İSTANBUL TEKNİK ÜNİVERSİTESİ ★FEN EDEBİYAT FAKÜLTESİ
MATEMATİK MÜHENDİSLİĞİ PROGRAMI

ORACLE DATA MINER İLE İMKB HİSSELERİNİN YILLIK NET KARLARI ÜZERİNE BİR VERİ MADENCİLİĞİ UYGULAMASI

BİTİRME ÖDEVİ

Armağan ERSÖZ 090070057

Teslim Tarihi: 17.05.2012

Tez Danışmanı: Yar. Doç. Dr. Ahmet KIRIŞ

MAYIS 2012

ÖNSÖZ

Bu çalışmayı hazırlamamda bana yol gösteren, yardımını ve bilgisini hiçbir zaman esirgemeyen Sayın Hocam Yard. Doç. Dr. Ahmet KIRIŞ' a, bu zorlu süreçte manevi desteğiyle ve bilgisiyle her zaman yanımda olan arkadaşım Fırat CİVANER' e, hayatım boyunca bana bilgisi ve tecrübesiyle yol gösteren Kontrol Mühendisi Ağabeyim Aydoğan ERSÖZ' e ve bana sevgi, güven ve her türlü desteği veren anne ve babama en içten teşekkürlerimi sunarım.

Mayıs, 2012

Armağan ERSÖZ

İÇİNDEKİLER

ÖZET

Bu çalışmada İstanbul Menkul Kıymetler Borsası hisselerinin yıllık net karları üzerine bir veri madenciliği uygulaması yapılmıştır. Bu uygulama ile yıllara göre net karları bilinen söz konusu hisselerin bir sonraki yılının net karının belirlenmesi amaçlanmaktadır.

Bu amaçla İstanbul Menkul Kıymetler Borsası resmi sitesinden "Sermaye Artımları ve Temettü ödemeleri (1986 – 2011/09)" verileri alınmış, veriler düzenlenerek tablolar oluşturulmuştur. Hisseler net karlarına göre sınıflandırma modeline uygun hale getirilmiştir. Bu problemi çözmek için de sınıflandırma modelinin algoritmalarından Karar Destek Vektörleri (Support Vector Machine) ve Naive Bayes algoritmaları seçilmiştir.

Veri madenciliği uygulamasını gerçekleştirmek için Oracle veritabanının 11g sürümü ve üzerine Oracle Data Miner paket programı yüklenmiştir. Daha sonra oluşturulan tablolar veritabanına aktarılmış ve gerekli modeller oluşturulmuştur. Bu modeller yardımıyla da istenilen doğrulukta tahminler elde edilmeye çalışılmış ve sonuçlar yorumlanmıştır.

1. GİRİŞ

Gelişen teknoloji ve tüm Dünya ülkelerinde bilgisayarın yaygın kullanılması elektronik ortamda saklanan veri miktarında büyük bir artış meydana getirmiştir. Bilgi miktarının her 20 ayda bir iki katına çıkması veritabanı sayısında hızlı bir artışa neden olmaktadır. Birçok farklı bilim dallarından toplanan veriler, hava tahmini simülasyonu, sanayi faaliyet testleri, süpermarket alışverişi, banka kartları kullanımı, telefon aramaları gibi veriler, daha büyük veritabanlarında kayıt altına alınmaktadır. Veri tabanında veri birikiminin artarak devam etmesinin bir nedeni de yüksek kapasiteli işlem yapabilme gücünün ucuzlamasıdır. Bu verilerden elde edilecek bilgiler doğrultusunda iş dünyasında şirket stratejileri belirlenir.

Bu bitirme projesi kapsamında; veri madenciliği uygulaması ile, İMKB hisselerinin geçmiş yıllardaki net karlarının kar/zarar durumları üzerine model oluşturulmuş ve herhangi bir hissenin gelecek yılda net karının ne kadar olacağı tahmin edilmeye çalışılmıştır. Herhangi bir hissenin gelecek yıldaki tahmini net karını belirlerken hangi aşamalardan geçildiği ve veri madenciliği uygulamasına neden gereksinim duyulduğu bu tez dahilinde ifade edilmektedir.

2. VERİ MADENCİLİĞİ

Günümüzde kullanılan veri tabanı yönetim sistemleri eldeki verilerden sınırlı çıkarımlar yaparken geleneksel çevrimiçi işlem sistemleri (on-line transaction processing systems) de bilgiye hızlı, güvenli erişimi sağlamaktadır. Fakat ikisi de eldeki verilerden analizler yapıp anlamlı bilgiler elde etme imkanını sağlamakta yetersiz kalmışlardır. Verilerin yığınla artması ve anlamlı çıkarımlar elde etme ihtiyacı arttıkça uzmanlar Knowledge Discovery in Databases (KDD) adı altında çalışmalarına hız kazandırmışlardır. Bu çalışmalar sonucunda da veri madenciliği (Data Mining) kavramı doğmuştur. Veri madenciliğinin temel amacı, çok büyük veri tabanlarındaki ya da veri ambarlarındaki veriler arasında bulunan ilişkiler, örüntüler, değişiklikler, sapma ve eğilimler, belirli yapılar gibi bilgilerin matematiksel teoriler ve bilgisayar algoritmaları kombinasyonları ile ortaya çıkartılması ve bunların yorumlanarak değerli bilgilerin elde edilmesidir.[1]

2.1 Tanım

İlişkisel veri tabanı sistemleriyle ulaşılan veriler tek başına bir anlam ifade etmezken veri madenciliği teknolojisi bu verilerden anlamlı bilgi üretilmede öncü rol oynamaktadır. Aşağıda bazı veri madenciliği tanımlarına yer verilmektedir.

1. “Veri madenciliği; veritabanında bilgi keşfi (KDD), eldeki verilerden önceden bilinmeyen fakat potansiyel olarak yararlı olabilecek bilgileri çıkarmaktır. Bu kümeleme, veri özetlemesi, öğrenme sınıflama kuralları, değişikliklerin analizi ve sapmaların tespiti gibi birçok farklı teknik bakış açısını içine alır.” [2].
2. “Veri madenciliği, otomatik veya yarı otomatik çözüm araçları (tools) ile büyük ölçeklerdeki verinin anlamlı yapılar ve kurallar keşfetmek üzere araştırılması (exploration) ve analiz edilmesidir.” [3].

3. “Veri madenciliği çok büyük tabanları içindeki veriler arasındaki bağlantılar ve örüntüleri araştırarak, gizli kalmış yararlı olabilecek verilerden değerli bilginin çıkarılması sürecidir.” [4].
4. “Veri Madenciliği, büyük veri ambarlarından daha önceden bilinmeyen, doğru ve eyleme geçirilebilir bilgiyi ayrıştırma ve çok önemli kararların alınması aşamasında ayrıştırılan bu bilgiyi kullanma sürecidir.” [5].

Yukarıdaki tanımları toplayıp veri madenciliği kavramına ek bir tanım daha getirilebilir. Veri madenciliği; matematiksel yöntemler yardımıyla, biriken veri yığınları içerisinde bulunan dataların birbirleriyle ilişkisini ortaya çıkartmak için yapılan analiz ve kurulan modeller sonucunda elde edilecek bilgi keşfi sürecidir.

Veri madenciliğinin, disiplinler arası bir teknoloji olarak dört ana başlıktan oluştuğu kabul edilmektedir. Bunlar sınıflama, kategori etme, tahmin etme ve görüntülemedir. Bu dört temel dışında istatistik, makine bilgisi, veritabanları ve yüksek performanslı işlem gibi temelleri de içerir.

2.2 Tarihsel Gelişim

Veri madenciliğinin kavram olarak oluşması 1960’lı yıllara kadar dayanmaktadır. Bu dönemlerde veri taraması (data dredging), veri yakalanması (data fishing) gibi isimler verilmiş ve bilgisayar yardımıyla gerekli sorgulama (query) yapıldığında istenilen bilginin elde edilebileceği düşünülmüştür. Fakat 1990’lar geleneksel istatistiksel yöntemlerinin yerine algoritmik bilgisayar modülleri ile veri analizinin gerçekleştirilebileceğinin kabul edildiği yıllar olmuştur. Veri madenciliğinin tarihsel süreci Tablo1.1 de gösterilmiştir [6].

Tablo 1.1 : Veri madenciliğinin tarihsel gelişimi

Gelişim Adımları	Cevaplanan Karar Problemi	Kullanılabilen Teknolojiler	Ürün Sağlayıcıları	Karakteristikler
Veri Toplama (1960'lar)	"Benim toplam karım geçen 5 yılda ne kadardı?"	Bilgisayarlar, Teypler, Diskler	IBM,CDC	Geriye dönük , statik veri dağıtımı
Veri Erişimi (1980'ler)	"İngiltere'de geçen mart ayında birim satışları ne kadardı?"	İlişkisel Veritabanları, SQL, ODBC	Oracle,Sybase, Informix,IBM, Microsoft	Kayıt düzeyinde geriye dönük, dinamik veri dağıtımı
Veri Ambarlama ve Karar Destek Sistemleri (1990'lar)	"İngiltere'de geçen mart ayında birim satışları ne kadardı?"	OLAP, Çok Boyutlu Veritabanı Sistemleri, Veri ambarları	Pilot, Comshare, Arbor,Cognos, Microstrategy	Çoklu düzeylerde, geriye dönük dinamik veri dağıtımı
Veri Madenciliği (Bugün)	"Gelecek ay Boston'daki birim satışlar muhtemelen ne olabilir, niçin?	İleri düzeyde algoritmalar, çok işlemcili bilgisayarlar, büyük veritabanları	Pilot, Lockheed, IBM,SGI, SPSS,SAS, Microsoft vs.	Geleceğe dönük ,proaktif enformasyon dağıtımı

2.3 Kullanım Alanları

Tarihsel süreç, gelişen teknoloji ile veri madenciliğinin işlevliğini etkin bir şekilde sürdürdüğünü göstermektedir. Veriler çok hızlı bir şekilde toplanabilmekte, depolanabilmekte, işlenebilmekte ve bilgi olarak kurumların hizmetine sunulabilmektedir. Günümüzde bilgiye hızlı erişim, firmaların sürekli yeni stratejiler geliştirip etkili kararlar almalarını sağlayabilmektedir. Bu süreçte araştırmacılar, büyük hacimli ve dağınık veri setleri üzerinde firmalara gerekli bilgi keşfini daha hızlı gerçekleştirebilmeleri için veri madenciliği üzerine çalışmalar yapmışlardır. Tüm bu çalışmalar doğrultusunda veri madenciliği günümüzde yaygın bir kullanım alanı bulmuştur. Veri madenciliği, perakende ve pazarlama, bankacılık, sağlık hizmetleri, sigortacılık, tıp, ulaştırma, eğitim, ekonomi, güvenlik, elektronik ticaret alanında yaygın olarak kullanılmaktadır. [1]

2.4 Veri Madenciliği Modelleri

IBM tarafından veri işleme operasyonları için iki çeşit model tanımlanmıştır.

2.4.1 Doğrulama modeli

Doğrulama modeli kullanıcıdan bir hipotez alarak testler yapar ve bu hipotezin geçerliliğini araştırır.

2.4.2 Keşif modeli

Sistem bu modelde önemli bilgileri gizli veriden otomatik olarak elde eder. Veri başka hiçbir aracıya ihtiyaç duymadan yaygın olarak kullanılan modeller, genelleştirmeler ile ayıklanır.

2.5 Veri Madenciliği Uygulamaları İçin Temel Adımlar

Veri madenciliği uygulamalarında sırasıyla takip edilmesi gereken temel aşamalar aşağıda sistematik biçimde verilmiştir.

2.5.1 Uygulama alanının ortaya konulması

Bu ilk adımda veri madenciliğinin hangi alan ve hangi amaç için yapılacağı tespit edilir.

2.5.2 Hedef veri grubu seçimi

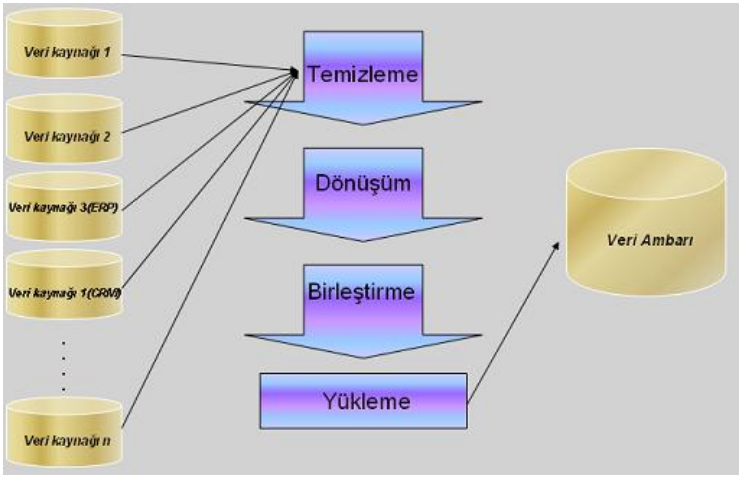
Belirlenen amaç doğrultusunda bazı kriterler belirlenir. Bu kriterler çerçevesinde aynı veya farklı veritabanlarından veriler toplanarak hedef (target) veri grubu elde edilir.

2.5.3 Model seçimi

Veri madenciliği probleminin seçimi datalar üzerinden belirlenir. (Sınıflandırma, Kümeleme, Birliktelik Kuralları, Şablonların ve İlişkilerin Yorumlanması v.b.)

2.5.4 Ön işleme

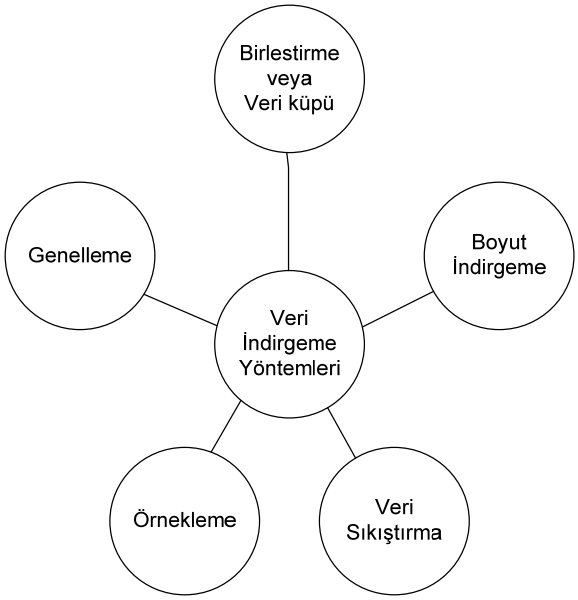
Bu aşamada seçilen veriler ayıklanarak silinir, eksik veri alanları üzerine stratejiler geliştirilir. Veriler tekrardan düzenlenip tutarlı bir hale getirilir. Bu aşamada yapılan işlem data temizleme ve data birleştirme olarak bilinen uyumlandırma işlemidir. Veri birleştirme (bütünleştirme), farklı veri tabanlarından ya da kaynaklarından elde edilen verilerin birlikte değerlendirmeye alınabilmesi için farklı türdeki verilerin tek türe dönüştürülebilmesi demektir. Bu süreç Şekil 2.1'de gösterilmektedir.



Şekil 2.1 : Ön işleme süreci

2.5.5 Veri indirgeme

Çözümleme işlemi veri madenciliği uygulamalarında uzun sürebilmektedir. İşlem yapılırken eksik ya da uygun olmayan verilerin oluşturduğu tutarsız verilerle karşılaşılabilir. Bu gibi durumlarda verinin söz konusu sorunlardan arındırılması gerekmektedir. Çözümlemeden elde edilecek sonuçta bir değişiklik olmuyorsa veri sayısı ya da değişkenlerin sayısında azaltmaya gidilir. Veri indirgeme çeşitli biçimlerde yapılabilir. Bu yöntemler Şekil 2.2'de gösterilmiştir [7].



Şekil 2.2 : Veri indirgeme yöntemleri

Veriyi indirirken bu verileri çok boyutlu veri küpleri biçimine dönüştürmek söz konusu olabilir. Böylece çözümler sadece belirlenen boyutlara göre yapılır. Veriler

arasında seçme işlemi yapılarak da veri tabanından veriler silinip boyut azaltılması yapılır.

2.5.6 Veri dönüştürme

Verileri direkt veri madenciliği çözümlerine katmak çoğu zaman uygun olmayabilir. Değişkenlerin ortalama ve varyansları birbirlerinden çok farklıysa büyük ortalama ve varyansa sahip değişkenlerin diğer değişkenler üzerindeki etkisi daha fazla olur. Bu nedenle bir dönüşüm yöntemi uygulanarak değişkenlerin normalleştirilmesi ya da standartlaşması uygun yoldur.

2.5.7 Algoritmanın belirlenmesi

Bu aşamada indirgenmiş veriye ve kullanılacak modele hangi algoritmanın uygulanacağına karar verilir. Mümkünse bu algoritmanın seçimine uygun veri madenciliği yazılımı seçilir değilse oluşturulan algoritmaya uygun programlar yazılır.

2.5.8 Yorumlama ve doğrulama

Uygulama sonucunda elde edilen veriler üzerine yorumlama yapılır, bu yorumların test verileri üzerinden doğrulanması hedeflenir. Doğruluğu onaylanan bu yorumlar gizli bilgiye ulaşıldığını göstermektedir. Elde edilen bu bilgiler çoğu kez grafiklerle desteklenir.

2.6 Temel Veri Madenciliği Problemleri ve Çözüm Yöntemleri

Veri madenciliği uygulaması gerektiren problemlerde, farklı veri madenciliği algoritmaları ile çözüme ulaşılmaktadır. Veri madenciliği görevleri iki başlık altında toplanmaktadır.

- Eldeki verinin genel özelliklerinin belirlenmesidir.
- Kestirimci/Tahmin edici veri madenciliği görevleri, ulaşılabilir veri üzerinde tahminler aracılığıyla çıkarımlar elde etmek olarak tanımlanmıştır.

Veri madenciliği algoritmaları aşağıda açıklanmaktadır.

2.6.1 Karakterize etme (Characterization)

Veri karakterizasyonu hedef sınıfındaki verilerin seçilmesi, bu verilerin genel özelliklerine göre karakteristik kuralların oluşturulması olayıdır.

Örnek: Perakende sektöründe faaliyet gösteren, uluslararası ABC şirketinin binlerce kurumsal müşterisi olsun. ABC şirketinin pazarlama biriminde, büyük kurumsal müşterilere yönelik kampanyalar için her yıl düzenli olarak bu şirketten 10 milyon TL ve üstü alım yapan kurumsal müşteriler hedeflenmektedir. Veritabanından hedef grup belirlenerek genelleme yapılır ve genel kurallar oluşturulur.

2.6.2 Ayrımlaştırma (Discrimination)

Belirlenen hedef sınıfa karşıt olan sınıf elemanlarının özellikleri arasında karşılaştırma yapılmasını sağlayan algoritmadır. Karakterize etme metodundan farkı mukayese yöntemini kullanmasıdır.

Örnek: ABC kurumsal müşterilerinden her yıl 10 milyon TL ve üstü alışveriş yapan fakat geri ödeme konusunda riskli olan müşteri grubunun belirlenmesi

2.6.3 Sınıflandırma (Classification)

Sınıflandırma, veri tabanlarındaki gizli örüntüleri ortaya çıkarabilmek için veri madenciliği uygulamalarında sıkça kullanılan bir yöntemdir. Verilerin sınıflandırılması için belirli bir süreç izlenir. Öncelikle var olan veritabanının bir kısmı eğitim amaçlı kullanılarak sınıflandırma kurallarının oluşturulması sağlanır. Bu kurallar kullanılarak veriler sınıflandırılır. Bu veriler sınıflandırıldıktan sonra eklenecek veriler bu sınıflardan karakteristik olarak uygun olan kısma atanır. Sınıflandırma problemleri için "Oracle Data Miner" (ODM)' ın uyumlu olduğu çözüm yöntemleri Naive Bayes (NB), Karar Destek Vektörleri (SVM), Karar Ağaçları ve Adaptive Bayes Network(ABN)' dir. [1]

Örnek: XYZ şirketi müşterilerinin alım durumlarını göz önünde bulundurarak, alım gücüne göre "Yüksek", "Orta", "Düşük" şeklinde sınıflandırır. Müşterilerinin risk durumlarını sınıflandırmak için de "Risksiz", "Riskli", "Çok Riskli" şeklinde etiketlerle sınıflandırılabilir.

2.6.4 Tahmin etme (Prediction)

Kayıt altında tutulan geçmiş verilerin analizi sonucu elde edilen bilgiler gelecekte karşılaşılacak aynı tarz bir durum için tahmin niteliği taşıyacaktır. Örneğin ABC şirketi geçen yılın satışlarını bölge bazlı sınıflandırmış ve bu sene için bir trend analizi yaparak her bölgede oluşacak talebi tahmin etmiştir. Bu tür problemler için ODM'nin kullandığı regresyon analizi yöntemi SVM'dir.

2.6.5 Birliktelik kuralları (Association rules)

Birliktelik kuralları gerek birbirini izleyen gerekse de eş zamanlı durumlarda araştırma yaparak, bu durumlar arasındaki ilişkilerin tanımlanmasında kullanılır. Bu modelin yaygın olarak Market Sepet Analizi uygulamalarında kullanıldığı bilinmektedir. Örneğin bir süpermarkette X ürününden alan müşterilerin büyük bir kısmı Y ürününden de almıştır. Birliktelik kuralı ile bu durum ortaya çıkarılarak, süpermarketin X ve Y ürününü aynı veya yakın raflara koyması sağlanır. ODM bu problem sınıfı için de Birliktelik Kuralları modelini kullanmaktadır.

2.6.6 Kümeleme (Clustering)

Yapı olarak sınıflandırmaya benzeyen kümeleme metodunda birbirine benzeyen veri grupları aynı tarafta toplanarak kümelenmesi sağlanır. Sınıflandırma metodunda sınıfların kuralları, sınırları ve çerçevesi belli ve datalar bu kriterlere göre sınıflara atanırken kümeleme metodunda sınıflar arası bir yapı mevcut olup, benzer özellikte olan verilerle yeni gruplar oluşturmak asıl hedeftir. Verilerin kendi aralarındaki benzerliklerinin göz önüne alınarak gruplandırılması yöntemin pek çok alanda uygulanabilmesini sağlamıştır. Örneğin, pazarlama araştırmalarında, desen tanımlama, resim işleme ve uzaysal harita verilerinin analizinde kullanılmaktadır. Tüm bu uygulama alanlarında kullanılması, ODM'nin desteklediği "K-means" ve "O-Cluster" kümeleme yöntemleri ile mümkün kılınmıştır.

2.6.7 Aykırı değer analizi (Outlier analysis)

İstisnalar veya sürpriz olarak tespit edilen aykırı veriler, bir sınıf veya kümelemeye tabii tutulamayan veri tipleridir. Aykırı değerler bazı uygulamalarda atılması gereken değerler olarak düşünülürken bazı durumlarda ise çok önemli bilgiler olarak değerlendirilebilmektedir.

Örneğin markette müşterilerin hep aynı ürünü iade etmesi bu metodun araştırma konusu içine girer. ODM; temizleme, eksik değer, aykırı değer analizi gibi birçok yöntemi veri hazırlama aşaması içine almakta ve desteklemektedir.

2.6.8 Zaman serileri (Time series)

Yapılan veri madenciliği uygulamalarında kullanılan veriler çoğunlukla statik değildir ve zamana bağlı olarak değişmektedir. Bu metot ile bir veya daha fazla niteliğin belirli bir zaman aralığında, eğilimindeki değişim ve sapma durumlarını inceler. Belirlenen zaman aralığında ölçülebilir ve tahmin edilen/beklenen değerleri karşılaştırmalı olarak inceler ve sapmaları tespit eder.

Örneğin ABC şirketinin Ocak-Haziran 2009 dönemi için önceki yılın satış miktarları göz önünde tutularak bir hedef ortaya konulmuştur. 2008 ve 2009 değerleri karşılaştırmalı olarak incelenerek sapma miktarı belirlenir. ODM her ne kadar çeşitli histogramlarla kullanıcıya görsel destek sağlasa da tam anlamıyla bu tür problemleri desteklememektedir.

2.6.9 Veri görüntüleme (Visualization)

Bu metot, çok boyutlu özelliğe sahip verilerin içerisindeki karmaşık bağlantıların/bağıntıların görsel olarak yorumlanabilme imkanını sağlar. Verilerin birbirleriyle olan ilişkilerini grafik araçları görsel ya da grafiksel olarak sunar. ODM zaman serilerinde olduğu gibi histogramlarla bu metodu desteklemektedir.

2.6.10 Yapay sinir ağları (Artificial neural networks)

Yapay sinir ağları insan beyninden esinlenilerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme yapılarıdır. Yapay sinir ağları öğrenme yoluyla yeni bilgiler türetebilme ve keşfedebilme gibi yetenekleri hiçbir yardım almadan otomatik olarak gerçekleştirebilmek için geliştirilmişlerdir. Yapay sinir ağlarının temel işlevleri arasında veri birleştirme, karakterize etme, sınıflandırma, kümeleme ve tahmin etme gibi veri madenciliğinde de kullanılan metotlar mevcuttur. Yüz ve plaka tanıma sistemleri gibi teknolojiler yapay sinir ağları kullanılarak geliştirilen teknolojilerdendir.

2.6.11 Genetik algoritmalar (Genetic algorithms)

Genetik algoritmalar doğada gözlemlenen evrimsel sürece benzeyen, genetik kombinasyon, mutasyon ve doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Genetik algoritmalar parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, yapay zeka uygulamaları, fonksiyonel ve kombinasyonel eniyileme problemleri, ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer eniyileme yöntemlerine kıyasla daha başarılı sonuçlar vermektedir.

2.6.12 Karar ağaçları (Decision trees)

Ağaç yapıları esas itibariyle kural çıkarma algoritmaları olup, veri kümelerinin sınıflanması için "if-then" tipinde kullanıcının rahatlıkla anlayabileceği kurallar inşa edilmesinde kullanılırlar. Karar ağaçlarında veri kümesini sınıflamak için "Classification and Regression Trees (CART)" ve "Chi Square Automatic Interaction Detection (CHAID)" şeklinde iki yöntem kullanılmaktadır.

2.6.13 Kural çıkarma (Rules induction)

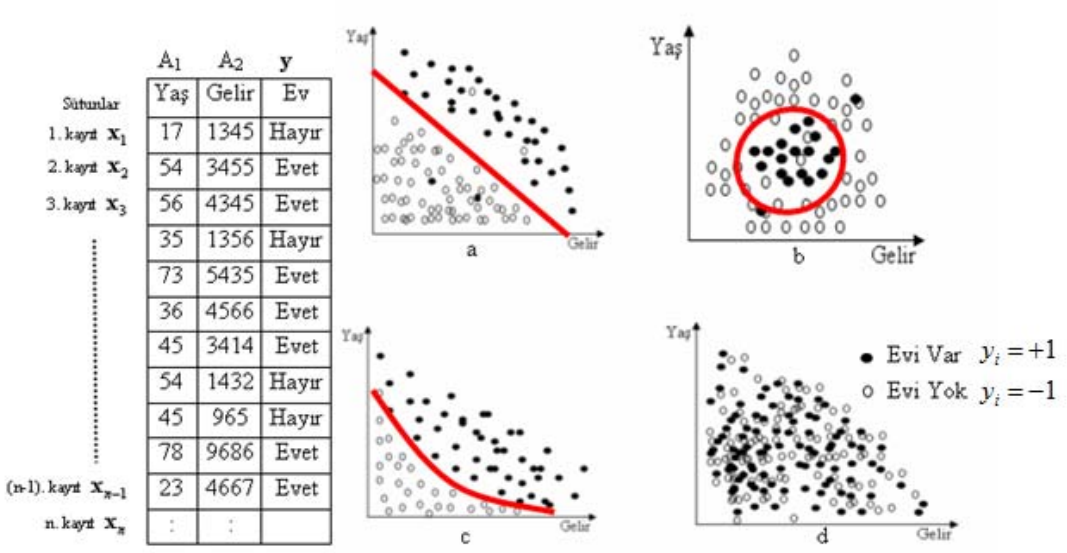
İstatistiksel öneme sahip yararlı "if-else" kurallarının ortaya çıkarılması problemlerini inceler.

3. TEMEL KAVRAMLAR VE MATEMATİKSEL ALTYAPI

Bu tez kapsamında, IMKB hisselerinin yıllık net kar verilerinden yararlanarak bir sonraki yılın net karını tahmin etme probleminin çözümü amacıyla Karar Destek Vektörleri ve Naive Bayes algoritmaları kullanılacağından sadece bu iki yöntem anlatılacaktır.

3.1. Karar Destek Vektörleri (Support Vector Machines)

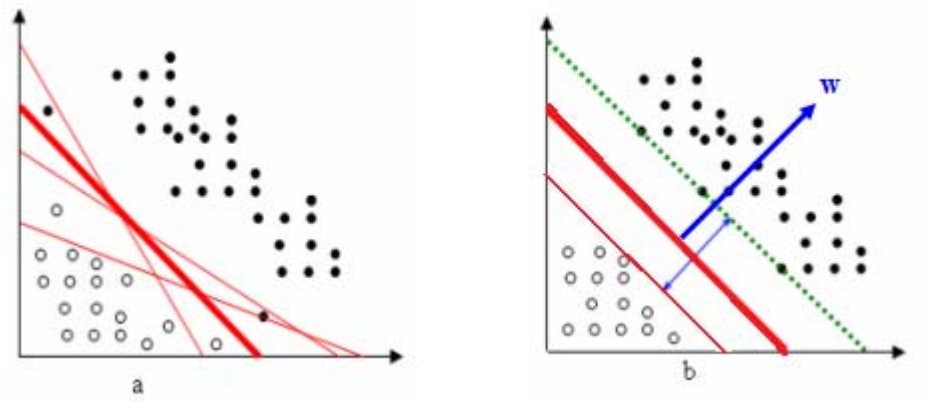
Karar Destek Vektörleri ODM tarafından hem sınıflandırma hem de Regresyon analizi amacıyla kullanılmaktadır. Burada olayı kolayca şekillendirebilmek için çizilecek şekillerde sadece iki tahmin edici kolon gösterilecek ancak yöntemin anlatılmasında m tane tahmin edici kolon olduğu kabul edilerek genelleştirilmiş ifadeler verilecektir. Aşağıdaki şekilde soldaki tablodaki veriler sağda iki boyutlu şekillerden biri ile gösterilebildiği varsayalım. Şekillerde belirtildiği gibi, tahmin edici kolonlar; Yaş ve Gelir koordinatlar olarak alınmıştır. Hedef kolon ev sahibi olup olmadığı ise, tabloda evet, hayır ve grafiklerde sırasıyla siyah ve beyaz daireler olarak gösterilmiştir. Eğer tablodaki veriler, “a” grafiği ile gösterilebilirse evi olanlar ve olmayanlar lineer bir doğru, “b” için bir elips, “c” için nonlineer bir eğri ile ayrılabilir. Verinin yapısına göre benzer başka bir çok ayrım da mümkün olabilir. Ancak bu veriler grafiklerle gösterildiğinde “d” deki gibi bir durumda ortaya çıkabilir. Bu durum tahmin edici kolonlar ile hedef kolon arasında bir ilişki olmadığı, başka bir deyişle Yaş ve Gelir verilerinin Ev sahibi olup olmadığını en azından bu tablodaki verilerle belirleyemediğini, dolayısıyla bu tabloya Veri Madenciliği uygulanamayacağı anlamına gelir.



Şekil 3.1 : SVM ile iki tahmin edici kolonlu bir sınıflandırma örneği

Yukarıda da bahsedildiği gibi amaç hedef kolonun sınıflarını ayıracak eğri denklemini bulmaktır. Konunun anlaşılması için öncelikle en basit olarak lineer bir eğri ile sınıfları ayırma durumu incelenecektir. Yöntemin başlangıcında belirtildiği gibi verilen şekiller iki boyutlu ama hesaplamalar genel olarak *m* boyut için yani *m* tahmin edici kolon için yapılacaktır. Ayrıca burada öncelikli olarak hedef kolonun sadece iki farklı değer alabildiği varsayılacak, daha sonra çoklu sınıflandırma anlatılacaktır.

Lineer Durum için ise, Şekil 3.1'deki tablodaki verilerin Şekil 3.2.a gibi lineer bir doğru (kesiksiz çizgi) ile ayrılabildiğini kabul edelim; amacımız beyaz ve siyah daireleri birbirinden ayıran doğru parçasının denklemini bulmak, ama şekilde de görüldüğü gibi bunları ayıran birçok doğru var, örneğin Şekil 3.2.a da ince çizgi ile gösterilen doğrularda bu ayrımı sağlamaktadır. Ancak bizim amacımız bu ayrımı en iyi şekilde sağlayacak ayrımı yapan doğrunun denklemini bulmak. Tabi ki burada doğru kavramı iki boyut olduğu için kullanılmaktadır, 3 boyutlu olduğunda doğru yerine düzlem, *m* boyutta ise doğru yerine *m* boyutlu hiper düzlemler ile bu ayrım sağlanmaktadır.



Şekil 3.2 : SVM iki boyut için sınıflandırma

Şekildeki beyaz ve siyah daireleri birbirinden ayırmak için, bu iki sınıf için aradaki maksimum uzaklık bulunmaya çalışılmalıdır. Bu amaçla Şekil 3.2.b'deki gibi gösterilen bir w vektörü tanımlansın. Benzer şekilde bileşenleri koordinatlar olan bir x vektörü ele alındığında Şekil 3.2.b'de kalın çizilen kesiksiz doğrunun denklemi

$$w.x + b = 0 \tag{3.1}$$

ile verilir.

Ayrıca Şekil 3.2.b'deki ince kesiksiz doğrunun denklemi

$$w.x + b = -1 \tag{3.2}$$

ve noktalı çizginin denklemi ise

$$w.x + b = 1 \tag{3.3}$$

ile verilebildiği kabul edilsin. Burada Şekil 3.2'de tabloda verilen her kaydın bir $\mathbf{x}_i$ noktasına karşı geldiğine ve bu kaydın hedef sütunun değerinin Evet veya Hayır olmasına göre noktanın beyaz veya siyah daire ile gösterileceğini hatırlatmak faydalı olabilir. Amaç, öyle bir w vektörü ve b skaleri bulmak ki, yukarıda verilen durumları gerçeklesin ve aynı zamanda bu iki sınıf yani beyaz ve siyah daireler arasındaki uzaklığı maksimum kılsın. Aslında bu ifade biraz daha yorumlanırsa burada bulunacak w vektörü ve b skalerinin, ince kesiksiz doğrunun ve noktalı çizginin denklemlerinin sağlanması demek, beyaz dairelerinin hepsinin ince kesiksiz doğrunun altında ve siyah dairelerin hepsinin de kesikli çizginin üstünde kalma şartı olduğu görülür. (3.2) ve (3.3) denklemleri arasında ki uzaklığı

$$uzaklık = m = \frac{1-(-1)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|} \tag{3.3}$$

ile verilir. Burada $\|\mathbf{w}\|$ ifadesi, $\mathbf{w}$ vektürünün normunu göstermekte olup,

$$\|\mathbf{w}\| = \sqrt{\sum_{i=1}^{m} w_i^2} \tag{3.4}$$

ile verilmektedir. Burada m tahmin edici kolon sayısı ve w_i ler $\mathbf{w}$ vektürünün *i.* bileşenini göstermektedir. m tane tahmin edici kolon olduğundan dolayı $\mathbf{w}$ ve $\mathbf{x}$ vektörlerinin boyutlarının da m olması gerektiği açıktır. Amaç, iki sınıf arasındaki uzaklığı maksimize edecek w vektörü ve b skalerini bulmak ve bu sırada beyaz dairelerin (3.2) ve siyah dairelerin (3.3) denklemleri ile verilen ince kesiksiz ve kesikli doğruların sırasıyla altında ve üstünde kalmalarını sağlamaktır. Bu amaç matematiksel olarak ifade edilmek istenirse şu şekilde olur.

Beyaz noktaların yani evi olmayanların $y_i = -1$ olarak kabul edildiği Şekil 3.1' den anlaşılmaktadır. (Bu kabul şekil ile uyumlu olması için yapılmıştır, hesaplamalarda Evi olanların veya olmayanların hedef kolonu -1 ile, diğer sınıfın ise +1 ile gösterilmesinde hiçbir sakınca yoktur ama şekille uyum açısından burada bu şekilde kabul edilmiştir). Beyaz dairelerin ince kesiksi doğrunun altında kalma şartı, her beyaz daire için yani hedef kolonu $y_i = -1$ olan her $\mathbf{x}_i$ noktası (kaydı, vektörü, satırı,...) için

$$w.x_i + b \leq -1 \tag{3.5}$$

denklemi sağlanmalı, siyah dairelerin kesikli çizginin altında kalma şartı, her siyah daire için yani hedef kolonu $y_i = 1$ olan her $\mathbf{x}_i$ noktası için

$$w.x + b \leq 1 \tag{3.6}$$

şartı sağlanmalı ve ayrıca

$$\underset{\mathbf{w}}{Maksimize}\ \frac{2}{\|\mathbf{w}\|} \tag{3.7}$$

olacak şekilde bir w vektörü ve b skaleri bulmaktır. Bundan sonraki işlemlerde kolaylık sağlamak için (3.7) yi maksimize etmek yerine bununla aynı anlama gelen $\|\mathbf{w}\|$ normunu minimize etmeye ve yine sadeleştirmelerin kolaylıkla

gerçekleşebilmesi için de $\|\mathbf{w}\|$ normu yerine $\frac{1}{2}\|\mathbf{w}\|^2$ ifadesini minimize etmeye çalışılacaktır. Açıktır ki bu ifadeyi minimize edecek w vektörü zaten (3.7)'yi maksimize eden w vektörü ile aynı vektördür.

Dolayısıyla problem,

$$\forall\ \mathbf{x}_i,\quad (i=1,...,n)\ \text{noktası için;} \qquad (\text{Burada } n \text{ toplam kayıt sayısıdır})$$

$$\begin{aligned} &y_i = 1 \quad \text{olan kayıtlar için} \qquad \mathbf{w}.\mathbf{x}_i + b \geq 1, \\ &y_i = -1 \quad \text{olan kayıtlar için} \qquad \mathbf{w}.\mathbf{x}_i + b \leq -1 \end{aligned} \tag{3.8}$$

şartlarını sağlayan ve

$$\underset{\mathbf{w}}{Minimize}\ \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.9}$$

ifadesini minimize eden bir w vektörü ve b skaleri bulmaktır. (3.18) şartlarını tek bir şart olarak ifade etmeye çalışılırsa her iki şartın birlikte

$$\forall\ \mathbf{x}_i \quad \text{noktası için} \quad (i=1,....,n)$$

$$y_i\left(\mathbf{w}.\mathbf{x}_i + b\right) \geq 1 \tag{3.10}$$

şeklinde yazılabileceği görülür. Dolayısıyla problem (3.10) koşulları altında (3.9) ifadesini minimize edecek bir w vektörü ve b skaleri bulma problemine dönüşür. Burada minimizasyon işleminin (3.10) gibi bazı kısıtlar altında gerçekleşmesi gerektiğinden doğrudan (3.9) ifadesinin değişkenlere göre türevleri sıfıa eşitlenerek sonuca gidilemez. Bazı kısıtlar altında gerçekleşmesi gereken extremum problemlerinde kısıtları da olayın içine katan Lagrange çarpanları yönteminin kullanılması gerekir. Lagrange formülasyonunda her kısıt "amaç fonksiyonuna" Lagrange çarpanı adı verilen bir ağırlık katsayısı ile eklenir. Lagrange formülasyonu orjinallinde "≤" eşitli kısıtlarla verildiğinden burada da buna uygun şekilde işlem yaparak yani (3.10) kısıtlarını eksi işareti ile çarparak Lagrange fonksiyonu aşağıdaki şekilde oluşturulacaktır. Ancak öncelikle (3.10) ifadesinin her kayıt için geçerli olması gerektiği, yani Şekil 3.1'de tabloda her kayıt veya şekildeki her beyaz ve siyah daireler için sağlanması gerektiğini söylenmişti. Dolayısıyla (3.10) ifadesinde toplam n tane koşul vardır. Şimdi Lagrange fonksiyonu

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w}.\mathbf{x}_i + b \right) - 1 \right) \tag{3.11}$$

şeklinde yazılabilir. (3.11)'de görüldüğü üzere ilk terim minimize etmek istenilen ifade, α_i 'ler Lagrange çarpanları ve parantez içindeki terim ise (3.20) kısıtlarıdır ve *n* tane kısıt olduğu için her kısıt farklı Lagrange çarpanı ile çarpılarak toplanmıştır. Dolayısıyla artık problem (3.11) Lagrange fonksiyonunu minimize edecek w vektörü ve *b* skalerini hesaplamaya indirgenir. Şimdi (3.11) ifadesinin w vektörü ve *b* skalerine göre türevlerini alarak, extremum olma şartları belirlenecektir. W vektörüne göre türevleri almak için önce herhangi bir bileşenine göre türev alınır, sonra bu genelleştirilir. Bunun için de öncelikle (3.11) ifadesi (3.4) yardımıyla daha açık olarak yazılmıştır.

$$\begin{aligned} L_P &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w}.\mathbf{x}_i + b \right) - 1 \right) \\ &= \frac{1}{2}\sum_{k=1}^{m} w_k^2 - \sum_{i=1}^{n} \alpha_i \, y_i \left(\sum_{k=1}^{m} w_k x_{ik} + b \right) + \sum_{i=1}^{n} \alpha_i \end{aligned} \tag{3.12}$$

ifadesinin w_j bileşenine göre türevi

$$\begin{aligned} &\frac{\partial L_P}{\partial w_j} = \frac{1}{2} 2 w_k \, \delta_{kj} - \sum_{i=1}^{n} \alpha_i \, y_i \left(\sum_{k=1}^{m} \delta_{kj} x_{ik} \right) = 0 \\ &\Rightarrow \; w_j - \sum_{i=1}^{n} \alpha_i \, y_i \, x_{ij} = 0, \; \Rightarrow \; w_j = \sum_{i=1}^{n} \alpha_i \, y_i \, x_{ij} \end{aligned} \tag{3.14}$$

dir ve buradan

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i \tag{3.15}$$

ve b skalerine göre türevinden

$$\frac{\partial L_P}{\partial b} = \sum_{i=1}^{n} \alpha_i \, y_i = 0 \tag{3.16}$$

elde edilir. (3.25) ve (3.26) ifadeleri Lagrange fonksiyonunda kullanıldığında ise

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, \mathbf{x}_i \mathbf{x}_j \tag{3.17}$$

ifadesi elde edilir. (3.17) denkleminde tek bilinmeyen olan α_i katsayıları bir quadratik programlama yazılımı ile çözülebilir. Bu katsayılar belirlendikten sonra ise (3.15) ifadesinden w vektörü belirlenir, b skaleri ise

$$b = \frac{\min_{\mathbf{x}_i}\left\{< w, x_i^{+1} >\right\} + \max_{\mathbf{x}_i}\left\{< w, x_i^{-1} >\right\}}{2} \tag{3.18}$$

formülü yardımıyla elde edilir. Burada "< >" ifadesi iki vektörün içi çarpımını göstermektedir. İfadenin yorumu x_i^{-1} de yani hedef kolonu "–" olan kayıtlar için **x** vekkörleri içinden $< w, x_i^{-1} >$ değeri en büyük olanla, x_i^{+1} de yani hedef kolonu "+" olan kayıtlar için **x** vekkörleri içinden $< w, x_i^{+1} >$ değeri en küçük olanın ortalama değeridir. (3.27) ve (3.28) ifadeleri yardımıyla w vektörü ve b skaleri belirlendikten sonra artık karar verici fonksiyon yani yukarıdaki kalın eğrinin denklemi

$$f(\mathbf{z}) = \mathbf{w}.\mathbf{z} + b \tag{3.19}$$

şeklinde yazılabilir. Her yeni kayıt için bu fonksiyonun verdiği değere göre

$$\begin{cases} f(\mathbf{z}) \geq 0 & \text{ise} \quad \mathbf{z} \text{ kaydının hedef kolonu } y_i = -1 \quad \text{yani "Evi Yok"} \\ f(\mathbf{z}) < 0 & \text{ise} \quad \mathbf{z} \text{ kaydının hedef kolonu } y_i = 1 \quad \text{yani "Evi Var"} \end{cases} \tag{3.20}$$

şeklinde bir sınıflandırma yapılır.

Yukarıda anlatılanlar sadece lineer bir eğri ile sınıfları ayırabilme ve hiçbir yanlış sınıflandırmaya izin vermeyen durum için geçerlidir. Gerçekte hatalı sınıflandırmaya izin verme yani kesikli çizgiler arasında genel yapıya uymayan ekstrem noktaların olabilmesi durumu da benzerdir. Ayrıca hem hedef kolon sadece ikili değil çoklu sınıflara ayrılmış olabilir hem de sadece lineer değil nonlineer eğrilerle sınıfları birbirinden ayırmak gerekebilir. Bu durumlarda yukarıdaki işlemlere benzer mantıkla işlemler silsilesi uygulanarak genel durum çıkarılabilmektedir. Ancak bu ayrıntılara bu proje kapsamında değinilmemiştir.

3.2 Naive Bayes Yöntemi

Bu bölümde veri kümesine uygulamak üzere seçilen sınıflandırma modelinin Karar Destek Vektörleri olmasına rağmen burada, sınıflandırma algoritmalarının en yaygını olan Geliştirilmiş Bayesian Ağlar' ın (Adaptive Bayes Network) açıklanmasına da

yer verilmiştir. Aşağıda bu modelin kullandığı Naive Bayes yönteminin matematiksel altyapısından bahsedilmiştir.

Sınıflandırma problemlerinin çözümünde kullanılan Naive Bayes yöntemi temel olarak olasılık teorisini kullanmaktadır. Bu bölümde önce yöntemin teorisi, sonrasında da yöntemle ilgili küçük örnekler verilmiştir.

3.2.1 Temel kavramlar

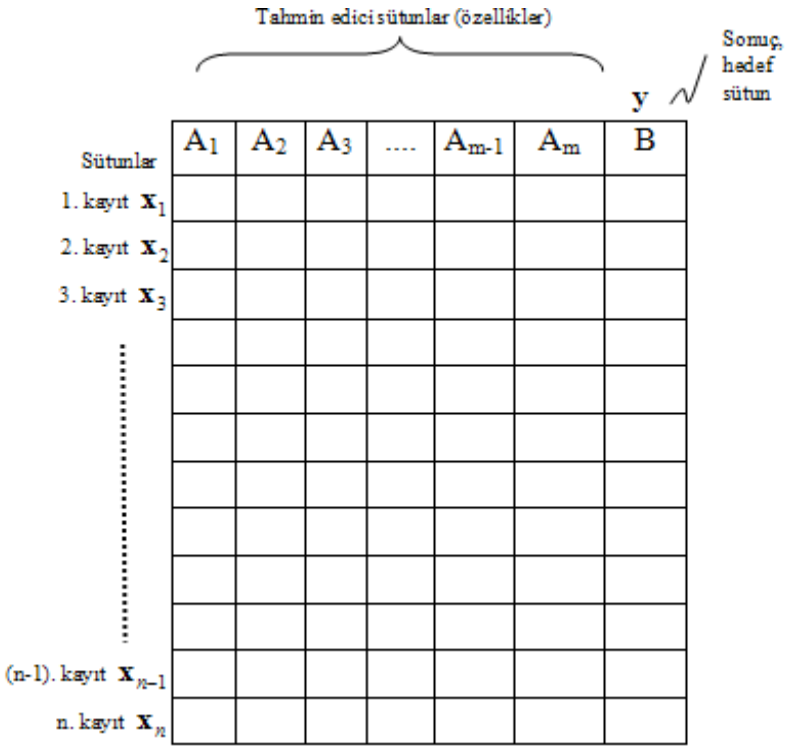
Olasılık: Bir olayın olabilirliğinin ölçüsüdür, [0-1] arasında değer alabilir, $P(A)$ ile gösterilir ve

$P(A) = 1$, A olayının mutlaka gerçekleşeceğini

$P(A) = 0$, A olayının gerçekleşmesinin mümkün olmadığını ifade eder.

Vektör: Burada kullanacağımız anlamıyla bir vektör $\mathbf{x} = \{x_1, x_2, x_3, ..., x_{m-1}, x_m\}$ şeklinde *m* elemanı ile belirlenen ve i. elemanı x_i ile verilen bir büyüklüktür.

Veri madenciliği uygulanacak olan veri kümesi aşağıda gösterilen tablonun formatındadır. Tabloda her satır (her kayıt) bir vektör ($\mathbf{x}_i$) olarak düşünülür, $\mathbf{x}_i$ vektörünün j. elemanı i. kaydın A_j sütunundaki değerine karşı gelir. Son sütun (*B*) yani $\mathbf{y}$ vektörü, veri madenciliği ile tahmin edilmek istenen hedef özelliktir. Dolayısıyla *n* kayıt ve (*m+1*) sütundan oluşan bir tabloda her biri *m* boyutlu *n* tane belirleyici $\mathbf{x}_i$ vektörü ve bir tane hedef sütun (*B*), yani $\mathbf{y}$ vektörü vardır.



Şekil 3.3 : Bir veri kaydı örneği

3.2.2 Teori

Naive Bayes yöntemi ile sınıflandırma koşullu olasılık hesabına dayanmaktadır. Şekil 4.1' de görüldüğü üzere tüm değerleri belirli geçmiş bir veri kümesinde, *B* yani sonuç sütunu, diğer $A_i, (i = 1,...,m)$ sütunlarına bağlı kabul edilerek, $P(B = b_j \mid A_i = a_{ik}, ...(i = 1,...,m))$, olasılıkları hesaplanır, burada $j = 1,...,s$ ve $k = 1,...,m_i$ dir. Bu ifade ile, her biri m_i tane farklı gruptan oluşan A_i sütunları a_{ik} değerlerini aldıklarında, bu A_i sütunlarına bağlı olarak, *B* sütununda bulunan s tane farklı grubun b_j değerlerinden her birini alma olasılıkları hesaplanmaktadır. Geçmiş veri kümesi yardımıyla hesaplanan bu olasılıklar, yeni gelecek verinin hangi gruba dahil edileceğinin, yani *B* sütununun tahmininde kullanılacaktır.

Konuyu anlaşılır kılmak için, tahmin edici sütun önce bir tane, A_1, sonra iki tane, A_1, A_2 alınarak, B sütununun bunlara bağlı olasılıkları hesaplanarak problem basitleştirilmiş daha sonra ise *m* sütun alınarak problem genelleştirilmiştir.

Öncelikle koşullu olasılık kavramının açıklanması gerekmektedir. A ve B iki olay olmak üzere, bu olayların olma olasılıkları $P(A)$ ve $P(B)$ ile verilir. Eğer *A* ve *B* olaylarının gerçekleşmesi birbirine bağlı değilse, bu iki olayın birlikte olma olasılığı

$$P(A,B) = P(A) \times P(B) \tag{3.21}$$

ile verilir. Örneğin A olayı, o gün havanın yağmurlu olması ve B olayı ise atılan bir madeni paranın yazı gelme olasılığı ise, bu iki olay birbirinden bağımsızdır ve bu iki olayın birlikte olma olasılıkları her bir olayın olma olasılıklarının çarpımına eşittir.

Eğer A ve B olayları birbirine bağlı ise, bu iki olayın birlikte olma olasılıkları; A' nın olma olasılığı ile A' dan sonra B' nin olma olasılığının çarpımı ile yani

$$P(A,B) = P(A)\,P(B \mid A) \tag{3.22}$$

veya B' nin olma olasılığı ile B' den sonra A' nın olma olasılığının çarpımı ile yani

$$P(A,B) = P(B)\,P(A \mid B) \tag{3.23}$$

ile verilir. Dolayısıyla buradan (3.22) ve (3.23) denklemleri birbirine eşitlenerek, A olayından sonra B olayının olma olasılığı

$$P(B \mid A) = \frac{P(B)P(A \mid B)}{P(A)} \tag{3.24}$$

ile verilir. Örneğin A olayı havanın yağmurlu olması, B olayı ise Ali*'* nin balığa çıkma olayı ise, B olayının A olayına bağlı olduğu açıktır ve A olayından sonra B olayının olma olasılığı yani hava yağmurlu iken Ali' nin balığa çıkma olayı (3.24) ifadesiyle hesaplanır.

Bir olayın olması ve olmaması olasılıkları toplamı $P(B) + P(B^{\perp}) = 1$ dir. Burada "$^{\perp}$" üst indisi B olayının değilini göstermektedir. Dolayısıyla Ali hava yağmurlu iken balığa çıktığı gibi, yağmur yağmazken de balığa çıkabilir, yani bir B olayına bağlı olarak A olayının olma olasılığı

$$P(A) = P(A,B) + P(A,B^{\perp}) = P(B)P(A \mid B) + P(B^{\perp})P(A \mid B^{\perp}) \tag{3.25}$$

şeklinde verilir. Bu ifade, (3.24)' te kullanılırsa,

$$P(B \mid A) = \frac{P(B)P(A \mid B)}{P(B)P(A \mid B) + P(B^{\perp})P(A \mid B^{\perp})} \tag{3.26}$$

elde edilir. Eğer A ve B olayları farklı değerler alabiliyorsa, örneğin Ali' nin balığa çıkması (b_1), işe gitmesi (b_2), spor yapması (b_3) gibi üç farklı B olayı varsa bu durumda $P(B = b_1) + P(B = b_2) + P(B = b_3) = 1$ dir. (3.25) ifadesine benzer bir şekilde bu kez A olayı r tane ayrık a_k ve B olayı s tane ayrık b_j değeri alıyorsa;

$$P\left(A=a_k\right)=\sum_{j=1}^{s}P\left((A=a_k),(B=b_j)\right)=\sum_{j=1}^{s}P\left(B=b_j\right)P\left((A=a_k)\,|\,(B=b_j)\right) \qquad (3.27)$$

elde edilir. (3.27) ifadesi (3.24)' te yerine yazıldığında ise,

$$P\left((B=b_j)\,|\,(A=a_k)\right)=\frac{P\left(B=b_j\right)P\left((A=a_k)\,|\,(B=b_j)\right)}{\sum_{k=1}^{s}P\left(B=b_k\right)P\left(A\,|\,(B=b_k)\right)} \qquad (3.28)$$

elde edilir. (3.28) ifadesinin *A* ve *B* olaylarının ikiden fazla değer alabildikleri durum için (3.26) ifadesinin genelleştirilmiş hali olduğu açıktır. Bu ifade Şekil.3.3' te verilen tabloda *B* sonuç sütununu tahmin edici tek bir A_1 sütunu olması halinde *B* sütununun alabileceği değerlerin olasılıklarının hesaplanmasında kullanılır. Ancak gerçek hayatta sadece biri tahmin edici, diğeri hedef sütun olmak üzere iki sütun olması değil, hedef sütunu tahmin edici birçok sütun bulunması beklenir.

Bu nedenle (3.28) ifadesinde *A* gibi sadece bir tahmin edici sütun yerine *m* tane A_i sütunu olduğunu ve bunların her birinin r_i tane bağımsız değer alabildiği yani örneğin A_1 sütunu $r_1=5$, A_2 sütunu $r_2=3$ farklı değer alabildiğini varsayalım. Bu durumda **Error! Reference source not found.** ifadesinde *A* yerine $A_1,A_2,...,A_m$ gibi m tane olay alınırsa;

$$P\left(B=b_j\,|\,A_1=a_{1j_1},A_2=a_{2j_2},...,\,A_m=a_{mj_m}\right)=$$
$$\frac{P\left(B=b_j\right)P\left(A_1=a_{1j_1},A_2=a_{2j_2},...,\,A_m=a_{mj_m}\,|\,B=b_j\right)}{\sum_{k=1}^{s}P\left(B=b_k\right)P\left(A_1=a_{1j_1},A_2=a_{2j_2},...,\,A_m=a_{mj_m}\,|\,B=b_k\right)} \qquad (3.29)$$

ifadesi elde edilir. Tahmin edici her sütunun yani her A_i olayının birbirinden bağımsız olduğu kabulü yapılırsa, sonuç olarak

$$P(B=b_k\,|\,A_1=a_{1j_1},A_2=a_{2j_2},...,\,A_m=a_{mj_m})=\frac{P(B=b_k)\times\prod_{i=1}^{m}P(A_i=a_{ij_i}\,|\,B=b_k)}{\sum_{\forall r|b_r\in B}\left(P(B=b_r)\times\prod_{i=1}^{m}P(A_i=a_{ij_i}\,|\,B=b_r)\right)} \qquad (3.30)$$

ifadesi elde edilir. Burada $j_i=1,...,m_i$ ve $k=1,...,s$ için bu olasılık değerleri hesaplanmalıdır, ayrıca $\forall r\,|b_r\in B$ terimi hedef sütunun alabileceği tüm farklı değerler üzerinde toplam alınacağını ifade etmektedir [8].

3.2.3 Örnekler

Örnek 1) *Tek boyut için:* Yapılan bir anket sonucunda 1000 deneğin gelir durumları "düşük", "orta", "iyi" ve "yüksek" olarak gruplanmış ve "Ev sahibi" olup olmadıkları ise ikinci bir sütunda Tablo 3.1a' da ki gibi belirtilmiş olsun.

Her ne kadar, ODM bu olasılık hesaplarını arka planda otomatik olarak işleyip kullanıcıya sadece sonucu bildirse de, burada amaç doğrultusunda arka planda neler döndüğü açıklanmıştır. Burada kısaltma amacıyla Gelir=G, Evet=E, Hayır=H şeklinde sembolize edilecektir. Tablo 3.1a verisinden elde edilen her farklı gruptaki kişi sayısı Tabl 3.1b ile gösterilmiştir.

Tablo 3.1.a : Gelir-Mülk ilişkisi

Gelir	**Ev**
Düşük	Evet
Orta	Evet
Yüksek	Hayır
İyi	Hayır
İyi	Evet
.	.
.	.

Tablo 3.1.b : Her gruptaki kişi sayısı

Gelir	**Ev=E**	**Ev=H**
Düşük	200	130
Orta	100	220
İyi	130	100
Yüksek	110	10

Tablo 3.1b yardımıyla sözü edilen olasılıklar (3.28) ifadesi kullanılarak;

$$P(Ev = E|Gelir = D) = \frac{P(Ev = E)P(G = D\backslash Ev = E)}{P(Ev = E)P(G = D\backslash Ev = E) + P(Ev = H)P(G = D\backslash Ev = H)}$$

$$= \frac{\frac{540}{1000} \times \frac{200}{540}}{\frac{540}{1000} \times \frac{200}{540} + \frac{460}{1000} \times \frac{130}{460}} = \frac{20}{33} = 0.6061$$

$$P(Ev = E|Gelir = D) = 0.6061$$

$$P(Ev = H|Gelir = D) = 1 - \frac{20}{33} = \frac{13}{33} = 0.3939$$

olarak hesaplanabilir. Burada bu sonuçlar çok daha kolay bir şekilde Tablo 3.1b' den de görülmektedir. Fakat hem hedef özelliğin ikiden fazla hem de kestirimci özellik sayısının birden fazla olduğu durumlarda tablodan okuma zorlaşacak ve yukarıdaki formülün uygulanması gerekecektir. Benzer şekilde diğer olasılıklar da hesaplanarak;

$$P(Ev = E|Gelir = O) = 0.3125$$

$$P(Ev = H|Gelir = O) = 1 - 0.3125 = 0.6875$$

$$P(Ev = E|Gelir = I) = \frac{13}{23} = 0.5652$$

$$P(Ev = H|Gelir = I) = 1 - \frac{13}{23} = 0.4348$$

$$P(Ev = E|Gelir = Y) = \frac{11}{12} = 0.9167$$

$$P(Ev = H|Gelir = Y) = 1 - \frac{11}{12} = 0.0833$$

yazılabilir.

Örnek 2) *İki Boyut için:* Yapılan bir anket sonucunda 100 deneğin gelir durumları "Düşük", "Orta" ve "Yüksek"; üniversite mezunu olup olmamaları ise "Evet" ve "Hayır" olarak belirlenmiş ve bu özelliklerdeki deneklerin haftalık televizyon izleme süreleri ise "İzlemiyor", haftada 5 saatten az izliyorsa "Az", 5 saatten fazla izliyorsa "Çok" olarak belirlenip 3. bir sütunda verilmiş olsun (Tablo 3.2). Kayıtların dağılımı Tablo 3.3' te verilmiştir

Tablo 3.2 : Model Verisi

Gelir	**Eğitim**	**Tv izleme**
Yüksek	Hayır	İzlemiyor
Orta	Evet	Az
Düşük	Evet	Az
Orta	Hayır	Çok
Düşük	Hayır	Çok
. .	. .	. .

Tablo 3.3 : Kayıt Dağılımları

Gelir	**Eğitim**	**Tv izleme**	**Kayıt Sayısı**
Düşük	Hayır	İzlemiyor	5
Düşük	Evet	İzlemiyor	1
Orta	Hayır	İzlemiyor	0
Orta	Evet	İzlemiyor	1
Yüksek	Hayır	İzlemiyor	1
Yüksek	Evet	İzlemiyor	4
Düşük	Hayır	Az	3
Düşük	Evet	Az	6
Orta	Hayır	Az	21
Orta	Evet	Az	10
Yüksek	Hayır	Az	6
Yüksek	Evet	Az	15
Düşük	Hayır	Çok	0
Düşük	Evet	Çok	2
Orta	Hayır	Çok	0
Orta	Evet	Çok	1
Yüksek	Hayır	Çok	4
Yüksek	Evet	Çok	20

Tablo 3.3 yardımıyla üniversite mezunlarının gelir verileri de kullanılarak televizyon izleme süreleri (3.30) ifadesiyle belirlenmeye çalışılmıştır. Burada iki belirleyici özellik olduğundan (3.30) ifadesi bu tabloya uygun formda yazılmalıdır. Burada T=Tv izleme, G=Gelir, E=Eğitim, D=Düşük, Y=Yüksek, O=Orta, İ=İzlemiyor, A=Az, Ç=Çok' u ifade etmektedir.

(3.30) ifadesi örneğe uygulandığında,

$$P(T = \text{İ} \mid G = D, E = H) = \frac{P(T = \text{İ})P(G = D \mid T = \text{İ})P(E = H \mid T = \text{İ})}{\begin{pmatrix} P(T = \text{İ})P(G = D \mid T = \text{İ})P(E = H \mid T = \text{İ}) \\ +P(T = A)P(G = D \mid T = A)P(E = H \mid T = A) \\ +P(T = \text{Ç})P(G = D \mid T = \text{Ç})P(E = H \mid T = \text{Ç}) \end{pmatrix}}$$

$$= \frac{\frac{12}{100} \times \frac{6}{12} \times \frac{6}{12}}{\left(\frac{12}{100} \times \frac{6}{12} \times \frac{6}{12}\right) + \left(\frac{61}{100} \times \frac{9}{61} \times \frac{30}{61}\right) + \left(\frac{27}{100} \times \frac{2}{27} \times \frac{4}{27}\right)} = 0.3885$$

sonucu elde edilir. Bu sonuca göre geliri düşük ve üniversite mezunu olmayan kişiler %38.85 olasılıkla televizyon izlemeyen gruptan olacaktır. Benzer şekilde diğer bazı olasılıklar da hesaplanırsa;

$P(T = \dot{I} \mid G = O, E = H) = 0.0253$ $P(T = \dot{I} \mid G = Y, E = Evet) = 0.0744$

$P(T = A \mid G = Y, E = Evet) = 0.3175$ $P(T = \dot{I} \mid G = D, E = Evet) = 0.3234$

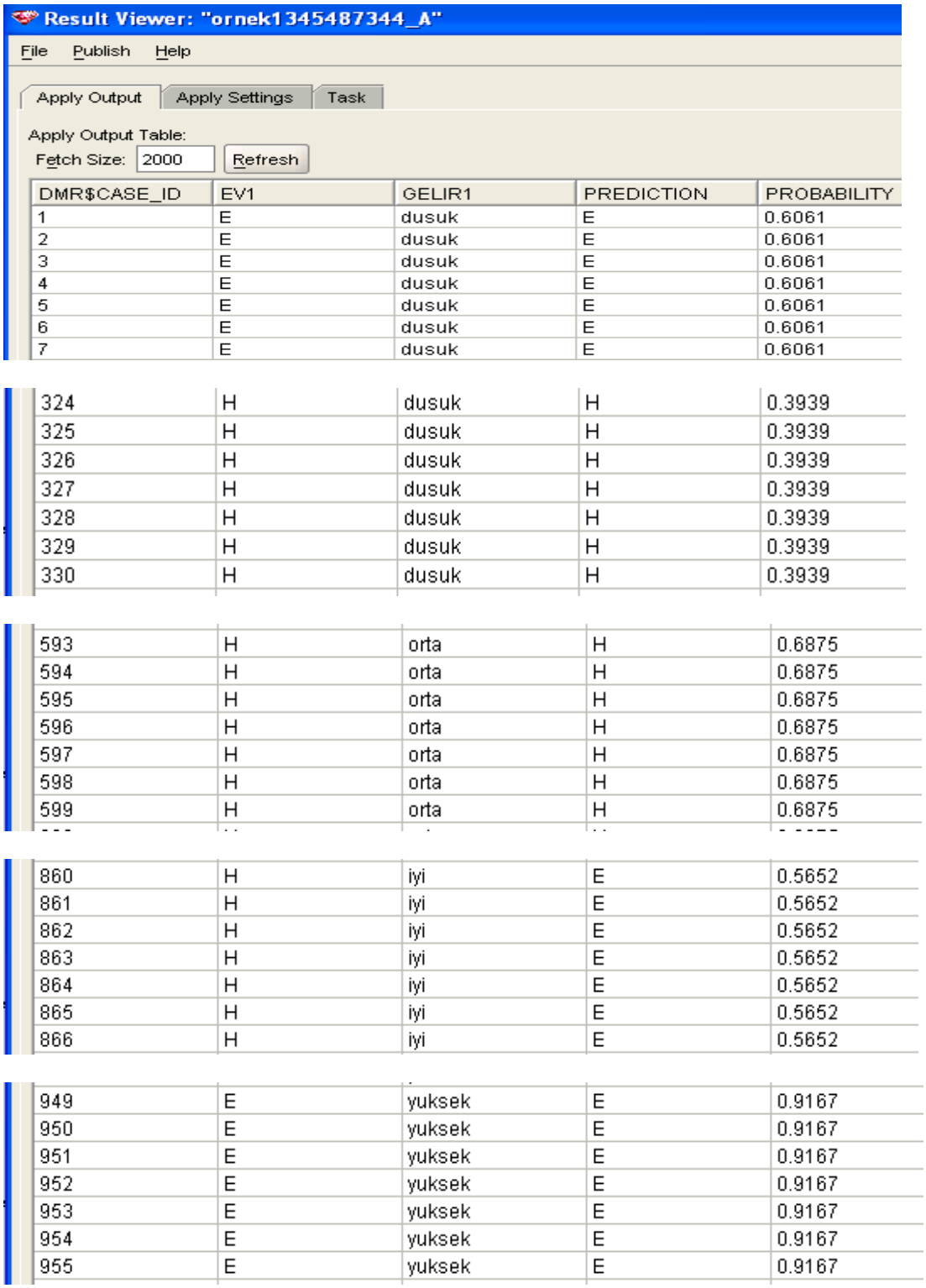
$P(T = Ç \mid G = O, E = H) = 0.0093$ $P(T = \dot{I} \mid G = Y, E = H) = 0.1526$

4. UYGULAMA VE SONUÇLAR

Bu bölümde Karar Destek Vektörlerinin ve Naive Bayes algoritmalarının Oracle Veri Madenciliği ile uygulanışı anlatılacaktır. Tez kapsamındaki asıl problemin, IMKB hisselerinin yıllara göre net karları yardımı ile bir sonraki yılın net karının tahmin edilmesi probleminin uygulanışı sürecinde Naive Bayes algoritmasına göre Karar Destek Vektörleri algoritması daha iyi sonuç vermiştir. Bu yüzden, bu tez kapsamında, modelin oluşturulma, uygulanma ve test süreçleri sadece Karar Destek Vektörleri için anlatılacaktır. Naive Bayes için ise bir önceki bölümde anlatılan örneklerin sonuçları ile Oracle Veri Madenciliği uygulamasının sonuçlarının karşılaştırılması amaçlanmıştır.

4.1. Naive Bayes Uygulaması (Tek boyutlu)

Bu bölümde, bir önceki örnekler bölümünde yer alan Örnek1 probleminin uygulanışı ve elde edilen sonuçların ODM' nin elde ettiği sonuçlarla karşılaştırılması yer almaktadır. Bunun için öncelikle Naive Bayes modeli oluşturulurken Discretize, Sample ve Split adımları atlanmalı, Cost Matrix seçeneği de kaldırılmalıdır. Ayrıca normalde ODM' de model oluşturulurken Apply aktivitesinde kullanılan tablo Build aktivitesinde kullanılandan farklı olmalıdır, çünkü Apply aktivitesindeki amaç yeni veri için tahmin kolonunun oluşturulmasıdır. Fakat burada sadece sonuçların doğruluğunun görülmesi amaçlandığından Apply aktivitesi de aynı tabloya uygulanmıştır. Aşağıda ODM' nin bu örneğe uygulanması sonucu elde edilen ekran çıktısı verilmiş, sonuçların aynı olduğu gözlenmiştir. [1]



DMR$CASE_ID	EV1	GELIR1	PREDICTION	PROBABILITY
1	E	dusuk	E	0.6061
2	E	dusuk	E	0.6061
3	E	dusuk	E	0.6061
4	E	dusuk	E	0.6061
5	E	dusuk	E	0.6061
6	E	dusuk	E	0.6061
7	E	dusuk	E	0.6061
324	H	dusuk	H	0.3939
325	H	dusuk	H	0.3939
326	H	dusuk	H	0.3939
327	H	dusuk	H	0.3939
328	H	dusuk	H	0.3939
329	H	dusuk	H	0.3939
330	H	dusuk	H	0.3939
593	H	orta	H	0.6875
594	H	orta	H	0.6875
595	H	orta	H	0.6875
596	H	orta	H	0.6875
597	H	orta	H	0.6875
598	H	orta	H	0.6875
599	H	orta	H	0.6875
860	H	iyi	E	0.5652
861	H	iyi	E	0.5652
862	H	iyi	E	0.5652
863	H	iyi	E	0.5652
864	H	iyi	E	0.5652
865	H	iyi	E	0.5652
866	H	iyi	E	0.5652
949	E	yuksek	E	0.9167
950	E	yuksek	E	0.9167
951	E	yuksek	E	0.9167
952	E	yuksek	E	0.9167
953	E	yuksek	E	0.9167
954	E	yuksek	E	0.9167
955	E	yuksek	E	0.9167

Şekil 4.1 : Örnek 1'in ekran çıktıları

ODM, Naive Bayes yöntemiyle sınıflandırmaya mümkün olan her durum için olasılıkları hesaplayarak bir model oluşturup, bu modeli yukarıdaki gibi aynı tablo üzerinde veya yeni kayıtların durumunu tespit için kullanmaktadır. Modelin doğruluğunun test edilmesi amacıyla formüllerle yapılacak işlemlerde aynı veri ve

hesaplanan olasılıklar kullanılarak yeni tahmin tablosu oluşturulabilir. Örneğin geliri düşük olanın ev sahibi olma olasılığı 0.6061 olarak hesaplandığı için tahmin evet ve sonucun güvenilirliği 0.6061'dir. Geliri orta olan kişinin ev sahibi olma olasılığı ise 0.3125 olduğu için modelin tahmini hayır ve sonucun güvenilirliği 0.6875 olacaktır. Bu şekilde işleme devam edilerek tüm tablo yeniden oluşturulur. [1]

Tablo 4.1 : Tablo 3.1a' nın yapılan hesaplamalarla elde edilen test sonuçları

Gelir	Ev	Tahmin	Güvenilirlik
Düşük	Evet	Evet	0.6061
Orta	Evet	Hayır	0.6875
Yüksek	Hayır	Evet	0.9167
İyi	Hayır	Evet	0.5652
İyi	Evet	Evet	0.5652
.	.	.	.

Modelin tüm güvenilirliği ise gerçek değerler ile tahmini değerlerin karşılaştırılması sonucu elde edilen aşağıdaki güvenilirlik matrisi ile verilebilir.

Tablo 4.2 : Güvenilirlik matrisi

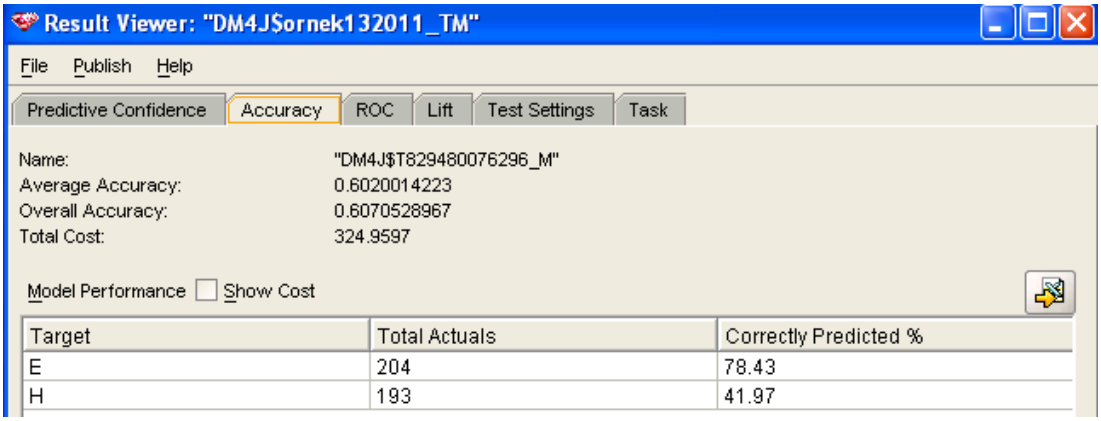
	E	H
E	440	100
H	240	220

Tablo 4.3'te görülen güvenilirlik matrisinde satırlar gerçek değerleri, sütunlar ise tahmin sonuçlarını göstermektedir. Örneğin gerçekte evi varken, modelin de evet yani "evi var" olarak tahmin ettiği kayıt sayısı 440 (doğru), gerçekte evi varken modelin hayır olarak tahmin ettiği kayıt sayısı (yanlış) 100 dür. Dolayısıyla matrisin köşegeni doğru kayıt sayısını, köşegen dışı ise yanlış kayıt sayısını göstermektedir. Buradan modelin doğruluğu

$$\frac{440 + 220}{440 + 100 + 240 + 220} = 0.66$$

olarak elde edilir. Modelin güvenilirliği ODM kullanılarak da hesaplanabilir. Ancak ODM ile model oluştururken verinin bir kısmını model, bir kısmını test için ayırma zorunluluğundan dolayı yukarıdaki veri %60 oranında model, %40 oranında test için ayrılarak ODM' den elde edilen güvenilirlik sonucu aşağıdaki ekran çıktısında

verilmiştir. Model, formüllerle hesaplanan duruma göre daha az veri kullandığı için güvenilirliğin biraz daha kötü çıkması doğaldır.



Şekil 4.2 : Örnek 1'in ODM ile güvenilirliği

4.2. Naive Bayes Uygulaması (2 boyutlu)

Bu bölümde de, bir önceki örnekler bölümünde yer alan Örnek2 probleminin uygulanışı ve elde edilen sonuçların ODM' nin elde ettiği sonuçlarla karşılaştırılması yer almaktadır.

76	E	cok	orta	izlemiyor	0.0292
51	E	az	orta	izlemiyor	0.0292
52	E	az	orta	izlemiyor	0.0292
43	E	az	orta	izlemiyor	0.0292
44	E	az	orta	izlemiyor	0.0292
50	E	az	orta	izlemiyor	0.0292
45	E	az	orta	izlemiyor	0.0292

100	E	cok	yuksek	az	0.3175
11	E	izlemiyor	yuksek	az	0.3175
12	E	izlemiyor	yuksek	az	0.3175
59	E	az	yuksek	az	0.3175
60	E	az	yuksek	az	0.3175
61	E	az	yuksek	az	0.3175
62	E	az	yuksek	az	0.3175

6	E	izlemiyor	dusuk	izlemiyor	0.3234
16	E	az	dusuk	izlemiyor	0.3234
17	E	az	dusuk	izlemiyor	0.3234
18	E	az	dusuk	izlemiyor	0.3234
75	E	cok	dusuk	izlemiyor	0.3234
20	E	az	dusuk	izlemiyor	0.3234
21	E	az	dusuk	izlemiyor	0.3234

1	H	izlemiyor	dusuk	izlemiyor	0.3885
2	H	izlemiyor	dusuk	izlemiyor	0.3885
3	H	izlemiyor	dusuk	izlemiyor	0.3885
15	H	az	dusuk	izlemiyor	0.3885
5	H	izlemiyor	dusuk	izlemiyor	0.3885
13	H	az	dusuk	izlemiyor	0.3885
14	H	az	dusuk	izlemiyor	0.3885

Şekil 4.3 : Örnek 2'nin ekran çıktıları

4.3. Karar Destek Vektörleri Uygulaması

Bu bölümde İstanbul Menkul Kıymetler Borsası'nın resmi sitesinden alınan verilerden yararlanarak IMKB hisselerinin net karları üzerine yapılan modellemenin adımları anlatılmıştır. Hisselerin 1986 ile 2011 yılları arasındaki net karlarını, hisse senedi olarak dağıtılan temettülerini, nakit olarak dağıtılan temettülerin TL ve döviz cinsinden değerlerini, efektif alışlarını içeren tablo, uygulamanın amacı doğrultusunda hisseler ve net karları içerecek bir şekilde düzenlenmiştir.

4.3.1 Veri Tablosunun Hazırlanması

Sınıflandırma probleminin daha iyi sonuç vermesi için öncelikle verilerimizin sınıflandırma problemine uygun hale getirilmesi gerekir. Burada veri temizleme işlemi sürecinde oluşturulan tüm tabloların ekran çıktılarına yer verilmemiştir fakat tüm tabloların oluşma süreci anlatılmıştır.

Öncelikle ilk tablomuzda hisselerin kodları, 1986 ve 2011 yılları arasındaki net karları, kodları, hisse senedi olarak dağıtılan temettüleri, nakit olarak dağıtılan temettüleri, efektif alışları gibi değerler bulunmaktadır.

Structure | Data

Fetch Size: 100 | Fetch Next | Refresh

COD	YEAR	NETKARTL	AMOUNTTL	HSODTRatio	NTBTTL	NTBTD	NDBHOBNTL	NetTL	Ex-DividendD...	CBExcRate
ABANA	2007	-523368								
ABANA	2006	-544762	0.00	0.00	0.00		0.0000	0.0000		
ABANA	2005	-243515	0.00	0.00	0.00		0.0000	0.0000		
ABANA	2004	-460530	0.00	0.00	0.00		0.0000	0.0000		
ACIBD	2010	11453399	0.00	0.00	0.00		0.0000	0.0000		
ACIBD	2009	17852738	0.00	0.00	0.00		0.0000	0.0000		
ACIBD	2008	-34617252	0.00	0.00	0.00		0.0000	0.0000		
ACIBD	2007	13419744	0.00	0.00	0.00		0.0000	0.0000		
ACIBD	2006	7244721	0.00	0.00	1919494.00	1449331.02	0.0356	0.0302	30.05.2007	1.324
ACIBD	2005	18647974	0.00	0.00	4417228.30	2833554.43	0.0818	0.0736	31.05.2006	1.559
ACIBD	2004	13388841	0.00	0.00	3821713.00	2800610.44	0.0708	0.0637	30.05.2005	1.365
ADANA	2010	102228498	0.00	0.00	58475763.52	36897882.08			31.05.2011	1.585
ADANA	2009	78726780	0.00	0.00	62858042.23	39922541.91			21.05.2010	1.574
ADANA	2008	107227141	0.00	0.00	79760661.00	52015560.85			29.05.2009	1.533
ADANA	2007	183973346	0.00	0.00	120491797.17	94481084.45			03.04.2008	1.275
ADANA	2006	133234965	0.00	0.00	100508229.49	73033156.15			03.04.2007	1.376
ADANA	2005	83705239	0.00	0.00	63106502.55	47370141.87			11.04.2006	1.332
ADANA	2004	36445523	0.00	0.00	25929925.64	18819803.77			02.05.2005	1.378
ADEL	2010	21735688	0.00	0.00	8820000.00	5565371.02	1.1200	0.9520	31.05.2011	1.585
ADEL	2009	16600678	0.00	0.00	6693750.00	4276063.63	0.8500	0.7225	31.05.2010	1.565
ADEL	2008	13716499	0.00	0.00	4567500.00	2978674.84	0.5800	0.4930	29.05.2009	1.533

Şekil 4.4 : Tüm verilerin bulunduğu tablo

Bu uygulamada hisselerin net karları üzerine veri madenciliği uygulaması yapmak amaçlandığından dolayı diğer bilgiler tablodan çıkarılmıştır. Yeni oluşturulan tabloda, hisselerin kodları ve o hisselerin yıllara göre net karları bulunmaktadır.

Structure | Data

Fetch Size: 100 | Fetch Next | Refresh

CODE	YEAR	NETPROFTL
ABANA	1991	1484
ABANA	1992	-8929
ABANA	1993	-21532
ABANA	1994	-22178
ABANA	1995	-7883
ABANA	1996	53303
ABANA	1997	180031
ABANA	1998	127501
ABANA	1999	813828
ABANA	2000	164308
ABANA	2001	719627
ABANA	2002	23821
ABANA	2003	-184309
ABANA	2004	-460530
ABANA	2005	-243515
ABANA	2006	-544762
ABANA	2007	-523368
ACIBD	1999	2022236
ACIBD	2000	3271064
ACIBD	2001	1036857
ACIBD	2002	4083881
ACIBD	2003	13567010
ACIBD	2004	13388841
ACIBD	2005	18647974
ACIBD	2006	7244721
ACIBD	2007	13419744
ACIBD	2008	-34617252
ACIBD	2009	17852738
ACIBD	2010	11453399

Şekil 4.5 : Gerekli verilerin bulunduğu tablo

Fakat tablonun bu hali uygulama için uygun değildir. Bir kolondaki değerin birden çok kez tekrar edilmesi(Şekil 4.5'te olduğu gibi) veritabanı yönetimi açısından uygun bir durum değildir. Bu yüzden tek bir hisseye ait bir satır, her bir yıla ait bir sütun olmasına dikkat edilmiştir. Verilerin bahsedilen duruma getirilmesi aşağıdaki SQL kodu ile gerçekleştirilmiştir.

“SELECT HİSSE_KODU,

MAX(CASE WHEN YIL= 2010 THEN NET_KAR END) AS NET2010,

MAX(CASE WHEN YIL= 2009 THEN NET_KAR END) AS NET2009,

MAX(CASE WHEN YIL= 2008 THEN NET_KAR END) AS NET2008,

MAX(CASE WHEN YIL= 2007 THEN NET_KAR END) AS NET2007,

MAX(CASE WHEN YIL= 2006 THEN NET_KAR END) AS NET2006,

MAX(CASE WHEN YIL= 2005 THEN NET_KAR END) AS NET2005,

MAX(CASE WHEN YIL= 2004 THEN NET_KAR END) AS NET2004

FROM UC_SUTUN560411502

GROUP BY HİSSE_KODU

ORDER BY HİSSE_KODU”

Bu işlemin sonucunda elde edilen tablo aşağıdaki gibidir.

Structure | Data

Fetch Size: 100 | Fetch Next | Refresh

CODE	Y1999	Y2000	Y2001	Y2002	Y2003	Y2004	Y2005	Y2006	Y2007	Y2008	Y2009
ACIBD	2022236	3271064	1036857	4083881	13567010	13388841	18647974	7244721	13419744	-34617252	17852738
ADANA	14481789	10787387	26090280	27828948	15363866	36445523	83705239	133234965	183973346	107227141	78726780
ADEL	1622572	1391305	1741836	3528510	2584940	3553017	5450406	8503937	7726990	13716499	16600678
AFYON	554466	470993	-436744	2488019	1018894	3921038	10304629	12717945	11522323	3658113	-709891
AKALT	495181	6426	8167740	6447340	-8089840	-11663476	-47078029	-16371397	3121426	18326874	6451711
AKBNK	318386799	343606000	-15015000	685448000	1324524000	1020528000	1438294000	1600192000	1994294000	1704553000	2725982000
AKCNS	11153837	2531981	6240619	17379493	35748074	63331801	113565095	146465575	185798034	104269708	75008370
AKENR	18440040	27928310	100211572	43928601	6571642	-16846949	-79091304	-59790065	-40280291	88950920	24249339
AKGRT	14468343	35831876	52580997	28246990	41230322	66480312	72247765	74184451	114496346	51970364	34965827
AKSA	17550429	18879296	62342993	79088709	24846533	37696889	-826562	61461963	4530504	72907209	50689317
AKSUE	859996	1504404	3239864	2393043	4145186	1628657	-550896	-1635028	379723	505252	3948202
ALARK	16969467	13161583	23258222	36205734	-23539725	-2446328	60238872	48311969	37644509	54819601	49853700
ALCAR	7956929	6847506	10233271	17615209	8208808	8123177	19728707	19894354	19083147	22836450	10437104
ALCTL	4488080	-5910940	-11638977	1623165	-14822527	-22294408	-624562	17586893	1443761	6574611	11214417
ALGYO	16838762	20384231	38185108	18477192	-8227120	-3787055	10183740	4916479	3106176	25809371	8387313
ALKIM	604825	3223370	9545724	9078728	9550151	8802967	7930893	11170067	18230111	22360616	20364150
ALNTF	33902031	11184000	-201099000	10125000	12483000	5134000	20765000	29654000	63320000	53016000	61544000
ALTIN	654215	372007	-22206003	702137	-1596154	642877	15557841	-1072323	11048926	-6707165	32558872
ALYAG	395406	624733	-18027025	-7313482	4770175	-4897629	-430642	-6158254	-4207963	-708250	-2959882
ANACM	1117488	3480251	2953473	32544360	71239556	61080557	53207158	37025898	70711886	10895800	15556709

Şekil 4.6 : Yılların satır olarak bulunduğu tablo

Sonraki adımda uç değerler tablodan çıkarılmıştır. Bunlar çok yüksek, çok düşük veya boş değerlerdir. Uygulamanın daha iyi sonuç verebilmesi için, hemen hemen tüm hisselerde net karları belirtilmeyen yıllar tablodan çıkarılmıştır. Bu işlemin sonucunda bazı hisseler toptan çıkarılmış, kalanlarının ise belirli yıllardaki net karları çıkarılmıştır. Bu adımın sonucunda elimizde söz konusu hisselerin 1999 ve 2010 yılları arasındaki net karları kalmıştır. Fakat 2005 yılında Türk Lirası’ndan 6 sıfır atıldığından dolayı, bu durumun uygulamamızda karışıklık yaratmaması için, tabloda

net karların USD karşılıkları bulunmaktadır. USD karşılıkları da yer yılın net karının 31 Aralıktaki USD kuruna bölümü ile elde edilmiştir.

Son adımda her yılın bir önceki yıla oranı bulunmuştur. Bu oran bulunulan yıl ile bir önceki yılın farkının bir önceki yıla bölümü ile bulunmuştur. Elde edilen oranlara bir gruplama yapılmıştır.

Tablo 4.3 : Kar ve zarar oranlarına göre gruplama

Kredi Tutarı	Verilen Değer
%100-%50 zarar	Z2
%50-%25 zarar	Z1
%25-%0 zarar	Z0
%0-%10 kar	K0
%10-%25 kar	K1
%25-%50 kar	K2
%50-%100 kar	K3
%100 ve üzeri kar	K4

Kullanıma hazır tablonun son hali şekildeki gibidir.

Structure | Data

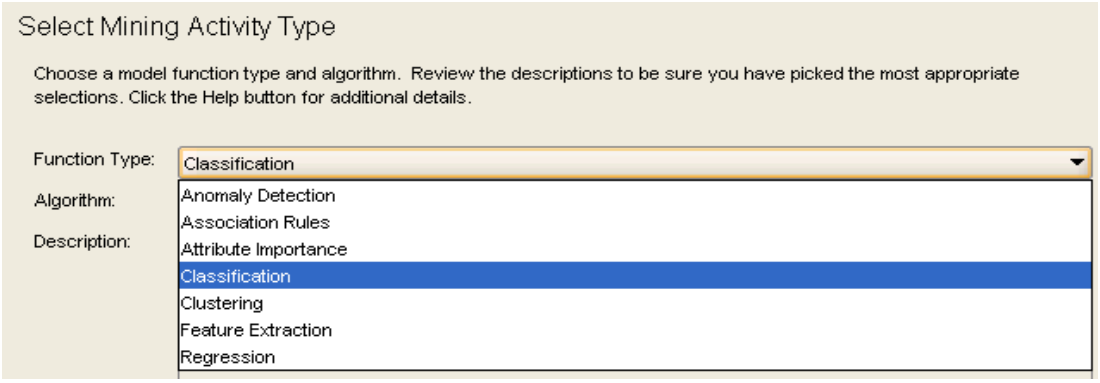
Fetch Size: 100 | Fetch Next | Refresh

CODE	Y1	Y2O	Y3O	Y4O	Y5O	Y6O	Y7O	Y8O	Y9O	Y10O	Y11O
ACIBD	3744202	K0	Z0	K0	K0	K0	K0	Z0	K0	Z0	Z0
ADANA	26813262	Z0	K0	Z0	Z0	K0	K0	K0	K0	Z0	Z0
ADEL	3004218	Z0	Z0	K0	Z0	K0	K0	K0	K0	K0	K0
AFYON	1026603	Z0	Z0	Z0	Z0	K0	K0	K0	K0	Z0	Z0
AKALT	916835	Z0	K5	Z0	Z0	K0	K0	Z0	Z0	K0	Z0
AKBNK	589498200	Z0	Z0	Z2	K0	Z0	K0	K0	K0	Z0	K0
AKCNS	20651506	Z0	K0	K0	K0	K0	K0	K0	K0	Z0	Z0
AKENR	34142026	K0	K0	Z0	Z0	Z0	K0	Z0	Z0	Z0	Z0
AKGRT	26788366	K0	Z0	Z0	K0	K0	K0	Z0	K0	Z0	Z0
AKSA	32494897	Z0	K0	K0	Z0	K0	Z0	Z3	Z0	K2	Z0
AKSUE	1592296	K0	K0	Z0	K0	Z0	Z0	K0	Z0	K0	K1
ALARK	31419237	Z0	Z0	K0	Z0	Z0	Z2	Z0	Z0	K0	Z0
ALCAR	14732380	Z0	Z0	K0	Z0	K0	K0	Z0	K0	Z0	Z0
ALCTL	8309751	Z0	Z0	Z0	Z1	K0	Z0	Z2	Z0	K0	K0
ALGYO	31177235	Z0	Z0	Z0	Z0	Z0	Z0	Z0	Z0	K1	Z0
ALKIM	1119843	K0	K0	Z0	K0	Z0	Z0	K0	K0	Z0	Z0
ALNTF	62770147	Z0	Z0	Z0	K0	Z0	K0	K0	K0	Z0	K0
ALTIN	1211289	Z0	Z2	Z0	Z0	Z0	K2	Z0	Z1	Z0	Z0
ALYAG	732100	K0	Z1	Z0	Z0	Z0	Z0	K2	Z0	Z0	K0
ANACM	2069047	K0	Z0	K1	K0	Z0	Z0	Z0	K0	Z0	K0

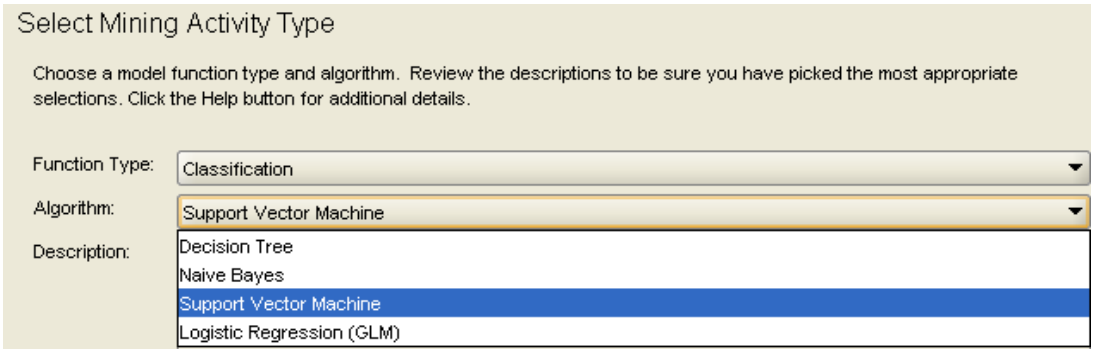
Şekil 4.7 : Kullanıma hazır tablonun son hali

4.3.2 Model Oluşturma

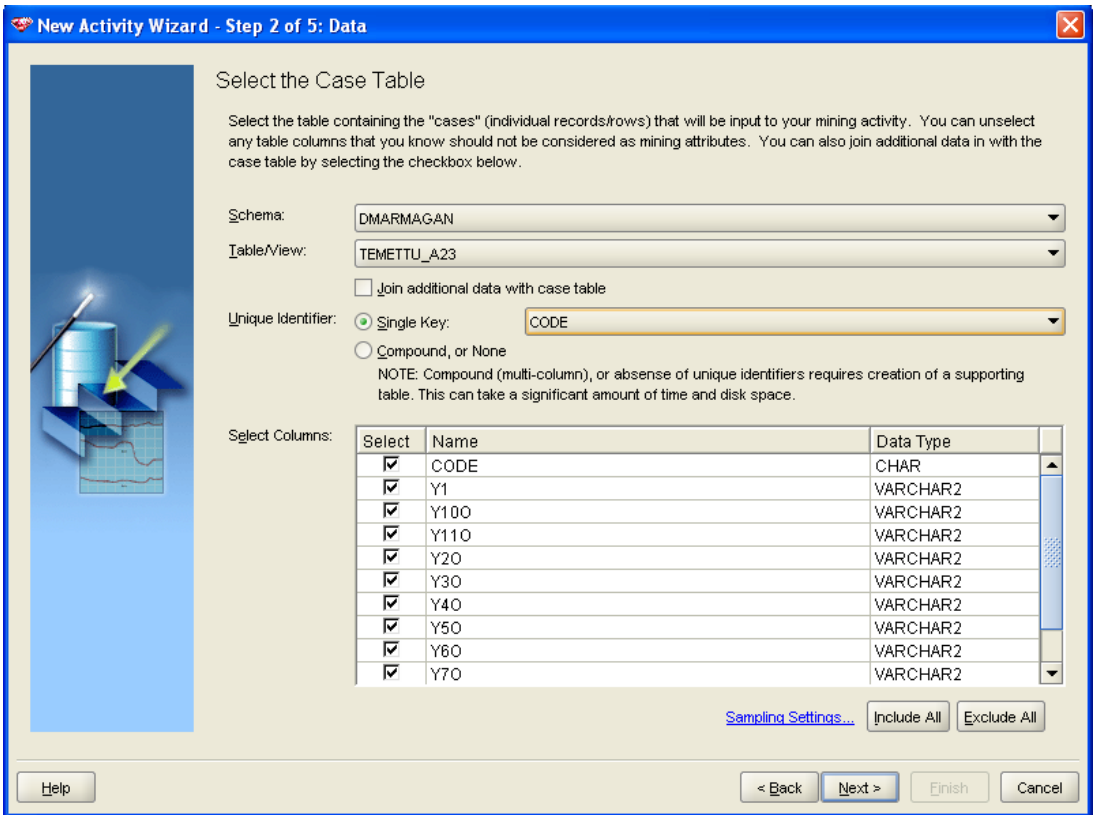
Veri tablosu üzerinde yapılan veri temizleme ve gruplama işlemleri sonrasında problemin yapısının sınıflandırma modeline uygun olduğu belirlenmiş ve bu bilgiler fonksiyon tipi olarak "Classification", çözüm algoritması olarak da "Support Vector Machine" seçilerek "TEMETTU_A23" için model oluşturma aşamaları ekran çıktıları ile birlikte aşağıda verilmiştir.



Şekil 4.8 : Fonksiyon tipi seçimi

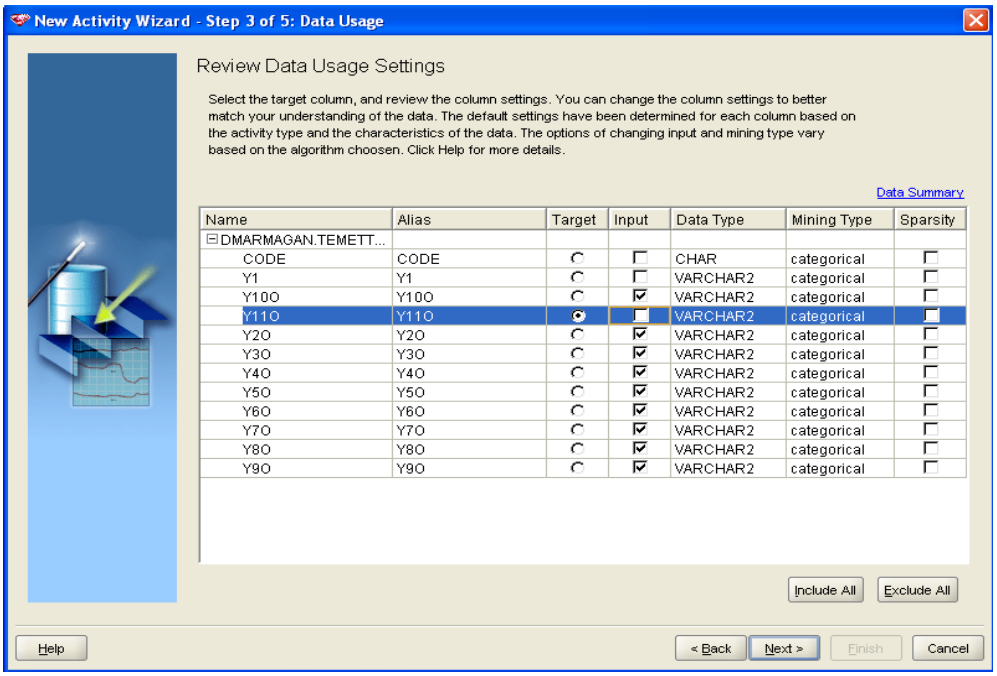


Şekil 4.9 : Algoritma seçimi



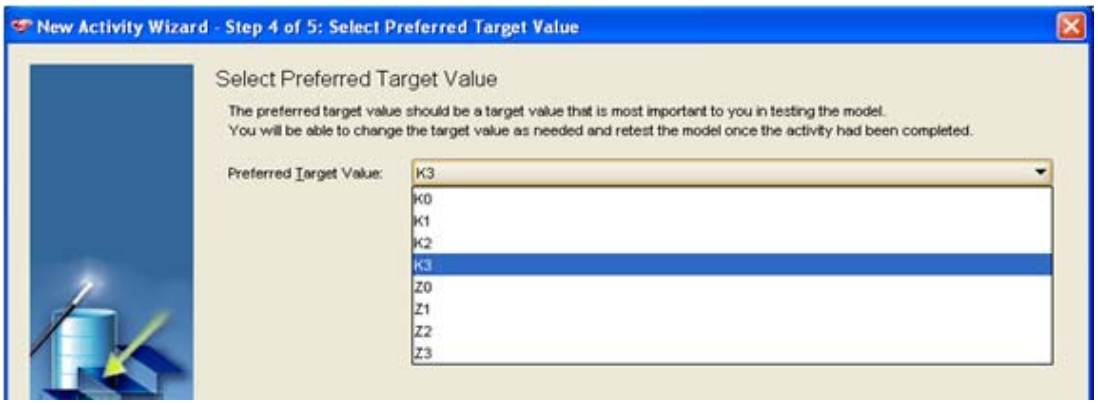
Şekil 4.10 : Model oluşturma, 1. aşama

Model gruplanmış tablo üzerinde oluşturulacağından "TEMETTU_A23" isimli tablo seçilmiştir. "Unique Identifier" kısmı zorunlu olup, tabloda benzersiz değerlere sahip olan sütun "Single Key" olarak seçilir. Bu tabloda benzersiz değer CODE sütunudur. Bu modeldeki CODE haricindeki tüm sütunlar tahmin edilecek sütunu etkileyeceğinden işleme hepsi dahil edilmiştir. CODE sütunu da "Single Key" olduğundan işleme dahil edilmiştir.



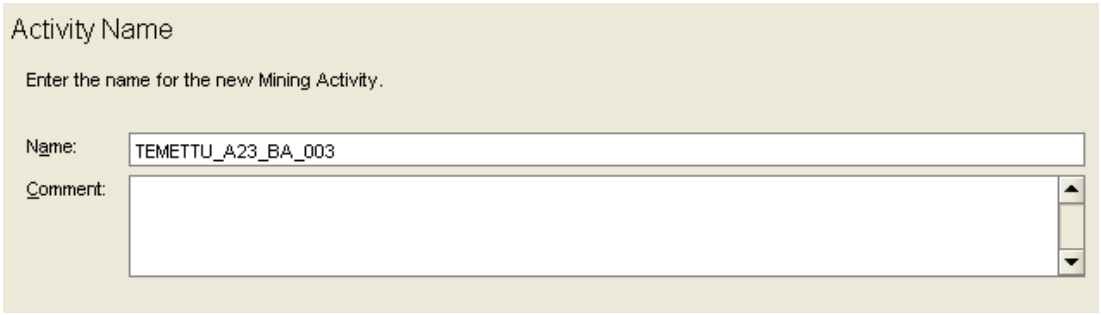
Şekil 4.11 : Model oluşturma, 2. aşama

Hedef sütun olarak Y11O(1999) seçilmiş ve Y11O sütununun tahmini için bir etkisi bulunmayan "Single Key" olan CODE ve gruplandırma yapılmamış sütun Y1 girdiler arasından çıkarılmıştır.

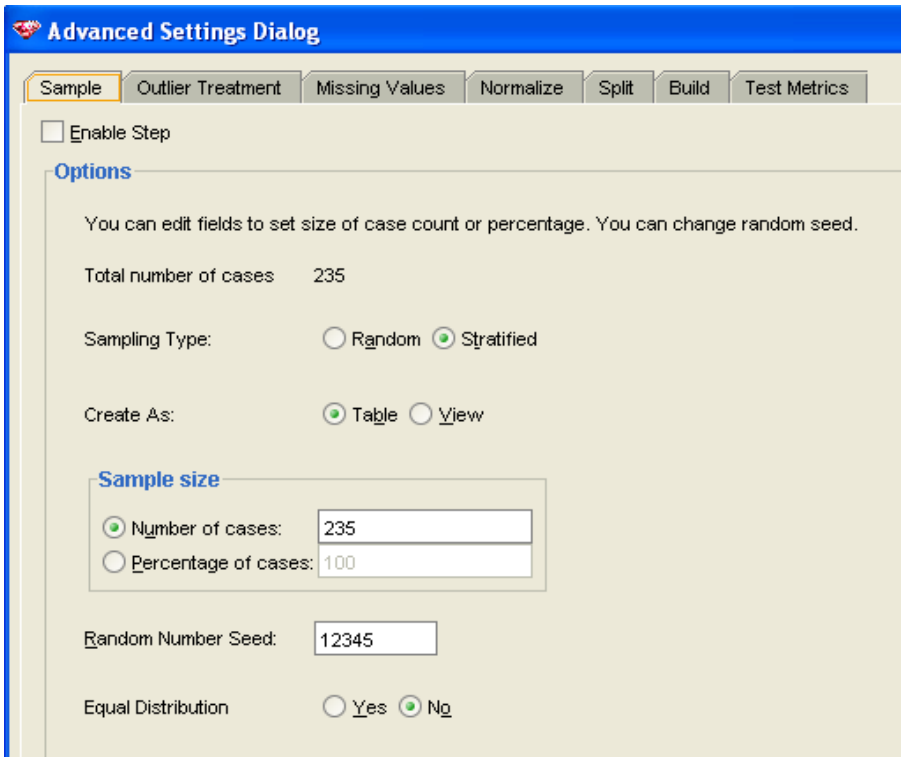


Şekil 4.12 : Tercih edilen hedef değer seçimi

Tercih edilen hedef değeri, görmek istenen sonuçlara göre şekillendirilir. Bu uygulamada tüm hisselerin 2010 yılı için hangi oranda kar veya zarar ettiğini görmek için yapılmış olduğundan "Target Value" seçeneği herhangi biri olarak seçilebilir. Yine de en çok oranda kar eden hisseleri görmek için "Target Value" K3 değeri olarak seçilmiştir. Bir sonraki adımda da modele ismi verilmiştir.

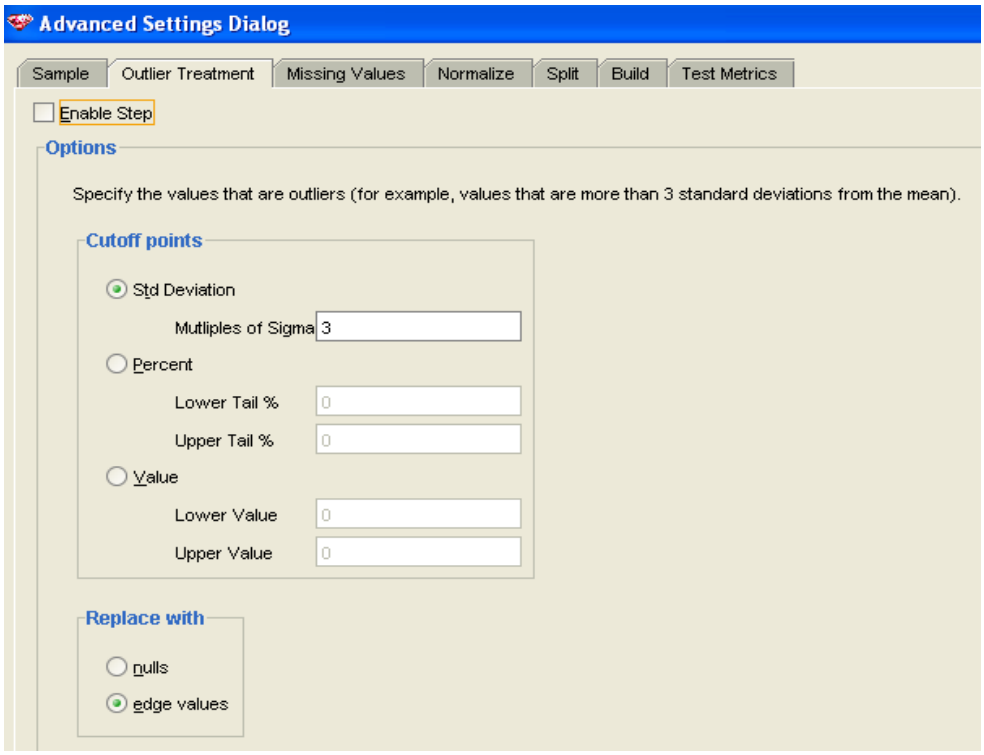


Şekil 4.13 : Modelin ismini belirleme



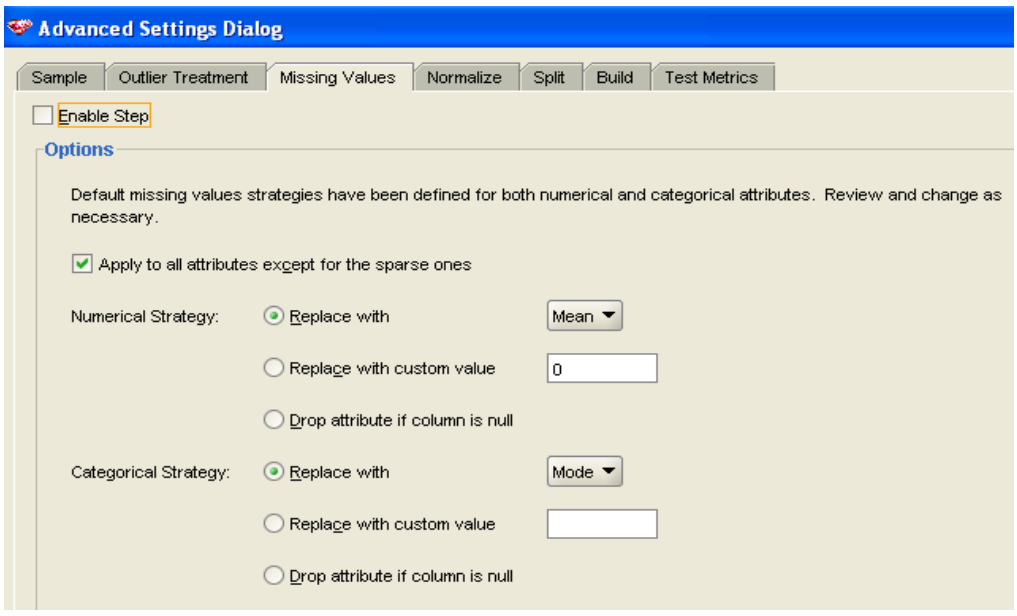
Şekil 4.14 : Gelişmiş ayarlar, "Sample" sekmesi

Son adımda yer alan "Advanced Settings" seçilerek model oluşturma aşamasında yürütülecek adımlar üzerinde gerekli ayarlar yapılmıştır. Tabloda çok fazla sayıda kayıt bulunduğu durumlarda kullanılan ve bu kayıtların içinden bir örneklem kümesi seçmeye yarayan "Sample" özelliği programda varsayılan olarak zaten seçili değildir ve aynı şekilde bırakılır.



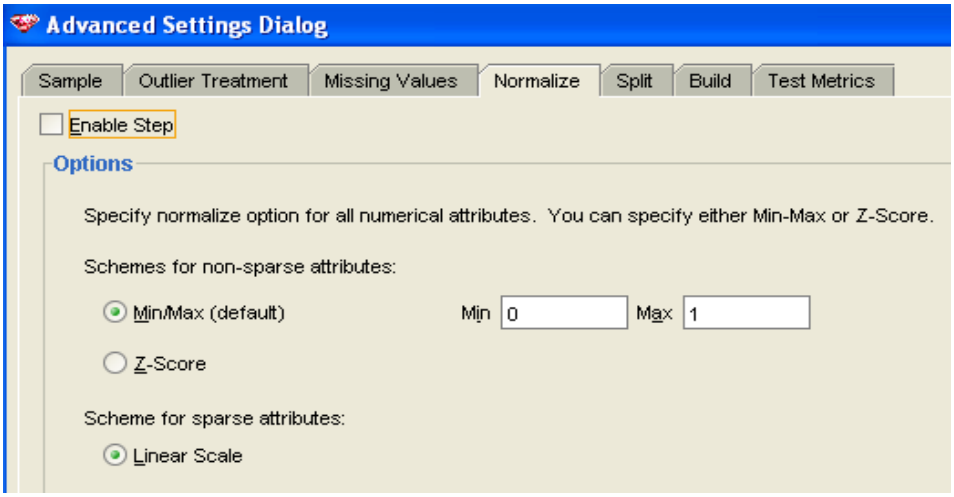
Şekil 4.15 : Gelişmiş ayarlar, "Outlier Treatment" sekmesi

"Outlier Treatment" sekmesinde aykırı değerler üzerine işlemler yapılır. Standart sapması 3'ten fazla olan değerlerin uygulamaya olabilecek olumsuz etkilerini engellemek amacıyla düzenlenmesi için kullanılan bir sekmedir. Bir önceki veri temizleme aşamasında bu tür uç değerler zaten tablodan çıkarıldığı için bu işlem uygulamaya katılmamıştır.



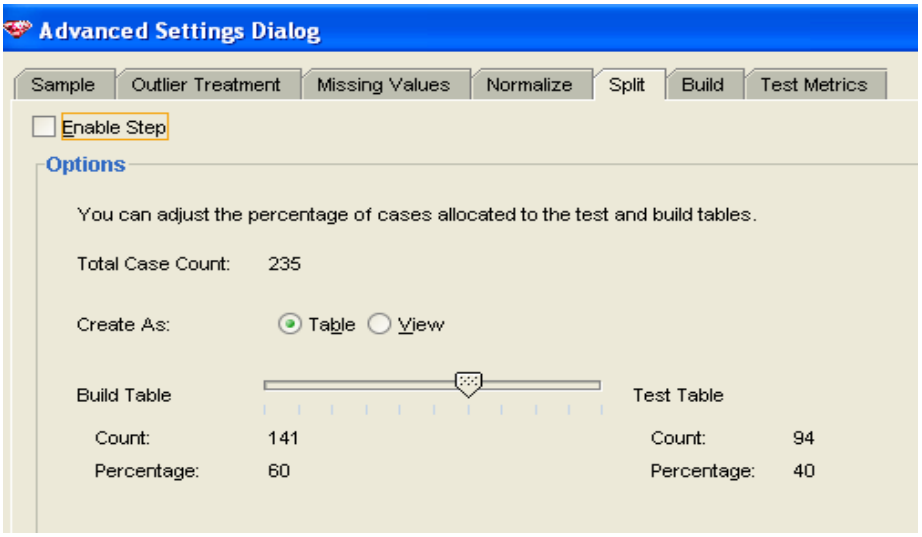
Şekil 4.16 : Gelişmiş ayarlar, "Missing Values" sekmesi

"Missing Values" sekmesinde kayıp değerler üzerine işlemler yapılır. Rakamsal değerler için kayıp değerlerin, varsayılan olarak atanan işlemi ortalama ile değiştirilmesi işlemidir. Fakat bu uygulamada veri temizleme aşamasında bu tür kayıp değerler tablodan çıkarıldığı için bu işlem uygulamaya katılmamıştır.



Şekil 4.17 : Gelişmiş ayarlar, "Normalize" sekmesi

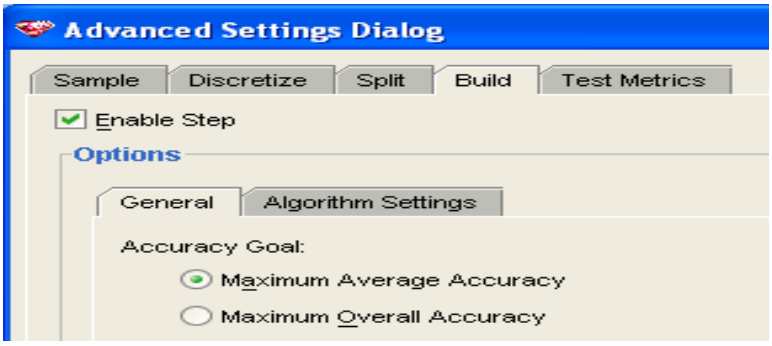
Bu sekme uç değerleri normalize etme sekmesidir. Veri temizleme aşamasından uç değerler tablodan çıkarıldığı için bu işlem uygulamaya katılmamıştır.



Şekil 4.18 : Gelişmiş ayarlar, "Split" sekmesi

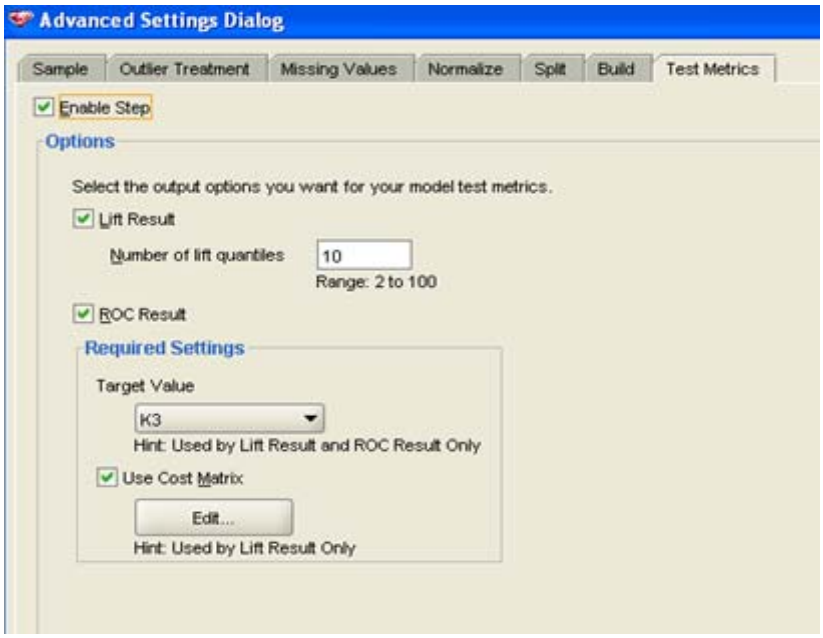
"Split" sekmesi, test için ayrılacak veri yüzdesini ayarlamak içindir. Bu uygulamada veri sayısı çok fazla olmadığından verilerin tümü "Built" aktivitesi için kullanılmış olduğundan bu işlemi gerçekleştirmeye yarayan "Split" seçeneğini kaldırmak için kendinden seçili olan "Enable Step" seçimi kaldırılıp çalışmaması sağlanmıştır.

Yukarıdaki tanımlar çerçevesinde, bu uygulama için “Sample”, “Outlier Treatment”, “Missing Values”, “Normalize” ve “Split” seçeneklerinin kullanılmaması gerektiği açıktır ve bu nedenle modelde bunların seçili olmamasına dikkat edilmiştir.



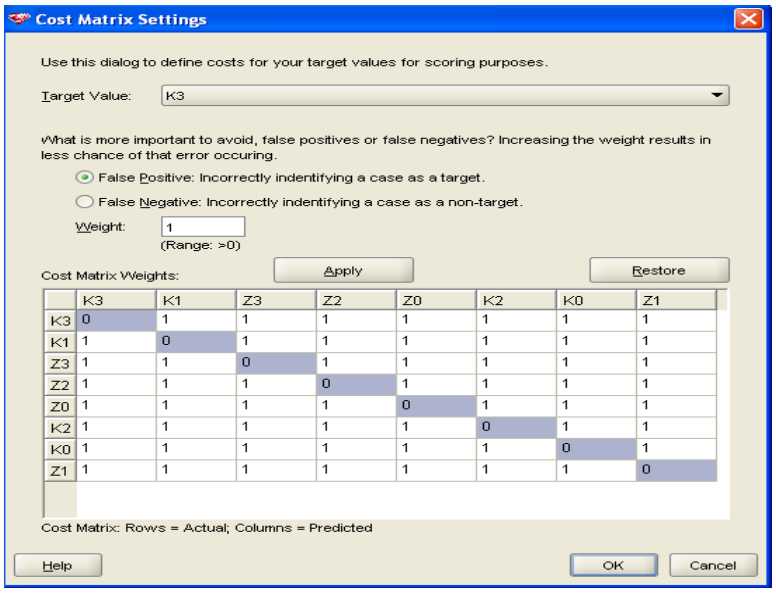
Şekil 4.19 : Gelişmiş ayarlar, “Build → General” sekmesi

“Build” ayarları iki bölüme ayrılmıştır. “General” sekmesinde, modelin genel doğruluğu artırıcı yönde mi (“Maximum Overall Accuracy”), yoksa her bir hedef değer için yüksek tahminde bulunan ve ortalama doğruluğu artırıcı yönde mi (“Maximum Average Accuracy”) olacağı seçilir. Örneğin model, en çok kar eden hissenin tahminini yüksek doğrulukla yaparken en çok zarar eden hissenin tahminini daha düşük doğrulukla yapıyor olabilir. Açıktır ki modelin hedef olarak seçilen sütundaki her bir değeri yüksek doğrulukla tahmin etmesi istenecektir. Bu sebeple “Maximum Average Accuracy” seçeneği model kurulurken varsayılan olarak seçilidir ve o şekilde bırakılmıştır [9].



Şekil 4.20a : Gelişmiş ayarlar, “Test Metrics” sekmesi

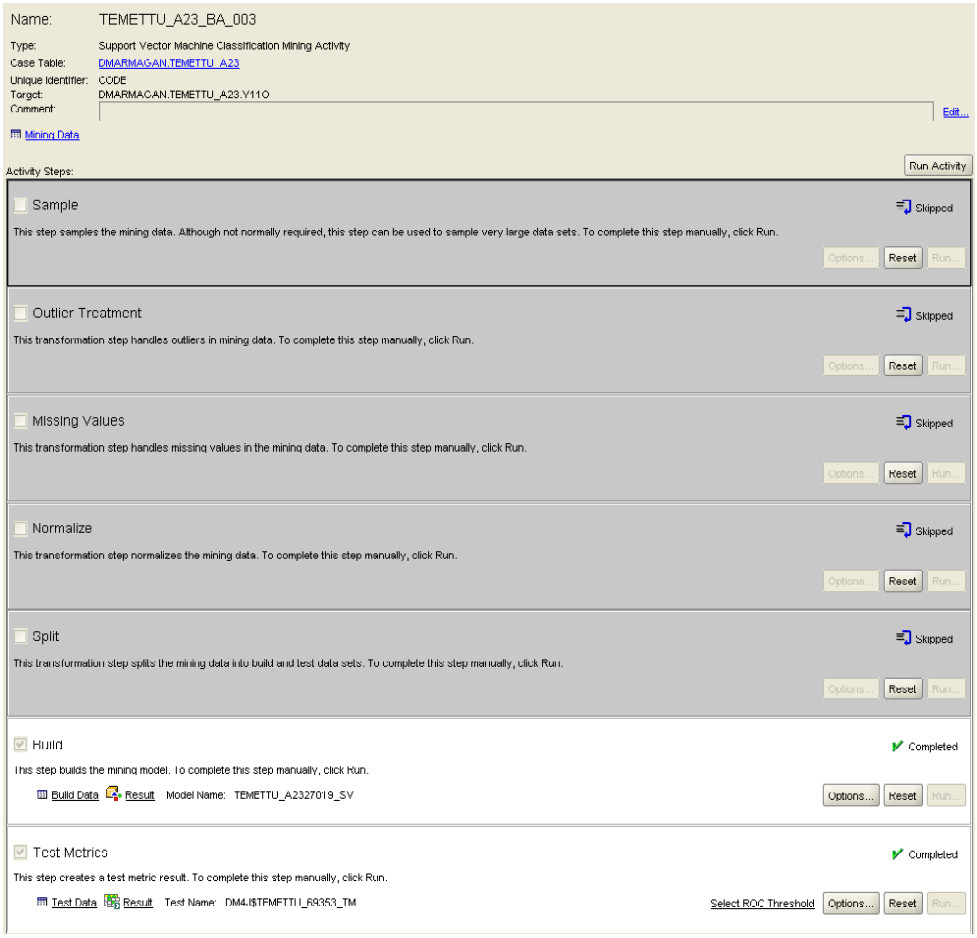
"Test Metrics" sekmesinde bulunan "Cost Matrix" seçeneği de seçilmelidir, çünkü burada tahmin edilecek olan Y11O sütununda K3 değerlerinin diğerlerine göre daha iyi tahmin edilmesi istenmektedir. "Use Cost Matrix" seçeneğinin altındaki "Edit" tuşuna basılarak modelde kullanılan cost matrix Şekil 4.20b'deki gibi görüntülenebilir.



	K3	K1	Z3	Z2	Z0	K2	K0	Z1
K3	0	1	1	1	1	1	1	1
K1	1	0	1	1	1	1	1	1
Z3	1	1	0	1	1	1	1	1
Z2	1	1	1	0	1	1	1	1
Z0	1	1	1	1	0	1	1	1
K2	1	1	1	1	1	0	1	1
K0	1	1	1	1	1	1	0	1
Z1	1	1	1	1	1	1	1	0

Şekil 4.20b : "Cost Matrix"in görüntülenmesi

Modelde, yanlış-negatif tahminler engellenmek istendiği için matrisin sol alt köşesinde bulunan katsayının sağ üst köşesinde bulunan katsayıya göre daha büyük olması gerekmektedir. Örneğin, yanlış-negatif tahminin verdiği zarar, yanlış-pozitif bir tahminin verdiği zararın 6 katı olan bir cost matrix olsun. "Evet" tahminin olasılığının 0.8 olduğu kabul edilsin. Böylece yanlış-negatif bir tahminin vereceği zarar 0.2 * 6 = 1.2 olarak bulunurken, yanlış-pozitif bir tahminin vereceği zarar ise 0.8 * 1 = 0.8 olarak hesaplanır. Böylece model tarafından, düşük zararlı "Evet" tahmini yapılacaktır [10]. Böylece yanlış-negatif bir tahminin modele verdiği zarar, yanlış-pozitif bir tahminin modele verdiği zarardan daha büyük olarak hesaplanacak ve yanlış-negatif tahminler azalacaktır. Gelişmiş ayarlar yapıldıktan sonra model oluşturma süreci başlatılır. Bu sürecin tamamlanmış hali Şekil 4.21'de görüldüğü gibidir.



Şekil 4.21 : Model oluşturma sürecinin son hali

4.3.3 Modelin Testi

Bu uygulamada çok fazla sayıda veri olmadığı için test, modelin üzerinden yapılmıştır. Şekil 4.21'deki "Result" kısmından yararlanarak modelin güvenilirliği incelenecektir.

File Publish Help

Predictive Confidence | Accuracy | ROC | Lift | Test Settings | Task

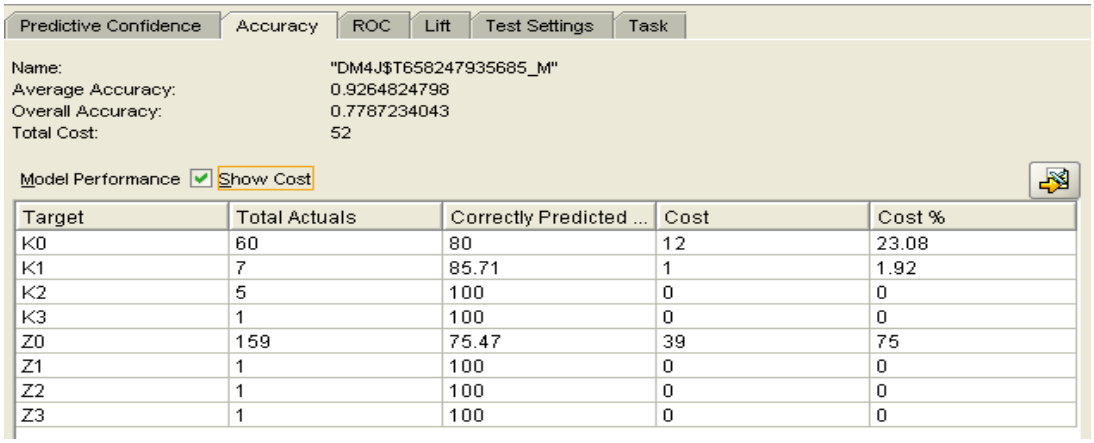
Name: "DM4J$T658247935685_M"
Average Accuracy: 0.9264824798
Overall Accuracy: 0.7787234043
Total Cost: 52

Model Performance ☐ Show Cost

Target	Total Actuals	Correctly Predicted %
K0	60	80
K1	7	85.71
K2	5	100
K3	1	100
Z0	159	75.47
Z1	1	100
Z2	1	100
Z3	1	100

Şekil 4.22a : "Test Metrics"in sonucu, "Accuracy" sekmesi

"Accuracy" sekmesine tıklandığında, modelin test tablosuna uygulandığında elde edilen sonucun doğruluk oranları görüntülenir. Tahmin edilmek istenen hedef sütunundaki gerçek değerler bilinmektedir, böylece modelin yaptığı tahminler gerçek değerlerle karşılaştırılabilir. Modelde, Y11O değeri "K0" olan 60 kişi vardır ve model bunların %80' ini doğru olarak tahmin etmiştir. Aynı şekilde Y11O değeri "Z0" olan 159 kişiden %75.47' sinin tahmini doğru olarak yapılmıştır. "Show Cost" butonuna tıklandığında, yanlış tahminin modele vereceği zarar aşağıdaki gibi görüntülenir. Düşük "Cost" değeri, modelin başarılı olduğu anlamına gelir [9].



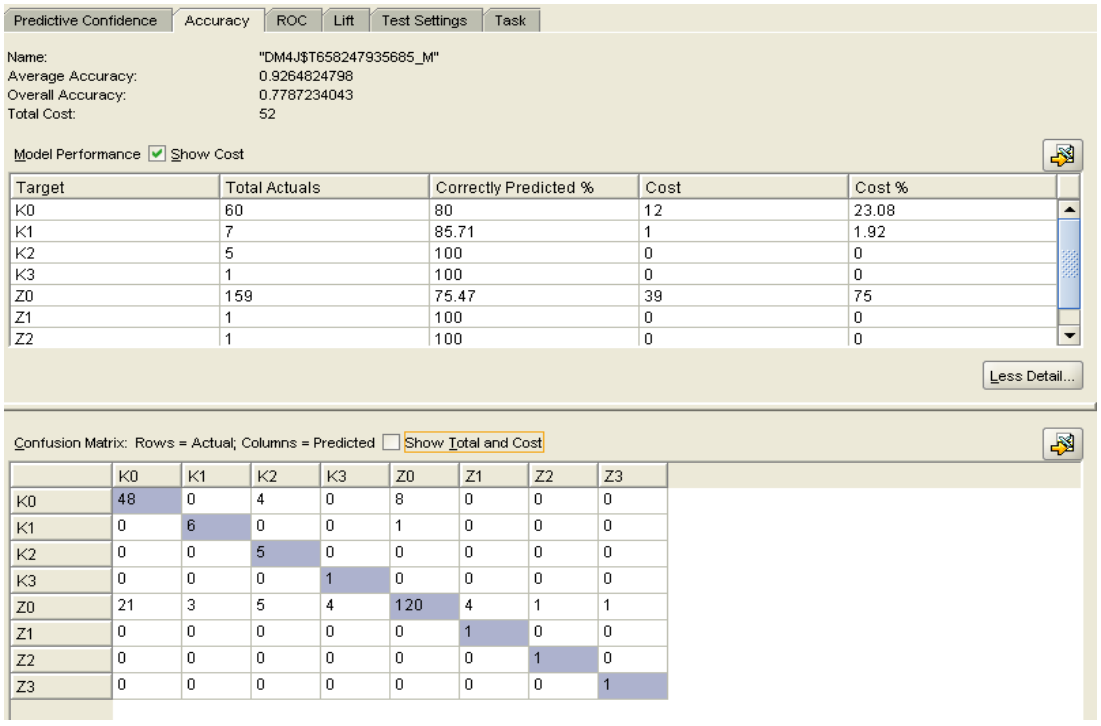
Predictive Confidence | Accuracy | ROC | Lift | Test Settings | Task

Name: "DM4J$T658247935685_M"
Average Accuracy: 0.9264824798
Overall Accuracy: 0.7787234043
Total Cost: 52

Model Performance ☑ Show Cost

Target	Total Actuals	Correctly Predicted ...	Cost	Cost %
K0	60	80	12	23.08
K1	7	85.71	1	1.92
K2	5	100	0	0
K3	1	100	0	0
Z0	159	75.47	39	75
Z1	1	100	0	0
Z2	1	100	0	0
Z3	1	100	0	0

Şekil 4.22b : : "Test Metrics"in sonucu, "Accuracy"de cost'un görüntülenmesi



Predictive Confidence | Accuracy | ROC | Lift | Test Settings | Task

Name: "DM4J$T658247935685_M"
Average Accuracy: 0.9264824798
Overall Accuracy: 0.7787234043
Total Cost: 52

Model Performance ☑ Show Cost

Target	Total Actuals	Correctly Predicted %	Cost	Cost %
K0	60	80	12	23.08
K1	7	85.71	1	1.92
K2	5	100	0	0
K3	1	100	0	0
Z0	159	75.47	39	75
Z1	1	100	0	0
Z2	1	100	0	0

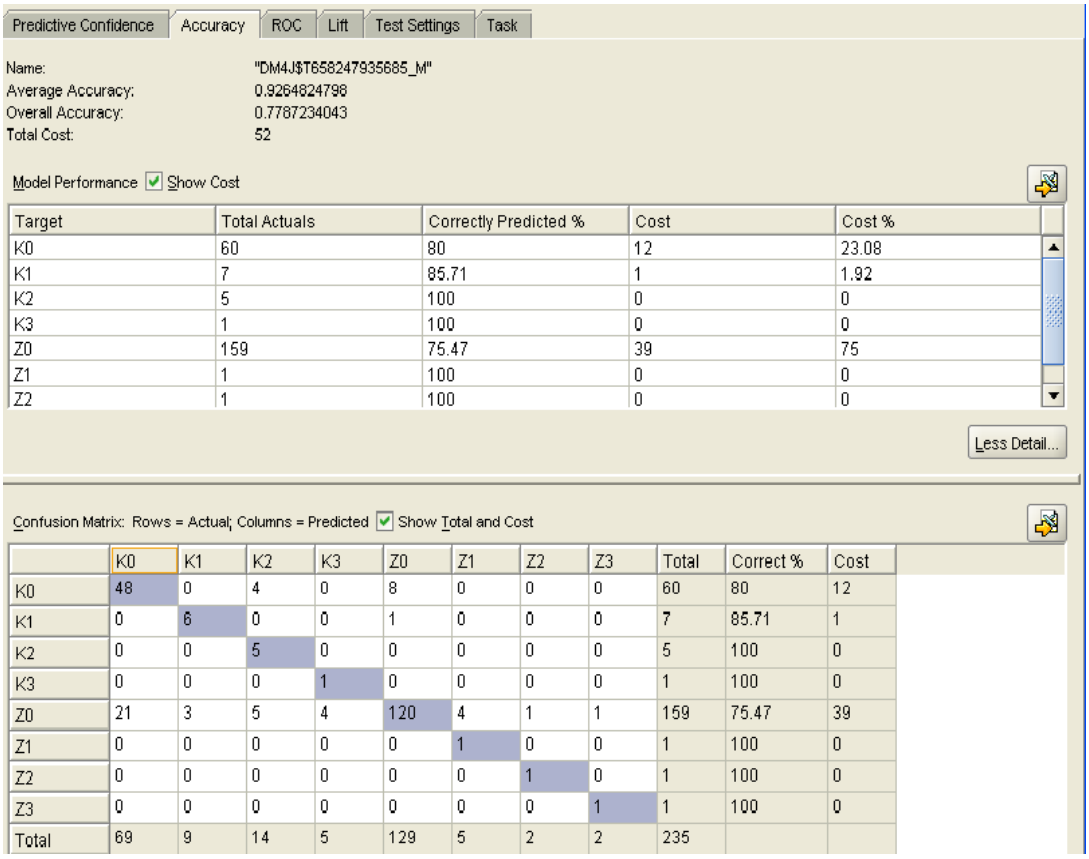
Less Detail...

Confusion Matrix: Rows = Actual; Columns = Predicted ☐ Show Total and Cost

	K0	K1	K2	K3	Z0	Z1	Z2	Z3
K0	48	0	4	0	8	0	0	0
K1	0	6	0	0	1	0	0	0
K2	0	0	5	0	0	0	0	0
K3	0	0	0	1	0	0	0	0
Z0	21	3	5	4	120	4	1	1
Z1	0	0	0	0	0	1	0	0
Z2	0	0	0	0	0	0	1	0
Z3	0	0	0	0	0	0	0	1

Şekil 4.22c : "Test Metrics"in sonucu, "Accuracy"de güvenilirlik matrisinin görüntülenmesi

Sonraki adımda “More Detail” a tıklanarak güvenilirlik matrisi (Confusion Matrix) görüntülenmiştir. Bu matriste, hedef sütunundaki gerçek değerler ile modelin test tablosuna uygulanarak yapılan tahmin değerlerinin sayısı gösterilmektedir. Test tablosundaki Y11O’nun gerçek değerleri bilinmektedir ve bu değerler güvenilirlik matrisinin satırlarındaki değerlerdir. Matrisin sütunları ise modelin yapmış olduğu tahminleri göstermektedir. Örneğin, matrisin 5.satır 1. sütununda yer alan 21 sayısı yanlış-pozitif tahminleri, yani gerçek değer Z0 iken K0 şeklinde tahmin edilen hedef sayısını gösterir. Buna benzer olarak 1.satır 5.sütunda bulunan 8 sayısı ise yanlış-negatif tahminleri, yani gerçek değer K0 iken Z0 şeklinde tahmin edilen hedef sayısını göstermektedir [9]. Matrisin köşegenindeki sayılar ise doğru olarak yapılan tahmin sayılarıdır.

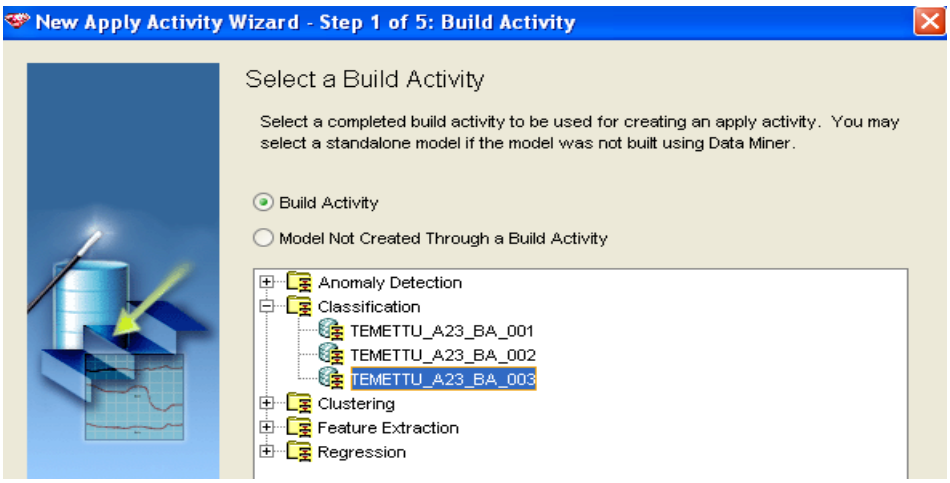


Şekil 4.22d “Test Metrics”in sonucu, “Accuracy”de detaylı güvenilirlik matrisinin görüntülenmesi

Son olarak, “Show Total and Cost”a tıklanarak güvenilirlik matrisinden elde edilen, kayıt sayıları, doğru tahmin yüzdesi, yanlış tahminin modele vereceği zarar gibi istatistiksel bilgiler görüntülenmiştir [9].

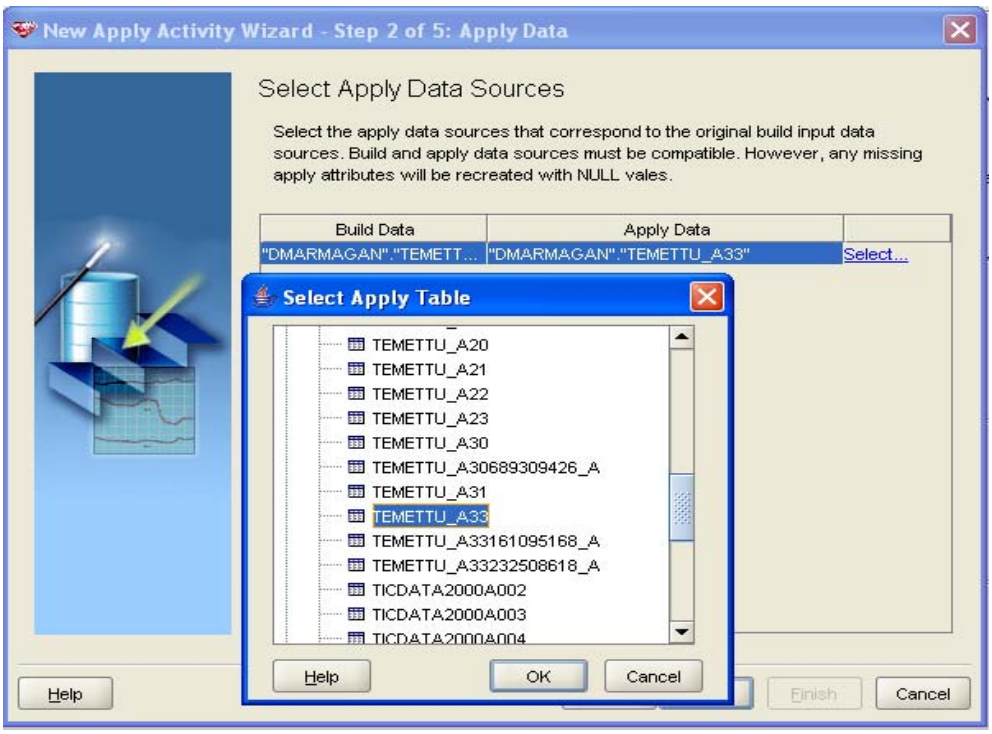
4.3.4 Modelin Uygulanması

Bu aşamada önceki adımda oluşturulan modelin, Build tablosuyla aynı formatta hazırlanan 2000-2010 yılları arasındaki net karların tablosuna uygulanması ayrıntılı olarak anlatılmaktadır.



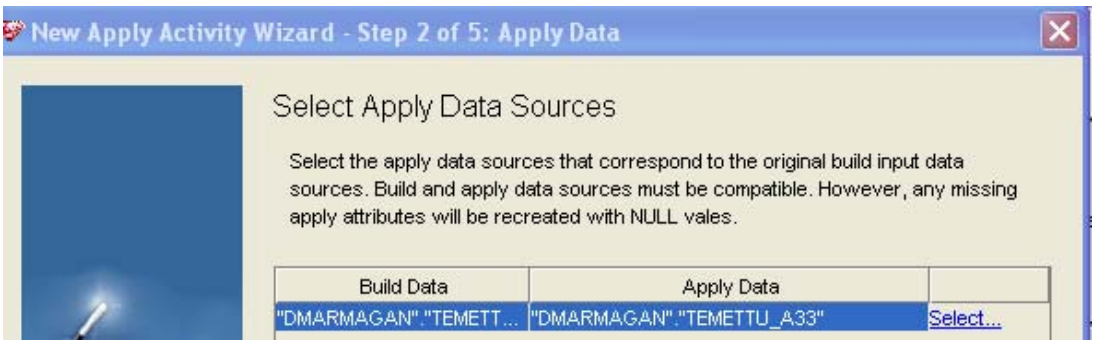
Şekil 4.23 : Uygulama 1. aşama, uygulama yapılacak modelin seçimi

Uygulama için listelenen modellerden, bir önceki aşamada oluşturulan TEMETTU_A23_BA_003 modeli seçilmiştir ve bir sonraki adımda modelin uygulanacağı tablo Şekil 4.24a'daki gibi görüntülenmiştir.

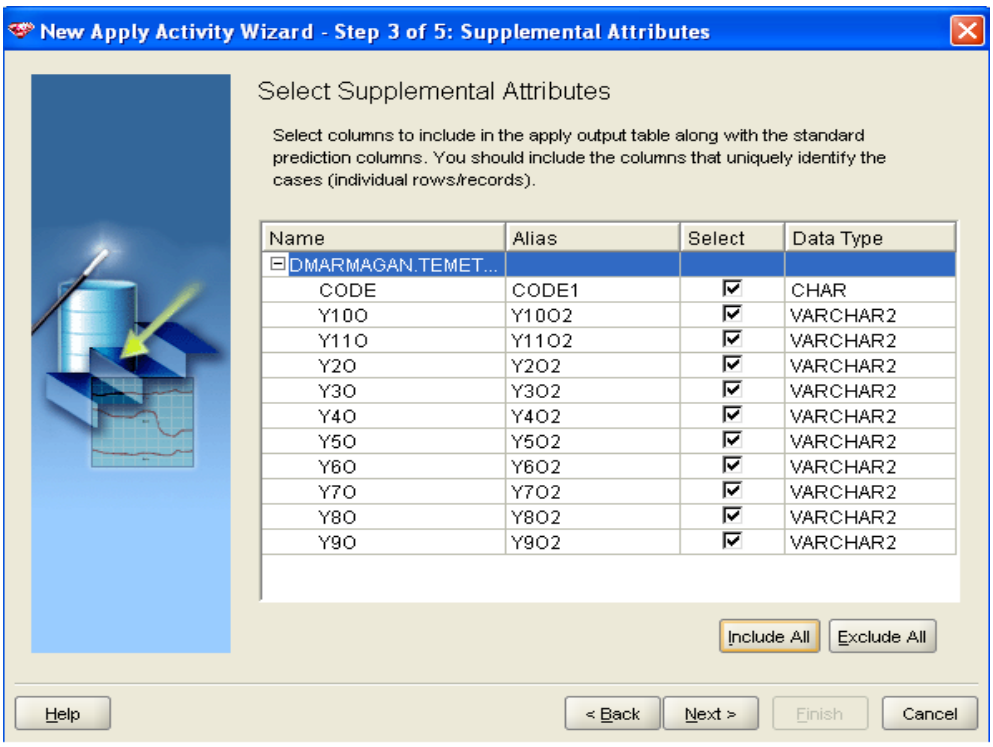


Şekil 4.24a : Uygulama 2. aşama, uygulama yapılacak tablonun seçimi

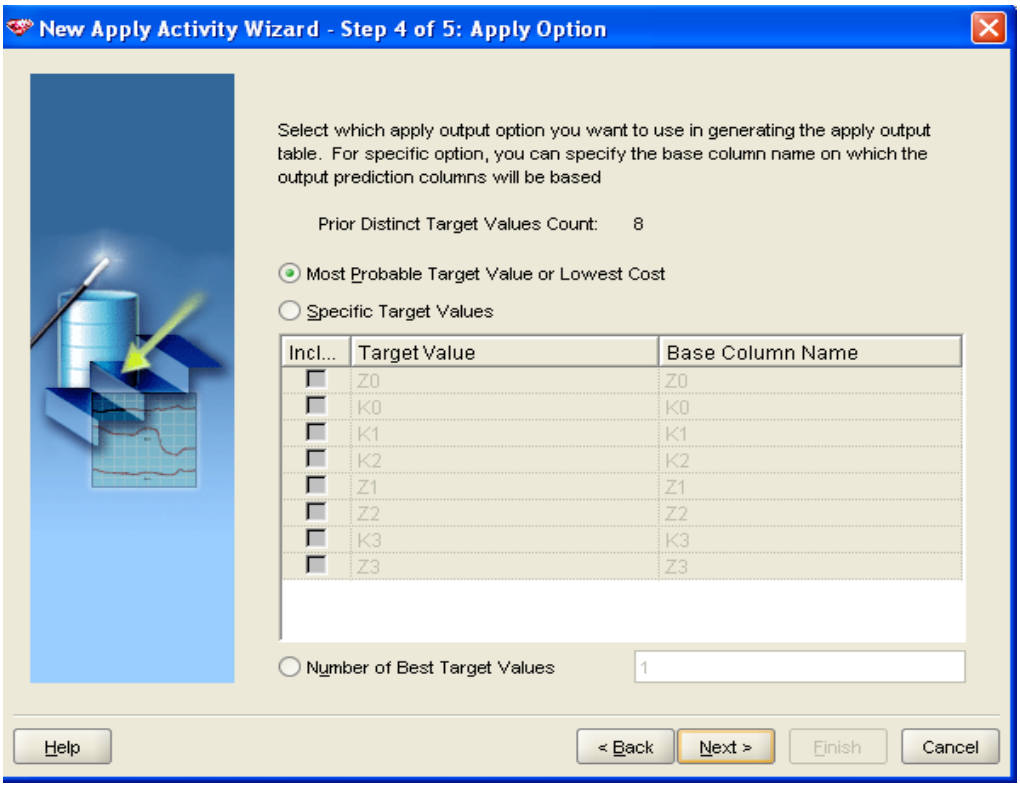
Uygulama sonucunda görüntülenmek istenen sütunlar seçilerek bir sonraki adıma geçilmiştir.



Şekil 4.24b : Seçili tablonun "Apply Data" altında gösterimi



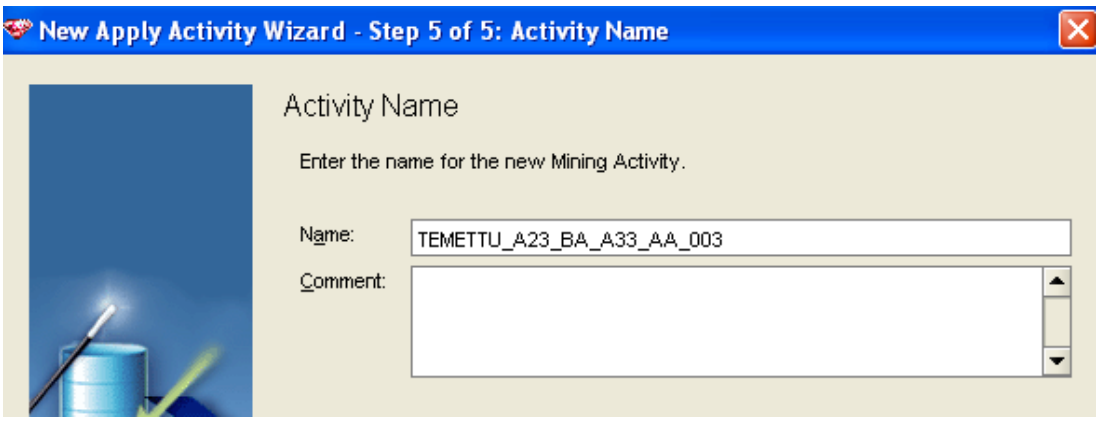
Şekil 4.25 : Uygulama 3. aşama, sonuç tablosunda gösterilecek sütunların seçimi



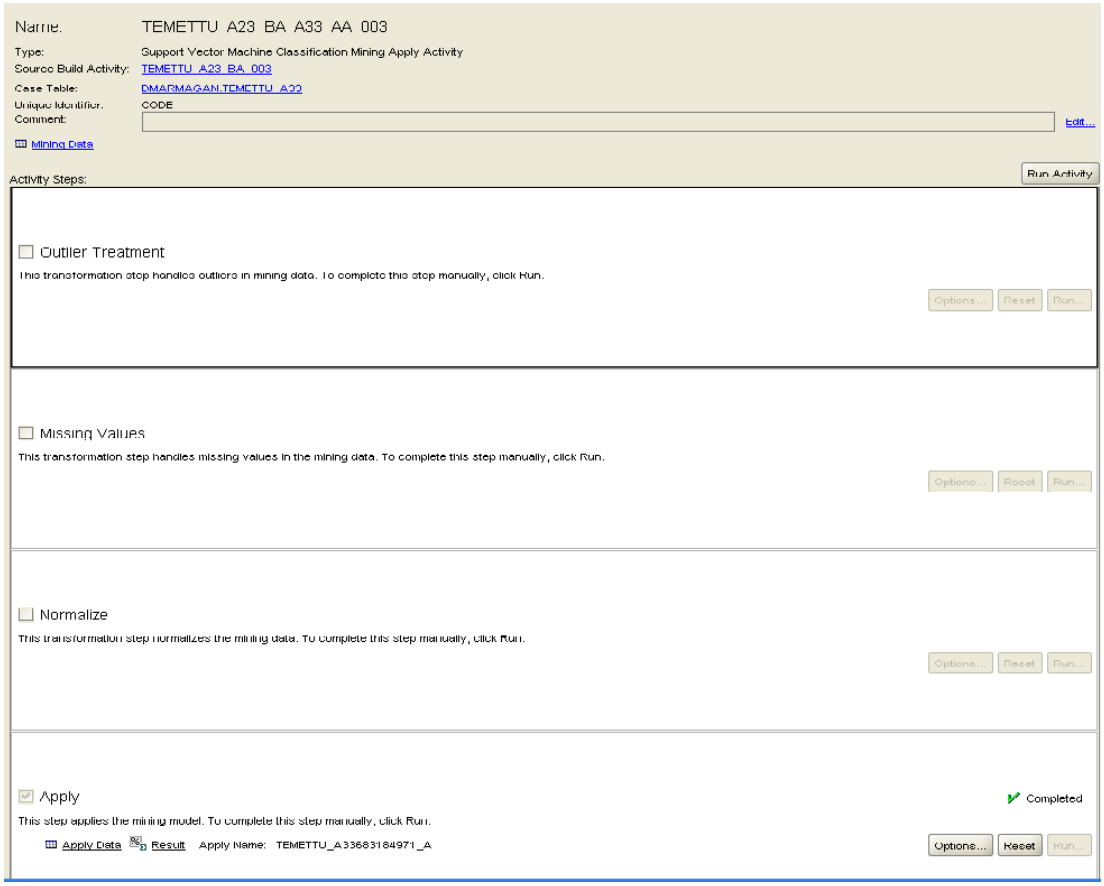
Şekil 4.26 : Uygulama 4. aşama, tahmin yönteminin seçimi

Her bir kayıt için en yüksek olasılığa sahip hedef değerler (Z0, Z1, Z2, Z3, K0, K1, K2, K3) görüntülenmek istendiğinden "Most Probable Target Value or Lowest Cost" seçeneği seçilmiştir. Bize vereceği değer, her bir kayıtta hedef değerler için hangi olasılık değeri daha yüksekse sonuç olarak o değer "Prediction" sütununda görüntülenir.

Son adımda uygulamaya isim verilerek uygulama süreci başlatılmıştır.



Şekil 4.27 : Uygulama 5. aşama, aktivitenin isminin belirlenmesi

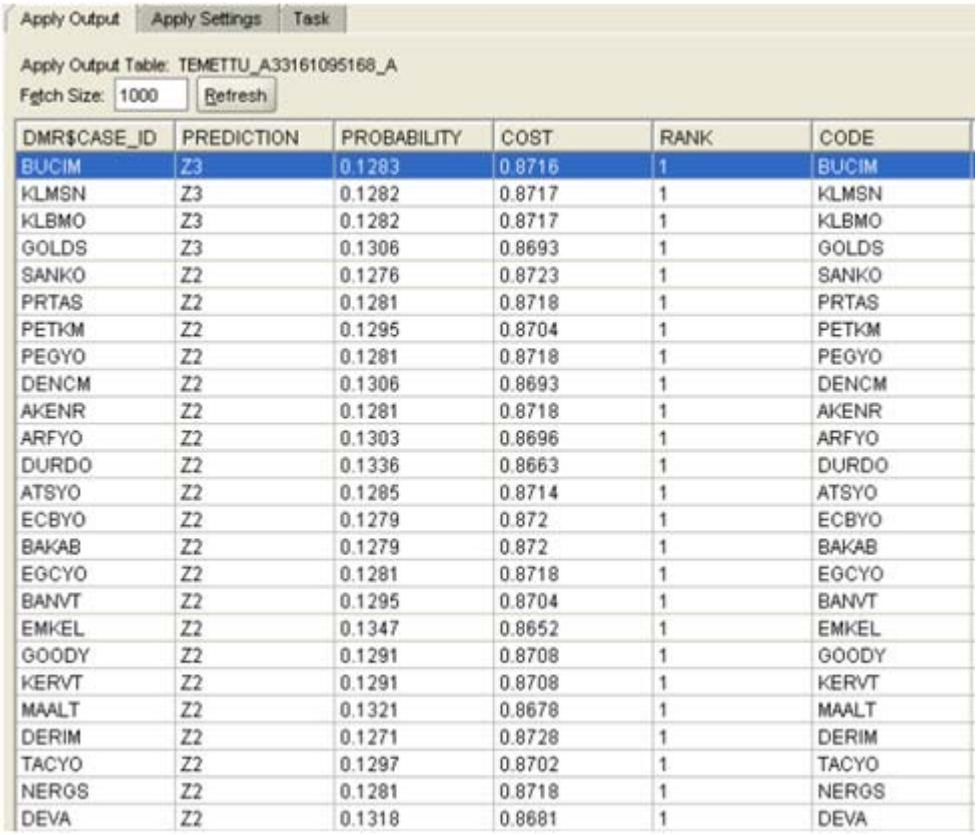


Şekil 4.28 : Uygulama sürecinin son hali

Süreç tamamlandıktan sonra "Result" yolundan istenen olasılık ve tahminler görüntülenebilmektedir.

4.3.5 Sonuçlar ve Yorumlar

Aşağıda sonuç olarak verilen tüm ekran çıktılarında, tahminleri "Prediction" sütununda, tahminlerin olma olasılığı ise "Probability" sütununda görülmektedir. Diğer sütunlar ise girdi verilerinde de bulunan ve karşılaştırma için buraya da dahil edilen sütunlardır.



Apply Output | Apply Settings | Task

Apply Output Table: TEMETTU_A33161095168_A

Fetch Size: 1000 Refresh

DMR$CASE_ID	PREDICTION	PROBABILITY	COST	RANK	CODE
BUCIM	Z3	0.1283	0.8716	1	BUCIM
KLMSN	Z3	0.1282	0.8717	1	KLMSN
KLBMO	Z3	0.1282	0.8717	1	KLBMO
GOLDS	Z3	0.1306	0.8693	1	GOLDS
SANKO	Z2	0.1276	0.8723	1	SANKO
PRTAS	Z2	0.1281	0.8718	1	PRTAS
PETKM	Z2	0.1295	0.8704	1	PETKM
PEGYO	Z2	0.1281	0.8718	1	PEGYO
DENCM	Z2	0.1306	0.8693	1	DENCM
AKENR	Z2	0.1281	0.8718	1	AKENR
ARFYO	Z2	0.1303	0.8696	1	ARFYO
DURDO	Z2	0.1336	0.8663	1	DURDO
ATSYO	Z2	0.1285	0.8714	1	ATSYO
ECBYO	Z2	0.1279	0.872	1	ECBYO
BAKAB	Z2	0.1279	0.872	1	BAKAB
EGCYO	Z2	0.1281	0.8718	1	EGCYO
BANVT	Z2	0.1295	0.8704	1	BANVT
EMKEL	Z2	0.1347	0.8652	1	EMKEL
GOODY	Z2	0.1291	0.8708	1	GOODY
KERVT	Z2	0.1291	0.8708	1	KERVT
MAALT	Z2	0.1321	0.8678	1	MAALT
DERIM	Z2	0.1271	0.8728	1	DERIM
TACYO	Z2	0.1297	0.8702	1	TACYO
NERGS	Z2	0.1281	0.8718	1	NERGS
DEVA	Z2	0.1318	0.8681	1	DEVA

Şekil 4.29 : Uygulama sonucu

Uygulama sonucu, sayfanın sağ üst köşesinde yer alan "Export" tuşuna basılarak dışarı aktarılıp Excel tablosuna dönüştürülerek incelenmiştir. Sonuçlar PREDICTION sütununa göre gruplanarak sıralanmıştır. (Yukarıdaki tabloda yer sıkıntısından dolayı sadece verilerin bir kısmı konmuştur.) Uygulamadaki veriler çok fazla olmadığı için tüm veriler birlikte incelenmiştir. Elde edilen sonuçlar açıktır. Örneğin, "BUCIM" hissesinin 2010 yılındaki kar-zarar aralığı 0.1283 olasılıkla Z3 aralığında tahmin edilmiştir.

KAYNAKLAR

[1] Yapıcı A., Özel A., Ayça C., 2010. Oracle Data Miner ile Kredi Ödemeleri Üzerine Bir Veri Madenciliği Uygulaması, pp. 9, 11, 17, 47, 48.

[2] Frawley W. J., Shapiro G. P., Matheus C. J., 1992. *Discovery in Databases: An Overview*, AI Magazine, **13**-3, 57-70.

[3] Berry J. A., Linoff G., 1997. *Data mining techniques for marketing, sales and customer support*, John Wiley & Sons Inc., New York.

[4] Holshemier M., Siebes A., 1994. *Data mining <http://www.pcc.qub.ac.uk>*

[5] Cabena P., Hadjnian P., Stadler R., 1998. *Discovering data mining from concept to implementation*, Prentice Hall PTR, NJ.

[6] Aldana, W. A., *Data Mining Industry: Emerging Trends and New Opportunities*, p.11.

[7] Özkan, Y., *Veri Madenciliği Yöntemleri*, p.41.

[8] Kırış A., Demiryürek U., 2005. Malzeme Talep ve Sarf Miktarlarının Veri Madenciliği Yöntemleri ile Öngörülmesi, p. 23.

[9] ORACLE CORPORATION, 2006. *Oracle 10g Release 2 Data Mining Tutorial,* pp. 78, 81, 82, 83, 84, 85, 86, 96.

[10] Taft M., Krishnan R., Hornick M., Muhkin D., Tang G., Thomas S., Stengard P., 2005. *Oracle Data Mining Concepts, 10g Release 2,* p. 14, Oracle Corporation, CA.