y'nin aldığı değerdir.

β; regresyon katsayısı olup, X'in kendi birimi cinsinden 1 birim değişmesine karşılık Y'de kendi birimi cinsinden meydana gelecek değişme miktarını ifade eder.

ε; tesadüfi hata terimi olup ortalaması sıfır varyansı σ2 olan normal dağılış gösterdiği varsayılır. Bu varsayım parametre tahminleri için değil katsayıların önem kontrolleri için gereklidir.

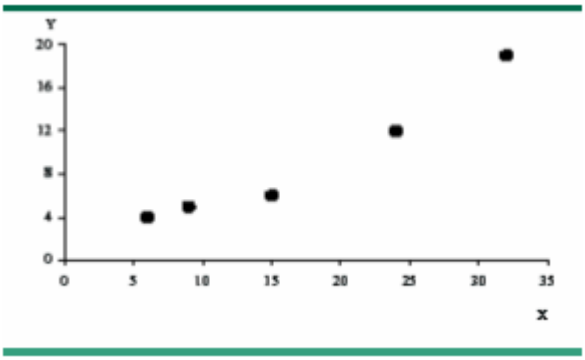
3.2.1 Parametrelerin (Katsayıların) Tahmini

Bir regresyon modeli oluşturulurken genelde en-küçük kareler ve en büyük olabilirlik (maximum likelihood) teknikleri olarak bilinen iki yaklaşımdan birisi kullanılır. Eğer hata teriminin normal dağılım göstermesi şeklinde bir varsayım varsa en büyük olabilirlik, hata teriminin dağılışı ile ilgili herhangi bir varsayım söz konusu değilse en-küçük kareler tekniği kullanılarak parametreler tahmin edilir. **En-küçük kareler tekniği** kullanılarak parametrelerin nasıl tahmin edildiğini örnek bir veri grubu üzerinde kısaca özetleyelim.

Tablo 3.1 Örnek Veri Tablosu

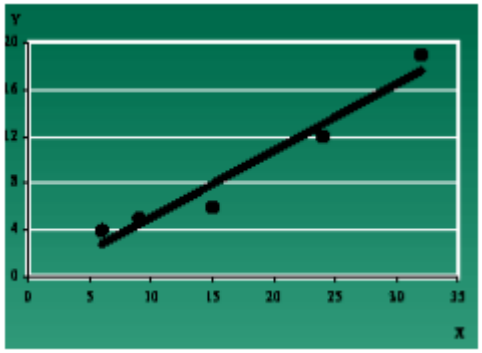
BOY(cm) X	ÇEVRE(cm) Y	
9	5	
15	6	
6	4	
24	12	
32	19	

Tabloda verilen X ve Y değişkenlerine ait beş gözlem çifti, koordinat eksenlerine yerleştirildiğinde elde edilen serpme diyagramının Şekil 3.1'deki grafik elde edilir.



Şekil 3.1 Gözlem Çiftlerinin Koordinat Eksenlerindeki Serpme Diyagramı

Şekil 3.1'de verilen noktaları temsil eden regresyon doğrusu oluşturulursa Şekil 3.2 elde edilir. Uydurulan regresyon doğrusu ile gözlem noktaları arasındaki fark hata (ε) olarak isimlendirilir. Regresyon doğrusuna ait parametreler öyle tahmin edilmelidir ki; doğru ile gözlem noktaları arasındaki fark (hata) en az olsun. Bunu sağlayacak teknik ise en-küçük kareler tekniğidir.

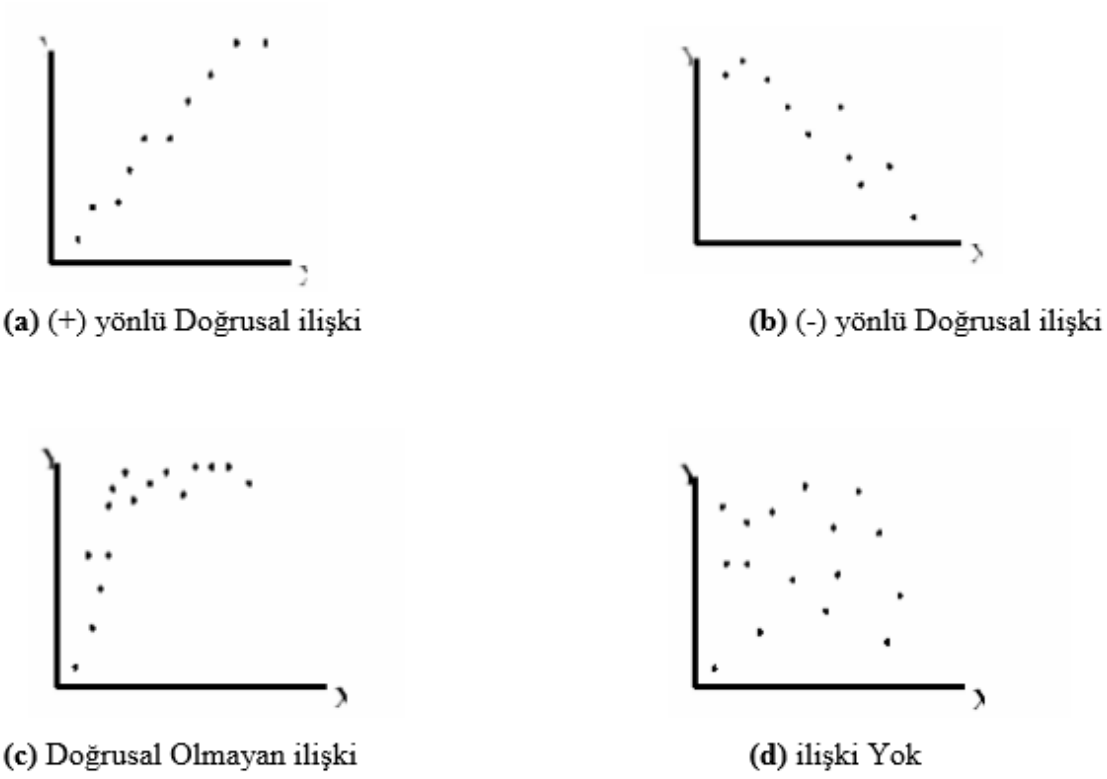


Şekil 3.2 Serpme Diyagramında Olan Noktaları Temsil Eden Regresyon Doğrusu

Regresyon analizi, bilinen bulgulardan, bilinmeyen gelecekteki olaylarla ilgili tahminler yapılmasına izin verir. Regresyon, bağımlı ve bağımsız değişken(ler) arasındaki ilişkiyi ve doğrusal eğri kavramını kullanarak, bir tahmin eşitliği geliştirir. Değişkenler arasındaki ilişki belirlendikten sonra, bağımsız değişken(ler)in skoru bilindiğinde bağımlı değişkenin skoru tahmin edilebilir.

Bağımlı Değişken (y): Bağımlı değişken, regresyon modelinde açıklanan ya da tahmin edilen değişkendir. Bu değişkenin bağımsız değişken ile ilişkili olduğu varsayılır. Bağımsız Değişken (x): Bağımsız değişken, regresyon modelinde açıklayıcı değişken olup; bağımlı değişkenin değerini tahmin etmek için kullanılır.

Değişkenler arasında doğrusal ilişki olabileceği gibi, doğrusal olmayan bir ilişki de olabilir. Bu nedenle, saçılım grafiği yapılmadan (ilişki yok/doğrusal ilişki var/doğrusal olmayan ilişki var) ve değişkenler arasında korelasyon varlığına rastlanmadan regresyon analizine karar verilmemesi gerekir. Bu bilgiler doğrultusunda, tek/çok değişkenli doğrusal regresyon analizlerinin yanı sıra, tek/çok değişkenli doğrusal olmayan regresyon analizleri de mevcuttur.



Şekil 3.3 Değişkenler Arasındaki Olabilecek İlişkileri Gösteren Grafikler

Örnek: Kardiyoloji kliniğine başvuran erkek hastalar üzerinde yapılan bir araştırmada, yas(x) ve kolesterol(y) değişkeni arasındaki korelasyondan yola çıkılarak kurulan regresyon modeli aşağıdaki gibi elde edilmiştir:

Y=3,42+0,326x

Bu modele göre, yastaki bir birimlik artısın, kolesterol değerinde 0.326 birimlik bir artışa neden olacağı, yeni doğan bir erkeğin (X=0) kolesterol değerinin ise 3.42 olacağı söylenebilir. Kurulan bu modele göre, 50 yasında bir erkeğin kolesterol değerinin ne kadar olacağını tahmin edebiliriz.

X=50 için :

Y=3,42+0,326*(50)=19,52 50 yasında bir erkeğin kolesterol değerinin 19.52 olacağı söylenebilir.

3.2.2 Tek Değişkenli Ve Çok Değişkenli Regresyon Analizi

Regresyon analizi bağımlı değişken ile bir veya daha çok bağımsız değişken arasındaki ilişkiyi incelemek amacıyla kullanılan bir analiz yöntemidir. Bir tek bağımsız değişkenin kullanıldığı regresyon tek değişkenli regresyon analizi, birden fazla bağımsız değişkenin kullanıldığı regresyon analizi de çok değişkenli regresyon analizi olarak adlandırılır.

Regresyon analizi ile bağımlı ve bağımsız değişkenler arasında bir ilişki var mıdır? Eğer bir ilişki varsa bu ilişkinin gücü nedir? Değişkenler arasında ne tür bir ilişki vardır? Bağımlı değişkene ait ileriye dönük değerleri tahmin etmek mümkün müdür ve nasıl tahmin edilmelidir? Belirli koşulların kontrol edilmesi durumunda özel bir değişken veya değişkenler grubunun diğer değişken veya değişkenler üzerindeki etkisi nedir ve nasıl değişir? gibi sorulara cevap aranmaya çalışılır.

Tek Değişkenli Regresyon Analizi: Tek değişkenli regresyon analizi bir bağımlı değişken ve bir bağımsız değişken arasındaki ilişkiyi inceler. Tek değişkenli regresyon analizi ile bağımlı ve bağımsız değişkenler arasındaki doğrusal ilişkiyi temsil eden bir doğrunun denklemi formüle edilir.

Çok Değişkenli Regresyon Analizi: İçinde bir adet bağımlı değişken ve birden fazla bağımsız değişkenin bulunduğu regresyon modelleri çok değişkenli regresyon analizi olarak bilinir. Regresyon analizi, birçok alanda veri analizi için başvurulan önemli bir istatistiksel teknik olup değişkenler arasındaki ilişkiyi açıklamak için kullanılır.

Kısaca regresyon analizi, bağımlı bir değişken ile bağımlı değişken üzerinde etkisi olduğu varsayılan bağımsız değişken(ler) arasındaki ilişkinin matematiksel bir model ile açıklanmasıdır.

Basit doğrusal regresyon analizinde bir bağımlı ve bir bağımsız değişken söz konusu iken, çoklu doğrusal regresyon analizinde ise bir bağımlı değişken varken iki ya da daha fazla bağımsız değişken vardır ve her iki analizde de değişkenler arsında doğrusal bir ilişki vardır. Ayrıca bağımlı ve bağımsız iki değişken arasında eğrisel bir ilişki var ise değişkenler arasındaki ilişki eğrisel regresyon modeli ile açıklanır. Korelasyon analizinde, değişkenlerin bağımlı ve bağımsız değişken olarak belirlenmesi hesaplamaların sonucu açısından önemli değilken, regresyon analizinde ise değişkenlerin hangisinin bağımlı hangisinin bağımsız değişken olduğunu tespit etmek çok önemlidir. Örneğin ,ithalat ve ihracat yapan bir işletmenin karlılık düzeyini işletmenin bulunduğu ülkedeki döviz kurları etkileyebilir halde, işletmenin karlılık düzeyi bağımlı(y),ülkedeki döviz kurları ise bağımsız(x) değişken olarak ele alınmalıdır.

3.3.3 Tek Değişkenli Regresyon Analizi

Tek değişkenli regresyon analizi bir bağımlı değişken ve bir bağımsız değişken arasındaki ilişkiyi inceleyen analiz tekniğidir. Bu analizle bağımlı ve bağımsız değişkenler arsındaki doğrusal (lineer) ilişkiyi temsil eden bir doğru denklemi formüle edilmektedir. Korelasyon analizinde olduğu gibi ,regresyon analizinde üzerinde durulan ilişki, değişkenler arasındaki doğrusal ilişkidir. Bu doğrunun hesaplanması ise en küçük kareler metodu yardımıyla yapılmaktadır. Regresyon analizi sonuçlarının yorumlanmasında birçok araştırmacı ve öğrenci tarafından ciddi hatalar yapılmaktadır. En yaygın hata, regresyon analizi sonuçlarının yorumlanmasında bağımsız değişkeninin y bağımlı değişkenine sebep olduğu şeklindeki yorumdur. Bağımsız değişkenlerin bağımlı değişkendeki değişimi açıklıyor olması sebepselliği gerekli kılmaz. Başka bir ifade ile, bağımlı ve bağımsız değişkenler arasında (pozitif ve negatif) bir ilişkinin olması her zaman bağımsız değişken(lerin) bağımlı değişkenin sebebi olduğu sonucunu doğurmayacaktır. İki değişken arasında bir ilişkinin olabilmesi için sebepsellik şart değildir.

İlişkinin sebebi belki de iki değişkenin üçüncü bir değişkenle olan ilişkilerinden kaynaklanıyor olabileceği gibi, söz konusu ilişki tamamen tesadüfi olarak da ortaya çıkmış olabilir. Sebepsellik ile ilişkiselliğin aynı şeyler olmadığı unutulmamalıdır. Regresyon analizi değişkenler arasındaki ilişkinin yapısı ve derecesi ile ilgilenmektedir

3.3.4 Çok Değişkenli Regresyon Analizi

Bir bağımlı değişken ve birden fazla bağımsız değişkenin yer aldığı regresyon modellerine çok değişkenli regresyon analizi denir. Çok değişkenli regresyon analizinde bağımsız değişkenler eş zamanlı olarak (aynı anda) bağımlı değişkendeki değişimi açıklamaya çalışmaktadır. Hesaplama ve yorum bakımından tek değişkenli regresyon analizine benzemektedir. Çok değişkenli regresyon analizinin yorumu tek değişkenli regresyon analizine benzemektedir. Ancak bazı farklılıklar vardır. Örneğin, tek değişkenli regresyon analizindeki karşılığı çoklu regresyon katsayısı R (multiple R) olarak ifade edilmektedir. Çoklu regresyon katsayısı R, bir bağımlı değişkendeki değişim ile eşzamanlı(aynı anda) ele alınan birden fazla bağımsız değişkendeki değişim arasındaki ilişkinin derecesini göstermektedir. Daha basit bir ifade ile , bağımlı değişken ile birlikte ele alınan bir grup bağımsız değişkendeki değişimin ilişkisinin (korelasyonunun) bir göstergesidir. Çok değişkenli regresyon analizi sosyal bilimlerin birçok dalında kullanım alanı bulmaktadır. Pazarlama, sosyoloji ve psikoloji gibi bilim dallarında davranışsal hareketlerin belirlenmesinde, ekonomide zaman serisi türü ekonomik değişkenleri etkileyen faktörlerin tespiti ve geleceğe yönelik projeksiyonlarında(tahmininde) kullanım alanı bulmaktadır.

3.3.5 Çoklu Regresyon Metodları

Enter metodu: Bağımsız değişkenleri bir blok olarak tek adımda girilip değerlendirildiği metod

İleri Doğru Seçim Metodu (Forward selection) :Bağımlı değişken ile en yüksek pozitif veya negatif korelasyonu olan bağımsız değişken ilkönce seçilir. Daha sonra girilen değişkenin katsayısının 0 olduğu hipotezi F testi ile yoklanır. Burada elde edilen F değeri, SPSS'in öngörülen F değerleri ile karşılaştırılır. SPSS'in iki F ölçütü vardır.

1. F değeri sizin belirleyeceğiniz minimum bir F değeri ile karşılaştırılır. Normal ayar 3,84'tür.
2. F istatistiği ile bağlantılı ihtimalin ayarlanması. Bunun normal ayarı da 0,05'tir. Eğer elde edilen F değeri bu ölçütlere eşit veya küçükse bağımsız değişken regresyon değerlendirmesine alınır ve seçim ileri doğru devam eder; yoksa işlem orada durdurulur. Bir değişken seçilip işleme alındığında, geride kalan bağımsız değişkenlerle bağımlı değişken arasındaki korelasyonlara bakılır ve en yüksek korelasyona sahip bağımsız değişken bir sonraki aday olur. Bu, aynı zamanda en geniş F değerine sahip değişkenin de seçimi olur.

Geriye Doğru Eleme (Backward Elimination) Metodu: İleri doğru seçimin tersine, burada önce bütün bağımsız değişkenler seçilir; sonra sıra ile belli ölçütlere göre eleme yapılır. SPSS; eleme için iki ölçüt koymaktadır.

1. Değişkenin formülde kalabilmesi için en küçük kareler F değeri normal ayar 2,71'dir.

2. En büyük F ihtimali (Probabability of F-to-remove,POUT).Bunun normal ayarı da 0,10'dur. İlk önce en küçük kısmi korelasyon katsayısına sahip değişken incelenir. Öngörülen değerlerden, büyük değere sahip değişken elenir.

Adım Adım Seçme (Stepwise Selection) Metodu: İlk olarak bağımsız değişken seçilir; eğer bu –ileri doğru seçmedeki- FIN veya PIN gereklerini yerine getirirse ikinci değişken seçilir; yoksa işlem orada biter. İkinci değişken olarak en yüksek kısmi korelasyona sahip değişken alınır. Seçimler yüksek korelasyondan düşüğe doğru yapılır. Bağımsız değişkenler ölçütlere uyarsa regresyon analizine başlanır. İlk değişken seçildikten sonra Adım Adım seçme, ileri doğru seçmeden farklılaşır. İlk değişkenin, geriye doğru elemedeki gibi FOUT veya POUT ölçütlerine uyup uymadığı kontrol edilir. Yani adım adım seçmede, hem ileri doğru seçme hem e geriye doğru eleme işlemleri yapılır.

3.3 En Küçük Kareler Metodu

En küçük kareler yöntemi, birbirine bağlı olarak değişen iki fiziksel büyüklük arasındaki matematiksel bağlantıyı, mümkün olduğunca gerçeğe uygun bir denklem olarak yazmak için kullanılan, standart bir regresyon yöntemidir.

Bir başka deyişle bu yöntem, ölçüm sonucu elde edilmiş veri noktalarına "mümkün olduğu kadar yakın" geçecek bir fonksiyon eğrisi bulmaya yarar. Regresyon fonksiyonun oluşturulmasını ve yeni değerin tahmin edilmesini sağlar.

Regresyon analizi yaparken en çok kullanılan yöntemlerden biri en küçük kareler yöntemidir. Büyük matematikçi C. F. Gauss'un 18 yaşındayken (1795) geliştirdiği bu yöntem, ilk kez 1801 de Cres astroidinin yörüngesinin belirlenmesinde kullanılmış ve ilk kez Gauss'un toplu eserlerinin yayınlandığı ciltlerden ikincisinde 1809 yılında yayınlanmıştır. Fransız matematikçi A. Legendre 1805 ve Amerikalı matematikçi R. Adrain de 1808 yıllarında aynı yöntemi Gauss'dan habersiz ve bağımsız olarak keşfetmişlerdir. En küçük kareler yöntemi, tıp, finans, mühendislik, ziraat, biyoloji ve sosyoloji gibi çeşitli bilim dallarında çeşitli değişkenler arasındaki ilişkiler belirlenirken kullanılan en önemli araçlar arasındadır.

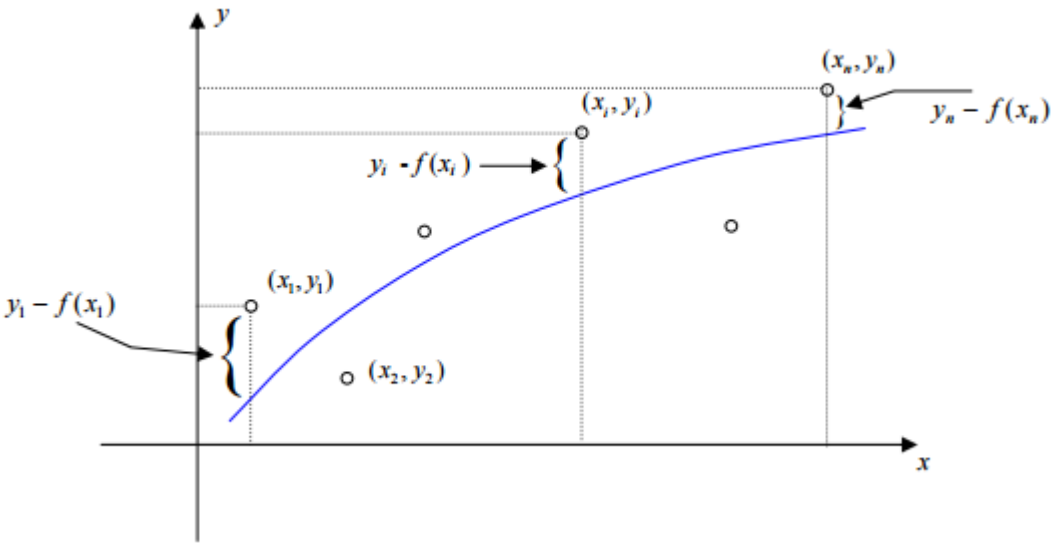
Belli ölçümler sonucunda i = 1, 2, . . . , n için (X_i, Y_i) verileri elde edilmiş olsun. Burada, her bir yi nin xi ye bağlı olarak değiştiği varsayılmaktadır. (X_i, Y_i) düzlemde noktalar olarak düşünüldüğünde, pratikte bu noktalar düzgün bir eğri üzerinde, başka bir deyimle, bilinen bir fonksiyonun grafiği üzerinde bulunmazlar. Hatta bazı durumlarda, (X_i, Y_i)'ler arasında ne tür bir bağıntı bulunduğu dahi bilinmeyebilir. Ancak, yapılan ölçümlerin doğası gereği, her i = 1, 2, . . . , n için Y_i = f (X_i) olacak biçimde bir fonksiyonun var olduğu, ölçümlerde yapılan hata nedeniyle bu eşitliklerin bazıları veya hepsinin sağlanmadığı kabul edilebilir. Bu düşünceyle, ölçülen yi değeri f (X_i) için yaklaşık değer kabul edilerek bu yaklaşımdaki hatanın minimum olduğu f fonksiyonu belirlenmeye çalışılır. Bu amacı gerçekleştirmek için f fonksiyonunun bir takım parametrelere bağlı bir ifadesi bulunduğu varsayılıp eldeki veriler yardımıyla bu parametreler belirlenmeye çalışılır.

Örneğin;

f fonksiyonu y = f (x) = mx + b ifadesinde olduğu gibi bir doğrusal fonksiyon veya

y = f (x) = ax 2 + bx + c ifadesinde olduğu gibi bir karesel fonksiyon olabilir ki bu durumda belirlenmesi gereken parametereler; a, b, c, m'dir.

Y_i değeri f (X_i) için yaklaşık değer, f $(X_i) \approx Y_i$, kabul edilince yapılan hata Y_i- f (X_i) dir ve amaç, bu hatalar minimum olacak şekilde bir f fonksiyonu bulmaktır.



Şekil 3.4 f Fonksiyonunun Gösterimi

Y_i - f (X_i) farklarından her birine bir artık denir.

En küçük kareler yönteminde aranan fonksiyon, ya da onun parametreleri, tüm artıkların kareleri toplamı olan

$$\sum_{i=1}^{n}(Y_{i-} f(X_i))^2 \qquad (3.2)$$

İfadesini minimum yapacak şekilde belirlenir. Bu, yönteme neden en küçük kareler yöntemi dendiğini açıklar.

Örnek:

Bir üretici, ürettiği ürünün çeşitli üretim seviyeleri için maliyetini belirliyor ve aşağıdaki tabloyu oluşturuyor:

Ürün Sayısı (X yüz)	2	5	6	9
Maliyet (Y bin TL)	4	6	7	8

Bu üretici için gider fonksiyonunu yukarıdaki tabloya en iyi uyan doğrusal fonksiyon olarak belirlenir. Eldeki veri tablosu, düzlemde şu (x,y) noktalarını verir:

(2,4) , (5,6) , (6,7) ve (9,8)

Bu noktaların hepsini üzerinde bulunduran bir doğru yoktur. Amaç, bu noktalara en iyi uyan doğruyu, yani regresyon doğrusunu bulmaktır. Regresyon doğrusunun denklemi y=C (x) = mx+b m ve b belirlenerek bulunacaktır. Artıkları hesaplanır ve veri tablosu aşağıdaki gibi genişletilir.

Ürün sayısı (x yüz)	2	5	6	9
Maliyet (y bin YTL)	4	6	7	8
$mx + b$	$2m + b$	$5m + b$	$6m + b$	$9m + b$
$y - mx - b$	$4 - 2m - b$	$6 - 5m - b$	$7 - 6m - b$	$8 - 9m - b$

Artıkların kareleri toplamı aşağıdaki iki değişkenli fonksiyonu tanımlar:

$F(m,b)=(4-2m-b)^2 + (6-5m-b)^2 +(7-6m-b)^2 +(8-9m-b)^2$

Bu fonksiyonun hangi m ve b değerleri için minimum değeri aldığını belirlenmeli. Kısmi türevleri hesaplanır:

$Fm(m,b)=2(4-2m-b)(-2)+2(6-5m-b)(-5)+2(7-6m-b)(-6) +2(8-9m-b)(-9)=0,$

$Fb (m,b)=2(4-2m-b)(-1)+2(6-5m-b)(-1)+2(7-6m-b)(-1) +2(8-9m-b)(-1)=0.$

Bir miktar aritmetik işlemden sonra aşağıdaki denklem sistemi elde edilir:

$146m+22b=152$

$22m+4b=25$

Bu sistemi yok etme yöntemi ile çözülür. İkinci denklem -11/2 ile çarpılıp birinci denkleme toplanırsa;

$25m = 14.5 \Rightarrow m = 0.58$ ve m nin bu değeri ikinci denklemde yerine konulursa

$12.76 + 4b = 25 \Rightarrow 4b = 11.24 \Rightarrow b = 3.06$

elde edilir. Görüldüğü gibi, sistemin tek çözümü vardır:

m = 0.58 , b = 3.06

m ve b'nin bu değerleri için F(m,b)'nin minimum olduğunu biliniyor. O halde regresyon doğrusu;

y=0.58x+3.06 elde edilmiş olur.

3.4 Hot Deck Algoritması

Hot Deck Imputation ile eksik veri değerlerini doldururken benzerlik tahmininde bulunmak için k-en yakın komşu en çok tercih edilen metodudur. Diğer bir deyişle eksik veri bulunduran satır ile tamamlanmış satır arasındaki uzaklık hesabı için k-en yakın komşu metoduyla yapılır.

Hot deck atfının en önemli dezavantajı, 'benzerlik' kavramının tanımlanmasındaki güçlüktür. Bu nedenle hot deck prosedürü kayıp veriler için standart bir yol sağlamamaktadır. Bu benzerliğin belirlenebilmesi için verici (donor) durumların seçimini başarabilecek bir yazılım gerekmektedir. Daha ileri bir hot deck algoritmasına göre, benzer bir kayıttan daha fazla sayıda kayıt belirlenir ve bu verici (donor) kayıtlardan biri kayıp değerlerin atfı için rassal olarak seçilir. Ayrıca eğer uygunsa, bu verici durumların ortalaması kayıp değerlerin atfı için kullanılır.

3.5 En Yakın k-Komşular (EYK) Algoritması

Eksik değerler benzerlik ölçütü vasıtasıyla tespit edilen en benzer kayıtların eksik değerlere karşılık gelen nitelik değerlerinin bir araya getirilmesiyle hesaplanmaktadır. Bu hesabın yapılabilmesi için benzerlik ölçütü ve en yakın k komşu adedinin belirlenmesi gerekmektedir. Diğer bir deyişle eksik veri bulunduran satır ile tamamlanmış satır arasındaki uzaklık hesabı için k-en yakın komşu metoduyla yapılır. Bunun için aşağıdaki adımlar uygulanır.

1) Veriler tamamlanmış ve tamamlanmamış (eksik) veri kümeleri olmak üzere ikiye bölünür.

2) Xi tamamlanmış veri kümesinin matrisidir. Xij i.durumun j.değişkenini ifade eder. Yi Tamamlanmamış veri kümesinin matrisidir. Yij i.durumun j.değişkenini ifade eder.

3)Her eksik veri içeren satır için öklid uzaklığı hesaplanır.

$$\text{Euclid(d)}= \sqrt{\sum_{j=1}^{n}(X_{kj-}\, Y_{ij})^2} \qquad (3.3)$$

Uzaklık hesabına göre eksik veri içeren tamamlanmamış satıra en yakın tamamlanmış satır belirlenir. Denkleme göre j ifadesi eksik değer içeren kaydın tam olan sütunlarını ifade etmektedir. Yani benzerlik ölçütü, eksik olan kaydın sadece tam nitelikleri ile o niteliklere karşılık gelen diğer komşuların nitelikler değerleri kullanılarak hesaplanmaktadır. Benzerlik ölçütü hesaplandıktan sonra k adet komşunun ağırlıklı olarak benzerlik oranları:

$$W_{ik}= \frac{\frac{1}{Euclid(d)}}{\sum_{k=1}^{K}\frac{1}{Euclid(d)}} \quad \text{elde edilir.} \qquad (3.4)$$

En yakın k adet komşuya olan uzaklık değerleriyle hesaplanan ağırlıklı benzerlik oranlarının toplamı 1 olmaktadır. Ağırlık değerleri (W_{ik})bulunduktan sonra eksik olan kaydın eksik nitelik değeri, o niteliğe karşılık gelen en yakın (k) komşu kayıtlarındaki niteliklerden yararlanılarak:

$$Y_{ij=}\sum_{k=1}^{K} W_{ik}\, X_{kj} \text{ elde edilir.} \qquad (3.5)$$

Denklemdeki Yij, i numaralı kaydın j sütunundaki eksik değeri, Wik, i kaydının k. komşu kayıtla olan ağırlıklı uzaklık ölçütü ve Xkj, k. komşunun j sütununda bulunan değerini ifade etmektedir

3.6 Naive Bayes ile Değer Atama Metodu

Naive Bayes sınıflandırıcı Bayes karar teorisine dayanan basit bir olasılıksal sınıflandırıcıdır. Herbir sınıf için olasılıkları hesaplar ve her bir örnek için olasılığı en yüksek sınıfı bulma eğilimindedir. Popüler olmasının sebebi sadece iyi performansı değil basit yapısı yüksek hesaplama hızı ve eksik verilere olan duyarsızlığıdır [1]. NBI Naive Bayesian Classifier kullanan tamamlama metodudur.

3.6.1 Önsel ve Sonsal Dağılımlar (Önsel ve Sonsal Olasılıklar)

Seçilen bir model için Bayesci istatistiksel analizin ilk aşaması önsel dağılımların tespit edilmesidir. Önsel dağılım; araştırmacının kendi görüşleri, benzer çalışmalar, uzman görüşleri vb. sayesinde elde edilebilen dağılımlardır ve iki gruba ayrılmaktadır. Bunlar:

3.6.1.1 Açıklayıcı Önsel Dağılım: Açıklayıcı önsel dağılım, olabilirlik fonksiyonu tarafından etkisi azaltılmayan önsel dağılımdır. Önceki çalışmaların meta analizinin yapılması, verinin histogramının çizilmesi veya kesikli önsel dağılımın kullanılması yardımıyla açıklayıcı önsel dağı- lımları belirlemek mümkün olmaktadır.

3.6.1.2 Açıklayıcı Olmayan Önsel Dağılım: Açıklayıcı olmayan önsel dağılım, çalışmalarda en yaygın kullanılan önsel dağılım türüdür ve parametrenin tanımlı olduğu aralık bilgisi dışında herhangi bir bilginin olmaması durumunda kullanılmaktadır. Bayesci yaklaşımda açıklayıcı olmayan önsel dağılım kullanıldığında, sonuçlar klasik yaklaşımla elde edilen sonuçlarla örtüşmektedir. Bunun nedeni, parametre tahmininde eldeki verinin sağladığı bilgi dışında farklı bir bilgi kullanılmamasıdır. Başlıca açıklayıcı olmayan önsel dağılımlar; düzgün, eşlenik, etkisiz ,yaygın/dağınık ve zayıf önsel dağılımlardır. Önsel bilgiler elde edildikten sonra bu bilgiler, araştırmacının gözlemlerine dayanan verilerden elde edilen ve olabilirlik fonksiyonu yoluyla olasılıksal olarak niceliksel hale getirilen bilgilerle birleştirilmekte ve sonsal olasılıklar elde edilmektedir. Sonsal olasılık, tahmin edilecek parametre hakkındaki tüm bilgileri içermektedir ve istatistiksel olarak şöyle gösterilmektedir.

Sonsal = Önsel * Olabilirlik

Naive Bayes sınıflandırıcı Bayes karar teorisine dayanan basit bir olasılıksal sınıflandırıcıdır. Her bir sınıf için olasılıkları hesaplar ve her bir örnek için olasılığı en yüksek sınıfı bulma eğilimindedir. Popüler olmasının sebebi sadece iyi performansı değil basit yapısı yüksek hesaplama hızı ve eksik verilere olan duyarsızlığıdır. NBI Naive Bayesian Classifier kullanan tamamlama metodudur.

Genellikle veri tabanlarında kayıp değer taşıyan özellik sayısı 1'den fazla olur. Bu durumda;
1. Tamamlama yapılacak ilk özellik tespit edilmeli.

2.Tamamlanacak özellikler için tamamlanma sıraları göz önünde bulundurulmalı.
3 farklı NBI stratejisi vardır:

Order Irrelevant Strategy (NBI-OI): Tamamlanacak özellikler tanımlandıktan sonra veri kümesi eksik değerlerinin tamamlanmasının sırasıyla ilgisi yoktur. Tamamlanmış özelliklerin değerleri daha sonraki özellikler için kullanılmaz. Kayıp değerleri tamamlanmış veri kümeleri tüm farklı tamamlama sıraları için aynıdır.

Order Relevant Strategy (NBI-OR): Tamamlanacak veri kümesi eksik değerlerinin tamamlanma sırasıyla ilgilidir. Tamamlanmış özelliklerin değerleri daha sonraki özellikler için kullanılır. Kayıp değerleri tamamlanmış veri kümeleri tüm farklı tamamlama sıraları için farklıdır.

Hybrid Strategy (NBI-Hm): İlk iki stratejinin birleşimidir. Birinci özellik tamamlama adımında sıralı stratejiyi kullanır kalanında sırasız stratejiyi kullanır.

NBI bu üç stratejiden de anlaşılacağı üzere 2 adımdan oluşur:

1.adım tamamlanacak özellikleri ve sırasını tanımlamak. Kayıp değerlere sahip özellikler birden fazla olabilir. Bu kayıp değerlerli özellikler arasında önceliğe iki açıdan bakılabilir. Birincisi kayıp değerlerin oranı(missing proportion), ikincisi özelliğin önem faktörü(important factor) dür.

2.adım kayıp veriler için NBI modelini kullanmak. Sıralı stratejide her adımda tamamlanan kayıp değerler ile değişen veri kümesi kullanılır.

3.7 C4.5 Karar Ağacı Algoritması

ID3 algoritmasını geliştiren Quinlian'ın geliştirdiği C4.5 karar ağacı oluşturma algoritmasıdır.ID3 algoritmasında bazı eksiklikler ve sorunlar vardır. Bu sorunlar C4.5 algoritması ile giderilmiştir.

C4.5 ağacının ID3 ağacından en büyük farkı normalleştirme (normalization) kullanıyor olmasıdır.

Yani ID3 ağacı üzerinde entropi hesabı yapılır (veya bilgi kazanımı (information gain)) ve bu değere göre karar noktaları belirlenir. C4.5 ağacında ise entropi değerleri birer oran olarak tutulur. Ayrıca ağaç üzerinde erişim sıklıklarına göre alt ağaçların (subtree) farklı

seviyelere taşınması da mümkündür. C4.5 ağacının diğer bir farkı ise tam bu noktada ortaya çıkar ID3 ağacının yaklaşımından farklı olarak C4.5 ağacında budama (prunning) işlemi yapılmaktadır.

C4.5 Algoritmasının çalışma şekli:

Her adımda bütün özellikler kontrol edilir
Her özelliğin normalize edilmiş bilgi kazanımı (information gain) hesaplanır
En iyi bilgi kazanımını veren özellik karar ağacında karar olarak taşınır.
Ardından bu yeni karar düğümünün altında bir alt liste oluşturularak alt karar ağacı inşa edilir.
Örnek olarak aşağıdaki veri kümesini (dataset) ele alalım:

Tablo 3.2 Örnek Dataset

Özellik1	Özellik2	Özellik3	Sınıf
Ali	70	Geçti	1
Ali	90	Geçti	2
Ali	85	Kaldı	2
Ali	95	Kaldı	2
Ali	70	Kaldı	1
Evren	90	Geçti	1
Evren	78	Kaldı	1
Evren	65	Geçti	1
Evren	75	Kaldı	1
Şadi	80	Geçti	2
Şadi	70	Geçti	2
Şadi	80	Geçti	1
Şadi	80	Kaldı	1
Şadi	96	Kaldı	1

Yukarıdaki veri kümesi için bir C4.5 ağacı oluşturmak istiyor olunsun. Yapılacak ilk adım bilgi kazanımını (information gain) hesaplamaktır. Bilgi kazanımı hesaplanırken, o anda veri kümesinde bulunan bütün veriler ve hesaplanması istenen belirli bir verinin üzerinden gidilir. Bu hesaplaması yapılacak olan belirli veriye örnekleme (misal, sampling) ismi verilir ve bütün veri kümesi üzerinden bu örneklemeye ait hesaplama yapılır.

Bilgi(M)= $-\sum_{i=1}^{k}(frekans(S_{i,}\text{M}) / |\text{M}|).\log_2(frekans(S_{i,}\text{M})/ |\text{M}|))$ (3.6)

Bilgi (information) hesaplaması sırasında kullanılacak olan formül yukarıdaki şekildedir. Buna göre herhangi bir misal (M ile gösterilmiştir) için o sınıftaki (S ile gösterilmiştir) değerlere göre frekansına bakılır. Ayrıca yukarıdaki formülde |M| değeri, o sınıftaki misallerin sayısını ifade etmektedir.

Yukarıdaki şekilde her örnek için bilgi değeri hesaplandıktan sonra kazanım (gain) hesaplanması mümkündür.

Genelde tam bu adımda bilgi parçalara bölünür ve bölünen parçalar (partition) üzerinden işlem yapılır. Bu durum için ise hesaplama aşağıdaki şekilde yapılabilir:

$Bilgi_x$(P)=$\sum_{i=1}^{n}((|P_i|/ |\text{P}|).$ Bilgi(P_i) (3.7)

Yukarıdaki formülde her bir i parçası için yapılan bilgi hesaplaması verilmektedir. Kazanım ise bu durumda aşağıdaki şekilde hesaplanabilir:

Kazanım(Özellik X)=Bilgi (P)-$Bilgi_x$(P) (3.8)

Örnekte sınıf değerinin bilgi kazanımını (information gain) hesaplamak istiyor olunsun. Yukarıdaki formüle göre, 14 toplam satırdan 5 tanesi sınıf 2 ve 9 tanesinin sınıf 1 olduğunu dikkate alarak aşağıdaki eşitlik yazılır. Önce bilgi değerlerini hesaplanır sonra da kazanımı bulunur:

Bilgi (P)= -9/14*$\log_2\left(\frac{9}{14}\right)$ – 5/14*$\log_2(\frac{5}{14})$

=0.940 bit

İlk bilgi değeri bütün parçanın hesaplandığı yani 14 satırın tamamının dikkate alındığı ve 9/14 ve 5/14 olarak iki ihtimalin hesaba katıldığı durumdur. Bu durum aynı zamanda entropi olarak da düşünülebilir.

İkinci bilgi hesabında özellik 1 kullanılır. Buna göre veri kümesinin ilk 5 satırında Ali, sonraki 4 satırında Evren ve son 5 satırında Şadi özellikleri var. Buna göre tabloyu 3 parçaya bölünür:

Özellik1	Özellik2	Özellik3	Sınıf
Ali	70	Geçti	1
Ali	90	Geçti	2
Ali	85	Kaldı	2
Ali	95	Kaldı	2
Ali	70	Kaldı	1
Evren	90	Geçti	1
Evren	78	Kaldı	1
Evren	65	Geçti	1
Evren	75	Kaldı	1
Şadi	80	Geçti	2
Şadi	70	Geçti	2
Şadi	80	Geçti	1
Şadi	80	Kaldı	1
Şadi	96	Kaldı	1

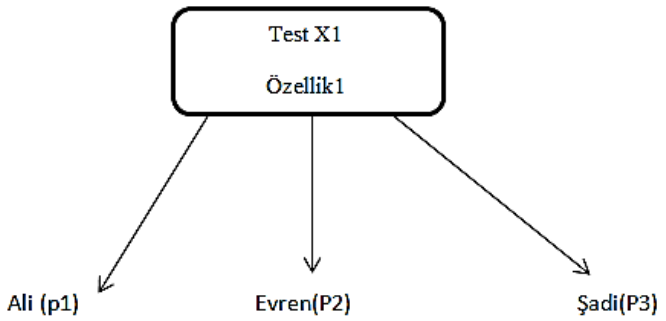
Yukarıdaki yeni tabloya göre her özellik parçasının ayrı ayrı hesaplanarak denklemde yerine yazılması gerekir:

$Bilgi_{x1}$(T)=0.694 bit

Kazanım(X1)= 0.940-0.694=0.246 bit

Olarak bulunur.

Yukarıda bulunan bilgi kazanımı, bütün veri kümesindeki Özellik1 için bütün sınıflar arasındaki kazanımı göstermektedir. Bu değerleri kullanarak aslında veri kümesinin 3 parçaya bölmüş ve her birisi için bilgi kazanımının olası değerini hesaplanmış oluyor.



Öz2	Öz3	Sınıf
70	Geçti	1
90	Geçti	2
85	Kaldı	2
95	Kaldı	2
70	Kaldı	1

Öz2	Öz3	Sınıf
90	Geçti	1
78	Kaldı	1
65	Geçti	1
75	Kaldı	1

Öz2	Öz3	Sınıf
80	Geçti	2
70	Geçti	2
80	Kaldı	1
80	Kaldı	1
96	Kaldı	1

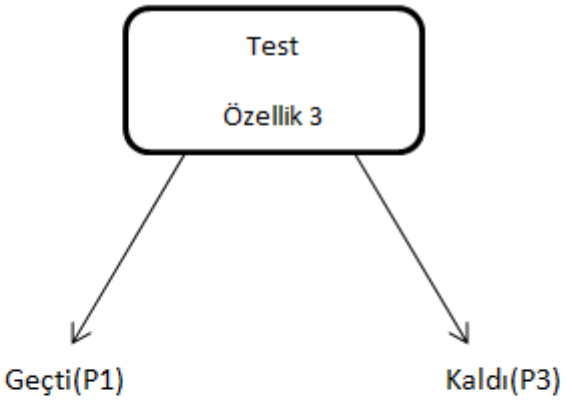
Yukarıdaki gösterim karar ağacının alabileceği olasılıklardan birisidir ve ağacın özellik 1'de bulunan isimlere göre karar noktası oluşturulması halinde alacağı vaziyeti gösterir. Şimdi aynı hesap, Özellik 3 için yapılır. Burada geçti / kaldı ihtimalleri bulunuyor dolayısıyla ağaç 2 dala ayrılabilir. Önce hesap yapılır:

Bilgi(P)=0.940 bit

$Bilgi_{ö3}(P)$=0.892 bit

Kazanım(Ö3)=0.940 – 0.892=0.048 bit

Özellik 3 için 3/6 ve 6/8 olmak üzere iki parça bulunuyor ve her ikisi için de hesap yapılıp toplam bilgiden çıkarıldığında kazanım değeri olarak 0.048 bit bulunuyor. Bu değer yorumlanmadan önce ağacın şu anda hesaplanan halini gösterilirse:



Öz1	Öz2	Sınıf
Ali	70	1
Ali	90	2
Evren	90	1
Evren	65	1
Şadi	80	2
Şadi	70	2

Öz1	Öz2	Sınıf
Ali	85	2
Ali	95	2
Ali	70	1
Evren	78	1
Evren	75	1
Şadi	80	1
Şadi	80	1
Şadi	96	1

Yukarıdaki şekilde ağacın ilk karar noktasını Geçti / Kaldı ihtimalleri üzerine kurulacak olursa bilgi kazanımı olarak 0.048 beklenmektedir.

Bu durumda C4.5 ağacı en yüksek kazanıma sahip olan değeri alacaktır. Bu değer Özellik 1 için isimler olduğundan karar ağacının bu adımda Özellik 1'e göre karar noktası eklemesi yerinde olacaktır.

Ardından diğer adımlar için benzer şekilde hesaplamalar yapılarak ağacın karar noktaları oluşturulmaya devam edecektir.

C4.5 Ağacının önemli bir diğer özelliği ise budama işlemidir. Esas olarak ağaçlarda iki tip budama yapılabilir. Birisi ön budama (preprunning) diğeri ise son budama (post prunning). C4.5 ağacı son budama (postprunning) yöntemini tercih etmektedir.

Hemen burada karar ağaçlarında (decision trees) ön budama ve son budamanın nasıl yapıldığından bahsedelim. Ön budama, genelde ağaç oluşturulurken bazı dalların oluşturulmaması yönündedir. Örneğin bazı dallar anlamsız olacağından veya hiç eleman içermeyeceğinden oluşturulmaz. Son budama ise ağacın bütün dallarını oluşturur ve sonra bazı şartlara göre budama yapar. Burada da eleman içermeyen dallar budanabileceği gibi, bazı durumlarda istatistiksel yaklaşımlar da kullanılabilir. Örneğin yukarıdaki veri kümemizi Özellik 2'yi kullanarak 10luk dilimlere bölmek istenilirse 100-90 arasında 90-80 ve 80-70 arasında verilerimiz olacak ancak 70'in altında veri olmayacak. Bunu verilere bakmadan anlaşılmaz. Şayet illaki ağaç oluşturulacaksa ve kural 0 ile 100 arasındaki notların 10'arlık dilimler halinde bölünmesi ise bu dalların da ağaçta yer alması gerekir ancak hiç veri içermeyeceği için bu dallanmalar anlamsız olacak ve budanacaktır.

C4.5 Karar ağacı algoritması ile kayıp verilerin tahmini yapılabilmektedir. Bunun için aşağıdaki adımlar izlenir;

T çalışılan veri kümesi ve genel bilgi kazancı bilgi(T) olsun.

X bu kümenin herhangi bir özelliği olsun ve X özelliğinin bilgi kazancı ise bilgiX(T) olsun.

Bilgi(T) hesap edilir. Ancak bilgiX(T) hesap edilirken olmayan veriler bu kümeden çıkarılır. Olay sayısı n ile ifade edilirse ve bilinmeyen veriler b ile ifade edilirse X özelliğin (n-b) adet eksik olmayan verileriyle sanki hiçbir veri eksik değilmiş gibi klasik formül uygulanır.

Ardından eksik olmayan değerlerin toplam değerlere oranı:

$$F= (n-b) / n \qquad (3.9)$$

Formülü ile hesaplanır. Bu durumda; F = (bilgi(T) - bilgiX(T)) formülü ile bilgi kazancı hesaplanmış olur. Yukarıdaki işlemi tüm eksik veri içeren satırlar için tekrarladığımızda

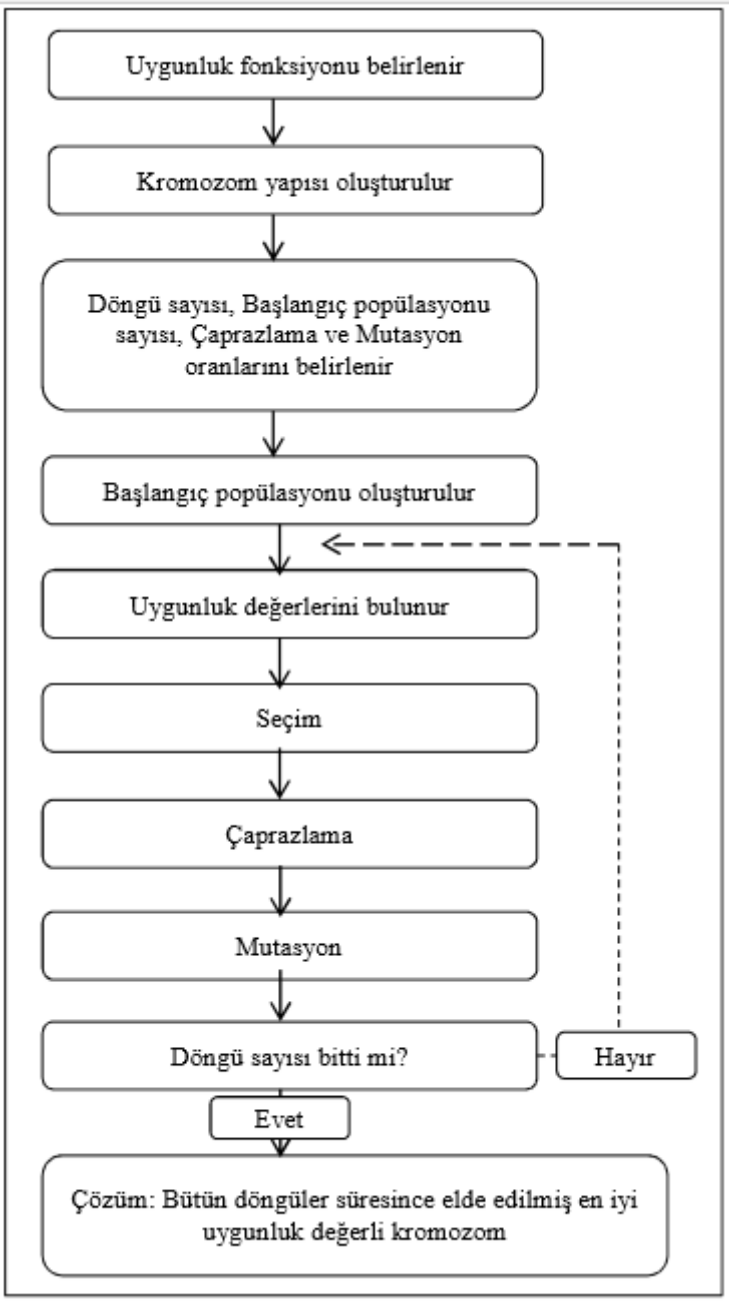
bir tablo elde edilecektir. Elde edilen tabloyu kullanarak her nitelik için geçerli sayıları ve kazanç değerlerini bulabiliriz. Bu tablo ile Karar Ağacı yapısı oluşturulur.

3.8 Genetik Algoritmalar(GA)

1975 yılında, John Holland birey havuzundan en uygun bireyin hayatta kalma kanunundan yararlanan doğal seçim yöntemi olan genetik algoritmaları (GA) tanıtmıştır. GA'ın temel prensibi aile neslinden daha iyi türler seçmek ve rastgele genleri karşılıklı değiştirerek daha iyi nesil üretmektir. GA'ın amacı doğal seçim prensiplerini taklit eden rehber eşliğinde çözüm uzayında daha iyi çözümü bulmaktır. Birkaç nesil sonra uygun olmayan genler elenerek daha uygun genler üretilmeye başlanmaktadır. GA çözüm uzayını gezmek ve en iyi çözümü bulmak konusunda dengeleme yapmaktadır. Fakat yine de bazı durumlarda çözüm uzayı tam gezilmeden en iyi çözüm bulunduğu durumlar oluşabilmektedir. Genetik algoritmalar genelden özele veya basitten karmaşık olana doğru giden ve geniş alanda kullanılan etkili bir arama tekniğidir. Bu doğal yöntem optimizasyon problemleri için kullanılmaktadır. Bu yüzden son yıllarda GA eksik değer hesaplama problemlerinde kullanılmıştır.

3.8.1 Genetik Algoritmalar Süreci

Genetik algoritmalar bir popülasyon havuzundan yeni popülasyon havuzu oluşturma sürecidir (Şekil 3.6). Bu süreç seçim, çaprazlama ve mutasyon temel aşamaları şeklinde devam etmektedir. En başta uygunluk fonksiyonu belirlenmekte ve kromozom yapısı oluşturulmaktadır. Daha sonra döngü sayısı, başlangıç popülasyon sayısı, çaprazlama ve mutasyon oranları belirlenmektedir. Başlangıç popülasyon sayısına göre başlangıç popülasyon havuzu oluşturulmakta ve popülasyon havuzundaki bireylerin uygunluk fonksiyonu değerleri hesaplanmaktadır. En sonda ise seçim işlemi gerçekleştirilmekte çaprazlama ve mutasyon işlemleri gerçekleştirilmektedir. Böylece genetik algoritmaların bir döngü süreci tamamlanmış olmaktadır. Genetik algoritmaların sonlanma kriteri olan döngü sayısı kadar algoritma bu şekilde devam etmekte ve sonlanmaktadır.



Şekil 3.5 Genetik Algoritmaların Çalışmasının İş Akışı

Başlangıç popülasyon havuzunun oluşturulması

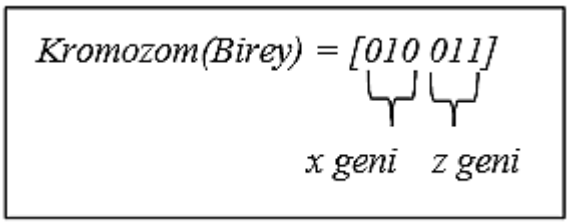
Genetik algoritmalarda kromozom ya da birey olarak adlandırılan yapılar çözümü istenen problemdeki Bağıntı(3.10)'daki gibi değişkenlerin birleşimden oluşmaktadır. Her bir değişkene gen adı verilmektedir.

Örneğin Bağıntı(3.11)'deki verilen y uygunluk fonksiyonunun maksimum olduğu değerin bulunması istendiğinde x ve z değişkenlerinin alabileceği değerler aralığı göz

önünde bulundurularak, Şekil 3.7'deki gibi ikili sayı sistemi ile oluşturulan bir kromozom yapısı oluşturulmaktadır.

Kromozom=[Değişken1 Değişken2…DeğişkenN] (3.10)

$Y=ax + bz,\ 1< x, z <10$ (3.11)



Şekil 3.6 Genetik Algoritmalarda Kromozom Yapısı

Genetik algoritmaların çalışabilmesi için başlangıç popülasyon sayısının başta belirlenmesi gerekmektedir. Başlangıç popülasyon sayısı çözüm havuzunda kaç adet kromozom yani bireyin olacağını belirleyen bir sayıdır. Bu sayı çok küçük olursa lokal minimum problemi olabilmektedir. Çok büyük olması durumunda ise çözüme ulaşılması çok fazla zaman alabilmektedir. Başlangıç popülasyon havuzu çözüm olabilecek bireyleri yani kromozomları değişken sınırları içinde rastgele bir şekilde üretilmesi sonucu oluşturulmaktadır. Rastgele bu üretim çözüme erişilmesinde bazen gecikmelere neden olabileceği gibi aynı zamanda tüm çözüm uzayını tarama imkanı da sunmaktadır.

Uygunluk fonksiyonu

Genetik algoritmaların, optimize yapılması istenen problemin çözümüne ulaşılıp ulaşılmadığını derecelendiren fonksiyona uygunluk fonksiyonu denmektedir. Genetik algoritmalar bir problemin çözümü için en düşük ya da en büyük uygunluk değerini bulmayı amaçlamaktadır. Karşılaşılan problemi en iyi bir şekilde ifade eden uygunluk fonksiyonunun belirlenmesi, yapılacak genetik algoritmalar çalışması bakımından oldukça önemli bir konudur. Her bir genetik algoritmalar döngüsü sonucunda uygunluk fonksiyonu tekrar tekrar hesaplanmakta böylece o anda bulunan bireylerin problemin çözümü için ne derece uygun olduğunun tespiti yapılmaktadır.

Uygunluk fonksiyonunu en küçük değere ulaştıran çözümün bulunması istendiğinde en büyük çözüm amaçlı uygunluk fonksiyonu -1 sayısı ile çarpılarak kullanılmaktadır.

Seçim

Birey havuzu oluşturulduktan sonra problemin çözümü için uygunluk değeri yardımıyla her bir bireye bir skor verilmektedir. Buna göre daha uygun olan bireyler havuzda kalarak güçlü olmayan bireyler havuzun dışında bırakılmaktadır. Seçim için rulet tekerleği, turnuva seçimi, sıra seçimi ve seçkinlik yöntemlerinden biri tercih edilmektedir.

$$Rulet\ tekerleği\ seçim\ değeri = \frac{Kromozom\ uygunluk}{Toplam\ uygunluk} \quad (3.12)$$

Rulet tekerleği seçimi değerlerinin oluşturulmasında Denklem 4.13 kullanılarak her bireye ait seçilme oranları belirlenmektedir. Kümülatif toplam haline getirilen seçim değerleri Tablo 3.3 ve Şekil 3.8'de gösterilmiştir.

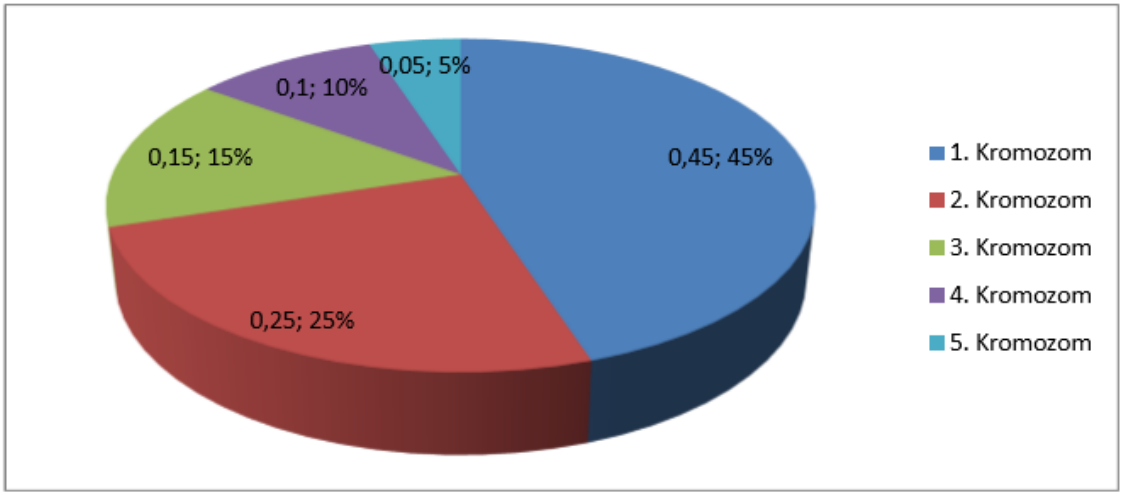
Tablo 3.3 Genetik Algoritmaların Seçim Aşamasında Rulet ve Sıra Değerleri

Burada başlangıç popülasyon sayısı kadar rastgele 0 ile 1 arasında üretilen değerler yardımıyla kümülatif değerlere göre yeni nesil bireyler oluşturulmaktadır. Rastgele

Kromozom	Uygunluk Değeri	Rulet Tekerleği Seçimi	Rulet Kümülatif	Sıra Seçimi	Sıra Kümülatif
1.Kromozom	45	0.45	0.45	0.33	0.33
2.Kromozom	25	0.25	0.70	0.26	0.59
3.Kromozom	15	0.15	0.85	0.20	0.80
4.Kromozom	10	0.10	0.95	0.13	0.93
5.Kromozom	5	0.05	1.00	0.06	1.00

üretilen sayı ilk kümülatif değerden itibaren sırasıyla karşılaştırılmaktadır. Eğer rastgele sayı kümülatif değerden küçük ise karşılık gelen kromozom seçilmektedir. Eğer havuzda uygunluk değeri diğerlerine göre daha yüksek birey ya da bireyler varsa rulet tekerleğinde çok fazla yer işgal etmektedir. Böylece bu uygun bireyler birkaç döngü sonra havuzda baskın hale gelmektedir.

Sıra seçiminde ise bireyler uygunluk fonksiyonu değerlerine göre büyükten küçüğe sıralanmakta ve sıralarına uygun kümülatif değerlere göre seçim işlemi gerçekleştirilmektedir. Kümülatif değerler sıralamaya göre verildiği için rulet tekerleğine göre her seferinde daha fazla çeşitlilik olmakta bu yüzden çözüme geç ulaşılmaktadır.



Şekil 3.7 Genetik Algoritmalarda Rulet Tekerleği Uygunluk Değerleri ve Yüzdesi Gösterimi

Sıra seçim değerleri Bağıntı(3.13)'e göre hesaplanmaktadır. Denklemdeki N en büyük sıra değerini göstermektedir. Turnuva seçimi ise karşılıklı iki bireyden uygunluk fonksiyonu daha yüksek olan bireyin havuzda kalması diğerinin ise çıkarılmasıyla yapılmaktadır. Seçkinlik yöntemi belli orandaki en iyi birey ya da bireylerin popülasyon havuzuna direk olarak seçilmesiyle gerçekleştirilmektedir.

$$Sıra\ Seçim\ Değeri = \frac{((N+1)-Kromozom\ sıra)}{(N*(N+1))/2} \tag{3.13}$$

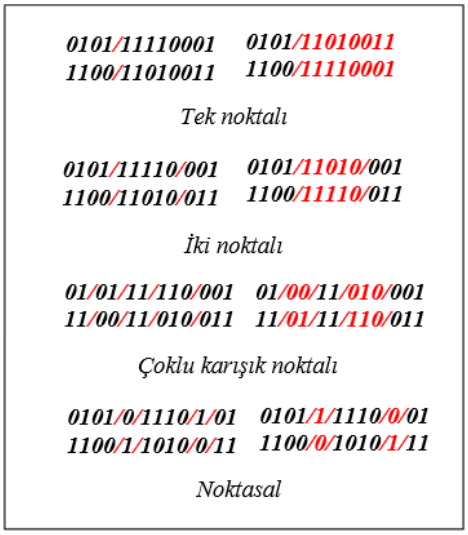
Genetik algoritmalar her uygunluk değeri hesaplamasından sonra çalışma süresince gelmiş geçmiş en iyi olan bireyi saklamaktadır. Her döngü sonucunda yeni nesilde daha iyi bir birey elde edilmiş ise bu birey en iyinin yerini alarak genetik algoritmalar sonlanana kadar saklanmaktadır. Bu saklanan en iyi birey genetik algoritmaların elde ettiği optimum çözüm olarak adlandırılmaktadır.

Genetik operatörler

Genetik algoritmaların başarısı var olan geçerli popülasyondan seçilmiş üyelerin karşılıklı olarak bilgilerinin değiştirilmesi ve mutasyona maruz kalmalarına bağlı olarak değişmektedir. Birey bilgilerinin karşılıklı değişmesini sağlayan çaprazlama operatörü ve bireyin mutasyon sonucu değişmesini sağlayan operatör ise mutasyon operatörü olarak

adlandırılmaktadır. Bu operatörler belirli yüzdelik ya da bindelik oranlarda bireyler veya bitler üzerinde uygulanmaktadır.

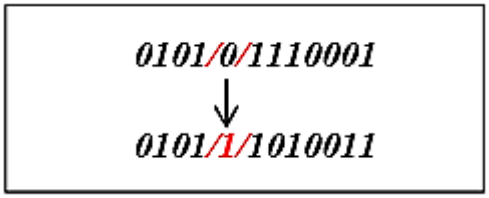
Çaprazlama operatörü anne baba olarak adlandırılan bit dizelerinden bilgileri almakta ve iki yeni birey üretmektedir. Belirli bir bit sırasından itibaren anne ve baba bireylerinin bit dize değerleri karşılıklı olarak yer değiştirilmektedir. Şekil 3.9'da gösterildiği gibi çaprazlama tek noktalı, iki noktalı, çoklu karışık noktalı ve noktasal olarak yapılmaktadır. Tek noktadan yapılan çaprazlama ile ebeveyn bireyleri belirlenmiş bir bit sırasından itibaren karşılıklı olarak diğer bitlerle yer değiştirmektedir. İki noktalı çaprazlama ise iki nokta olarak belirlenmiş başlangıç ve bitiş yer değişim sırası boyunca karşılıklı olarak bitler yer değiştirmektedir. Çoklu karışık noktalı çaprazlama bir veya birden çok bit değerleri karşılıklı olarak yer değiştirmektedir. Noktasal çaprazlama ise tek bir bit değeri diğer ebeveynin aynı pozisyonundaki bit değeri ile yer değiştirmekte yeni iki adet birey oluşmasını sağlamaktadır.



Şekil 3.8 Çaprazlama Yöntemleri

Şekil 3.10'da mutasyon operatörü gösterilmektedir ve bir birey üzerinde meydana gelen değişiklikleri ifade etmektedir. Mutasyon işlemi çaprazlama operatörüne göre çok daha küçük bir değişime sebep olmakla birlikte

güçlü olanın ayakta kalma ve yeni nesil oluşturma prensibi için bir aşama olarak görülmektedir.



Şekil 3.9 Mutasyon Operatörü

3.8.2 Genetik Algoritmaların Örnek Uygulaması

Matematikte türevi alınabilen bir fonksiyonun en büyük ya da en küçük yapan değeri fonksiyonun türevi alınarak bulunmaktadır. Fakat türevi alınamayan fonksiyonların çözümü için genetik algoritmalar kullanışlı bir hale gelmektedir. Örneğin; $y = 24x - 3x^2 + 5$, 0< x <16 denklemini en büyük yapan x değerini bulmak için fonksiyonun matematiksel türevi $\frac{dy}{dx} \rightarrow 24 - 6x = 0$ olmaktadr. Bu eşitlikten x=4 olarak y fonksiyonunu en büyük yapan değer olarak bulunur. Aynı fonksiyon genetik algoritmalar ile çözülmek istendiğinde ilk başta uygunluk fonksiyonu tespiti yapılmalıdır. Uygunluk fonksiyonu olarak y fonksiyonunun kendisini kullanılmıştır. Fakat gerçekte bu uygunluk fonksiyonu üzerinde çalışılan probleme uygun belirlenmelidir. Örneğin en uygun banka kredi maliyet hesabı ya da nesnelerin bir bölgeye en uygun şekilde yerleştirilmeleri için kullanılacak uygunluk fonksiyonu yapılacak çalışmanın amacına hizmet edecek şekilde oluşturulmalıdır. Uygunluk fonksiyonu seçiminden sonra başlangıç popülasyon sayısı belirlenmektedir. Çözümü istenen y fonksiyonu için 5 adet kromozom tercih edilmiştir. Kromozomdaki x değişkeni 1 ile 15 arasında tam sayı değerleri almaktadır. Bu nedenle bir kromozom ikili sayı düzeninde 4 bit uzunluğunda bir sayı ile temsil edilmiştir. Genetik operatörler çaprazlama oranı bu örnek için 0.6 ve mutasyon oranı 0.01 olarak alınmıştır. Çizelgedeki gibi başlangıç popülasyonu kromozom değerleri için rastgele 1 ile 15 arasında 5 adet sayı üretilmesiyle oluşturulurmuştur. Daha sonra Tablodaki gibi bu kromozomlara ait uygunluk fonksiyonu değerleri hesaplanmıştır.

Örneğin 1.kromozom uygunluk değeri x=2 için y=41 hesaplanmıştır. Diğer değerler de hesaplanarak Tablo 3.4'te verilmiştir.

Tablo 3.4 Başlangıç Popülasyonu

Kromozom	Onluk Sayı Değerleri	İkili Sayı Değerleri
1.Kromozom	2	0010
2.Kromozom	11	1011
3.Kromozom	7	0111
4.Kromozom	14	1110
5.Kromozom	5	0101

Tablo 3.5'deki gibi uygunluk değerlerinin negatif olması durumunda rulet tekerleği seçilme oranları negatif olamayacağından dolayı bu değerleri pozitife çevirmek için bütün uygunluk değerleri çok büyük bir sabit sayı ile örneğin 1000 ile toplanmıştır.

Tablo 3.5 Uygunluk Fonksiyonu Değerleri

Kromozom	Onluk Sayı Değerleri	İkili Sayı Değerleri	Uygunluk Değerleri
1.Kromozom	2	0010	41
2.Kromozom	11	1011	-94
3.Kromozom	7	0111	26
4.Kromozom	14	1110	-247
5.Kromozom	5	0101	50

Ayrıca normalizasyon ile [0, 1] aralığına dönüştürülerek kullanılabilir. Bu şekilde elde edilmiş rulet tekerleği kümülatif değerleri Tablo 3.6'da verilmiştir.

Tablo 3.6 Rulet Tekerleği Kümülatif Değerleri

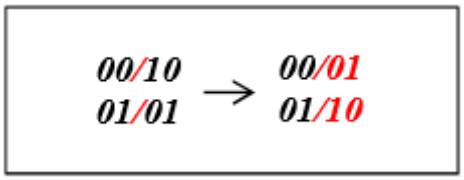
Kromozom	Sayı Değerleri	İkili Sayı Sistemi	Düzeltilmiş Uygunluk Değerleri	Rulet Tekerleği Seçimi	Rulet Tekerleği Kümülatif
1.Kromozom	2	0010	1041	0.2180	0.2180
2.Kromozom	11	1011	906	0.1897	0.4077
3.Kromozom	7	0111	1026	0.2148	0.6225
4.Kromozom	14	1110	753	0.1577	0.7802
5.Kromozom	5	0101	1050	0.2198	1.00
			Toplam: 4776	Toplam: 1.00	

Rulet tekerleği kümülatif değerleri elde edildikten sonra seçim aşamasına geçilmiştir. Seçim aşamasında popülasyon sayısı kadar yani 5 adet rastgele 0 ile 1.00 değerleri arasında ondalık sayı üretilmiştir ve ilk bireyden başlayarak kümülatif değerler ile üretilen sayı karşılaştırılmıştır. Eğer rastgele sayı karşılık gelen bireyin kümülatif değerinden küçük ise o birey seçim havuzuna Tablo 3.7'deki gibi seçilmiştir. Bu örnekte görüldüğü gibi kümülatif değerleri büyük olan 1., 3. ve 5. bireylerin seçilme oranları daha yüksek olmaktadır.

Tablo 3.7 Yeni Birey Havuzu

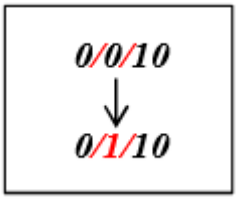
Kromozom	Onluk Sayı Değerleri	İkili Sayı Değerleri
1.Kromozom	2	0010
2.Kromozom	11	1011
3.Kromozom	7	0111
4.Kromozom	5	0101
5.Kromozom	5	0101

Oluşturulan birey havuzunda çaprazlama yapmak için ilk bireyden başlanarak rastgele 0 ile 1.00 arasında ondalık sayı üretilmiştir. Rastgele sayı en başta verilen 0.6 çaprazlama oranından küçük ise o birey rastgele seçilen bir birey ile yine rastgele seçilen bir noktadan itibaren Şekil 3.11'deki gibi yer çaprazlanmıştır. Bu süreç yeni birey havuzundaki tüm bireyler için tekrarlanmıştır.



Şekil 3.10 Çaprazlama İşlemi

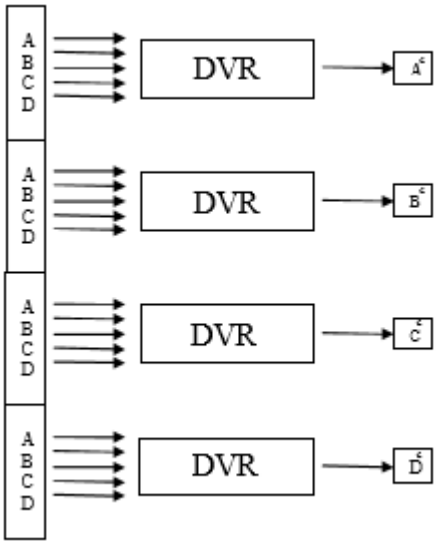
Mutasyon için ilk bireyden başlanarak rastgele 0 ile 1.00 arasında ondalık sayı üretilmiştir. Rastgele sayı belirtilmiş 0.01 mutasyon oranından küçük ise o bireyin herhangi bir bit değeri değiştirilmiştir (Şekil 3.12). Çaprazlama veya mutasyon x değişkenin başlangıç aralık sınırları dışında olması durumunda birey bazında tekrarlanmıştır.



Şekil 3.11 Mutasyon İşlemi

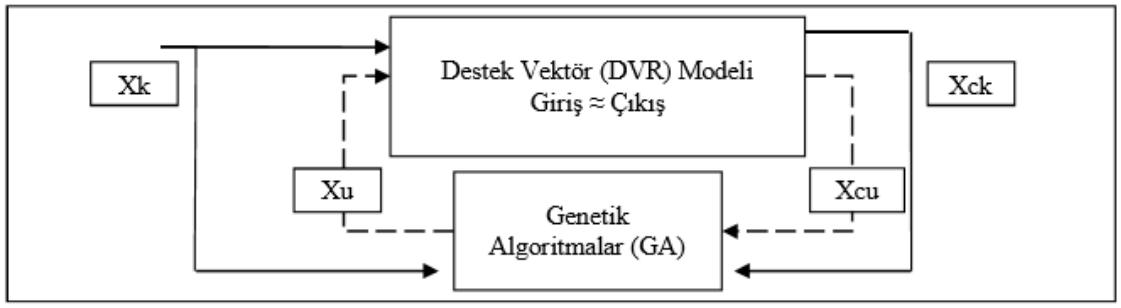
Genetik algoritmalar mutasyon işleminden sonra tekrar döngünün en başına dönmektedir. Daha sonra birey havuzunda bulunan yeni bireylerin uygunluk değerleri yeniden hesaplanarak sonlanana kadar devam ettirilmiştir. Sonlanma kriteri en başta belirlenen döngü sayısıdır. Zaten belirli bir döngü sayısı sonrası bireylerin en uygun çözüm olan =4 değerine yakınsadığı görülmüştür.

Anlatılanlara göre Destek vektör regresyonu (DVR) ve genetik algoritmalar (GA) ile eksik değer hesaplamasının yapılabilmesi için önce veri kümesinden eksik değer içermeyen tam kayıtlar seçilmektedir. Bu tam kayıtların Şekil 3.13'de görüldüğü gibi her seferinde bir tanesi çıkış diğer tüm nitelikler giriş olacak şekilde kullanılarak veri kümesindeki nitelik sayısı kadar destek vektör regresyonu yapısı oluşturulmaktadır. Böylece tüm regresyon yapıları eğitilerek genel regresyon modeli oluşturulmaktadır. Bu sayede veri kümesinin her bir tam kaydının çıkışta yaklaşık olarak benzer bir şekilde geri elde edilmesinin sağlandığı bir model oluşturulmuştur.



Şekil 3.12 Destek Vektör Regresyonu (DVR) Modeli

Şekil 3.14'de bahsedilmiş bu destek vektör regresyon modeli ve genetik algoritmalar optimizasyonu kullanılarak eksik değeri tamamlamak üzere oluşturulmuş yapı görülmektedir.

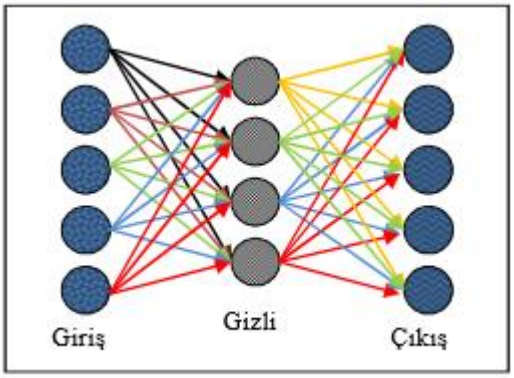


Şekil 3.13 Destek Vektör Regresyonu (DVR) ve Genetik Algoritmalar (GA) ile Eksik Değer Hesaplama

3.9 Yapay Sinir Ağları(YSA)

Yapay sinir ağları (YSA) insan sinir sistemi gibi davranır ve insan sinir sistemi gibi öğrenme işlemlerini gerçekleştirir. YSA giriş, gizli ve çıkış katmanlarından oluşmaktadır. Sinir hücreleri bir araya gelerek bir katmanı oluşturmakta ve her bir sinir hücresi sonraki katmandaki sinir hücrelerine ağırlık değerleri denilen parametre değerleri ile bağlanmaktadır. Ağırlık değerlerinin optimizasyonundan sonra, eğitilmiş bir yapay sinir ağı önceden öğrenmesi için verilen bilgi kategorisinde bir uzman olarak kabul edilmektedir.

Bu uzman olan sistem yeni bilinmeyen durumlar için tahminlerde bulunması amacıyla kullanılmaktadır. Yapay sinir ağları örüntü tanıma, sinyal işleme, zaman serileri tanıma, doğrusal olmayan kontrol, problem tanımlama ve benzeri alanlarda geniş bir şekilde kullanılmaktadır. Yapay sinir ağları, sınıflama değerlerinin bilindiği bir eğitim veri kümesiyle yapılan öğrenme yöntemidir. Bu yöntem değişik sınıflama görevlerinde fazlaca başarı hikayesi olan pratik bir yaklaşım olarak kendisini kanıtlamıştır. YSA, giriş katmanındaki sinir hücresi sayısı, gizli katman sayısı ve çıkış katmanındaki sinir hücresi sayısındaki değişiklikle bağlı olarak ağ yapısı itibariyle farklılıklar göstermektedir.



Şekil 3.14 Yapay Sinir Ağları Yapısı

Yapay sinir ağlarında öğrenme işlevi geri yayılım öğrenme algoritması ile yapılmaktadır. Bu öğrenme algoritması ağ çıkışının gerçekte olması gereken sınıf değeri ile bulunan değer arasındaki toplam farkı azaltmaya çalışmaktadır. Bu sayede katmanları birbirine bağlayan ağırlık parametre değerleri güncellenmektedir. En uygun yapay sinir ağ yapısının tam olarak oluşturulmasının kolay olmaması ve öğrenilen kuralların insanlar tarafından kolay yorumlanamaması ayrıca ağ yapısının eğitiminin uzun süre alabilmesi nedenleriyle yapay sinir ağları bazen eleştirilmektedir. Fakat gürültülü veriler üzerinde yüksek toleransa sahip, iyi bir şekilde çalışması ve birçok uygulamada başarılı sınıflama yapabilmesi yapay sinir ağlarının kullanılmasını avantajlı bir hale getirmektedir.

3.9.1 Geri Yayılım Algoritması

Geri yayılım algoritması ileri beslemeli çok katmanlı yapay sinir ağlarında öğrenmeyi gerçekleştirmektedir. İleri beslemeli yapay sinir ağı ağırlık değerleri vasıtasıyla giriş katmanını orta katman olarak da adlandırılan gizli katman veya gizli katmanlara oradan da en son çıkış katmanına aktaran ağ modelidir. Geri besleme ise adından da anlaşılabileceği üzere çıkış katmanındaki gerçek değer ile tahmin edilen değer arasındaki hata oranını temel alarak ağ yapısı üzerinde geriye doğru düzeltmeler yapılmasını sağlamaktadır. Geri yayılım algoritmasının çalışma adımları şu şekilde ifade edilmektedir.

Ağırlık ve bias değerlerine başlangıç değerlerinin verilmesi aşamasında yapay sinir ağında bulunan tüm ağırlık ve bias değerlerine rastgele -1.00 ile +1.00 veya -0.5 ile +0.5 arasında küçük değerli ondalık sayılar verilmektedir. Bundan sonra giriş değerleri vasıtasıyla gizli katmandaki nöron değerleri hesaplanmaktadır. Gizli nöron değeri hesaplaması ilgili nörona gelen tüm giriş nöron değerleri ile ağırlık değerlerinin

çarpımlarının toplamı şeklinde Bağıntı (3.10) ile hesaplanmaktadır. Burada Ij gizli nöron değerini, ise o nörona girişinden gelen ağırlık değerini temsil etmektedir. Denklemde görülen θ, bias olarak adlandırılarak gizli nöron değerinin sıfır olmasını engellemek ve Bağıntı (3.11)'deki aktivasyon fonksiyonun sonuç üretebilmesi amacıyla eşik değeri olarak kullanılmaktadır. Aktivasyon fonksiyonu büyük ölçekli verileri 0 ile 1 arasındaki bir değer aralığına dönüştürmekte ve öğrenme işlemini kolaylaştırmaktadır.

$$I_j = \sum(W_{ij} * 0_i) + \theta \quad (3.14)$$

$$0_j = \frac{1}{1+e^{-I_j}} \quad (3.15)$$

Bu şekilde çıkış katmanı nöron değerleri hesaplandıktan sonra geri yayılım yapılarak hata değerleri hesaplanmaktadır. Hata eğitim verisindeki gerçek çıkış değeri ile hesaplanmış değer arasındaki fark temel alınarak Bağıntı(3.12) ile hesaplanmaktadır.

$$Hata_j = 0_j(1- 0_j)(H_j - 0_j) \quad (3.16)$$

Buradaki 0_j hesaplanmış çıkış değeri, H_j ise gerçek, hedef çıkış değerini temsil etmektedir. Bu şekilde bütün çıkış nöronları ve çıkış nöronlarındaki hata değerleri hesaplanmaktadır.

$$Hata_j = 0_j (1-0_j)\sum_k (Hata_k * W_{jk}) \quad (3.17)$$

Bundan sonra gizli katmandaki her bir nöron için hata değerleri hesaplaması yapılmaktadır. Bunun için Bağıntı(3.13) kullanılır. W_{jk}, gizli katmanda bulunan nörondan çıkış katmana giden ağırlık değerini, $Hata_k$ ise o ağırlık değerinin ulaştığı çıkış katmanındaki nöron değerinin hatasıdır. En son aşamada ise hata oranlarına karşılık ağırlık ve bias değerleri güncellenmektedir. Yeni ağırlık değerleri Bağıntı(3.14) deki gibi yeni bias değerleri ise Bağıntı(3.15) 'daki gibi hesaplanmaktadır:

$$W_{iJ} = W_{iJ} + (\alpha * Hata_j * 0_j) \quad (3.18)$$

$$\theta_j = \theta_j + (\alpha * Hata_j) \quad (3.19)$$

α öğrenme katsayısı olarak bilinmektedir. 0 ile 1 arasında ondalık sayı değeri almakta ve bu katsayı karar verme uzayında yerel minimuma sıkışıp kalınmasını önlemektedir. Eğer bu katsayı düşük tutulursa öğrenme işlemi yavaş adımlarla ilerlemekte, yüksek tutulur ise

çözümün etrafında sarkaç gibi yakınlaşıp uzaklaşan dalgalanmalara neden olmaktadır. Geri yayılım algoritması eğitim veri kümesindeki her bir kayıttan sonra ağırlık ve bias değerlerinde güncelleme yapabildiği gibi algoritmanın çalışma hızı açısından tüm eğitim veri kümesi kayıtlarının en sonunda kümülatif toplam güncelleme de yapabilmektedir. Fakat daha iyi sınıflama başarısı için her kayıt sonrasında güncelleme tercih edilmektedir. Geri yayılım algoritması belli bir döngü sayısı veya ağırlık değerlerindeki değişim belli bir eşik seviyesini geçmeyinceye kadar devam etmekte ve sonlanmaktadır. Farklı tipteki aktivasyon fonksiyonu, öğrenme katsayısı veya hata fonksiyonu seçimiyle değişik şekillerde geri yayılım algoritması uygulanmaktadır.

3.9.2 Yapay Sinir Ağlarıyla Eksik Veri Tamamlama Örneği

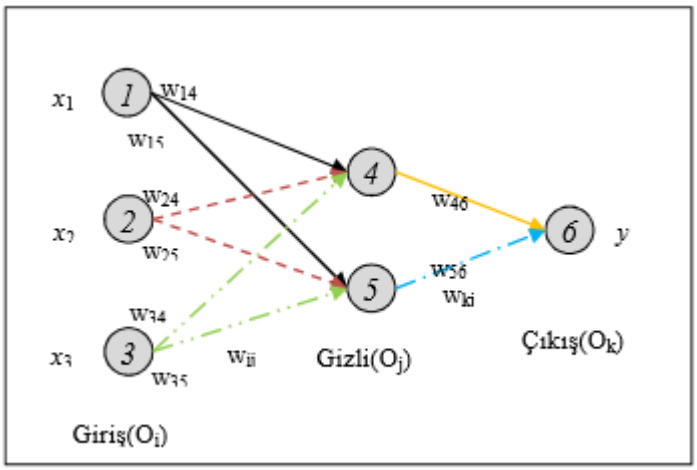
Tablo 3.3 'de verilen veri kümesinin yapay sinir ağları ile öğrenilmesi istenmiştir. Veri kümesinde X_1, X_2, X_3 olmak üzere 3 giriş ve sınıflama yapan y çıkış nitelik değeri bulunmaktadır.

Yapay sinir ağının yapısı veri kümesinin yapısına uygun olarak Şekil 3,7'deki gibi oluşturulmuştur. 1, 2, 3 giriş, 4, 5 gizli, 6 ise çıkış katmanında bulunan nöronları temsil etmektedir.

Tablo 3.7 Yapay Sinir Ağları Eğitim Örneği Veri Kümesi

X1	X2	X3	Y
1	0	1	1
1	0	0	0
0	1	0	1
1	1	0	0

1, 2, 3 giriş nöronlarına verilen veri kümesinin 1, 0, 1 eğitim örneğine karşılık 6 numaralı çıkış katmanında bulunan nöronda veri kümesinin y nitelik değeri olarak 1 elde edilmek istenmiştir.



Şekil 3.15 Yapay Sinir Ağlarının Bir Örneği

Başlangıç yapay sinir ağının değerleri Tablo 3.4'de gösterilmiştir. Başlangıç ağırlık değerleri ve bias değerleri küçük sayılarla rastgele oluşturulmuştur. Öğrenme katsayısı olarak 0.9 alınmıştır.

Tablo 3.8 Başlangıç, Giriş ve Bais Değerleri

X_1	1	W_{34}
X_2	0	W_{35}
X_3	1	W_{46}
W_{14}	0,3	W_{56}
W_{15}	-0,2	θ_4
W_{24}	0.5	θ_5
W_{25}	0.2	θ_6

Bu değerlere karşılık 4, 5 gizli katman nöronlarının ve aktivasyon fonksiyonu çıkış değerleri Çizelge 3.4'deki gibi hesaplanmıştır. Bu değerler tespit edildikten sonra çıkış katmanında bulunan 6 numaralı nöron, 4, 5 nöron değerleri yardımıyla elde edilmiştir.

Tablo 3.9 4,5,6 Numaralı Nöron Değerlerinin Hesaplanması

Nöron	Giriş	Aktivasyon Çıkış
4	0.3+0-0.4-0.3 = -0.4	$\frac{1}{(1+e^{0.4})} = 0.401$
5	-0.2+0+0.3+0.3 = 0.4	$\frac{1}{(1+e^{-0.4})} = 0.599$
6	(-0.2)(0.401)-(0.1)(0.599)+0.2 = 0.06	$\frac{1}{(1+e^{-0.06})} = 0.515$

Böylece yapay sinir ağının ileri besleme aşaması yapılmış olmaktadır. Bundan sonra geriye doğru hataları yayma aşaması olan geri yayılım algoritması uygulanmaktadır. Gerçekte Tablo 3.3'deki veri kümesine göre yapay sinir ağının 1, 2, 3 numaralı nöron girişlerine 1, 0, 1 değerleri verilince 6 numaralı çıkış katmanındaki nörondan 1 değeri elde edilmeliydi fakat bu değer Tablo 3.5'de görüldüğü gibi 0.515 olarak hesaplanmıştır. Buna göre hatalar Tablo 3.6'daki gibi hesaplanmıştır.

Tablo 3.10 6, 5, 4 Numaralı Nöron Hata Değerlerinin Hesaplanması

Nöron	Hata
6	(0.515)(1-0.515)(1-0.515) = 0.1211
5	(0.599)(1-0.599)(0.1211)(-0.1) = -0.0029
4	(0.401)(1-0.401)(0.1211)(-0.2) = -0.0058

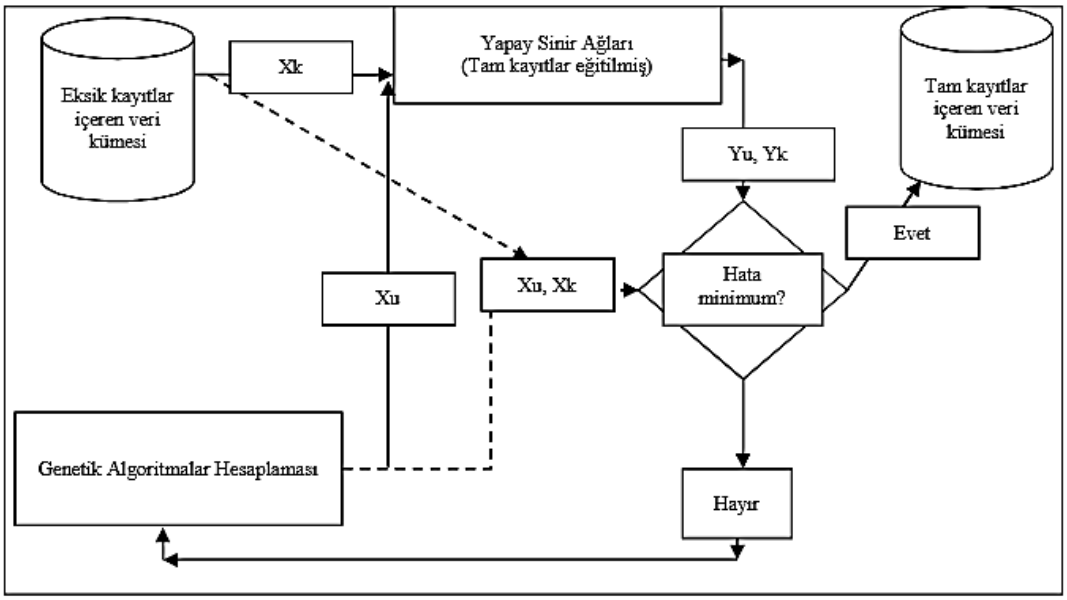
En son aşamada ise hata değerlerine karşılık ağırlık ve bias değerleri güncellenmiştir. Buna göre yeni ağırlık ve bias değerleri Tablo 3.11'de verilmiştir.

Tablo 3.11 Yeni Ağırlık ve Bias Değerleri

w_{14}	0.3+(0.9)(-0.0058)(1) = 0.2948
w_{15}	-0.2+(0.9)(-0.0029)(1) = -0.2026
w_{24}	0.5+(0.9)(-0.0058)(0) = 0.5
w_{25}	0.2+(0.9)(-0.0029)(0) = 0.2
w_{34}	-0.4+(0.9)(-0.0058)(1) = -0.4052
w_{35}	0.3+(0.9)(-0.0029)(1) = 0.2974
w_{46}	-0.2+(0.9)(0.1211)(0.401) = -0.1563
w_{56}	-0.3+(0.9)(0.1211)(0.599) = -0.2347
θ_4	-0.3+(0.9)(-0.0058) = -0.3052
θ_5	0.3+(0.9)(-0.0029) = 0.2974
θ_6	0.2+(0.9)(0.1211) = 0.3090

Tablo 3.11'deki güncellemeler yapıldıktan sonra veri kümesinde bulunan bir sonraki eğitim veri kaydı yapay sinir ağı girişine verilmiş ve yeniden ağırlık ve bias değerleri güncellenmiştir. Bu şekilde yapay sinir ağı eğitim sonlanma kriterine kadar devam edilerek öğrenme işlemi gerçekleştirilmiştir.

Abdella ve Marwala, Şekil 3.8'deki modeli kullanarak eksik verileri hesaplamıştır. Yapay sinir ağ yapısı veri kümesinin özel olarak modellemesinde yani giriş katman değeri çıkış katman değeriyle neredeyse aynı olacak şekilde oluşturulmuştur. Giriş ile çıkış arasındaki fark genetik algoritmaların uygunluk fonksiyonu olarak eksik veri hesaplamasında kullanılmaktadır.



Şekil 3.16 Yapay Sinir Ağları (YSA) ile Eksik Değer Hesaplama

Yapay sinir ağ yapısı giriş(X), çıkış(Y) ve ağırlık(w) değerlerinden oluşmaktadır. Matematiksel olarak yapay sinir ağı ifade edilirse aşağıdaki şeklinde yazılmaktadır.

$$Y = f(X,W) \tag{3.20}$$

Eğer ağ yapısı girişte verilen vektör değerlerini çıkışta tahmin edecek üzere eğitilirse giriş(X) ile çıkış(Y) çıkış neredeyse birbirine eşit olmaktadır .

$$X \sim Y \tag{3.21}$$

Gerçekte giriş ile çıkış vektörü her zaman mükemmel bir şekilde birbirine eşit olamazlar. Bu yüzden giriş ve çıkış vektörünün farkını ifade eden bir hata (e) fonksiyonu Denklem 4.32'deki gibi gösterilmiştir.

$$e = X - Y \tag{3.22}$$

Burada Y yerine Bağıntı (3.16) eşitliği yazılırsa hata fonksiyonu Bağıntı (3.19) halini almaktadır.

$$e = (X - \mathrm{f(X,W)}) \tag{3.23}$$

Hatanın sıfır olmayan minimum değeri tercih edildiğinden dolayı, hata fonksiyonu karesi alınarak ifade edilmektedir.

$$e = (X - \mathrm{f(X,W)})^2 \tag{3.24}$$

Eksik değer, giriş vektöründe (X) bazı değerlerin olmadığı durumlarda oluşmaktadır. Giriş (X) vektöründe bilinen değerler için (X_K), bilinmeyen değerler ise (X_U) olarak ayrı ayrı yazılabilmektedir.

$$e = (\binom{X_K}{X_U} - f\binom{X_K}{X_U}, W))^2 \tag{3.25}$$

Böylece genetik algoritmaların (GA) uygunluk fonksiyonu Bağıntı(3.21)'de elde edilmiştir. GA uygunluk fonksiyonunu minimize edecek değerler eksik değerler olarak kabul edilmekte ve böylece eksik değerlerin hesaplaması yapılmış olmaktadır.

Buna göre destek vektör regresyon modelinin girişine (Bağıntı 3.22) tam olmayan bir kayıt verilmektedir. Bu kayıtta X_k bilinen nitelikler, Xu ise eksik nitelik değerleri olarak kabul edilmektedir. Destek vektör çıkışı f fonksiyonu (Bağıntı 3.23) olarak adlandırılmakta ve değerleri başta bilinen Xck ve sonradan hesaplanmış Xcu vektörlerinden oluşmaktadır. Eğitilmiş DVR modelinin girişinin çıkışına yaklaşık olarak eşit olması istenmekte ve aradaki fark hata (Bağıntı 3.24) olarak adlandırılmaktadır. Genetik algoritmaların amacı giriş ile çıkış arasındaki bu hatayı negatif olmayan uygunluk fonksiyonu (Bağıntı 3.25) yardımı ile minimum hata yapan eksik değerleri hesaplamaktır:

$$DVR\ giriş = \binom{X_K}{X_U} \tag{3.26}$$

$$DVR\ çıkış = \binom{X_{ck}}{X_{cu}} = f\binom{X_K}{X_U} \tag{3.27}$$

$$Hata = \binom{X_K}{X_U} - f\binom{X_K}{X_U} \tag{3.28}$$

$$GA\ uygunluk\ fonksiyonu = (\binom{X_K}{X_U} - f\binom{X_K}{X_U})^2 \tag{3.29}$$

3.10 Bulanık c-ortalamalar (BCO) ile Eksik Değer Hesaplama

Kümeleme yapmanın genel amacı veri kümesinde var olan verilmiş bir takım nesneleri, nesnelerin benzerliğine göre alt gruplara bölme ve bu alt kümeler arasındaki benzerliği en aza indirmektir. Bulanık c-ortalamalar (BCO) yöntemi var olan bir nesneyi bir veya birden çok kümeye üye yapmaktadır. Bu yöntem 1973 yılında Dunn tarafından ortaya atılmış 1981 yılında Bezdek (Wang, 1983) tarafından iyileştirilmiş ve sık sık örüntü tanıma alanında kullanılmaya devam edilmektedir. Yöntemin başlıca hedefi amaç fonksiyonunu (Bağıntı 3.26) minimum hale getirmeye çalışmaktır.

$$J_M = \sum_{\text{İ}=1}^{N} \sum_{J=1}^{C} Uij^2 \, (X_i - C_j)^2 \qquad (3.30)$$

$$2 \leq C \leq N \qquad (3.31)$$

$$1 \leq m \leq \infty \qquad (3.32)$$

$$U_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{|X_i - C_j|}{X_i - C_k|})^{\frac{2}{m-1}}} \qquad (3.33)$$

$$C_j = \frac{\sum_{i=1}^{K} {U_{ij}}^m * X_i}{\sum_{i=1}^{N} {U_{ij}}^m} \qquad (3.34)$$

Bağıntı(3.27)'deki c parametresi küme merkez sayısını göstermektedir 83.28ve 2 ile veri kümesi kayıt adedi (N) arasında bir tam sayı değeri almaktadır. Bağıntı(3.28)'de ise m parametresi ağırlık faktörü olarak adlandırılmakta ve 1 ile ∞ arasında ondalık sayı değeri almaktadır. Ağırlık faktörü parametresi kümeleme sürecinde bulanıklık miktarını kontrol etmektedir. Teorik olarak c ve m parametresinin optimum seçim değeri olmamaktadır. Bu parametreler veri kümesinin karakteristik özelliğine ve veri kümesinde bulunan niteliklerin birbirleriyle olan ilişki düzeylerine bağlı olarak değişmektedir. Bu tez çalışmasında önerilen bulanık c-ortalamalar ile eksik veri hesaplaması yapılırken o anda kullanılan veri kümesi yapısına en uygun, optimum c ve m parametre değerlerinin tespit edilmesi sağlanmıştır. Bulanık kümelemede her veri nesnesi bir üyelik fonksiyonu (Bağıntı 3.29) değerine sahiptir. Bu değer nesnenin hangi küme merkezine (Bağıntı 3.30) ne derece ait olduğunu tanımlamaktadır.

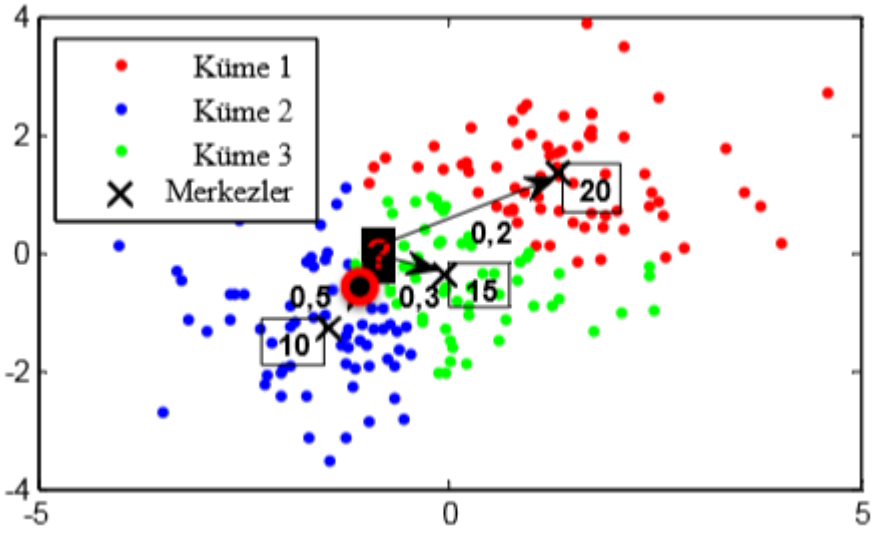
Üyelik fonksiyon değerlerini ve küme merkezlerini güncelleme sürecinde sadece veri kümesinde eksik değer içermeyen tam kayıtlar hesaba alınmaktadır. Bu süreçte temel klasik k-ortalamalar kümeleme yöntemine nazaran her bir veri nesnesi tek bir kümeye ait

olmamaktadır. Veri nesnesi bütün küme merkezlerine üyelik fonksiyon değerleri nispetince üye olmaktadır. Tam olmayan veri kaydının eksik değeri bu üyelik fonksiyon değerleri ve küme merkezi olarak kabul edilen değerler yardımıyla hesaplanmaktadır. Deneysel çalışmalar bulanık eksik değer tamamlama algoritmasının temel klasik kümeleme algoritmasına göre daha iyi performans sonuçları ortaya koyduğunu göstermiştir.

3.10.1 Bulanık c-ortalamalar Örnek Uygulaması

Şekil 3.9'da eksik bir değerin bulanık c-ortalamalar ile hesaplanması bir örnek üzerinden açıklanmıştır. Örneğin soru işareti (?), veri kümesindeki bir eksik değer olarak kabul edilmektedir.

Eksik değer içermeyen veri kümesinin tam elemanları 3 küme merkezine ayrılmıştır. Bulanık c-ortalamalar m, ağırlık faktörü parametre değeri 2 olarak tercih edilmiştir. Bu durumda eksik değerin (?), küme merkezlerine olan üyelik fonksiyonu değerleri sırasıyla $U_{?-1}$ =0,2, $U_{?-2}$=0.5 ve $U_{?-3}$=0.3 olarak hesaplanmıştır.



Şekil 3.17 Bulanık c-ortalamalar (Bco) ile Eksik Değer Hesaplaması

Küme merkez değerleri Bağıntı 3.30 ile sırasıyla c_1= 20, C_2 =10 ve C_3=15 olarak hesaplanmıştır. Böylece eksik olan soru işareti (?) değeri

c_1*$U_{?-1}$+ c_2*$U_{?-2}$+c_3*$U_{?-3}$'den yani 0.2*20+0.5*10+0.3*15 işleminin hesabından sonra 13.5 olarak bulanık c-ortalamalar eksik değer hesaplama yöntemi ile bulunmuştur.

3.11 Beklenti Maksimizasyonu Algoritması(EM Algoritması)

Beklenti maksimizasyonu algoritması, literatürde Expectation Maksimation (EM) diye geçmektedir. Bu algoritmanın amacı, örnek küme üzerinde bazı tespitler yapılır ancak bunlardan bir kısmı eksik veya hatalı parametrelerle temsil edildiği durumlarda kullanılabilir. Özellikle istatistiksel uygulamalar ile istatistiksel örüntü tanıma sistemlerinde oldukça kullanılabilen bir yöntemdir.

Tanım yaparken üzerini vurguladığım gibi bu algoritma istatistiksel yöntemlere dayanmaktadır. Bu algoritmayı uygulamak için ilkönce hatalı veya eksik olan verilen tespit edilir. Daha sonra ise tespit edilen parametrelerin örnek küme üzerindeki büyüklükleri yani tespit edilen parametreler kaç elemana ait olduğu çıkartılır. Çıkartılan bu sayıların kullanım amacı örnek kümeye ait olasılık yoğunluk fonksiyonun içinde kullanılacağındandır. Bu tür dağılımlar tekrar eden dağılımlardır. EM (Expectation Maximization) Algoritması bir objenin hangi kümeye ait olduğunu belirlemede kesin mesafe ölçütlerini kullanmak yerine tahminsel ölçütleri kullanmayı tercih eder. Maksimum benzerlik prensibine dayanan Beklenti Maksimizasyonu (BM) algoritması ilk olarak Dempster, Laird ve Rubin tarafından 1977 yılında ortaya konulmuştur. Regresyon atamasının iteratif süreçli bir halidir ve 2 iteratif adımdan oluşur.

EM algoritması son yıllarda bir çok araştırmada kullanılan popüler bir yaklaşım olmuştur. EM algoritması, tam olmayan veri problemlerini çözmek için maksimum olasılık tahminlerini yapan tekrarlı bir algoritmadır. EM Algoritmasının her tekrarı iki adımda gerçekleşir. Bu adımlar, bekleneni bulma (E-Adımı) ve maksimizasyon (M Adımı) olarak adlandırılır [2]. E-adımında gözlenen verilerin parametrelerine ait kestirimler kullanılarak bilinmeyen (kayıp) veri ile ilgili en iyi olasılıklar tahmin edilirken, M-Adımında ise tahmin edilen kayıp veri yerine konulup bütün veri üzerinden maksimum olabilirlik hesaplanarak parametrelerin yeni kestirimleri elde edilir.

Benzer bir uygulama olarak:

$$F(x,p) = \frac{100!}{10!x65!x15!x10!}(0.2)^{x_1+x_3}(0.6-0.5p)^{x_2}(0.25p)^{x_4+x_5} \qquad (3.35)$$

Daha sonra tüm veriler işlemlerde kullanmak için yeni değişkenlere atanır. Hatalı verilerin değişkenleri en az iki tane olmalıdır, çünkü eğer tek değişkenden ibaret olursa zaten ona eşit olurdu. Hatalı veriler bu sebepten en az iki değişkene atanmalıdır.

Çİ:X_1 N:X_2 H:X_3+X_4 ÇH:X_5 (3.36)

Yukarıda görülen örnekte görüldüğü gibi hatalı veriye sahip 'H' verisi iki tane değişken biçiminde yer almıştır. Daha sonra örnek küme üzerindeki veriler incelenir ve bu verilere ait olasılık dağılımları belirlenir. Bu dağılımlar hatalı veya eksik veri olasılığı olan "p" ile yazılırlar. Yani burada p' yi kullanarak beklentinin makzime edilmesinin yolu açılmış olur. Örnek olarak dağılım:

P(Çİ)=0.2 P(N)=0.6-0.5P P(H)=0.2+0.25P P(ÇH)=0.25P (3.37)

Yukarıdaki gibi olasılık dağılımları çıkartılır ve daha sonra olasılık yoğunluk fonksiyonunun kurulmasına geçilir. Bu kurulum yukarıda ki f(x,p) fonksiyonudur. Bu fonksiyonda ilk kısım yani, faktöriyel işleminin hesaplanmasına gerek yoktur. Çünkü bu işlem ilerde görüleceği gibi türev alma işleminden sonra "0" a eşit olacaktır. Bunun yerine işlem kolaylığı açısından "A" demek yeterli olacaktır.

Olasılık dağılımında dikkat edilirse bazı parametreler aynı üst altında yazılmıştır. Bunun sebebi değişken atamadır. Değişken atamalardan sonra aynı köke sahip kökler çıkacaktır. Yani:

P(X_1)=0.2 P(x_2)=0.6-0.5p P(X_3)=0.2 P(X_4)=0.25 P(X_5)=0.25p

Burada görüleceği gibi x1 ile x2 , x4 ile x5 aynı köklere sahiptir. Bu da olasılık yoğunluk fonksiyonu üzerinde görülebilir.

Olasılık yoğunluk fonksiyonunun bu şekliyle işlemlerde kullanılması oldukça zordur. İşlemleri kolaylaştırma için bu fonksiyonun doğal logaritması alınır.

$$\ln\big(f(x,p)\big) = \ln A + (X_1 + X_3)\ln(\quad) + X_2\ln(0.6-0.5p) + (X_4 + X_5)\ln(0.25p)$$

Daha sonra literatürde E-Step denilen beklenti adımına geçilir. Bu adımda olasılık yoğunluk fonksiyonun beklentisi alınır. Fonksiyon üzerinde hatalı veriye ait değişkenler hariç diğerlerinin hepsi bilinir. Yani beklenti adımından sonra:

$$E(\ln(f(x,p)) = \ln A + (X_1 + X_3)\ln() + x_2 \ln(0.6 + 0.5p) + x_5 \ln(0.25p)$$

Fonksiyon bu hale gelecektir. Burada x3 ve x4 haricindeki bütün x'ler bilinmektedir. Bu x'ler problemin tanımında bilinene değişkenlerdir. Yine burada k'lı yazımın sebebi k. iterasyonu vurgulamaktır.

Daha sonra bu fonksiyonun türevi alınır ve sonuç sıfıra eşitlenir. İşlem içindeki p'ler yalnız bırakılır. Bu adıma literatürde M-Step yani maksimizasyon adımı denir. Amaç yakınsayan "p" değerleri için bilinmeyen parametrelerin bulunmasıdır.

3.12 Çoklu Atama Metodu

Çoklu atama, Tekli atma yöntemlerinin birleşimini oluşturur. Çoklu atamada Monte Carlo Tekniği kullanılır.

m =Ataması yapılmış ve analiz edilmiş küme sayısı

Qi= analiz edilmiş i. kümeden tahmin

Vi = analiz edilmiş i. kümeden varyans tahmini

Çoklu atamalardan elde edilen nokta tahmini:

$$\emptyset = \frac{1}{m}\sum_{i=1}^{m} \theta_i \qquad (3.38)$$

Nokta tahmini için varyans tahmini:

$$V = \frac{1}{m}\sum_{i=1}^{m} \gamma_i + \frac{m+1}{m}[\frac{1}{m-1}\sum_{i=1}^{m}(\theta_i - \theta)^2 \qquad (3.39)$$

3.12.1 Monte Carlo Yöntemi

Rastgele sayı seçme ilkesi üzerine kurulmuş algoritmalara rastlantısal algoritmalar denir. Her ne kadar yanıt vermedikleri, hatta kimileyin yanlış yanıt verdikleri bile olsa, bazı problemler için, rastlantısal algoritmalar kesin algoritmalara tercih edilebilirler, çünkü daha basittirler ve daha hızlı çalışırlar.

N sayısı ilgili bir algoritmanın 2^n zamanda sonuçlanması bilgisayar biliminde pek arzu edilmez, çünkü böyle bir algoritma çok yavaş işler. Ama daha kısa bir algoritma bilinmiyorsa ne yapmalı? Rastlantısal algoritmalar bu hız sorununa kısmi bir çözüm

getirirler: yüzde yüz doğruluktan vazgeçerek hızlı algoritmalar bulabiliriz. Elbette rastlantısal algoritmaların uçaklarda kullanılmamasının büyük yararları vardır. Rastlantısal algoritmalar, Monte Carlo, Las Vegas ve Sherwood olarak üç ana başlıkta toplanabilir.

Monte Carlo algoritmaları her zaman bir sonuç verir ve sonucun doğruluk olasılığı program çalıştıkça artar. Örneğin her denemede yüzde 90 başarı şansı varsa, iki denemede şansı yüzde 99'a çıkar.

Öte yandan Las Vegas algoritmaları yanlış sonuç üretmezler, ama bazen hiç sonuç üretmezler. Sherwood algoritmaları ise her zaman sonuç verir ve verdikleri sonuç doğrudur. Ancak Sherwood algoritmalarının uygulanabildiği problemler sınırlıdır.

Şimdi çok bilinen bazı problemler üzerinde Monte Carlo algoritmasının çalışmasını inceleyelim:

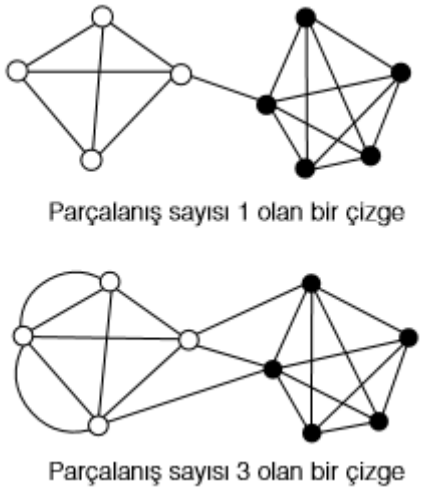
Baskın Eleman Problemi: Bir sayı dizisinde, sayıların yarıdan fazlası aynıysa, o elemana baskın eleman denir. Bir dizideki baskın elemanı bulma algoritması, en basit olarak tüm elemanların diğerleriyle karşılaştırması olabilir, bu da O(n log n) zaman alır.

Bu problem için oluşturulabilecek bir Monte Carlo algoritması şöyle olabilir: Programımız, verilen n sayı arasından rastgele bir sayı seçsin. Eğer bu sayı verilen n sayının en az yarısına eşitse program dursun ve yanıt olarak bu sayıyı versin. Eğer bu sayı verilen n sayının en az yarısına eşit değilse program gene dursun ve yanıt olarak "baskın eleman bulunamadı" desin. Eğer problem "bulundu" yanıtını verirse, baskın eleman bulunmuş demektir. "Bulunamadı" yanıtını verirse ya yanlış eleman seçilmiştir ya da baskın eleman yoktur. Eğer baskın eleman varsa, program en az yüzde elli olasılıkla baskın elemanı seçecektir ve doğru yanıtı verecektir. Program beş kez çalıştırıldığında, programın baskın elemanı bulma olasılığı 1 – (1/2)5 = 0,96875'tir, yani nerdeyse yüzde 97, fena sayılmaz. Ve program 5n zamanda biter, oldukça çabuk. Eğer program› alt› kez çalıştırırsak, doğru yanıtı bulma olasılığı 1 – (1/2)5 = 0,984375'dir ve program 6n zamanda biter.

Asallık Testi: Monte Carlo asallık testi algoritması 2 ile √n arasında rastgele sayılar üretir ve bu sayıların n'yi bölüp bölmediğine bakar. Eğer bölen bir say› bulunursa, n asal

değildir. Ama tam bölen bir sayı bulunamazsa, kesin olarak asaldır diyemeyiz. Bu algoritmanın çok iyi bir algoritma olduğu söylenemez. Örneğin, 60329 sayısı 23, 43 ve 61'in çarpımıdır. Algoritma 2 ile 60329'un köküne en yakın tamsayı olan 245 arasında rastgele sayılar seçecektir. Ancak bu aralıkta sadece üç say› doğru sonuç üretilmesini sağlar. Bu nedenle, bu örnekteki Monte Carlo asallık testi algoritması ancak %1,224 olasılıkla doğru sonucu verir, hiç de iyi bir sonuç sayılmaz.

Çizgelerin Parçalanış Sayısı: Bir çizgede, çıkarıldığı taktirde çizgeyi iki parçaya bölecek olan kenarlardan oluşan bir kümeye o çizgenin parçalanış kümesi diyelim. Çizgenin parçalanış sayısı, bu kümelerin sahip olabileceği en düşük öğe sayısıdır. Parçalanış sayısı, bir anlamda, çizgenin ne kadar "sağlam", yani ne derece tekparça olduğu konusunda bir fikir verir: Parçalanış sayısı ne kadar büyükse, çizge o kadar tekparçadır, noktalar o derece birbirine bağlanmıştır diyebiliriz.



Şekil 3.18 Parçalanış Sayısına Göre Çizgeler

3.13 Yerine Ortalama Koyma Yöntemi

Bu yöntem, veri setinde kayıp verinin olduğu alandaki diğer verilerin ortalamasını alarak kayıp olan verileri doldurmaya yarayan yöntemdir. Veri aralığı düşük olan verilerde kullanıldığında yararlı olabilir. Aksi halde hata oranını arttırır.

4.UYGULAMA

Bu çalışmada ele alınan yöntemler kayıp veri bulunduran İris datası üzerinde uygulanıp, yöntemler karşılaştırıldı. Algoritmalar ve yöntemler RapidMiner, SPSS ve Weka programları yardımıyla uygulandı.

Tablo 4.1 İris Datasının Orijinal Hali

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	?	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	?
8	id_8	Iris-setosa	5	?	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	?	1.500	?
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	?	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	?

Tablo 4.1'de iris datasının belli bir kısmı gösterilmiştir. Kayıp veriler görüldüğü üzere '?' ile gösterilmiştir.

4.1 Regresyon Analizi (En Küçük Kareler Metodu)

İris datasındaki kayıp veriler ilk olarak Regresyon yöntemi ile atandı. RapidMiner programında gerekli Tool'lar kullanılarak gerçekleştirildi.

Tablo 4.2 Regresyon Analizi ile Atanmış Kayıp Veriler

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3.046	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	1.206
8	id_8	Iris-setosa	5	3.046	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.046	1.500	1.206
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	5.851	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	1.206

Yine Tablo 4.1'de gösterilen iris datasının orijinal halinde olan kayıp veriler (?) regresyon analizi ile atanmıştır.

4.2 Hot Deck Yöntemi (En Yakın Komşu Algoritması)

Kayıp veriler için yöntem olarak KNN algoritması uygulanmıştır.

Tablo 4.3 KNN Algoritması ile Kayıp Verilerin Atanması

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3.600	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	0.200
8	id_8	Iris-setosa	5	3.400	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.100	1.500	0.100
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	4.400	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	0.200

4.3 Naive Bayes Algoritması

Tablo 4.4 Naive Bayes ile Kayıp Verilerin Atanması

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3.312	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	0.316
8	id_8	Iris-setosa	5	3.318	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.734	1.500	0.380
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	4.532	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	0.052

Yine Tablo 4.1'de gösterilen iris datasının orijinal halinde olan kayıp veriler (?)knn algoritması ile atanmıştır.

4.4 Yerine Ortalama Koyma Yöntemi

Tablo 4.5 Yerine Ortalama Koyma Metodu ile Eksik Veri Atama

Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3.046	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	1.206
8	id_8	Iris-setosa	5	3.046	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.046	1.500	1.206
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	5.851	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	1.206
16	id_16	Iris-setosa	5.700	4.400	1.500	0.400

Yine Tablo 4.1'de gösterilen iris datasının orijinal halinde olan kayıp veriler (?) yerine ortalama koyma metodu ile atanmıştır.

Bu tez çalışmasında Rapid Miner programı kullanılarak elde edilen sonuçlarda;

Yerine ortalamayı koyma yöntemi, korelasyonun düşmesine ve verilerin dağılımlarını olumsuz olarak değiştirilmesine yol açar. Bu yöntem az kayıp verisi az ve aralığı düşük setlerde kullanılabilir.

Regresyon ataması, bu yöntem için kayıtlarda önce korelasyonu yüksek iki alan seçilip ona göre bir regresyon formülü üretilebilir. İlişkili olamayan alanlarda işe yaramaz. Bu fonksiyonun oluşturulmasında hatanın göz önünde bulundurulması gerekir.

Hot-Deck atama, veriler arasındaki mesafeye bakarak sabit bir sayıyı boş alanlara eklediği için hata bayı oldukça yüksek çıkan bir algoritmadır. Avantajı, uygulamasının kolay olması ve az veri kaybında hatayı fazla etkilememesidir.

Naive Bayes yöntemi, olasılıksal yöntemleri kullandığı için tüm veriyi kullanır. Bayes bir kümeleme algoritması olduğu için mümkün olduğu kadar fazla veri bulabileceği değerler arasındaki anlamlılığı arttırır. Az sayıdaki verilerde hata oranı yüksektir.

5. SONUÇ

Veri madenciliğinde sık ortaya çıkan kayıp veriler, araştırmacılar için dikkate alınmadığında araştırmacıları yanlış sonuçlara götürebilir. Kayıp verilerin tahmininde gözlem sayısı ve verinin özelliği oldukça önemlidir.

Bu çalışmada kayıp veriler ile ilgili sonuçlar aşağıda verilmiştir. Buna göre herhangi bir yöntemin diğerlerinden tamamen üstün olduğu söylenemez. Ancak veri özelliğine, kayıtların birbirleri ile olan ilişkisine, kayıt sayısına ve kayıp veri sayısının toplam kayıt oranına bakarak bir yöntem seçimi yapılabilir.

Her şeyden önce kayıp verileri Durum Düzeyinde Silme işlemi yapılırsa kullanımı basit olmasına rağmen fazla veri kaybında varyans artar ve Rassal Olarak Kayıpta (ROK) hatalı sonuçlar üretir.

Beklenti maksimizasyonu, Maksimum benzerlik prensibine dayandığı için tüm verilerin kullanılması gerekir. Buradaki benzerlik olabilmesi için ise kayıp verili kayıta benzer yani değer aralığı az olan ve veri seti büyük olan kayıtlarda uygulanması daha doğru sonuçlar verir.

Son gözlemi ileri Taşıma, birbirine yakın değerleri olan alanlarda kullanılabilir, ancak veri aralığı yüksek veri setlerinde hatayı yükseltir.

Karar ağacı yönteminde kayıp verilerin fazla olması ağaçtaki tutarsızlığı artırmaktadır.

Karar ağacında en önemli nokta; onu oluşturan eğitim kümesi ve sağlama kümesi arasındaki ilişkidir. Ağaç karmaşıklaştıkça eğitim kümesi için doğruluğu artmakta, ancak sağlama kümesi için ise doğruluğu azalmaktadır.

Sonuç olarak, elimizde bulunan verinin yapısına ve içeriğine göre algoritmaların farklı problemlerde farklı başarı oranları göstermesi doğaldır. Dolayısıyla en iyi algoritma budur diye genel bir şey yoktur problemin tipine göre kayıp veriyi tespit etme yöntemi değişebilmektedir. Yapay sinir ağları, genetik algoritmalar gibi yapay zekâ algoritmaları da kayıp verinin tahmininde kullanılabilecek olup, araştırmalar yapılabilinir.

daha kaliteli, tutarlı eğitim verisine sahip veri kümelerinin eksik değerlerinin en yakın k-komşu algoritmasıyla daha hassas ve doğru başarıyla hesaplanabildiği ortaya çıkmıştır. Fakat genetik algoritmaların çözümü bulamadığı ya da gerçekte bulunması gereken eksik değer veri kümesinde hesaplanmasının zor olduğu aykırı, gürültülü veri olması durumunda önerilen yöntem en yakın benzer k adet komşunun ağırlıklı ortalaması almıştır. Alınan ağırlıklı ortalama eksik değer tahminini daha yumuşak, kabul edilebilir

bir değerde olmasına neden olmuştur. Ayrıca önerilen EykYsa yaklaşımın veri kümesinin tek bir kaydında birden çok eksik değer olma durumunda bile başarılı sonuçlar ürettiği görülmüştür. Veri kümesi kayıtlarında birden çok eksik değer olması durumunda bile araştırmacıların en yakın k-komşu yaklaşımını tercih etmeleri yapılan çalışmalara daha fazla avantaj sağlayacağı görülmüştür.

KAYNAKLAR

Molenberghs, G., Thijs, H., Jansen, I., Beunckens, C., Kenward, M.G., Mallinckrodt, C., Carroll, R.J.:Analyzing Incomplete Longitudinal Clinical Trial Data 2004.

Aydın S., Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Doktora Tezi, 2007.

Alpaydın, E., 2000. Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2knotlar.doc, Erişim Tarihi: 12.06.2013

Veri tabanı yönetimi veri ambarı ders notu Yrd. Doç. Dr. Altan MESUT Trakya Üniversitesi Bilgisayar Mühendisliği

Veri madenciliği yöntemleri Bilgisayar bilimleri ve mühendisliği 2.basım Dr.Yalçın Özkan

]H. Coflkun Gündüz, www.cs.bilgi.edu.tr/~cgunduz.

Afifi, A.A., R.M., Elashoff, (1966), "Missing Observations in Multivariate Statistics I: Review of the Literature", Journal of American Statist. Assoc., 61 , 595-604.

Dalsgaard, T., C., Andre ve P., Richardson (2001), "Standard Shocks In The Oecd Interlink Model" Economıcs Department Workıng Papers No. 306, 32

[1] Peng Liu, Lei Lei, **Missing Data Treatment Methods and NBI Model**, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06) 2006 IEEE.

[2] Baygül, Arzu , **Kayıp Veri Analizinde Sıklıkla Kullanılan Etkin Yöntemlerin Değerlendirilmesi**, İstanbul Üniversitesi Sağlık Bilimleri Enstitüsü Yüksek Lisans Tezi 2007.

[3] Molenberghs, G., Thijs, H., Jansen, I., Beunckens, C., Kenward, M.G., Mallinckrodt, C., Carroll, R.J.:**Analyzing Incomplete Longitudinal Clinical Trial Data** 2004.

ÖZGEÇMİŞ

Ad Soyad: TUĞÇE KİRAZ

Doğum Tarihi: 02.05.1994

Doğum Yeri: İSTANBUL

Lise: Cahit Elginkan Anadolu Lisesi

Stajlar:

Ziraat Teknoloji / 15.06.2015-18.07.2015 / Stajyer

Arçelik-LG Klima / 22.07.2015-04.09.2015 / Stajyer

İş Tecrübesi:

Milliyet Gazetesi/ Milliyetemlak.com / Ürün Geliştirme/ Uzman Yardımcısı 19.07.2016-20.01.2017