

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BELİRLİ PERİYOTLARDA ATMLERDEKİ PARA
TALBİNİN BELİRLENMESİNDE VERİ
MADENCİLİĞİNİN KULLANILMASI**

YÜKSEK LİSANS TEZİ

Bil.Müh. Öner ÇELİK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Tez Danışmanı : Yrd. Doç. Dr. Nilüfer YURTAY

Eylül 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BELİRLİ PERİYOTLARDA ATMLERDEKİ PARA
TALEBİNİN BELİRLENMESİNDE VERİ
MADENCİLİĞİNİN KULLANILMASI**

YÜKSEK LİSANS TEZİ

Bil.Müh. Öner ÇELİK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Bu tez 11/09/2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



Prof. Dr. Emin GÜNDOĞAR
Jüri Başkanı



Doç. Dr. Cemil ÖZ
Üye



Yrd. Doç. Dr. Nilüfer YURTYAY
Üye

ÖNSÖZ

Bankalar gibi finansın merkezi olan kuruluşlarda nakit yönetimi kavramı her zaman önemli bir unsur olmuştur ve geçmişte olduğu kadar gelecekte de önemini koruyacaktır. Öncelerde insan gücü ve tecrübesi ile çözülmekte olan bu finans problemini; gelişen bilgisayar teknolojileri sayesinde, artık minimum insan gücü ve minimum maliyet ile çözmek mümkün olabilmektedir. Günümüzde veri tabanı ve veri madenciliği teknolojilerinin de karar destek yapılarında kullanılması ve başarılı sonuçlarının gözlenmesi, nakit yönetimi konusunda bu alanda bir eksiklik olduğu tespitini yapabilmemizi sağlamaktadır. VM teknolojilerinin, bu alanda kullanılması ile bankaların likiditelerinin daha verimli kullanmaları, müşteri memnuniyetinin artırılması, iş gücünün daha verimli kullanılması, güvenlik maliyetlerinin düşürülmesi gibi daha bir çok konuda katkılar sağlayacağı görülmektedir.

İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
TABLolar LİSTESİ	viii
ÖZET.....	ix
SUMMARY.....	x
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
VERİ MADENCİLİĞİ.....	5
2.1. Neden Veri Madenciliği.....	5
2.1.1. Müşteri kaybetme analizi (churn analyse).....	6
2.1.2. Çapraz satış (cross selling).....	6
2.1.3. Sahtekarlık tespiti (fraud detection).....	6
2.1.4. Risk yönetimi (risk management).....	7
2.1.5. Müşteri kümeleme (customer segmentation).....	7
2.1.6. Hedef kampanyalar (targeted ads).....	7
2.1.7. Satış tahmini (sales forecast).....	7
2.2. Veri Madenciliğinde Karşılaşılan Problemler.....	7
2.2.1. Veri tabanı boyutu.....	8
2.2.2. Gürültülü veri.....	9
2.2.3. Boş (null) değerler.....	9
2.2.4. Eksik veri.....	10

2.2.5. Artık veri.....	10
2.2.6. Dinamik veri.....	11
2.3. Veri Madenciliği Algoritmaları.....	11
2.3.1. Hipotez testi.....	12
2.3.2. Sınıflama algoritması.....	12
2.3.3. Kümeleme algoritması.....	12
2.3.4. Eşleştirme algoritması.....	13
2.3.5. Zaman serileri arasındaki bağımlılıklar.....	13
2.3.6. Sıra örüntüler.....	14

BÖLÜM 3.

ATM'LERDE NAKİT YÖNETİMİ.....	15
3.1. ATM Cihazları.....	15
3.2. Nakit Yönetimi.....	15
3.2.1. Karlılık açısından nakit yönetimi.....	17
3.2.2. Nakit akımının unsurları.....	17
3.2.2.1. Operasyonel (dahili) nakit akımı.....	17
3.2.2.2. Operasyon dışı (dahili) nakit akımı.....	18
3.2.2.3. Finansal (dış) nakit akımı.....	18
3.2.3. Etkin bir nakit yönetiminin sağlanması.....	18
3.2.4. Nakit yönetimi varlığının sağladığı yararlar.....	19
3.2.5. Nakit tahminleri uygulamalarının geliştirilmesinde karşılaşılan zorluklar.....	19
3.2.6. Tahmin yöntem ve teknikleri.....	20
3.2.6.1. Kısa dönem tahmin teknikleri.....	20
3.2.6.2. Uzun dönem tahmin teknikleri	21
3.2.6.3. Stratejik tahmin yararları.....	21
3.3. ATM'lerde Tutulacak Nakit Miktarının Belirlenmesinde Dikkat Edilecek Etkenler	22
3.3.1. ATM konumunun etkisi.....	22

3.3.2. Takvim etkisi.....	22
3.3.3. Mevsimsel etki.....	23
BÖLÜM 4.	
KURAMSAL ÇALIŞMA.....	24
4.1. Kullanılan Algoritmanın Seçilmesi.....	24
4.2. Karar Ağacı Algoritması.....	25
4.3. Karar Ağacının Budanması.....	30
4.4. Tez Kapsamında Kullanılan Algoritmalar.....	31
4.4.1.Sınıflandırma ve regresyon ağaçları (classification and regression trees– CART).....	31
4.4.2.Ki-Kare otomatik etkileşim dedektörü (Chi-squared automatic interaction detector - CHAID).....	32
BÖLÜM 5.	
UYGULAMA.....	34
5.1. Veri Kümesinin Tanımlanması.....	34
5.2. Karar Ağacı Algoritmalarının Veri Üzerinde Uygulanması.....	36
5.2.1. CART uygulaması.....	36
5.2.2. CHAID uygulaması.....	40
BÖLÜM 6.	
SONUÇLAR.....	44
KAYNAKLAR.....	46
ÖZGEÇMİŞ.....	48

SİMGELER VE KISALTMALAR LİSTESİ

ATM	:	Otomatik para Ödeme Makinesi (Automated Teller Machine)
VM	:	Veri Madenciliği
VTYS	:	Veri Tabanı Yönetim Sistemi
VT	:	Veri Tabanı
CART	:	Sınıflandırma ve Regresyon Ağaçları (Classification and Regression Trees)
CHAID	:	Ki-Kare Otomatik Etkileşim Belirleme (Chi-Squared Automatic Interaction Detector)
YSA	:	Yapay Sinir Ağları

ŞEKİLLER LİSTESİ

Şekil 2.1.	Zaman serileri arasındaki bağımlılıkları gösteren örnek grafik....	14
Şekil 4.1.	Örnek bir veri kümesi.....	27
Şekil 4.2.	Örnek veri kümesinin karar ağacına dönüştürülmesinin birinci adımı.....	28
Şekil 4.3.	Örnek veri kümesinin karar ağacına dönüştürülmesinin son adımı.....	29
Şekil 5.1.	CART algoritmasının ürettiği kural setinin bir kısmı.....	37
Şekil 5.2.	CART algoritması kural ağacının bir kısmı.....	38
Şekil 5.3.	CART algoritması tahminlemede kullanılan değişkenlerin önem dereceleri.....	39
Şekil 5.4.	CHAID algoritmasının ürettiği kural setinin bir kısmı.....	41
Şekil 5.5.	CHAID algoritmasının ürettiği ağaç yapısının bir bölümü.....	42
Şekil 5.6.	CHAID algoritması tahminlemede kullanılan değişkenlerin önem dereceleri.....	43

TABLULAR LİSTESİ

Tablo 5.1.	Eğitim setini oluşturan verilerin tablosu.....	35
Tablo 5.2.	Modellemede kullanılacak nitelikler tablosu.....	36
Tablo 5.3.	CHAID algoritması analiz sonuçları.....	40
Tablo 5.4.	CART algoritması analiz sonuçları.....	43

ÖZET

Anahtar kelimeler: ATM, nakit yönetimi, veri madenciliği

Finansal kurumlar özellikle de bankalar için çok önemli bir kavram olan nakit yönetiminin en iyi uygulanması gereken alanlardan biri de ATM'lerde tutulacak olan nakit miktarının optimize edilmesi problemidir. Bankalar için, ATM makinelerinde bekletilecek nakit, o paranın kullanılmaması ve olası getirilerinden yararlanılamaması demektir. Eksik miktarda para koyulması müşteri memnuniyetini kaybettireceği gibi ekstra maliyetlere sebep olacak, fazla nakit tutulması ise kuruluşu olası kardan edecektir.

Firmaların kuruluş amaçları temelde kar etmek olsa da bu amaç doğrultusunda, müşteri memnuniyeti, talep edilen hizmeti zamanında sunmak, ve en ekonomik biçimde sunmak gibi planlanması gereken yükümlülükleri vardır. Anlaşıldığı üzere tüm bu yükümlülükler temelde nakit yönetiminin en iyi şekilde yapılması yani otomatize edilmesi gereğini doğurmaktadır. Bu amaç için kullanılacak en iyi yöntemlerden biri ise Veri Madenciliğidir. VM, minimum maliyet ile maksimum verim alınmasını sağlayacaktır.

THE USING OF THE DATA MINING FOR DETERMINING THE OPTIMUM CASH AMOUNT WHICH IS HOLDING IN AN ATM IN A REGULAR PERIOD OF TIME

SUMMARY

Key Words: ATM, cash management, data mining

Cash management and optimizing which is the one of the most important milestone for the financial foundations, especially banks, have to implement without any mistake at arranging amount of money which should stored in ATMs. The meaning of the money which is holding in the ATMs that can not using it and missing its gains. If the money is less then the optimum amount it will cause to loss of satisfaction of customers, or more money then the optimum amount will cause to loss of the possible gains.

The main purpose of the foundations is earning much money so they have to provide satisfaction of their workers and customers, also they have to supply the products and services according to the criterion required by customer and on time with using the most economic way. It clearly seems that it is essential to automatize the cash management. The one of the best way of gain this purpose is using Data Mining. Data Mining can supply the maximum gain with minimum loss.

BÖLÜM 1. GİRİŞ

Bilgisayar teknolojilerinin geldiği noktalar, donanımların hızlanması ve fiyatların düşmesi, zaten bir çok alanda kullanılan bilgisayarların daha farklı alanlarda ve daha verimli kullanılmasını da beraberinde getirmiştir. Bu sayede her alanda bir çok veri hızla toplanabilmekte, toplanan veriler kısa bir zamanda büyük miktarlara ulaşmakta ve her geçen gün de artmaya devam etmektedir. Doğal olarak bu süreç, hızla biriken bu verilerin saklanması, saklanan verilerin işlenmesi, işlenen bu verilerin analiz edilmesi ve karar aşamalarında kullanılabilmesi gibi problemleri de beraberinde getirmektedir. Bu problemlerin çözümünde ise çoğu zaman kullanılan teknoloji veri tabanları (VT) olmuştur.

Bir VT içerisinde sınırlı sayıda ilişkisel veriyi saklayabileceği gibi milyonlarca veriyi de saklayabilmektedir. Hal böyle olunca da bu verilerin analiz edilmesi sürecinde insan gücünün kullanılması çok büyük bir zaman ve iş gücü kaybına yol açacaktır. İşte bu , verilerin toplanması, saklanması, analiz edilip hızlı bir şekilde operasyonel sonucun çıkarılması süreci, beraberinde Veri Madenciliği (VM) kavramını bir çözüm olarak karşımıza çıkarmıştır.

VM, geniş veri yığınları içerisinde, yararlı olma potansiyeline sahip, aralarında beklenmedik / bilinmedik ilişkilerin olduğu verilerin keşfedilerek, veri sahibi için hem anlaşılır hem de kullanılabilir bir biçime getirilmesine yönelik geliştirilmiş yöntemler topluluğudur. Bahsi geçen bu yöntemler karar verme sürecinde oldukça etkili rol oynamaktadırlar. Nihayetinde amaç bilgiyi keşfederek ona ulaşmak ve bu yolla fayda sağlamaktır. Veri madenciliği, aynı zamanda bir süreçtir. Veri yığınları arasında, soyut kazılar yaparak veriyi ortaya çıkarmanın yanı sıra, bilgi keşfi sürecinde örüntüleri ayrıştırarak süzmek ve bir sonraki adıma hazır hale getirmek de bu sürecin bir parçasıdır[1].

Büyük boyutlu yapısal veriyi saklama ve bu verilere etkin bir şekilde erişim sağlamakla yükümlü olan veri tabanı yönetim sistemlerinde (VTYS) veri düzenlemesi, ilgili organizasyonun işletimsel veri ihtiyacı doğrultusunda gerçekleştirilir. Bu işlem her zaman bilgi keşfi perspektifi ile birebir örtüşmez. Bu açıdan VM algoritması uygulanmadan önce veri ön işleme basamakları gerçekleştirilir. Veri tabanında bulunan veriler üzerinde gerçekleştirilen bu basamaklar kısaca şöyledir: temizleme, boyut indirgeme, tür dönüşümleri, transfer, vb işlemlerdir.

VM algoritmalarında kullanılan veriler genelde çok büyük boyutlu verilerdir. Bu nedenle verimli bir VM algoritması geliştirmek için öncelikle amaca uygun veriler toplanmalı, toplanan veriler etkin şekilde kullanılabilir, ölçeklenebilir veriler olmalı, bir birinden bağımsız olmalı ve bu şekilde algoritmaya tabi tutulmalıdır.

Veri Madenciliği yaklaşımları temel olarak iki grupta toplanmaktadır [2]:

- Kullanıcı güdümlü yaklaşım: Bilginin çıkarılması için yürütülen işlemler kullanıcının güdümündedir. Geliştirilen hipotezler mevcut veriler kullanılarak doğrulanmaya çalışılır. Eğer kural doğrulanmıyorsa, hipotezde değişiklik yapılarak doğru kurallar bulunmaya çalışılır. Kullanıcı güdümlü VM her alanda kullanılmasa da belirsizlik düzeyi düşük ortamlarda, konuya hakim uzman kullanıcılar tarafından geliştirilen hipotezlerle çok çabuk sonuç elde edilebilmektedir.

- Otomatik yaklaşım: Bu yaklaşım genelde yapay anlayış (Artificial Intelligence) ve makine öğrenmesinde kullanılan metotlardan yararlanır. Otomatik yaklaşımda kullanılacak teknikler verinin türü ve yapılacak işe göre değişmektedir. Otomatik yaklaşım kullanıcı güdümlü yaklaşımın aksine uzman kişilere daha az ihtiyaç duyar. Bu yaklaşım, uygulama zamanındaki yüksek maliyetli olması durumuna karşın, ortaya çıkarttığı değerli bilgilerden ötürü tercih edilir.

Tez kapsamında, bankacılık sektöründe çok önemli bir yere sahip olan nakit yönetimi (cash optimization) alanında ATM cihazlarında belirli periyotlarda (günlük, haftalık, vb..) tutulması gereken para miktarının VM yöntemleri ile belirlenmesi konusu işlenecektir. Günümüz ekonomik şartlarında nakit yönetiminin önemi anlaşılmalı olsa da genelde insan gücü ile yapılan, en iyi sonuca yaklaşılamadan

yapılan tahminler, hem mevcut nakdin kullanılamaması hem de fazladan bir takım maliyetlere yol açmaktadır. Sektörde bir açık olarak görünen bu problemin çözülmesi için tez kapsamında bir çalışma gerçekleştirilmiştir. VM'den faydalanılarak, ATM cihazlarında tutulması gereken optimum nakit miktarının tespit edilmesi bu tezin amacını oluşturmaktadır.

Günümüzde kullanılan nakit tahminleme yöntemlerine bakılacak olursa, ülkemizde yaygın olarak kullanılan bir yöntemin olmadığı görülmektedir. Daha çok geçmiş zaman verileri kullanılarak ve geçmişte elde edilen tecrübelerle dayanılarak çözüm üretilmesi yoluna gidilmiştir. Bu iş için bankalar ve diğer çeşitli finansal kuruluşlar bir ekip oluşturmuşlar ve nakit yönetimi işlemlerini bu ekiplere bırakmışlardır. Bilgisayar teknolojilerindeki gelişimin desteklendiği ve bu teknolojilerin günlük hayata yansıtılmaya özen gösterildiği bazı ülkelerde ise otomatik sistemlerin kullanılmaya başlandığı görülmektedir. Ve bu eğilim hızla yayılmaktadır.

Nakit yönetimi ve tahminleme alanında en bilinen ticari yazılımlar iCom (Carreker Corporation), MorphisCM (Morphis, Inc), Optica\$h, (Transoft International) and ProCash Analyser (Wincor Nixdorf) ürünleridir[13-14]. Fakat bu çözümlerinde zayıf kaldığı ve üstünde özenle durulması gereken noktalar vardır. Tahminleme için verileri algoritmaya sunmadan önce her bir makine kendine has özellikleri ile ele alınmalıdır. Her bir ATM'in kendisine has özelliklerinin olduğu(konum, mevsim, kültürel etkinlikler, vs..) düşünülerek tahminleme amacıyla kullanılacak olan veri seti esnek bir yapıda oluşturulmalı ve değişken girişlere izin verebilmelidir. yine tahminleme için kullanılan ve ATM'den bağımsız olan değişkenler de dinamik seçilmelidirler. Çünkü modellerde kullanılan veriler sabit olmasına rağmen, gerçek hayatta finansal veriler çok hızlı bir şekilde değişebilmektedir.

Mevcut çözümlerde bulunan eksiklerin giderilebilmesi ancak yeni bir tahminleme yöntemi kullanılarak mümkün olmaktadır. Kullanılacak algoritmalar için iki alternatif düşünülebilir; Daha esnek bir yapıya sahip yapay sinir ağı algoritmaları veya uygulaması diğerlerine nazaran daha basit ve algoritma olarak geliştirmeye daha elverişli olan(açık kodlu versiyonlarına internet kullanılarak rahatlıkla erişilebilmektedir) ve daha esnek veri girişlerine izin veren karar ağacı algoritmaları.

Bu teziz hazırlık aşamalarında bahsi geçen iki yöntem de denenmiş ve karar ağacı algoritmalarının daha başarılı sonuçlar verdiği gözlenmiştir. Burada unutulmaması gereken ise her algoritmanın mevcut verilerden oluşan setlere farklı tepkiler vereceğidir. Çalışan algoritmaların performansları uygulanacak olan veri setlerinin içeriklerine ve yapılarına bağlıdır.

BÖLÜM 2. VERİ MADENCİLİĞİ

Bu bölümde veri madenciliğinden, veri tabanında bilgi keşfinden, kısaca veri madenciliği algoritmalarından, veri madenciliği ön veri işlemeden ve veri madenciliğinde karşılaşılan sorunlardan bahsedilecektir. En son olarak veri madenciliğinde kullanılan araçlara değinilecektir.

Veri madenciliği, çok büyük veri tabanlarındaki yada veri ambarlarındaki veriler arasında bulunan ilişkiler, örüntüler, değişiklikler, sapma ve eğilimler, belirli yapılar gibi ilginç bilgilerin ortaya çıkarılması ve keşfi işlemidir. Veri madenciliğinde kullanılan yöntem ve araçlar, çok kısa zamanlarda işin niteliğine yönelik stratejik soruları cevaplamada yardımcı olurlar. Ham veride gizli kalmış olan örüntüleri ve ilişkileri tahmini bilgilere dönüştürebilirler[3].

2.1. Neden Veri Madenciliği

Gelişen teknoloji ile birlikte hızla düşmeye devam eden fiyatlar nedeniyle her sektördeki firmalar, teknolojiye daha çok yatırım yapmaya başladılar ve her alanda rekabet iyice arttı. Müşteri memnuniyetini sağlamak ve bunu yaparken de en az maliyetle en yüksek verimi almak temel amaç olarak belirlendi. Bankacılık sektörü de bu müşteri-firma ilişkisinin en sık yaşandığı alanlardan birisidir. Milyonlarca müşterisi olan yada gün içerisinde binlerce işlem gerçekleştirilen bankacılık sektöründe müşteri yatkınlıklarını takip etmek, ihtiyaçlarını takip edip çözümler üretmek, onlara özel kampanyalar düzenlemek ve beğenilerini kazanmak yada müşteriden bağımsız düşünürsek borsaların eğilimlerini takip etmek, gelecek için tahminlerde bulunup bir takım kararlar almak, parayı kullanarak para üretmek, eldeki

mevcut paranın en iyi kullanılmasını sağlamak gibi bir çok operasyon veriler ve bu verilerin işlenmesi ile gerçekleşmektedir. Öncelikle sağlıklı bir şekilde toplanması gereken bu verilerin en verimli şekilde kullanılabilmesi günümüzde bir takım bilgisayar programları ile mümkün olmaktadır. Bu programların temel kaynağı ise VM algoritmalarıdır. Evet VM, çok büyük veri yığınları içerisinde amaca yönelik keşfedilmemiş bilgiyi çıkarma işlemidir. VM teknolojilerinin çözüm ürettiği iş teknolojileri aşağıdaki gibi özetlenebilir.

2.1.1. Müşteri kaybetme analizi (churn analyse)

Telekom, bankacılık ve sigorta sektörleri günümüzde bu tehlike ile sürekli karşı karşıyalar. Churn analizi, şirketlere müşterilerinin neden başka şirketler ile çalışmak üzere göç ettiklerini anlamaları için yardımcı olur, müşteri ilişkilerini kuvvetlendirir ve sonunda müşteri sadakatini artırır. Bu teknik genel olarak “hangi müşterilerimiz rakip firmaya kaymaya daha meyilli” sorusuna cevap aramaktadır.

2.1.2. Çapraz satış (cross selling)

Ürün satan şirketler için çapraz satış önemli bir dinamiktir. Özellikle çevrimiçi(online) satıcılar, satışlarını arttırmak için bu tekniği kullanırlar. Örneğin çevrimiçi alışveriş yapmak için herhangi bir internet sitesi müşterilerine, “ilgilendiğiniz ürünümüzü alan müşterilerimiz bu ürünleri de aldı...” gibi bir başlık altında çeşitli tavsiyelerde bulunur. Alışveriş sitelerinde sıklıkla görünmeye başlayan bu özellik müşteri verilerini kaydetme daha sonra da kullanılacak anlamlı veriyi çıkarma operasyonudur. Bu tavsiyeler veri madenciliği sonucu çıkarsanabilir.

2.1.3. Sahtekarlık tespiti (fraud detection)

Sigorta şirketleri veya bankalar bir çok müşterisi vardır ve günde binlerce talebi işleme alırlar. Her birinin gerçekliğini ayrı ayrı araştırmak çok da mümkün değildir. Müşteriye en iyi ve en hızlı şekilde hizmet edebilmek için hızlı ve doğru karar almak gerekecektir. Veri madenciliği, gelen talebin sahte olup olmadığını tanımlamak ve kısa bir sürede geri dönüşün yapılabilmesinde yardımcı olabilir.

2.1.4. Risk yönetimi (risk management)

Bankacılıktaki en sık karşılaşılan sorulardan birisi müşterilerin kredi taleplerine verilecek cevaptır. Veri madenciliği teknikleri, müşterinin risk seviyesi derecelendirilerek kararın alınması süreci kolaylaştırılabilir. Böylece her müşteri için doğru kararın verilmesine yardımcı olunabilir.

2.1.5. Müşteri kümeleme (customer segmentation)

Müşteri kümeleme, satış yöneticilerinin farklı müşteri profillerini anlamaları, eğilimlerini takip ederek en iyi hizmeti sunmalarını sağlama konusunda yardımcı olacaktır.

2.1.6. Hedef kampanyalar (targeted ads)

Çevrimiçi satış yapan şirketler ve web portalları, müşterileri için içeriklerini özelleştirmekten hoşlanırlar. Müşterinin sayfalar ve ürünler arası geçişlerini ve satın alma modellerini kullanarak müşteriye uygun ürünlerin reklamlarını göstermek için bu siteler veri madenciliği çözümlerini kullanabilirler.

2.1.7. Satış tahmini (sales forecast)

Veri madenciliği tahmin teknikleri, zaman ilişkili sorulara cevap vermek için çok etkili yöntemlerdir. Geçmiş verilerin en iyi şekilde analiz edilmesi, mevcut şartların da algoritmaya tabi tutulması ile tüm olasılıklar hesaplanarak mümkün olan en iyi tahminin elde edilmesi sağlanır. Böyle maliyet en aza indirgenmiş olur.

2.2. Veri Madenciliğinde Karşılaşılan Problemler

VM büyük hacimli gerçek dünya verileriyle uğraşır. Veriler her zaman istenildiği kadar düzenli olmayacağından, bu durum kurulan algoritmalar açısından büyük sorunlar oluşturur. Mesela küçük veri setleriyle ve yapay hazırlanmış verilerle doğru

çalışan sistemler büyük hacimli, eksik, gürültülü, NULL değerli, artık, dinamik verilerle yanlış çalışabilir. Bundan dolayı bu sorunların aşılması gerekmektedir.

2.2.1. Veri tabanı boyutu

VM' de örnek veri setinin boyu ne kadar büyük olursa elde edilen sonuç da o kadar net ve iyi olur. Tabi ki daha çok veri ile çalışmak bazı dezavantajları da beraberinde getirecektir. Bunların başında da donanım ihtiyacı ve zaman problemleri gelmektedir. Tüm bu dezavantajlarının toplamına da maliyet gözüyle bakarsak, ne kadar iyi düzenlenmiş verilerle çalışırsak maliyet her açıdan düşecektir.

VT' de tutulan verilerin boyutu iki boyutlu olarak genişlemektedir:

- Yatay Boyut: Her bir nesnenin(tablo) özellik(sütun) sayılarıyla genişlemektedir.
- Dikey Boyut: Her bir nesnenin kayıt sayısı ile genişlemektedir.

Geliştirilen pek çok algoritma yüzler mertebesindeki verilerle uğraşacak şekilde geliştirildiğinden aynı algoritmanın yüz binlerce kat daha fazla kayıtlarla çalışabilmesi için azami dikkat gerekmektedir. Veri hacminin büyüklüğünden kaynaklanan sorunun çözümü için uygulanacak alternatif çözümlerden bazıları:

- Örneklem kümesinin yatay ve dikey boyutta indirgenmesi,
 - a) Yatay indirgeme: Nitelik değerlerinin önceden belirlenmiş genelleme sıradüzenine göre, bir üst nitelik değeri ile değiştirilme işlemi yapıldıktan sonra aynı olan çokluların çıkarılma işlemidir.
 - b) Dikey indirgeme: Artık niteliklerin indirgenmesi işlemidir.
- VM yöntemleri sezgisel/buluşsal bir yaklaşımla arama uzayını taramalıdır, vb.

Örneklem kümesinin geniş olması bulunacak örüntüleri ne kadar iyi tanımlıyorsa, bu büyük kümeyle uğraşma zorluğu da o kadar artmaktadır.

2.2.2. Gürültülü veri

Veri girişi veya veri toplanması esnasında oluşan sistem dışı hatalara gürültü denir. Veri toplanması esnasında oluşan hatalara ölçümden kaynaklanan hatalar da dahil olmaktadır. Bu hataların sonucu olarak VT' de birçok niteliğin değeri yanlış olabilir.

Günümüz ticari ilişkisel veri tabanları bu tür hataların ele alınması için az bir destek sunmaktadır. VM' de kullanılan gerçek dünya verileri için bu sorun ciddi bir problemdir. Bu sebepten dolayı VM tekniklerinin gürültülü verilere karşı daha az duyarlı olması gerekir. Sistemin gürültülü veriye daha az duyarlı olmasından kasıt, gürültülü verilerin sistem tarafından tanınması ve ihmal edilmesidir.

Sınıflama üzerine yaptığı çalışmalardan tanınan Quinlan'nın gürültünün sınıflama üzerine etkileri konusunda yaptığı çalışmada; etiketli öğrenmede etiket üzerindeki gürültünün öğrenme algoritmasının performansını doğrudan etkileyerek düşürdüğünü tespit etmiştir [4].

2.2.3. Boş (null) değerler

Eğer VT' de bir nitelik değeri NULL ise o nitelik bilinmeyen ve uygulanamaz bir değere sahiptir. VT' de birincil anahtar haricindeki herhangi bir niteliğin özelliği NOT NULL (NULL olamaz) şeklinde tanımlanmadığı sürece bu niteliğin değeri NULL olabilir. Kurulacak bir ilişkide kullanılacak verilerin aynı sayıda niteliğe ve NULL olsa bile aynı sayıda nitelik değerine sahip olması gerekir. Lee, NULL değerini ilişkisel veri tabanlarını genişletmek için aşağıdaki üç gruba girecek şekilde ayırmıştır [5]:

- bilinmeyen,
- uygulanamaz,
- bilinmeyen veya uygulanamaz.

Veri kümelerinde var olan NULL değerleri için çeşitli çözümler söz konusudur [4]:

- NULL değerli kayıtlar tamamıyla ihmal edilebilir,
- NULL değerli kayıtlardaki NULL değerleri olası bir değerle güncellenebilir.

Bu güncelleme için çeşitli yöntemler söz konusudur:

- a) NULL değeri yerine o nitelikteki en fazla frekansa sahip bir değer veya ortalama bir değer konulabilir,
- b) NULL değeri yerine varsayılan bir değer konulabilir,
- c) NULL değerinin bulunduğu kaydın diğer özelliklerine göre, NULL değerinin kendine en yakın değerle güncellenmesi sağlanabilir, vb.

2.2.4. Eksik veri

VM' de ilişkilerin kurulabilmesi ve istenen problemin çözümüne ulaşabilmek için gereken örneklem kümesindeki 2 boyutun (yatay ve dikey boyutlar) eksik olmaması gerekir. Bu boyuttaki eksiklikler şu şekilde olabilir:

- Yatay boyutta: Örneklem kümesinde olması gereken nitelik veya niteliklerin olmamasıdır. Örneğin: eğer insanların göz rengiyle alakalı bir hastalığın neye bağlı olduğu bulunmaya çalışılıyorsa, niteliklerden göz renginin örneklem kümesinde bulunması gerekir.
- Dikey boyutta: Dikey Örneklem kümesindeki kayıtların eksik olmasıdır. Örneğin bir süper markette yaşı 10 ve 25 yaşındaki kişiler her yaptıkları alışverişte bir ürünü sürekli alıyorsa, bu örüntünün keşfedilmesi için örneklem kümesinde yeterli sayıda 10-25 yaş aralığına giren kayıtların bulunması gerekir. Eğer örneklem kümesinde bu kayıtlar bulunmazsa gerçek hayatta var olan bir örüntü kaçırılmış olur.

2.2.5. Artık veri

Artık veri, problemde istenilen sonucu elde etmek için kullanılan örneklem kümesindeki gereksiz niteliklerdir. Artık nitelikleri elemek için geliştirilmiş algoritmalar, özellik seçimi olarak adlandırılır. Özellik seçimi arama uzayını küçültür ve sınıflama işleminin kalitesini de artırır [6].

2.2.6. Dinamik veri

İçeriği sürekli değişen veri tabanlarıdır. Bunlara örnek kurumsal çevrim-içi veri tabanları gösterilebilir. Bir veri tabanındaki içeriğin sürekli değişmesi VM uygulamalarının uygulanabilmesini önemli ölçüde zorlaştırıcı sorunlar doğurmaktadır. Bu sorunlardan bazıları şunlardır:

- Ortaya çıkan VM örüntülerinin sürekli değişim halinde olan verilerden hangisini ifade ettiğinin tespitinin zorluğu ve bu üretilen sonuçların zaman içinde eski üretilen sonuçlardan farkının tespiti ve gereken yerlerin güncellenme zorluğu,
- VM algoritmalarının çalışabilmesi için verilerin üzerine okuma kilidi(lock) konulması gerektiğinde, bu verilerin başka uygulamalar tarafından değişime açık olmaması,
- VM algoritmalarının ve çevrimiçi VT uygulamalarının aynı anda uygulanmasından kaynaklanan ciddi performans düşüşlerinin olması, vb.

2.3. Veri Madenciliği Algoritmaları

Veri madenciliği algoritmaları verilerde var olan bilgiyi anlaşılabilir kurallar olarak çıkartmaya yarayan metotlardır. Veri madenciliği algoritmaları genel olarak iki ana gruba ayrılır [7]:

- Doğrulamaya dayalı algoritmalar: Kullanıcı tarafından ispatlanmak istenen bir hipotez ortaya sürülür ve VM algoritmalarıyla bu hipotez ispatlanmaya çalışılır. Çok boyutlu analizlerde ve istatistiksel analizlerde tercih edilen metottur. Hipotez testi buna örnektir.
- Keşfe dayalı algoritmalar: Doğrulamaya dayalı algoritmaların tersine bu algoritmalarda ortada ispatlanması istenen hipotezler yoktur. Tam tersine bu algoritmalar otomatik keşfe dayanmaktadır. Keşfe dayalı algoritmaların birçok kullanım alanı vardır: istisnai durumların keşfi, karar ağacı, kümeleme gibi algoritmalar bu yaklaşıma göre kurulmuştur.

2.3.1. Hipotez testi

Hipotez testi algoritmaları doğrulamaya dayalı algoritmalar. Doğrulanacak hipotez VT üzerindeki verilerle belli doğruluk ve destek değerlerine göre sınırlanır.

Sınama işlemi uzman tarafından aşağıdaki ihtiyaçlardan dolayı yapılır:

- Bir kural ortaya çıkarılmak istendiğinde
- Ortaya çıkarılmış bir kuralın budanması veya genişletilmesinde.

2.3.2. Sınıflama algoritması

Sınıf olmak için her kaydın belli ortak özellikleri olması gerekir. Ortak özelliklere sahip olan kayıtların hangi özellikleriyle bu sınıfa girdiğini belirleyen algoritma, sınıflama algoritmasıdır. Sınıflama algoritması, denetimli öğrenme kategorisine giren bir öğrenme biçimidir. Denetimli öğrenme, öğrenme ve test verilerinin hem girdi hem de çıktığı içerecek şekilde olan verileri kullanmasıdır.

Sınıflama sorgusuyla, bir kaydın önceden belirlenmiş bir sınıfa girmesi amaçlanmaktadır. Bir kaydın önceden belirlenmiş bir gruba girebilmesi için sınıflama algoritması ile öğrenme verileri kullanılarak hangi sınıfların var olduğu ve bu sınıflara girmek için bir kaydın hangi özelliklere sahip olması gerektiği otomatik olarak keşfedilir. Test verileriyle de bu öğrenmenin testi yapılarak ortaya çıkan kurallar optimum sayısına getirilir.

Sınıflama algoritmasının kullanım alanları sigorta risk analizi, banka kredi kartı sınıflaması, sahtecilik tespiti, vb. alanlardır.

2.3.3. Kümeleme algoritması

Kümeleme algoritması denetimsiz öğrenme kategorisine giren bir algoritmadır. Kümeleme algoritmasındaki amaç verileri alt kümelere ayırmaktır. Sınıflama algoritmasında olduğu gibi ortak özellikleri olan veriler bir kümeye girer. Alt kümelere ayrılmak için keşfedilen kurallar yardımıyla bir kaydın hangi alt kümeye

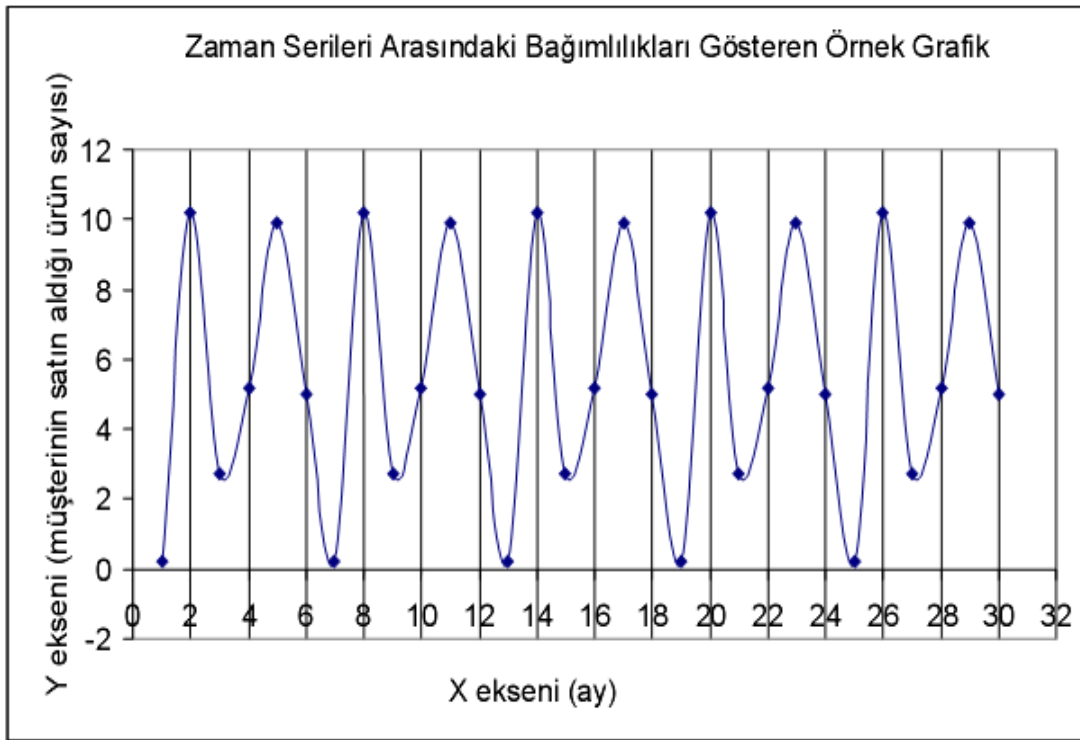
girdiği kümeleme algoritması sayesinde bulunur. Kümeleme algoritması genelde astronomi, nüfus bilimi, bankacılık uygulamaları, vb. uygulamalarda kullanılır.

2.3.4. Eşleştirme algoritması

Eşleştirme algoritması denetimsiz öğrenme kategorisine giren bir algoritmadır. Eşleştirme algoritması sınıflama algoritmasının benzeridir. Sınıflama algoritmaları ile eşleştirme algoritmaları arasındaki fark, eşleştirmede sınıflandırmada olduğu gibi bir sınıfa sokulması amaçlanmaz. Eşleştirmedeki amaç örneklem kümesindeki nesnelerin nitelikleri arasındaki ilişkilerin saptanmasıdır. Nitelikler arasındaki bütün kombinasyonlar çıkarılarak bütün niteliklerin farklı kombinasyonlarındaki farklı değerleri deneyerek örüntüler keşfedilmeye çalışılır[6]. Bu, ilişkilendirme algoritmasının sınıflandırma algoritmasından farklıdır. Her bir ilişkilendirme kuralı farklı ifadeleri verecek şekilde ortaya çıkar.

2.3.5. Zaman serileri arasındaki bağımlılıklar

Zaman serilerindeki örüntü belli bir periyotta, belli bir sıklıkta gerçekleşen olaylardır. Belli frekansla tekrarlanan bu olaylar zaman serileriyle yapılan VM algoritmaları sayesinde keşfedilir. Örneğin, müşteriler e-ticarette yazın yazlık ürünlere, kışın da kışlık ürünlere rağbet gösteriyorsa bu 6 ay periyotla tekrarlanan bir örüntüdür. Aşağıdaki örnek grafikte X eksenini sıfır zamanını referans alan ve geçen süreyi ay cinsinden ifade ediyorsa, Y eksenini müşterinin satın aldığı ürün sayısını gösteriyorsa bu grafikteki 6 ay süreyle tekrarlanan örüntü keşfedilebilmektedir.



Şekil 2.1. Zaman serileri arasındaki bağımlılıkları gösteren örnek grafik

Bu grafiğin insan gözüyle fark edilebilmesindeki sebep nitelik sayısının az, ve örüntünün fark edilebilecek türden basit olmasıdır. Oysa gerçek hayattaki verilerde genelde bu kadar basit örüntüler ve az nitelik sayıları bulunmaz.

2.3.6. Sıra örüntüler

Belli bir olayın bir başkasını izlemesi sıra örüntülerini oluşturur. Sıra örüntülerine örnek şu şekilde olabilir: Zayıflamak için diyetisyene giden bir kişinin daha sonra spor malzemeleri alması, ardından da bir baskül satın alması gibi. Sıra örüntüleri perakende satış, telekomünikasyon ve tıp alanında kendine geniş bir uygulama alanı bulmaktadır.

BÖLÜM 3. ATMLERDE NAKİT YÖNETİMİ

3.1. ATM Cihazları

ATM(Otomatik Para Çekme Makinesi)'ler yüksek miktarlarda nakit tutulan noktalardır. Gerekenden fazla para bulundurmak, kurumlara gecelik faiz kaybı getireceği gibi; talebin karşılanamaması da müşteri memnuniyetsizliği ve ciro kaybı ile sonuçlanır. Tüm bunların önüne geçmek için doğru bir şekilde tahminleme yapmak gerekmektedir. VM, ATM'lerde tutulması gereken optimum para miktarının tahminlenmesini sağlar.

Günümüzde, çoğu ATM'den sadece para çekilmekle kalmayıp artık para da yatırılabilir. Lakin yatırılan miktarlar toplanmadığı sürece kuruma fonlama maliyeti getirmektedir. VM ile ATM'lere yatırılacak para miktarları da tahmin edilebilir.

3.2. Nakit Yönetimi

Büyük küçük her türlü işletmenin karşı karşıya olduğu en önemli günlük sorunlardan bir tanesi, nakdin sürekli, etkili ve sağlıklı bir şekilde yönetimini sağlamaktır. Bu sorunun en önemli nedeni ise işletmelerimizde henüz bir "Nakit Akımı Yönetim" bilincinin gelişmemiş olmasıdır. Bu nedenle çoğu zaman nakit akımı ile ilgili günlük problemlerin yöneticilerin çok büyük bir zamanını aldığını ve genellikle de elde edilen sonuçlarında istenildiği ölçüde etkili olmadığını

görülmektedir. Yöneticiler etkin bir nakit yönetimi sistemi yardımı ile günlük ihtiyaçları karşılamaya yönelik dar kapsamlı faaliyetlerle uğraşmaktan kurtularak, nakit ihtiyaçları ve/veya fazlalıkları ile ilgili olarak daha stratejik kararları alabilme yeterliliğine ulaşacaklardır.

Nakit yönetimi günlük gelişmelerle oluşturulacak bir politika değildir. Etkin bir nakit yönetim sistemi günlük değişikliklere uyum sağlayabilmek için yeterli ölçüde esnek olmalıdır. Ancak bu söz konusu politikaların günlük olarak belirlenmesi anlamına kesinlikle gelmemektedir. Nakit yönetimi bir işletmenin faaliyetlerine sağlıklı bir şekilde devam edebilmesi için olmazsa olmaz bir şarttır. Ancak bu şekilde stratejik anlamda mevcut konumu korumak ve gelişimi sürdürmek mümkün olabilecektir.

Nakit akımı yönetimi stratejilerine değinmeden önce kısa bir şekilde temel kavramlara açıklık getirmeye çalışalım. Nakit; bankada ve/veya kasamızda bulunan ve kullanımında herhangi bir engel olmayan parasal tutarları ifade etmektedir. Bu noktada "Kar" ile "Nakit" arasındaki ayırımı dikkat çekmek zannediyorum yararlı olacaktır. Kar ile nakit birbirinden farklı kavramlardır. Kar tüm müşterilerin yapmasını beklediğiniz ödemelerin zamanında gerçekleştirilmesi ve giderlerinde beklenen ölçüde gerçekleştirilmesi sonucunda elde edilmesi planlanan parasal tutarı ifade etmektedir. Ancak kar kavramı işletmelerin günlük realitelerinden uzaktır. Nakit, kar elde etmek için uğraş verirken işletmemizi açık tutabilmek için elde bulunması gereken bir kavramdır. Başka bir deyişle nakit işlerin devam edebilmesi ve dolayısıyla kar yapabilmek için elde bulunması gereken minimum parasal büyüklükleri ifade etmektedir. Bu anlamda firmaların elde ettikleri karlar eğer pozitif net nakit akışına dönüşemiyor ise pek fazla değerleri bulunmamaktadır[10].

Karlar harcanamaz, sadece nakit harcanabilir. Uygulamada çoğu kez karlı olmasına rağmen nakit sıkışıklığı ile karşılaşan işletmelere rastlanılmaktadır[10].

3.2.1. Karlılık açısından nakit yönetimi

Nakit yönetimi basit olarak belirli bir zaman biriminde işletmeye giren ve çıkan nakdin izlenmesini ifade etmektedir. Nakit akımının izlenmesi ve kontrol altında

tutulması yönetimlerin önem vermeleri gereken temel konulardan bir tanesidir. Nakit yönetimi esas olarak gelecekle ilgilidir ve tahminlere dayanmaktadır. Bu nedenle sağlıklı bir nakit yönetiminden söz edebilmek için öncelikle etkin bir tahmin sisteminin geliştirilmesi ve uygulanması gerekmektedir. Elde edilen sonuçlarına göre nakit yönetimini ikiye ayırmak mümkündür;

- Pozitif Nakit Akımı : Eğer işletmeye girmesi beklenen nakit tutarı işletmeden çıkması beklenen nakit tutarından fazla ise firma pozitif nakit akımına sahiptir. Pozitif nakit akımı, nakit fazlasının nasıl değerlendirileceği bilindiği sürece işletme için son derece olumlu bir durumu ifade etmektedir.
- Negatif Nakit Akımı : Eğer işletmeye girmesi beklenen nakit tutarı işletmeden çıkması beklenen nakit tutarından daha az ise firma negatif nakit akımına sahiptir.

3.2.2. Nakit akımının unsurları

Nakdin ve nakit kaynaklarının sağlıklı bir şekilde takip edilebilmesi için nakit yönetimi sürecini üç aşamada incelemek yararlı olacaktır.

3.2.2.1. Operasyonel (dahili) nakit akımı

Operasyonel nakit akımı, genellikle çalışma sermayesi olarak adlandırılmaktadır. Dahili ve temel faaliyetlerden yaratılan nakit akımını ifade etmektedir. Temel olarak hizmet ve ürün satışlarından elde edilen nakdi ve bunlarla ilgili yapılan temel nakit çıkışlarını kapsamaktadır. İşlerin devam edebilmesi için son derece önemlidir. Büyük bir oranda yönetimin kontrolü altındadır.

3.2.2.2. Operasyon dışı (dahili) nakit akımı

Operasyon dışı nakit akımı temel faaliyetler dışında elde edilen ve elden çıkartılan elde edilen nakdi ifade etmektedir.

3.2.2.3. Finansal (dış) nakit akımı

Yatırımcılar, borç verenler ve hisse senedi sahipleri gibi dış kaynaklardan elde edilen veya ödenen nakit tutarları bu başlık altında değerlendirilmektedir.

3.2.3. Etkin bir nakit yönetiminin sağlanması

Etkin bir nakit yönetiminin sağlanması için dikkat edilmesi gereken temel unsurlar şu şekilde sıralanabilmektedir:

- Nakit girişi gerektiren tüm gelir kalemleri ve nakit çıkışı gerektiren gider kalemleri işletmenin özellikleri dikkate alınarak ayrıntılı bir şekilde belirlenmelidir.
- Etkin bir "Nakit Akımı Tahmin" sistemi ve süreci oluşturulmalı, söz konusu sistemin işletmenin genelinde uygun bir şekilde uygulanması sağlanmalıdır.
- Nakit yönetimi süreçleri ve bunların ürünleri gelişmelere uyum gösterebilecek şekilde esnek olarak yapılandırılmalıdır.
- Beklenmedik ani nakit çıkışlarına karşı gerekli güvenlik marjları her zaman düşünülmeli ve dikkate alınmalıdır.
- Genel ekonomik gelişmeler ve bunların mevcut nakit yönetimi politikaları üzerine olabilecek potansiyel etkileri zamanlı ve verimli bir şekilde izlenmelidir.
- Parasal satın alma gücünde yaşanacak düşüşler (enflasyon) yapılacak çalışmalarda dikkate alınmalıdır.
- Borç profilinin faiz ve diğer şartlar açısından iyileştirilmesi için yeniden yapılandırılmasına çalışılması.
- Tasarruf ve yatırımlar için potansiyel kaynakları tanımlamak.
- Önemli nitelikteki harcamalar için tahminde bulunmak ve bu harcamaları karşılayabilecek fon kaynaklarını belirlemek.
- Nakit yönetimi ile ilgili tüm taraflar (alacaklar, borçlar, tahsilat, pazarlama, satın alma vb.) arasında gerekli düzeyde bilgi alışverişinin sağlanarak etkinliğin artırılması.

3.2.4. Nakit yönetimi varlığının sağladığı yararlar

Bu yararları şu şekilde sıralayabiliriz:

- Yanlış finansman kararlarını önlemek
- Daha karlı yatırım enstrümanlarının seçilmesinin sağlanması
- Fonlama maliyetlerini azaltmak
- Verimliliği arttırmak
- Kredibilitiyi arttırmak
- Firma içi iletişimi geliştirmek

3.2.5. Nakit tahminleri uygulamalarının geliştirilmesinde karşılaşılan zorluklar

- Bir tahmin sürecinin geliştirilememesi ve/veya mevcut sistemlerin uygulanmalarına yeterince önem verilmemesi
- İzole edilmiş, durgun tahminler. Genellikle finans bölümleri bu konuda etkili olmaktadır. Bu nedenle genellikle diğerlerinin görüşlerine yeterince önem verilmemektedir.
- Tahmin teknik ve araçlarının gelişmemesi. İşletmede yaşanan önemli büyümeler, yapıda meydana gelen değişiklikler ve yeni iş alanlarının eklenmesi gibi değişiklikler sonrasında tahmin tekniklerinde gerekli düzenlemelerin zamanında hayata geçirilmemesi.
- Tahmin sistemlerinin aşırı karmaşık olması.

3.2.6. Tahmin yöntem ve teknikleri

Tahmin yöntem ve tekniklerinin seçilmesi ve uygulanması unutulmaması gereken en önemli nokta sadece tek bir "En İyi" yolun bulunmadığıdır. Yapılacak tahminlerden maksimum faydanın alınabilmesi için işletmenin, içerisinde bulunan iş kolunun ve genel ülke şartlarının bir şekilde anlaşılması gerekmektedir. Özellikle temel amaçlardan bir tanesinin de nakit çıkışlarını disiplin altına almak olduğu unutulmamalıdır. Temel Tahmin Yöntem ve Teknikleri ise şunlardır;

- Kısa Dönem Tahmin Teknikleri (haftalık, aylık)
- Uzun Dönem Tahmin Teknikleri (yıllık)
- Uzun Dönem Tahmin Teknikleri (stratejik)

3.2.6.1. Kısa dönem tahmin teknikleri

İşletmelerin günlük, haftalık ve en fazla aylık nakit giriş ve çıkışlarını izlediği tahminlerdir. İşletmelere kısa zamanlı yatırım ve borçlanma konularında kolaylıklar sağlamaktadır.

- Tahsilatlar ve Ödemeler Yöntemi: Hazine birimi nakit girişleri ve çıkışları için ayrı planlar hazırlamaktadır. Nakit girişleri müşterilerden gerçekleştirilecek tahsilatlara ilişkin tahminleri (nakit tahsilatı ve/ veya alacaklardan tahsilat) ve kar payı tahsilatları ve faiz gelirleri gibi diğer tahsilatları kapsar. Demirbaş ve varlık satışlarından beklenen nakit tutarları da bu plan içerisinde yer almaktadır. Ödemeler tahminlerinde ise alımlar ve ücretler, vergiler, borç ödemeleri, kira ve kar payı ödemeleri gibi diğer nakit çıkışları yer almaktadır. Tahminleri gerçekleştirenler ise bu ayrı planları birleştirerek pozitif veya negatif durumu belirlemekte ve buna bağlı olarak da borçlanma ve yatırım ihtiyaçlarını tespit etmektedirler.
- Dağıtım Yöntemi: İşletmenin yapısına uygun olarak çeşitli kriterlere göre belirlenmiş olan zaman dilimleri için belirlenen nakit akım tahminlerinin günlük temele yayılması esasına dayanmaktadır.

3.2.6.2. Uzun dönem tahmin teknikleri

Uzun dönem tahmin teknikleri yıllık nakit akımlarının hazırlanmasına yardımcı olmaktadır. Yıllık bütçeleme faaliyetlerinin önemli destekleyicisi olmakla birlikte aylık periyotlarla gözden geçirilmeli ve güncelleştirilmelidirler. Uzun dönem tahmin tekniklerinin en önemlileri "Satışların Yüzdesi veya Proforma Yöntemi" ve "Düzeltilmiş Net Gelir Yöntemi"dir[10].

Satışların Yüzdesi veya Proforma Yöntemi: Finansal tablo tahminleri gelecekteki beklenen satış rakamlarına ve geçmişte satışların diğer bilanço kalemleri ile olan ilişkileri temel alınarak belirlenmektedir. Çalışma sermayesi kalemlerinin hemen

hemen çoğu satışlar ile belli oranlar dahilinde ilişkilidir. Bu yöntemde tahmin yapanlar proforma bilançoları oluşturmakta ve eğer toplam varlıklar borç ve öz sermaye toplamından fazla ise nakit fazlası aksi takdirde ise nakit ihtiyacı olacağı belirlenmektedir. Söz konusu yöntemin en önemli eksikliği satışlar tutarına son derece bağlı olmasıdır[10].

Düzeltilmiş Net Gelir Yöntemi: Planlanan nakit kaynakları ve nakdin kullanımı finansal tablolar yardımı ile hesaplanmaktadır. Net nakit akımının hesaplanabilmesi için nakit çıkışı gerektirmeyen giderler ve bilanço hesaplarındaki değişiklikler net gelir tahminine eklenmektedir. Nakit kaynakları faaliyetlerden sağlanan nakit, varlıkların elden çıkartılmasından elde edilen nakit ve borçlarda meydana gelecek artışlardır. Nakit kullanımları ise yeni varlıkların alınması, borç geri ödemeleri, sermaye harcamaları ve kar payı ödemeleridir. Nakit kullanımlarından nakit kaynakları düşürüldüğünde elde edilen net nakit akımıdır[10].

3.2.6.3 Stratejik Tahmin Yararları

- Ne zaman pozitif nakit akımına ulaşılacağı
- İşe yeni başlanıyorsa, yatırım harcamalarının ne zaman geri elde edileceği
- Kar payı ve borç geri ödeme yeterliliğinin belirlenmesi
- Gelişim trendleri ve nakit akımına olabilecek potansiyel etkileri
- Yeni yatırım kararlarının alınabilmesi (ne zaman ve ne kadar)

3.3. ATM'lerde Tutulacak Nakit Miktarının Belirlenmesinde Dikkat Edilecek Etkenler

ATM'lerden çekilen para miktarları dağılımları incelendiğinde ortaya çıkan resmi anlamlandırabilmek için iniş çıkışların nedenlerini tespit etmek gerekir. Nedenler; 'ATM konumunun etkisi', 'takvim etkisi' ve 'mevsimsel etki' olarak başlıca 3 sınıfa ayrılabilir.

3.3.1. ATM konumunun etkisi

Bilindiği üzere ATM'ler dış mekanlarda (ör: cadde üzeri) konumlanabildiği gibi iç mekanlarda da bulunabilir (ör: alışveriş merkezi). Peki konum bilgisinin çekilen para miktarları üzerinde nasıl bir etkisi vardır? Şöyle ki, hava koşullarına bağlı olarak insanların davranış biçimleri değişkenlik gösteriyor. Çoğumuz yağmurlu bir günde alışveriş merkezinde gezinmeyi seçerken güneşli havada dış mekanları tercih ediyoruz. Haliyle bu da ATM kullanım oranları üzerinde etkili oluyor. ATM konumunu etkili bir faktör kılan nedenlerden biri de çevresel özellikler; civardaki etkinlikler. Stadyum yakınındaki bir ATM'nin maç günlerinde, eğlence mekanları yakınındaki bir ATM'nin Cuma akşamlarında yoğun kullanılması konum bilgisinin çekilen para miktarı üzerindeki etkisini gösterir. O halde, tahminleme yapılırken ATM konumu mutlaka hesaplamaya dahil edilmelidir.

3.3.2. Takvim etkisi

Birkaç yıllık para çekim miktarı verisi tarih bazında incelendiğinde dağılımda bazı tekrarlar olduğu dikkat çekecektir. Veri ayrıntılı kırılımda değerlendirildiğinde o günün takvimsel özelliğinin para çekim miktarına doğrudan etkisi olduğu sonucuna varılır. Örneğin bayram ve yılbaşı arifeleri gibi zamanlarda para çekim miktarları diğer günlere oranla oldukça yüksektir. Maaş ödemesi de periyodik olarak ATM'lere olan talebi arttıran bir takvim etkisidir. Verinin sergilediği dağılıma bakılarak hafta içi/hafta sonunun, ayrı ayrı günlerin, aybaşının, aysonunun vb. etkisi tespit edilebilir. Tahmin yapılacak günler için de para miktarı günün takvimsel özelliğine bağlı olarak hesaplanmalıdır.

3.3.3. Mevsimsel etki

Kimi zaman ATM konumu ve takvim etkisi birbirinden bağımsız değerlendirilemez ve bu ikisinin ATM'den çekilen para miktarına olan etkilerini beraberce ele almak gerekir. Konum ve takvim etkisi birlikte değerlendirildiği zaman ortaya dönemsel görülen mevsim etkisi çıkar. Buna verilebilecek en iyi örnek yaz aylarında tatil yörelerinde yaşanan talep yoğunluğudur. Antalya'daki bir ATM'den Ocak ve Temmuz aylarında çekilen para miktarları karşılaştırıldığında mevsimsel etki çok net

ortaya çıkar. Burada göz önünde bulundurulması gereken aynı anda hem ATM'nin bulunduğu yer (tatil yöresi), hem de tarih (Temmuz ayı) bilgisidir. Özetle, ATM'ler için nakit miktarı tahminlemede tarih ve mekan bilgilerinin birbirinden bağımsız düşünülmesi mümkün değildir.

BÖLÜM 4. KURAMSAL ÇALIŞMA

4.1. Kullanılan Algoritmanın Seçimi

Mevcut çözümler incelendiğinde, kurulan sistemlerin, değişken verilerle çalışabilme potansiyelinin düşük olduğu gözlenmektedir. Nakit tahminlemesi bir çok durağan ve değişken etkene bağlı olarak ele alınması gereken bir problemdir. Her ATM'in kendine has koşullar altında değerlendirilebilmesi gerektiği gibi, algoritmanın işletilmesinde kullanılacak olan diğer verilerin de zaman içerisinde değişebileceği (gecelik faiz oranları, hava şartları, vs..) göz önünde bulundurulmalıdır. Bu şartlar altında, kullanılacak olan algoritmanın seçimindeki baş aktör esneklik olarak karşımıza çıkmaktadır.

İhtiyaç duyulan esnekliği sağlayabilecek, uluslararası kabul görmüş olan iki algoritma çözüm olarak düşünülmüştür. Nakit tahminlemesini etkileyen faktörler arasındaki ilişkiyi kurup, yorumlayabilecek olan Yapay Sinir Ağları algoritması ilk çözüm olarak karşımıza çıkmaktadır. Problemle ilgili düşünülebilecek diğer bir yöntem ise Karar Ağacı Algoritmalarıdır. Karar ağacı algoritmaları, veri setleri hazırlanırken, değişken verilerin kullanılmasına olanak sağlayarak daha doğru bir tahminlemeye gidilmesini sağlamaktadırlar. Kolay uygulanabilirlik özelliği ile de her bir ATM için değişecek olan o anki mevcut şartlarında hızlıca hesaplanabilmesini sağlamaktadırlar.

Algoritma seçiminde dikkat edilmesi gereken bir diğer husus da mevcut verilerdir. Bir algoritma kullanılacak olan bir 'x' veri setinde en iyi sonucu verirken, daha farklı bir 'y' veri setinde en kötü sonucu verebilmektedir. Bu nedenle algoritma seçimi

yapılırken mevcut veriler düşünölmeli ve bu veriler arasındaki ilişkileri en iyi şekilde ortaya koyacak bir yöntemin seçilmesi gerektiđi unutulmamalıdır. Çünkü temelde tüm algoritmaların yaptığı işlem aynıdır: geçmiş verilerin incelenmesi, aralarındaki ilişkilerin ve bağımlılık derecelerinin ortaya konulması, ardından da muhtemel sonuçların üretilmesidir.

Bu tezin hazırlık aşamalarında, bir test seti oluşturularak, YSA ve karar ağaçları algoritmalarına bu set uygulandığında, karar ağacı algoritmalarının daha net sonuçlar ürettiđi görölmüştür. Bu nedenle bu tez kapsamında karar ağacı algoritmaları kullanılacaktır.

4.2. Karar Ağacı Algoritması

Karar ağaçları, ağaç şeklinde sınıflandırıcılardır. Bu ağaçtaki her düğüm bir yaprađı veya karar düğümünü belirtir. Yaprak düğümü hedef niteliđin deđeridir. Karar düğümü ise, bir nitelikte uygulanacak olan test deđeridir, bu düğümü, o niteliđe ait olan tüm olası nitelik deđerleri izler, bu deđerler ise ağacın dallarını oluşturur. Karar ağacı, bir örneđi, kökten yaprađa kadar inceleyerek sınıflandırır. Karar ağaçlarının öğrenme algoritmaları, bir hipotezi sunmak için bir küme karar ağacı kullanırlar. Öğrenme kümesinde, ham veri incelenerek mümkün olan en iyi şekilde sınıflandırılır. Algoritma bu işlemi öz devinimli (recursive) olarak tekrar eder ve en son ortaya çıkardığı karar ağacı en son hipotezi oluşturur. İdeal olan karar ağacı, öğrenme kümesi dışındaki verilerde de aynı kuralları oluşturur yada az hata payıyla aynı hipotez sonuçlarını ortaya çıkartır [8].

Aşađıda, bir karar ağacı algoritması gösterilmektedir [9] :

Algoritma: Karar-Ağacı-Oluştur

Giriş Deđişkenleri : Örnek kümesi, nitelik-listesi

Metot :

1. N düğümünü oluştur
2. Eğer *örnek* deđerlerinin tümü aynı sınıftaysa (Sınıf C)
 - N düğümünü C etiketinde bir yaprak düğümü olarak döndür
3. Eğer nitelik-listesi boşsa

- N düğümünü Örnek kümesindeki en fazla kullanılan sınıf etiketinde bir yaprak düğümü olarak döndür

4. Nitelik-listesi'nden en fazla bilgi kazancı test-niteliği'ni seç

5. N düğümünün etiketine test-niteliği ismini ver

6. Test-niteliği'nin her bir bilinen değeri için dön

- N düğümünden test-niteliği=ai olacak şekilde bir dal çiz

7. $s_i = \text{örnek}$ kümesinde test-niteliği= ai olan örnekler olsun

8. Eğer s_i boşsa

- Örnek kümesi içinde en fazla kullanılan sınıf etiketinde bir yaprak çiz

9. Değilse Karar-Ağacı-Oluştur(s_i , nitelik-listesi, test-nitelik)' dan dönen düğüm değerini ağaca ekle.

Yukarıda belirtilen algoritma, ID3 algoritmasının bir versiyonudur, ve karar ağacı indirgenmesi (decision tree induction) olarak da bilinir. Temel strateji şu şekildedir [9]:

1. Ağaç, örnek kümesinin tek bir düğüm ile temsil edilmesiyle başlar.

2. Eğer örnek kümesindeki tüm örnek değerleri aynı sınıfa (yani hedef değere) gidiyorsa, bu düğüm bir yaprak haline gelir ve bu hedef değerinin adını alır. Ağaç, yaprak değerinden sonra o yönde daha fazla uzamaz.

3. Diğer koşulda, algoritma nitelik-listesinden bilgi kazancı en yüksek olan niteliği seçer ve bu nitelik, test-niteliği haline gelir.

4. Test niteliğine ait olan tüm olası değerler ve bunları oluşturan örnek kümesi değerleri, bir dal şeklinde ağaca eklenir. Burada önemli olan nokta, bu değerlerin sürekli değişken şeklinde olmamasıdır. Ağacın boyutlarının kontrol edilebilir olabilmesi için, sürekli değişkenler kategorik değişken haline çevrilmiş olmalıdır.

5. Algoritma bundan sonraki aşamada, yinelemeli olarak her örnek kümesi değeri için yaprak düğümüne ulaşana dek program yeni örnek kümeleri ve yeni nitelik listesi ile kendi içinde döndürülür.

6. Yinelemeli döndürme sadece aşağıdaki koşullarda durur :

- Tüm örnek küme hedef değerleri aynı sınıfa aitse
- Örnekleme devam edebileceği bir nitelik listesi kalmamışsa
- Örnekleyecek değerler kümesi kalmamışsa.

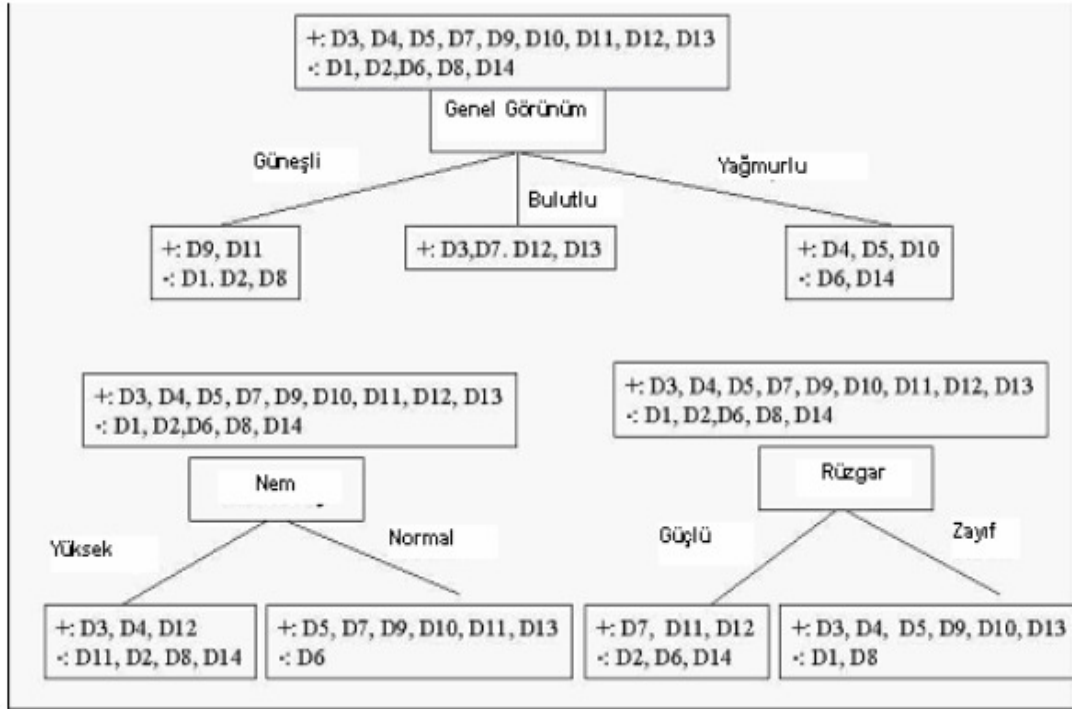
Algoritmayı daha iyi anlayabilmek için aşağıdaki örnekten faydalanılabilir :

Elimizde 14 adet örnekten ve 3 adet nitelikten oluşan bir veri kümesi olsun. Bu kümenin değerleri Şekil 3.1’de görülebilir. Havanın durum kombinasyonlarına göre, hedef değeri olan “tenis oynama/oynamama” durumu en sağdaki kolonda listelenmiştir.

Örnek	Nitelikler			Hedef Tenis Oynama
	Genel Görünüm	Nem	Rüzgar	
D1	Güneşli	Yüksek	Zayıf	Hayır
D2	Güneşli	Yüksek	Güçlü	Hayır
D3	Bulutlu	Yüksek	Zayıf	Evet
D4	Yağmurlu	Yüksek	Zayıf	Evet
D5	Yağmurlu	Normal	Zayıf	Evet
D6	Yağmurlu	Normal	Güçlü	Hayır
D7	Bulutlu	Normal	Güçlü	Evet
D8	Güneşli	Yüksek	Zayıf	Hayır
D9	Güneşli	Normal	Zayıf	Evet
D10	Yağmurlu	Normal	Zayıf	Evet
D11	Güneşli	Normal	Güçlü	Evet
D12	Bulutlu	Yüksek	Güçlü	Evet
D13	Bulutlu	Normal	Zayıf	Evet
D14	Yağmurlu	Yüksek	Güçlü	Hayır

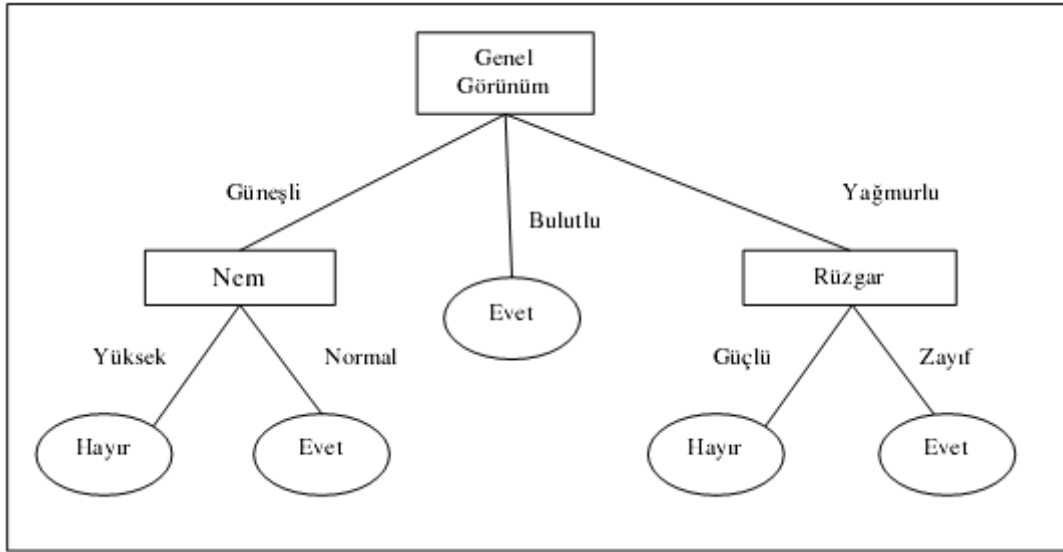
Şekil 4.1. Örnek bir veri kümesi

Karar ağacı algoritması, bu üç nitelikten en yüksek bilgi kazancı olan değeri “Genel Görünüm” olarak belirler. Bu sebeple, ilk önce tüm örnek kümesi değerleri “Genel Görünüm” niteliği altında gruplanmak üzere etiketlenir. Bu niteliğin olası tüm değerleri için etiketin altına birer dal çizilir (örnekte bu değerler Güneşli / Bulutlu /Yağmurlu). Bu değerlere sahip olan tüm örnekler de etiketlerin altında kümeler halinde tutulur.



Şekil 4.2. Örnek veri kümesinin karar ağacına dönüştürülmesinin birinci adımı

Sekil 3.2’de görüldüğü gibi, her bir daldan sonra, aslında aşağıda ağacın diğer alt-ağaçları oluşmaktadır. Örneğin, “Genel Görünüm” niteliği “Bulutlu” değerini aldığı anda, tüm örnek değerler aynı hedef değerine gittiklerinden (Tennis oynanır : Evet) , “Bulutlu” değerinin altında bir alt-ağaç oluşmamış, artık bu değer bir yaprak haline gelmiştir. Ancak “Güneşli” değeri için henüz örnekleme kümesinin tümü aynı değere gitmediğinden ve nitelik listesinin de tüm elemanları kullanılmamış olduğundan, ağaç oluşturma işlemi devam edecek demektir. Bu durumda, “Genel Görünüm”tan sonra “Güneşli” değeri için o örnek kümesi içinden hangi niteliğin daha fazla kazançlı bilgi sağlayacağını program hesaplar, ve bunun “Nem” olduğuna karar verir. Nem alt-ağacı da oluşturulduktan sonra, görüldüğü gibi, tüm değerler aynı hedef değerine gittiğinden bu alt-ağaç da burada noktalanır. Oysa “Genel Görünüm” niteliğinde “Yağmurlu ” değeri için bir alt ağaç çizilmeye çalışıldığında, bilgi kazancının maksimum olduğu nitelik bu kez Nem değil, “Rüzgar” niteliği olacaktır. ağacın aldığı son hali aşağıdaki şekilde görülebilir :



Şekil 4.3. Örnek veri kümesinin karar ağacına dönüştürülmesinin son adımı

Karar ağaçlarında ayırt edici olan nokta, test-niteliğin seçiminde kullanılan bilgi kazancıdır. Bu değer, ağacın ayrılış noktalarındaki verimliliği temsil eder. Algoritmada, en yüksek bilgi kazancı değeri olan nitelik, test-nitelik olarak seçilir. Bu nitelik, seçilen örnek kümesinin sınıflandırılması için gereken bilgi boyutunu minimize eder. Bu bilgi teorisi merkezli yaklaşım, bir objenin sınıflandırılmasında kullanılan test sayısını en küçük hale getirerek daha basit (ama en basit olmayan) bir ağaç yapısı ortaya çıkartır.

S, s adet veri örneğinden oluşan bir küme olsun. Cif ($i=1, \dots, m$) ise, bir niteliğe ait m adet değerlerin tanımlı sınıf değerleri olsun. s_i , Cif sınıfında bulunan S örneklerinin sayısı olsun. Bir örnek kümesini sınıflandırmak için kullanılan bilgi miktarının beklenen değeri, aşağıdaki formülle hesaplanır :

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (4.1)$$

Bu formüldeki p_i , bir örneğin Cif sınıfında bulunma olasılığıdır ve s_i/s değerine eşittir. A niteliğine ait v adet değer bulunsun $\{a_1, a_2, \dots, a_v\}$. A niteliği, S'yi v adet altkümeye ayıkabilir, $\{S_1, S_2, \dots, S_v\}$. Eğer, A test niteliği olarak seçildiyse, o zaman bu alt kümeler, S kümesini içeren düğümün birer dalı olacak demektir. s_{ij} , S_j alt kümesinin Cif sınıfında bulunan örneklerin sayısı olsun. Entropy, yada A alt

kümelerine bölünmede beklenen bilgi miktarı su şekilde hesapla Entropy, yada A alt kümelerine bölünmede beklenen bilgi miktarı su şekilde hesaplanır :

$$E(A) = \sum_{j=1}^v \frac{(s_{1j} + \dots + s_{mj})}{s \cdot I(s_{1j}, \dots, s_{mj})} \quad (4.2)$$

Entropy değeri küçüldükçe, alt küme bölünmelerinin saflık derecesi artar.

$$I(s_{1j}, \dots, s_{mj}) = -\sum_{i=1}^m p_{ij} \log_2(p_{ji}) \quad (4.3)$$

A niteliğinin üzerinden dallanmadan kazanılan bilgi kazancı ise :

$$Gain(A) = I(s_{1j}, \dots, s_{mj}) - E(A) \quad (4.4)$$

Başka bir deyişle, Gain(A), A niteliğinin değeri bilindiğinde Entropy'de beklenen düşüşü simgeler.

4.3. Karar Ağacının Budanması

Bir karar ağacı oluşturulduğunda, birçok dalda, öğrenme verisindeki gürültü ve kayıplardan dolayı anomaliler oluşacaktır. Ağacın budanma metodu, bu sorunu ortadan kaldırmaya yardımcı olabilir. Bu metoda, tipik olarak en az güvenilir olan dalı istatistiksel olarak hesaplayıp kaldırmaktan ibarettir ve daha hızlı ve güvenilir bir sınıflandırma ile sonuçlanır. İki adet budama yöntemi vardır.

Bunlardan birincisi, önceden-budama yöntemidir. Bu yöntemde öğrenme verisi sınıflandırılırken ağacın o dalının ileriye yönelik devam edip etmeyeceğine önceden karar verilir ve gerekiyorsa, geri kalan bölünmeden sonra geriye kalan verinin sınıflandırılması durdurularak, en fazla hedef değeri taşıyan değer yaprak yapılır. Bu yöntemde, önceden bir eşik değeri belirlenir. Bu eşik değerini aşmayan bilgi

kazançlarına sahip olan nitelikler gruplandırılır. Program devam ederken bu bilgi kazancının düştüğü noktada ağacın büyümesine izin verilmeden diğer dala geçilir.

Her iki koşulda da, bu eşik değerini belirlemek isin en zor kısmıdır. Çünkü eşik değeri çok yüksek tutulursa ortaya çıkan ağaç çok fazla basit ve genel kurallardan oluşan bir ağaç olur. eşik değeri çok düşük tutulursa ise, ağacın sınıflandırması çok özele inebilir ve test verisi üzerinde doğru sonuçlar ortaya çıkmayabilir.

İkinci yöntem, sonradan-budama yöntemidir. Tamamen büyümüş bir ağaç üzerinde uygulanır. Tüm dalların çıkardığı kurallar denenerek, bunlardan en fazla hata oranını oluşturan dal budanır. Böylece ortaya daha basit bir ağaç yapısı çıkartılabilir [9].

Alternatif olarak, hem önceden hem de sonradan budama yöntemi birleştirilerek, yeni bir yöntem olarak kullanılabilir. Sonradan-budama yöntemi önceden-budama'ya göre çok daha fazla hesaplama gerektirir, ancak ortaya daha güvenilir bir ağaç çıkartır.

4.4. Tez Kapsamında Kullanılan Algoritmalar

Tez kapsamında karar ağacı algoritmasını esas alan iki farklı yöntem kullanılmıştır.

4.4.1. Sınıflandırma ve regresyon ağaçları (classification and regression trees– CART)

Bu algoritma, ağaç yapısına dayalı olarak sınıflandırma ve regresyon modellerinin türetilmesi için yaygın olarak kullanılan bir istatistiksel prosedürdür. CART ağaç modeli, tek değişkenli ikili kararların bir hiyerarşisini içerir. CART verileri iki alt kümeye ayırdığı için her bir alt küme içindeki durumlar, bir önceki alt kümeden daha homojen olacaktır. Bu ardışık süreç, homojenlik kriterine ulaşıncaya veya diğer bazı durma kriterleri sağlanıncaya değin kendini tekrar eder. Aynı kestirim değişkeni ağaçta farklı düzeylerde pek çok kez kullanılabilir. Ağacın yapısı önceden belirlenmemekte, verilerden türetilmektedir. CART, kök düğümünde, verilerin iki gruba bölünmesi için, en iyi değişkenin seçilmesini sağlar ve farklı bölümlendirme

(splitting) kriterleri kullanır. Bu bölümlendirme kriterlerinin tümü, her bir alt kümedeki sınıf etiketlerini mümkün olduğunca homojen olacak biçimde bölümlendirir. Bölümlendirme prosedürü çocuk düğümlere (child node) veya alt düğümlerin her birine ardışık olarak uygulanır.

CART ağaçları, kesin bir heterojenlik (impurity) ölçüsüne bağlı olarak düğümlere ayrılmış iki değerli (binary) ağaçlardır ve bu nedenle de sonuçta homojen dallar oluşmaktadır. Ağacın hedefi benzer veya aynı çıktı değerlerine sahip olma eğiliminde olan alt gruplar yaratmaktır. CART modelleri için bölünmelerin bulunmasında kullanılan dört farklı heterojenlik ölçüsü mevcuttur. Kategorik hedef değişkenler için Gini, Twoing veya (sıralayıcı hedef değişkenleri için) sıralı Twoing, sürekli hedef değişkenler için ise en küçük kareli sapma (LSD) kullanılabilir.

CART analizi ağaç yapısına dayalı diğer sınıflama teknikleri ile kıyaslandığında çok sayıda avantaja sahiptir. İlki ve belki de en önemli olan özelliği parametrik olmayışıdır. Diğer bir söyleyişle ön kestirici veya aynı anlama gelmek üzere bağımsız değişken değerlerine ilişkin varsayımlar gerektirmemektedir. Bu nedenle CART analizinde kullanılacak değişkenler çok çarpık sayısal değişkenler olabileceği gibi, sınıflayıcı veya sıralayıcı yapıya sahip kategorik değişkenler de olabilir. Bu önemli bir özelliktir ve analizi yapacak araştırmacıya, normallik araştırma ve dönüşüm yapma gibi işlemler gerektirmediğinden zaman kazandırmaktadır. CART analizi, ele alınan problem yüzlerce mümkün bağımsız değişken içerse bile, bölümlendirilecek tüm mümkün değişkenleri araştırma gücüne sahiptir[12].

4.4.2. Ki-Kare otomatik etkileşim dedektörü (Chi-Squared automatic interaction detector - CHAID)

Bu algoritma, optimal bölünmelerin tespiti için ki-kare istatistiğini kullanan bir yöntemdir. CHAID, bölümlendirme amaçlı kullanılan etkili bir istatistiksel tekniktir. Bir istatistiksel testin anlamlılığını kriter olarak kullanarak, bir potansiyel ön kestirici değişkenin tüm değerlerini değerlendirir. Hedef değişkene veya aynı anlama gelmek üzere bağlı değişkene göre istatistiksel olarak homojen (benzer) olarak değerlendirilen tüm değerleri birleştirir ve diğer tüm değerleri heterojen (benzer olmayan) olarak değerlendirir. Ardından karar ağacındaki ilk dalın formuna göre en

iyi ön kestirici deęişkenin seçilmesiyle, her bir düęümün seçilen deęişkenin homojen deęerlerinin bir grubunu oluşturmasını sağlar. Bu süreç ardıl olarak ağaç tamamıyla büyüyene kadar sürer. Kullanılan istatistiksel test, hedef deęişkenin ölçüm düzeyine bağlıdır. Eğer hedef deęişken sürekli bir deęişken ise, F testi kullanılır. Eğer hedef deęişkeni kategorik ise ki-kare testi kullanılmaktadır[12].

BÖLÜM 5. UYGULAMA

5.1. Veri Kümesinin Tanımlanması

Nakit optimizasyonu modellemelerinde zor olan taraf, birbiriyle ilgisi yokmuş gibi görünen bir çok verinin aynı amaca yönelik birleştirilmesi ve sonunda bir çıkarım yapılmasıdır. Çok değişik başlıklardan(Müşteri davranışları, mevsimsel özellikler, konum etkisi, tarihsel etki, vb.) bir çok veri bu amaçla kullanılabilir.

Güçlü bir model oluşturabilmek için işin amaç ve hedefleri doğrultusunda ve doğru miktarda veri ile çalışılmalıdır. Fazla veri algoritmayı yorup, sonucun sapmasına neden olabileceği gibi, eksik veri de doğru bir modelin oluşturulmasını önleyebilir.

Tablo 5.2’de, model oluşturulurken kullanılan nitelikler ve kısa açıklamaları yer almaktadır. Tabloda belirtilen niteliklerden ID, ATM numarasıdır, ve bankanın güvenlik stratejileri doğrultusunda maskelenerek kullanılmıştır.

Tabloda verilen nitelik bilgilerine sahip toplam 32181 adet kayıt kullanılmıştır. Bu kayıtlardan 26842 adedi eğitim için, 5339 adedi de test amaçlı olarak kullanılmıştır. En doğru modelin karar aşamasında kullanılması amacı ile veriler değişik algoritmalar ile denenmiştir.

Kullanılacak olan veriler üzerinde amaca yönelik olarak türetmeler veya kırpmalar yapılması sonucun netliği açısından önemlidir. Bu nedenle yapılmak istenen tahminin periyodunun belirlenmesi gerekmektedir. Örneğin Aylık bir tahmin yapılacaksa veriler bu hedefe yönelik şekillendirilmeli, haftalık, günlük veya saatlik bir tahmin yapılacaksa veriler bu şekilde değerlendirilmelidir. Tez kapsamında

günlük bazda tahmin yapacak bir model oluşturulmuş ve bu nedenle veriler bu hedefe yönelik olarak tekrar düzenlenmiştir. Bu kapsamda; TUTAR ve ADET verilerine kümeleme işlemi yapılarak günlük bazda kayıtlar oluşturulmuştur. Yine bu kapsamda SAAT verisi günlük bazda tahmin yapılmak istendiğinden ve TÜR verisi de yapılan işlem türünden daha çok çekilen işleme odaklanılması amacıyla modellemeden çıkarılmıştır. Fakat günlük bazda hangi tür işlemin ne kadar gerçekleşeceği gibi bir tahminleme yapılmak istenirse TÜR verisi yeniden modelleme eklenmek zorundadır.

Tablo 5.1. Eğitim setini oluşturan verilerin tablosu

Sıra No	Nitelik Adı	Açıklama
1	ID	Maskelenmiş ATM numarası
2	MEVSİM	İşlemin yapıldığı mevsim bilgisi. I: İlkbahar, Y: Yaz, S: Sonbahar, K: Kış
3	HD	İşlemin yapıldığı tarihteki hava durumu bilgisi. A:Açık, K:Kapalı-Yağışlı
4	HS	Hafta sonu bilgisi. E: Evet, H: Hayır
5	AB	Ay başı bilgisi. E: Evet, H: Hayır
6	AS	Ay sonu bilgisi. E: Evet, H: Hayır
7	TÜR	İşlem tipi. 10: Para Çekme, 12: Nakit Avans
8	SAAT	Saat aralığı bilgisi.
9	ADET	Belirli saat aralığında yapılmış toplam para çekme işlemi adedi.
10	TUTAR	Belirli saat aralığında çekilen para miktarı.

Tablo 5.2. Modellemede kullanılacak nitelikler tablosu

Sıra No	Nitelik Adı	Açıklama
1	ID	Maskelenmiş ATM numarası
2	Mevsim	İşlemin yapıldığı mevsim bilgisi. I: İlkbahar, Y: Yaz, S: Sonbahar, K: Kış
3	HD	İşlemin yapıldığı tarihteki hava durumu bilgisi. A:Açık, K:Kapalı-Yağışlı
4	HS	Hafta sonu bilgisi. E: Evet, H: Hayır
5	AB	Ay başı bilgisi. E: Evet, H: Hayır
6	AS	Ay sonu bilgisi. E: Evet, H: Hayır
7	Adet	Günlük toplam yapılan para çekme adedi.
8	Tutar	Günlük toplam çekilen para miktarı.

5.2. Karar Ağacı Algoritmalarının Veri Üzerinde Uygulanması

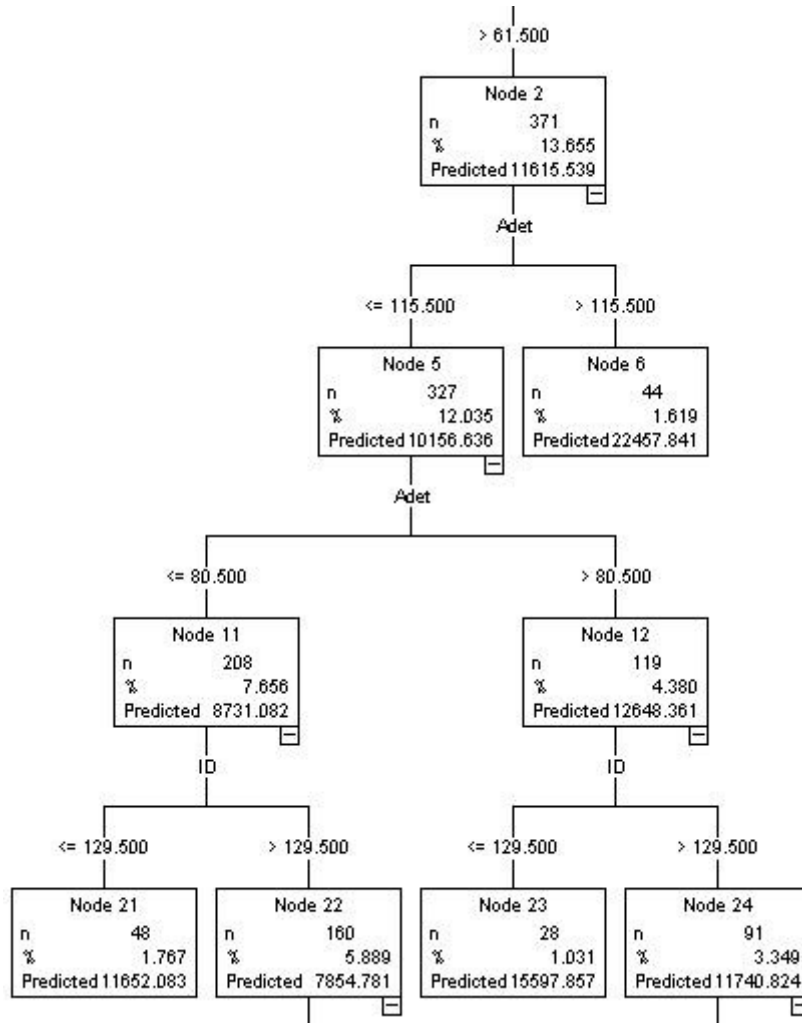
5.2.1. CART uygulaması

Algoritma öncelikle, eğitim seti verileri üzerinde uygulanmıştır. Gerekli model oluşturulup, algoritma işletilmeye başlandığında öncelikle, program tarafından, bir kural seti üretilmektedir. Şekil 5.1’de algoritma doğrultusunda üretilen kural setinin bir kısmı görülmektedir.



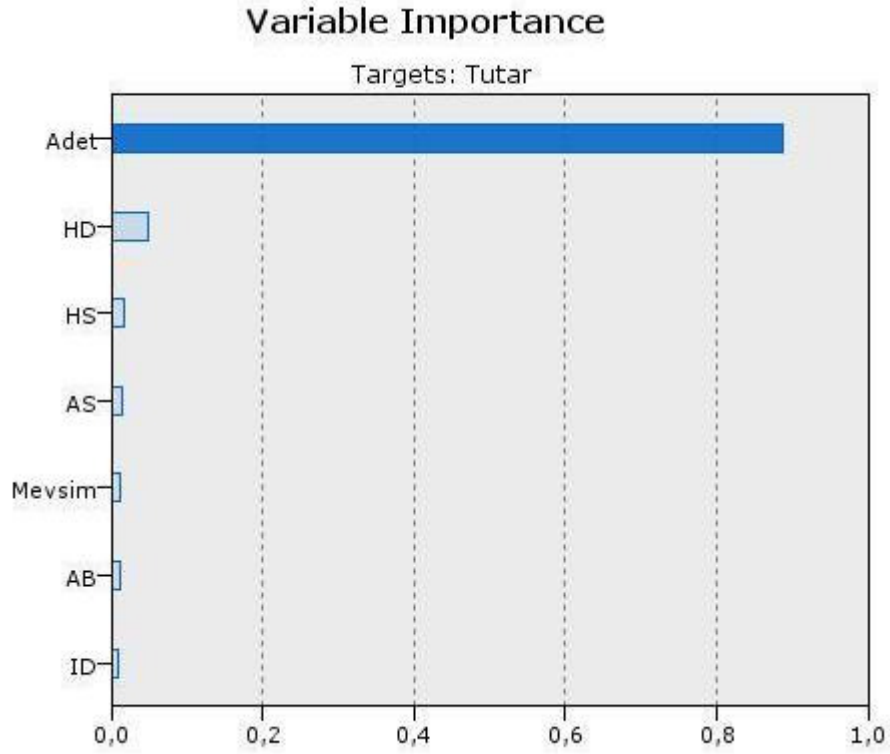
Şekil 5.1 CART algoritmasının ürettiği kural setinin bir kısmı

Oluşturulan kurallar doğrultusunda elde edilen ağaç görünümü Şekil 5.2' de görülmektedir.



Şekil 5.2 CART algoritması kural ağacının bir kısmı

Tahminleme de kullanılan veriler ve program tarafından belirlenen öncelik dereceleri Şekil 5.3'de görülmektedir.



Şekil 5.3 CART algoritması tahminlemesinde kullanılan değişkenlerin önem dereceleri

Tablo 5.3' de oluşturulan modele test seti uygulandıktan sonra alınan analiz sonuçları görülmektedir. Analizde tahmin edilen tutar değeri ile test setinde yer alan tutar değerleri karşılaştırılarak tablodaki değerler türetilmiştir. En küçük hata (Minimum Error) ve en büyük hata değerleri (Maximum Error), TL cinsinden yapılan tahminin gerçek değerlerle olan farkını vermektedir. Mutlak hata değeri (Mean Absolute Error) değerinin de 0'a yakın olması istenir. Yine aynı şekilde Standart sapma (Standard Deviation) değerinin 0'a yakın olması ve lineer korelasyon değerinin de 1'e yakın olması istenir. Bu değerler karar ağacının ürettiği kuralların doğruluk derecelerini göstermektedir.

Tablo 5.3. CART algoritması analiz sonuçları

En Küçük Hata Değeri	-12
En Büyük Hata Değeri	116
Mutlak Hata Değeri	0,028
Standart Sapma	0,425
Lineer Korelasyon(Linear Correlation)	0,941
Uygulama Adedi(Occurrences)	561

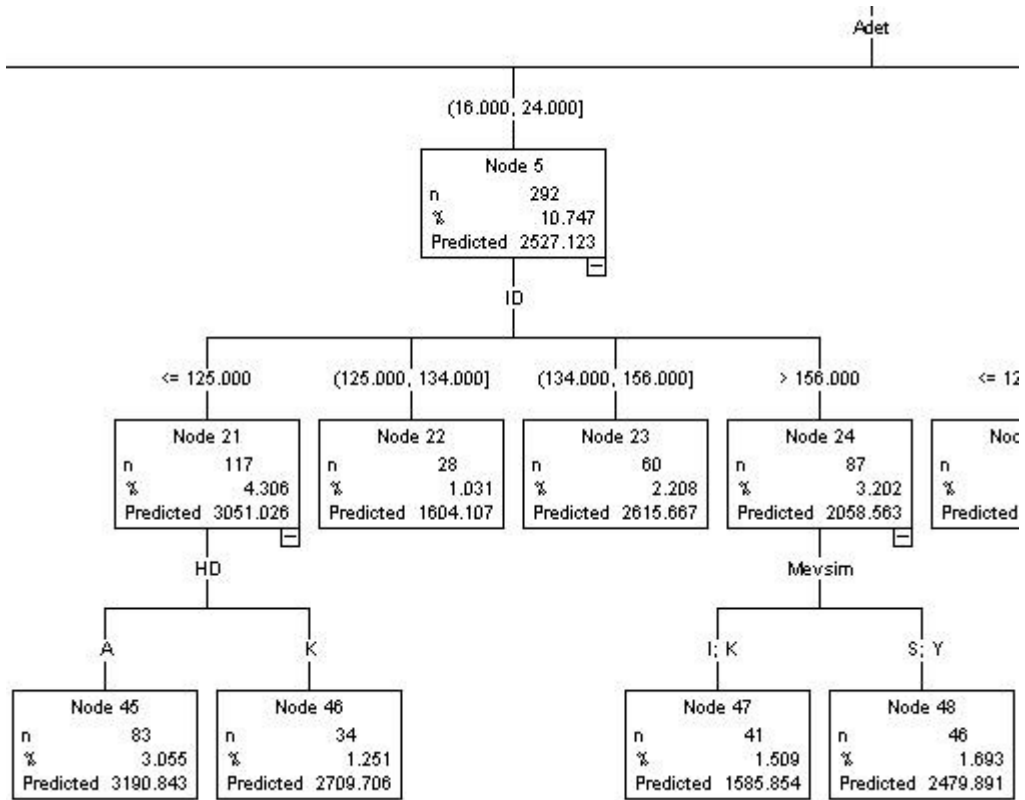
5.2.2. CHAID Uygulaması

Tahminleme de kullanılacak olan model oluşturulup, algoritma işletildiğinde üretilen kural ağacı Şekil 5.5'de görülmektedir. CART algoritmasından farklı olarak ana niteliğin farklı durumlarına göre diğer nitelikler homojen olarak değerlendirilerek sonuca gidilmiştir.



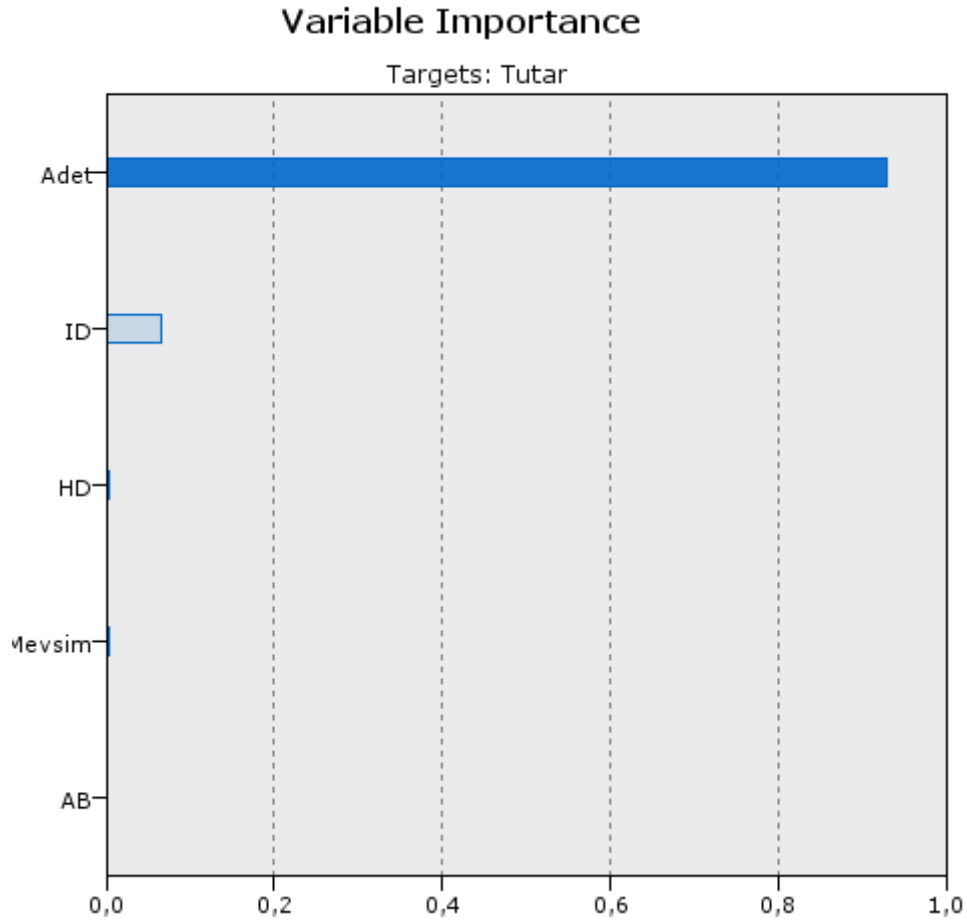
Şekil 5.4. CHAID algoritmasının ürettiği kural setinin bir kısmı

Oluşturulan kuralların yapısından da anlaşılacağı gibi, ağaç yapısının derin değil daha geniş olması beklenir. Şekil 5.6'da CHAID algoritmasının işletilmesi ile elde edilen ağaç yapısının bir bölü görünmektedir.



Şekil 5.5. CHAID algoritmasının ürettiği ağaç yapısının bir bölümü

Tahminleme de kullanılan veriler ve program tarafından belirlenen öncelik dereceleri Şekil 5.7'de görülmektedir.



Şekil 5.6. CHAID algoritması tahminlemede kullanılan değişkenlerin önem dereceleri

Tablo5.4’de oluşturulan modele test seti uygulandıktan sonra elde edilen analiz sonuçları görülmektedir. CART algoritmasına kıyasla daha başarısız sonuçların elde edildiği kolaylıkla söylenebilmektedir. CHAID algoritmasında sonuca etki eden parametrelere bakıldığında, CART algoritmasındaki homojen dağılımdan söz etmek mümkün olmamaktadır.

Tablo 5.4. CHAID algoritması analiz sonuçları

En Küçük Hata Değeri	-83
En Büyük Hata Değeri	92
Mutlak Hata Değeri	0,171
Standart Sapma	0,549
Lineer Korelasyon	0,927
Uygulama Adedi	561

BÖLÜM 6. SONUÇLAR

Günümüzde VM, büyük firmalar, bankalar, alışveriş merkezleri gibi müşteri ve kar odaklı çalışan sektörler için, popüler konulardan biridir. Bu tez kapsamında, daha çok bankacılıkla ilgili görünse bile nakit akışının ve nakit optimizasyonunun önemli olduğu her sektör için kullanılması gereken, bir tasarım gerçekleştirilmiş ve başarı ile uygulanmıştır. Bu tasarım gerçekleştirilirken kümeleme, ilişkilendirme ve sınıflandırma algoritmalarından yararlanılmıştır.

Geliştirilen sistemde:

- İki farklı algoritma kullanılmış olup en iyi sonucu CART algoritmasının verdiği görülmüştür.
- Başarılı bir alt yapı kurulmuş olup, modelde yapılacak ufak değişikliklerle istenilen periyotlar boyunca tahminler yapılabilmektedir.
- Geliştirilen sistem ihtiyaçlara göre genişletilebilir bir şekilde tasarlanmış ve bu sayede sistem esnekliği sağlanmıştır.
- Kurallar otomatik olarak üretildiği için her yeni veri eklenmesi sonrasında, manuel olarak kural tanımlamak gerekmemektedir.

Oluşturulan sistemde nakit tahminlemesi yapılmış olup sistem geliştirmeye açıktır. Tahminlemeye etki edecek daha bir çok etkenin daha belirlenerek eğitim setine katılması ile elde edilen sonuçlar daha net olacaktır. Bir nakit optimizasyonu sistemi geliştirilmek isteniyorsa özellikle gecelik faiz oranları, verilerin toplanma tarihlerine göre enflasyon oranları, para transfer maliyetleri (ulaşım, güvenlik, personel, vs..), konum etkisi, tarih etkisi, mevsimsel etkiler gibi sonuca kesin olarak etki edecek bir çok parametre ele alınmalıdır.

Sistemin ticari amaçlı kullanılabilmesi için istisnai durumların kontrol edilmesi, ve güvenlik açısından geliştirilmesi gerekmektedir.

KAYNAKLAR

[1]	ÖĞÜT, S., Veri Madenciliği Kavramı Ve Gelişim Süreci, sf. 5-6, İstanbul, 2006.
[2]	YARIMAĞAN, Ü., Veri tabanı sistemleri, Akademi & Türkiye Bilişim Vakfı, Ankara, 2000.
[3]	YAN, L., MILLER, J., MOZER, M., WOLNIEWICZ, R., Improving Prediction of Customer Behaviour in Nonstationary Environments
[4]	QUINLAN, J. R., Induction of decision trees, Machine Learning, vol. 1, pp. 81- 106, 1986.
[5]	LEE, S. K., An extended relational database model for uncertain and imprecise Information, In Proceedings Of The 1st VLDB conference, Vancouver, British Columbia, Canada, pp. 211-218, 1992.
[6]	AYDOĞAN, F., E-Ticarette Veri Madenciliği Yaklaşımlarıyla Müşteriye Hizmet Sunan Akıllı Modüllerin Tasarımı Ve Gerçekleştirimi, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, sf. 15-17, 2003.
[7]	SIMOUDIS, E., Reality check for data mining, In IEEE Expert: Intelligent Systems and Their Applications, pp. 26-33, 1996.
[8]	TOSUN, T., Veri Madenciliği Teknikleriyle Kredi Kartlarında Müşteri Kaybetme Analizi, sf. 12-17, 2006.
[9]	HAN, J. and KAMBER , M., Data Mining Concepts and Techniques, 2001.
[10]	http://www.karenstitusu.org/makaleler.php?aid=47 , Nisan 2009.
[11]	www.solicetec.com/tr/Nakit_Miktari_Tahminlemesi2.asp , Nisan 2009.
[12]	OĞUZLAR, A., TÜZÜNTÜRK, S., Borsada İşlem Gören Şirketlerin Finansal Göstergelerinin Analizi, sf. 8-9
[13]	SIMUTIS, R., DILIJONAS, D., BASTINA, L., Cash Demand Forcasting For ATM Using Neural Networks And Support Vector .Regression

	Algorithms, sf. 416-418, 2008
[14]	SIMUTIS, R., DILIJONAS, D., BASTINA, L., FRIMAN, J., DROBINOV, P. Optimization of cash management for ATM network, sf. 117–121, 2007

ÖZGEÇMİŞ

Öner ÇELİK, 04.08.1983 de İstanbul' da doğdu. İlk, orta ve lise eğitimini Beykoz'da tamamladı. 2001 yılında Beykoz Fevzi Çakmak Süper Lisesi, sayısal bölümden mezun oldu. 2002 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü 2006 yılında bitirdi. 2007 yılında Sakarya Üniversitesi, Bilgisayar ve Bilişim Mühendisliği Bölümünde yüksek lisansa başladı. 2007 – 2008 yılları arasında yazılım eğitimleri veren bir firmada Microsoft sertifikasyon eğitimleri verdi. Bu süre içerisinde .NET teknolojileri ve MSSQL veri tabanı ile ilgilendi. Şu anda Türkiye Halkbankası A.Ş.'de BİM Uzman Yardımcısı olarak görev yapmaktadır.