T.C.
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

MUTABAKAT FONKSİYONU KULLANARAK İŞ AKIŞLARI OPTİMİZASYONU

HACI MEHMET YILDIRIM KOÇDAĞ

Doktora Tezi

Tez Danışmanı
Yrd. Doç. Dr. Turgay Tugay Bilgin

İSTANBUL – 2013

T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

MUTABAKAT FONKSİYONU KULLANARAK İŞ AKIŞLARI OPTİMİZASYONU

DOKTORA TEZİ

HACI MEHMET YILDIRIM KOÇDAĞ

Tez Danışmanı
Yrd. Doç. Dr. Turgay Tugay Bilgin

İSTANBUL – 2013

ÖZET

Doktora Tezi, Mutabakat Fonksiyonu Kullanarak İş Akışları Optimizasyonu, Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu tez çalışmasında iş akışlarını karar ağaçları yardımıyla optimize edebilecek bir mutabakat fonksiyonu geliştirilmesi amaçlanmıştır. Geliştirilecek mutabakat fonksiyonunun hali hazırda bulunan karar ağaçları ile elde edilebilecek doğruluk değerlerinden daha yüksek doğruluk değeri bulması hedeflenmiştir.

Çalışma kapsamında kullanıcı verisinin, veri madenciliğine uygun hale getirilebilmesi için temizlenmesi, ardından bu veri üzerinden farklı boyutta örnek veriler seçilerek karar ağaçları oluşturulması planlanmıştır. Farklı algoritma ve veri büyüklükleri ile oluşturulan karar ağaçları bir mutabakat fonksiyonuna girdi olarak verilmiştir. Geliştirilen mutabakat fonksiyonu ile elde edilen tahmin doğruluğunun tüm karar ağaçları içerisindeki en yüksek doğruluk oranına sahip algoritmadan daha yüksek doğruluk oranına sahip olması hedeflenmiştir.

Tez yedi bölümden oluşmaktadır. Birinci bölümde tez hakkında genel bilgiler verilerek konuya giriş yapılmıştır. İki, üç ve dördüncü bölümlerde sırasıyla karar ağaçları, regresyon analizi ve mutabakat fonksiyonu hakkında genel bilgiler verilmiştir. Beşinci bölümde uygulama ortamı tasarımı ve kullanılan veri setlerinden bahsedilmiştir. Altıncı bölümde karar ağaçları ile yapılan uygulamanın sonuçları tablolar ile açıklanmıştır. Sonuç ve referanslar bölümleri ile tez tamamlanmıştır.

Bu tez 2013 yılında tamamlanmıştır ve 70 sayfadan oluşmaktadır.

Anahtar Kelimeler: Veri Madenciliği, Karar Ağaçları, İş Akışları, Mutabakat Fonksiyonu

ABSTRACT

Doctorate Thesis, Workflow Optimization via Consensus Function, Maltepe University, Institute of Science, Computer Engineering Department.

The objective of this thesis is to prepare a consensus function that can optimize workflow via decision trees. It is aimed to create a consensus function that will have a higher degree of accuracy in comparison with the current decision tree algorithms.

The study is planned to cover data cleaning for mining purposes and the preparation of decision trees by way of selecting various samples. Decision trees generated by various algorithms were assembled using a consensus function. By means of that consensus function, it is aimed to obtain greater accuracy value than the accuracies of the individual decision tree algorithms.

The thesis consists of seven sections. The first section provides an introduction by giving general information about the thesis. General information regarding decision trees, regression analysis and consensus function are given in sections two, three and four respectively. The design of the test platform and the detailed descriptions of the data sets used are covered in the fifth section. Results of the experiments have been given in section six. The thesis is finalized with conclusion and references.

This thesis has been completed in 2013 and consists of 70 pages.

Keywords: Data Mining, Decision Trees, Work Flows, Consensus Function

TEŞEKKÜR

Tez konusunu seçmemde beni yönlendiren, tez süreci boyunca destek ve yardımlarını esirgemeyen, değerli bilgilerinden istifade ettiğim danışman hocam Yrd. Doç. Dr. Turgay Tugay BİLGİN'e, maddi ve manevi desteğini benden hiçbir zaman esirgemeyen çok değerli aileme ve çalışmalarım sırasında emeği geçen herkese teşekkürlerimi sunarım.

İÇİNDEKİLER

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
CHAID	Chi-Squared Automatic Interaction Detector	Ki-Kareli Otomatik Etkileşim Dedektörü
CART	Classification & Regression Trees	Sınıflandırma ve Regresyon Ağaçları
MARS	Multivariate Adaptive Regression Splines	Çoklu Değişkenli Adaptif Regresyon Çubukları
QUEST	Quick, Unbiased, Efficient Statistical Tree	Çabuk, Önyargısız, Etkin İstatistiksel Ağaç
SLIQ	Supervised Learning in Quest	QUEST'te Kontrollü Öğrenme
SPRINT	Scalable Parallelizable Induction of Decision Trees	Karar Ağaçlarının Ölçeklendirilebilir Paralel İndüksiyonu

ŞEKİLLER

ÇİZELGELER

1. BÖLÜM : Giriş ve Temel Kavramlar

1.1. Giriş

Bu tez çalışmasında iş akışlarındaki kullanıcı verisi dikkate alınarak akışın en iyileştirilmesi üzerine bir çalışma yapılmıştır. Çalışma sırasında bir Telekom firmasındaki eğitim seyahat akışındaki gerçek kullanıcı verisi maskelenerek kullanılmıştır. İş akışının sonucuna karar vermek için karar ağaçları ve bu karar ağaçlarından en iyi mutabakat fonksiyonu oluşturulmaya çalışılmıştır. Farklı boyutta veri seti örnekleri alınarak CART, C4.5, ID3 ve CHAID algoritmaları ile sınıflandırma kuralları belirlenmiştir. Oluşturulan sınıflandırma kurallarından yararlanılarak karar ağaçları üretilmiştir ve tüm veri setine uygulanarak her bir karar ağacı için kesinlik düzeyleri çıkartılmıştır.

Oluşturulan karar ağaçları ile kesinlik düzeyleri dikkate alınarak en iyi mutabakat fonksiyonu oluşturulmaya çalışılmıştır. Mutabakat fonksiyonu kullanılarak elde edilen kesinlik düzeyinin, karar ağaçlarının tek başına uygulanması ile elde edilen kesinlik düzeyinden daha yüksek olması hedeflenmiştir.

Tez çalışmasında elde edilen sonuçlar grafikler ve tablolar ile gösterilmiştir. Farklı ağaçların sonuçları karşılaştırmalı olarak gösterilmiştir. Mutabakat fonksiyonu farklı veri setleri üzerinde test edilmiş ve iş akışı türünden veri setlerinde en iyi sonuç verebilecek genel bir mutabakat fonksiyonu önerilmiştir.

Uygulama sırasında karar ağacı algoritmalarının çalıştırılabilmesi için Rapid Miner adlı veri madenciliği yazılımı ve buna ait Whibo eklentisi kullanılmıştır. Oluşturulan karar ağaçlarının tüm veri setine uygulanması için Tahminci (Forecaster) adını verdiğimiz bir yazılım hazırlanmıştır. Ayrıca mutabakat fonksiyonunun oluşturulması ve test edilmesi için Mutabakat Fonksiyonu Uygulaması hazırlanmıştır.

1.2. Veri Madenciliği

Her geçen gün üretilen veri miktarı artmaktadır. Sürekli artan disk kapasiteleri bu bilginin depolanmasını kolaylaştırmakla birlikte artan bu verinin analiz edilmesi, değerlendirilmesi ve karar verilmesi daha da zorlaşmaktadır. Bu noktada veri madenciliğinin önemi ön plana çıkmakta ve artmaktadır.

Bu bölümde veri madenciliğinin tanımı, uygulama alanları, yaşam döngüsü ve yöntemleri detaylı olarak incelenmiştir.

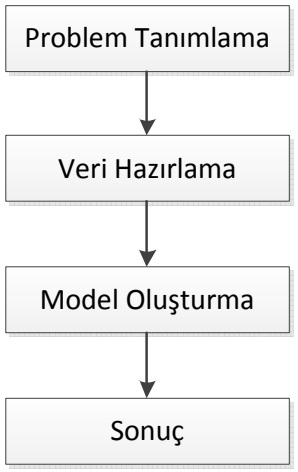
1.2.1. Veri Madenciliği Tanımı

Veri madenciliği, büyük miktarda veri içinden, gelecekle ilgili tahmin yapılmasını sağlayacak bağıntı ve kuralların aranmasıdır [1]. Veri madenciliği ile büyük miktarda veriden yararlı bilgi, desenler, ve eğilimlerin (genelde önceden belli olmayan) çıkarılabilmesi amaçlanır [2]. Geleneksel yöntemler kullanılarak çözülmesi çok zaman olan problemlere veri madenciliği süreci kullanılarak daha hızlı bir şekilde çözüm bulunabilir [3].

Veri madenciliğinin bir başka tanımı ise; verinin sahibine anlamlı ve yararlı olacak şekilde veri kümesinin içinde anlamlı ilişkileri bulmak ve veriyi yeni bir şekilde özetlemek için veri kümelerinin incelenmesidir [4, 5].

1.2.2. Veri Madenciliği Süreci

Veriden bilgiye ulaşma süresince yapılan çalışmaların tümüne veri madenciliği süreci denir. Şekil 1.1'de görüldüğü üzere, veri madenciliği süreci sırasıyla problem tanımlama, veri hazırlama, model oluşturma ve sonuç adımlarından oluşmaktadır.

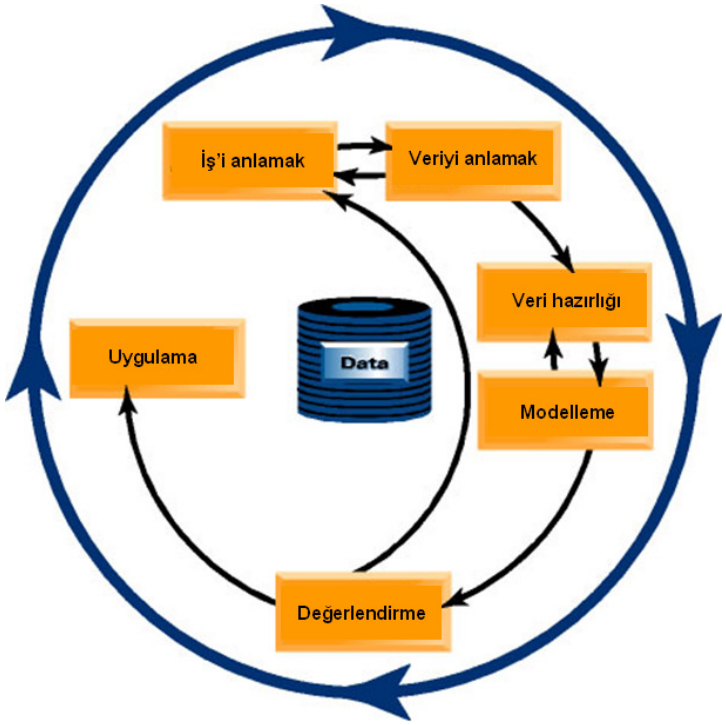


Şekil 1.1 Veri Madenciliği Süreci

Problem tanımlama adımında öncelikli olarak problem incelenir ve genel tanımı yapılır. Veri hazırlama adımında öncelikle veri toplanır, daha sonra bu veriler temizlenir ve sınıflandırılır. Model oluşturma adımında problem çözümü için modeller hazırlanır ve test edilir. Başarılı olan model veya modeller probleme uygulanır. Sonuç adımında, uygulanan model veya modeller işletildiğinde ortaya çıkan sonuç yorumlanır, raporlanır ve bir sonraki problemin tanımına girdi olabilecek şekilde saklanır.

Bazı kaynaklarda veri madenciliğinin aslında bir süreç olmadığı aynı zamanda bir yaşam döngüsü olduğu ifade edilmektedir [6]. Her bir problem çözümü

bir sonraki veri madenciliğinin problem tanımı olarak ele alınırsa bu bilginin yanlış olmadığı gözlemlenir. Şekil 1.2'de Veri Madenciliği Yaşam Döngüsü gösterilmiştir [6].



Şekil 1.2 Veri Madenciliği Yaşam Döngüsü

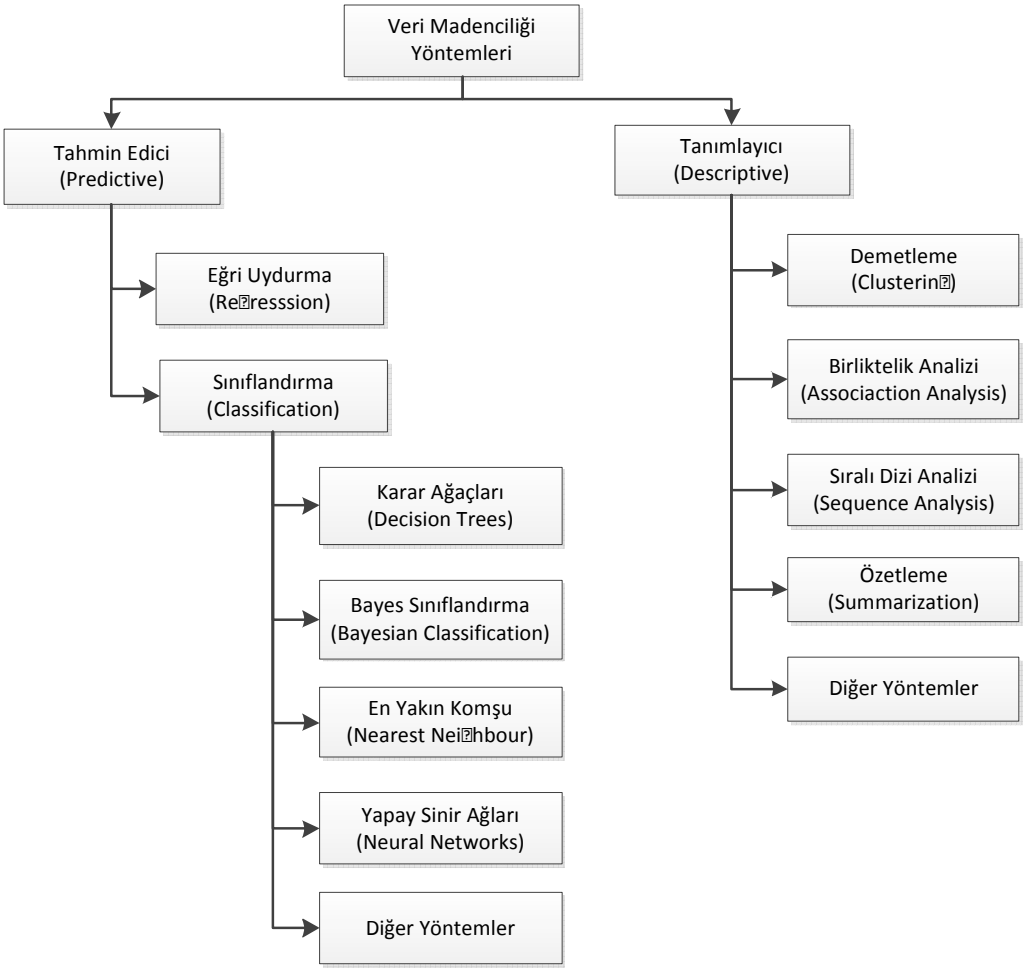
Veri madenciliği yaşam döngüsü 6 aşamadan oluşur. Her aşamada o aşamanın bir sonraki aşamasının ne olduğu gösterilir. Veri madenciliği yaşam döngüsünde elde edilen çıkarımlar, bir sonraki aşamada kullanılır. Aşağıda veri madenciliği yaşam döngüsü aşamalarının kısa açıklamaları verilmiştir:

- **İşi Anlamak:** Bu aşamada problem kısıtları, sonuçları ile birlikte anlaşılmaya çalışılır.
- **Veriyi Anlamak:** Toplanan veri incelenir.
- **Veri Hazırlığı:** İncelenen veri üzerinde temizlik yapılır. Veri madenciliğine uygun hale getirilir.

- **Modelleme:** Problemin çözümü için veri madenciliği teknikleri kullanılarak modelleme yapılır.
- **Değerleme:** Uygulama aşamasına geçmeden önce çözüm modeli baştan aşağı kontrol edilir ve tüm kısıtların dikkate alınıp alınmadığı değerlendirilir.
- **Uygulama:** Oluşturulan model ve veri ile uygulama hazırlanır.

1.2.3. Veri Madenciliği Yöntemleri

Veri madenciliği yöntemleri temelde iki ana başlıkta incelenmektedir. Bunlar Tahmin Edici (Predictive) ve Tanımlayıcıdır (Descriptive) yöntemlerdir [7]. Şekil 1.3.'de veri madenciliği yöntemleri detaylı olarak gösterilmiştir.



Şekil 1.3 Veri Madenciliği Yöntemleri

1.2.3.1. Tahmin Edici (Predictive) Yöntemler

Tahmin edici yöntemler sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesidir [7]. Tahmin edici yöntemler eğri uydurma ve sınıflandırma olarak ikiye ayrılır. Karar ağaçları yöntemi sınıflandırma yöntemlerinden biridir.

Örneğin bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır.

1.2.3.2. Tanımlayıcı (Descriptive) Yöntemler

Tanımlayıcı yöntemlerde amaç, büyük veri kümelerindeki desen ve ilişkileri tespit ederek, incelenen sistemin anlamını kavramaktır [7]. Demetleme ve birliktelik analizi yöntemleri, tanımlayıcı yöntemlerden bir kaçıdır. Örneğin 25 yaş altı bekar kişiler ile 25 yaş üstü evli kişiler üzerinde yapılan ve ödeme performanslarını gösteren bir analiz tanımlayıcı modellere örnek olarak verilebilir.

1.3. İş Akışları Madenciliği

İş akışları madenciliğinde, iş akışları ile alınan veri setleri yorumlanarak iş akışlarının nasıl daha efektif hale getirilebileceği incelenmektedir. Bu bölümde iş akışları madenciliği konusunda temel kavramlar, literatür incelemesi, problemin tanımı ve tezin amacı açıklanmıştır.

1.3.1. İş Akışları Madenciliği Kavramı

İş akışları madenciliği, iş akışlarının geçmiş çıktılarını kullanarak iş akışının nasıl daha iyi hale getirebileceğini bulmaya çalışmaktır. Bulunan sonuçlar iş akışını

besleyerek akışın adımlarını değiştirebilir veya akışa çeşitli fonksiyonlar ekleyebilmek için kullanılabilir. Birçok kurumsal firmada iş akışlarının kullanılması ile birlikte özellikle 1990'lı yılların sonlarında iş akışlarının nasıl daha iyi hale getirilebileceği bir problem halini almıştır. Bu konuda öncelikle akışlardaki adımlarda geçen süreler ve kayıtlar dikkate alınarak bazı çözümler üretilmiştir [8, 9]. Ardından yapılan araştırmalar ile veri madenciliği kullanılarak iş akışı kullanıcı veri setleri ile çeşitli fonksiyonlar çıkartılmaya çalışılmıştır [10].

1.3.2. Literatür İncelemesi

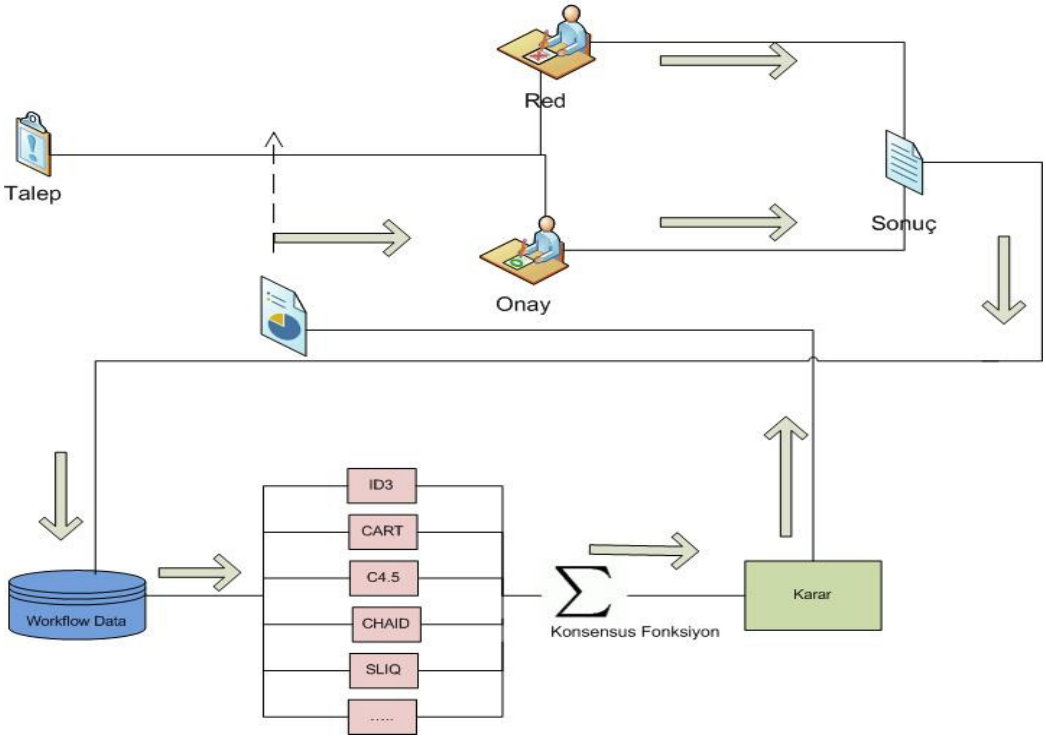
Süreç madenciliği üzerine yapılan ilk çalışma 1998 yılında Cook ve Wolf tarafından gerçekleştirilmiştir [8]. Süreç madenciliği ile iş akışları madenciliğini bağlayan ilk çalışma ise 1998 yılında Agrawal, Gunopulos ve Leymann tarafından hazırlanmıştır [9]. Bu konuda Herbst ve Karagiannis detaylı bir çalışma yapmış ve yapılan bu ilk detaylı çalışma 1999 yılında yayınlanmıştır [10]. İş akışlarının modellenmesinde veri madenciliğinin önemi, 2001 yılında Weijters ve Van Der'in hazırladığı çalışma ile ortaya çıkarılmıştır [11]. 2005 Yılında doküman yönetimi bazlı bilgilerin, iş akışları faaliyet kayıtları (activity log) olarak kullanılması konusunda detaylı bir çalışma yapılmıştır [12]. 2005 Yılındaki bir başka çalışma Stanislaw tarafından hazırlanan, gelişim seviyesinin karar ağaçları ile tahmini üzerinedir [13]. 2006 Yılında ise ev fiyatlarının tahmini üzerine bir çalışma hazırlanmıştır [14]. Aynı yılda iş akışlarının tasarımı yapılırken faaliyet kayıtlarından nasıl yararlanılabileceğini gösteren bir çalışma gerçekleştirilmiştir [15]. İş akışlarında karar ağaçlarının kullanımı konusundaki ilk çalışma 2007 yılında yapılmış olup, karar ağaçlarının iş akışlarını ne kadar etkin hale getirebileceğini ortaya koymuştur [16]. Karar ağaçları ile öğrenci performanslarını tahmin etme amaçlı çalışma 2009 yılında Ning Fang be Jingui Lu tarafından hazırlanmıştır [17]. Aynı yılda yapılan bir başka çalışma, işin azalma riskinin karar ağaçları ile tahmin edilmesi konusunda Vineet Kumar tarafından yapılmıştır [18]. Konuya ışık tutacak bir başka çalışma, uzman sistemler ve karar ağaçları hakkında 2010 yılında yapılmıştır [19].

1.3.3. Problemin Tanımı ve Tezin Amacı

Bu tez çalışmasında iş akışlarındaki onay/ret kararının hızlı alınması (Otomatikleştirilmesi veya kullanıcıya öneri oluşturulması) için karar ağaçları kullanılarak bir mutabakat fonksiyonunun oluşturulması hedeflenmektedir. Oluşturulacak mutabakat fonksiyonu diğer veri setleri ile test edilerek genel amaçlı bir mutabakat fonksiyonu oluşturulmaya çalışılacaktır. Mutabakat fonksiyonunun karar ağacı algoritmalarının tek başına ürettiği kesinlik düzeyinden daha iyi bir kesinlik düzeyine ulaşması hedeflenmektedir.

Yapılacak çalışma sırasıyla aşağıda açıklanmıştır:

- İş akışları verisi, veri madenciliği ve karar ağaçları kullanılabilecek uygun yapıya getirilecektir.
- Karar ağaçları ile bu veri seti kullanılarak tahmin edici ağaçların oluşturulması sağlanacaktır.
- Oluşturulan karar ağaçları tüm veri seti üzerinde test edilecektir.
- Her bir karar ağacı için doğruluk oranları hesaplanacaktır.
- Oluşturulan ağaçlar birbirleri arasında karşılaştırılacaktır.
- Oluşturulan ağaçların bir kısmı veya tümü alınarak bir mutabakat fonksiyonu oluşturulacaktır.
- Bu mutabakat fonksiyonu ile tüm veri seti için sonuçlar tekrar oluşturulacak ve kesinlik düzeyleri incelenecektir.
- Mutabakat fonksiyonu için hesaplanan kesinlik düzeyinin kullanılan tüm karar ağacı algoritmalarının kesinlik düzeyinden daha yüksek olması hedeflenmektedir.
- Mutabakat fonksiyonu farklı veri setlerine uygulanarak etkinliği deneyler ile ölçümlenecektir.



Şekil 1.4 İş Akışı İçin Planlanan Çözüm

Şekil 1.4.'de Seyahat akışı ve mutabakat fonksiyonunun birlikte oluşacağı yapı gösterilmiştir. Kısaca açıklamak gerekirse, hali hazırda kullanılan iş akışı ile girilen akış talebi bir sonraki adımda onay veya red aksiyonları ile sonuçlandırılır. Bu sonuçlar ve talep bilgileri veri tabanında depolanır. Sistemin otomatikleştirilmesi veya kullanıcıya bir öneri sunulması amacı ile oluşturulacak çözüm için biriktirilen toplu veri bir sonraki akışlar için karar noktasında kullanılacaktır. Veri uygun büyüklüğe ulaştıktan sonra karar ağaçları oluşturulur. Karar ağaçları farklı algoritmalar kullanılarak hazırlanır. Hazırlanan karar ağaçlarının kesinlik düzeyleri hesaplanır. Kesinlik düzeyleri dikkate alınarak mutabakat fonksiyonu oluşturulur.

Mutabakat fonksiyonu bir sonraki aşamada girilen talepler için onaya gitmeden işleme alınır. Yeni talepler akışın sonuçlanması beklenmeden, sonuç mutabakat fonksiyonu ile tahmin edilir. Tahmin edilen durum, onay/red adımındaki kullanıcıya iletilir veya sistem bu sonucu kullanarak iş akışını otomatik onaylayarak

veya reddederek kapatır. Sistem kendisini geliştiren bir yapı ile çalışır, yeni bilgiler geldikçe belirli aralıklar ile sistem kendini yeniler, tekrardan mutabakat fonksiyonu oluşturur. Sistemin kendini yenilemesi, artan veriye veya önerilen karar ile kullanıcının verdiği kararın belirli bir yüzde ile başarısız olması durumunda tetiklenir.

2. BÖLÜM : İş Akışı Madenciliğinde Kullanılan Yöntemler ve Kalite Ölçümü

2.1. Karar Ağaçları

Karar ağaçları (decision trees), bir tahmin tekniğidir. Genelde sınıflandırma, kümeleme, tahmin modellerinde ve sorunla ilgili araştırma alanını alt gruplara ayırma için kullanılır [20].

Karar ağaçları ile veri oluşturulduktan sonra kökten yaprağa doğru inilerek kurallar (Eğer-O zaman kuralları) (if-then rules) yazılabilir. Karar ağaçlarında kök ve her düğüm bir soruyla etiketlenir. Düğümlerden ayrılan dallar ise ilgili sorunun olası yanıtlarını belirtir. Her dal düğümü de söz konusu sorunun çözümüne yönelik bir tahmini temsil eder. Bunun gibi kural çıkarma, veri madenciliği çalışmasının sonucunu doğrulamak için kullanılır. Bu kurallar uygulama konusunda uzman bir kişiye gösterilerek sonucun anlamlı olup olmadığı denetlenebilir [20].

Bir karar ağacı modeli, üç bölümden oluşur:

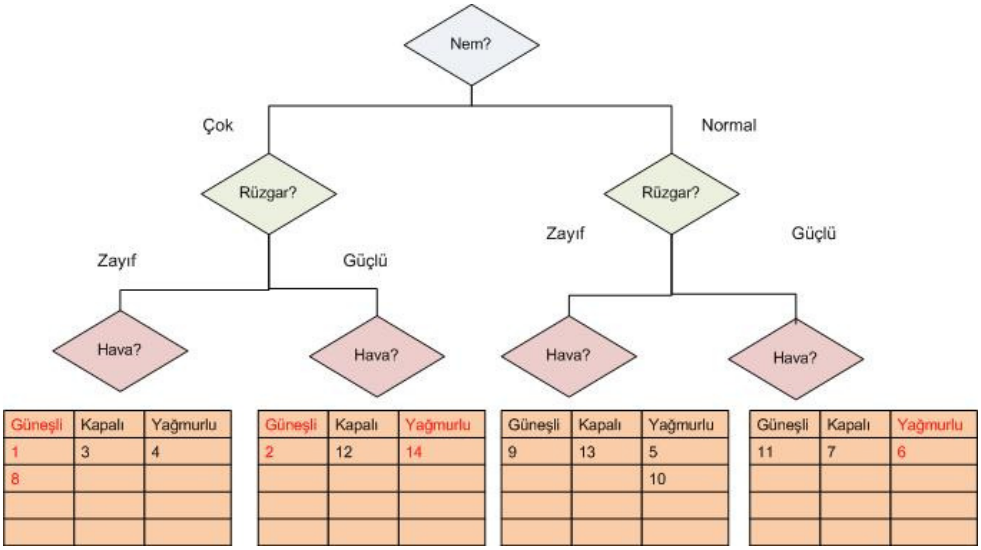
i. Karar ağacı,
ii. Ağacı oluşturacak bir algoritma,
iii. Ağacı veriye uygulayacak ve söz konusu sorunu çözecek bir algoritma,

Örneğin Çizelge 2.1'de bulunan veriler bir karar ağacı algoritması ile Şekil 2.1'de görülen şekilde bir karar ağacına dönüştürülebilir. Burada dikkat edilmesi gereken nokta hedef sütunun belirlenmesi ve hedef sütuna ulaşım için kullanılacak sütunların seçilmesidir.

Çizelge 2.1 Örnek Karar Ağacı Verisi

Gün	**Hava**	**Sıcaklık**	**Nem**	**Rüzgar**	**Futbol Oynanır mı?**
1	Güneşli	Sıcak	Yüksek	Zayıf	Hayır
2	Güneşli	Sıcak	Yüksek	Güçlü	Hayır
3	Kapalı	Sıcak	Yüksek	Zayıf	Evet
4	Yağmurlu	Normal	Yüksek	Zayıf	Evet
5	Yağmurlu	Serin	Normal	Zayıf	Evet
6	Yağmurlu	Serin	Normal	Güçlü	Hayır
7	Kapalı	Serin	Normal	Güçlü	Evet
8	Güneşli	Normal	Yüksek	Zayıf	Hayır
9	Güneşli	Serin	Normal	Zayıf	Evet
10	Yağmurlu	Normal	Normal	Zayıf	Evet
11	Güneşli	Normal	Normal	Güçlü	Evet
12	Kapalı	Normal	Yüksek	Güçlü	Evet
13	Kapalı	Sıcak	Normal	Zayıf	Evet
14	Yağmurlu	Normal	Yüksek	Güçlü	Hayır

Şekil 2.1'de görüldüğü üzere havanın durum bilgisi hedef alınarak bir ağaç oluşturulmuştur [20]. Oluşturulan ağaçta nem ve rüzgar sonuca ulaşmak için sorulan soruların sütunlarıdır.



Şekil 2.1 Örnek Karar Ağacı

Literatürde kullanılan başlıca karar ağacı algoritmaları şunlardır;

- CHAID (Chi-Squared Automatic Interaction Detector),
- CART (Classification and Regression Trees),
- ID3,
- Exhaustive CHAID,
- C4.5,
- MARS (Multivariate Adaptive Regression Splines),
- QUEST (Quick, Unbiased, Efficient Statistical Tree),
- C5.0,
- SLIQ (Supervised Learning in Quest),
- SPRINT (Scalable Parallelizable Induction of Decision Trees).

2.1.1. ID3 Algoritması

ID3 algoritması J. Ross Quinlan tarafından 1986 yılında geliştirilmiştir. Veri tabanında çok nitelik varsa ve eğitim kümesi çok fazla kayıt içeriyorsa, bunun yanında az hesaplama yaparak makul bir ağaç oluşturulmak isteniyorsa ID3 algoritması kullanılabilir [20]. ID3 algoritmasının çalışma şekli şöyledir;

- C bir eğitim kümesi olmak üzere, eğer C'deki bütün kayıtlar aynı sınıf üyesi iseler, sınıfın adında bir düğüm oluşturulur ve algoritma sonlanır, değilse bir test niteliği seçilerek karar düğümü oluşturulur.
- C kümesi, karar düğümüne göre alt kümelere ayrılır: C_1, C_2,..,C_n.
- Algoritma her bir C_i kümesine özyinelemeli bir şekilde uygulanır.

2.1.2. C4.5 Algoritması

ID3 algoritması yine aynı kişi tarafından genişletilerek C4.5 adını almıştır. ID3 algoritmasında bir özellik sayısal değerlere sahip ise sonuç alınamamaktadır. Bu yüzden C4.5 algoritması geliştirilmiştir [21]. C4.5 algoritmasının çalışma şekli şöyledir;

- Sayısal değerler ile çalışılırken bir eşik değeri belirlenir. Bu eşik değeri bulunurken özelliğin değerleri sıralanır ve $[v_i,v_i+1]$ aralığının orta noktası alınır, bu değer t eşik değeri olarak belirlenir.
- Özellik değeri bu t eşik değerinden büyük veya küçük eşit olmak üzere ikiye ayrılır.

2.1.3. CART (Classification and Regression Trees) Algoritması

CART, Leo Breiman, Jerome Friedman, Charles J. Stone ve Richard A. Olshen tarafından 1984'de geliştirilmiştir [22]. CART algoritmasının özellikleri şöyledir;

- Her aşamada ilgili kümenin, kendinden daha homojen olan iki alt kümeye ayrılması sağlanmaktadır.
- Ayrım işlemi kategorik bağımlı değişkenler için GINI, TWOING, sürekli değişkenler için en küçük kareler sapması (Least-Squared Deviation) indeks hesaplamalarına göre yapılmaktadır.
- Bu hesaplamalarda kar, maliyet değerleri ve değişken kategorileri arasındaki önceliklerin tanımlanabilmesi gibi sağlanan çeşitli esneklikler, CART algoritmasının günümüzde de yoğun olarak tercih edilmesine neden olmaktadır.

2.1.4. CHAID (Chi-Square Automatic Interaction Detector) Algoritması

CHAID, CART algoritmasına benzemektedir. 1980 yılında Gordon tarafından geliştirilmiştir [23]. CHAID algoritmasının özellikleri şöyledir;

- CART algoritması ile arasındaki en büyük fark veriyi bölümlere ayırırken farklı bir yol kullanmasıdır. Optimum bölümleri seçmek için kullanılan ENTROPY veya GINI metrikleri yerine ki kare testi uygulayan bir teknik kullanılır.
- Kategorik ve sürekli değişkenler üzerinde çalışabilmesi, ağaçta her düğümü ikiden fazla alt gruba ayırabilmesi gibi nedenlerle günümüzde de tercih edilen bir algoritmadır.

2.1.5. SLIQ (Supervised Learning In Quest) Algoritması

IBM Quest 1996 yılında diğer algoritmalardaki (QUEST) hafıza problemi üzerine geliştirilmiştir [24]. SLIQ algoritmasının özellikleri şöyledir;

- Büyük veri kümelerini bölümlere ayırarak karar ağacı oluştururken ön-sıralama tekniğini kullanır. Bu teknik her düğümdeki sıralama masrafını büyük ölçüde önlemiş olur.
- SLIQ her düğüm için sınıf listesi olarak adlandırılan ayrık sıralanmış bir liste tutar. Bu listedeki her eleman verideki niteliklere karşılık gelmektedir ve bir sınıf etiketine sahiptir.
- SLIQ, karar ağacını oluştururken genişlik öncelikli yolu kullanır. Her nitelik için uygun sıralanmış listeyi tarar ve her değer için ENTROPY değerini hesaplar.
- Her nitelik için ENTROPY hesaplandıktan sonra veriyi bölmek için bir nitelik seçilir. Bu işlem veri sınıflara ayrılana kadar yinelemeli olarak devam eder.

2.2. Hata Matrisi

Hata matrisi, veri madenciliğinde bulunan sınıflandırma sonuçlarını değerlendirmek amacıyla kullanılmaktadır. Çizelge 2.2'de kullanılan hata matrisi gösterilmiştir.

Çizelge 2.2 Hata Matrisi

Mevcut Sınıf /Tahmin Edilen Sınıf	**C1**	**-C1**
C1	TP	FN
-C1	FP	TN

Bir sınıflandırıcının 4 tane sonuç üretme olasılığı vardır ve Çizelge 2.2'de bu sonuçların gösterimi yapılmaktadır. Eğer tahminlenen sonuç pozitif ve gerçek değerde pozitif ise doğru pozitif (*true positive - TP*) olarak adlandırılır. Ancak gerçek değer negatif ise üretilen sonuç yanlış pozitif (*false positive - FP*) olarak isimlendirilir. Tam tersi olarak tahminlenen ve gerçek değerlerin her ikiside negatif ise doğru negatif (*true negative - TN*), tahminlenenin negatif fakat gerçek değeri pozitif olduğu durumda ise üretilen sonuç yanlış negatif (*false negative - FN*) olarak adlandırılır [25].

TP gerçek ve tahmin sonuçlarının tümünde pozitif olarak bulunan kümeyi temsil etmektedir. TN ise gerçek ve tahmin sonuçlarının tümünde negatif olarak bulunan kümeyi temsil etmektedir. Sistemin kesinlik düzeyi (accuracy) hesaplanırken, toplam doğru bulunan sonuç sayısı toplam sayıya bölünür. Çıkan sonuç tahmin algoritmasının kesinlik düzeyini göstermektedir. Bu oran 1'e ne kadar yakın olursa tahmin sisteminin başarısı o kadar büyük olur.

$$\textbf{\textit{Kesinlik Düzeyi = (TP + TN) / Toplam}} \quad (1)$$

Hata düzeyi (error rate) hesaplanırken kesinlik düzeyi 1'den çıkarılır. Bu oran ne kadar yüksek çıkarsa tahmin sisteminin başarısı o kadar düşüktür.

$$\textbf{\textit{Hata Düzeyi = 1- Kesinlik Düzeyi}} \quad (2)$$

Çizelge 2.3 Seyahat Akışı Veri Seti İçin Hata Matrisi

Gerçek / Tahmin	seyahat_onay = evet	seyahat_onay = hayır	None	Toplam
seyahat_onay = evet	6934	26	40	7000
seyahat_onay = hayır	412	2500	88	3000
Toplam	7346	2526	128	10000

Çizelge 2.3'de gerçek sonuçlar ve karar ağacı ile bulunan sonuçlar karşılaştırmalı olarak gösterilmiştir. Toplam 10000 adet satırdan oluşan veri seti bulunmaktadır. Bunların 7000 tanesinin sonucu "evet" ve 3000 tanesinin sonucu ise "hayır" 'dır. Karar ağacı algoritması 6934 tane kaydı "evet" olarak etiketlemiştir. 26 kaydı ise normalde "evet" olarak etiketlemesi gerekirken "hayır" olarak etiketlemiştir. 40 kayıtta ise herhangi bir sonuç üretememiştir. Seyahat onayı "hayır" olduğu bilinen 3000 kayıttan 2500 tanesi "hayır" olarak etiketlenmiştir. 412 tanesi ise hatalı bir şekilde "evet" olarak etiketlenmiştir. 88 tanesinde ise herhangi bir sonuç elde edilememiştir. Bu durumda sistemin kesinlik düzeyi (accuracy) aşağıdaki gibi hesaplanır;

$$\textbf{\textit{Kesinlik Düzeyi}} = (2500 + 6934) / 10000$$
$$= 0{,}9434$$

Sistemin hata düzeyi ise aşağıdaki şekilde hesaplanır;

$$\textbf{\textit{Hata Düzeyi}} = 1-0{,}9434$$
$$= 0{,}0566$$

2.3. Karar Ağaçları Performans Analizleri

Bu bölümde çalışmada kullanılan karar ağaçları algoritmalarının performans analizleri tablo ve grafikler ile gösterilmiştir. Performans analizi seyahat akışı veri seti kullanılarak yapılmıştır. Yapılan çalışmada veri seti büyüklüğü gerçek veri seti içerisinden rasgele seçilerek elde edilmiştir. Oluşturulan veri seti MS-SQL veri tabanında saklanmıştır. Rapid Miner ve Whibo eklentisi yardımıyla veri setinin karar ağaçları bulunmuştur. Uygulamanın karar ağaçları oluşturma süreleri analiz edilerek performansları not edilmiştir. Kullanılan donanımın bellek ve işlemcisi yetersiz kaldığından çalışma 30.000 kayıt ile sınırlandırılmıştır. Performans ölçümleri 2.40 GHz Intel i5 işlemcisi ve 4 GB ana belleği bulunan bir bilgisayarda gerçekleştirilmiştir.

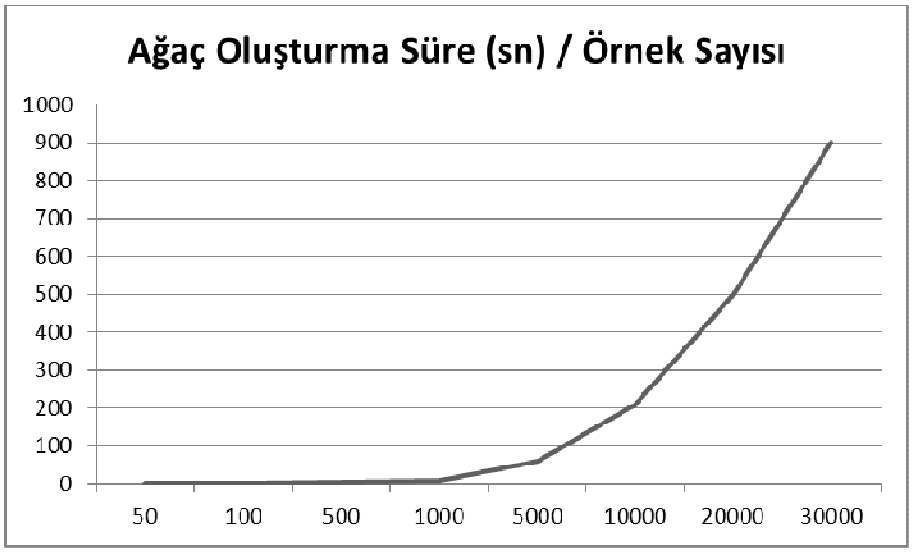
2.3.1. CHAID Algoritması Performans Analizi

CHAID algoritması kullanılarak farklı veri setlerinin performansları zaman ve örnek sayısı dikkate alınarak Çizelge 2.4.'de gösterilmiştir.

Çizelge 2.4 CHAID Algoritması Performans Analizi

Örnek Sayısı	Ağaç Oluşturma Süre (sn)
50	1
100	1
500	6
1000	10
5000	60
10000	200
20000	500
30000	900

Çizelge 2.4.'deki örnek sayıları ve ağaç oluşturma süreleri dikkate alınarak oluşturulan eğri Şekil 2.2.'de gösterilmiştir.



Şekil 2.2 CHAID Algoritması Performans Analizi Eğrisi

Şekil 2.2.'de gösterilen eğri dikkate alınarak CHAID algoritmasında örnek sayısındaki artış, ağaç oluşturma süresini üstel bir şekilde etkilemektedir.

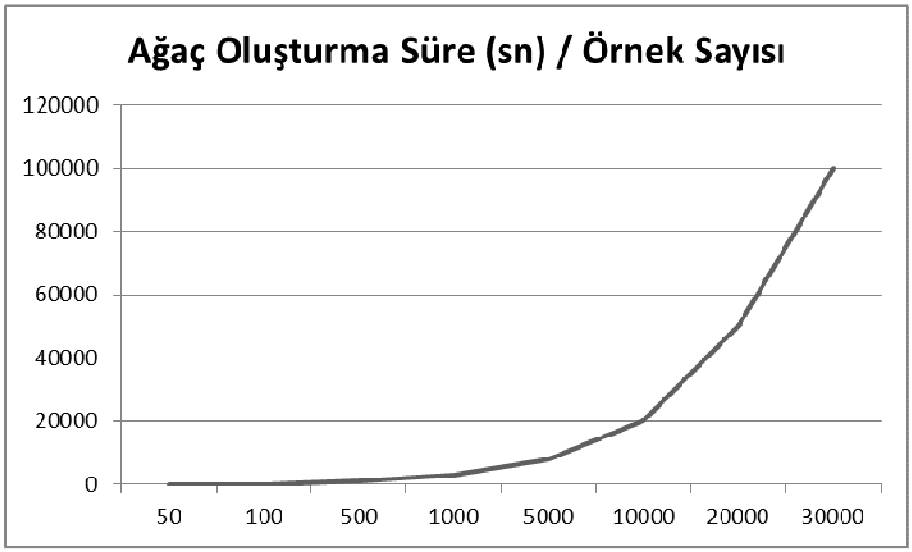
2.3.2. CART Algoritması Performans Analizi

CART algoritması kullanılarak farklı veri setlerinin performansları zaman ve örnek sayısı dikkate alınarak Çizelge 2.5.'de gösterilmiştir.

Çizelge 2.5 CART Algoritması Performans Analizi

Örnek Sayısı	Ağaç Oluşturma Süre (sn)
50	2
100	3
500	1000
1000	3000
5000	8000
10000	20000
20000	50000
30000	100000

Çizelge 2.5.'deki örnek sayıları ve ağaç oluşturma süreleri dikkate alınarak oluşturulan eğri Şekil 2.3.'de gösterilmiştir.



Şekil 2.3 CART Algoritması Performans Analizi Eğrisi

Şekil 2.3.'de gösterilen eğri dikkate alınarak CART algoritmasında örnek sayısındaki artış, ağaç oluşturma süresini üstel bir şekilde etkilemektedir.

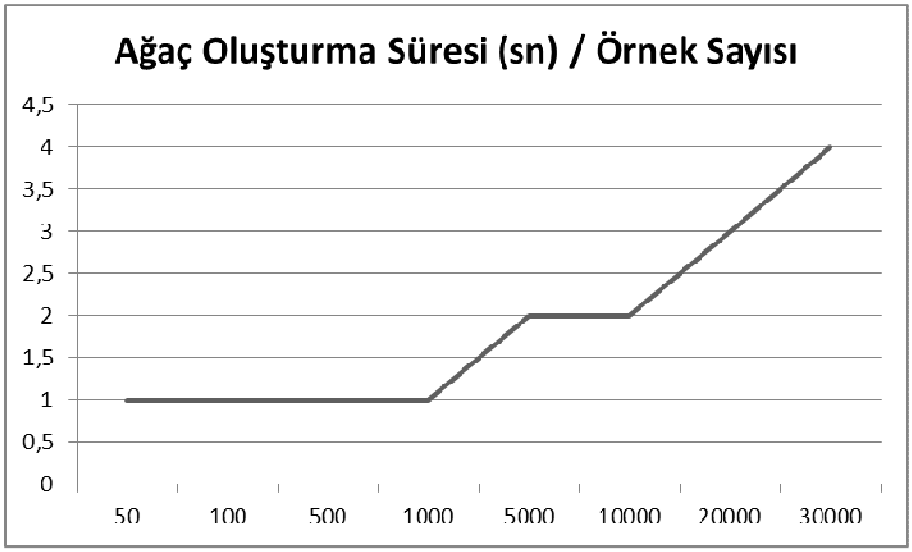
2.3.3. C4.5 Algoritması Performans Analizi

C4.5 algoritması kullanılarak farklı veri setlerinin performansları zaman ve örnek sayısı dikkate alınarak Çizelge 2.6.'de gösterilmiştir.

Çizelge 2.6 C4.5 Algoritması Performans Analizi

Örnek Sayısı	Ağaç Oluşturma Süre (sn)
50	1
100	1
500	1
1000	1
5000	2
10000	2
20000	3
30000	4

Çizelge 2.6.'deki örnek sayıları ve ağaç oluşturma süreleri dikkate alınarak oluşturulan eğri Şekil 2.4.'de gösterilmiştir.



Şekil 2.4 C4.5 Algoritması Performans Analizi Eğrisi

Şekil 2.4.'de gösterilen eğri dikkate alınarak C4.5 algoritmasında örnek sayısındaki artış, ağaç oluşturma süresini doğrusal bir şekilde etkilemektedir.

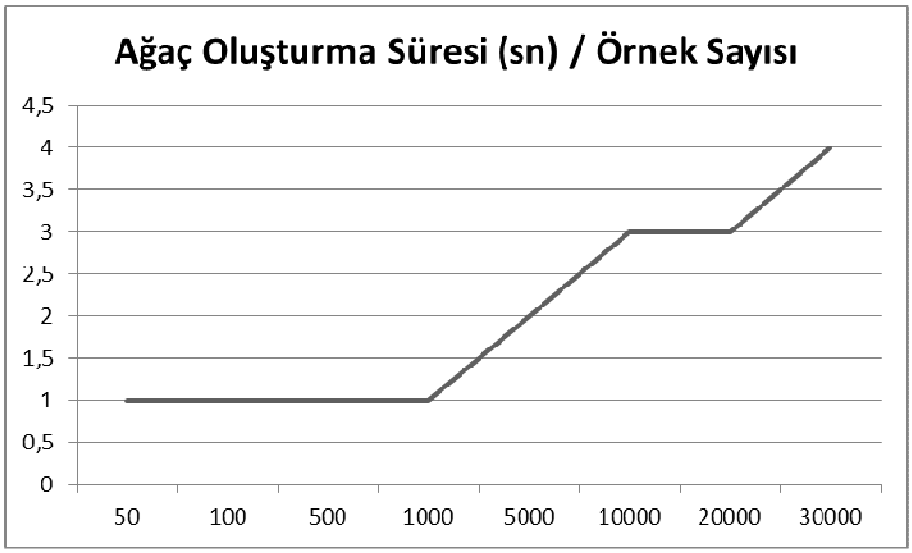
2.3.4. ID3 Algoritması Performans Analizi

ID3 algoritması kullanılarak farklı veri setlerinin performansları zaman ve örnek sayısı dikkate alınarak Çizelge 2.7.'de gösterilmiştir.

Çizelge 2.7 ID3 Algoritması Performans Analizi

Örnek Sayısı	Ağaç Oluşturma Süre (sn)
50	1
100	1
500	1
1000	1
5000	2
10000	3
20000	3
30000	4

Çizelge 2.7.'deki örnek sayıları ve ağaç oluşturma süreleri dikkate alınarak oluşturulan eğri Şekil 2.5.'de gösterilmiştir.



Şekil 2.5 ID3 Algoritması Performans Analizi Eğrisi

Şekil 2.5.'de gösterilen eğri dikkate alınarak ID3 algoritması örnek sayısındaki artış, ağaç oluşturma süresini doğrusal bir şekilde etkilemektedir

3. BÖLÜM : Lojistik Regresyon Analizi Giriş ve Temel Kavramlar

3.1. Giriş

Tez çalışmasında mutabakat fonksiyonunu bulabilmek için lojistik regresyon analizi kullanılmıştır. Lojistik regresyon analizi ile tüm veri setleri modellerinin katsayıları hesaplanmıştır. Hesaplanan katsayılar Forecaster uygulaması ile veri seti üzerinde denenmiş ve kesinlik düzeyleri bulunmuştur. Bulunan kesinlik düzeylerinin, en başarılı modellerin kesinlik düzeylerinden daha yüksek olduğu görülmüştür.

Basit ve çoklu regresyon analizlerinin uygulanabilmesi için değişkenlerin bazı varsayımları yerine getirmesi gerekir. Bu koşulların sağlanmadığı veri setlerine basit ya da çoklu regresyon analizleri uygulanamaz. Lojistik regresyon analizi, normal dağılım varsayımı ve süreklilik ön koşulu gerektirmeyen bir regresyon yöntemidir [26].

Bu bölümde lojistik regresyon analizi tanımlanmış ve matematiksel ifadesi gösterilmiştir. Çalışmada lojistik regresyon analizi için SPSS 15.0 programı kullanılmıştır.

3.1.1. Lojistik Regresyon Analizi

Lojistik regresyon, cevap değişkenin kategorik olarak, ikili (binary, dichotomous), üçlü ve çoklu kategorilerde gözlendiği durumlarda açıklayıcı değişkenlerle sebep-sonuç ilişkisini belirlemede yararlanılan bir yöntemdir. Açıklayıcı değişkenlere göre cevap değişkenin beklenen değerlerinin olasılık olarak elde edildiği sınıflama ve atama işlemi yapmaya yardımcı olan bir regresyon yöntemidir. Lojistik regresyon yönteminde bağımlı değişken üzerinde açıklayıcı

değişkenlerin etkileri olasılık olarak hesaplanarak risk faktörlerinin olasılık olarak belirlenmesi sağlanır [26].

Lojistik regresyon analizinde üç yöntem vardır.

i. İkili Lojistik Regresyon,
ii. Sıralı Lojistik Regresyon,
iii. İsimsel Lojistik Regresyon.

Bu çalışmada ikili lojistik regresyon analizi kullanılmıştır. İkili lojistik regresyon, ikili cevap içeren bağımlı değişkenlerle yapılan lojistik regresyon analizidir. Bir ya da daha fazla açıklayıcı değişken ile ikili cevap değişken arasındaki bağıntıyı ortaya çıkarır.

3.1.2. Lojistik Regresyon Modeli

Lojistik regresyon, bağımlı değişkenin tahmini değerlerini olasılık olarak hesaplayan ve olasılık kurallarına uygun sınıflama yapma imkanı veren bir istatistiksel yöntemdir. Lojistik regresyon tablolaştırılmış ya da ham veri setlerini analiz eden bir yöntemdir. İki değişkenli lojistik regresyon modeli Denklem 3'de gösterilmiştir.

$$P(Y) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} \quad (3)$$

Regresyon katsayılarının hesaplanması aşağıdaki gibi yapılır;

$$ln\left(\frac{P(Y)}{Q(Y)}\right) - \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots . + \beta_p X_p \quad (4)$$

$$\frac{P(Y)}{Q(Y)} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p} = e^{\beta_0} e^{\beta_1 X_1} e^{\beta_2 X_2} \ldots . e^{\beta_p X_p} \quad (5)$$

Birden çok bağımsız değişken içeren bir model için regresyon analizi denklemi aşağıdaki gibi gösterilmiştir.

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots . + \beta_n X_n \quad (6)$$

$X_1 \ldots X_n$ sebepleri gösterirken, Y sonucu göstermektedir. α sembolü ise sabit katsayı değeridir. Bu denklemi bulabilmek için her bir gözlem aşağıdaki gibi formüle edilir.

$$Y_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots . + \beta_n X_{in} + \varepsilon_i \quad (7)$$

Her bir gözlemin numarası i ile gösterilmiştir. Regresyon analizi matris olarak aşağıdaki gibi gösterilir;

$$X = \begin{bmatrix} X_{10} & X_{11} & \ldots & X_{1n} \\ X_{20} & X_{21} & \ldots & X_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ X_{m0} & X_{m1} & \ldots & X_{mn} \end{bmatrix}$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_m \end{bmatrix} \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_m \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix}$$

Y sonuç kümesini, ε hata kümesini ve β katsayıları göstermektedir.

3.2. Lojistik Regresyon Analizi Kullanım Gerekçesi

Tez çalışmasında kullanılan veri setlerinde 4 adet bağımlı ve 1 adet bağımsız değişken bulunmaktadır. Bu değişkenlere uygulanacak olan regresyon analizi için verilerin sürekli ve normal dağılımı varsayımını sağlaması gerekmektedir. Ancak kullanılan veri setlerinin yapısı ikili yapıda olduğu için bu veri setlerine uygulanacak en uygun analiz olarak lojistik regresyon analizi seçilmiştir.

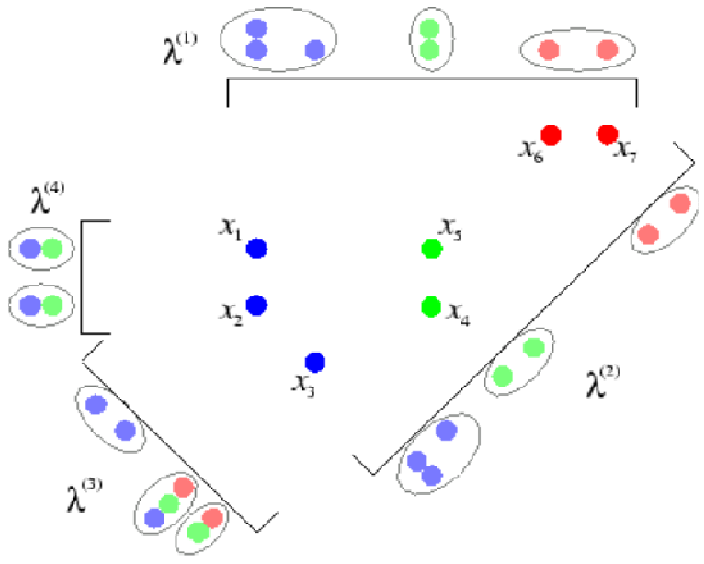
Lojistik regresyon analizinin uygulanabilmesi için en az bir değişkenin iki ve daha fazla yapıda değer alması gerekir. Tez çalışmasında 4 adet bağımsız ve 1 adet bağımlı ve ikili yapıda girilmiş değerler bulunmaktadır; dolayısıyla lojistik regresyon modeli tercih edilmiştir.

4. BÖLÜM : Mutabakat Fonksiyonu Giriş Ve Temel Kavramlar

4.1. Mutabakat Fonksiyonu

Mutabakat Fonksiyonu, bir konu, bir olay veya bir vaka durumunda uzlaşma durumudur. Fikir birliği veya oydaşım kelimeleri de Türkçemizde mutabakat ile eşanlamlı kelimeler olarak kullanılmaktadır.

Çeşitli algoritmalar ile oluşturulmuş karar ağaçları bir durum için farklı tahminler oluşturabilmektedir. Tüm karar ağaçlarını kullanarak bir mutabakat fonksiyonu hazırlayıp, ortak bir tahmin sonucunda uzlaşma oluşturulduğunda tahmin kesinliği arttırılabilir. Bunun için farklı modellerin farklı örnekler için oluşturduğu tahminlerin tek bir modelde toplanması gerekir. Şekil 4.1'de mutabakat fonksiyonu için bir örnek gösterilmiştir.



Şekil 4.1 Mutabakat Fonksiyonu Örneği

Şekil 4.1'de λ_1, λ_2, λ_3 ve λ_4 farklı modelleri, X_1, X_2, X_3 ve X_4 ise farklı örnekleri göstermektedir [27]. λ'ların yanındaki kümeler *1, 2* ve *3* şeklinde bulunan tahminlerdir. Bulunamayan sonuçlar *?* ile gösterilmiştir.

λ_1= *{1,1,1,2,2,3,3}*
λ_2= *{2,2,2,3,3,1,1}*
λ_3= *{1,1,2,2,3,2,3}*
λ_4= *{1,2,?,1,2,?,?}*

1. ve *2.* modellerin tahmin kümesi benzerlik göstermektedir. *3.* tahmin kümesi bunlardan farklıdır. *4.* tahmin kümesi ise tutarsızlık ve eksikler içermektedir. Bu tahmin kümelerinin tümü veya bir kısmı kullanılarak daha başarılı ve tek tahmin kümesi elde edilmesi beklenmektedir. Böyle bir mutabakat fonksiyonu oluşturabilmek için birliktelik metotları (ensemble methods) kullanabilir.

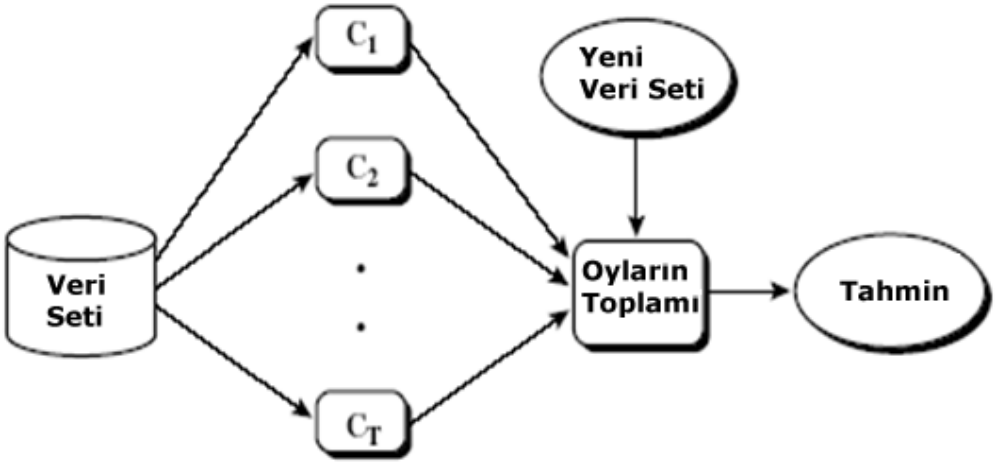
4.2. Birliktelik Metotları (Ensemble Methods)

Veri madenciliğinde kullanılan temel tekniklerden birisi birliktelik metotlarıdır. Birliktelik metotları, veri seti analiz edilerek bu veri seti içindeki birliktelik davranışlarının bulunmasını ve tahminine yönelik çalışmalar yapılmasını sağlarlar.

Birliktelik metotları, oluşturulan modellerin tahmin kesinliğini arttırmak ve bu modelleri birleştirilmek amacıyla kullanılır. Örneğin birliktelik metotları ile üç farklı modelin sonuçları bir uzlaşma sağlayarak tek bir sonuç üretilebilir.

Şekil 4.2.'de basit olarak birliktelik metotlarının nasıl çalıştığı gösteren bir örnek bulunmaktadır [27]. *"Data"* veri setini, *"*$C_1,C_2,..C_T$*"* ise farklı modellerin sonuçlarını gösteren kümedir. $C_1,C_2,..C_T$ modellerinin tüm oy sonuçları dikkate

alınarak mutabakat fonksiyonu oluşturulur. Oluşturulan mutabakat fonksiyonu dikkate alınarak girdilere istinaden sonuçlar tahmin edilir.



Şekil 4.2 Birliktelik Metotları Örneği

Farklı modelleri dikkate alarak mutabakat fonksiyonu oluşturmanın birçok yöntemi vardır. Bu yöntemlerden en çok kullanılanları [27];

- Paketleme (Bagging)
- Ağırlıklı Paketleme (Boosting)

Paketleme modellerin aritmetik ortalamasını alırken, ağırlıklı paketleme metodu ise modellerin ağırlıklı ortalamasını almaktadır.

4.2.1. Paketleme

Bu metot birden fazla doktorun hastalık tanısı koyma sırasında oylama yapılarak toplam en yüksek oyun tanı oluşturma kararına varılmasına benzetilebilir.

Oluşturulan modeller bir örnek küme üzerinde çalıştırılıp tahminler oluşturulur. Herhangi bir örnek için modellerden tahminleri istenir, paket model tahminleri sayar ve en yüksek oyu alan tahmin seçilerek o örneğin tahmini olarak atanır. Genellikle oluşturulan yeni paket model tek bir modelin oluşturduğu sonuçlara göre daha fazla kesinlik içermektedir [27].

T farklı karar ağacı küme sayısı olsun, n örnek veri setinin uzunluğu, H oluşturacak yeni fonksiyon olarak adlandırılırsa; Paketleme metodu sözde kodu aşağıdaki gibi yazılabilir.

1. Örnek veri seti $(x_1,y_1).....(x_n,y_n)$
2. For I=1..T
    a. $H(x)=majority(h_1(x).....h_t(x))$
3. Result H

4.2.2. Ağırlıklı Paketleme

Ağırlıklı Paketleme birden fazla doktora danışarak ardından her bir doktorun verdiği kararı belirli bir ağırlığa göre (Örnek: Ünvan) bir tanı oluşturulması şeklinde açıklanabilir.

Uygulamada oluşturulan modellere farklı ağırlıklar verilir. Her bir modelden bir örnek için tahmin istenir. İstenen tahminler ağırlıklar ile çarpılır ve toplanır. Oluşan sonuç içinde en yüksek kat sayıya sahip tahmin sonuç tahmin olarak örneğe atanır.

Bu yöntem oldukça başarılıdır. Fakat ağırlıkları değiştirip ideale yakın sonuç oluşturulması durumunda farklı örnek gruplarında başarısız olabilir [27].

T farklı karar ağacı küme sayısı olsun, n örnek veri setinin uzunluğu, H oluşturacak yeni fonksiyon olarak adlandırılırsa; Ağırlıklı Paketleme metodu sözde kodu aşağıdaki gibi yazılabilir.

1. Örnek veri seti $(x_1, y_1) \ldots\ldots (x_n, y_n)$
2. For I=1..T
    a. $H(x) = avg(h_1(x) w_1 + \ldots\ldots + h_t(x) w_t)$
3. Result H

5. BÖLÜM : Uygulama Ortamı Tasarımı ve Kullanılan Veri Setleri

5.1. Uygulama Hakkında

Uygulamada üç farklı veri seti için karar ağaçları oluşturulmuş ve kesinlik düzeyleri hesaplanmıştır. Seyahat akışı, banka kredi bilgileri ve yetişkin bilgileri veri setleri, kullanılan veri setleri bölümünde anlatılmıştır. Uygulama sırasında adım adım yapılan çalışmalar Şekil 5.1.'de gösterilmiştir.

Şekil 5.1 Uygulama Akışı

Uygulamanın ilk adımında tüm veri üzerinden rasgele örnek veri seçilmiştir. İkinci adımda örnek veri seti kullanılarak karar ağaçları elde edilmiştir. Takip eden adımda tüm veri setinden örnek veri seti çıkartılmıştır ve uygulama veri seti elde edilmiştir. Dördüncü adımda uygulama veri seti üzerinde karara ağaçları algoritmaları kullanılarak kesinlik düzeyleri hesaplanmıştır. Beşinci adımda lojistik regresyon analizi ile mutabakat fonksiyonu hesaplanmıştır. Altıncı ve yedinci adımda sırasıyla mutabakat fonksiyonu kesinlik düzeyi bulunur ve diğer algoritmaların kesinlik düzeyleri ile mutabakat fonksiyonu kesinlik düzeyi karşılaştırılmıştır.

5.2. Uygulama Geliştirmede Kullanılan Araçlar

Uygulamanın geliştirilmesi ve deneylerin yapılabilmesi için birden fazla araç kullanılmıştır. Kullanılan araçlar aşağıda listelenmiştir;

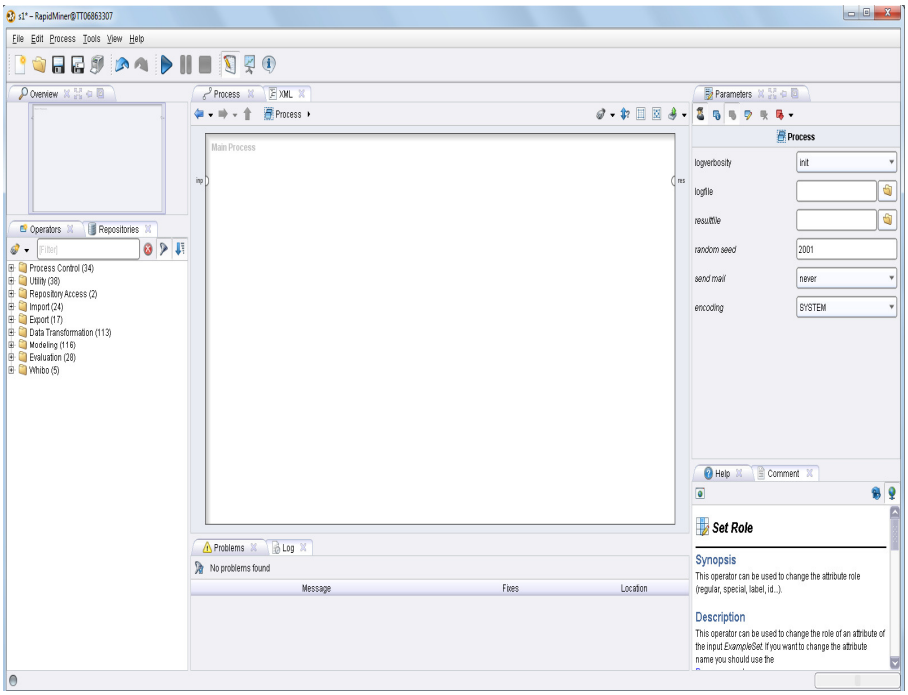
- Rapid Miner,
- Whibo,
- Forecaster,
- SPSS,
- MS Visual Studio .NET 2008,
- MS SQL Server 2008.

Rapid Miner ve Whibo eklentisi ile karar ağaçları hesaplanmıştır. Forecaster uygulaması ile kesinlik düzeyleri hesaplanmıştır. SPSS ile mutabakat fonksiyonu hazırlanmıştır. MS Visual Studio .NET 2008 uygulaması ile C# dili kullanılarak Forecaster uygulaması yazılmıştır. MS SQL Server 2008 uygulaması ile Forecaster'da kullanılmak üzere uygulama verisi hazırlanmış ve tüm veriler bu veri tabanı uygulamasında tutulmuştur.

5.2.1. Rapid Miner Aracı

Rapid Miner, veri madenciliği için geliştirilmiş açık kaynak kodlu yazılımlardan biridir. Java tabanlı geliştirilen yazılım birçok veri madenciliği projesinde kullanılmıştır. Tez hazırlanırken Rapid Miner uygulaması farklı karar ağaçları üretilmek için kullanılmıştır.

Rapid Miner uygulamasının kullanıcı arayüzü aracılığıyla tasarlanan model ile farklı algoritmalar ve farklı veri seti formatları birleştirilerek çalışmanın modeli oluşturulur. Çalıştır tuşu ile uygulama çalıştırılarak sonuçlar ekranda gözlemlenir veya farklı formattaki dosyalara çıktı olarak verilir. Şekil 5.2'de Rapid Miner uygulaması kullanıcı arayüzü gösterilmiştir.

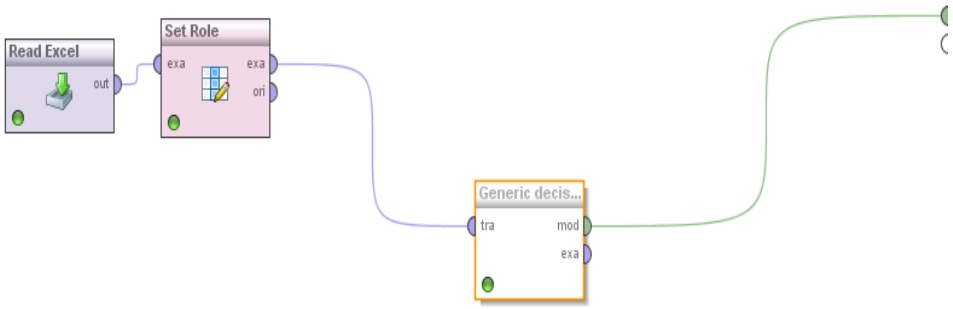


Şekil 5.2 Rapid Miner Kullanıcı Arayüzü

5.2.2. Whibo Eklentisi (Rapid Miner Eklentisi)

Rapid Miner uygulaması için bir eklentidir. Karar ağaçlarından C4.5, ID3, CART ve CHAID'i uygulayabilmek için kullanılmıştır. Açık kaynak kodlu bir eklentidir. Rapid Miner ve Whibo kullanılarak uygulama için hazırlanan tasarım Şekil 5.3.'de bulunmaktadır.

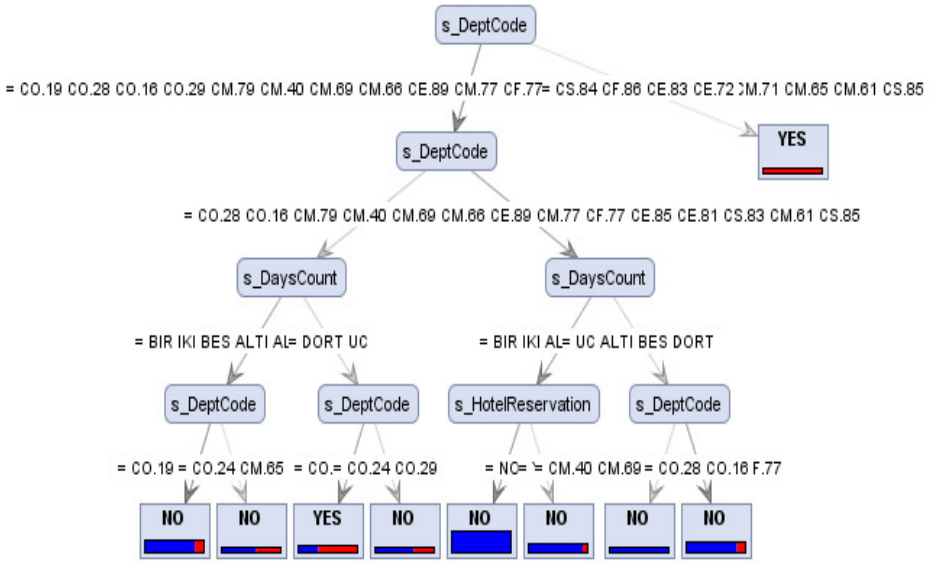
Read Excel bileşeni ile akış verisi alınmaktadır. "Set role" bileşeni ile tahmin edilecek sütun belirlenmektedir. "Generic decision tree" bileşeni ile kullanılacak ağaç modeli seçilmektedir.



Şekil 5.3 Rapid Miner Üzerinde Whibo ile Geliştirilen Model

Model çalıştırıldıktan sonra oluşturulan ağaç yapısı iki farklı şekilde çıktı olarak alınabilir. Bunlar;

- Görsel Ağaç Gösterimi,
- Metin Ağaç Gösterimi.

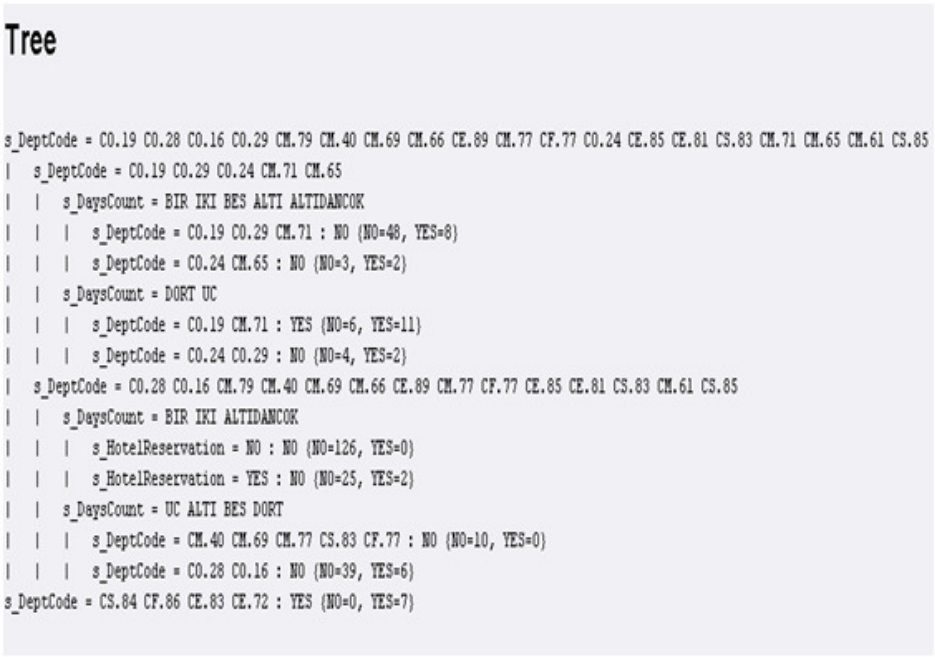


Şekil 5.4 Görsel Karar Ağacı Çıktısı

Şekil 5.4.'de elde edilen karar ağacının görsel olarak çıktısı görülmektedir. Mavi oval kareli kutucukların içerisinde hedefe ulaşmak için gerekli sütunların ismi bulunur. Bunların hemen altında "=" ifadesinden sonra gelen kısım ise aldıkları değerlerdir. Ağaç üzerinde bulunan NO ve YES alanları hedef sütunun çıktılarıdır, altlarında bulunan kırmızı mavi işaretler ise, bu alanların ne kadar doğru olduğunu gösterir. %100 NO veya YES sonucuna ulaşılan kısımlar komple kırmızı veya mavi olarak gösterilir. Bir ağacın %100 tüm sonuçları karşılaması durumu ideal durumdur, gerçek hayatta büyük veri setlerinde bu tahmin oranı ile karşılaşılması oldukça güçtür.

Şekil 5.5.'de karar ağacının metin olarak çıktısı gösterilmiştir. Metin üzerinden incelendiğinde "|" ifadesi ile gösterilen kısımlar dallanmayı ifade etmektedir. ":" ifadesinden sonra gelen kısım hedef sütundaki sonucu gösterir. Parantez içerisinde yazan kısımlar bu sonucu hangi sayı ile oluşturduğunu gösterir.

Örnek olarak 4. satırda 48 adet NO sonucu 8 adet YES Sonucu bulunuştur, NO sonucu genel sonuç olarak yazılmıştır.



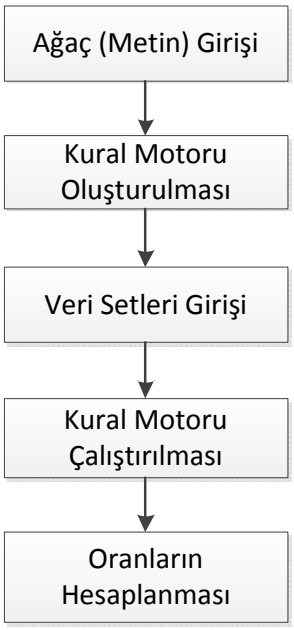
Şekil 5.5 Metin Karar Ağacı Çıktısı

5.2.3. Sonuçları Yorumlamak için Geliştirilen Forecaster Aracı

Rapid Miner ve Whibo eklentisi kullanılarak hazırlanan metin şeklinde ağacın alınıp, işlenip, ardından veri seti ile ilişkilendirilip sonuçların oluşturulması için yeni bir yazılıma ihtiyaç duyulmuştur. Bunun içim Forecaster adı verilen ek bir yazılım geliştirilmiştir. Forecaster uygulaması windows uygulaması şeklinde hazırlanmış, uygulama yazılım dili olarak C#, veri tabanı olarak ise MS-SQL Server kullanılmıştır.

Forecaster'ı çalıştırmadan önce hazırlanan metin şeklinde ağaçlar veri tabanında hazırlanan tablolara girilir. Forecaster ile toplam 12 adet farklı karar ağacı aynı anda veri setine uygulanarak sonuçlar oluşturulabilir ve kesinlik düzeyleri bulunabilir. Oluşturulan programın çalışma adımları kısaca Şekil 5.6.'da gösterilmiştir.

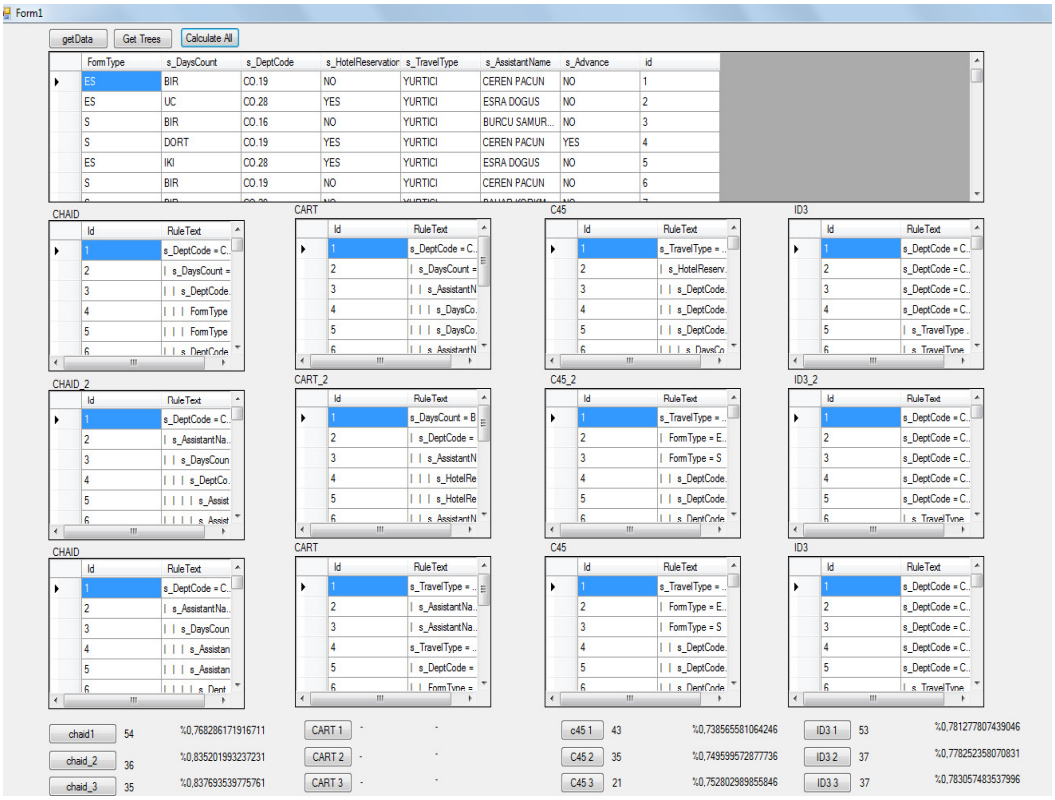
Whibo ile üretilen karar ağacı metin olarak sisteme alınır ve veri tabanına kaydedilir. Veri tabanına kaydedilirken herhangi bir işlem uygulanmaz. Her bir dal ayrı bir satır olarak tabloya eklenir. Kural motoru oluşturulurken, veri tabanındaki ağaç üzerinde gezilir, geliştirilen dönüştürücü aracılığıyla if-else yapısına dönüştürülür. Kural motoru oluşturulduktan sonra kullanıcı verisi sisteme girilir. Kullanıcı verisi de veri tabanında bir tablo üzerinde kayıt edilir. Girilen kullanıcı verisi üzerinde kural motoru çalıştırılır. Her bir satır için birer tahmin oranı oluşturulur. Kural motoru sonrası bulunan sonuçlardan Oranların Hesaplanması modülü ile başarı oranları analiz edilir.



Şekil 5.6 Forecaster Çalışma Adımları

Şekil 5.7.'de Forecaster uygulamasının kullanıcı ekranı verilmiştir, üst alanda bulunan getData butonu ile tüm veri seti sisteme alınır, getTrees butonu ile karar ağaçları uygulama içerisine alınır, calculateAll butonu ile tüm ağaçlar veriler üzerine uygulanarak sonuçlar ve doğruluk oranları hesaplanır. Uygulamanın alt kısmında bulunan butonlar ile farklı karar ağaçları tüm veri setine tek tek uygulanabilir.

Uygulanan karar ağaçları algoritmaları sırasıyla CHAID, CART, C4.5 ve ID3'dir. Uygulama farklı boyutlardaki veri örnekleri ile hazırlanan ağaçlar kullanmaktadır. Oranların hesaplanması için Hata Matrisi (Confusion Matrix) kullanılmıştır.



Şekil 5.7 Forecaster Kullanıcı Arayüzü

Örnek olarak uygulamada chaid1 butonuna tıklandığında çalışacak metodun sözde kodu aşağıdaki gibidir.

```
1. Satir=0
2. Veri tabanından verileri al
3. Veri tabanından karar ağacını al
```

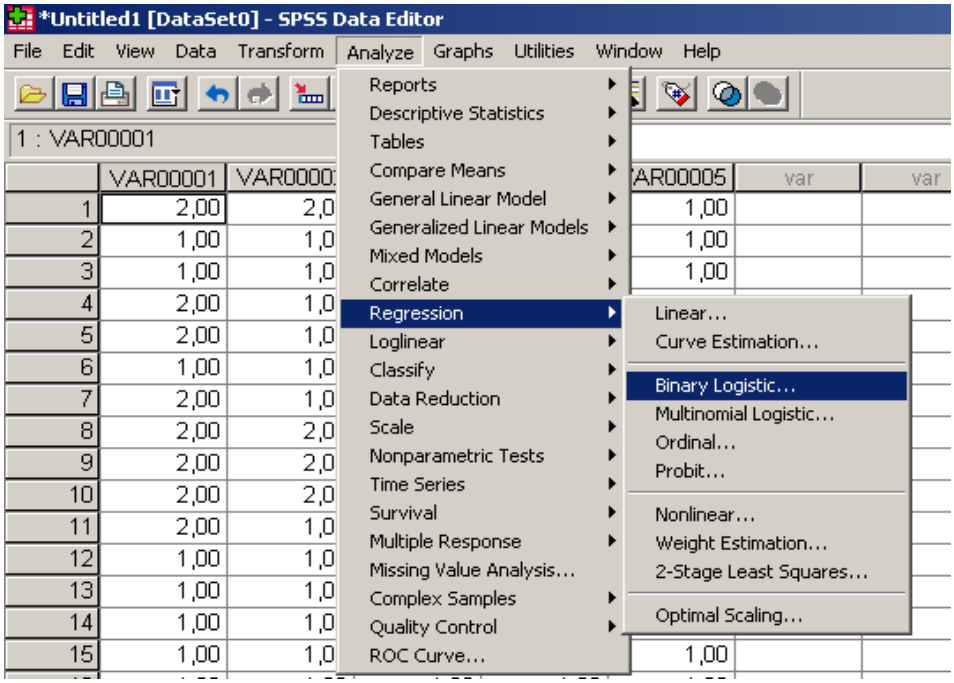
4. *Sonuç bulunacak satırın sütunlarını al*
5. *ID=0,Level=0*
6. *Ağacın ID ve Level elemanından başla ve bu elemanın değeri ile ilgili satırın sütununu karşılaştır*
7. *Eğer sonuç YES veya NO ise 11'e git*
8. *Sonucu aramak için ağacın aynı levelda bulunan bir sonraki elemanına bak, bir sonraki eleman yoksa 10'a git,ID=ID+1*
9. *Git 7*
10. *Level=Level+1, Git 6*
11. *Sonuç bulundu, bir sonraki veri satırı için satır=satır +1*
12. *Satırlar bitti ise 13, bitmediyse Git 4*
13. *Oranları hesapla*

5.2.4. SPSS Uygulaması

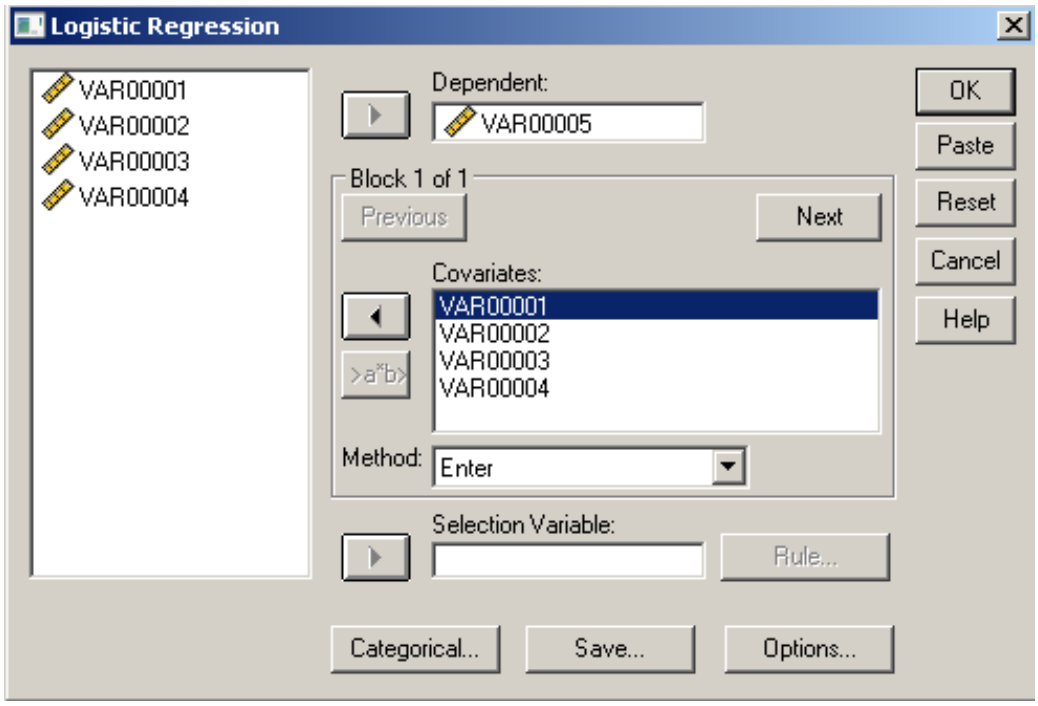
SPSS (Statistical Packages for the Social Sciences) uygulaması Sosyal Bilimler için verilerin analizinin yapılması, anlamlandırılması, veriler arasında ilişikilerin ortaya çıkarılması amacıyla kullanılan bir programdır [29].

Tez'de SPSS 15.0 versiyonu kullanılarak, lojistik regresyon tekniği ile karar ağaçlarından mutabakat fonksiyonu oluşturulmaya çalışılmıştır.

SPSS ile lojistik regresyon analizi yapabilmek için öncelikle uygulama verisinin SPSS'e alınması gerekmektedir. Veri setini herhangi bir Excel dosyasından kopyalayarak SPSS üzerine yapıştırarak SPSS'e veri girişi yapılabilir. Veri girişi tamamlandıktan sonra veri üzerinde yapılacak çalışma için "Analyze" menüsü altından "Regression" seçeneği ve ardından "Binary Logistic" seçilir. Şekil 5.8'de bu seçim gösterilmiştir.



Şekil 5.8 SPSS Lojistik Regresyon Analizi Seçimi



Şekil 5.9 SPSS Lojistik Regresyon Bağımlı ve Bağımsız Değişken Seçimi

Lojistik regresyon analizi seçildikten sonra açılan pencerde bağımlı ve bağımsız değişkenler seçilir. Ardından "OK" butonu ile regresyon analizi sonuçları elde edilir. Şekil 5.9'da bağımlı ve bağımsız değişkenlerin seçildiği ekran gösterilmektedir.

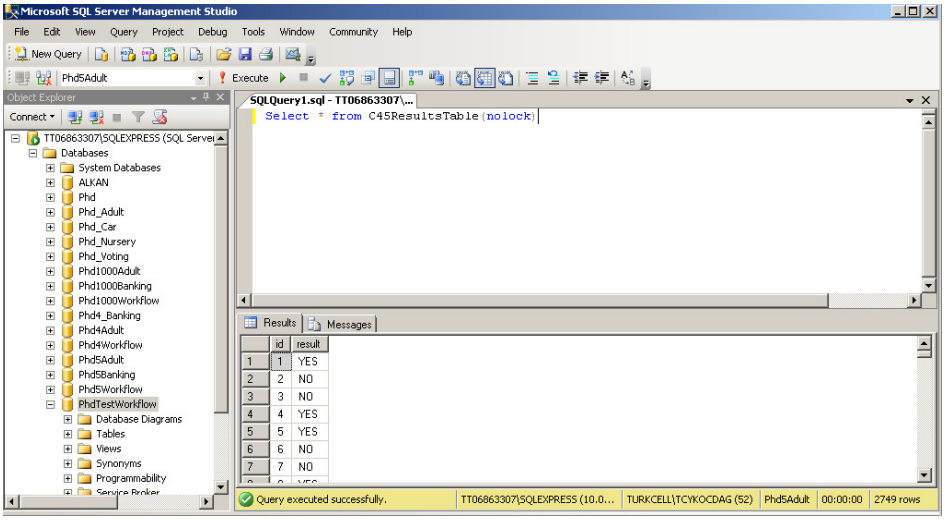
5.2.5. Microsoft Visual Studio.NET 2008

Forecaster uygulamasının geliştirilmesi için Microsoft Visual Studio.NET 2008 kullanılmıştır. Microsoft Visual Studio 2008, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır. Forecaster uygulaması geliştirilirken C# programlama dili kullanılmıştır.

5.2.6. Microsoft SQL Server 2008

Forecaster uygulamasının ihtiyaç duyduğu iş akışı verisi ile işlemler sırasında ihtiyaç duyulan verilerin saklanması ve birbiri ile ilişkilendirilmesi için Microsoft SQL Server 2008 yazılımı kullanılmıştır. Microsoft SQL Server, bilgi yönetimi ve depolama amacıyla kullanılan bir ilişkisel veri tabanı yönetimi sistemidir.

Veri tabanından sorgulama yapılabilmesi için öncelikle bağlantı sonrası açılan ekrandan ilgili veri tabanı seçilir, ardından sorgu dili kullanılarak sorgulanmak istenen bilgi istenir. Şekil 5.10.'da örnek sorgulama ekranı gösterilmektedir.



Şekil 5.10 Microsoft SQL Server 2008 Veri Tabanı Sorgulama Ekranı

5.3. Kullanılan Veri Setleri

Çalışmalar üç farklı veri seti üzerinde gerçekleştirilmiştir. Bunlar; Seyahat Akışı, Banka Kredi (Bank Credits) ve Yetişkin Bilgileri (Adult) 'dir. Banka Kredi ve Yetişkin Bilgileri veri setleri kısa adı UCI olan "University of California, Irvine" bünyesinde bulunan "Machine Learning Repository" adlı veri tabanından alınmıştır [28].

5.3.1. Seyahat Akışı Veri Seti

Seyahat akışı veri seti, bir iletişim firmasının hali hazırda kullanılan seyahat akışı veri setinin gerçek bilgileri maskelenerek oluşturulmuştur. Oluşturulan veri setinde bazı satırlar ve sütunlar çalışmaya etki etmeyeceği düşünüldüğünden çıkarılmıştır. Veri madenciliği çalışmasına uygun 7 sütun 5500 satırdan oluşan bir veri seti elde edilmiştir. Son sütun tahmin edilecek onay sütununu göstermektedir. Çizelge 5.1'de bu veri setine ilişkin detaylı sütun bilgileri gösterilmiştir.

Çizelge 5.1 Seyahat Akışı Veri Seti

Sütun İsmi	**Açıklama**	**Alabileceği Değerler**
FormType	Seyahat formunun tipini gösterir.	ES(Eğitim Seyahat), E(Eğitim)
DaysCount	Seyahat süresini gösterir.	Sayısal değerler.
DeptCode	Çalışanın çalıştığı bölümün kodunu gösterir.	Metin değerler.
HotelReservation	Otel rezervasyonu olup olmayacağını gösterir.	Evet/Hayır
TravelType	Seyahatin yurtiçi mi yurtdışı mı olduğunu gösterir.	Yİ/YD
AssitantName	Çalışanın bağlı olduğu asistanı gösterir.	Metin değerler.
Approve	Akışın onaylanıp onaylanmadığı bilgisini gösterir.	Evet/Hayır

Seyahat akışı 4 adımlıdır. Bu adımlar sırasıyla;

i. Akış başlatan bilgi girişi,
ii. Bölüm asistanı onayı,
iii. Acente onayı,
iv. Yönetici onayı.

şeklindedir.

5.3.2. Banka Kredi Bilgileri Veri Seti

Banka kredileri bilgi seti bir özel bankanın müşterilerinin bilgileri ile hazırlanmıştır. Tüm veri "University of California, Irvine" bünyesinde bulunan "Machine Learning Repository" adlı veri tabanından alınmıştır. 1000 Satır ve 20 sütundan oluşan veri setinden 7 adet sütun seçilerek karar ağaçları oluşturulmuştur. Çizelge 5.2'de bu veri setine ilişkin detaylı sütun bilgileri gösterilmiştir.

Çizelge 5.2 Banka Kredi Veri Seti

Sütun İsmi	**Açıklama**	**Alabileceği Değerler**
Credit History	Kredi Geçmişi Bilgisi	A31-A34
Purpose	Amaç	A41-A49 ve A100.
Savings	Müşteri Hesap Bilgisi	A61-A65.
Present Employment	Çalışma Geçmişi	A71-A75.
Personal Status and Sex	Medeni Hali ve Cinsiyeti	A91-A95.
Other Debtors / Guarantors	Kefalet Bilgileri	A101-A103.
Result	Kredi Sonucu	Evet/Hayır

5.3.3. Yetişkin Bilgileri Veri Seti

Yetişkin bilgileri veri seti, farklı özelliklere sahip yetişkin bireylerin özellikleri ile hazırlanmıştır. Yetişkinlerin verilerine göre yıllık gelirlerini tahmin etmeye yönelik bir veri setidir. Bu veri seti' de "University of California, Irvine" bünyesinde bulunan "Machine Learning Repository" adlı veri tabanından alınmıştır. 48842 Satır ve 14 sütundan oluşan veri setinden 3000 satır ve 7 adet sütun seçilerek karar ağaçları oluşturulmuştur. Çizelge 5.3'de bu veri setine ilişkin detaylı sütun bilgileri gösterilmiştir.

Çizelge 5.3 Yetişkin Bilgileri Veri Seti

Sütun İsmi	**Açıklama**	**Alabileceği Değerler**
Workclass	İş Sınıfı	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
Education	Eğitim Bilgisi	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
Marital Status	Medeni Hali	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
Occupation	Meslek Bilgisi	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
Relationship	İlişki Bilgisi	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
Race	Irk.	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
Result	Yıllık Kazanç Sonucu (50.000<X).	Evet/Hayır

6. BÖLÜM : Karar Ağaçları ve Mutabakat Fonksiyonu Uygulama Sonuçları

6.1. Uygulama Sonuçları Hakkında

Bu bölümde Seyahat Akışı, Banka Kredi Bilgileri ve Yetişkin Bilgileri Uygulama sonuçları farklı bölümler şeklinde ele alınarak anlatılmıştır. Her bir bölümde yapılan çalışma özetlenmiş, karar ağaçları ve mutabakat fonksiyon sonuçları gösterilmiş, hata matrisleri ortaya çıkarılmış, ek deneylerden bahsedilmiş ve lojistik regresyon sonuçları yorumlanmıştır. Son bölümde tüm sonuçlar bir arada yorumlanmıştır.

6.2. Seyahat Akışı Uygulama Sonuçları

5619 satırdan ve 7 sütundan oluşan veri seti üzerinde 250 satırlık rasgele örnek seçilmiştir. Bu örnek veri üzerinde Rapid Miner ve Whibo eklentisi kullanılarak ID3, C4.5, CHAID ve CART algoritmaları çalıştırılmıştır. Oluşan karar ağaçları tüm veriye uygulanarak tahmin sonuçları elde edilmiştir. Tahmin sonuçları gerçek verilerin sonuçları ile karşılaştırılarak karar ağaçlarının kesinlik düzeyleri hesaplanmıştır. Bu çalışmaya ek olarak farklı boyutlardaki örnek veri setleri ile deneyler desteklenmiştir.

6.2.1. Karar Ağaçları ve Mutabakat Fonksiyon Sonuçları

Çizelge 6.1'de Karar ağaçları ve lojistik regresyon analizi sonucunda bulunan mutabakat fonksiyonu sonuçları karşılaştırılmıştır. Bu çalışma örnek alınan 250'lik

veri seti ile oluşturulan karar ağacı ve oluşturulan mutabakat fonksiyonu tüm veri setine uygulanmıştır.

Çizelge 6.1 Karar Ağaçları ve Mutabakat Fonksiyonu Kesinlik Düzeyleri

Yöntem		**Kesinlik Düzeyi**
Karar Ağacı Algoritmaları	CHAID	0,838
	CART	0,685
	C4.5	0,741
	ID3	0,765
Mutabakat Fonksiyonu		0,842

Mutabakat fonksiyonu ile karar ağaçlarına göre daha yüksek kesinliğe sahip bir sonuç ortaya çıkarılmıştır. SPSS ile bulunan mutabakat fonksiyonu aşağıda gösterilmiştir.

*Mutabakat Fonksiyonu = 1,446 * CART + 1,785 * CHAID*

Çizelge 6.2'de oluşturulan mutabakat fonksiyonu hata matrisi gösterilmiştir.

Çizelge 6.2 Mutabakat Fonksiyonu Hata Matrisi

Gerçek Sonuç / Mutabakat Fonksiyonu	**Evet**	**Hayır**
Evet	272	325
Hayır	521	4251

Toplam 5619 satırdan 250 adet örnek veri çıkartılmıştır. 5369 satır içerisinde başarılı bulunan iki küme (Evet / Evet ve Hayır / Hayır) toplanarak 4523 satır başarılı bulunmuştur. Buradan yola çıkılarak;

Mutabakat Fonksiyonu Kesinlik Düzeyi = 4523 / 5369
= 0,8424

olarak bulunmuştur.

Bu çalışmaya ek olarak, rasgele farklı örnek veri setleri seçilerek karar ağaçları yeniden oluşturulmuştur. Toplam veri üzerinden örnek veri setleri çıkartılıp kalan veri seti üzerinde karar ağaçları ve mutabakat fonksiyonu uygulanmıştır. Uygulama sonucunda kesinlik düzeyleri bulunmuş ve bulunan sonuçlar karşılaştırmalı olarak gösterilmiştir. Bulunan sonuçlar Çizelge 6.3.'de gösterilmiştir.

Çizelge 6.3 Ek Deney Sonuçları

		Kesinlik Düzeyi				
Yöntem		**Deney 1**	**Deney 2**	**Deney 3**	**Deney 4**	**Deney 5**
Karar Ağacı Algoritmaları	**CHAID**	0,816	0,786	0,745	0,756	0,798
	CART	0,766	0,719	0,72	0,701	0,742
	C4.5	0,800	0,799	0,754	0,760	0,802
	ID3	0,808	0,778	0,745	0,764	0,832
Mutabakat Fonk.		0,816	0,786	0,791	0,789	0,832

Deney 1 ve Deney 2'de mutabakat fonksiyonu, en iyi kesinlik düzeyini veren CHAID algoritması ile aynı değeri yakalamıştır. Deney 3 ve Deney 4'de mutabakat fonksiyonu tüm algoritmaların kesinlik düzeyinden daha yüksek kesinlik düzeyine ulaşmıştır. Deney 5'de ise mutabakat fonksiyonu, en iyi kesinlik düzeyine sahip ID3 algoritması ile aynı kesinlik düzeyine ulaşmıştır. Çizelge 6.4'de yapılan ek deney sonuçlarının ortalaması gösterilmiştir.

Çizelge 6.4 Ek Deney Sonuçları Ortalaması

Yöntem		Kesinlik Düzeyi
Karar Ağacı Algoritmaları	CHAID	0,708
	CART	0,730
	C4.5	0,783
	ID3	0,785
Mutabakat Fonksiyonu		0,803

Yapılan ek deneylerin sonuçları dikkate alındığında, ek deney sonuçlarının ortalamasındaki mutabakat fonksiyonu kesinlik düzeyi diğer algoritmaların ortalama kesinlik düzeyinden yüksektir. Bu sonuca göre mutabakat fonksiyonunun başarılı olduğu görülmektedir.

6.2.2. Lojistik Regresyon Sonuçları

Bu bölümde seyahat akışı için yapılan SPSS programı kullanılarak lojistik regresyon çalışması açıklanmıştır. Çizelge 6.5'de bağımlı değişkenin SPSS tarafında kodlaması gösterilmektedir.

Çizelge 6.5 Ek Bağımlı Değişkenin Kodlaması

Gerçek Değer	SPSS Değeri
Evet	0
Hayır	1

SPSS ilk adımda (Adım 0) lojistik regresyon sabitinin katsayısını hesaplamıştır. Çizelge 6.6'da sabit terimin katsayısı gösterilmektedir. SPSS sabit terimin katsayısını -2,028 olarak bulmuştur. Buradaki iterasyon 5. adımda bitmiştir. 5. Adımda bitmesinin sebebi iterasyon değişkeninin değişimi 0,001'den küçük olmasıdır.

Çizelge 6.6 Sabit Terimin Katsayısının Hesaplanması

Adım 0		
Iterasyon	**Iterasyon Değişkeni**	**Sabitin Katsayısı**
1	3579,643	-1,535
2	3446,653	-1,953
3	3443,828	-2,026
4	3443,826	-2,028
5	3443,826	-2,028

Adım 0'da bulunan sabitin katsayısını Wald İstatistiği yöntemi ile test edilmesi gerekmektedir. Çizelge 6.7.'de Sabit terimin katsayısının lojistik regresyon için uygun olup olmadığının testi için Wald İstatistiği yönteminin sonucunu göstermektedir. Çizelge 6.8. ise Wald istatistiğindeki kolonlardaki kısaltmaların adlarını göstermektedir.

Çizelge 6.7 Sabit Terimin Uyumunun Wald İstatistiği Sonucu

	B	**S.E.**	**Wald**	**Df**	**Sig(p)**	**Exp(B)**
Adım 0 Sabit	-2,028	0,045	2025,164	1	0,000	0,132

Çizelge 6.8 Wald İstatistiği Tablosu Kolonların Açıklaması

Terim	**Açıklama**
B	Sabit terime ait regresyon katsayısı
S.E.	Katsayıya ait standart hata (Standart Error)
Wald	Model uyumunu test eden wald istatistik sonucu
Df	Serbestlik derecesi (degree of freedom)
Sig(p)	Modelin uyumlu olup olmadığına karar vermemizi sağlayan p değeri
Exp(B)	Risk faktörü

Wald istatistiğinin sonucunda elde edilen p değerinin 0,05'den küçük olması durumunda %5 anlamlılık düzeyinde katsayının uyumlu olduğu söylenebilir. Çizelge 6.7'deki değer 0,05'in altında olduğu için katsayı denklem için uyumludur. Çizelge 6.9'da bağımsız değişkenlerin katsayıları gösterilmektedir.

Çizelge 6.9 Bağımsız Değişkenlerin Katsayıları

		B	S.E.	Wald	Df	Sig(p)	Exp(B)
Adım 1	2.Değişken	2,527	0,113	498,522	1	0,000	12,518
Adım 2	2.Değişken	1,446	0,108	179,015	1	0,000	4,245
	1.Değişken	1,785	0,124	206,274	1	0,000	5,959

Çizelge 6.9'da gösterildiği üzere Adım 2'deki Sig. (p) değerleri 0,05'in altındadır. Denklemi oluşturmak için Adım 2'deki 1. ve 2. değişken katsayıları kullanılmıştır.

6.3. Banka Kredi Bilgileri Uygulama Sonuçları

1000 satırdan ve 7 sütundan oluşan veri seti üzerinde 250 satırlık rasgele örnek seçilmiştir. Bu örnek veri üzerinde Rapid Miner ve Whibo eklentisi kullanılarak ID3, C4.5, CHAID ve CART algoritmaları çalıştırılmıştır. Oluşan karar ağaçları tüm veriye uygulanarak tahmin sonuçları elde edilmiştir. Tahmin sonuçlarını gerçek verilerin sonuçları ile karşılaştırılarak karar ağaçlarının kesinlik düzeyleri hesaplanmıştır. Bu çalışmaya ek olarak farklı boyutlardaki örnek veri setleri ile deneyler desteklenmiştir.

6.3.1. Karar Ağaçları ve Mutabakat Fonksiyonu Sonuçları

Çizelge 6.10'da Karar ağaçları ve lojistik regresyon analizi sonucunda bulunan mutabakat fonksiyonu sonuçları karşılaştırılmıştır. Bu çalışma örnek alınan

250'lik veri seti ile oluşturulan karar ağacı ve oluşturulan mutabakat fonksiyonu tüm veri setine uygulanmıştır.

Çizelge 6.10 Karar Ağaçları ve Mutabakat Fonksiyonu Kesinlik Düzeyleri

Yöntem		**Kesinlik Düzeyi**
Karar Ağacı Algoritmaları	CHAID	0,653
	CART	0,637
	C4.5	0,632
	ID3	0,605
Mutabakat Fonksiyonu		0,654

Mutabakat fonksiyonu ile karar ağaçlarına göre daha yüksek kesinliğe sahip bir sonuç ortaya çıkarılmıştır. SPSS ile bulunan mutabakat fonksiyonu aşağıda gösterilmiştir.

*Mutabakat Fonksiyonu = 0,642 * CHAID + 0,358 * C4.5*

Çizelge 6.11 Mutabakat Fonksiyonu Hata Matrisi

Gerçek Sonuç / Mutabakat Fonksiyonu	**Evet**	**Hayır**
Evet	403	141
Hayır	113	88

Çizelge 6.11'de oluşturulan mutabakat fonksiyonu hata matrisi gösterilmiştir. Toplam 1000 satırdan 250 adet örnek veri çıkartılmıştır. 750 satır içerisinde başarılı bulunan iki küme (Evet / Evet ve Hayır / Hayır) toplanarak 491 satır başarılı bulunmuştur. 4 Adet satır için sonuç bulunamamıştır.

Buradan yola çıkılarak;

Mutabakat Fonksiyonu Kesinlik Düzeyi = 491 / 750
= 0,6546

olarak bulunmuştur.

Bu çalışmaya ek olarak, rasgele farklı örnek veri setleri seçilerek karar ağaçları yeniden oluşturulmuştur. Toplam veri üzerinden örnek veri setleri çıkartılıp kalan veri seti üzerinde karar ağaçları ve mutabakat fonksiyonu uygulanmıştır. Uygulama sonucunda kesinlik düzeyleri bulunmuş ve bulunan sonuçlar karşılaştırmalı olarak gösterilmiştir. Bulunan sonuçlar Çizelge 6.12'de gösterilmiştir.

Çizelge 6.12 Ek Deney Sonuçları

		Kesinlik Düzeyi				
Yöntem		**Deney 1**	**Deney 2**	**Deney 3**	**Deney 4**	**Deney 5**
Karar Ağacı Algoritmaları	**CHAID**	0,569	0,619	0,640	0,643	0,605
	CART	0,709	0,705	0,682	0,713	0,681
	C4.5	0,699	0,68	0,735	0,654	0,697
	ID3	0,694	0,664	0,703	0,654	0,729
Mutabakat Fonk.		0,725	0,705	0,719	0,675	0,729

Deney 1'de Mutabakat Fonksiyonu tüm karar ağaçları içerisinde en yüksek kesinlik düzeyine sahiptir. Deney 2'de en yüksek kesinliği sahip CART, Deney 5'te ise en yüksek kesinlik düzeyine sahip ID3 algoritması ile aynı kesinlik düzeyine erişmiştir. Deney 3 ve Deney 4'de Mutabakat Fonksiyonu diğer algoritmaların kesinlik düzeylerinin altında kalmıştır. Çizelge 6.13'te yapılan ek deney sonuçlarının ortalaması gösterilmiştir.

Çizelge 6.13 Ek Deney Sonuçları Ortalaması

Yöntem		**Kesinlik Düzeyi**
Karar Ağacı Algoritmaları	CHAID	0,615
	CART	0,698
	C4.5	0,693
	ID3	0,689
Mutabakat Fonksiyonu		0,711

Yapılan ek deneylerin sonuçları dikkate alındığında, ek deney sonuçlarının ortalamasındaki mutabakat fonksiyonu kesinlik düzeyi diğer algoritmaların ortalama kesinlik düzeyinden yüksektir. Bu sonuca göre mutabakat fonksiyonu başarılı olduğu söylenebilir.

6.3.2. Lojistik Regresyon Sonuçları

Bu bölümde banka kredi bilgileri için yapılan SPSS programı kullanılarak lojistik regresyon çalışması açıklanmıştır. Çizelge 6.14'de bağımlı değişkeninin aldığı değerlerin SPSS işlem yaparken hangi değerlere çevrildiği gösterilmektedir .

Çizelge 6.14 Bağımlı Değişkenin Kodlaması

Gerçek Değer	**SPSS Değeri**
Evet	0
Hayır	1

SPSS ilk adımda (Adım 0) lojistik regresyon sabitinin katsayısını hesaplamıştır. Çizelge 6.15'de lojistik regresyon sabitinin katsayısı gösterilmektedir. SPSS sabit terimin katsayısını 0,802 olarak bulmuştur. Buradaki iterasyon 3. adımda bitmiştir. 3. adımda bitmesinin sebebi iterasyon değişkeninin değişimi 0,001'den küçük olmasıdır.

Çizelge 6.15 Sabit Terimin Katsayısının Hesaplanması

Adım 0		
Iterasyon	**Iterasyon Değişkeni**	**Sabitin Katsayısı**
1	887,609	0,762
2	887,358	0,802
3	887,358	0,802

Adım 0'da bulunan sabitin katsayısını Wald İstatistiği yöntemi ile test edilmesi gerekmektedir. Çizelge 6.16'da sabit terimin katsayısının denklem için

uygun olup olmadığının testi için Wald İstatistiği yönteminin sonucunu göstermektedir.

Çizelge 6.16 Sabit Terimin Uyumunun Wald İstatistiği Sonucu

	B	S.E.	Wald	Df	Sig(p)	Exp(B)
Adım 0 Sabit	0,802	0,081	98,550	1	0,000	2,230

Wald istatistiğinin sonucunda elde edilen p değerinin 0,05'den küçük olması durumunda %5 anlamlılık düzeyinde katsayının uyumlu olduğu söylenebilir. Çizelge 6.16'daki değer 0,05'in altında olduğu için katsayı lojistik regresyon için uyumludur. Çizelge 6.17'de bağımsız değişkenlerin katsayıları gösterilmektedir.

Çizelge 6.17 Bağımsız Değişkenlerin Katsayıları

		B	S.E.	Wald	Df	Sig(p)	Exp(B)
Adım 1	1.Değişken	0,642	0,199	10,427	1	0,001	1,901
	3.Değişken	0,358	0,196	3,331	1	0,048	1,431

Çizelge 6.17'de gösterildiği üzere Adım 1'deki Sig. (p) değerleri 0,05'in altındadır. Denklemi oluşturmak için Adım 1'deki 1. ve 3. değişken katsayıları kullanılmıştır.

6.4. Yetişkin Bilgileri Uygulama Sonuçları

3000 satırdan ve 7 sütundan oluşan veri seti üzerinde 250 satırlık rasgele örnek seçilmiştir. Bu örnek veri üzerinde Rapid Miner ve Whibo eklentisi kullanılarak ID3, C4.5, CHAID ve CART algoritmaları çalıştırılmıştır. Oluşan karar ağaçları tüm veriye uygulanarak tahmin sonuçları elde edilmiştir. Tahmin sonuçlarını gerçek verilerin sonuçları ile karşılaştırılarak karar ağaçlarının kesinlik düzeyleri

hesaplanmıştır. Bu çalışmaya ek olarak farklı boyutlardaki örnek veri setleri ile deneyler desteklenmiştir.

6.4.1. Karar Ağaçları ve Mutabakat Fonksiyon Sonuçları

Çizelge 6.18'de Karar Ağaçları ve lojistik regresyon analizi sonucunda bulunan mutabakat fonksiyonu sonuçları karşılaştırılmıştır. Bu çalışma örnek alınan 250'lik veri seti ile oluşturulan karar ağacı ve oluşturulan mutabakat fonksiyonu tüm veri setine uygulanmıştır.

Çizelge 6.18 Karar Ağaçları ve Mutabakat Fonksiyonu Kesinlik Düzeyleri

Yöntem		**Kesinlik Düzeyi**
Karar Ağacı Algoritmaları	CHAID	0,731
	CART	0,748
	C4.5	0,727
	ID3	0,722
Mutabakat Fonksiyonu		0,761

Mutabakat fonksiyonu ile karar ağaçlarına göre daha yüksek kesinliğe sahip bir sonuç ortaya çıkarılmıştır. SPSS ile bulunan mutabakat fonksiyonu aşağıda gösterilmiştir. Çizelge 6.19'da oluşturulan mutabakat fonksiyonu hata matrisi gösterilmiştir.

$$Mutabakat\ Fonksiyonu = 1{,}404 * CHAID + 0{,}466 * CART + 0{,}345 * C4.5 + 0{,}306 * ID3$$

Çizelge 6.19 Mutabakat Fonksiyonu Hata Matrisi

Gerçek Sonuç / Mutabakat Fonksiyonu	**Evet**	**Hayır**
Evet	441	412
Hayır	242	1653

Toplam 3000 satırdan 250 adet örnek veri çıkartılmıştır. 2750 satır içerisinde başarılı bulunan iki küme (Evet / Evet ve Hayır / Hayır) toplanarak 2094 satır başarılı bulunmuştur. 2 Adet satır için sonuç bulunamamıştır. Buradan yola çıkılarak;

$$\textit{Mutabakat Fonksiyonu Kesinlik Düzeyi} = 2094 / 2750 = 0{,}7614$$

olarak bulunmuştur.

Bu çalışmaya ek olarak, rasgele farklı örnek veri setleri seçilerek karar ağaçları yeniden oluşturulmuştur. Toplam veri üzerinden örnek veri setleri çıkartılıp kalan veri seti üzerinde karar ağaçları ve mutabakat fonksiyonu uygulanmıştır. Uygulama sonucunda kesinlik düzeyleri bulunmuş ve bulunan sonuçlar karşılaştırmalı olarak gösterilmiştir. Bulunan sonuçlar Çizelge 6.20'de gösterilmiştir. Çizelge 6.21'de yapılan ek deney sonuçlarının ortalaması gösterilmiştir.

Çizelge 6.20 Ek Deney Sonuçları

		Kesinlik Düzeyi				
Yöntem		**Deney 1**	**Deney 2**	**Deney 3**	**Deney 4**	**Deney 5**
Karar Ağacı Algoritmaları	**CHAID**	0,682	0,713	0,690	0,719	0,731
	CART	0,747	0,775	0,772	0,767	0,761
	C4.5	0,669	0,713	0,660	0,728	0,675
	ID3	0,678	0,695	0,695	0,724	0,731
Mutabakat Fonk.		0,765	0,775	0,772	0,784	0,787

Çizelge 6.21 Ek Deney Sonuçları Ortalaması

Yöntem		Kesinlik Düzeyi
Karar Ağacı Algoritmaları	CHAID	0,707
	CART	0,765
	C4.5	0,689
	ID3	0,704
Mutabakat Fonksiyonu		0,777

Yapılan ek deneylerin sonuçları dikkate alındığında, ek deney sonuçlarının ortalamasındaki mutabakat fonksiyonu kesinlik düzeyi diğer algoritmaların ortalama kesinlik düzeyinden yüksektir. Bu sonuca göre mutabakat fonksiyonu başarılı olduğu söylenebilir.

6.4.2. Lojistik Regresyon Sonuçları

Bu bölümde yetişkin bilgileri için yapılan SPSS programı kullanılarak lojistik regresyon çalışması açıklanmıştır. Çizelge 6.22'de bağımlı değişkenin SPSS tarafında kodlaması gösterilmektedir.

Çizelge 6.22 Ek Bağımlı Değişkenin Kodlaması

Gerçek Değer	SPSS Değeri
Evet	0
Hayır	1

SPSS ilk adımda (Adım 0) lojistik regresyon sabitinin katsayısını hesaplamıştır. Çizelge 6.23'de sabit terimin katsayısı gösterilmektedir. SPSS sabit terim katsayısını -1,141 olarak bulmuştur. Buradaki iterasyon 4. adımda bitmiştir. 4. Adımda bitmesinin sebebi iterasyon değişkeninin değişimi 0,001'den küçük olmasıdır.

Çizelge 6.23 Sabit Terimin Katsayısının Hesaplanması

Adım 0		
Iterasyon	**Iterasyon Değişkeni**	**Sabitin Katsayısı**
1	2665,982	-1,031
2	2660,603	-1,138
3	2660,599	-1,141
4	2660,599	-1,141

Adım 0'da bulunan sabit katsayısı değerini Wald İstatistiği yöntemi ile test edilmesi gerekmektedir. Çizelge 6.24'de sabit terim katsayısının lojistik regresyon için uygun olup olmadığının testi için Wald İstatistiği yönteminin sonucunu göstermektedir.

Çizelge 6.24 Sabit Terimin Uyumunun Wald İstatistiği Sonucu

	B	**S.E.**	**Wald**	**Df**	**Sig(p)**	**Exp(B)**
Adım 0 Sabit	-1,141	0,048	573,852	1	0,000	0,320

Wald istatistiğinin sonucunda elde edilen p değerinin 0,05'den küçük olması durumunda %5 anlamlılık düzeyinde katsayının uyumlu olduğu söylenebilir. Çizelge 6.24'deki değer 0,05'in altında olduğu için katsayı denklem için uyumludur. Çizelge 6.25'de bağımsız değişkenlerin katsayıları gösterilmektedir.

Çizelge 6.25 Bağımsız Değişkenlerin Katsayıları

		B	**S.E.**	**Wald**	**Df**	**Sig(p)**	**Exp(B)**
Adım 1	1.Değişken	2,088	,106	390,359	1	0,000	8,065
Adım 2	1.Değişken	1,727	0,127	186,051	1	0,000	5,626
	4.Değişken	0,674	0,133	25,795	1	0,000	1,962
Adım 3	1.Değişken	1,494	0,148	101,277	1	0,000	4,456
	2.Değişken	0,496	0,164	9,213	1	0,002	1,643
	4.Değişken	0,495	0,146	11,496	1	0,001	1,641
Adım 4	1.Değişken	1,404	0,155	81,701	1	0,000	4,073
	2.Değişken	0,466	0,165	8,012	1	0,005	1,594
	3.Değişken	0,345	0,176	3,858	1	0,049	1,413
	4.Değişken	0,306	0,177	3,990	1	0,042	1,358

Çizelge 6.25'de gösterildiği üzere Adım 4'deki Sig. (p) değerleri 0,05'in altındadır. Denklemi oluşturmak için Adım 4'deki 1., 2., 3. ve 4. değişken katsayıları kullanılmıştır.

6.5. Sonuçların Toplu Olarak Yorumlanması

Gerçekleştirilen çalışmalarda üç farklı veri seti üzerinde dört farklı karar ağacı algoritması kullanılarak elde edilen sonuçlar kaydedilmiştir. Karar ağaçları algoritmalarının sonuçlarından yola çıkarak en iyi kesinlik düzeyini verecek bir mutabakat fonksiyonu geliştirilmiştir. Elde edilen mutabakat fonksiyonu üç veri setine uygulandığında birkaç durum dışında karar ağaçlarının her birinin ürettiği kesinlik düzeyinden daha yüksek bir değer sağladığı deneysel olarak ispatlanmıştır. Bu durumda mutabakat fonksiyonunun, iş akışları için karar verilmesi noktasında, sistemin en iyileştirilmesine katkı sağladığı görülmüştür.

Yapılan ek deneylerde mutabakat fonksiyonu ve karar ağaçlarının kesinlik düzeyleri karşılaştırıldığında farklı sonuçlar elde edilmiştir. Bir kaç durumda mutabakat fonksiyonu kesinlik düzeylerinin karar ağaçları kesinlik düzeyi ile aynı değerde veya biraz daha kötü olduğu görülmüştür. Buna rağmen, ortalamalar alındığında mutabakat fonksiyonunun kesinlik düzeyinin her bir veri seti için başarılı olduğu söylenebilir.

Bir başka nokta ise karar ağaçlarının başarılı olduğu sonuçlar her bir deneyde farklı karar ağaçları algoritmalarına aittir, mutabakat fonksiyonu kullanıldığında is sistemin hangi karar ağacını seçmesi gerektiğini bilmesine gerek olmayacaktır.

7. BÖLÜM : DEĞERLENDİRME VE ÖNERİLER

İş akışları onay veya reddinin tahmini ile ilgili olarak farklı karar ağacı algoritmaları incelenmiştir. İş akışları kullanıcı verisi üzerinden farklı boyutlarda örnekler alınmıştır. Rapid Miner ve Whibo eklentisi ile birlikte veriler karar ağaçlarına dönüştürülmüştür. Bu çalışma sırasına ID3, CART, C4.5 ve CHAID karar ağaçları algoritmaları kullanılmıştır. Çalışma sırasında üç farklı veri seti kullanılmıştır. Kullanılan veri setleri sırasıyla, Seyahat Akışı, Banka Kredileri ve Yetişkin Bilgileridir.

Farklı veri setlerinden alınan örnekler ile ortaya çıkarılan karar ağaçlarının tahminlerini oluşturabilmek için tüm veri seti üzerinde karar ağaçlarını uygulayan Forecaster programı geliştirilmiştir. Forecaster programı ile karar ağaçları sonuçları kesinlik düzeyleri hesaplanmıştır.

Karar ağaçları kullanılarak Lojistik Regresyon analizi yöntemi ile her bir veri seti için mutabakat fonksiyonu bulunmuştur. Lojistik regresyon analiziyle geliştirilen mutabakat fonksiyonu ile daha yüksek kesinlik düzeyleri bulunması hedeflenmiştir. Her bir karar ağacı algoritmasının tek başına uygulanması ile elde edilebilen en iyi kesinlik düzeyinden daha iyi değer üreten mutabakat fonksiyonu elde edilmiştir. Bu durumda hedeflenen sonuca ulaşıldığı ifade edilebilir.

Takip edecek çalışmalarda tahmin kesinliğinin arttırılması için karar ağaçları dışındaki tahmin modelleri incelenecektir. Yeni tahmin modelleri ile oluşturulacak mutabakat fonksiyonlarının kesinlik düzeyleri farklı veri setleri için hesaplanacaktır.

KAYNAKLAR

1. Ethem Alpaydın, "Zeki Veri Madenciliği: Ham Veriden Altın Veriye Ulaşma Yöntemleri", Bilişim 2000 Eğitim Semineri, 2000.

2. Bhavani Thuraisingham, ''Web Data Mining & Applications in Business Intelligence & counter Terrorism'', CRC Press 2003, 2003.

3. Shin-Yuan Hung, Hsiu-Yu Wang, ''Applying Data Mining to Telecom Churn Management'', PACIS 2004 Proceedings, Paper 89, 2004.

4. Daniel T. Larose, ''Discovering Knowledge in Data: An Introduction to Data Mining'', John Wiley & Sons Inc., pp. 42-70, 2005.

5. Ercan Akkuş, "Designing a Customer Relationship Management Model for an Insurance Company", Yüksek Lisans Tezi, İstanbul, pp. 10-25, 2002.

6. Robert Nisbet, John Elder & Gary Miner, ''Handbook of Statistical Analysis & Data Mining Applications'', Academic Press, pp. 35-36, 2009.

7. Haldun Akpınar, "Veri tabanlarında bilgi keşfi ve veri madenciliği", İ.Ü. İşletme Fakültesi Dergisi, C:29, pp. 1-22, 2000.

8. J. E. Cook & A. L. Wolf, ''Discovering Models of Software Processes from Event-Based Data''. ACM Trans. Softw. Eng. Methodol. 7, pp. 215–249, 1998.

9.R. Agrawal, D. Gunopulos, F. Leymann, ''Mining Process Models from Workflow Logs'' In: Proceedings of the 6th International Conference on Extending Database Technology, Springer-Verlag, pp. 469–483, 1998.

10. J. Herbst & D. Karagiannis, ''An Inductive approach to the Acquisition & Adaptation of Workflow Models'', Proceedings of the IJCAI, 1999.

11. A. Weijters & W. van der Aalst, ''Process mining: discovering workflow models from event-based data'' In: Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), pp. 283–290, 2001.

12. E. Kindler,V. Rubin, & W. Schafer, ''Incremental Workflow mining based on Document Versioning Information'' In M. Li, B. Boehm & L. J. Osterweil, eds.: Proc. of the Software Process Workshop 2005, Beijing, China. Vol. 3840 LNCS, Springer, pp. 287–3019, 2005.

13. Stanislaw Matusik, ‘‘Modeling a Level of Development in Malopolskie Using Decision Trees’’, International Advances in Economic Research, 2005.

14. Gang-Zhi Fan, Seow Eng Ong & Hian Chye Koh, ‘‘Determinants of House Price: A Decision Tree Approach’’, Urban Studies, 2006.

15. E. Kindler, E., V. Rubin & W. Schafer, ‘‘Activity mining for discovering software process models’’ In B. Biel, M. Book & V. Gruhn, eds.: Proc. of the Software Engineering 2006 Conference, Leipzig, Germany. Vol. P-79 LNI., Kollen Druck,pp. 175–180, 2006.

16. Liu Yingbo, Jianmin Wang & Jiaguang Sun, ‘‘Using Decision Tree Learning to Predict Workflow Activity Time Consumption’’, ICEIS (2), pp.69-75, 2007.

17. Ning Fang & Jingui Lu, ‘‘Work in progress - a decision tree approach to predicting student performance in a high-enrollment, high-impact, & core engineerings’’, Education Conference, 2009.

18. Vineet Kumar Jain, ‘‘Decision Tree Approach Takes Multiple Business Interruption Risks Into Account’’, 2009.

19. Wei Hou, Bingru Yang, Chensheng Wu, Zhun Zhou & Wei Hou, ‘‘RedTrees: A relational decision tree algorithm in streams’’, Expert Systems with Applications, 2010.

20. J.Ross Quinlan, ‘‘Induction of Decision Trees’’; Mach. Learn. 1, vol. 1, pp.81-106, 1986.

21. J.Ross Quinlan, ‘‘C4.5: Programs for Machine Learning’’, Morgan Kaufmann Publishers, 1993.

22. Leo Breiman, Jerome Friedman, Charles J. Stone & Richard A. Olshen, ‘‘Classification & regression trees’’; Monterey, Calif., U.S.A.: Wadsworth, Inc. 1984.

23. Gordon V. Kass, ‘‘An Exploratory Technique for Investigating Large Quantities of Categorical Data’’, Applied Statistics, Vol. 29, No. 2, pp. 119–127, 1980.

24. Manish Mehta, Rakesh Agrawal & Jorma Rissanen, ‘‘SLIQ: A Fast Scalable Classifier for Data Mining’’, Advances in Database Technology, 1996.

25. Tom Fawcett, “An introduction to ROC analysis”, Pattern Recognition Letters, 27(8):861–874, 2006.

26. Kazım Özdamar, ‘‘Paket Programlar ile İstatistiksel Veri Analizi’’, Kan Kitabevi, 5.Baskı, 2004.
27. Ünal Sezer, ‘‘Karar Ağaçlarının Birliktelik Kuralı ile İyileştirilmesi’’, Kocaeli Üniversitesi Tez çalışması, 2008.
28. <http://archive.ics.uci.edu/ml/> erişim: 08.12.2011
29. <www.spss.com>
30. İş Akışı Yönetim Sistemi (EBİ)
<www.btvizyon.com.tr/download/ebi_06_03_15.ppt>
31. Mind2Biz Informatics - İş Akış Yönetimi,
<http://www.mind2biz.com.tr/corporate/other2.php>
32. U. Hauptmanns, ‘‘A decision-making framework for protecting process plants from flooding based on fault tree analysis’’, Reliability Engineering & System Safety, 2010.
33. F. Hammann, H. Gutmann, N. Vogt, C. Helma & J. Drewe, ‘‘Prediction of adverse drug reactions using decision tree modeling’’, Clinical Pharmacology & Therapeutics, 2010.
34. Ken Farion, Wojtek Michalowski, Szymon Wilk, Dympna O'Sullivan, Stan Matwin & Ken Farion, ‘‘A Tree-Based Decision Model to Support Prediction of the Severity of Asthma Exacerbations in Children’’, Journal of Medical Systems, 2010.
35. William J. Clancey, Maarten Sierhuisa & Chin Seaha, ‘‘Workflow agents versus expert systems: Problem solving methods in work systems design’’, Cambridge University Press, 2009.
36. N.R. Sakthivel, V. Sugumaran & S. Babudevasenapati, ‘‘Vibration based fault diagnosis of monoblock centrifugal pump using decision tree’’, Expert Systems with Applications, 2010.
37. Mehmet Seval Kaygulu, ‘‘Supervising & Unsupervising Techniques of Learning in Data Mining’’, 2008.
38. Elif Özge Özdamar, ‘‘Veri Madenciliği Teknikleri ve Bir Uygulama’’, 2002.
39. Wikipedia, http://www.wikipedia.com

40. Yao Jung Yanga, Tien-Wen Sunga, Chuni Wua & Hsiang-Yang Chena, ‘‘An agent-based workflow system for enterprise based on FIPA-OS framework’’, Expert Systems with Applications, 2009.

41. Taylor & Francis Group, "Business Process Management System (BPMS) Standards", 2006.

42. P.D. O'Brien & M.E. Wieg, ‘‘Agent based process management: applying intelligent agents to workflow’’, The Knowledge Engineering Review, Vol. 13:2, 1998.

43. J. Vaarkamp1, C.S. Hamilton, M. Escreetv & C. Percy, “Managing workflow in treatment planning using standard spreadsheet software”, Journal of Radiotherapy in Practice, vol. 7, pp. 213-221, 2008.

44. Y. Xiang, S.H. Zhang, Y.Z. Shen & M.L. Shi, ‘‘Pattern-Oriented Workflow Generation & Optimization’’, Journal of Universal Computer Science,Vol. 5:9; pp. 1924-1944, 2009.

ÖZGEÇMİŞ

Hacı Mehmet Yıldırım Koçdağ, 1980 yılında İstanbul Üsküdar'da doğdu. Öğrenimlerini sırasıyla Pendik Merkez İlkokulu, Eczacıbaşı Ortaokulu, MTS Lisesi'nde tamamladı. 2004 yılında Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümünden mezun oldu. 2006 yılında Maltepe Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans programını bitirdi. Yüksek Lisansı tamamladıktan sonra, yaklaşık 1 sene boyunca Sydney College Avusturalya'da çeşitli sertifikasyon programlarını tamamladı 2008 yılında Maltepe Üniversitesi MBA programını tamamladı. Son 9 senedir özel firmalarda çeşitli pozisyonlarda çalıştı. Çalıştığı firmalar, Philip Morris, BIS, Arçelik, Siemens ve Turkcell. Eylül 2008'de Maltepe Üniversitesi'nde başladığı Bilgisayar Mühendisliği Doktora programı tez aşamasında devam etmektedir.