T.C.
DİCLE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ ALGORİTMALARI KARŞILAŞTIRILMASI

Cengiz COŞKUN

YÜKSEK LİSANS TEZİ

MATEMATİK ANABİLİM DALI

DİYARBAKIR
HAZİRAN 2010

T.C.
DİCLE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ ALGORİTMALARI KARŞILAŞTIRILMASI

Cengiz COŞKUN

YÜKSEK LİSANS TEZİ

DANIŞMAN : Yrd. Doç. Dr. Abdullah Baykal

MATEMATİK ANABİLİM DALI

DİYARBAKIR
HAZİRAN 2010

T.C

DİCLE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ

DİYARBAKIR

……..... tarafından yapılan bu çalışma, jürimiz tarafından ……. Anabilim Dalında DOKTORA/YÜKSEK LİSANS tezi olarak kabul edilmiştir.

Jüri Üyesinin

Ünvanı Adı Soyadı

Başkan :

Üye :

Üye :

Yukarıdaki bilgilerin doğruluğunu onaylarım.

…../…../……

……..

ENSTİTÜ MÜDÜRÜ

(MÜHÜR)

ÖZ

Bu tezde veri madenciliğinin genel bir tanımı yapılmış; veri madenciliği metotları ve algoritmaları hakkında bilgi verilmiş, model oluşturma basamakları ve oluşturulan modellerin karşılaştırılması için kullanılan metriklerden bahsedilmiş ve bu teorik bilgiler ışığında bir veri kaynağı üzerinde veri madenciliğinde yaygın olarak kullanılan birkaç algoritmanın karşılaştırmalı değerlendirmesi yapılmıştır.

Çalışmanın teorik kısmında veri madenciliği uygulamasının bir veri kaynağı üzerinde hangi aşamalardan geçtiği üzerinde durulmuş, kullanılmakta olan metotlar ve bu metotlara ait algoritmalar anlatılmış ve oluşturulan modeli değerlendirme kriterleri hakkında bilgi verilmiştir.

Çalışmanın uygulama kısmında ise, teorik kısımda anlatılmış olan bilgiler ışığında; J48, NaiveBayes, Lojistik Regresyon ve KStar algoritmalarının karşılaştırması yapılmıştır. Karşılaştırma yapılırken veri önişlemeden başlamak üzere, hangi bilgilerin kullanıldığı, nasıl bir metot izlendiği, algoritmaların oluşturduğu modellerin istatistiksel sonuçları ve bu sonuçların nasıl değerlendirildiği detaylı bir şekilde anlatılmıştır.

Yapılan karşılaştırma sonucuna göre eldeki veri üzerinde çalıştırılan dört farklı algoritmadan J48 algoritmasının ürettiği modelin en iyi karşılaştırma ölçütlerine sahip olduğu sonucuna ulaşılmış, ancak modeller arasında belirgin bir farklılık oluşmadığı sonucu da vurgulanmıştır.

ABSTRACT

In this thesis, a theoretical study of metrics and methods of data mining algorithm comparison is documented and a comparison of several well known data mining algorithms is studied. Different kinds of data mining algorithms exist for different methodologies such as association, classification, clustering methods. A comparison of classification algorithms was performed using a breast cancer data including 204,949 records as a case study.

In the theoretical phase of the study, a general information about data mining, methodologies used as part of data mining process, preprocessing of the data, and description of comparison metrics is given. Since the main aim of this study is related to the comparison of algorithms, information about data mining and the methods is briefly mentioned and not detailed. The information given is just an overview of the whole process in order to enlighten how the comparison proceeds and on what basics it holds on.

In the application phase, a comparison of classification algorithms including J48, NaiveBayes, Logistic Regression and KStar is performed. It starts with the description of the tool, Weka, used for the application of algorithms on the data. Then, data source that was used to train and test the models is described. Structure and semantics of the source is studied in detail. Preprocess of the data and related work is mentioned that includes both the introduction of Arff format and data cleansing and restructuring. At the end, algorithms are applied and results of those

algortihms in terms of statistical figures are given and a comparison is performed using those metrics mentioned in the theoretical phase.

TEŞEKKÜR

Tez çalışmam süresince büyük yardımlarını gördüğüm, bilgi ve deneyiminden yararlandığım değerli hocam sayın Yrd. Doç. Dr. Abdullah BAYKAL'a, manevi desteklerinden dolayı eşim Zelal COŞKUN'a ve oğlum Azad COŞKUN'a, teşekkürlerimi sunmayı bir borç bilirim.

İÇİNDEKİLER

AMAÇ

Bu çalışmanın temel amacı veri madenciliği algoritmalarının karşılaştırılma yöntemleri üzerine teorik bir çalışma ortaya koyarak bir uygulama yapmak, sınıflandırma metotlarına ait temel algoritmalardan bazılarını büyük bir veri kaynağı üzerinde çalıştırarak ortaya çıkarılan modelleri karşılaştırmak, algoritmaların karşılaştırmalı bir değerlendirmesini yapmaktır.

1. GİRİŞ

Veri madenciliği alanı, bilgisayar teknolojisinin gelişmeye başladığı yıllardan sonra, eldeki verilerin büyük bir hızla artmasıyla ortaya çıkan veri analizi ihtiyacına bağlı olarak büyük bir hızla gelişmeye başlamıştır. Konu ile ilgili olarak bu kısa zaman diliminde pek çok akademik araştırma ve geliştirme yapılmıştır.

Veri madenciliği araç ve metotlarının gelişmesiyle iş dünyasından kaynaklanan, konuya yönelik talepten ötürü, algoritmaların ve yazılım araçlarının geliştirilmesine yönelik, hem iş çevresinde hem de akademik çevrede konuya yoğun bir ilgi oluşmuş, veri büyüklüğü ve algoritmaların kompleksliği nedeniyle daha iyi sonuçlar almanın yolları araştırılmış, bu yapılırken ortaya çıkan pek çok yöntemden hangisinin daha iyi olduğu gibi sorular ortaya çıkmıştır. Uygulanan teknoloji ve algoritmaların verimliliği her ne kadar problem alanına bağımlı olsa da akademik anlamda karşılaştırma yapılması ihtiyacı doğmuştur.

Veri analizinin istatistik bilimine bağlı olması, ayrıca yapay zeka ve makine öğrenme gibi temelde istatistik ve matematik bilimine bağlı farklı akademik disiplinlerin oluşturduğu veri madenciliği yöntemlerinin değerlendirilmesi doğal olarak yine istatistik biliminin temel kuramları üzerinden yapılagelmiştir.

Veri madenciliği çalışmalarında çok çeşitli yöntemler kullanılmaktadır. Farklı alanlarda çok geniş bir uygulama alanına sahip olduğu için bu alandaki çalışmalar durmaksızın devam etmekte, var olan yöntemler üzerinde iyileştirmeler yapılmakta

ve yeni yöntemler geliştirilmektedir. Aynı zamanda, matematik, istatistik, enformatik ve bilgisayar bilimlerindeki gelişmeler de bu alana yansımaktadır. Bu sebeple, veri madenciliği, geniş bir uygulama alanına sahip olup, gelişmeye açık, sadece akademik değil aynı zamanda iş dünyasının da yoğun ilgisini çeken bir alandır.

Veri madenciliği uygulama alanının çok geniş olması bu konuya olan ilgiyi de arttırmaktadır. Kullanılan pek çok model ve bu modellere ait farklı algoritmalar vardır. Bu algoritmalardan hangisinin daha efektif sonuçlar ürettiği, hangi algoritmanın hangi alanda daha başarılı olduğu sorusuna verilen cevaplar uygulamaların başarımını arttıracak ve yapılan işin verimini arttıracaktır. Bu sebeple algoritmaların karşılaştırılarak değerlendirilmesi büyük önem arz etmektedir.

Bu tez veri madenciliği algoritmalarının karşılaştırılması üzerine bir çalışmadır. Farklı sınıflandırma algoritmalarının nasıl karşılaştırılabileceği ve kullanılabilecek metrikler üzerinde durulmuş, çok bilinen dört farklı sınıflandırma algoritması karşılaştırılarak veri önişlemeden başlamak üzere model oluşturulması ve modellerin karşılaştırılması konusunda bilgi verilmesi amaçlanmıştır.

2. KAYNAK ARAŞTIRMASI

Veri madenciliği alanında karşılaştırma amaçlı yapılan çalışmalar veri madenciliği araçlarının ve algoritmalarının karşılaştırılmaları yönünde olmuştur. Çok sayıda algoritma olması, her algoritmanın kendi içinde farklı parametrelerle çalışması, her algoritmanın birden çok versiyonunun bulunması, farklı algoritmaların farklı amaca yönelik olması, kullanılan veri kaynağının farklı olması, algoritmaların farklı veri tiplerini desteklemesi ve veri üzerinde yapılan önişlemlerin uygulayıcıya bağlı olması gibi sebeplerle farklı sonuçlar elde edilmiştir.

Hızla gelişen teknolojiyi gözönüne aldığımızda, son on yıllar içerisinde yapılan bu çalışmaların güncelliğini koruyamayacağı aşikardır. Ancak temelini istatistik biliminden alan ve yapılan her türlü iyileştirmeye rağmen algoritmik olarak aslına benzeyen pek çok algoritmanın literatürde bulunan karşılaştırma sonuçları incelenmeye değerdir ve yapılacak olan yeni çalışmalarda bir temel olarak ele alınmaya devam edecektir.

Veri madenciliği araçlarının karşılaştırılmasına yönelik yapılan çalışmalar, ücretli ya da ücretsiz olarak kullanımda bulunan pek çok uygulamanın sınırlı bir sayıdaki kümesini kapsar. Bu karşılaştırmalarda daha çok uygulamaların kullanışlığı, arayüzleri, görselliği, desteklenen algoritmalar, platform uyumlulukları ele alınmıştır. Böyle bir çalışmaya Elder ve ark.'nın [1] yaptığı, var olan çeşitli ücretli ve ücretsiz veri madenciliği araçlarının karşılaştırması örnek olarak gösterilebilir. Bu çalışmada platform uyumluluğunu, desteklenen algoritmaları, veri analiz arayüzlerini, kullanım

kolaylığını, görsel yetkinliklerini ve modelleme metotlarını karşılaştırmışlardır. O yıllarda henüz gelişme aşamasında olan veri madenciliği araçlarının günümüzde kullanılan diğer araçları kapsamaması; karşılaştırılan araçların henüz bugünkü gelişmişlik seviyesine gelmemesi çalışma sonuçlarının geçerliliğini sorgulatır durumdadır.

Veri madenciliği algoritmalarının karşılaştırmasına yönelik pek çok akademik makale bulunmaktadır. Bu çalışmalar farklı veri grupları kullanılarak -ki bu farklı veri gruplarında farklı veri tipleri, farklı veri dağılımı, farklı veri nitelikleri söz konusudur- belli başlı temel algoritmalar kullanılarak yapılmıştır. Dolayısıyla farklı çalışmalarda farklı sonuçlar elde edilmiş, çoğu çalışma sonucu birbiriyle örtüşmediği gibi birbiriyle tamamen zıt sonuçlara da ulaşılmıştır.

1994 yılında altısı üniversitede akademik çalışma yapan, diğer altısı üniversite dışında endüstriyel araştırmalarda bulunan toplam 12 kişilik bir ekiple European StatLog Projesi kapsamında sınıflandırma algoritmaları üzerine çalışma yapılmıştır. Yapılan bu çalışma sonuçları Michie ve ark. [2] tarafından "Machine Learning, Neural and Statistical Classification" adında kitap haline getirilmiştir. Bu kitapta hangi algoritmanın endüstri ihtiyaçlarını daha iyi karşıladığını bulmaya yönelik kapsamlı testlerden oluşturdukları sonuçları yayınlamışlardır. Çalışmalarında sınıflandırma tekniğine ait istatistiksel metotları, makine öğrenme metotlarını ve yapay sinir ağları metotlarını karşılaştırmışlar. Ağırlıklı matris, kredi risk değerlendirme, resim ve diğer tarz veri kümelerini farklı algoritmalarla analiz ederek, elde ettikleri analiz sonuçlarına yer vermişler. Kredi riski veri kümesi üzerinde

yapılan veri madenciliği çalışmalarında en başarılı algoritmaların karar ağaçları olduğu görülmüş, resim veri kümesi üzerinde yapılan çalışmalar obje tanıma ve segmentasyon gibi iki temel işlem olarak irdelenmiş; obje tanımada istatiksel metotlar ve yapay sinir ağları genel olarak daha başarılı bulunurken segmentasyon için makine öğrenme algoritmalarının iyi sonuçlar verdiği, istatistiksel metotların çok kötü sonuçlar verdiği ifade edilmiştir. Özel olarak resim veri kümesinde en iyi sonuçların "k-en yakın komşuluk algoritması"nda - k-nearest neighbour - alındığı belirtilmiştir. Ağırlıklı Matris veri kümesinde ise makine öğrenme algoritmasının çok kötü sonuçlar verdiği ki tahminlenecek yeni verinin tümü hiçbir model kullanılmaksızın belirli bir sınıfa atandığında bile daha doğru sonuçlar verdiği belirlenmiştir. Diğer veri kümelerinde ise bazen makine öğrenme algoritmalarının, bazen de yapay sinir ağları algoritmalarının başarılı olduğu sonucuna ulaşılmıştır. Bu sonuçlardan, yazarlar, farklı problem alanında farklı algoritmaların başarılı olduğu gözlemini yaparken, bir genelleme olarak da yapay sinir ağları algoritmalarının başarılı olduğu veri kümelerinde istatistiksel metotların da başarılı olduğu sonucuna görüşlerinde yer vermişlerdir.

Wilson ve Sharda [3], şirketlerin iflas tahminlemesinde, diskriminant analizi ve yapay sinir ağları metotlarını karşılaştırmışlardır. Çok hisseli ya da halka açık şirketlerin finansal geleceklerini tahminlemek için kullanılagelen temel istatistik metotlarından diskriminant analizi ve lojistik regresyon metotlarını veri madenciliğinin yapay sinir ağları metoduyla karşılaştırarak yapay sinir ağları sonuçlarının daha doğru sonuçlar ürettiği sonucunu çıkarmışlardır.

Lin ve ark. [4] 2004 yılında yaptıkları çalışmada öğrencilerin, seviyelerine göre, uygun seviye grubuna atanması konusunda Fischer diskiriminant analizini ve çekirdek-tabanlı diskriminant analizini de içeren farklı diskriminant analizlerini kullanarak yaptıkları karşılaştırmada öğrenci seviye tespitinde çekirdek-tabanlı diskriminant analizinin daha uygun sonuçlar oluşturduğu sonucuna ulaşmışlardır.

King ve ark. [5] sembolik öğrenme (CART, C4.5, NewID, ITrule,Cal5, CN2), istatistik (NaiveBayes, k-en yakın komşuluk, çekirdek-yoğunluk -kernel density-, doğrusal diskriminant -linear discriminant-, ikilenik diskriminant -quadratic diskriminant-, lojistik regresyon, projection pursuit, Bayes Ağları -Bayesian Networks-) ve yapay sinir ağları (geri yayınım -backpropogation-, merkez tabanlı uzaklık fonksiyonu -radial basis function-) metotları arasında karşılaştırma yapmışlardır. Bu metotları beş adet resim, iki adet tıp, ikişer mühendislik ve finans veri kümeleri üzerinde uygulamışlardır. Daha iyi sonuçlar üreten algoritmanın, üzerinde araştırma yapılan veri kümesine bağlı olduğu sonucuna ulaşmışlardır ve örnek olarak ikili değerli niteliklerin %38 oranının üzerinde olduğu veri kümelerinde sembolik öğrenme algoritmalarının daha verimli sonuçlar ürettiğini belirtmişlerdir.

Wu ve ark. [6] 2007 yılında yazdıkları bir araştırma makalesinde 2006 yılı aralık ayında düzenlenen Uluslararası Veri Madenciliği Konferansında belirlenmiş olan en yaygın 10 veri madenciliği algoritmasını ele almış, bunlar hakkında teorik bilgileri derlemiş ve algoritmaların detaylı tanımıyla beraber ileriye dönük yapılabileceklere makalelerinde yer vermişlerdir. Makale tarama ve oylama metodu ile

belirledikleri en yaygın 10 veri madenciliği algoritmaları şunlardır: C4.5, *k*-en yakın komşuluk, SVM, Apriori, EM, PageRank, AdaBoost, *k*NN, Naive Bayes, ve CART.

Sabzecari ve ark.[7] özel bir bankanın kredi derecelendirme amaçlı veri kümesi üzerinde uyguladıkları veri madenciliği metotlarını karşılaştırmışlardır. Bankalar, kredi verirken, veri madenciliği yöntemleri ile kredi başvurusunda bulunan müşterileri değerlendirerek müşteriye kredi verilmesinin uygun olup olmadığını belirlemektedirler. Sabzecari ve ark. Probit ve lojistik regresyon, CART, yapay sinir ağları, bagging ve MARS algoritmalarını karşılaştırarak sonuçlarını değerlendirmişlerdir. Oldukça küçük bir veri kümesi üzerinde yaptıkları bu çalışma sonucunda istatistiksel modeller arasından lojistik regresyon ve makine öğrenme modelleri arasından da bagging modelinin daha başarılı sonuçlar ürettiğini görmüşlerdir.

Veri madenciliği algoritmaları sürekli geliştirilmekte, var olan algoritmalar üzerinde geliştirmeler yapılmakta, farklı metotlar uygulanarak daha verimli algoritmalar elde edilmeye çalışılmaktadır. Bu yeni yaklaşımlar var olan eski ve yaygın algoritmalarla karşılaştırılarak değerlendirilmektedir. Rajavarman ve ark. [8] geliştirdikleri genetik algoritma tabanlı sınıflandırma metodunu yaygın olan ID3, ID3 boosting, yapay sinir ağları ve NaiveBayes algoritmaları ile karşılaştırarak sonuçlarını yayınlamışlardır. Üç farklı veri kümesi ile yaptıkları değerlendirme sonuçlarına göre kendi genetik algoritma sonuçlarının diğer algoritmalar karşısında daha başarılı olduğunu ifade etmişlerdir.

Zurada ve ark. [9] sağlık endüstrisinde kötü kredilerin belirlenmesinde karşılaştırdıkları yapay sinir ağları, karar ağaçları, lojistik regresyon, hafıza-tabanlı sebepleme ve bütünleştirilmiş model arasından yapay sinir ağlarının, lojistik regresyon algoritmasının ve bütünleştirilmiş modelin daha iyi kesinlik oranına sahip sonuçlar ürettiğini, karar ağaçlarının ise iyi kredi sınıflandırmasını daha yüksek bir doğruluk derecesiyle tesbit ettiğini belirtmişlerdir.

Bugüne kadar yapılan deneysel araştırmaların çoğunda görülmektedir ki algoritmaların başarı oranı büyük oranda kullanılan veri kümesine bağlıdır. Bu yüzden var olan pek çok farklı makalede farklı sonuçlara ulaşılmış, kimi araştırmada bulunan sonuç diğer araştırmalarda bulunan sonuçtan farklı çıkmıştır.

3. ÖN BİLGİLER

3.1. VERİ MADENCİLİĞİ TARİHÇESİ

Veri madenciliği son on yıllarda iş dünyası ve yazılım dergilerinde kendine yer edinmiştir. Halbu ki son on yıllar öncesinde pek az kişi veri madenciliği terimini duymuştur. Aslında veri madenciliği kökeni çok eskilere dayansa da bir alan olarak literatürde 1980'lerden itibaren yer almıştır.

Veri madenciliğinin esası üç temel kola bağlı olarak gelişmiştir. Bu kollardan ilki, en önemlisi ve en eskiye dayananı klasik istatistik bilimidir. Regresyon analizi, standart dağılım, standart sapma, diskriminant analizi, güven aralıkları gibi verileri ve veriler arasındaki ilişkiyi inceleyen bu yöntemler klasik istatistik çalışmalardır. Bu yöntemler ileri düzey istatistiksel analizin temelini oluşturan yapı taşlarıdır. Açıkça, klasik istatistiksel yöntemler bugün kullanılmakta olan veri madenciliği araç ve metotlarının esasını oluşturur.

Veri madenciliğinin üzerinde yeşerdiği bir diğer kol yapay zeka (AI) dır. Yapay zeka, sezgisel - heuristic - yaklaşımı temel alarak, insan-benzeri-düşünebilme prensibiyle, istatistikten farklı metotlarla, istatistiksel problemlere yaklaşır. Bu yaklaşım uygulanabilirlik açısından yüksek kapasitede bilgisayar gücü gerektirdiği için, güçlü bilgisayar sistemlerinin kullanıcının hizmetine sunulmaya başlandığı 1980'li yıllara kadar pratik uygulamalarda yer edinememiştir. Hala pek çok uygulama, süper bilgisayarlar gibi kişisel bilgisayarlardan daha güçlü makineler

gerektirdiği için, bu uygulamaların pek çoğunun büyük şirket ya da devlet kurumları ile sınırlı kaldığı söylenebilir.

Diğer ve son temel kol da yine köklerini istatistik ve yapay zekadan alan makine öğrenmesidir. Aslında makine öğrenme, yapay zekanın sezgisel - heuristic - yöntemleri ileri düzey istatistiksel yöntemlerle harmanlayıp evrimleşerek geliştiği ileri düzey halidir denebilir. Makine öğrenme, uygulandığı bilgisayar sistemlerinde, istatistiksel ve yapay zeka algoritmaları kullanarak eldeki verinin değerlendirilmesine, bu verilerden sonuçlar çıkarılmasına ve bu sonuçlara bakılarak kararlar alınmasına olanak sağlar.

Temel olarak veri madenciliği, öğrenme yöntemlerinin iş ve bilimsel verilere uygulanarak bilgi çıkarılmasıdır. Veri madenciliği, istatistik, yapay zeka ve makine öğrenme disiplinlerinin gelişmesiyle ortaya çıkan, eldeki veriden öğrenme yoluyla gizli bilgileri ve örüntüleri ortaya çıkararak ileriye dönük tahminler yapmayı amaçlayan ve geçmişi on yıllara dayalı yeni bir bilim dalıdır. İş ve bilim alanında, normalde çok yoğun veri kümelerinden çıkarılması imkansız bilgiyi çıkarmada gün geçtikçe daha çok kabul görmektedir.

3.2. VERİ MADENCİLİĞİ HAKKINDA TEMEL BİLGİLER

Veri madenciliği büyük veri kaynaklarındaki gizli, önemli ve yararlı bilgilerin bilgisayar yardımıyla keşfedilmesidir. Veriler arasındaki benzerliklerin, örüntülerin ya da ilişkilerin çıkarılması amacıyla uygulanan işlemler bütünüdür. Veri

madenciliğinin ekonomi alanında pazar araştırması, müşteri profilinin çıkarılması, sepet analizi; bankacılıkta risk analizi, sahtekarlıkların saptanması; bilişimde web verilerinin analizi, ağ güvenliği, belgelerin sınıflandırılması gibi uygulamaları mevcuttur. Bunların dışında meteorolojide, tıpta, temel bilimlerde, ilaç biliminde ve diğer alanlarda da uygulamaları mevcuttur. Her ne kadar veri madenciliği yeni bir alan olsa da, aslında daha önceleri ekonomistler, istatistikçiler, hava durumu tahminleyicileri, eldeki verileri kullanarak ileriye dönük tahminler yapmakla uğraşıyorlardı. Son onyıllarda veri miktarlarındaki hızlı büyüme, farklı tarzlardaki verilerin farklı algoritma ihtiyacı, bu disiplinin kendi ayakları üzerinde durma gereksinimine sebep olmuştur.

Gelişen teknoloji ile birlikte hayatımızdaki veriler gün be gün büyümekte, daha önceleri kilobaytlarla ifade edebildiğimiz kişisel bilgisayarlardaki veriler artık megabaytlar, gigabaytlar ile ifade edilebilmektedir. Daha önceleri çöpe atılabilir tarzdaki veriler bile, bilgi depolama aygıtlarının gelişmesiyle beraber depolanmaya başlanmıştır. Günlük hayatımızı kolaylaştıran bankacılık işlemleri, online sistemler, internetin yaygınlaşması, bilgiye kolay erişim ve bilgi aktarım gereksinimini arttırmıştır. Bu gibi gelişmeler veri miktarının hızlı bir şekilde artmasında bir faktör olmuştur. Doğrulanabilirliği mümkün olmasa da, bir tahmine göre dünyadaki toplam veri miktarı her 20 ayda bir ikiye katlanmaktadır. Büyük şirketlerin, okulların, hastanelerin, bankaların, alışveriş merkezlerinin, diğer özel ve kamu kurumlarının veri bankaları büyük veri yığınlarından oluşmaktadır. Bu veriler analiz edilerek ileriye dönük politika belirlemede, geleceği öngörmede ya da varolan sistem hakkında karar alıcı mekanizmalarda önemli rol oynarlar. Veri madenciliği bu büyük

miktardaki verilerin analiz edilmesi için uygulanması gereken basamakların bütününü kapsar.

3.2.1. Veri

Veri, nesneleri, işlem ya da eylemleri niteliklerinin değerleriyle tanımlayan bilgi kümesidir. Nesneler ya da işlemler niteliklerden oluşur. Örneğin nesnemiz 'insan' ise, 'göz rengi', 'yaşı', 'adı', 'ikametgah adresi' onun nitelikleridir. "İnsan" nesnelerinin oluşturduğu küme ise veridir. Para çekme işlemini ele alırsak 'çekilen tutar', 'çekildiği hesap numarası', 'çekiliş saati', 'çekildiği yer' bu işlemi tanımlayan niteliklerdir. İçerisinde birden çok para çekme işlemine ait bilgiyi barındıran küme ise veridir.

3.2.2. Veri Önişleme

Nitelikler, sayısal, nominal ya da katarlar şeklinde değer alabilirler. İdeal bir veri kümesinde bütün verilerin eksiksiz, hatasız ve tutarlı olması beklenir. Ancak gerçek hayatta bu pek geçerli değildir. Verilerin analize uygun bir yapıya getirilmesi işlemine *veri önişleme* denir.

Veri önişleme adımı bir veri madenciliği çalışmasının oldukça büyük bir kısmını kapsar ve analizin doğru sonuçlara ulaşmasında ve efektif bir şekilde uygulanmasında büyük öneme sahip olup problem alanında bilgi sahibi olmayı gerektirir.

Bu adım veri madenciliği aşamalarının ilk ve en uzun basamağını oluşturur. Veri temizleme, veri birleştirme, veri dönüşümü ve veri azaltma işlemlerini kapsar.

3.2.2.1. Veri temizleme

Kullanıcı hataları, program hataları, bazı otomatize edilebilecek işlemleri kullanıcıya bırakma, veri girişinin önemsenmemesi gibi sebeplerle veri kümelerinde eksik ya da gürültülü veriler oluşabilir. Veri üzerindeki bazı nitelikler yanlış değer taşıyabilecekleri gibi, eksik, geçersiz veriler de olabilir. Veriler üzerinden faydalı ve doğru sonuçlar çıkarabilmek için bu tip bilgilerin düzeltilmesi ya da gözardı edilmesi gerekir. Veri temizleme basamağında bu tip veriler tamamlanır, ayıklanır ya da tutarsız veri varsa bu tutarsızlıklar belirli mantıksal işlemlerle düzenlenir.

Eksik nitelik değeri taşıyan veriler, gözardı edilebilir, rastgele değerlerle doldurulabilir, bu niteliğe özgü global bir değerle doldurulabilir, o niteliğe ait ortalama değerle doldurulabilir, aynı sınıfa ait verilerin nitelik değerlerinin ortalamasıyla doldurulabilir, ya da veri kümesinin o niteliğe ait olasılığı en yüksek değeriyle doldurulabilir.

Verilerimiz içerisinde tutarsız, ya da gerçeğe aykırı olan gürültülü veri diye adlandırılan hatalı veriler de bulunabilir. Gürültülü veriler, bölmeleme, demetleme ya da eğri uydurma gibi metotlarla düzeltilebilir. Bölmeleme işlemi eldeki verinin sıralanarak eşit bölmelere ayrılması ve her bölmenin kendisine ait ortalama ya da uç değerlerle ifade edilmesidir. Böylece verilerdeki hata miktarlarının minimize

edilmesi amaçlanır. Demetleme ile benzer veriler aynı demette olacak şekilde gruplanır ve dışarıda kalan veriler göz ardı edilir. Böylece muhtemel yanlış ölçümler ayıklanmış olur. Eğri uydurma metodu ile ise nitelik değeri diğer niteliklere bağlı olarak belli bir fonksiyona uydurulur. Bu fonksiyon kullanılarak nitelik değerindeki tutarsızlıklar giderilir.

3.2.2.2. Veri birleştirme

Bazı durumlarda birçok veri kaynağından yararlanarak veri kümemizi oluşturmamız gerekir. Veri birleştirme denilen bu işlemde farklı kaynaklardan gelen veriler aynı veri kümesi altında birleştirilir. Farklı kaynaklarda aynı nitelik için farklı değerler, ölçü birimleri ya da derecelendirmeler kullanılmış olabilir. Bu durumlarda nitelik değerlerini birleştirirken dönüşüm yapmak gerekir. Farklı kaynaklarda aynı nitelikler farklı nitelikmiş gibi ele alınmış olabilir ya da birleştirme sonucunda gereksiz veriler oluşabilir. Bu tip niteliklerin belirlenmesi, gereksiz verilerin ayıklanması gerekir.

3.2.2.3. Veri dönüşümü

Verilerde bazı nitelik tipleri uygulanacak olan algoritmaya uygun olmayabilir ya da veri nitelikleri belirleyici olmayabilir. Veri dönüşümü yapılarak nitelikler algoritmaya uygun hale getirilir ve nitelikler daha belirleyici olacak şekilde dönüştürülebilir. Bunun için normalizasyon ya da nitelik oluşturma işlemleri yapılabilir.

Normalizasyon işlemi numerik değerli nitelikler için uygulanır. min-max normalizasyon, z-score normalizasyon ve ondalık normalizasyon numerik nitelikler üzerinde uygulanabilecek yöntemlerdir.

Sürekli ve numerik değerli bir A niteliğinin aldığı maximum değer a_{max}, minimum değer a_{min}, niteliğin değeri X, aralığın taşınacağı yeni aralık ise $[a'_{min}, a'_{max}]$ olsun. Bu niteliğin X değerinin dönüşüm sonucu alacağı değer min-max normalizasyon ile (1), z-score normalizasyon ile (2), ondalık normalizasyon ile (3) hesaplanabilir.

$$\frac{(X - \text{ı}_{min})}{(a_{max} - \text{ı}_{min})}(a'_{max} - \text{ı}'_{min}) + \text{ı}'_{min} \qquad \text{Min-max normalizasyon} \qquad (1)$$

$$\frac{(X - \text{nean})}{\text{Standard Deviation}} \qquad \text{z-score normalizasyon} \qquad (2)$$

$$\frac{X}{10^j} \qquad \text{Ondalık norrmalizasyon} \qquad (3)$$

(j | max(|v/10^j|) < 1 ifadesini sağlayan en küçük tamsayı)

Bazı algoritmalar belli tip nitelikler üzerinde çalışırlar. Özellikle sürekli veriler algoritmalar tarafından beğenilmeyen veri tipleridir. Ayrık veriler ise algoritmaların uygulanabilirliği için daha uygundur. Bu yüzden sürekli verileri ayrık değerli verilere çevirmek gerekebilir. Genelde sınırsız sayıda olabilecek sürekli değerlerin, sınırlı sayıda olan ayrık değerlere dönüştürülmesiyle, nitelik, kavram özelliği kazanmış olur. Ayrıca, niteliklerin dijital ortamda daha büyük hafıza alanı gerektiren kompleks değişkenler yerine küçük alanlar gerektiren basit tiplere

dönüştürülmüş olmasından dolayı, veri bankası hacmi de küçülmüş olur. Niteliklere ayrık değerler vermek için farklı yöntemler uygulanabilir. Niteliğin sürekli değerleri gruplara ayrılarak her bir gruba ayrık değerli bir etiket verilebilir. Histogramlar kullanılabilir. Benzer değerler demetlenerek her demete ayrık değerli etiket verilebilir veya nitelik değeri matematiksel değerlendirmelerden geçirilerek ayrık değerlere dönüştürülebilir. Böylece ayrık değerli nitelikler oluşturulabileceği gibi nominal değerli yeni kavramlar da oluşturulabilir.

Nitelik oluşturma işlemi, orijinal niteliklerden daha değerli bilgi içeren yeni nitelikler oluşturmak ve veri madenciliği algoritmasının başarımını arttırmak için uygulanır.

3.2.2.4. Veri azaltma

Analiz edilecek olan verinin aşırı büyük olması, uygulanacak olan algoritmanın daha uzun bir sürede tamamlanmasına ve aslında sonucu etkilemeyecek gereksiz işlemlere sebep olur; ayrıca bazı algoritmalar belirli tip veriler üzerinde çalışır, bu tipte olmayan verilerin gözardı edilmesi ya da dönüştürülmesi gerekir. Bu sebeple veri önişleme aşamasında uygulanacak olan bir diğer işlem de sonucu etkilemeyecek bir şekilde gereksiz olan bilgilerin silinmesi, birleştirilmesi ya da diğer bazı yöntemlerle daha anlamlı ve algoritmaya uyumlu hale getirilmesidir. Nitelik birleştirme, nitelik azaltma, veri sıkıştırma, veri küçültme, veri ayrıştırma ve kavram oluşturma gibi yöntemlerle eldeki veri, sonucu değiştirmeyecek şekilde daha verimli bir hale getirilmektedir.

Veri tanımındaki bazı nitelikler birbirleriyle ilişkili olabilir. Birden fazla sayıdaki bu tip nitelikler birleştirilerek veri bankası küçültülebilir. Veri tanımındaki bazı nitelikler sonuca etki etmeyecek ya da sonucu değiştiremeyecek kadar düşük önemde iseler bu tip nitelikler seçilerek elimine edilirler. Nitelik seçme, problem alanına yönelik bilgiyi değerlendirerek yapılabileceği gibi istatistiksel yöntemlerle, karar ağaçlarıyla ya da bilgi kazancı değerleriyle tespit edilebilir. Veri sıkıştırma, büyük verinin sıkıştırma algoritmalarıyla boyutunu küçültmeyi, böylece veri saklamayı ve veri erişimini hızlandırmayı amaçlar. Bu yöntemin verimli olması için uygulanacak olan algoritmanın sıkıştırılmış veri üzerinde çalışabilmesi gerekir.

3.2.3. Veri Madenciliği Modelleri

Veri madenciliği modelleri kestirime dayalı ve tanımlayıcı olarak ikiye ayrılır:

- *Kestirime dayalı modeller*: sınıflandırma, eğri uydurma, zaman serileri
- *Tanımlayıcı modeller*: demetleme, özetleme, bağıntı kuralları

3.2.3.1. Bağıntı kuralları metodu

Nesnelerin nitelikleri arasında ilişkiler bularak, ilişki kurallarından oluşan bir model oluşturur. 1993 yılında Agrawal ve ark. [10] tarafından ilk kez ortaya atıldıktan sonra büyük ilgi toplamış ve üzerinde pek çok araştırma, geliştirme ve iyileştirmeler yapılmıştır.

Sınıflandırma tekniğinden farkı sadece sınıfı değil, diğer herhangi bir niteliği ya da birkaç niteliği de tahminleyebilir model oluşturmasıdır. Çok küçük veri

kümelerinden bile aşırı sayıda bağıntı kuralı elde edilebilir. Oluşturulmuş olan bu kurallardan çoğu, verileri doğru tahminlemeden uzak kalabilir; bu nedenle bağıntı kuralları bulunurken belirli bir doğrulama yetisinin üzerindeki kurallar dikkate alınırlar. Kural sayısını aza indirgemek ve geçerliliği olan ilişkiler üzerinde yoğunlaşmak için destek ve güven sınırları tanımlanmıştır. Bağıntı kuralları incelenirken minimum destek ve minimum güven değerleri belirlenerek bu değerlerin altında kalan değerlere sahip ilişkiler gözardı edilirler.

$I=\{i_1,i_2,i_3,\ldots\ldots i_n\}$ ve $T=\{\ t_i \mid t_i \subseteq\ \ \}$ olsun.

Örneğin, bir markette bulunan nesnelerin {bira, süt, peynir, …} kümesi I; bu marketten yapılan alışveriş işlemlerine ait {{kola, makarna}, {yumurta, süt, peynir}, {ekmek, peynir}, {makarna, süt}, … {bira, süt}} gibi kümeler de T kümesi olarak düşünülebilir.

$X \subset I$, $Y \subset I$ ve $X \cap Y = \emptyset$ olacak şekilde X ve Y birer alt küme ise; $X \rightarrow Y$ bir bağıntı kuralını ifade eder. Örneğin X={yumurta, peynir} Y= {bira} şeklinde düşünülürse bu bağıntı, yumurta ve peynir alanların bira da aldığını gösteren bağıntı kuralıdır. Her bağıntı kuralının destek ve güven değeri hesaplanır.

Minimum destek ve minimum güven parametreleri her bağıntı için hesaplanan destek ve güven değerleri ile sınanarak bağıntının göz ardı edilip edilmeyeceği belirlenir. Destek ve güven değerlerinin bir $X \rightarrow Y$ bağıntısı için hesaplanması (4) ve (5)'te verilmiştir.

$$\text{Destek} = \frac{s(X \cup Y)}{s(T)} \tag{4}$$

$$\text{Güven} = \frac{s(X \cup Y)}{s(X)} \tag{5}$$

Bulunmuş olan bir bağıntı kuralının, T veri kümesindeki geçerli olduğu verilerin oranına *destek* (4); kuralı oluşturan elemanlar kümesinin geçerli olduğu sınanan kümedeki oranına ise *güven* denilmektedir (5).

Bağıntı kurallarını bulmak için pek çok algoritma vardır. Bunlardan en bilineni Apriori Algoritmasıdır. Algoritma, "Aşağı Kapalılık Özelliğini" kullanarak sık geçen elemanlar kümesini oluştururken gereksiz işlemler yapmayı önler. "Aşağı Kapalılık Özelliği", minimum destek sınırını sağlayan bir elemanlar kümesinin boş olmayan tüm alt kümelerinin de minimum destek sınırını sağlayacağını söyler. Bu doğruluğu açık bir önermedir. Çünkü bir elemanlar kümesi minimum destek sayısından fazla sayıda bulunuyorsa bu kümenin boş olmayan bütün alt kümeleri de en azından bu küme kadar destek sayısına sahip olmalıdır. Literatürde Apriori algoritmasının pek çok versiyonu mevcuttur.

Bağıntı kurallarında kullanılan minimum destek sınırı, hem önemsiz ilişkilerin bulunarak modelin gereksiz derecede büyük olmasını önler, hem de algoritmaların daha verimli çalışması için zemin hazırlar. Ancak, bazen, aslında bulunması gereken, fakat minimum destek sınırı sebebi ile göz ardı edilen ilişkilerin gözden kaçmasına sebebiyet verebilir. Bir markette yapılan işlemlerdeki bağıntı kurallarını incelediğimizi düşünelim. Bu markette televizyon satışı da yapılıyor,

ekmek satışı da yapılıyor. Satılan ekmek sayısı günde yüzleri bulurken üç günde bir televizyon satışı yapılıyorsa içerisinde televizyon geçen kümelerin destek sayısının minimum destek sayısı altında kalacağı ve bu yüzden bazı ilişkilerin bulunamayacağı görülebilir [11]. Bu probleme, benzer frekansa sahip nesneler gruplandırılarak, farklı gruplara farklı minimum destek seviyesi kullanma yolu ile çözüm bulunabilir. Ancak bu durumda farklı gruplardaki nesnelerin çapraz ilişkileri gözden kaçmış olacaktır. Diğer bir çözüm ise her nesneye kendine özgü bir minimum destek seviyesi tanımlamakla olur. Çoklu Minimum Destek Seviyeli Bağıntı Algoritmaları farklı nesnelere farklı minimum destek seviyeleri verilerek yapılan bağıntı kuralları algoritmalarıdır ve MS-Apriori Algoritması bu yöntemi kullanır [12].

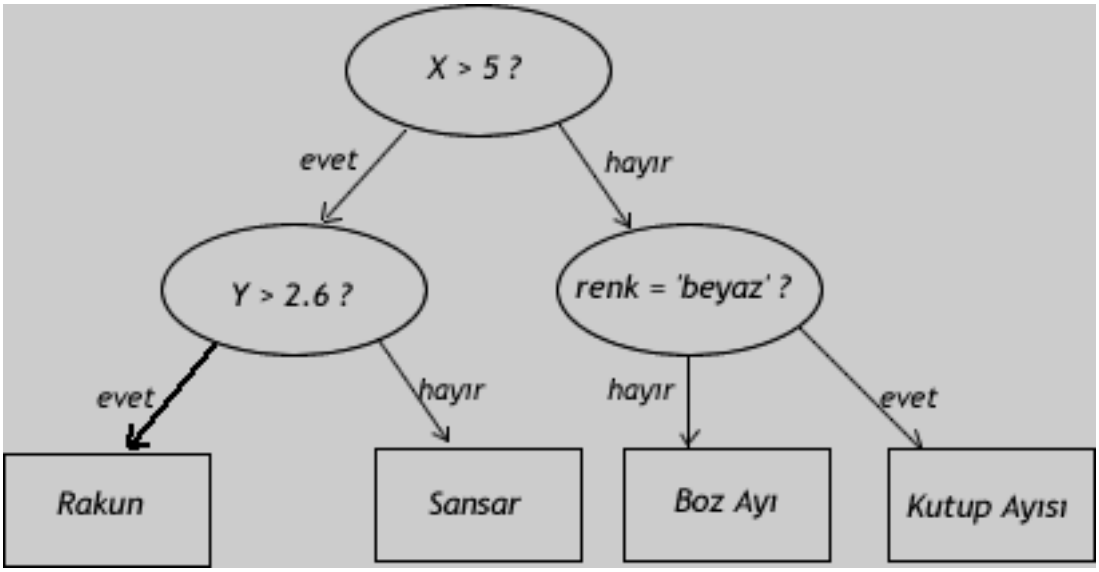
3.2.3.2. Sınıflandırma

Eldeki yoğun veriyi analiz edip, nesnelerin niteliklerini kullanarak her nesneyi yine bu nesnelerin bir niteliği olan özel bir sınıfa atama işlemidir. Elimizdeki nesneler $A=\{A_1, A_2, A_3, \ldots A_{|A|}\}$ şeklinde niteliklerden oluşsun ($|A|$, nitelik sayısı, başka bir ifade ile A nitelikler kümesinin eleman sayısını ifade eder). Elimizdeki her bir nesne, $\{A_1, A_2, A_3, \ldots A_{|A|}\} - \{C_1\}$ nitelikleri değerlendirilerek, bir c sınıfına atanır. Burada $C=\{C_1\}$ sınıf niteliği kümesi, c ise C_1 niteliğinin alabileceği değerlerden biridir. Sınıf niteliği, A kümesinin elemanlarından birisidir. Bu niteliğin alabileceği değerler sayısı ayrık ve sınırlıdır. Amaç, elimizdeki D veri kümesindeki örnekleri analiz ederek, örnekleri sınıflandırabilecek bir tahminleyici/sınıflandırıcı fonksiyon bulmak ve daha sonra bu kümede bulunmayan yeni bir örnekle karşılaş-tığımızda bu fonksiyonla yeni örneği de sınıflandırabilmektir. Karar ağaçları, Naive-

Bayes, SVM (Support Vector Machines), bağıntı tabanlı sınıflandırma, yapay sinir ağları ve genetik algoritmalar en bilinen sınıflandırma teknikleridir.

Karar ağaçları: Karar ağaçları çok bilinen, yaygın kullanımlı ve güçlü sınıflandırma metodudur. Diğer sınıflandırma metotlarına göre en güçlü yanı diğer tekniklere göre ürettiği modelin okunulabilirliğinin ve insanlar tarafından değerlendirilebilirliğinin yüksek olmasıdır. Ağaç formunda, yaprak düğümler ve sınama düğümlerinden oluşur. Yaprak düğümler bu düğüme ait nesnelerin sınıflarını ihtiva eden uç düğümlerdir. Sınama düğümleri ise o düğüme ait nesneleri bir niteliğe göre karşılaştırmaya alarak dallanan, iç düğümlerdir. Her nesne, ağacın kökünden başlanarak, varılan her sınama düğümünde o düğümdeki karşılaştırma sonucuna göre uygun yol seçilerek uç düğüme ulaştığında, ulaşılmış olan yaprak düğümün sınıf değerini alır.

Sınıflandırılması istenen yeni bir nesne geldiğinde karar ağacı kökten aşağı doğru taranarak, yaprak düğüme ulaşıldığında o düğümün sınıf etiketi ile sınıflandırılır. Grafik 3.2.1.'de sanal nitelik ve değerlerden oluşturulmuş örnek bir karar ağacı görülüyor. Karar ağacının kök düğümünde X niteliği 5 değeri ile karşılaştırılıyor, karşılaştırmanın değerine göre ağaç farklı dallara ayrılıyor. Ulaşılan alt düğümlerde farklı nitelikler karşılaştırılarak, bir uç düğüme ulaşıncaya kadar aynı yöntem devam eder. Yaprak düğümlerde ise o düğüme ulaşan nesnelerin sınıfları yer alır.



Grafik 3.2.1. Karar ağacı örneği

Karar ağaçlarının uygulandığı ID3, C4.5, C5.0, J48, CART algoritmaları en bilinen uygulamalardır.

Karar ağaçlarının implementasyonu ağaç oluşturma ve ağaç budama basamaklarından oluşur.

Ağaç oluşturma: Veri kaynağındaki bütün nesneleri içeren kök düğümden başlar, yinelemeli olarak her düğümde var olan nesneleri seçilecek olan bir niteliğe göre farklı dallara ayırarak bütün nesneleri sınıflandıracak şekilde yaprak düğümlere bölene kadar, ya da ayırım yapıcı bir nitelik kalmayana kadar devam eder. Sınama düğümlerinde eldeki nesnelerin hangi niteliğe göre alt düğümlere bölündüğünde en çok verimin alınacağını bulmak ve dallanmayı bu niteliğe göre yapmak algoritmanın gücünü arttırır. Nesneler alt düğümlere bölündüğünde alt düğümlerdeki nesnelerin homojenitesi ne kadar yüksek olursa o düğümdeki dallanma o kadar verimli olur. Bu sebeple, her düğümde, sınaması yapılacak olan nitelik (o düğümdeki nesneleri alt

düğümlere böldüğünde) homojenite bakımından en yüksek kazancı sağlayacak nitelik olarak seçilir.

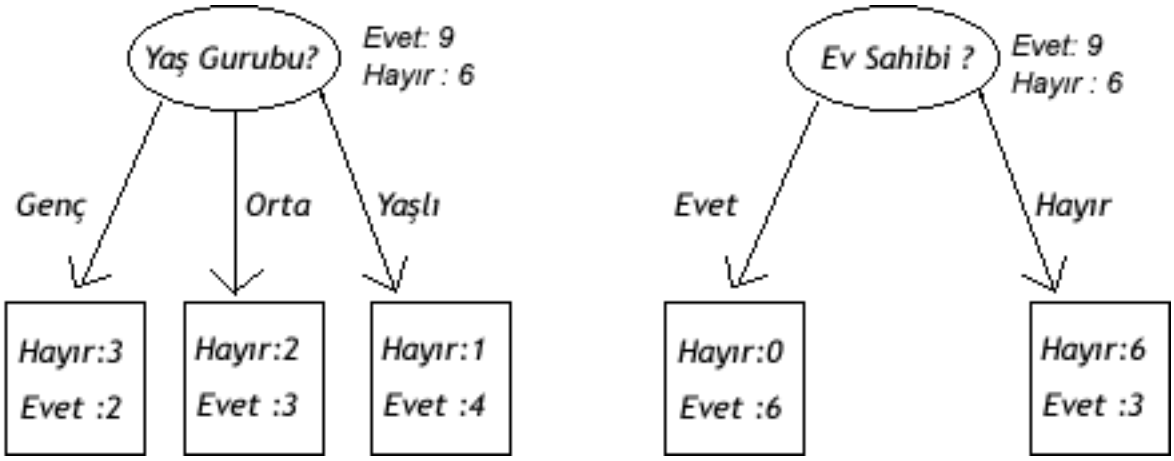
Bilgi Kazancı – Information Gain: Nitelik seçiminde, düğümlerin rastgelelik ya da düzensizlik değerini veren entropilerinden faydalanılır. Düğümlerin entropileri hesaplanarak ana düğümden alt düğümlere geçerken elde edilen bilgi kazancı -Information Gain- hesaplanır, en fazla bilgi kazancı sağlayan nitelik o düğümde sınama niteliği olarak kullanılır.

Entropi değeri 0 –1 arasında değişir. Aynı sınıf nesnelerden oluşan bir küme için 0, farklı sınıfa ait eşit sayıdaki nesnelerden oluşan bir küme için 1 değerini alır. Entropi değeri D düğümünde var olan her sınıfın olasılık değerinin logaritmik değeri ile çarpımlarının toplamıdır, (6) ile hesaplanır.

$$\text{Entropi (D)} = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j) \qquad (6)$$

Elimizde kredi derecelendirme verisi olsun. Karar ağacını oluştururken kök düğümdeki sınama kriterini, yaş grubu mu yoksa kişinin ev sahibi olup olmadığı niteliği olarak mı seçmek doğru olur? Grafik 3.2.2.'ye bakıldığında uç düğümlerde sınıf etiketi olarak baskın sınıf değerini kullanırsak "yaş grubu" kriteri kullanıldığında 5 hata; "ev sahibi" kriteri kullanıldığında ise 3 hata oluştuğunu görürüz. Dolayısı ile "ev sahibi" niteliğini sınama kriteri olarak seçmek daha doğru olur. Günümüzdeki algoritmaların çoğu düğümlerin entropi bilgilerini kullanarak

bilgi kazancı üzerinden uygun niteliği bulur. Bilgi kazancı, ana düğüm ile alt düğümlerin entropileri arasındaki fark olarak tanımlanır.



Grafik 3.2.2. Karar ağacında bir seviyedeki sınama kriterini belirleme

- Yaş Grubu Niteliği kriter olarak kullanıldığında

  Ana düğümde Entropi :

  Entropi (YG) = (- (6/15) x $\log_2$(6/15)) + (- (9/15) x $\log_2$(9/15)) =0.971

  Alt Düğümlerde Entropi:

  Entropi($YG_{genç}$) = (-(3/5) x $\log_2$(3/5)) + (-(2/5) x $\log_2$(2/5))= 0.971

  Entropi(YG_{orta}) = (-(2/5) x $\log_2$(2/5)) + (-(3/5) x $\log_2$(3/5))= 0.971

  Entropi($YG_{yaşlı}$) = (-(1/5) x $\log_2$(1/5)) + (-(4/5) x $\log_2$(4/5))= 0.722

  Alt düğümlerdeki Entropiler Toplamı:

  Entropi(YG_{ALT}) = (5/15 x 0.971) + (5/15 x 0.971) + (5/15 x 0.722)

  = 0.888

  Elde Edilen Bilgi Kazancı :

  Bilgi Kazancı = Entropi(YG) – Entropi(YG_{ALT}) = 0.971 -0.888

  = 0.083

- Ev Sahibi Niteliği kriter olarak kullanıldığında

  Ana düğümde Entropi :

  $$\text{Entropi (YG)} = (- (6/15) \times \log_2(6/15)) + (- (9/15) \times \log_2(9/15)) = 0.971$$

  Alt Düğümlerde Entropi:

  $$\text{Entropi}(YG_{evet}) = (-(0/6) \times \log_2(0/6)) + (-(6/6) \times \log_2(6/6)) = 0$$

  $$\text{Entropi}(YG_{hayır}) = (-(6/9) \times \log_2(6/9)) + (-(3/9) \times \log_2(3/9)) = 0.918$$

  Alt düğümlerdeki Entropiler Toplamı:

  $$\text{Entropi}(YG_{ALT}) = (6/15 \times 0) + (9/15 \times 0.918) = 0.551$$

  Elde Edilen Bilgi Kazancı :

  $$\text{Bilgi Kazancı} = \text{Entropi}(YG) - \text{Entropi}(YG_{ALT}) = 0.971 - 0.551 = 0.42$$

Sırasıyla 0.083 ve 0.42 bilgi kazancı sağlayan "yaş grubu" ve "ev sahibi" kriterlerinden "ev sahibi" kriteri daha büyük bilgi kazancına sahip olduğu için bu düğümde "ev sahibi" niteliği sınama kriteri olarak kullanılır.

Gini İndisi: Nitelik seçiminde kullanılan bir diğer metot IBM InteligentMiner aracının kullandığı Gini indis hesaplamasıdır. Bir düğüme ait Gini indis değeri aşağıdaki şekilde hesaplanır.

$$\text{gini}(S) = 1 - \sum_{j=1}^{n} p_j^2 \qquad (7)$$

Burada n o düğümdeki sınıf sayısı, p_j ise sınıfın o düğümdeki olasılığıdır. Düğümde kullanılan niteliğe göre alt dalların gini indis değeri ise (8) ile hesaplanır.

$$\text{gini}_{\text{alt}}(S) = \sum_{j=1}^{v} [\frac{N_j}{N} \text{gini}(S_j)] \quad (8)$$

Bu formülde v alt düğüm sayısı, N_j alt düğümdeki örnek sayısı, S_j ise alt düğümlere ait gini indis değeridir. O düğümde kullanılabilecek bütün nitelikler için alt dalların gini indis değeri hesaplanarak en küçük gini indis değerine sahip olan nitelik o düğümdeki ayrıştırıcı olarak seçilir.

Ağaç Budama: Ağaç oluşturma basamağı, verileri tamamen aynı sınıf üyelerinden oluşan yapraklara bölünceye ya da karşılaştıracak nitelik kalmayıncaya kadar bölmeler. Bu algoritmanın sonucu olarak, çok derin ya da çok az deneme kümesi örneği içeren yaprak düğümlere sahip ağaçlar oluşabilir. Böyle bir ağacı öğrenme kümesi üzerinde test edince elbette ki doğruluğu çok yüksek sonuçlar verir. Ancak böyle bir model henüz görülmemiş örneklerle karşılaşırsa çok kötü doğruluklu sonuçlar üretebilir. Böyle bir model verimli değildir ve veriyi genellemekten uzaktır. Böyle bir modelin sahip olduğu bu özelliğe *aşırı uyum -overfitting-* denir. Aşırı uyum bir modelde istenmeyen bir sonuçtur.

Aşırı uyum genelde verideki gürültüden -hatalı sınıf değeri, yanlış değerli nitelikler- kaynaklandığı gibi problem alanının karmaşıklığından ya da rastgelelikten kaynaklanabilir.

Aşırı uyumu azaltmak için ağaçlarda budama işlemi uygulanır. Budama işlemi, bazı dalların ya da alt dalların kaldırılarak o dala ait nesnelerin yoğun olduğu sınıfla etiketlenmiş yaprak düğümlerle yerleştirilmesiyle gerçekleştirilir. Ağaç oluş-

turulurken erken-dur yöntemiyle erken-budama yapılabileceği gibi ağaç oluşturulduktan sonra budama *geç-budama* yapılabilir. Geç-budama yönteminin daha başarılı olduğu bilinmektedir. Zira erken-budama yöntemi hatalı sonuçlara yol açabilir, çünkü henüz dallanma yapılmamış bir dal budandığında, ağacın o noktadan sonra ne şekil almış olacağı o aşamada bilinmemektedir. Ancak geç-budama yapılırken ağaç zaten oluşmuş bulunmaktadır ve hangi dalların aslında gereksiz olduğu, aşırı uyum yarattığı bilinmektedir. Geç-budama yapılırken düğümlerdeki beklenen hata değerine bakılır. Eğer bir düğümdeki beklenen hata miktarı, o düğüme ait alt dallardaki beklenen hata miktarından küçük olursa alt dallar budanır.

3.2.3.3. İstatistiksel sınıflandırma

Bayes Teoremi: $B_1,B_2,...,B_k$, S örnek uzayına ait olaylar ise ve $\forall i \in \{1,2,\ldots,k\}$ için $P(B_i) \neq 0$ olmak üzere S örnek uzayındaki her A ($P(A) \neq 0$) olayı için aşağıdaki denklem geçerlidir [20]. (r=1, 2, … , k)

$$P(B_r \mid A) = \frac{P(B_r) \cdot P(A|B_r)}{\sum_{i=1}^{k} P(B_i) \cdot P(A|B_i)} \tag{9}$$

İstatistiksel sınıflandırma metodunda Bayes teoreminden yararlanılır. Eldeki veri kümesinden her sınıfa ait olasılık değeri niteliklere bağlı olarak bulunur. Oluşturulan bu olasılıklara göre bir nesnenin hangi sınıfa ait olduğu olasılıklı olarak hesaplanabilir. Bir örneğin sınıfı, o örneğe ait niteliklerin olasılık fonksiyonunu

maksimal değerine taşıyan sınıf olarak belirlenir. Bayes uygulamasında, niteliklerin birbirinden bağımsız değişken olması gerekir.

$A_1,A_2,...,A_{|A|}$ bir D veri kaynağındaki nesnelere ait nitelikler olsun. C ise bu nitelikler arasında bulunan |C| sayıda $c_1,c_2,..c_{|c|}$ şeklinde değerler taşıyan sınıf niteliği olsun. $a_1,a_2,..a_{|A|}$ değerlerini taşıyan bir d örneğinin ait olduğu c_j sınıfı için (10) değeri maksimum olur.

$$\Pr(C=c_j \mid A_1=a_1,A_2=a_2,\ldots,A_{|A|}=a_{|A|}) \tag{10}$$

c_j sınıfını tahminlerken bütün c sınıfları için $\Pr(C=c \mid A_1=a_1,A_2=a_2,\ldots,A_{|A|}=a_{|A|})$ olasılığı hesaplanır ve bu olasılıklardan maksimum olanının c_j değeri, d örneğinin sınıfını tahminler.

Bayes teoreminden faydalanarak $\Pr(C=c_j \mid A_1=a_1,A_2=a_2,\ldots,A_{|A|}=a_{|A|})$ olasılığı (11)'e dönüştürülebilir.

$$\frac{\Pr(A_1 = a_1,\cdots,A_{|A|} = a_{|A|} \mid C = c_j)\Pr(C = c_j)}{\sum_{k=1}^{|C|}\Pr(A_1 = a_1,\cdots,A_{|A|} = a_{|A|} \mid C = c_k)\Pr(C = c_k)} \tag{11}$$

Niteliklerin bağımsız olduklarını bildiğimizden,

$$\Pr(A_1 = a_1,\ldots,A_{|A|} = a_{|A|} \mid C = c_j) = \prod_{1}^{|A|}\Pr(A_i = a_i \mid C = c_j) \tag{12}$$

kullanılarak

$$\frac{\Pr(C = c_j)\prod_{i=1}^{|A|}\Pr(A_i = a_i \mid C = c_j)}{\sum_{k=1}^{|C|}\Pr(C = c_k)\prod_{i=1}^{|A|}\Pr(A_i = a_i \mid C = c_k)} \qquad (13)$$

ifadesine dönüştürülebilir.

Verilen d örneğinin sınıfını belirlerken (13)'deki ifadenin değerini, d örneğinin nitelik değerleri ile maksimize edecek j indisini (c sınıfını) bulmamız yeterlidir. İfadenin paydasının j indisinden bağımsız ve bütün sınıflar için o sınıfın oluşma olasılığı ile niteliklerin sınıflara bağlı koşullu olasılığı ile ilişkili olduğundan fonksiyonu maksimize edecek c_j sınıfını tahminlerken paydayı göz ardı edip, kesrin payını değerlendirmemiz yeterli olacaktır. Dolayısı ile bulunacak olan c_j sınıfı aşağıdaki gibi formüle edilir ve *Maximum A Posteriori* (*MAP*) tahminleyicisi olarak adlandırılır.

$$\arg_{c_j}\max[\Pr(C = c_j)\prod_{i=1}^{|A|}\Pr(A_i = a_i \mid C = c_j)] \qquad (14)$$

İstatistiksel sınıflandırma yapan veri madenciliği algoritmalarının çoğu Bayes Teoremine dayanmaktadır. Pekçok istatistiksel sınıflandırma algoritması ve versiyonları mevcuttur. NaiveBayes Algoritması ve Bayes Ağları, Bayes Teoremine dayanan istatiksel veri madenciliği algoritmalarının en yaygınlarındandır.

3.2.3.4. Regresyon

Karar ağaçları, normal olarak nominal nitelikli veriler üzerinde etkilidir. Elbette ki numerik değerler ile de çalışabilirler, ancak numerik değerler üzerinde daha verimli algoritmalar vardır ki doğası gereği numerik nitelikler üzerine kuruludur. Regresyon modeller, numerik değerli nitelikleri matematiksel bir fonksiyona uydurarak, sınıf tahminlemesinde bu modeli kullanırlar.

Doğrusal regresyon: Doğrusal regresyon, istatistik anlamda bir bağımsız değişkenin bağımlı değişkenlerin belirli ağırlıklarının doğrusal bileşimi şeklinde ifade edilmesidir (15). Bu model, veriye ait tüm nitelikler ve sınıf değeri numerik olduğunda uygulanabilecek bir modeldir. Ana fikri, sınıf niteliğini diğer niteliklerin belirlenecek ağırlık değerlerinin doğrusal bileşimi şeklinde formüle etmek ve tahminlemeleri bu model üzerinden yapmaktır.

$$y = v_0 a_0 + v_1 a_1 + \cdots + v_k a_k \qquad (15)$$

Doğrusal regresyon numerik tahminlemede oldukça iyi ve basit bir yöntem-dir ve onlarca yıldır istatistik uygulamalarda yaygın bir şekilde kullanılagelmektedir. Kötü yanı doğrusal olmasıdır. Normal hayattaki verilerin çoğu doğrusal değildir; her ne kadar hatalar karesi minimize edilerek en iyi doğrusal ilişki kurulmuş olsa da, ilişki, veriyi tam yansıtmakta başarısız kalabilir.

Lojistik Regresyon: Doğrusal regresyon numerik nitelikli tüm veriler üzerinde sınıflandırma yapmak için kullanılabilir. Aslında, veri dağılımı doğrusal

olsun olmasın sınıflandırma için regresyon yöntemi kullanılabilir. Bunun için, her sınıfa o sınıfın tüm nesnelerini 1, diğerlerini 0 değerine taşıyacak bir regresyon uygulanır. Böylece her sınıf için ayrı bir doğrusal ifade elde edilmiş olur. Sınıfı belli olmayan bir test nesnesi ile karşılaşıldığında tüm doğrusal formüller hesaplanır, hangi sınıfın doğrusal ifadesi daha büyük çıkarsa nesne o sınıfa tahminlenir. Bu yönteme *çokyanıtlı doğrusal regresyon – multiresponse linear regression*- denir. Özetle her sınıf için ayrı bir üyelik fonksiyonu oluşturarak, nesnelerin hangi sınıfa ait olduğu tahminlemesini bu fonksiyon değerlerini karşılaştırarak yapmaktır.

Pratikte çokyanıtlı doğrusal regresyon genelde iyi sonuçlar üretir. Ancak iki kötü yanı vardır. Birincisi, üretilen üyelik değerleri, 0-1 aralığı dışında, olasılık temeline aykırıdır. İkincisi, kullanılmış olan en küçük kareler metodu, hata terimlerinin istatistiksel olarak bağımsız olduğunu, aynı standart sapma değeri ile normal dağıldığını varsaymaktadır, ancak sınıflandırma problemlerindeki örnek nesnelerde bu gerçekleşmemektedir.

Bir diğer istatiksel model, Lojistik Regresyon, ise bu problemleri ortadan kaldırır. Üyelik fonksiyonunda 0-1 aralığına yönelmek yerine, eksi sonsuz ve artı sonsuz aralığına taşıyan bir yaklaşım benimser. Lojistik Regresyon modeli, her biri

$$\Pr\left[\,\middle|\, a_1, a_2, \cdots, a_k \right] \tag{16}$$

şeklinde olan üyelik fonksiyonunu

$$\frac{\log(\Pr[\,|\,a_1, a_2, \cdots, a_k}{(-\Pr[\,|\,a_1, a_2, \cdots, a_k} \tag{17}$$

şekline dönüştürür. Böylece fonksiyonun değeri [0-1] aralığı yerine eksi ve artı sonsuz aralığına taşınır.

Regresyon, aynı zamanda karar ağaçlarında ya da yapay sinir ağlarında da numerik nitelikler için yararlanılan bir yöntemdir. CART - Clasification and Regression Trees - sisteminde karar ağaçlarında numerik değerler için kullanıldığı gibi sonraki diğer algoritmalarda da kullanılmıştır.

3.2.3.5. Örnek tabanlı yöntemler

Günlük hayatımızda bir nesne gördüğümüzde bu nesnenin ne olduğunu belirlemek için belleğimizde yer edinen eski nesnelerle karşılaştırırız ve bu yeni nesne belleğimizdeki nesnelerden en çok hangisini andırıyorsa bu nesneyi de onunla benzer sınıfa atarız. Örnek tabanlı yöntemlerde, öğrenme kümesi ile edinmiş olduğumuz bilgiyle yeni nesnenin kimi hatırlattığı ya da kime benzediği sorusuyla yeni nesneyi tahminleriz.

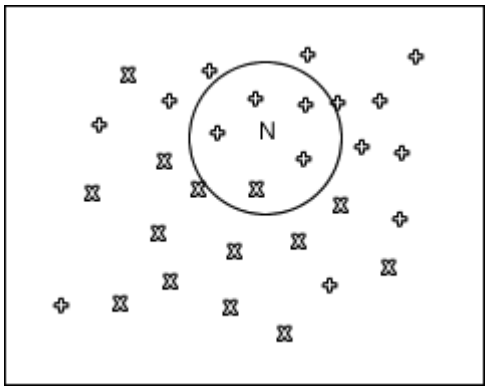
Örnek tabanlı yöntemlerin diğer yöntemlerden en büyük farkı bütün işlemleri tahminleme aşamasına bırakmış olmasıdır. Diğer yöntemlerde öğrenme aşamasında yapılan tüm işlemler, bu yöntemde yeni nesneyle karşılaşma anına bırakılmıştır. Bu nedenle bu yöntemlere *tembel* - *lazy* - yöntemler de denir. Yeni bir nesneyle karşılaşıldığında bu nesnenin nitelikleri ile eldeki nesnelerin nitelikleri arasındaki

benzerlik ya da farklılıklara göre yeni nesne en benzerin sınıfına atanır. Nesnelerin benzerliği, nitelikler arasında uzaklık metriği sayesinde matematiksel olarak hesaplanan değerlerle tesbit edilir. Genelde k sayıdaki benzer - yakın - nesnenin majoritesini oluşturduğu sınıf değeri kullanılır. Bu sebeple yöntem *k-en yakın komşuluk* adıyla anılır.

İki nesnenin uzaklığını hesaplamada kullanılan matematiksel yöntem -eğer nesneler sadece bir tane numerik nitelikten oluşuyorsa- gayet açıktır: iki nesnenin sahip olduğu numerik değerlerin farkı alınarak nesnelerin birbirlerinden uzaklığı bulunabilir. Eğer birden fazla sayıda numerik değer varsa bu durumda Euclid uzaklığı kullanarak nesneler arası uzaklık hesaplanabilir. Ancak nominal nitelikler mevcutsa bu nitelikler arasında da bir uzaklık kavramı oluşturmak gerekir. Kırmızı, yeşil ve mavi gibi değerleri barındırabilen bir nitelik için uzaklık nasıl hesaplanabilir? Bunun için en bilinen yaklaşım nitelik değeri aynı ise 0 farklı ise 1 değeri vermektir. Bazı durumlarda 0 ve 1 değerleri yeterli olmayabilir. Kırmızı ve sarı renkleri arasındaki uzaklığın, beyaz ve siyah renkleri arasındaki uzaklıktan daha küçük olması beklenebilir. Bu tip nitelikler için kendine özgü uzaklık metriği düşünmek gerekir. Örneğin renkler arasındaki uzaklık için renk skalasındaki sıralama kullanılabilir.

Bazı durumlarda bazı niteliklerdeki uzaklık değerinin diğer niteliklerdeki uzaklık değerinden daha fazla önemli olduğu düşünülebilir. Böyle durumlarda her nitelik için önem derecesine karşılık gelen bir ağırlık ölçütü verilmektedir.

Şekil 3.2.1.'de '+' ve 'x' sınıflarına ait nesnelerin iki boyutlu XY düzleminde yerleşimleri gösterilmiştir. Yeni karşılaşılan N nesnesi k-en yakın komşuluk algoritmasına göre (k=5), en yakın 5 komşusunda + sınıfının ağırlıkta olmasından dolayı + sınıfı ile tahminlenir.



Şekil 3.2.1. k-en yakın komşuluk algoritmasına göre yeni nesnenin sınıflandırılması

Nesneler arası benzerlik ve farklılık: Nesneler arasındaki benzerlik ya da farklılık değeri matematiksel olarak Öklit uzaklığı, Minkowski uzaklığı, Manhattan uzaklığı kullanılarak ölçülendirilir. Her nesne arasındaki uzaklık n x p boyutunda (n: nesne sayısı; p: nitelik sayısı) matrisle ifade edilir. Bu matrise *Uzaklık Matrisi* denir.

Bir veri kümesinin matris formda ifadesi (18)'de; (19)'da da bir uzaklık matrisi verilmiştir. Veri kümesine ait matris formunda her satır bir nesneyi, sütun ise nesnelere ait nitelik değerlerini taşır. Uzaklık matrisinde ise d_{ij}, (i,j | $i \in N$ ve $i<n$; $j \in N$ ve $j<n$, $i \neq j$ olmak üzere) i ve j indisine sahip nesneler arasındaki uzaklığı ifade eder.

$$\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} \tag{18}$$

$$\begin{bmatrix} 0 & . & . & . & . \\ d(2,1) & 0 & . & . & . \\ d(3,1) & d(3,2) & 0 & . & . \\ \vdots & \vdots & \vdots & \cdots & . \\ d(n,1) & d(n,2) & d(n,3) & \cdots & 0 \end{bmatrix} \tag{19}$$

Uzaklık matrisinde i ve j indisine sahip nesneler arasındaki uzaklık d_{ij} değerinin hesaplanmasında öklid uzaklığı, minkowski uzaklığı, manhattan uzaklığı gibi uzaklık hesaplama yöntemleri kullanılabilir. (20)'de Öklid uzaklığı ile; (21)'de Minkowski uzaklığı ile d_{ij} uzaklıklarının hesaplanması ifade edilmiştir. (21)'de q tamsayısı değerinin 1 alınması durumunda Manhattan uzaklığı elde edilmiş olur.

$$d(i, j) = \sqrt{\left|X_{i,1} - \text{ }_{j,1}\right|^2 + \text{ }_{i,2} - \text{ }_{j,2}\right|^2 + \cdots + \text{ }_{i,p} - \text{ }_{j,p}\right|^2} \tag{20}$$

q pozitif bir tamsayı olmak üzere

$$d(i, j) = \sqrt{\left|X_{i,1} - \text{ }_{j,1}\right|^q + \text{ }_{i,2} - \text{ }_{j,2}\right|^q + \cdots + \text{ }_{i,p} - \text{ }_{j,p}\right|^q} \tag{21}$$

3.2.3.6. Demetleme

Veri kümesi birbirine benzeyen nesnelerden oluşan demetlere bölünür. Aynı demetteki veriler birbirine daha çok benzerken farklı demetlerdeki nesneler birbirine daha az benzer. Verinin demetlere ayrılmış olması kullanıcıya veri hakkında genel

bilgiler sunar, kullanıcının veri dağılımını anlamasını sağlar. Sınıf sayısı ve hangi nesnenin hangi sınıfta olduğu belli değildir. Bu sebeple *gözetimsiz öğrenme* - *unsupervised learning* - diye adlandırılır. Demetleme yöntemleri olarak bölünmeli yöntemler, hiyerarşik yöntemler, yoğunluk tabanlı yöntemler, model tabanlı yöntemler sayılabilir.

Bölünmeli Yöntemler: Demetleme algoritmalarının en bilineni "k-orta" - k-means - algoritmasıdır. Veri kümesinin kaç tane demete bölüneceği girilen parametre ile belirlenir. Bu k parametresidir. Verilen bu parametreye göre k tane orta nokta - mean point - algoritma tarafından rastgele belirlenir. Veri kümesindeki tüm nesneler Euclid uzaklığı kullanılarak en yakın orta noktanın bulunduğu demete atanır. Oluşturulan bu demetlerin orta noktaları demete eklenmiş olan nesneler göze alınarak yeniden hesaplanır. Oluşturulmuş olan bu yeni orta noktalara göre algoritma en baştan itibaren yeniden çalıştırılarak yeni demetler oluşturulur ve oluşan bu demetlerin orta noktaları tekrar hesaplanarak yeni orta noktalar elde edilir. Bu yineleme işlemi artık orta noktalar değişmeyecek şekilde sürdürülür.

Algoritmada demetler oluşturulurken kullanılan Euclid Uzaklığı yerine yalın uyum katsayısı, jaccard katsayısı, jaccard uzaklığı, cosine benzerliği gibi farklı yöntemler de kullanılabilir.

k-orta algoritmasının çok çeşitli varyasyonları geliştirilmiştir. Bunlardan hiyerarşik k-orta demetleme yöntemi k parametresini 2 değeri ile başlatarak tüm veri kümesini 2 demete böler. Sonrasında her demeti kendi içinde yinelemeli olarak

demetleyerek istenilen demet sayısına ulaşana kadar demetleme işlemine devam eder. Bölünmeli yöntem yaklaşımını kullanan bir diğer algoritma da k-medoids demetleme algoritmasıdır.

Diğer Demetleme Algoritmaları: Bilinen diğer demetleme algoritmaları hiyerarşik demetleme yöntemlerinden AGNES ve DIANA; yoğunluk tabanlı yöntemlerden DBSCAN, OPTICS, DENCLUE, CLIQUE; model tabanlı yöntemlerden EM (Expectation Maximization) algoritmalarıdır.

3.2.4. Model Başarımını Denetleme

Model başarımını değerlendirirken kullanılan temel kavramlar hata oranı, kesinlik, duyarlılık ve F-ölçütüdür. Modelin başarısı, doğru sınıfa atanan örnek sayısı ve yanlış sınıfa atılan örnek sayısı nicelikleriyle ilgilidir.

Modelin başarı testinin, öğrenme veri kümesinin dışında bir veri kümesi ile yapılması gerekir. Bu kümeye test kümesi denir. Bu sebeple eldeki veri kümesi, öğrenme veri kümesi ve test veri kümesi olarak ayrılır. Öğrenme veri kümesi üzerinde model oluşturulduktan sonra oluşan model test veri kümesinde sınanır.

Test sonucunda ulaşılan sonuçların başarılılık bilgileri karışıklık matrisi ile ifade edilebilir. Karışıklık matrisinde satırlar test kümesindeki örneklere ait gerçek sayıları, kolonlar ise modelin tahminlemesini ifade eder. Tablo 3.2.1'de iki sınıflı bir veri kümesinde oluşturulmuş bir modelin karışıklık matrisi verilmiştir. Sınıf sayısı

ikiden fazla olduğunda 2x2 boyutundaki bu matris, n sınıf sayısı olmak üzere, n x n boyutlarında genişletilmiş bir matris şeklini alacaktır. TP ve TN değerleri doğru sınıflandırılmış örnek sayısıdır. False Pozitif (FP), aslında 0 (negatif) sınıfındayken 1 (pozitif) olarak tahminlenmiş örneklerin sayısıdır. False Negative (FN) ise 1 (pozitif) sınıfındayken 0 (negatif) olarak tahminlenmiş örneklerin sayısını ifade eder. Genel olarak n x n boyutlarındaki bir karışıklık matrisinde ana köşegen doğru tahminlenmiş örnek sayılarını; ana köşegen dışında kalan matris elemanları ise hatalı sonuçları ifade etmektedir.

		Öngörülen Sınıf	
		Sınıf =1	Sınıf=0
Doğru Sınıf	Sınıf=1	a	b
	Sınıf=0	c	d

a: TP (True Pozitif) örnek sayısı
b: FN (False Negatif) örnek sayısı
c: FP (False Pozitif) örnek sayısı
d: TN (True Negatif) örnek sayısı

Tablo 3.2.1 İki sınıflı bir veri kümesinde oluşturulmuş modelin karışıklık matrisi

3.2.4.1. Doğruluk – Hata oranı

Model başarımının ölçülmesinde kullanılan en popüler, basit ve belirleyici ölçüt, modele ait doğruluk oranıdır. Doğru sınıflandırılmış örnek sayısının (TP +TN), toplam örnek sayısına (TP+TN+FP+FN) oranıdır (22).

$$\text{Doğruluk} = \frac{TP + TN}{TP + FP + FN + TN} \quad (22)$$

Hata oranı ise bu değerin birimsel tamlayanıdır. Diğer bir ifadeyle yanlış sınıflandırılmış örnek sayısının (FP+FN), toplam örnek sayısına (TP+TN+FP+FN) oranıdır (23).

$$\text{Hata Oranı} = \frac{FP + FN}{TP + TN + FP + FN} \tag{23}$$

3.2.4.2. Kesinlik

Kesinlik, sınıfı 1 olarak tahminlenmiş True Pozitif örnek sayısının, sınıfı 1 olarak tahminlenmiş tüm örnek sayısına oranıdır:

$$\text{Kesinlik} = \frac{TP}{TP + FP} \tag{24}$$

3.2.4.3. Duyarlılık

Doğru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır (25).

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \tag{25}$$

Duyarlılık, kesinlik ölçütü ile ters orantılıdır.

3.2.4.4. F-Ölçütü

Kesinlik ve duyarlılık ölçütleri tek başına anlamlı bir karşılaştırma sonucu çıkarmamıza yeterli değildir. Her iki ölçütü beraber değerlendirmek daha doğru sonuçlar verir. Bunun için F-ölçütü tanımlanmıştır. F-ölçütü, kesinlik ve duyarlılığın harmonik ortalamasıdır:

$$\text{F - ölçütü} = \frac{2 \times \text{Duyarlilik} \times \text{Kesinlik}}{\text{Duyarlilik} + \text{Kesinlik}} \tag{26}$$

4. MATERYAL VE METOT

4.1. UYGULAMADA KULLANILAN VERİ MADENCİLİĞİ ARACI

Veri madenciliği algoritmalarının karşılaştırılıdığı bu tez çalışmasında Waikato Üniversitesinde java programlama diliyle geliştirilmiş olan Weka (Waikato Environment for Knowledge Analysis) programı kullanılmıştır. Weka, kullanımı ücretsiz, açık kaynak kodlu, içerisinde pek çok sınıflandırma, regresyon, demetleme, bağıntı kuralları, yapay sinir ağları algoritmaları ve önişleme metotları barındıran, yaygın kullanımlı bir veri madenciliği aracıdır.

WEKA, ham verinin işlenmesi, öğrenme metotlarının veri üzerinde istatistiksel olarak değerlendirilmesi, ham verinin ve ham veriden öğrenilerek çıkarılan modelin görsel olarak izlenmesi gibi veri madenciliğinin tüm basamaklarını destekler. Geniş bir öğrenme algoritmaları yelpazesine sahip olduğu gibi pek çok veri önişleme süzgeçleri içerir. 4 temel uygulamayı barındırır, bunlar:

- Explorer
- Experimenter
- Knowledge Flow
- Simple CLI

Explorer, çeşitli veri madenciliği algoritmalarının uygulanabileceği, veri önişlemenin yapılabileceği, kullanımı kolay bir arayüzdür. Knowledge Flow,

grafiksel olarak taşı-bırak metodu ile çeşitli veri işleme metotları ikonlarının ekrana taşınarak bir veri akış diyagramı ile veri madenciliği uygulamalarının yapılmasına imkan veren arayüzdür. Experimenter arayüzü, eldeki probleme hangi metodun, hangi parametrelerle uygulanarak daha iyi bir sonuç alınabileceğini analiz etmeye yarayan arayüzdür. Simple CLI uygulaması ise metin tabanlı bir konsolden komut satırlarının girilerek veri madenciliğinin uygulanabilmesine imkan vermektedir.

Desteklediği temel veri kaynakları; metin tabanlı arff, csv, c45, libsvm, svmlight, Xarff formatlarıdır. jdbc sürücüsü bulunan veritabanlarına direk bağlantı yapabilir ve internet üzerinden http protokolünü kullanarak bu formatlara uygun dosyaları okuyabilme yeteneğine sahiptir.

4.2. UYGULAMADA KULLANILAN VERİ KAYNAĞI

Algoritmaların karşılaştırması SEER (Surveillance Epidemiology and End Results) veri kaynağı kullanılarak yapılmıştır.

SEER veri tabanı farklı kanser gruplarını içeren ve bilimsel araştırmalarda son derece önemli bir yer tutan, güvenilir, dokümante edilmiş, eşine az rastlanır bir veri kaynağıdır. National Cancer Institute (NCI)'in sağladığı Amerika Birleşik Devletleri'nin belli başlı coğrafi bölgelerini kapsayan, nüfusunun %26'sını ilgilendiren ve bu kanser vakaları hakkında istatistiksel önem taşıyan bilgiler içerir. Yıllık olarak güncellenen bu veritabanı bilimsel çalışma yapanlara, sağlık sektöründe çalışanlara, halk sağlığı konusunda görevli kurumlara açık bir veri kaynağı olup,

binlerce bilimsel çalışmada kaynak olarak kullanılmıştır. Veri kaynağı, kurumun web sitesinden veri kullanma talep formu doldurularak imzalandıktan sonra elektronik olarak indirilebilmektedir.

1973 yılı itibarı ile başlanmış olan kanser verileri farklı yılları kapsayan, farklı tümör tiplerine göre gruplar altında metin formatında, 118 nitelikten oluşan oldukça büyük veri kaynağıdır. Bazı nitelikler daha önceki yıl verilerinde yokken sonraki verilerde eklenmiş, bazı niteliklerin sonraki yıllarda değerleri alınmamış ve bazı nitelikler farklı bir tümör tipinde değer taşırken bazı tiplerde bir anlam ifade etmediği için değer kullanımdan kaldırılmıştır. Her ne kadar bu veri kaynağı oldukça düzenli ve dokümante edilmiş olsa da yaptığım çalışma için bir önişlemden geçirilmesi gerekmiştir.

Bu tez çalışmasında yıllık olarak güncellenen SEER veri kaynağının 2008 yılına ait olan versiyonu kullanılmıştır.

NCI kurumunun sağlamış olduğu bu veri kaynağı incidence ve populations adlı iki klasör ve bir readme dosyasından oluşmaktadır. “Readme.txt” metin belgesi, bu klasörlerde var olan dosyalar hakkında detaylı bilgiler içermektedir.

Incidence klasörü içerisinde mevcut olan dosyalar şunlardır:

- *“seerdic.pdf” dosyası*: SEER veri sözlüğü olup veri dosyalarında bulunan kayıtların niteliklerini ve detaylı açıklamalarını kapsamaktadır.

- *"yr1973_2006.seer9" klasörü*: 1973-2006 yıllarına ait Atlanta, Connecticut, Detroit, Hawaii, Iowa, New Mexico, San Francisco-Oakland, Seattle-Puget Sound ve Utah verilerini kapsamaktadır; bu bölgelerden The Seattle-Puget Sound ve Atlanta bölgeleri sırasıyla 1973, 1974'ten itibaren programa katılmıştır.

- *"yr1992_2006.sj_la_rg_ak" klasörü*: 1992-2006 yıllarına ait San Jose-Monterey, Los Angeles, Rural Georgia ve Alaska verilerini kapsamaktadır.

- *"yr2000_2006.ca_ky_lo_nj" klasörü*: 2000-2006 yıllarına ait Greater California, Kentucky, Louisiana ve New Jersey verilerini kapsamaktadır. Louisiana bölgesi 2005 yılına ait veriler sadece Ocak-Haziran aylarında yapılan teşhis verilerini içerir. Temmuz-Aralık 2005 tarihleri arasında Katrina Kasırgasının Louisiana halkı üzerinde uzun süren yıkıcı etkileri nedeniyle 2005 yılının son altı aylık verileri bu klasörde yer almamaktadır.

- *"yr2005.lo_2nd_half" klasörü*: 2005 yılı Temmuz-Aralık aralığına ait Louisiana bölgesi verileri bulundurmaktadır.

Veri kaynağı dosyaları, yukarıda belirtilen klasörler içerisinde hastalara ilişkin bilgileri içeren kayıtlardan oluşmaktadır. Bu kayıtlar 118 nitelikten oluşmaktadır. Bu niteliklerin ayrıntılı tanımı seerdic.pdf dosyasında bulunmaktadır.

4.2.1. SEER Veri Kaynağının Nitelikleri

Bir veri kaynağı üzerinde çalışma yapabilmek için her şeyden önce veri kaynağındaki bilgilerin anlamını, alabileceği değer ya da değer aralıklarını, formatını bilmek gerekir. Uygulama alanlarının çok geniş ve farklı disiplinlerden olabileceği düşünülürse, matematikçi, istatistikçi ya da bilişim uzmanlarının konu hakkında iyi derecede bilgili olması beklenemez. Bu açık, uygulamayı yapan kişinin konuya hakim kişilerle görüşme yapma ya da varsa konuyla ilgili belgeleri araştırarak veri kaynağındaki verilerle ilgili bilgi edinme yoluyla kapatılabilmektedir.

SEER veri kaynağıyla ilgili belgeler kaynak dosyalarıyla beraber araştırmacılara verilmektedir. Ayrıca bu belgelere internet üzerinden de erişilebilmektedir. Yapmış olduğum çalışma öncesinde kaynak dosyalarla birlikte gelmiş olan bu belgeler incelenmiş ve veri yapısına ilişkin çalışılmıştır. Aşağıda, bu çalışmaya ait, veri dosyasındaki bilgilerin açıklamaları verilmektedir.

1- ***Hasta Kayıt Numarası:*** Hasta bazında SEER programının her hastaya verdiği özgün bir numaradır. Hastada yeni bir kanser bulgusu rastlanırsa aynı kayıt numarası ile listelenir.

2- ***Kayıt Kodu:*** Programa kayıtlı bulunan her coğrafik bölgeye özgün kayıt kodudur.

Kayıt Kodları	
0000001501	San Francisco-Oakland SMSA (1973)
0000001502	Connecticut (1973)
0000001520	Metropolitan Detroit (1973)
0000001521	Hawaii (1973)
0000001522	Iowa (1973)
0000001523	New Mexico (1973)
0000001525	Seattle (Puget Sound) (1974)
0000001526	Utah (1973)
0000001527	Metropolitan Atlanta (1975)
0000001529	Alaska
0000001531	San Jose-Monterey
0000001535	Los Angeles
0000001537	Rural Georgia
0000001541	Greater California (excluding SF, Los Angeles & SJ)
0000001542	Kentucky
0000001543	Louisiana
0000001544	New Jersey

3- ***Medeni Durum (Teşhis Zamanında):*** Hastanın teşhis tarihindeki medeni hali bilgisidir.

Kodlar	
1	Bekar (hiç Evlenmemiş)
2	Evli

3	Ayrı Yaşamakta
4	Boşanmış
5	Dul
9	Bilinmiyor

4- ***Uyruğu:*** Kişinin etnisitesini ifade eden kod.

5- ***İspanyol Menşeylilik:*** Bu nitelik hastanın soyisminin ispanyolca olması durumunda hangi ülkeden olduğunu ifade eder. Meksikalı, Porto Rikolu, Kübalı, Güney ya da Orta Amerikalı, Avrupalı, İspanyol, Dominic Cumhuriyeti menşeyli, bilinmeyen gibi değerlere karşılık gelen kodları alır.

6- ***İspanyol Menşeylilik (Otomatik değerlendirme***): NHIA algoritması ile İspanyol menşeyliliğin tayin edildiği kodlama.

7- ***Cinsiyet:*** Hastanın Cinsiyeti.

Kod	
1	Erkek
2	Bayan

8- ***Hastanın Yaşı:*** Teşhis tarihinde hastanın yıl cinsinden yaşı. Bilinmeyen yaşlar için 999 değeri kullanılmıştır.

9- ***Doğum Yılı:*** YYYY formatında hastanın doğum yılı. Bilinmeyen doğum tarihleri için 9999 değeri kullanılmıştır.

10- ***Doğum Yeri:*** Hastanın doğum yeri kodudur. 3 karakterli numerik kodlardan oluşur.

11- ***Sıra Numarası-Merkezi (Sequence Number–Central):*** Teşhis edilen tümörün aynı hastadaki kaçıncı tümör olduğunu ifade eden niteliktir. Kötü huylu (malign) ve in-situ kanserlerde ilk teşhiste 00 değerini alır ki hastada sadece bir malign tümör teşhis edildiğini ifade eder. İkinci bir malign tümör teşhisinde ilk kayıttaki değer 01 değerini, yeni yapılan teşhis 02 değerini alır. Maksimum 35 değerini alabilir. Kaçıncı tümör olduğu bilinmeyenler için 99 değerini alır. Non-malign (iyi huylu) tümörler ise 60 değeri ile başlar. Maksimum 87 değerini alır. Kaçıncı olduğu bilinmiyorsa 88 değerini alır. Servix Karsinoma hastaları için ise bu değer 89 değerini taşır.

12- ***Teşhis Tarihi (Ay):*** Hastalığın teşhisinin yapıldığı tarihin ay bilgisidir. 01-12 arasında değerler alır.

13- ***Teşhis tarihi (Yıl):*** Hastalığın teşhisinin yapıldığı tarihin yıl bilgisidir. 1973-YYYY arasında değerler alır.

14- ***Primary Site:*** Tümörün ortaya çıktığı orijin bölgesini ifade eden kodlamadır. Nitelik 1977 öncesinde "Manual of Tumor Nomenclature and Coding", 1968 (MOTNAC)'a göre, 1977 sonrası ise "International Classification of Diseases for Oncology", 1976 Edition (ICD-O-1976)'a göre kodlanmıştır. 1973-1991 yılları arasındaki değerler sonradan makine-çevirmeyle ICD-O-2 kodlarına çevrilmiş olup tam bir elle kontrolü yapılmamıştır.

15- ***Laterality:*** Parçalı organlarda tümörün hangi parçada olduğunu ya da vücudun hangi kısmında olduğunu ifade eden kodlamadır.

16- ***Histoloji:*** Tümörün morfolojik özelliğini belirten histolojik kodudur. 1977'den itibaren genellikle mikroskobik olarak teşhisi yapılmış vakalar için girilmiştir. Program boyunca farklı yıllarda farklı kodlama metotları kullanıldığı için bu bilginin özellikle morfolojik araştırmalarda kullanılmasında dikkat edilmesi gerekir.

17- ***Davranış Kodu ICD-O-2 (Behavior):*** Tümörün malign, benign, karsinoma in situ, invazif olması durumuna göre kodlanmıştır. 1992 tarihinden itibaren kullanılmıştır. 2000 yılından sonra geçerli olan ICD-O-3 kodlaması ICD-O-2 kodlamasına da çevrilmiştir.

18- ***Histoloji Tipi ICD-O-3:*** Tümör hücresinin mikroskobik muhteviyatını belirtir. Histoloji tipi tümörün sınıflandırılmasında ve tedavi seçeneklerinin belirlenmesinde temel unsurdur. Tedavi ve hastalığın gidişatında önemli bir

etkendir. 2001 yılı itibarı ile ICD-O-3 formatında kodlanmıştır. 1973-2000 yılları arasındaki ICD-O-2 kodlamaları ICD-O-3 kodlamasına çevrilmiştir.

19- ***Davranış Kodu ICD-O-3 (Behavior):*** Davranış kodunun ICD-O-3 formatındaki kodlamasıdır. 1973-2000 yılları arasındaki tüm ICD-O-2 kodlamaları bu formata çevrilmiştir.

20- ***Teşhis Doğrulaması (Diagnostic Confirmation):*** Tümör hücresinin teşhisinde kullanılmış olan en iyi metodu belirtir. Sadece teşhis anındaki metodu değil hastalık boyunca kullanılmış olan teşhis metotlarından en iyi olanı ifade eder.

Mikroskobik Doğrulama	
1	Pozitif Histoloji
2	Pozitif Sitoloji
4	Pozitif Mikroskobik doğrulama, (Metot belirtilmemiş.)
Mikroskobik Olmayan Doğrulama	
5	Pozitif Labaratuvar test/marker uygulaması
6	Mikroskobik doğrulama olmadan direk görselleme
7	Mikroskobik doğrulama olmadan Radyoloji ve diğer görselleme metodu
8	Sadece Klinik Teşhis (5,6,7 olmayan)
Bilinmeyen Doğrulama	
9	Mikroskobik doğrulama olup olmadığı bilinmeyen (ölüm sertifikası)

21- ***EOD – Tümör Büyüklüğü:*** Esas tümör hücresinin en uzun olduğu kesitinin milimetre cinsinden uzunluğu (999 bilinmeyen büyüklükler için kullanılmıştır). 1988-2003 yılları arasındaki vakalar için geçerlidir.

22- ***EOD-Yayılım:*** Tümörün kesintisiz ya da uzak metastaz şeklinde teşhis edilmiş olan yayılımının olup olmadığı ve hangi organa yayıldığı bilgisi. 1988-2003 yıllarındaki kayıtlar için geçerlidir.

23- ***EOD-Lenf Nodu yayılımı:*** Tümörün kendini gösterdiği lenf nod zincir sayısıdır. 1998-2003 yılları için geçerlidir.

24- ***Bölgesel Lenf Nodları Yayılımı:*** Patalojist tarafından belirlenmiş metastaza uğramış bölgesel lenf nodu sayısıdır.

25- ***Tümor Markerları 1,2,3:*** Göğüs, prostat ve testis kanserlerine ait tümör markerları.

26- ***CS-Tümor Büyüklüğü:*** Esas tümör hücresinin en uzun olduğu kesitinin milimetre cinsinden uzunluğu (999 bilinmeyen büyüklükler için kullanılmıştır). 2004 yılı sonrası vakalar için geçerlidir.

27- ***CS-Yayılım:*** Tümörün teşhis edilmiş olan kesintisiz yayılımının olup olmadığı ve hangi organa yayıldığı bilgisi. 2004 yılı sonrası vakalar için geçerlidir.

28- ***CS-Lenf Nodu yayılımı:*** Tümörün lokal olarak kendini gösterdiği lenf nodu zincir sayısıdır. 2004 yılı sonrası vakalar için geçerlidir.

29- ***CS METS AT DX:*** Teşhis anında tümörün yayılmış olduğu diğer bölgeler. 2004 yılı sonrası vakalar için geçerlidir.

30- ***Derived AJCC T:*** CS kod değerlerinden algoritmik yollarla elde edilmiş AJCCT değeri. 2004 yılı sonrası vakalar için geçerlidir.

31- ***Derived AJCC N:*** CS kod değerlerinden algoritmik yollarla elde edilmiş AJCCN değeri. 2004 yılı sonrası vakalar için geçerlidir.

32- ***Derived AJCC M:*** CS kod değerlerinden algoritmik yollarla elde edilmiş AJCCM değeri. 2004 yılı sonrası vakalar için geçerlidir.

33- ***Derived AJCC STAGE GROUP:*** CS kod değerlerinden algoritmik yollarla elde edilmiş AJCC aşaması. 2004 yılı sonrası vakalar için geçerlidir.

34- ***Derived SS1977:*** CS kod değerlerinden algoritmik yollarla elde edilmiş SS1977 – SEER Summary Stage 1977" kodlaması. 2004 yılı sonrası vakalar için geçerlidir.

35- ***Derived SS2000:*** CS kod değerlerinden algoritmik yollarla elde edilmiş SS2000 – SEER Summary Stage 1977" kodlaması. 2004 yılı sonrası vakalar için geçerlidir.

36- ***RX Summ - Surgery of Primary Site:*** Tümörün esas görüldüğü bölgede herhangi bir operasyonda bulunulup bulunulmadığı, yapıldıysa yapılan müdahaleyle dokunun alınıp alınmadığı.

"RX Summ - Surgery of Primary Site" Kodları	
00	Uygulanmadı; Tümörlü esas bölgede herhangi bir operasyon yapılmadı; Sadece Otopsi
10-19	Tümör alındı; Patalojik örnek yok ya da olup olmadığı bilinmiyor
20-80	Operasyonel resection; Patalojik örnekli
90	Opere edilmiş, Bölgeye operatif işlem uygulanmış ama ameliyat hakkında bilgi yok.
98	Hematopeietik, reticuloendothelial, immunoproliferative, myeloproliferative hastalık
99	Operasyon olup olmadığı bilinmiyor

37- ***RX Summ – Scope of Regional Lymph Node Surgery:*** Bölgesel lenf nodlarının alınması, biyopsisi ya da aspirasyonu yönteminin tanımı.

"RX Summ – Scope of Regional Lymph Node Surgery" Kodları	
0	Bölgesel Lenf Nodları alınmamış;
1	Biyopsi ya da aspirasyon uygulanmış
2	Sadece sentinel lenf nodu biyopsisi
3	Bölgesel lenf nodları alınmış ama alınan Lenf nodu sayısı bilinmiyor, ya da belirtilmemiş,

4	1-3 lenf nodu alınmış
5	4 ya da daha fazla lenf nodu alınmış
6	Sentinel lenf nodu biyopsisi ve 3,4,5 kodları aynı anda geçerli
7	Sentinel lenf nodu biyopsisi ve 3,4,5 kodları farklı zamanlarda uygulanmış
9	Bilgi yok ya da kodlama uygun değil

38- ***Surgical procedure of Other Site:*** Esas tümör bölgesinden uzaktaki lenf nodlarına, dokulara ya da organlara operasyonla müdahale

"Surgical procedure of Other Site" Kodları	
0	Uygulanmamış; sadece otopsi esnasında
1	Uygulanmış
2	Başka bölgelerde uygulanmış
3	Uzak Lenf nodlarında uygulanmış
4	Uzak tümörlerde uygulanmış
5	2,3,4 kodlarının hepsi geçerli
9	Bilgi yok

39- ***Reason For No Surgery:*** Operasyon yapılmama sebebi. Sebep önemli olmadığı için aşağıdaki tabloda kodların gruplanarak açıklamaları verilmiştir.

"Reason For No Surgery" Kodları	
0	Operasyon yapılmış
1,2,3,4,5,6,7	Operasyon yapılmamış
8,9	Bilinmiyor

40- ***RX Summ – Radiation:*** Uygulanan radyoterapi yöntemi

"RX Summ – Radiation" Kodları	
0	Uygulanmamış
1	Beam (ışın) Rayoterapisi
2	Radyoaktif implant
3	Radioisotop
4	1,2,3 kodlarındaki uygulamaların bir kombinasyonu
5	Radyoterapi – metot belirtilmemiş
6	Diğer Radyoterapi yöntemleri (1973-1987 arası)
7	Hasta ya da hasta yakını radyoterapi tedavisini reddetti
8	Radyoterapi önerildi, yapılıp yapılmadığı bilinmiyor
9	Radyoterapinin yapılıp yapılmadığı bilinmiyor

41- ***RX Summ – Surgery/Radiation Therapy Sequence:*** Radyoterapi ve Operasyon uygulanan hastalarda uygulamanın hangi sırada uygulandığı

"RX Summ – Surgery/Radiation Therapy Sequence" Kodları	
0	Radyoterapi ve Operasyon birlikte uygulanmamış
2	Önce radyoterapi sonra operasyon
3	Önce operasyon sonra radyoterapi
4	Ameliyat öncesi ve sonrasında radyoterapi
5	Intraoperatif radyoterapi tedavisi
6	Ameliyat öncesinde ve sonrasında diğer radyoterapi uygulamaları ile intraoperatif radyoterapi
9	Ameliyat ve radyoterapi uygulanmış ancak sıralama bilinmiyor

42- ***Age Recode:*** 5 yıllık aralıklarla hasta yaşı grupları

Yaş Grup Kodları	
00	0 Yaş
01	01-04 Yaşlar
02	05-09 Yaşlar
03	10-14 Yaşlar
04	15-19 Yaşlar
05	20-24 Yaşlar
06	25-29 Yaşlar
07	30-34 Yaşlar
08	35-39 Yaşlar
09	40-44 Yaşlar
10	45-49 Yaşlar
11	50-54 Yaşlar
12	55-59 Yaşlar
13	60-64 Yaşlar
14	65-69 Yaşlar
15	70-74 Yaşlar
16	75-79 Yaşlar
17	80-84 Yaşlar
18	85+ Yaşlar
99	Bilinmiyor

43- ***Behavior Recode for Analysis:*** 2001 yılından itibaren ICD-O-3 kodlaması gerektiren histoloji ve behavior kodlamaları -ki 2001 öncesinde ICD-O-3 kodlaması kullanılıyordu- bu nitelikle yeniden kodlanmıştır.

Behavior Recode for Analysis	
0	Benign
1	Borderline malignite
2	In Situ
3	Malign
4	ICD-O-3 kodlamasına göre malign
5	ICD-O-3 kodlamasında geçerliliği yok

44- ***Histoloji Recode – Broad Groupings:*** Histoloji kodları

45- ***Histoloji Recode – Brain Groupings:*** Beyin tümörü histoloji kodları

46- ***Race Recode:*** Hasta ırkının tekrar kodlaması

"Race Recode" Kodları	
1	Beyaz
2	Siyah
3	Diğer (Amerika yerlisi, Asyalı)
7	Diğer-belirtilmemiş
9	Bilgi yok

47- ***Actual Number of Primaries:*** SEER verisinde kaydı bulunan her hastanın sahip olduğu esas tümör sayısı. Bu değer her hastanın farklı kayıtlarında aynı değeri taşır.

48- ***Yaşama Süresi (Survival Time Recode):*** Bu nitelik değeri hesaplanırken teşhis tarihine ek olarak şu alanlar göze alınmıştır; ölüm tarihi, hastanın hayatta olduğu bilindiği son tarih ya da takip evresinde en son muayeneye geldiği tarih. YYMM formatındadır. Örneğin Mayıs 1970'te kanser teşhisi konulmuş ve Mayıs 1980'de hayatını kaybetmiş bir hasta için bu niteliğin değeri 4 yıl ve 0 ay anlamına gelen 0400 şeklinde olur.

49- ***Cause of Death Recode to SEER Site Recode:*** Ölüm Sebebi kodlamalarıdır. Özetle, 38000 kodu ve 5 ile başlayan kodlar kanser dışı ölüm sebeplerini; 00000 kodu hastanın hayatta olduğunu; 41000 kodu tanımlı olmayan bir ölüm sebebini; 99999 ise sisteme girilmemiş bilgiyi ifade eder. Bu kodların dışındaki 2 ve 3 rakamları ile başlayan kodlar ise kanser sebepli ölümlerdir.

50- ***Vital Status Recode:*** Hastanın hayatta mı yoksa değil mi olduğunu belirtir.

"Vital Status Recode"	
1	Hayatta
4	Ölü

4.2.2. SEER Veri Kaynağı Üzerinde Benzer Çalışmalar

SEER veri kaynağı üzerinde tıbbi amaçlı, istatistiksel ya da veri analizi üzerine pek çok araştırma yapılmış, konuyla ilgili pek çok bilimsel makale yayınlanmıştır.

Göğüs kanseri verileri ile yaşamda kalma tahminlemesi üzerine Delen ve ark. [13] yapay sinir ağlarını, karar ağaçlarını ve bir diğer istatistiksel metot olan lojistik regresyon metodunu 200,000'den fazla veri üzerinde karşılaştırmışlar. Modelleri oluştururken 10-kat çarpraz geçerleme metodunu kullanmışlar. Oluşturulacak sınıfları veri kaynağında var olan "Survival Time Recode" - STR - niteliğini kullanarak belirlemişler. Bu niteliğin değeri 60 ay ve üzeri ise "hayatta kaldı" sınıfına, 60 ayın altında ise "hastalığa yenik düştü" sınıfına atarak belirlemişler. Kullandıkları 1973-2000 yıllarını kapsayan 433,272 kayıttan oluşan göğüs kanseri veri kümesini önişlemler ile 202,932 kayıda düşürmüşler ve çalışmalarını bu kayıtlar üzerinde gerçekleştirmişler. Sonuç olarak C4.5 algoritmasının %93.6'lık bir doğruluk oranıyla en iyi modeli ürettiğini –ki bu oran literatürde görülmüş en iyi doğruluk oranıdır-, yapay sinir ağlarının %91.2 doğruluk oranıyla ikinci sırada yer aldığını, lojistik regresyon modelinin ise %89.2 oranıyla üçü arasında en kötü sonucu ürettiğini ifade etmişler.

Yine aynı veri kaynağı üzerinde, Delen ve ark.'nın çalışmalarını takiben, Abdelghani Bellaachia ve Erhan Guven [14], göğüs kanseri hastalarında yaşamda kalma tahminlemesi amacıyla çeşitli veri madenciliği algoritmalarının karşılaş-

tırmasını yapmışlar. NaiveBayes, geri yayınımlı yapay sinir ağları ve C4.5 algoritmalarından C4.5 algoritmasının daha iyi sonuçlar verdiği sonucuna ulaşmışlar. Modelleri oluştururken ücretsiz bir yazılım olan Weka Aracını kullanmışlar. Bu çalışmada Bellaachia ve ark., Delen ve ark.'nın yaptığı çalışmadan farklı olarak, ön işleme esnasında sınıflandırma amacıyla kullandıkları STR niteliğine ek olarak "Vital Status Recode" –VSR– ve "Cause of Death" –COD– niteliklerini de göz önünde bulundurarak sınıfları oluşturmuşlar. COD niteliğinin değeri önemlidir, çünkü ilk 60 ayda yaşanan ölümler kanser sonucu değil, doğal yollardan, trafik kazalarından ya da farklı sebeplerden de olabilir. Benzer şekilde, VSR niteliği de hastanın hayatta olup olmadığına dair en kuvvetli bilgiyi verir, yalnız başına STR niteliği ise hastanın teşhisten sonraki yaşamış olduğu süreyi verebildiği gibi en son kontrol tarihine kadar, ya da bilinen en son yaşamış olduğu tarihe kadar geçen süreyi ihtiva ediyor olabilir. Bu sebeplerle bu çalışmada hastanın "hayatta kaldı" ya da "hastalığa yenik düştü" sınıflarından birine atayabilmek için araştırmacılar;

if STR ≥ 60 and VSR ?= "Hayatta" **then**
    Sınıf <= "Hayatta Kaldı"
else if STR < 60 and COD ?= "Göğüs Kanseri" **then**
    Sınıf <= "Hastalığa Yenik Düştü"
else
    Kaydı göz ardı et;
end if

şeklinde bir yaklaşım uygulamışlar [14].

4.3. Veri Kaynağının Önişleme Prosedürü

SEER veri kaynağı, hasta kayıtlarını, niteliklerin değerleri arasında herhangi bir ayraç kullanılmadan, her bir kayıt bir satırda olacak şekilde farklı kanser tipleri için farklı metin dosyalarında tutmaktadır. Her satır 264 karakterden oluşmaktadır.

Çalışmada veri madenciliği algoritmalarının karşılaştırmasını yaparken 2000-2006 yılları arası Greater California, Kentucky, Louisiana ve New Jersey bölgelerine ait göğüs kanseri verileri kullanılmıştır. İlgili veri kaynağı incidence/ yr2000_2006.ca_ky_lo_nj/BREAST.txt dosyasında yer almaktadır.

Veri madenciliği analizi için kullanılan Weka Programı csv, arff, c4.5 libsvm, xarff gibi formatları desteklemektedir. Bunun için C programlama dili ile veri dosyasını arff formatına dönüştürecek bir program yazılarak weka programının bu dosyayı okuması sağlanmıştır. Bu transfer yapılırken aynı zamanda veri önişleme -nitelik azaltma- işlemi uygulanmış, analizde kullanılmayacak nitelikler göz ardı edilerek yeni dosyada bu değerlere yer verilmemiştir. Veri içerisinde değerleri girilmemiş olan nitelikler de, değerlerin girilmediğini ifade eden, global değerlerle tamamlanmıştır.

4.3.1. Arff formatı

Arff formatı Weka programının temel veri kaynağı olup temelde bir metin belgesidir. Metin belgesi içerisinde '%' karakteri ile başlayan satırlar yorum

satırlarıdır. Yorum satırları program tarafından değerlendirmeye alınmayan bilgi amaçlı yazılmış kontrol cümleleridir. Genel itibarı ile Arff dosyaları üç bölümden oluşur. İlk bölüm dosya içerisindeki veriyi tanımlayan isimlendirme bölümüdür. @relation kelimesi ile veri kaynağına isim verilir:

@relation SeerDataByCengizCoskun

İkinci bölüm veriyi tanımlayan - verinin hangi niteliklerden oluştuğunun tanımının yapıldığı - bölümdür. Her nitelik @attribute kelimesinden sonra nitelik adı ve tipi yazılarak tanımlanır. Veri tipi katar ise string, sayı ise numeric, tarih ise date veri tipleri kullanılır. Nominal değerler için niteliğin alacağı değerler küme parantezi "{ }" içerisinde listelenir:

@attribute "PatientID" string
@attribute "RegistryID" {00001541, 00001542, 00001543, 00001544}
@attribute "AgeAtDiagnosis" numeric

Yukarıdaki örneklerde PatientID katar, RegistryID nominal, AgeAtDiagnosis ise sayı tipinde tanımlanmıştır. Nitelikler, bir sonraki veri bölümünde bu bölümdeki tanımlanma sırasına göre yer almalıdır.

Üçüncü ve son bölüm ise verilere ait nitelik değerlerinin verildiği veri bölümüdür. Veri bölümü @data belirteci ile başlar ve metin dosyanın sonuna kadar devam eder. Her satırda bir nesneye ait tüm niteliklerin değerleri virgülle ayrılmış

şekilde yer alır. Katar tipindeki niteliklerin değerleri çift tırnak (“ “) içerisinde yazılırlar.

@data

"84074921",0000001544,058

"84074973",0000001544,059

4.3.2. Kullanılan Veri Kaynağındaki Niteliklerin Düzenlenmesi

Nitelik azaltma işleminde bütün nitelikler titiz bir şekilde incelenerek, sınıflandırma modeli üzerinde etkisinin az olduğu düşünülen nitelikler elenmiştir. Ham dosyada yer alan Registry ID, Marital Status, Sex, Age At Diagnosis, Primary Site, Laterality, Grade, Regional Nodes Positive, Tumor Marker 1, Tumor Marker 2, CS Tumor Size, Rx Summ- Surgery of Primary Site, Rx Summ-Scope of Regional Lymph Node Surgery, Rx Summ- Surgical procedure of Other Site, Rx Summ-Radiation, RX Summ- Radiation sequence with surgery, Behavior Recode for Analysis, Histology Recode Broad Groupings, Race Recode(W,B,AI,API), SEER Historic Stage A, Number of Primaries, Survival Time Recode, Cause of Death, Vital Status Code nitelikleri analizde değerlendirmek üzere seçilmiş, diğer nitelikler elenerek devam eden işlemlerde gözardı edilmiştir.

Arff formatına dönüştürme programında, değeri olmayan ya da anlamlı olmayan, analizde sorun yaratabilecek değerlere sahip bazı nitelikler de uygun değerlerle tamamlanmıştır. Bunlar:

1- ***Teşhis Anındaki Yaş*** (Age at Diagnosis): 3 karakterden oluşan hastanın yaşını belirten bu nitelik numerik değer taşımaktadır. 000-130 arası veriler hastanın yaşını belirtir. 999 değeri ise hasta yaşının bilinmediğini ifade etmektedir. 999 değerini taşıyan kayıtlar için Weka programında bu değerin missing value- eksik değeri - ifade etmesi için '?' karakteri ile değiştirilmiştir.

2- ***"Regional Nodes Positive" Niteliği***: Numerik bilgi taşıyan bu nitelik için maksimum değer 90 olduğundan 90 üzerindeki veriler '?' karakteri ile değiştirilip missing value - eksik değer - olarak güncellenmiştir. Zaten 90 üzerindeki bilgiler 'bilgi yok', 'lenf incelenmedi', 'Pozitif lenf mevcut ancak sayı belirtilmemiş' gibi temelde aynı anlam taşıyan değerleri ifade etmektedir.

3- ***"CS Tumor Size" Niteliği***: 2004 sonrası verilerde yer alan CS tumor size bilgisi 2004 öncesi verilerde yer almamaktadır. 2004 öncesi verilerde EOD-Tumor Size niteliği mevcuttur. Dolayısı ile CS- Tumor Size bilgileri 2004 öncesi veriler için EOD- Tumor Size değerleri ile tamamlanmıştır. 998 ve 999 kodları 'bilgi yok' anlamına geldiğinden '?' karakteri ile değiştirilmiştir. 991,992, … ,997 kodları ise sırasıyla maksimum 10mm, 20mm, … , 70mm anlamlarını taşıdığından 010, 020, … , 070 değerleri ile değiştirilmiştir.

4- ***Tumor Marker 1 ve Tumor Marker 2 Nitelikleri***: 2004 sonrası verilerde bu niteliklerin değerleri olmadığı için, değeri boş olan bu niteliklere "Bilinmiyor ya da bilgi girilmemiş" anlamına gelen '9' değeri atılmıştır.

5- ***"RX Summ – Surgery of Primary Site" Niteliği***: Boş olan bu nitelik değerleri için "Operasyon olup olmadığı bilinmiyor" anlamına gelen '99' değeri atılmıştır. 10-19 arası kodlar aynı anlama geldiği için 10 koduyla; benzer sebeple, 20-80 arası kodlar da 20 değeriyle değiştirilmiştir.

6- ***"RX Summ – Scope of Regional Lymph Node Surgery" Niteliği***: Boş olan bu nitelik değerleri için "Bilgi yok ya da Kodlama uygun değil" anlamına gelen '9' değeri atılmıştır.

7- ***"RX Summ - Surgical procedure of Other Site" Niteliği***: Boş olan bu nitelik değerleri için "Bilgi yok" anlamına gelen '9' değeri atılmıştır.

8- ***Sınıf Ataması***: SEER veri kaynağında hastaların hastalığı yenip yenemediğini ifade eden bir alan olmamakla birlikte, bu bilgiyi çıkarabileceğimiz Survival Time Recode –STR-, Cause of Death Recode to Seer Site Recode -COD- ve Vital Status Recode –VTR-, nitelikleri mevcuttur. Delen ve ark.'ın kullandığı STR niteliğine [13] Bellaachia ve arkadaşlarının ek olarak kullandığı VTR ve COD niteliklerini [14] göz önüne alarak sınıflandırma yapılmıştır. Özetle, sınıfı belirlemek için;

```
if (VSR ?= '1')                          /* hayatta mı? - Evet VSR=1 */
  {   if ( STR? > 60) Class= '1';        /*  60 aydan fazla ise savaşı kazandı  */
  }
else                                     /* hayatta değil -  VSR< >1  */
  {          /* Eğer COD kanser sebepli ölüm ise savaşı kaybetti */
     if (COD ?='Kanser') Class='2';
  }
```

yöntemi kullanılmıştır. Bu kriterlerin dışında kalanlar ise gözardı edilmiştir.

Yukarıda belirtilen ön işlemler weka arayüzü dışında, C programlama ile hazırlanmış olan ConvertSeerDataToArff programında yapılmıştır. İlgili C programı ve kütüphanesi EK1 ve EK2'de verilmiştir.

Program, 204,949 kayıt içeren BREAST.txt dosyası için çalıştırılmış ve elenen kayıtlar sonucu 60,984 kayıt içeren out.txt dosyası oluşturulmuştur (Şekil 4.2.1).

```
D:\csdsc>ConvertSeerDataToArff.exe BREAST.TXT
Okunan Kayıt Sayısı = 204949
Yazılan Kayıt Sayısı = 60948
Devam etmek için bir tuşa basın . . .
D:\csdsc>
```

Şekil 4.2.1. Seer Veri kaynağının hazırlanması için ConvertSeerDataToArff Programının çalıştırılması

4.3.3. Weka Arayüzünde Verinin Önişlemesi

SEER veri kaynağı, Weka programının okuyabileceği Arff formatına çevrildikten sonra Weka arayüzünde de önişlemden geçirilmiştir.

Katar olarak tanımlanmış “PrimarySite”, “Rx_SummSurgPrimeSite”, “HistologyRecode_BroadGroupings” nitelikleri “StringToNominal” süzgeci kullanılarak nominal niteliğe dönüştürülmüştür. “CS-TumorSize” niteliği 0-989 aralığında geçerli, genelde 0-70 aralığında yoğunlaşmış değerler taşımaktaydı. Bu alan 94 değeri maximum olacak şekilde 0-94 aralığında sınırlandırılmıştır. Orijinal veri kaynağındaki 118 adet nitelik sayısı niteliklerin üzerinden geçilerek 22 adede indirgenmişti. Bu sayı yine de sınıflandırma yapmak için çok fazlaydı ve detayın fazla olmasından dolayı oluşan modelde aşırı dallanma ve overfittinge sebep olacağı düşünülmüştür. Bu sebeple Weka programındaki “Attribute Selection” arayüzü kullanılarak istatiksel olarak hangi niteliklerin analizde daha verimli olacağı sorusuna yanıt aranmıştır. Weka’nın sunduğu 10’a yakın seçim algoritması çalıştırılmış, çıkan sonuçlar incelenerek nitelik sayısı 10’a düşürülmüştür. Bu nitelikler: “AgeAtDiagnosis”, “TumorNarker1”, “TumorMarker2”, “CS_TumorSize”, “RxSumm_SurgPrimeSite”, “RxSumm_ScopeRegLNSur”, “RxSumm_SurgOtherReg”, “BehaviorRecodeForAnalysis”, “HistologyRecode_BroadGroupings”, “SEERHistoricStageA” ve “MyClass” nitelikleridir.

4.4. Modellerin Oluşturulması ve Değerlendirilmesi

Önişlemler sonucunda elde edilmiş olan Arff formatındaki 60,948 kayıt içeren göğüs hastalıkları veri kaynağı üzerinde karar ağaçları, bayes, regresyon, örnek tabanlı sınıflandırma modellerinden birer algoritma seçilerek bunların başarıları karşılaştırılmıştır. Karşılaştırılacak algoritmalar seçilirken bu algoritmaların popülerliği ve literatürde benzer konuda yapılan çalışmalar dikkate alınmıştır.

Algoritmaları çalıştırırken test yöntemi olarak "10-kat çarpraz doğrulama" metodu kullanılmıştır. Bu yöntemle veri kaynağı 10 bölüme ayrılır ve her bölüm bir kez test kümesi, kalan diğer 9 bölüm öğrenme kümesi olarak kullanılır.

Oluşturulan modeller karşılaştırılırken doğruluk, kesinlik, duyarlılık ve F-Ölçütü değerleri kullanılmıştır.

4.4.1. Karar Ağacı Modelinin Başarım Ölçütleri

Weka aracında model oluştururken kullanılabilecek pek çok karar ağaçları algoritmaları mevcuttur. ADTree, BFTree, Decision Stump, FT, J48, J48graft, LADTree, LMT, RBTree, RandomForest, RandomTree, RepTree algoritmalarından J48 algoritması işlenmiş olan SEER veri kaynağı üzerinde çalıştırılmıştır. Tablo 4.2.1'de oluşturulan modelin test sonuçlarına ait istatistikler ve karışıklık matrisi görülmektedir. Tablo 4.2.2.'de ise karışıklık matrisini kullanarak hesaplanan karşılaştırma ölçütleri verilmiştir.

Tablo 4.2.1. J48 algoritmasına ait modelin test istatistiği (Weka çıktısı)

```
Correctly Classified Instances    52640     86.3687 %
Incorrectly Classified Instances  8308      13.6313 %
Kappa statistic                   0.6537
Mean absolute error               0.2107
Root mean squared error           0.3261
Relative absolute error           49.2942 %
Root relative squared error       70.5484 %
Total Number of Instances         60948

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class
```

```
0.965   0.363   0.856     0.965  0.907     0.873    1
0.637   0.035   0.892     0.637  0.743     0.873    2
0.864   0.262   0.867     0.864  0.856     0.873  WtdAvg

=== Confusion Matrix ===

   a     b   <-- classified as
 40629  1459 |     a = 1
  6849 12011 |     b = 2
```

Tablo 4.2.2. J48 Algoritmasına ait modelin karşılaştırma ölçütleri

Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü
%86.36	%85.57	%96.53	%90.72

4.4.2. Bayes (İstatistiksel) Sınıflandırma Modelinin Başarım Ölçütleri

Bayes Sınıflandırma için Weka'da var olan BayesNet, NaiveBayes, NaiveBayesSimple, NaiveBayesUpdateable algoritmalarından NaiveBayes algoritması seçilerek veri kümesi üzerinde çalıştırılmıştır. Tablo 4.2.3.'te oluşturulan modelin test sonuçlarına ait istatistikleri ve karışıklık matrisi görülmektedir. Tablo 4.2.4.'te ise karışıklık matrisini kullanarak hesaplanan karşılaştırma ölçütleri verilmiştir.

Tablo 4.2.3. NaiveBayes Algoritmasına ait modelin test istatistiği (Weka çıktısı)

```
Correctly Classified Instances    51932     85.2071 %
Incorrectly Classified Instances  9016      14.7929 %
Kappa statistic                   0.6318
Mean absolute error               0.1698
Root mean squared error           0.3491
Relative absolute error           39.7238 %
Root relative squared error       75.5217 %
Total Number of Instances         60948
```

```
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.942   0.348   0.858     0.942  0.898     0.883    1
0.652   0.058   0.833     0.652  0.732     0.883    2
0.852   0.258   0.85      0.852  0.846     0.883  WtdAvg

=== Confusion Matrix ===

   a     b  <-- classified as
 39627  2461 |     a = 1
  6555 12305 |     b = 2
```

Tablo 4.2.4. NaiveBayes Algoritmasına ait modelin karşılaştırma ölçütleri

Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü
%85.21	%85.80	%94.15	%89.78

4.4.3. Regresyon Modelinin Başarım Ölçütleri

Karşılaştırma amaçlı olarak regresyon tabanlı yöntemlerden lojistik regresyon algoritması seçilerek veri kaynağına uygulanmıştır. Tablo 4.2.5.'de oluşturulan modelin test sonuçlarına ait istatistikleri ve karışıklık matrisi görülmektedir. Tablo 4.2.6'da ise karışıklık matrisini kullanarak hesaplanan karşılaştırma ölçütleri verilmiştir.

Tablo 4.2.5. Lojistik regresyon algoritmasına ait modelin test istatistiği (Weka çıktısı)

```
Correctly Classified Instances    52025     85.3597 %
Incorrectly Classified Instances  8923      14.6403 %
Kappa statistic                   0.6385
Mean absolute error               0.2104
Root mean squared error           0.3337
Relative absolute error           49.2248 %
Root relative squared error       72.1804 %
Total Number of Instances         60948
```

```
=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.937   0.332   0.863     0.937  0.898     0.878    1
0.668   0.063   0.825     0.668  0.739     0.878    2
0.854   0.249   0.851     0.854  0.849     0.878

=== Confusion Matrix ===

   a     b   <-- classified as
 39420  2668 |     a = 1
  6255 12605 |     b = 2
```

Tablo 4.2.6. Lojistik regresyon algoritmasına ait modelin karşılaştırma ölçütleri

Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü
%85.36	%86.30	%93.66	%89.83

4.4.4. Örnek Tabanlı Sınıflandırma Modelinin Başarım Ölçütleri

Örnek tabanlı yöntemlerden Weka'da bulunan KStar algoritması kullanılarak model oluşturulmuştur. Tablo 4.2.7.'de oluşturulan modelin test sonuçlarına ait istatistikleri ve karışıklık matrisi görülmektedir. Tablo 4.2.8.'de ise karışıklık matrisini kullanarak hesaplanan karşılaştırma ölçütleri verilmiştir.

Tablo 4.2.7. KStar Algoritmasına ait modelin test istatistiği (Weka çıktısı)

```
Correctly Classified Instances    52076      85.4433 %
Incorrectly Classified Instances  8872       14.5567 %
Kappa statistic                   0.6194
Mean absolute error               0.2135
Root mean squared error           0.3275
Relative absolute error           49.96   %
Root relative squared error       70.8471 %
Total Number of Instances         60948
```

```
=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class
0.98    0.426   0.837     0.98   0.903     0.896    1
0.574   0.02    0.928     0.574  0.709     0.896    2
0.854   0.3     0.865     0.854  0.843     0.896  WtdAvg

=== Confusion Matrix ===

   a     b   <-- classified as
 41247   841 |     a = 1
  8031 10829 |     b = 2
```

Tablo 4.2.8. KStar Algoritmasına ait modelin karşılaştırma ölçütleri

Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü
%85.44	%83.70	%98.00	%90.28

4.5. Oluşturulan Modellerin Karşılaştırılması

Önişlemden geçirilen SEER veri kaynağı J48, NaiveBayes, Lojistik Regresyon ve KStar algoritmaları ile analiz edilerek her algoritma için oluşmuş olan modele ait test istatistiği bir önceki bölümde verilmişti. Karşılaştırma yapabilmek için her modele ait karşılaştırma ölçüt değerleri Tablo 4.2.9.'da genel bir tabloda yeniden verilmiştir.

Tablo 4.2.9.

Algoritma / Ölçüt	J48	NaiveBayes	Lojistik Regresyon	KStar
Doğruluk	0.863687	0.852071	0.853597	0.854433
Kesinlik	0.855744	0.858062	0.863054	0.837027
Duyarlılık	0.965335	0.941527	0.936609	0.980018
F-Ölçütü	0.907242	0.897859	0.898328	0.902896

Tablo 4.2.9'daki veriler incelendiğinde J48 algoritmasının model testine ait %86.36 doğruluk derecesiyle en iyi sonucu ürettiği söylenebilir. Doğruluk ölçütü oldukça basit ve önemli bir kriterdir. Bu ölçüte göre J48 algoritmasını sırasıyla KStar, Lojistik Regresyon ve NaiveBayes algoritmaları izlemektedir.

Kesinlik ölçütü bakımından lojistik regresyon en iyi sonucu oluşturmuş olup, diğer algoritmalar bu ölçüte göre NaiveBayes, J48 ve KStar şeklinde sıralanabilir. Ancak kesinlik ölçütü tek başına yorumlanırsa değerlendirme yanlış sonuçlara götürebilir. Bu ölçütü duyarlılık ölçütüyle beraber ele almak gerekir. Tablodan görüleceği üzere algoritmalar, duyarlılık ölçütüne göre KStar, J48, NaiveBayes ve Lojistik Regresyon olarak sıralanabilir. Görüleceği üzere, kesinlik ölçütü ve duyarlılık ölçütü birbiriyle zıt bir sıralama ortaya koymuştur.

Kesinlik ve duyarlılık ölçütlerini beraber değerlendirmek için, her iki değerin harmonik ortalaması olan F-ölçütüne baktığımızda sıralamanın J48, KStar, Lojistik Regresyon ve NaiveBayes şeklinde olduğunu görüyoruz. Bu sıralamanın yine doğruluk kriter sıralamasıyla birebir aynı olduğu görülmektedir.

5. BULGULAR VE TARTIŞMA

İstatistik, örnek verilerden hareket ederek popülasyon hakkında yorumlama, genelleme ve tahminleme yapma bilimi diye tanımlanır. Veri madenciliği de dijital ortamdaki büyük veri yığınlarından bilgi çıkarmayı ve bu bilgiyi değerlendirmeyi amaçlar.

Bu amaçla yapılan çalışmalarda karşılaşılan en büyük problem veri kaynağının hatalı veriler içermesi ya da çok sayıda nitelik değerinin eksik girilmiş olmasıdır. SEER veri kaynağı son derece düzenli ve dünya çapında bilinen; istatistiksel çalışmalarda yoğun bir şekilde kullanılan veri kaynağıdır. Veri kaynağındaki nitelikler belirli bir format dahilindedir ve niteliklerin açıklamaları veri kaynağı ile birlikte kullanıma verilmektedir. Bu sebeplerle SEER veri kaynağının böyle bir uygulamada kullanılması uygulamanının sonuçlarını daha güvenilir kılacaktır.

Veri kaynağı ne kadar düzenli ve güvenilir olursa olsun, bir veri madenciliği uygulamasında kullanabilmek için veriler üzerinde önişlem yapmak gerekir. Bu çalışmada da kullanmış olduğum veri kaynağı üzerinde işlemler yapılmış, veri kaynağı analize uygun yapıya dönüştürülmüştür. Veri önişleme basamağında yapılan en önemli çalışma verinin temizlenmesi, analizde kullanılmayacak gereksiz bilgilerin veriden çıkarılarak verinin düzenlenmesi işlemidir. Yaptığım çalışmada kullandığım SEER veri kaynağı çalışma öncesinde incelenmiş ve analizde sonucu etkilemeyecek olan bazı nitelikler veriden çıkarılmıştır. Bu nitelikleri belirlerken tıbbi bilgimin

yeterli olmaması optimal bir seçim yapamamış olmama sebep olmuş olabilir; tıbbi sonuçlara ulaşmayı hedef alan bir çalışma yapıyor olsaydım burada yapılmış hataların sonuçlar üzerinde olumsuz etkileri olacaktı. Ancak buradaki amacım tıbbi sonuçlar elde etmek yerine model başarımı konusunda istatistiksel sonuçlara ulaşmak olduğu için bu hataların etkisinin sınırlı olacağını düşünüyorum. Ayrıca, nitelik seçimi yaparken oldukça titiz davrandığımı, mümkün olduğunca her niteliğin etkisini değerlendirerek ve veri dokümantasyonunu dikkatle inceleyerek karar aldığımı, bu noktada, vurgulamak isterim.

Veri madenciliğinde bilgiye erişmede farklı metotlar kullanılmaktadır. Bu metotlara ait pek çok algoritma vardır. Bu algoritmalardan hangisinin daha üstün olduğu üzerine pek çok çalışma yapılmış, yapılan bu çalışmalarda farklı sonuçlar elde edilmiştir. Bunun en önemli sebebi, işlem başarımının kullanılan veri kaynağına, veri üzerinde yapılan önişleme, algoritma parametrelerinin seçimine bağlı olmasıdır. Farklı kişiler tarafından, farklı veri kaynakları üzerinde, farklı parametrelerle yapılan çalışmalarda farklı sonuçlar oluşması doğaldır. Ancak, yaptığım çalışma, "benzer veri kümelerinde belli yöntemlerin daha başarılı olduğu" şeklindeki çıkarıma [2] uygun olarak, diğer çalışmalarla [13,14] benzer sonuç vermiştir. Göğüs kanseri vakalarının farklı yıllarını içeren SEER veri kaynağının kullanıldığı her iki çalışmada, bir karar ağacı algoritması olan C4.5 algoritmasının diğer algoritmalardan daha iyi sonuç ürettiği sonucuna ulaşıldığı belirtilmiştir [13,14]. Bu tez çalışmasında da, yapılan karşılaştırma sonucunda, C4.5 algoritmasının Weka implementasyonu olan J48 karar ağacı algoritması, benzer şekilde diğer algoritmalara göre daha başarılı bulunmuştur.

Yaptığım çalışmada, üretilen modellerin doğruluk oranı, Delen ve ark.'nın benzer veriler üzerinde yaptığı çalışmaya ait modellerin belirtilen doğruluklarından düşük olmuştur. Bunu, aşağıda belirttiğim sebeplere bağlayabiliriz:

- Kullanmış olduğum veri kaynağı 2003-2006 yıllarına aittir. Delen ve arkadaşları ise daha eski (1973-2000) yıllara ait SEER verileri üzerinde çalışmışlardır. Veri kaynakları farklıdır ve Delen ve ark.'nın kullanmış olduğu veri kaynağı daha fazla kayıt içermektedir. Kanser çalışmalarında ve tıp biliminde yaşanan gelişmeler neticesinde SEER veri kaynağındaki bilgiler değişmiştir. Bu sebeple eski yıllara ait veri kaynağı ile sonraki yıllara ait veri kaynağı arasında hem içerik hem format anlamında farklılıklar oluşmuştur.

- Veri önişleme aşamasında yaptığım nitelik seçimi, verilerin tamamlanması gibi analiz sonuçlarını etkileyici işlemlerde farkında olmadan model çıkarımını etkileyici işlemler yapılmış olabilir. Farklı önişlemelerle oluşturulan verilerin analiz sonuçlarının farklı olması kaçınılmazdır.

- Yaptığım çalışmada, algoritmaların kullandığı parametreler varsayılan değerler olarak seçilmiştir. Bundaki amacım, algoritmalar arasında pozitif ayrımcılık denebilecek durumlara yol açmamak; amacımın, modelleri

daha iyi oluşturmak olmayan bir çalışmada, çalışmanın farklı bir istikamete yönelmesini önlemektir.

Özetle, model başarımlarının Delen ve ark.'larınınkinden düşük olmasının sebepleri veri kaynağının hem içerik hem format açısından farklı olması, yapılan veri önişleme çalışmasının farklı olması, algoritmalar çalıştırılırken muhtemelen farklı parametreler kullanılması olarak söylenebilir.

Bir önceki bölümde yaptığımız karşılaştırmayı, J48 algoritmasının SEER veri kaynağındaki göğüs kanseri kayıtları üzerinde diğer algoritmalara göre daha iyi tahmin sonuçları oluşturduğu şeklinde özetleyebiliriz. Ancak, Tablo 4.2.9.'daki rakamlara baktığımızda değerler arasında büyük farklar olmadığını, en azından J48 ile en yakın takipçisi arasında doğruluk ve F-ölçütü açısından %1'in altında fark olduğunu görürüz. Bu sonuç da bizi böyle bir genelleme yapmaktan alıkoymaktadır.

Veri madenciliği algoritmalarının karşılaştırma yolu ile yapılan deneysel çalışmalar bilim dünyasında keskin eleştirilere maruz kalmaktadır. Hand [17], veri madenciliği algoritmalarının karşılaştırılması hakkında karşılaştırma sonuçlarının doğru olmayacağını, literatürde yer alan makalelerdeki çalışmaların aslında bir illüzyon yarattığını, deneysel çalışmaların ortaya koyduğu sonuçların gerçekle bağdaşmayacağını belirtmiştir. Doğası gereği veri madenciliği model başarımlarının veriye bağlı olduğunu, veri üzerinde yapılan önişleme işlemlerinin ve kullanılan algoritma parametrelerinin oluşan sonuç üzerinde farklı etkileri olacağını, kullanıcıya bağlı olarak aynı modelle farklı sonuçlar elde edilebileceğini belirtmiştir. Literatürde

yer alan ve yeni geliştirilmiş olan bir algoritmanın eski bir algoritmayla karşılaştırılması yapılarak yeni algoritmanın daha başarılı olduğunun ispatlanmaya çalışıldığı makalelerde, bu sebeplerden ötürü ve geliştiricinin isteyerek ya da istemeden yanlı yaklaşımlarının etkili olacağı belirtilmektedir. Literatürdeki diğer karşılaştırma çalışmalarında sonucun kullanıcının yatkın olduğu modele bağlı olduğu, bu yüzden farklı makalelerde farklı sonuçlara ulaşılacağı belirtilmiştir. Bunun ötesinde bazı çalışmalarda kompleks algoritmaların klasik algoritmalara karşı daha başarılı olduğu şeklindeki iddiaların da aslında illüzyondan ibaret olduğu ifade edilmektedir [17].

Salzberg, Prechelt'in 1996 yılında yapay sinir ağları hakkında var olan 200 makale üzerinde yaptığı araştırmasında, incelenen çalışmaların çoğunda büyük deneysel hatalar olduğu, yeni geliştirilen algoritmaların büyük bir oranının (%29) gerçek verilerle test edilmediği, sadece %8'lik kısmının gerçek veriler kullanılarak eski algoritmalarla karşılaştırmasının yapıldığı sonucuna ulaştığını ifade etmektedir. Salzberg, makalesinde hiçbir sınıflandırıcının diğerlerine göre tamamen daha iyi olamayacağını, algoritma karşılaştırmasının hatalara ve yanlış yorumlamalara açık olabileceğini belirtmiştir. Ancak buna rağmen deneysel karşılaştırmaların gerekliliğini, bu çalışmalarda hatalı çıkarımlara yol açmayacak şekilde dikkat edilmesinin önemini vurgulamıştır [18].

Deneysel çalışmalar üzerine yapılan bu eleştirilerin haklılık payı büyüktür. Dolayısı ile yapılan bir karşılaştırma işlemine dayanarak bir algoritmanın diğer bir algoritmaya kesin bir üstünlüğünden söz etmek doğru olmayacaktır. Ancak model başarımı karşılaştırmalarının, bir veri madenciliği çalışmasında önemli katkıları

olacağı açıktır. Bir kullanıcının bir problem üzerinde yapacağı model oluşturma işleminde farklı algoritmaları karşılaştırarak en başarılıyı bulmasının ve modelini o algoritma ile kurmasının elbette sonuçlar üzerinde olumlu etkisi olacaktır. Ancak, burada dikkat edilmesi gereken nokta öğrenme kümesinin seçimidir. Çünki farklı öğrenme kümeleriyle yapılan farklı karşılaştırmalar farklı sonuçlar verebilir [17]. Ayrıca yeni geliştirilen bir algoritmanın bilimsel anlamda geçerliliğinin belirlenmesinde deneysel çalışmaların önemli bir yeri vardır.

6. SONUÇLAR VE ÖNERİLER

Veri madenciliğinde sınıflandırma modellerinden karar ağaçları, Naivebayes, lojistik regresyon ve örnek tabanlı sınıflandırma yöntemlerinden seçilen dört algoritmanın, 2000-2006 yılları arası SEER veri kaynağındaki göğüs kanseri hasta kayıtları üzerinde yapılan karşılaştırması sonucunda bir karar ağacı algoritması olan J48 algoritmasının diğer algoritmalara göre nispeten daha iyi model oluşturduğu görülmüştür.

Sınıflandırma algoritmalarının karşılaştırma yöntemlerini inceleyen bu tez çalışmasında veri madenciliği ve karşılaştırma ölçütleri üzerinde durulmuştur. Genel anlamda hangi algoritmanın daha iyi model ürettiği şeklinde bir çalışmada farklı veri kaynakları üzerinde, daha çok sayıda algoritma kullanarak karşılaştırma yapılması gerekecektir. Michie ve ark'ın araştırma sonuçlarını yayınladıkları kitaplarında belirtmiş oldukları "farklı veri kaynaklarında farklı algoritmaların daha başarılı olduğu" [2] tezi göz önüne alınarak farklı veri kaynaklarındaki karşılaştırmaların kategorize edilmesi gerekebilecektir.

Bu çalışmada, modellerin oluşturulması için ücretsiz bir yazılım olan Weka aracı kullanılmıştır. Var olan diğer veri madenciliği araçları üzerinde aynı algoritmalar çalıştırılarak farklı araçların benzer sonuçlar üretip üretmediği kontrol edilebilir.

Algoritmaların veri kaynağı üzerinde çalıştırılması sırasında algoritma parametreleri olarak her algoritmanın o parametre için varsayılan değeri kullanılmıştır. Her algoritma ve her veri kaynağı için başarım derecesini maksimize edecek parametre değerleri tespit ederek bu parametrelerle algoritma sonuçlarını karşılaştırmak farklı sonuçlara götürebilecektir. Ancak, böyle bir karşılaştırmada yanlılık oluşabilecektir.

Bu çalışmada, algoritmaların ürettiği modellerin başarım sonuçları karşılaştırılmıştır. Benzer şekilde, algoritmaların hızı ve hafıza kullanımı ile algoritmaların performans karşılaştırması da yapılabilir.

7. KAYNAKLAR

1. Elder, J. F. ; Abbot, D. W. ; *A Comparison of Leading Data Mining Tools;* Fourth International Conference on Knowledge Discovery& Data Mining, New York, **1998**.

2. Michie, D. ; Spiegelhalter, D. J. ; *Machine Learning, Neural and Statistical Classification*; Taylor, C. C. ; Prentice Hall, **1994**.

3. Wilson, R. L. ; Sharda, R. ; *Bankruptcy prediction using neural networks*; Decision Support Systems, Vol. 11, Issue 5 ; Elsevier Science Publishers B. V. Amsterdam, The Netherlands; **1994**; 545-557.

4. Lin, M. ; Huang, S. ; Chang, Y. ; *Kernel-based discriminant technique for educational placement*; Journal of Educational and Behavioral Statistics, Vol 29; **2004**; 219-240.

5. King, R. D. ; Feng, C. ; Sutherland, A. ; *StatLog: Comparison of Classification Algorithms on Large Real-World Problems*; Applied Artificial Intelligence, Vol 9, Issue 3 ; **1995**; 289-333.

6. Wu, X. ; Kumar, V. ; Quinlann, J. R. ; Ghosh, J. ; Yang, Q. ; Motoda, H. ; McLachlan, G. J. ; Angus, N. ; Bing, L. ; Yu, P. S. ; Zhou, Z. H. ; Steinbach, M. ; Hand, D. J. ; Steinberg, D. ; *Top 10 algorithms in data mining*; Knowledge of Information Systems, Vol 14; **2008**; 1-37.

7. Sabzevari, H. ; Soleymani, M. ; Noorbakhsh, E. ; *A comparison between statistical and Data Mining methods for credit scoring in case of limited available data*; Eleventh Annual APRIA Conference; **2007**.

8. Rajavarman, V.N. ; Rajagopalan, S.P. ; *Comparison between Traditional data mining Techniques and Entropy-based Adaptive Genetic Algorithm for Learning*

Classification Rules; International Journal of Soft Computing Vol 2 Issue 4; **2007**; 555-561.

9. Zurada, J. ; Lonial, S. ; *Comparison Of The Performance Of Several Data Mining Methods For Bad Debt Recovery In The Healthcare Industry*; The Journal of Applied Business Research Vol 21; **2005**; 37-53.

10. Agrawal, R. ; Imielinski, T. ; Swami, A. N. ; *Mining association Rules between Sets of Items in Large Databases* ; International Conference on Management of Data; **1993** ; 207-216.

11. Liu, B. ; Hsu, W. ; Ma. Y. ; *Mining Association Rules with Multiple Minimum Supports*. International Conference on Knowledge Discovery and Data Mining; **1999**; 337-341, 1999.

12. Liu, B. ; *Web Data Mining*; Carey, M.J. ; Ceri, S. ; Springer; Berlin; **2007**.

13. Delen, D. ; Walker, G. ; Kadam, A. ; *Predicting breast cancer survivability:a comparison of three data mining methods*; Artificial Intelligence in Medicine, Vol 34, issue 2 ; **2004**; 113-127.

14. Bellaachia, A. ; Guven, E. ; *Predicting breast cancer survivability: a comparison of three data mining method* ; Ninth Workshop on Mining Scientific and Engineering Datasets in conjunction with the Sixth SIAM International Conference on Data Mining (SDM 2006) ; **2006**.

15. SEER veri kaynağı ve dokümantasyon ; (21 Temmuz 2010) Erişim: http://www.seer.cancer.gov

16 . Witten, I. H. ; Frank, E. ; *Data Mining, Practical Machine Learning Tools and Techniques* ; Morgan Kaufmann , USA; **2005**.

17. Hand, D. J. ; *Classifier Technology and the Illusion of Progress*; Statistical Science, Vol. 21; Institute of Mathematical Statistics, **2006**; 1-15.

18. Salzberg, L. ; *Methodological Note On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach* ; Data Mining and Knowledge Discovery 1 ; Kluwer Academic Publishers, Boston ; **1997**; 317–328.

19. Baykal, A. ; Ders Notları, Veri Madenciliği.

20. Freund's, J. E. ; *Mathematical Statistics with Applications*; Prentice Hall, USA ; **2004**; 7. baskı.

EKLER

EK 1. (DataTypeDefinitions.h C kütüphanesi kaynak kodu) C programlama dili ile yazılmış C kütüphanesidir. Bu kütüphane SEER veri kaynağını Arff formatına dönüştüren ConvertSeerDataToArff programı tarafından kullanılmaktadır. Ana program tarafından kullanılan fonksiyonlar ve veri tipleri tanımlanmıştır. SEER kaynağındaki satırların içeriği *recs[118]* dizisinde tanımlanmıştır. Bu dizinin her bir elemanı SEER satırındaki niteliklerin başlangıç pozisyonunu, uzunluğunu, adını, tipini (0: nominal, 1: numerik, 2: katar), analizde yer alıp almayacağını ve eğer nitelik nominal ise alabileceği nominal değerler kümesini içerir. Kütüphanede yer alan *int prepareData(char *str)* fonksiyonu ise her bir satırı arff formatına çevirirken eksik değerlerin doldurulması, niteliklerin birleştirilmesi gibi veri önişleme işlemlerini uygular. Ayrıca STR, VTR, COD bilgilerini kullanarak örneğin sınıfını belirler.

```
/*-------------------------------------------------------------------------------------------*/
/*  Library “DataTypeDefinitions.h” used by ConvertSeerDataToArff            */
/*  implemented by Cengiz Coşkun.                                            */
/*  Includes Data Type Def.’s and functions                                  */
/*-------------------------------------------------------------------------------------------*/
typedef char * string;
char myClass;
string substrf(string pstr, int bas, int length){
    string temp;
    temp =malloc((length+1)*sizeof(char));int k=bas,i=0;
    for (;i<length;k++,i++) temp[i]=pstr[k];
```

```
    temp[i]='\0';
    return temp;
}
int ctoint (char c) {return c - '0';}

/* SEER Data definition */
const int numfields = 117;
enum data_types {sstring=0, snumeric=1, snominal=2} mynums;
char *dataTypeName[3]={"string","numeric","nominal"};
struct { int  bas;              //Start position of the field
         int  len;               //Length of the field
         char ad[100];          //Field Definition
         int type;              //0 nominal ; 1 numeric ; 2 katar
         int included;          //0: not included, 1:included
        char values[100];   //nominal değerler
    } recs[118] = {
          {  0,  8,  "\"PatientIDs\"",0,0,""},

            /* RegistryID Hastanın hangi bölgeden olduğu
            1541 California, 1542 Kentucky,1543 Louisiana, 1544 New Jersey */
          {8, 10, "\"RegistryID\"", 2, 1,
"{0000001541,0000001542,0000001543,0000001544}"},

          /* Medeni Durum: 1 bekar, 2 evli, 3 ayrı, 4 boşanmış, 5 dul,  9 bilinmiyor */
```

```
{ 18,  1,  "\"MaritalStatus\"",2,1,"{1,2,3,4,5,9}"},
{ 19,  2,  "\"Race\"",2,0,"{01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13,
14, 20, 21, 22, 25, 26, 27, 28, 30, 31, 32, 96, 97, 98, 99}"},//White, Black,
AmericanIndianAlaskan, Chinese, Japanese, Filipino, Hawaiian, Korean...
{ 21,  1,  "\"SpanishOrHispanic\"",0,0,""},
{ 22,  1,  "\"NHIAderivedHispanic\"",0,0,""},
 /* Cinsiyet 1 Erkek 2 Dişi */
{ 23,  1,  "\"Sex\"",2,1,"{1,2}"},
{ 24,  3,  "\"AgeAtDiagnosis\"",1,1,""},
{ 27,  4,  "\"YearOfBirth\"",0,0,""},
{ 31,  3,  "\"BirthPlace\"",0,0,""},
{ 34,  2,  "\"SequenceNumberCentral\"",0,0,""},
 /* Month and Year of diagnosis (mmyyyy) */
{ 36,  6,  "\"DateOfDiagnosis\"",0,0,""},
{ 42,  4,  "\"PrimarySite\"",0,1,""},     //
{ 46,  1,  "\"Laterality\"",2,1,"{0,1,2,3,4,9}"},
{ 47,  4,  "\"Histology\"",0,0,""},
{ 51,  1,  "\"Behaviour\"",0,0,""},
{ 52,  4,  "\"HistologicType\"",0,0,""},
{ 56,  1,  "\"BehaviourCode\"",2,0,"{0,1,2,3}"},
{ 57,  1,  "\"Grade\"",2,1,"{1,2,3,4,5,6,7,8,9}"},
{ 58,  1,  "\"DiagnosticConfirmation\"",0,0,""},
{ 59,  1,  "\"TypeOfReportingSource\"",0,0,""},
{ 60,  3,  "\"EOD_TumorSize(mm)\"",1,0,""},
```

```
{ 63,  2,  "\"EOD_Extension\"",0,0,""},
{ 65,  2,  "\"EOD_ExtensionProstPath\"",0,0,""},
{ 67,  1,  "\"EOD_LymphNodeInvolv\"",2,0,"{0,1,2,3,4,5,6,7,8,9}"},
{ 68,  2,  "\"RegionalNodesPositive\"",1,1,""},
{ 70,  2,  "\"RegionalNodesExamined\"",0,0,""},
{ 72, 13,  "\"EOD_Old13Digit\"",0,0,""},
{ 85,  2,  "\"EOD_Old2Digit\"",0,0,""},
{ 87,  4, "\"EOD_Old4Digit\"",0,0,""},
{ 91,  1,  "\"CodingSystemForEOD\"",0,0,""},
{ 92,  1,  "\"TumorMarker1\"",2,1,"{0,1,2,3,4,5,6,8,9}"},
{ 93,  1,  "\"TumorMarker2\"",2,1,"{0,1,2,3,4,5,6,8,9}"},
{ 94,  1,  "\"TumorMarker3\"",0,0,""},
{ 95,  3,  "\"CS_TumorSize\"",1,1,""},
{ 98,  2,  "\"CSExtension\"",0,0,""},
{100,  2,  "\"CSLymphNodes\"",0,0,""},
{102,  2,  "\"CSMetsAtDx\"",0,0,""},
{104,  3,  "\"CSsiteSpecificF1\"",0,0,""},
{107,  3,  "\"CSsiteSpecificF2\"",0,0,""},
{110,  3,  "\"CSsiteSpecificF3\"",0,0,""},
{113,  3,  "\"CSsiteSpecificF4\"",0,0,""},
{116,  3,  "\"CSsiteSpecificF5\"",0,0,""},
{119,  3,  "\"CSsiteSpecificF6\"",0,0,""},
{122,  2, "\"DerivedAJCC_T\"",0,0,""},
{124,  2,  "\"DerivedAJCC_N\"",0,0,""},
```

{126, 2, "\"DerivedAJCC_M\"",0,0,""},

{128, 2, "\"DerivedAJCC_StageGroup\"",0,0,""},

{130, 1, "\"DerivedSS1977\"",0,0,""},

{131, 1, "\"DerivedSS2000\"",0,0,""},

{132, 1, "\"DerivedAJCCFlag\"",0,0,""},

{133, 1, "\"DerivedSS1977Flag\"",0,0,""},

{134, 1, "\"DerivedSS2000Flag\"",0,0,""},

{135, 6, "\"CS_version1\"",0,0,""},

{141, 6, "\"CS_versionLatest\"",0,0,""},

{147, 2, "\"RxSumm_SurgPrimeSite\"",0,1,""},

{149, 1, "\"RxSumm_ScopeRegLNSur\"",2,1,"{0,1,2,3,4,5,6,7,9}"},

{150, 1, "\"RxSumm_SurgOtherReg\"",2,1,"{0,1,2,3,4,5,9}"},

{151, 2, "\"RxSumm_RegLNExamined\"",0,0,""},

{153, 1, "\"RxSumm_ReconstructFirst\"",0,0,""},

{154, 1, "\"ReasonForNoSurgery\"",0,0,""},

{155, 1, "\"RxSumm_Radiation\"",2,1,"{0,1,2,3,4,5,6,7,8,9}"},

{156, 1, "\"RxSumm_RadToCNS\"",0,0,""},

{157, 1, "\"RxSumm_SurgRad_Seq\"",2,1,"{0,2,3,4,5,6,9}"},

{158, 2, "\"RxSumm_SurgeryType\"",0,0,""},

{160, 2, "\"RxSumm_SurgSite98_02\"",0,0,""},

{162, 1, "\"RxSumm_ScopeReg98_02\"",0,0,""},

{163, 1, "\"RxSumm_SurgeOther98_02\"",0,0,""},

{164, 2, "\"SEER_RecordNumber\"",0,0,""},

{166, 1, "\"OverrideAgeSiteMorph\"",0,0,""},

{167, 1, "\"OverrideSeqnoDxconf\"",0,0,""},

{168, 1, "\"OverrideSiteLatSeqno\"",0,0,""},

{169, 1, "\"OverrideSurgDxconf\"",0,0,""},

{170, 1, "\"OverrideSiteType\"",0,0,""},

{171, 1, "\"OverrideHistology\"",0,0,""},

{172, 1, "\"OverrideReportSource\"",0,0,""},

{173, 1, "\"OverrideIllDefineSite\"",0,0,""},

{174, 1, "\"OverrideLeukLymph\"",0,0,""},

{175, 1, "\"OverrideSiteBehaviour\"",0,0,""},

{176, 1, "\"OverrideSiteEodDxDt\"",0,0,""},

{177, 1, "\"OverrideSiteLatEOD\"",0,0,""},

{178, 1, "\"OverrideSiteLatMOrph\"",0,0,""},

{180, 1, "\"SEERtypeOfDFollowup\"",0,0,""},

{182, 2, "\"AgeRecode<1\"",2,0,""},

{184, 5, "\"SiteRecode\"",0,0,""},

{189, 5, "\"SiteRecWithKaposiAndMesoThelioma\"",0,0,""},

{194, 4, "\"RecodeICD_O2to9\"",0,0,""},

{198, 4, "\"RecodeICD_O2to10\"",0,0,""},

{202, 3, "\"ICCCsiteRecodeICD_O2\"",0,0,""},

{205, 3, "\"SEERModifiedICCCsiteRecode\"",0,0,""},

{208, 3, "\"ICCsiteRecodeICD_O3\"",0,0,""},

{211, 3, "\"ICCsiteRecodeExtendedICD_O3\"",0,0,""},

{214, 1, "\"BehaviourRecodeForAnalysis\"",2,1,"{0,1,2,3,4,5}"},

{215, 1, "\"ICD_OcodingScheme\"",0,0,""},

```
{216,  2,  "\"HistologyRecode_BroadGroupings\"",0,1,""},
{218,  2,  "\"HistologyRecode_BrainGroupings\"",0,0,""},
{220,  2,  "\"CSschema\"",0,0,""},
/* Race Recode (W_B_AI_AP) hastanın menşeyi.
   1 white, 2 black, 3 American/Alaskalı(yerli), 4 Asian or Pacific Islander,
   7 OtherUnspecified, 9 Unknown) */
{222,  1,  "\"RaceRecode(WhiteBlackOther)\"",2,0,"{1,2,3,7,9}"},
{223,  1,  "\"RaceRecode(W_B_AI_AP)\"",2,1,"{1,2,3,4,7,9}"},
{224,  1,  "\"OriginRecodeNHIA(hisp_NonHisp)\"",0,0,""},
{225,  1,  "\"SEERhistoricStageA\"",2,1,"{0,1,2,4,8,9}"},
{226,  2,  "\"AJCCstage3edition\"",0,0,""},
{228,  2,  "\"SEERmodifiedAJCCstage3\"",0,0,""},
{230,  1,  "\"SEERsummaryStage1977\"",0,0,""},
{231,  1,  "\"SEERsummaryStage2000\"",0,0,""},
{232,  2,  "\"NumberOfPrimaries\"",1,1,""},
{234,  1,  "\"FirstMalignantPrimaryIndicator\"",0,0,""},
{235,  5,  "\"StateCountryCode\"",0,0,""},
/* STR, COD ve VSR alanları sınıfı belirleyecek: 1:hayatta; 2:değil */
{240,  4,  "\"*__SurvivalTimeRecode__*\"",1,0,""},
{244,  5,  "\"*__CauseOfDeathtoSEERsiteRecode__*\"",0,0,""},
{249,  5,  "\"*__CODtoSiteRecKM__*\"",0,0,""},
{254,  1,  "\"*__VitalStatusCode__*\"",2,0,"{1,4}"},
{255,  3,  "\"IHSlink\"",0,0},
{258,  1,  "\"SummaryStage2000\"",0,0,""},
```

```
            {259,  2,  "\"AYAsiteRecode\"",0,0,""},
            {261,  2,  "\"LymphomaSubtypeRecode\"",0,0,""},
            {263,  1,  "\"VitalStatusRelativetoSiteAtXx\"",0,0,""}
          };

int prepareData(char *str){
   /* return 9 as class to ignore if not satisfying my class conditions.*/
   myClass='9';
   // Eğer yaş bilinmiyor ise (999 ise) ? ile değiştir.
   if ((str[24]=='9')&& (str[25]=='9') &&(str[26]=='9'))
     {str[24]='?';  str[25]=' ';  str[26]=' ';  }

   /* RegionalNodesPositive bilgisi yoksa 99 değerini ata bilinmiyor; girilmemiş;
      hasta kaydında yok 90 değerinin altındakiler sayısal değer ifade ettiği için 90
      üzerindekileri missing value yap */
   if (str[68]==' ') { str[68]='9';str[69]='9';}
   if ((str[68]=='9') && (str[69]!='0'))  { str[68]='?'; str[69]=' '; }

   /* CS-tumor size boş ise EOD-tumor size değerini kullan
      998 ya da 999 ise missing value olarak güncelle
      991 992 993 994 995 değerlerinden biri ise 10mm 20mm..50mm olarak güncelle
*/
   if (str[95]==' ') { str[95]=str[60]; str[96]=str[61];str[97]=str[62];}
   if (((str[95]=='9') && (str[96]=='9') &&(str[97]=='8'))||
```

```
    ((str[95]=='9') && (str[96]=='9') &&(str[97]=='9')))
     { str[95]='?'; str[96]=' '; str[97]=' '; }
   if ((str[95]=='9')&&(str[96]=='9') && str[97]<'9')
     { str[95]='0'; str[96]=str[97]; str[97]='0'; }

   // CS-extension boş ise EOD-extension değerini kullan
   if (str[98]==' ') { str[98]=str[63]; str[99]=str[64];}
   // CS-lymph Nodes boş ise EOD-lymphNodes değerini kullan
   if (str[100]==' ') { str[100]=str[67]; str[101]=str[68];}

   /* 2004 sonrası verilerde tumor marker 1 ve tumor marker2 olmadığı için bunların
değerlerine 9 ata. */
   if (str[92]==' ') { str[92]='9'; }
   if (str[93]==' ') { str[93]='9'; }

   // RxSumm_SurgPrimeSite değeri yoksa değerlerine 99 ata. /*bilinmiyor */
   if (str[147]==' ') { str[147]='9'; str[148]='9';}

   // RX Summ SurgPrimeSite değerini gurupla
   int rxsumSurgPrSite=ctoint(str[147]);
   if (rxsumSurgPrSite==1) str[148]='0';
   if ((rxsumSurgPrSite>1) && (rxsumSurgPrSite<9))
     {  str[147]='2';
        str[148]='0';   }
```

```
  // RxSumm_ScopeRegLNSur değeri yoksa değerlerine 9 ata. /*bilinmiyor */
   if (str[149]==' ') { str[149]='9';}
   // RxSumm_SurgOtherReg değeri yoksa değerlerine 9 ata. /*bilinmiyor */
   if (str[150]==' ') { str[150]='9';}

   // Sınıfı Belirle
   if (str[254]=='1')  /* hayatta mı? */
      {    int duration = (ctoint(str[240])*10 + ctoint(str[241]))*12 + (ctoint(str[242])
*10 + ctoint(str[243]));
      if (duration >60) myClass= '1'; /*  60 aydan fazla ise kurtuldu olarak sınıfla */
      }
   else /* hayatta değil */
      {    /* Eğer COD- Cause Of Death 2 ya da 3 ile başlıyorsa kanser sebepli ölüm */
          if (str[244]=='2' || str[244]=='3') myClass='2';
       }
   return 1;
 }
```

EK 2. (ConvertSeerDataToArff programı kaynak kodu) C programlama dili ile yazılmıştır. Bu program parametre olarak girilen SEER veri kaynağı dosyasını arff formatına çevirerek out1.arff adında bir çıktı oluşturur. Bu dosya Weka uygulamasında analiz edilmek amacıyla oluşturulur. Program, komut satırından veri kaynağı dosyasının adı girilerek çalıştırılır.

```
/*-------------------------------------------------------------------------------------------*/
/*  ConvertSeerDataToArff  - converts SEER data file to Arff format                  */
/*  implemented by Cengiz Coşkun .                                                   */
/*  uses “DataTypeDefinitions.h” library                                             */
/*-------------------------------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <DataTypeDefinitions.h>

extern char myClass;
int totalRecords   = 0;
int totalIncluded  = 0;

int main(int argc, char *argv[])
{ FILE *fp, *wp; char *myStr;
  if  (argc==1){
      printf("\nHatalı kullanım: Dosya adı giriniz.");
      return -1;
```

```
    }

if (!(fp=fopen(argv[1],"r"))) {
                printf("\ndosya bulunamadı(%s)",argv[0]);
                return -2;
                }

if (!(wp=fopen("out1.arff","w"))){
                printf("\nCikti dosyasi olusturulamadi");
                return -3;
                  }
int i;

fprintf(wp,"@relation SeerDataByCengizCoskun\n");
for (i=0; i<numfields;i++)
if (recs[i].included)
if (recs[i].type==snominal)
        fprintf(wp,"@attribute %s %s \n",recs[i].ad,recs[i].values);
else fprintf(wp,"@attribute %s %s \n",recs[i].ad, dataTypeName[recs[i].type]);

fprintf(wp,"@attribute myClass {1,2,3,4}\n");
fprintf(wp,"@data\n");

myStr=(malloc(266 * sizeof(char) ));
```

```
  while (fgets(myStr,266,fp)!=NULL){
        prepareData(myStr);
        ++totalRecords ;
        if (myClass!='9')
        {
        for (i=0; i<numfields;i++)
        if (recs[i].included) {
           if ( (recs[i].type==sstring))
              fprintf(wp,"\"%s\",",substrf(myStr,recs[i].bas,recs[i].len));
           else
              fprintf(wp,"%s,",substrf(myStr,recs[i].bas,recs[i].len));
        }
        fprintf(wp,"%c\n",myClass);
        ++totalIncluded;
        }
      }
  fclose(wp);
  fclose(fp);
  printf ("Okunan Kayıt Sayısı = %d \n",totalRecords);
  printf ("Yazılan Kayıt Sayısı = %d \n",totalIncluded);
  system("PAUSE");
  return 0;
}
```

ÖZGEÇMİŞ

Adı Soyadı : Cengiz COŞKUN

Doğum Yeri : Sivas

Doğum Tarihi : 22.05.1976

Medeni Hali : Evli

Yabancı Dil : İngilizce

Eğitim Durumu (Kurum ve Yılı)

Lise	:	Diyarbakır Cumhuriyet Fen Lisesi	1991-1994
Lisans	:	Orta Doğu Teknik Üniversitesi – Bilgisayar Mühendisliği	1994-1998

Çalıştığı Kurum/Kurumlar ve Yıl:

ASELSAN	1997-1998 Yarı Zamanlı
Koçbank A.Ş.	1998-2001
Demir Halk Bank Nederland NV	2001-2006
SSS Yazılım	2006-2007
HSBC Bank A.Ş.	2007-2008
Dicle Üniversitesi	2008- …

Yayınları (SCI ve diğer):

Coşkun C. , Baykal A. ,"Dicle Üniversitesi Web Etkinlik Yönetim Sistemi", Akademik Bilişim 2009,sayfa:57 ,11-13 Şubat 2009, Şanlıurfa

Baykal A. , Coşkun C. ,"Web Madenciliği Teknikleri" Akademik Bilişim 2009,sayfa:59 ,11-13 Şubat 2009, Şanlıurfa

Baykal A., Coşkun C., "Dicle Üniversitesi Bilgi İşlem Online Talep Takip Sistemi" Akademik Bilişim 2010, 10-12 Şubat 2010, Muğla