T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ORTA ÖĞRETİM OKULLARI İÇİN ÖĞRENCİ OTOMASYONU TASARIMI VE ÖĞRENCİ VERİLERİ ÜZERİNE VERİ MADENCİLİĞİ UYGULAMALARI

Kerem ÜÇGÜN

YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR-KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Ali BULDU

İSTANBUL 2009

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ORTA ÖĞRETİM OKULLARI İÇİN ÖĞRENCİ OTOMASYONU TASARIMI VE ÖĞRENCİ VERİLERİ ÜZERİNE VERİ MADENCİLİĞİ UYGULAMALARI

Kerem ÜÇGÜN
(141100420060235)

YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR-KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Yrd. Doç. Dr. Ali BULDU

İSTANBUL 2009

TEŞEKKÜR

Bu tez çalışmasında bana yol gösteren, olumlu yaklaşımları ile sürekli teşvik eden, her konuda destek veren tez danışanım Yrd. Doç. Dr. Ali BULDU' ya,

Çalışmalarım sırasında beni yalnız bırakmayarak her konuda bana sabırla destek olan aileme,

teşekkürlerimi sunarım.

Temmuz, 2009 **Kerem ÜÇGÜN**

İÇİNDEKİLER

SAYFA

TEŞEKKÜR ..... **İ**

İÇİNDEKİLER ..... **İİ**

ÖZET ..... **V**

ABSTRACT ..... **Vİ**

SEMBOLLER ..... **Vİİ**

KISALTMALAR ..... **Vİİİ**

ŞEKİLLER ..... **İX**

TABLOLAR ..... **Xİİ**

BÖLÜM I. GİRİŞ VE AMAÇ **1**

I.1. GİRİŞ **1**

I.2. TEZ ÇALIŞMASININ AMACI **2**

BÖLÜM II. VERİ MADENCİLİĞİ KAVRAMINA GENEL BAKIŞ **3**

II.1. GENEL BAKIŞ **3**

II.1.1. Veri Madenciliği İle İlgili Önceden Yapılmış Çalışmalar **4**

II.1.1.1. Öğrenci Bilgi Sisteminde Veri Madenciliği Uygulaması . **4**

II.1.1.2. Apriori Algoritması İle Öğrenci Başarı Analizi **4**

II.1.1.3 Zeki Öğretim Sistemlerinde Veri Madenciliği Kullanılması **4**

II.1.1.4. Öğrenci Seçme Sınavında Öğrenci Başarımını Etkileyen Faktörlerin Veri Madenciliği Yöntemleriyle Tespiti **5**

II.1.1.5. Üniversite Öğrencilerinin İnterneti Eğitimsel Amaçlar İçin Kullanmalarını Etkileyen Faktörlerin Veri Madenciliği Yöntemleriyle Tespiti **5**

II.1.1.6. Web Tabanlı Uzaktan Eğitimde İki Farklı Öğretim Modelinin Öğrenci Başarısı Üzerindeki Etkilerinin İncelenmesi ... **5**

II.2. VERİ TABANI VE VERİ AMBARI KAVRAMLARI **5**
II.2.1. Veri Tabanı **5**
II.2.2. Veri Ambarı **6**
II.3. VERİ TABANLARINDA BİLGİ KEŞFİ SÜRECİNDEKİ ADIMLAR **6**
II.4. VERİ MADENCİLİĞİNİN BAZI UYGULAMA ALANLARI **8**
II.5. VERİ MADENCİLİĞİNDE KARŞILAŞILAN PROBLEMLER **9**
II.6. VERİ SEÇİMİ, ÖN İŞLEME VE İNDİRGEME **9**
II.6.1. Veri Seçimi **9**
II.6.2. Veri Önişleme **10**
II.6.2.1. Veri Tipi Dönüşümü **10**
II.6.2.2 Gruplama **11**
II.6.2.3. Kayıp Değerleri Yönetmek **11**
II.6.2.4. Gürültülü Veri **11**
II.6.3. Veri İndirgeme **11**
II.7. VERİ MADENCİLİĞİNDE KULLANILAN TEKNİKLER **12**
II.7.1. Sınıflama (Classification) ve Regresyon (Regression) **13**
II.7.2. Kümeleme **13**
II.7.3. Ardışıl Örüntü **14**
II.7.4. Birliktelik Kuralları **14**
II.7.4.1. Birliktelik Kuralları Temel Kavramları **15**
II.7.4.2. Güven (Confidence) ve Destek (Support) Ölçüleri **17**
II.7.4.3. Birliktelik Kurallarının Başarılı Ve Başarısız Olduğu Durumlar **19**
II.7.4.4. Birliktelik Kurallarının Belirlenmesinde Kullanılan Temel Algoritmalar **19**
BÖLÜM III. OKUL OTOMASYON YAZILIMININ GELİŞTİRİLMESİ **21**
III.1. ARAŞTIRMA YÖNTEMİ **22**
III.2. ARAŞTIRMA ARAÇLARI **22**
III.2.1. Donanım **22**
III.2.2. Yazılım **22**
III.3. OTOMASYON YAZILIMININ BÖLÜMLERİ **22**

III.4. VERİ TABANINDA TABLO YAPILARININ OLUŞTURULMASI **23**
III.5. OTOMASYON YAZILIMININ OLUŞTURULMASI **29**
III.5.1. Yazılımı Geliştirme Ortamı **29**
III.5.2. Otomasyon Yazılımının Aşamaları **30**
III.5.2.1. Temel Bilgiler Menüsü **33**
III.5.2.2. Öğrenci Bilgileri Menüsü **40**
III.5.2.3. Devamsızlık Bilgileri Menüsü **45**
III.5.2.4. Belgeler Menüsü **47**
III.5.2.5. Öğretmen Bilgileri Menüsü **50**
BÖLÜM IV. UYGULAMA VE SONUÇLAR **52**
IV.1. APRIORI ALGORİTMASI **53**
IV.2. OTOMASYON YAZILIMININ BAŞARI ANALİZİ BÖLÜMÜNÜN TASARIMI **57**
IV.2.1. Genel Liste **57**
IV.2.2. Maden Liste **57**
IV.2.3. Ders Listesi **58**
IV.2.4. Diğer İşlemler **59**
IV.2.4.1. 1.Adım **59**
IV.2.4.2. 2.Adım **59**
IV.2.4.3. 3. Adım **61**
IV.2.4.4. 4. Adım **62**
IV.2.4.5. 5. Adım **66**
IV.2.4.6. 6. Adım **67**
IV.2.4.7. 7. Adım **72**
IV.2.4.8. 8. Adım **75**
IV.2.4.9. 9. Adım **76**
IV.2.4.10. 10. Adım **77**
IV.2.4.11. Kurallar **78**
IV.3. VERİ MADENCİLİĞİ UYGULAMASININ ÇALIŞTIRILMASI **78**
BÖLÜM V. DEĞERLENDİRME VE ÖNERİLER **89**
KAYNAKLAR **91**
ÖZGEÇMİŞ **93**

ÖZET

ORTA ÖĞRETİM OKULLARI İÇİN ÖĞRENCİ OTOMASYONU TASARIMI VE ÖĞRENCİ VERİLERİ ÜZERİNE VERİ MADENCİLİĞİ UYGULAMALARI

Teknolojinin hızla gelişmesi ile günümüzde her alanda saklanan veri miktarı sürekli artmaktadır. Bu verilerden veri madenciliği teknikleri uygulanarak, önceden bilinmeyen, anlamlı, değerli bilgiler elde edilmek istenmektedir. Birliktelik-ilişki kuralı da bu tekniklerden biridir. Birliktelik-ilişki kurallarının belirlenmesinde en sık kullanılan algoritma ise apriori algoritmasıdır.

Apriori algoritması, veri madenciliğinde sık geçen öğelerin keşfedilmesi için en çok kullanılan tekniklerden biridir. Sık geçen öğeleri belirlemek için veri tabanı birçok kez taranır. Bu tarama esnasında apriori algoritmasının birleştirme, budama işlemleri ve minimum destek ve güven değerleri yardımları ile birbirleri ile ilişkili öğeler bulunur.

Okul otomasyon yazılımları öğrencilere ve onların okul durumlarına ait birçok veriyi barındırırlar. Bu verilerin çokluğuna ve zenginliğine rağmen, değerli bilgiler veritabanında gizlidir. Okullarda öğrenimin verimliliğini artırmak ve gelecek planları yapabilmek için yararlı bilgilere ihtiyaç vardır. Veri madenciliği teknikleri öğrenciler ile ilgili verilerden gizli kalmış önemli bilgileri ortaya çıkarır ve böylece bu teknikler eğitim-öğretim için değerli bilgiler sağlarlar.

Bu tez çalışmasında okul otomasyon yazılımı hazırlanması ve bu yazılımda öğrenci verileri üzerinde veri madenciliği uygulaması anlatılmıştır. İ.M.K.B Ticaret Meslek Lisesi öğrenci verileri üzerinde yapılan bir veri madenciliği çalışması ile ilgili bilgiler verilmiştir. Yazılım VB.Net programlama dilinde hazırlanmıştır. Veri tabanındaki veriler üzerinde apriori algoritması uygulanarak her aşaması kullanıcı tarafından gözlemlenmiştir. Elde edilen sonuçlardan öğrencilerin başarısız olduğu dersler arasındaki ilişkiler ortaya çıkarılmıştır.

Temmuz, 2009 **Kerem ÜÇGÜN**

ABSTRACT

THE DESIGN OF STUDENT AUTOMATION FOR MIDDLE EDUCATION SCHOOLS AND DATA MINING APPLICATIONS ON STUDENTS' DATA

In today's world, due to the rapid development of technology, the amount of data stored has been constantly increasing in every field. It is intended to obtain meaningful, valuable information that is not previously known from these data by applying data mining techniques. Association-relation rule is one of these techniques. In specification of the association-relation rules, the most frequently used algorithm is apriori algorithm.

Apriori algorithm is one of the most widely used techniques for exploring the frequent items in the data mining. In order to identify the frequent items, database is scanned many times. During this scanning, inter-related items are found with the help of merging and truncation processes, and minimum support and confidence values of apriori algorithm.

School automation softwares accommodate a great deal of data related to the students and their school status. Despite the abundance and richness of these data, valuable information is hidden inside the database. In order to increase the efficiency of education and to make future plans, useful information is needed. Data mining techniques reveal the significant information about students among those hidden ones and therefore these techniques provide valuable information for education.

In this work, development school automation software and data mining application on student's data are studied. Data on students of I.M.K.B Commercial Vocational High School are used. The software is developed based on VB.Net programming language. Every stage of the process can be observed by user by applying apriori algorithm on the data recorded in database. From the results obtained, the relationship between the courses in which students are unsuccessful are revealed and presented.

July, 2009 **Kerem ÜÇGÜN**

SEMBOLLER

c	: İlgili kuralın minimum güvenirliği
C_k	: Aday öğe kümesi
Ij	: I alanındaki tek bir nesne
k	: Küme sayısı
L_k	: Sık geçen öğe kümesi
s	: İlgili kuralın minimum destek değeri
$X \Rightarrow Y$	: Birliktelik ilişki kuralı
$\Im(I)$	: Ürünler kümesi ailesi
∞	**:** Apriori algoritmasında birleştirme (join) işlemi
$\varnothing$	: Boş küme

KISALTMALAR

ADO : ActiveX Data Objects

DHP : Direct Hashing and Pruning

GB : Gigabyte

Ghz : Gigahertz

HDD : Hard disk drive

Hds : Haftalık ders saati

İ.M.K.B : İstanbul Menkul Kıymetler Borsası

OLE DB : Object Linking and Embedding for Databases

Ram : Random access memory

SQL : Structured Query Language

T.C. : Türkiye Cumhuriyeti

TV : Televizyon

VB.Net : Visual Basic .Net

VM : Veri Madenciliği

VTBK : Veri tabanlarında bilgi keşfi

ŞEKİLLER

SAYFA NO

Şekil II.1 Veri ambarı **6**

Şekil II.2 Fayyad'a göre veri tabanlarında bilgi keşfi (VTBK) sürecinde yer alan adımlar [5]. 7

Şekil II.3 Örnek bir kümeleme modeli **14**

Şekil III.1 Okul otomasyon yazılımının aşamaları **21**

Şekil III.2 Ders_havuzu tablosunun veri yapısı **23**

Şekil III.3 Devam tablosunun veri yapısı **24**

Şekil III.4 Dönemler tablosunun veri yapısı **24**

Şekil III.5 Genel_bilgiler tablosunun veri yapısı **25**

Şekil III.6 Kullanıcı tablosunun veri yapısı **25**

Şekil III.7 Notlar tablosunun veri yapısı **26**

Şekil III.8 Öğrenci_bilgileri tablosunun veri yapısı **27**

Şekil III.9 Öğretmen_bilgileri tablosunun veri yapısı **28**

Şekil III.10 Resmi_adres tablosunun veri yapısı **28**

Şekil III.11 Sınıflar tablosunun veri yapısı **28**

Şekil III.12 Uygulamada kullanılan veri tabanındaki bazı tablolar arasındaki ilişkiler **29**

Şekil III.13 Otomasyon yazılımına giriş ekranı **30**

Şekil III.14 Kimlik doğrulama ekranı **32**

Şekil III.15 Yazılımın ana penceresinin görünümü **33**

Şekil III.16 Temel Bilgiler menüsü altındaki Genel Bilgiler penceresinin görünümü **34**

Şekil III.17 Temel Bilgiler menüsü altındaki Resmi Adres penceresinin görünümü **35**

Şekil III.18 Temel Bilgiler menüsü altındaki Ders Havuzu penceresinin görünümü **36**

Şekil III.19 Temel Bilgiler menüsü altındaki Sınıf Bilgileri penceresinin görünümü **37**

Şekil III.20 Temel Bilgiler menüsü altındaki Toplu Sınıf Bilgileri penceresinin görünümü **38**
Şekil III.21 Temel Bilgiler menüsü altındaki Kullanıcı Bilgileri penceresinin görünümü **39**
Şekil III.22 Temel Bilgiler menüsü altındaki Sınıf Tanımlamaları penceresinin görünümü **39**
Şekil III.23 Öğrenci Bilgileri menüsü altındaki Öğrenci Bilgileri penceresinin görünümü **41**
Şekil III.24 Öğrenci Bilgileri menüsü altındaki Yeni Öğrenci Kaydı penceresinin görünümü **42**
Şekil III.25 Öğrenci Bilgileri menüsü altındaki Toplu Not Girişi penceresinin görünümü **43**
Şekil III.26 Öğrenci Bilgileri menüsü altındaki Öğrenci Bazında Not Girişi penceresinin görünümü **44**
Şekil III.27 Devamsızlık Bilgileri menüsü altındaki Devamsızlık Girişi penceresinin görünümü **45**
Şekil III.28 Devamsızlık Bilgileri menüsü altındaki Sınıf Devamsızlık Listesi penceresinin görünümü **47**
Şekil III.29 Belgeler menüsü altındaki Öğrenci Belgesi, Hasta Sevk Kağıdı, Transkript penceresinin görünümü **48**
Şekil III.30 Öğretmen Bilgileri menüsü altındaki Öğretmen Bilgileri penceresinin görünümü **50**
Şekil III.31 Öğretmen Bilgileri menüsü altındaki Yeni Öğretmen Kaydı penceresinin görünümü **51**
Şekil IV.1 Başarı Analizi bölümünün aşamaları **52**
Şekil IV.2 Klasik Apriori algoritması özet kodu **54**
Şekil IV.3 Apriori-Gen işlevinin algoritma kesiti **54**
Şekil IV.4 Yazılımda kullanılan Apriori algoritmasının aşamaları **56**
Şekil IV.5 Genel Liste ekranından bir kesit **57**
Şekil IV.6 Maden Liste ekranından bir kesit **58**
Şekil IV.7 Ders Listesi ekranından bir kesit **58**
Şekil IV.8 L_1 sık geçen öğe kümesinin oluşumu **60**
Şekil IV.9 L_1 Sık geçen öğe kümesi **61**
Şekil IV.10 C_2 aday öğe kümesinin $L_1 \infty L_1$ birleştirme işlemi ile oluşumu **62**

Şekil IV.11 L_2 sık geçen öğe kümesinin oluşumu **65**
Şekil IV.12 L_2 sık geçen öğe kümesi **66**
Şekil IV.13 C_3 aday öğe kümesinin $L_2 \infty L_2$ dönüşümü ile oluşumu **67**
Şekil IV.14 C_3 aday öğe kümesinde budama işlemi **71**
Şekil IV.15 L_3 sık geçen öğe kümesinin oluşumu **74**
Şekil IV.16 L_3 sık geçen öğe kümesi **75**
Şekil IV.17 C_4 aday öğe kümesinin oluşumu **76**
Şekil IV.18 C_4 aday öğe kümesinde budama işlemi **77**
Şekil IV.19 Algoritma sonunda oluşan birliktelik kuralları **78**
Şekil IV.20 Birliktelik kuralları **78**
Şekil IV.21 Genel Liste ekranı **79**
Şekil IV.22 Maden Liste ekranı **80**
Şekil IV.23 Ders Listesi ekranı **80**
Şekil IV.24 C_1 aday öğe kümesi, destek ve güven değerlerini giriş ekranı **81**
Şekil IV.25 L_1 sık geçen öğe kümesi ekranı **82**
Şekil IV.26 C_2 aday öğe kümesi ekranı **82**
Şekil IV.27 L_2 sık geçen öğe kümesi ekranı **83**
Şekil IV.28 C_3 aday öğe kümesi ekranı **84**
Şekil IV.29 C_3 aday öğe kümesi (budama sonrası) ekranı **85**
Şekil IV.30 L_3 sık geçen öğe kümesi ekranı **85**
Şekil IV.31 C_4 aday öğe kümesi ekranı **86**
Şekil IV.32 C_4 aday öğe kümesi (budama sonrası) ekranı **87**
Şekil IV.33 Birliktelik kuralları ekranı **87**

TABLOLAR

SAYFA NO

Tablo II.1 Ürün hareket tablosu.....**18**

Tablo IV.1 Algoritmada kullanılan değişkenler.....**53**

Tablo IV.2 Kurallar.....**88**

BÖLÜM I

GİRİŞ ve AMAÇ

Bu bölümde veri madenciliğinin genel özellikleri ve önemi üzerinde durulmuş, veritabanları üzerinde veri madenciliği uygulamalarının gereksinim ve nedenleri ortaya konularak çalışmanın amacı açık bir şekilde ifade edilmiştir.

I.1. GİRİŞ

Veriler hacim olarak sayfalarca yer kaplarlar ama kullanım değerleri azdır. Oysa sayıları düzenleyip özetlersek, harfleri düzenleyerek anlamlı cümleler haline dönüştürürsek, notaları sıralayıp bir melodi oluşturursak ve bilgisayar ekranındaki noktaları (verileri) bir araya getirerek bir ağaç resmi veya bir grafik oluşturursak ancak o zaman verileri bilgiye dönüştürmüş oluruz. Bilgi verilere göre hacim olarak daha az yer tutar ama kullanım değeri olarak daha güçlüdür [1].

Veri madenciliği, (data mining) son 10 yılda dünyada hızla yaygınlaşmaya başlayan bir disiplinler arası disiplin olarak göze çarpmaktadır [2]. Temel olarak veri madenciliği, geniş veri setleri arasındaki desenlerin ya da düzenin, verinin analizi ve yazılım tekniklerinin kullanılması ile ilgilidir. Veriler arasındaki ilişkiyi, kuralları ve özellikleri belirlemekten bilgisayar sorumludur. Amaç, daha önceden fark edilmemiş veri desenlerini tespit edebilmektir.

Veri madenciliği (VM), eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır [3]. Başka bir deyişle, VM, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir [3]. Bu noktada kendi başına bir çözüm değil çözüme ulaşmak için verilecek karar sürecini destekleyen, problemi çözmek için gerekli olan bilgileri sağlamaya yarayan bir araçtır.

Veri madenciliği için yapılan tanımlamalardan bazıları da şunlardır:

- Veri madenciliği büyük miktardaki veriler içerisinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır [6].
- Veri madenciliği, çeşitli veri analiz araçlarının kullanılmasıyla veriler üzerinde gelecek ile ilgili geçerli tahminler yapmamız için kullanılacak örüntü ve ilişkilerin keşfedilmesi süreci olarak da Two Crows Corparation (2006) tarafından tanımlanmıştır.
- Önceden bilinmeyen ve potansiyel olarak faydalı olabilecek, veri içinde gizli bilgilerin çıkarılmasına veri madenciliği denir [7].
- Hand'e göre veri madenciliği, veri tabanı teknolojisi, istatistik, makine öğrenim, örüntü tanımı, yapay sinir ağları, verilerin görselleştirilmesi ve uzaysal veri analizi gibi farklı disiplinlerde yer alan tekniklerin bir bileşimini içerir [8].
- Holsheimer' e göre veri madenciliği, büyük veri kümesi içinde saklı olan genel örüntülerin bulunmasıdır [9].

I.2. TEZ ÇALIŞMASININ AMACI

Bu tez çalışmasının hazırlanma amacı;

- Okul otomasyon yazılımı tasarlayarak okul idarecilerin öğrenci işlerini daha rahat ve hızlı bir şekilde sürdürmesi,
- Veri madenciliği ve veritabanlarında bilgi keşfi süreci içinde yer alan temel kavramların, yöntem ve tekniklerin araştırılması,
- Veri madenciliği tekniklerinden birliktelik-ilişki yönteminin araştırılması ve bu yöntemin en çok kullanılan apriori algoritmasının örnek bir okulun öğrenci verileri üzerine uygulanması,
- Veri madenciliği uygulaması sonrasında öğrenimin düzenlenmesine yön verecek sonuçlara ulaşılması

olarak belirlenmiştir.

BÖLÜM II

VERİ MADENCİLİĞİ KAVRAMINA GENEL BAKIŞ

II.1. GENEL BAKIŞ

Veri madenciliği, kavramsal olarak 1960'lı yıllarda, bilgisayarların veri analiz problemlerini çözmek için kullanılmaya başlamasıyla ortaya çıkmıştır. O dönemlerde, bilgisayar yardımıyla, yeterince uzun bir tarama yapıldığında, istenilen verilere ulaşmanın mümkün olabileceği gerçeği ortaya çıkmıştır. Bu işleme veri madenciliği yerine önceleri veri taraması, veri yakalanması gibi isimler verilmiştir.

1990'lı yıllara gelindiğinde veri madenciliği ismi, bilgisayar mühendisleri tarafından ortaya atılmıştır. Amaç, geleneksel istatistiksel yöntemler yerine, veri analizinin algoritmik bilgisayar modülleri tarafından değerlendirmesini vurgulamaktı. Bu noktadan sonra bilim adamları, veri madenciliğine çeşitli yaklaşımlar getirmeye başlamışlardır. Bu yaklaşımların kökeninde istatistik, makine öğrenimi, veritabanları, otomasyon, pazarlama, araştırma gibi disiplinler ve kavramlar yatmaktaydı.

Günümüzde veri madenciliği birçok alanda kullanılmaktadır. Operasyonel kararların ötesinde, stratejik ve politik karar verme süreçlerinde önemli bir yere sahiptir. Gerek özel sektör gerekse kamusal sektör, çeşitli uygulamalar ve teknikler vasıtasıyla veri madenciliği yapmaktadır. İstatistik ile olan yakın ilişkisi veri madenciliğini tıp ve ekonomi gibi bilim dalları için de önemli kılmaktadır. Bilginin bu kadar değerli olduğu çağımızda, bilgiye ulaşmak için kat edilen yolda veri madenciliği oldukça önemli bir safhadır. Zira bu kadar yoğun bir veri akışının altında bilgiyi keşfetmek yalnızca bilimsel bir çaba değil aynı zamanda bir sanat halini almıştır.

Bu tez çalışması kapsamında geniş bir literatür taraması yapılmış, önceden yapılmış benzer çalışmalar incelenmiş ve konu ile ilgili geniş çaplı bilgi edinilmiştir. Tez kapsamında üzerinde yoğun olarak durulan birliktelik-ilişki kuralları, genelde mağazalarda müşteri ilişkilerinin belirlenmesinde kullanılmaktadır. Bu bağlamda

yapılan çalışmalar dikkate alınarak eksik kalmış yönler tespit edilip eğitim konusunda yapılan bu tez çalışmasına uyarlanmıştır. Ayrıca daha önceden veri madencili ile ilgili eğitim alanında yapılan çalışmalar incelenerek çalışmaların eksik kalmış ve geliştirilmeye açık noktaları üzerinde araştırma yapılmıştır.

II.1.1. Veri Madenciliği İle İlgili Önceden Yapılmış Çalışmalar

II.1.1.1. Öğrenci Bilgi Sisteminde Veri Madenciliği Uygulaması

Yapılan bu çalışmada, bir üniversitesinde okumakta olan 2002-2003 yılı girişli tüm öğrencilerin 1. ve 2. sınıfta almış oldukları notlar değerlendirilmiş ve öğrencilerin başarısız oldukları dersler değerlendirilerek bu dersler arasındaki birliktelik kuralları ortaya çıkarılmıştır.

II.1.1.2. Apriori Algoritması İle Öğrenci Başarı Analizi

Bu çalışmada veri madenciliği teknikleri kullanılarak Fırat Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Eğitimi bölümü öğrencilerinin notları kullanılarak öğrenci başarılarının analizi yapılmıştır. Bu analizi yapmak için Veri Madenciliğinde, birliktelik kuralı çıkarım algoritmalarından biri olan Apriori algoritması kullanılmıştır. Çalışmada kullanılan öğrenci notları, Fırat Üniversitesi öğrenci işleri biriminden SQL server veri tabanı ortamında alınarak gerekli dönüşümler yapılmış ve Apriori algoritması uygulanabilecek boolean tipe çevrilmiştir. Bu aşamadan sonra Matlab ortamında bir program hazırlanarak veritabanı programa giriş olarak verilmiş ve bilgisayar ortamından otomatik olarak kurallar üretilmiştir.

II.1.1.3 Zeki Öğretim Sistemlerinde Veri Madenciliği Kullanılması

Yapılan bir tez çalışmasında veri madenciliği yöntemlerinin, zeki öğretim sistemlerindeki öğrenci verisi üzerinde kullanılması ile bu sistemlerdeki verilerin analiz edilmesi sağlanmıştır. Zeki öğretim sisteminden alınan verilerin analiz edilmesi için ise, veri madenciliği tekniklerinden "birliktelik kuralları" ve "kümeleme analizi" kullanılmıştır. Zeki öğretim sisteminden alınan verilerde, aynı konu içerisinde birlikte yanlış yapılan veya doğru yapılan sorular tek boyutlu birliktelik kuralı analizi ile bulunmaktadır. Ayrıca farklı konulardaki sorular arasında, birlikte yanlış veya doğru yapılan soru dizileri çok boyutlu birliktelik kuralı

analizi ile bulunmaktadır. Öğrencilerin zeki öğretim sisteminde konulardan aldıkları başarı puanlarına göre kümeleme analizi gerçekleştirilmektedir. Yapılan bu çalışma; zeki öğretim sisteminden alınan verilerde; veri madenciliği tekniklerinden, tek ve çok boyutlu birliktelik kuralı analizi, kümeleme analizinin yapılmasını ve kümeleme analizi sonuçlarının görselleştirilmesini sağlayarak bu alana katkı sağlamıştır.

II.1.1.4. Öğrenci Seçme Sınavında Öğrenci Başarımını Etkileyen Faktörlerin Veri Madenciliği Yöntemleriyle Tespiti

Yapılan bu çalışmada öğrenci seçme ve yerleştirme merkezinin resmi internet sitesinde 2008 yılında uygulanmış olan öğrenci seçme sınavı öğrenci anketinden elde edilen veriler üzerinde sınıflandırma ve kümeleme yöntemleri uygulanmış ve öğrencilerin öğrenci seçme sınavında göstermiş olduğu başarıyı etkileyen faktörler tespit edilmiştir.

II.1.1.5. Üniversite Öğrencilerinin İnterneti Eğitimsel Amaçlar İçin Kullanmalarını Etkileyen Faktörlerin Veri Madenciliği Yöntemleriyle Tespiti

Bu çalışmada üniversitede okumakta olan lisans öğrencilerinin eğitim amacıyla internet kullanımlarının içerdiği gizli veya açık örüntüleri, veri madenciliği yöntemleriyle tespit edilmiştir. Öğrencilerin eğitimsel amaçlar için internet kullanımlarına yönelik davranış ve düşüncelerinde sık görülen örüntüler bulunmuştur.

II.1.1.6. Web Tabanlı Uzaktan Eğitimde İki Farklı Öğretim Modelinin Öğrenci Başarısı Üzerindeki Etkilerinin İncelenmesi

Bu çalışmada video destekli ve animasyon destekli öğretim modeline uygun hazırlanan iki farklı öğretim materyalinin öğrenci başarısı üzerindeki etkileri veri madenciliği teknikleri kullanılarak analiz edilmiştir.

II.2. VERİ TABANI VE VERİ AMBARI KAVRAMLARI

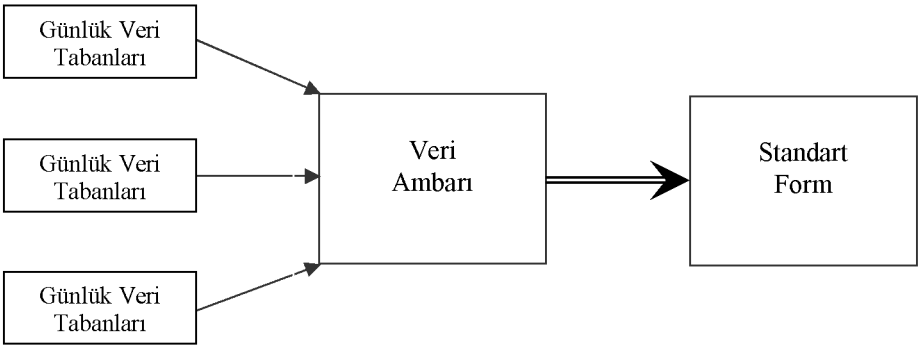
II.2.1. Veri Tabanı

Veritabanı birbiri ile ilişkili verilerin tekrara yer vermeden çok amaçlı kullanımına imkan verecek şekilde depolanması olarak tanımlanabilir. Veritabanları ayrıca verilerimizi güvenli ve hızlı bir biçimde işlememize yardımcı olurlar.

Veritabanı bir depo görevi görmektedir fakat yanlış kullanıldığında ise çöp yığınına dönüşmektedir. Veritabanının doğru kullanımı, doğru ilişkilendirilmesi çok önemlidir. Bunun en başında hız, zaman, sorgu sayısı gibi kazançlarımız olacaktır.

II.2.2. Veri Ambarı

Veri ambarı (data warehouses) birden çok veri tabanı tarafından toplanan bilgilerin, gelecekte değerlendirilmek üzere arka planda üst üste yığılarak birleştirilmesiyle oluşan veri deposudur. Veri ambarı sayesinde verinin analizi ve ileriye yönelik tahmin ve raporlamalar yapılır. Veri ambarları sadece okuma amaçlı kullanılırlar.



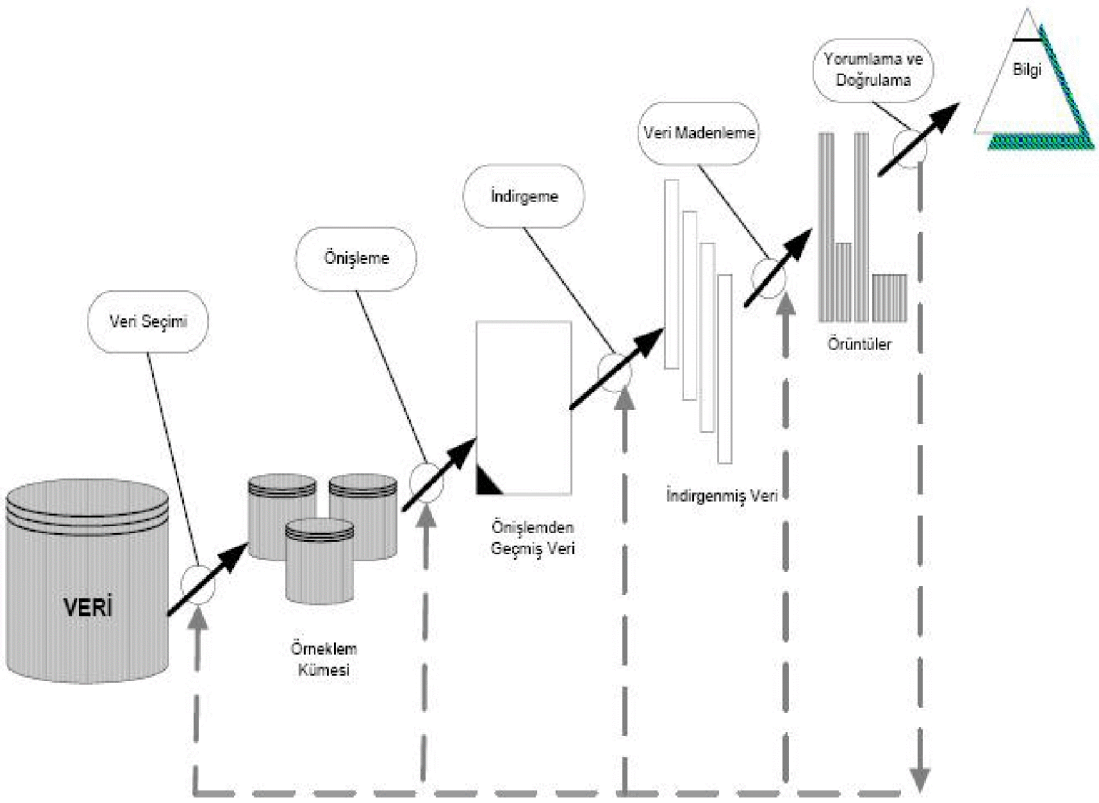
Şekil II.1 Veri ambarı

Günlük veri tabanlarından istenen özet bilgi seçilerek, gerekli ön işlemeden sonra veri ambarında saklanır. Ardından amaç doğrultusunda gerekli veri, ambardan alınarak veri madenciliği çalışması için standart bir forma çevrilir [6].

II.3. VERİ TABANLARINDA BİLGİ KEŞFİ SÜRECİNDEKİ ADIMLAR

Veri madenciliği sürecinin ilk adımı, uygulama alanının öğrenilmesi ile başlar. Sonra gürültülü, hatalı ve tutarsız verilerin temizlendiği veri temizleme ve ön işleme aşaması gelir. Sonra örnek veri kümesindeki ilgisiz nitelikte ve tekrarlı verilerin çıkarıldığı indirgeme aşaması gelir. Bir sonraki adım ise veri madenciliği yöntemlerinin ve algoritmalarının uygulandığı adımdır. Daha sonra, veri madenciliği adımında ortaya çıkarılan örüntüler, içlerinden yararlı olanların belirlenmesi için değerlendirilir. Veritabanlarında bilgi keşfinin son aşaması ise, elde edilen bilginin görüntüleme ve bilgi gösterimi yöntemleri kullanılarak kullanıcıya sunulması

aşamasıdır. Şekil II.2'de veri tabanlarında bilgi keşfi sürecinde yer alan adımlar görülmektedir.



Şekil II.2 Fayyad'a göre veri tabanlarında bilgi keşfi (VTBK) sürecinde yer alan adımlar [5].

Fayyad'a göre ise veri tabanlarında bilgi keşfi (VTBK) sürecindeki aşamalar özetle şu şekilde sıralanmaktadır [5].

- **Veri Seçimi (Data Selection):** Bu aşama birden fazla veri kümesi içerisinden, üzerinde sorgu yapılmasına uygun örnek bir veri kümesi oluşturma aşamasıdır. Veri toplama (data collection) ve farklı kümelerdeki verilerin birleştirilmesi işlemi de bu süreçte yer alır.
- **Veri Önişleme (Data Preprocessing):** Veri seçimi ile elde edilen örnek veri kümesinde yer alan hatalı ve eksik değerlerin düzenlendiği ve çıkarıldığı aşamadır. Veri temizleme (data cleaning) ve veri dönüştürme (data transformation) veri önişleme işlemleridir [5].
- **Veri İndirgeme (Data Reduction):** Seçilen örnek veri kümesindeki ilgisiz nitelikte ve tekrarlı verilerin çıkarıldığı aşamadır. Bu işlem ile verinin boyutu indirgendiğinden veri madenciliği uygulanırken çalıştırılacak sorguların daha hızlı sonuç üretmeleri sağlanır [5].

- **Veri Madenciliği (Data Mining):** Bu aşama veri madenciliği yöntemlerinin ve algoritmalarının uygulandığı adımdır [5].
- **Değerlendirme (Evaluation):** Bilgi keşfi sürecinde bu aşamadan önceki aşamalar sonucunda elde edilen bilginin geçerlilik, yenilik, yararlılık ve basitlik kıstaslarına göre değerlendirilmesi aşamasıdır [5].

II.4. VERİ MADENCİLİĞİNİN BAZI UYGULAMA ALANLARI

Veri madenciliği verinin yoğun olarak üretildiği ve bunun sonucu olarak veritabanlarının oluştuğu hemen her ortamda uygulama kullanım alanı bulabilir. Bazı uygulama alanları aşağıdaki gibi özetlenebilir;

- **Pazarlama-Perakendecilik:** Müşterilerin satın alma örüntülerine göre bir müşteri portföyü belirlemek, mevcut müşterileri kaybetmeden yeni müşteriler kazanmak, pazar sepeti analizi ile müşterilerin aldığı ürünler arasında bağıntı kurmak, satış tahmini ve satış noktası veri analizinde, mağaza yerleşim optimizasyonunda kullanılmaktadır.
- **Bankacılık, Sigortacılık, Borsa:** Kredi kartı dolandırıcılıklarının önlenmesi, kredi başvurularının puanlanması, kredi kartı harcamalarına göre müşteri gruplarının belirlenmesinde, risk analizi ve risk yönetiminde, hisse senedi fiyat tahmininde, alım-satım stratejilerinin optimizasyonunda kullanılmaktadır.
- **Telekomünikasyon:** Kalite ve iyileştirme analizlerinde, hatların yoğunluk tahminlerinde kullanılmaktadır.
- **Sağlık ve İlaç:** Hastalıkların teşhis edilmeleri sürecinde, tedavi yönteminin belirlenmesi, ilaç üretiminin ve geliştirilmesinin sağlanması amacı ile sağlık sektöründe kullanılmaktadır.
- **Endüstri ve Mühendislik:** Ürün üretimim iyileştirilmesi ve bu ürünlerin kalite kontrol analizlerinin yapılması sürecinde, bilimsel ve teorik problemleri çözmek için kullanılmaktadır.
- **Biyoloji ve Genetik:** Bitki türlerinin belirlenmesi ve ıslahında, genetik hastalıkların tespiti ve insan geni haritasının çıkarılmasında kullanılmaktadır.
- **Meteoroloji:** Hava durumu tahminlerinin yapılması, ekolojik dengenin gözetilmesi amacı ile kullanılmaktadır.

- **Web Madenciliği:** İnternet üzerindeki verilerin sınıflandırılarak, en kısa sürede ulaşabilmemizi sağlamak için web madenciliği kullanılmaktadır.

II.5. VERİ MADENCİLİĞİNDE KARŞILAŞILAN PROBLEMLER

Veri madenciliği sistemleri işlenmemiş veriyi tutmak için dinamik, bitirilmemiş, gürültü içeren ve büyük olma eğiliminde olan veritabanlarına bel bağlamaktadır. Problemler saklanan bilginin yeterli ve konuyla ilgili olup olmaması sonucunda ortaya çıkmaktadır. Yani, veri madenciliğine girdi olarak kullanılan veritabanlarının dinamik, eksiksiz, geniş ve net veri içermemesi ve verinin konu ile uyumsuzluğu durumunda sorunlar ortaya çıkmaktadır. Sınıflandırmak gerekirse başlıca sorunlar şunlardır [10]:

- **Sınırlı Bilgi:** Veritabanları genel olarak veri madenciliği dışındaki amaçlar için tasarlanmışlardır. Bu yüzden, öğrenme görevini kolaylaştıracak bazı özellikler bulunmayabilir. Yetersiz veri problemlere sebep olmaktadır çünkü bazı veriler geçerli etki alanında sunulamaz.
- **Gürültü ve Eksik Değerler:** Veri özellikleri ya da sınıflarındaki hatalara gürültü adı verilir. Veritabanlarındaki eksik bilgi ve bu yanlışlardan dolayı veri madenciliği amacına tam olarak ulaşmayabilir.
- **Kayıp Veri:** Veri madenciliği sistemleri içinde; basit, önemsenmeyen, kayıp değerler ve uygun kayıtların ihmal edilmesi istenebilir.
- **Belirsizlik:** Hatanın boyutu ve verideki gürültünün derecesi ile ilgilidir. Veri tahmini bir veri madenciliği sisteminde önemli bir husustur.
- **Boyut, Güncellemeler ve Konu Dışı Veriler:** Veritabanları büyük ve dinamik olma eğilimindedir, içerikleri her zaman ekleme, düzeltme ve silme işlemleri ile değişim halindedir. Veri madenciliği perspektifinden bakıldığında, kuralların hala aynı kalıp kalmadığı ve istikrarlılığı problemi ortaya çıkar.

II.6. VERİ SEÇİMİ, ÖN İŞLEME VE İNDİRGEME

II.6.1. Veri Seçimi

Veri seçimi (veri toplama) işlemi bir seçme işlemi olmasına rağmen, toplanan verinin tek bir ortamda birleştirilmesinden dolayı veri bütünleştirme de bu aşamanın bir sürecidir [11].

Veri seçimi aşamasında yapılması gerekenler:

- Farklı ortamdaki verilerin mevcut yapılarının incelenmesi ve tablo yapılarının ortaya çıkarılması,
- Veri madenciliği çalışmasında istenilen sonuca ulaşmak için gerekli verilerin, veri madenciliği uygulamak için belirlenen veri depolama ortamına, veri transfer teknikleri kullanılarak transfer edilmesidir.

II.6.2. Veri Önişleme

Veri ön işleme, veri madenciliği için önemli bir konudur. Veri madenciliğinde güvenilirliğin artırılması için, veri ön işleme yapılmalıdır. Aksi halde hatalı girdi verileri bizi hatalı çıktıya götürecektir.

Neden veri ön işlemeye ihtiyaç duyulduğu kısaca aşağıdaki gibi sıralanabilir [11]:

- Eksik, tutarsız ve gürültülü veriler olduğu için.
- Veriler üzerinde herhangi bir analiz türünün uygulanmasını engelleyecek veri problemlerinin çözümü için.
- Verilerin doğasının anlaşılması ve anlamlı veri analizinin başarılması için.
- Verilen bir veri kümesinden daha anlamlı bilginin çıkarılması için.

Veri temizleme ve veri dönüşümü işlemleri veri ön işleme aşamasını oluşturmaktadır. Veri temizlemenin amacı, aykırı değerlerin tespit edilmesiyle gürültünün düzeltilmesi ve verilerdeki tutarsızlığın giderilmesidir. Veri dönüşümünün amacı ise, kaynak veri içindeki farklı biçimdeki veri tip ve değerlerini veri madenciliği için uygun formlara dönüştürmektir. Veri ön işlemede uygulanan çeşitli teknikler vardır:

II.6.2.1. Veri Tipi Dönüşümü

Veri tipi dönüşümü basit olarak veri tipi değişimidir. Örnek olarak boolean tipteki bir veriyi integer tipine dönüştürme işlemi verilebilir. Bu dönüşümün sonucunda, bazı veri tabanı algoritmalarında boolean veri yerine integer veri kullanılması daha iyi verim alınmasını sağlayabilir [11].

II.6.2.2 Gruplama

Bazen bir kolon içinde oldukça fazla oranda tekrarlayan benzer veriler olur. Bu verileri birkaç grup içine indirgeme işlemi uygulanarak verinin kalitesi artırılır. Gruplama tekniği ile yorumlamanın daha kolay olması sağlanabilir [11].

II.6.2.3. Kayıp Değerleri Yönetmek

Birden fazla veri tabanından gelen veriler tek bir tablo içinde birleştirildiğinde bazı veri alanlarının veri olmayabilir. Operasyondan gelen verinin ilgili veri alanı boş bırakıldığından veya farklı veri tabanından gelen veride bu tip bir veri alanı bulunmadığından bazı kayıtlarda bu alan boş kalabilir. Bu durumu düzeltmek için,

- Kayıp değerler en çok kullanılan değerler ile doldurulabilir,
- Bir kayıtta çok oranda kayıp değer varsa kayıtın tamamı silinebilir,
- Kayıp değerlerin içi el ile doldurulabilir,
- En olası ortalama değer ile doldurulabilir [11]

II.6.2.4. Gürültülü Veri

Gürültülü veri, veri kümesi içinde yer alan ama bir anlamı olmayan, veri madenciliği uygulamasında kullanılmayacak veridir. Tipik olarak verilerde ne kadar çok oranda gürültü varsa, o derece de güvenilir sonuçlara ulaşmak güçleşecektir. Gürültülü veriden kurtulmak için veri temizleme işlemi uygulanır. Gürültülü verilerin teşhis edilmesi amacıyla histogram, kümeleme analizi ve regresyon gibi teknikler kullanılabilir [2].

Uygunsuz değerler kümeler ile belirlenebilir. Benzer değerler aynı grup veya küme içinde yer alırken, uygunsuz değerler kümelerin dışında yer alacaktır. Kümelerin dışında kalan bu değerler, veri madenciliği uygulanacak veri bütünlüğünden çıkarılır [11].

II.6.3. Veri İndirgeme

Veri indirgeme, daha küçük hacimli olarak ve veri kümesinin indirgenmiş bir örneğinin elde edilmesi amacıyla uygulanır. Veri indirgeme yöntemi ile çok büyük miktardaki verinin analiz edilmesi uzun süreceğinden, veri bütünlüğünü bozmadan verinin tamamını temsil edecek az miktarda veri alınır ve analiz edilir. Bu sayede elde edilen indirgenmiş veri kümesine veri madenciliği teknikleri uygulanarak daha etkin sonuçlar elde edilebilir.

Veri indirgeme yöntemleri aşağıdaki biçimde özetlenebilir:

- Veri birleştirme veya veri küpü (Data aggregation or data cube)
- Boyut indirgeme (Dimension reduction)
- Veri sıkıştırma (Data compression)
- Kesikli hale getirme (Discretization)

II.7. VERİ MADENCİLİĞİNDE KULLANILAN TEKNİKLER

Veri madenciliği sürecinde ilk yapılması gereken şey, bilgi ihtiyacının belirlenmesi ve problemin tanımlanmasıdır. Bu aşamadan sonra veri kaynaklarının belirlenmesi ve verilerin incelenmesi gerekir. Veriler hazırlanır. Bundan sonraki aşama modelin oluşturulmasıdır. İşlemin en önemli adımı sayılabilecek bu aşama gerekli bilgiye ulaşmaya giden yoldur. Bundan sonrası modelin değerlendirilmesi ve sonuçların yorumlanması adımıdır [12].

Veri madenciliğinde kullanılan modeller, tanımlayıcı (Descriptive) ve tahmin edici (Predictive) olarak iki ana başlık altında incelenmektedir. Tanımlayıcı modellerde, karar vermeye yardımcı olarak kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır. X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile çocuğu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir [13]. Tahmin edici modellerde, sonuçları bilinen verilerden hareketle bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin bir banka, önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde, bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır [12].

Gerek tanımlayıcı gerekse tahmin edici modellerde yoğun olarak; sınıflama (classification) ve regresyon (regression), kümeleme (clustering), birliktelik kuralları (association rules), ardışıl örüntü (sequential pattern) yöntemleri kullanılmaktadır. Sınıflama ve regresyon tahmin edici, kümeleme, birliktelik kuralları ve ardışıl örüntü tanımlayıcı modellerden bazılarıdır.

II.7.1. Sınıflama (Classification) ve Regresyon (Regression)

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen veri analiz yöntemleridir. Sınıflama; belli özelliklere göre gruplanmış değerleri tahmin ederken, regresyon; mevcut değerleri kullanarak diğer değerlerin ne olacağını tahmin etmeye çalışır.

Sınıflandırma ile ilgili bir uygulama kısaca şöyledir:

"Genç kadınlar küçük araba satın alır, yaşlı, zengin erkekler büyük, lüks araba satın alır." Amaç bir malın özellikleri ile müşteri özelliklerini eşlemektir. Böylece bir müşteri için ideal ürün veya bir ürün için ideal müşteri profili çıkarılabilir. Örneğin bir otomobil satıcısı şirket, geçmiş müşteri hareketlerinin analizi ile yukarıdaki gibi iki kural bulursa genç kadınların okuduğu bir dergiye reklam verirken küçük modelinin reklamını verir [6].

Örneğin sınıflandırma modeli banka kredi uygulamalarının güvenli veya riskli olmalarını sınıflamak amacıyla kullanılırken, regresyon modeli geliri ve mesleği bilinen potansiyel müşterilerin herhangi bir alışveriş esnasında yapacakları harcama davranışlarını tahmin etmek için kullanılabilir [11].

Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler şunlardır [13]:

- Genetik algoritmalar (Genetic algorithms),
- K-en yakın komşu (K-nearest neighbor),
- Naive-Bayes,
- Karar ağaçları (Decision trees)
- Yapay sinir ağları (Artificial neural networks)
- Bellek temelli nedenleme (Memory based reasoniing)

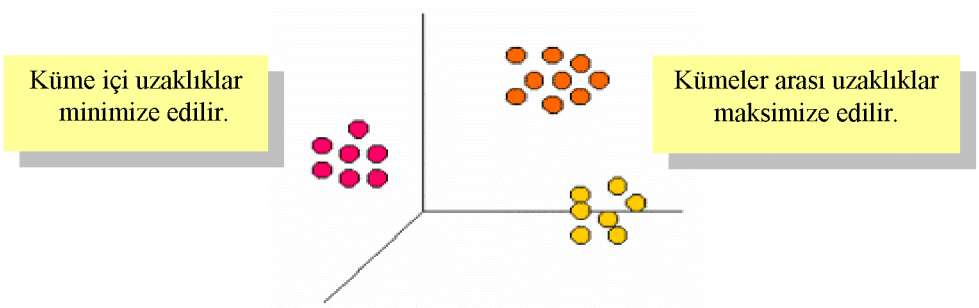
II.7.2. Kümeleme

Kümeleme, birbirine benzer özellikler gösteren veriyi sınıflara veya kümelere ayırma işlemidir. Kümeleme işlemi, nesneleri aralarındaki benzerliklere göre gruplara ayırır. Böylece benzer nesneler aynı grup içinde toplanırken farklı nesneler farklı gruplarda yer alır. Benzerlik nesneleri tanımlayan özelliklerdir.

Sınıflama işleminin kümeleme işleminden farkı şudur: Sınıflama işleminde sınıfları belli olan verilere bakılarak yeni gelecek olan verinin sınıfları tahmin edilir. Kümeleme işleminde ise kümesi belli olan veri yoktur. Veriler gruplar halinde kümelere ayrılır. Küme içindeki öğelerin benzerliği olmalı, kümeler arasında ise benzerlik az olmalıdır.

Pek çok kümeleme algoritması bulunmaktadır. Başlıca kümeleme algoritmaları şöyle sıralanabilir [14].

- Bölme yöntemleri
- Hiyerarşik yöntemler
- Yoğunluk tabanlı yöntemler
- Izgara tabanlı yöntemler
- Model tabanlı yöntemler



Şekil II.3 Örnek bir kümeleme modeli

II.7.3. Ardışıl Örüntü

Ardışıl örüntü, birbirleri ile ilişkisi olan ve birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır. Belli bir olayın gerçekleşmesinden sonra aynı olayla ilintili bir başka olayın gerçekleşmesi, ardışıl örüntü kavramını oluşturur. Örneğin; otomobil alan bir kişinin kısa vadede otomobili için aksesuar alması ardışıl bir örüntüdür [11].

II.7.4. Birliktelik Kuralları

Birliktelik kuralları, veri kümesindeki nesneler arasındaki ilişkiyi ortaya çıkarmak olarak tanımlanabilir. Birliktelik kuralları, veri kümesi içindeki hareketlerin (transactions) analiz edilerek bu hareketler ya da kayıtlar arasında sıklıkla bir arada görülenlerin tespit edilmesi işlemidir. Birliktelik kuralları ile bir ilişkide yer alan niteliklerin değerleri arasındaki bağımlılıklar, anahtarda yer almayan diğer niteliklerin gruplandırılması ile bulunur. Bu kurallar ilk olarak Agrawal (1994) tarafından bulunmuştur.

Birliktelik kurallarının analizi süreci market sepet analizi olarak da adlandırılır. Market sepet analizinde müşteri ile ilgili veri hareketlerinden, gelecekte müşterinin nasıl bir tercih yapacağına dair sonuçlar tahmin edilir. Bununla birlikte bu teknikler,

tıp, finans ve farklı olayların birbirleri ile ilişkili olduğunun belirlenmesi sonucunda değerli bilgi kazanımının söz konusu olduğu ortamlarda da önem taşımaktadır.

Birliktelik kuralı, bir ilişkide bir niteliğin aldığı değerler arasındaki bağımlılıkları, anahtarda yer almayan diğer niteliklere göre gruplama yapılmış verileri kullanarak bulur. Keşfedilen örüntüler veri kümesinde sıklıkla birlikte geçen nitelik değerleri arasındaki ilişkiyi gösterir [4]. Örneğin bir süper markette ekmek ve peynir satın alınan satış hareketlerinin % 75 'inde zeytin de satın alınmıştır. Bu tür birliktelik-ilişki örüntüleri ancak, örüntüde yere alan öğelerin birden fazla harekette tekrarlandığında potansiyel olarak mevcut olabilirler [4].

II.7.4.1. Birliktelik Kuralları Temel Kavramları

Birliktelik kuralının matematiksel modeli 1993 yılında Agrawal, Imielinski ve Swami tarafından ifade edilmiştir [15]. Bu modelde $I=\{ i_1,i_2,..,i_m \}$verilen bağlamda yer alan ürün kodlarını; D bağlamda yer alan hareketleri; T bir harekette yer alan ürün kodlarını ($T \subseteq I$) (bir harekette yer alan her ürün miktar bilgisi ile değil, o harekette satın alınıp alınmadığını gösteren (alındı/alınmadı) mantıksal bir değişkenle önemlidir), tid her harekete ait tekil (unique) numarayı, k-öğe kümesi k adet ürün içeren ürün kümesini temsil etmektedir. X bir ürün kümesi olmak üzere T hareketi X ürün kümesini ancak ve ancak $X \subseteq T$ şartını sağlıyor ise içerir. Bir birliktelik-ilişki kuralı X ve Y, I ürün kümesinin bir alt kümesi ve $X \cap Y= \varnothing$ olmak üzere $X \Rightarrow Y$ biçiminde bir bağımlılık ifadesidir. Bu ifade ile X, Y' yi belirler (X ürününü içere hareketler kümesi Y'yi içeren hareketler kümesi içinde kapsanır) veya Y, X' e bağımlıdır denir (Y kümesinin varlığı X kümesinin var olmasına bağlıdır). Hareket numaraları gruplandırılarak elde edilen ürünler arasındaki bağımlılık ilişkisinin yüzde yüz doğru olması beklenemez. Benzer şekilde, çıkarsama yapılan kuralın eldeki hareketler kümesinin önemli bir kısmı tarafından desteklenmesi istenir. Bu nedenlerden dolayı, $X \Rightarrow Y$ eşleştirme kuralı kullanıcı tarafından minimum değeri belirlenmiş güvenilirlik (c:confidence) ve destek (s:support) eşik değerlerini sağlayacak biçimde üretilir. $X \Rightarrow Y$ eşleştirme kuralına, c güvenirlik ölçüsü ve s destek ölçütü iliştirilir ve biçimsel olarak $\phi(D)=<X \Rightarrow Y,c,s>$ ile gösterilir. Burada D örneklemi; $X \Rightarrow Y$ birliktelik-ilişki kuralını; c eşik değeri, ilgili kuralın minimum güvenirliğini (X ürünlerini içeren hareketlerin en az %c oranında Y içeren hareketler kümesinde yer aldığını); s ilgili kuralın, minimum desteğini (X ve Y

ürünlerini içeren hareket tutanaklarının toplam hareket tutanakları içinde en az %s oranında var olduğunu) gösterir [4,11].

Ürünler kümesi ailesini $\mathfrak{I}(I)$ gösterelim ve X ve Y' nin her ikisi de $\mathfrak{I}(I)$ üzerinde değişebilen iki rasgele değişken olsun. Pr(X), Pr(X$\cap$Y), ve Pr(X$\cup$Y) marjinal olasılık değerleri, sırasıyla, X kümesi içinde yer alan tüm ürünlerin herhangi bir sepet varlığında bulunma olasılığını, X ve Y rasgele değişkenlerince paylaşılan ortak ürünlerin herhangi bir sepet varlığında bulunma olasılığını, ve X ve Y rasgele değişkenlerinin birleşiminde yer alan ürünlerin herhangi bir sepet varlığında bulunma olasılığını göstersin. O zaman, güvenilirlik eşiği, Pr(Y/X)=Pr(X$\cap$Y)/Pr(X) ile ve destek eşiği Pr(X$\cup$Y) ile ifade edilir. Güvenilirlik metriği, birliktelik-ilişki kuralının doğruluk derecesini, destek metriği ise kuralda yer alan öğelerin (ürünlerin) geçiş sıklığını gösterir. Yüksek güvenirlik ve destek değerine sahip kurallara güçlü kurallar adı verilir. Birliktelik-ilişki kuralı çıkarımı büyük veritabanlarından güçlü ilişkilendirme (eşleştirme) örüntülerinin elde edilmesini gerektirir [11].

Birliktelik-ilişki kuralı formülsel olarak şu şekilde tanımlanabilir;

$$A_1,A_2,\ldots,A_m \rightarrow B_1,B_2,\ldots,B_n \qquad \text{(II.1)}$$

Buradaki A_i ve B_j yapılan iş veya nesnelerdir. Bu kural genellikle "A_1, A_2,..., A_m meydana geldiğinde, sık olarak B_1, B_2,..., B_n aynı olay veya hareket içinde yer aldığını anlamına gelmektedir [11].

Örneklendirmek gerekirse; aşağıdaki kural bir marketin satış hareketlerinden elde edilmektedir.

Ürün(X,"Cips") $\rightarrow$ Ürün(X,"Kola") (II.2)

Burada X bir hareketteki değişkeni simgelemektedir. Bu kuraldan da cips alan müşterilerin aynı zamanda kola almaya yöneldikleri anlamı çıkarılmaktadır.

Aşağıdaki kural ise üç boyutlu bir veri ambarından gelmektedir:
Yaş, meslek ve ürün.

Yaş(X,"12–17"), Meslek(X,"Öğrenci")$\rightarrow$Ürün(X,"oyun konsolu-playstation") (II.3)

Bu kural ile "12–17 yaşları arasındaki öğrenci en çok "oyun konsolu (playstation) almaktadır" anlamı elde edilmektedir [11].

Başka bir örnek; bir elektronik ve dijital cihaz satışı yapan firmanın müşterilerinin alışveriş alışkanlıkları üzerine birliktelik kuralının formu;

$$P(X:müşteri, W) \wedge Q(X,Y) \Rightarrow alış(X,Z) \quad (II.4)$$

X, müşteri ilişki anahtarı; P ve Q, işe bağlı verinin bir parçası olan bağıl nitelik veya boyutlardan o an yüklenen değişkenler ve W, Y ve Z, müşteri X'in o ana kadar ki işlemlerinin nesne değişkenleridir.

Nitelik kuralları için arama yukarıdaki temel kurala uygun olmalıdır [11].

$$Yaş(X, \text{“30…39”}) \wedge gelir(X, \text{“60K…69K”}) \Rightarrow alış(X, \text{”Plazma TV”}) \quad (II.5)$$

[%4,5, %20]

$$Meslek(X, \text{“öğrenci”}) \wedge yaş(X, \text{“15…20”}) \Rightarrow alış(X, \text{“Oyun Konsolu”}) \quad (II.6)$$

[%5,2, %59]

Yukarıdaki ilk kuralda, otuzlu yaşlarındaki, yıllık gelirleri 60K-69K arasında olan müşterilerinin %20'si Plazma TV satın almış ve tüm satışların %4,5 'ini temsil eder. Bu açıklamadaki %20 ifadesi güven-confidence, %4,5 ifadesine ise destek-support olarak tanımlanır. Bir sonraki kuralda, yirmi yaş altı öğrenci olan müşterilerin %59'u oyun konsolu satın almış ve tüm satışların %5,2 'sini temsil eder. Bu açıklamaya göre ise %59 güven, %5,2 ise destek olarak tanımlanır.

II.7.4.2. Güven (Confidence) ve Destek (Support) Ölçüleri

Birliktelik kurallarını oluşturabilmek için destek (support) ve güven (confidence) değerlerini kullanarak, kullanıcı tarafından belirlenmiş minimum destek ve minimum güven değerlerinden yaygın birlikteliklerin belirlenmesi amaçlanır. Birliktelik kurallarının kullanışlı olması için hem konu ile ilgili hem de anlaşılabilir olması gerekir.

İşe bağlı veri satırları verilmiş olsun, (A⇒B) güveni aşağıdaki gibi tanımlanır [11].

Güven (A⇒B)=(A ve B'nin bulunduğu satır sayısı)/(A'nın bulunduğu satır sayısı) (II.7)

İşe bağlı veri kümemizin bir markette satılan gıda ürünleri olduğunu varsayalım. Birliktelik kuralı için güven %85; düşük yağlı peynir alan müşterilerin %85 'i diyet süt de satın aldığı anlamına gelir.

Güven değeri %100 olması demek kural bütün veri analizlerinde doğrudur ve bu kurallara **kesin(exact)** denir [15].

A ve B kümelerinin birliktelik kuralı (A⇒B) olarak gösterilirse destek değeri aşağıdaki gibi tanımlanır.

Destek (A⇒B)=(A ve B'nin bulunduğu satır sayısı)/(toplam satır sayısı) (II.8)

İşe bağlı veri kümemizin bir yapı markette satılan hırdavat ürünleri olduğunu varsayalım. Birliktelik kuralı için Destek %35; bütün müşterilerin %35'i hem çekiç hem de çivi satın aldığı anlamına gelir.

Yüksek güven ve destek değerine sahip kurallara **güçlü (strong)** kurallar adı verilir [15].

Güven ve destek ile ilgili başka bir örnek; Tablo II.1'deki alış hareketlerine bakarak toplam alış hareketlerine göre {Bisküvi, Yoğurt} ile Kola arasındaki ilişki şu şekilde açıklanabilir;

Tablo II.1 Ürün hareket tablosu [16].

ID	Ürünler
1	Kola, Ekmek, Bisküvi, Yoğurt
2	Kola, Bisküvi, Ekmek, Balık
3	Meyve suyu, Ekmek, Bisküvi, Yoğurt
4	Ekmek, Bisküvi, Yoğurt
5	Kola, Meyve suyu, Bisküvi, Yoğurt

$$\text{Destek-support} = \frac{(\text{Bisküvi, Yoğurt, Kola})}{\text{Toplam Hareket}} = \frac{2}{5} = 0{,}4$$

$$\text{Güven-confidence} = \frac{(\text{Bisküvi, Yoğurt, Kola})}{(\text{Bisküvi, Yoğurt})} = \frac{2}{4} = 0{,}5 \qquad \text{(II.9)}$$

Bu eşitlikten de anlaşılacağı gibi, {Bisküvi, Yoğurt} ⇒ Kola kuralı % 40 destek, %50 güven ölçülerine sahiptir [16]. Buradan elde edilen kuralda, bütün alışverişlerin %40'ında Bisküvi, Yoğurt, Kola ürünlerinin birlikte alındığı (veya

Bisküvi, Yoğurt, Kola'nın toplam fiş hareketlerinde birlikte bulunması olasılığı %40' dır), Bisküvi ve Yoğurt ürünlerini satın alan müşterilerin %50'sinin aynı alışverişte Kola ürününü de satın aldığı (veya Bisküvi ve Yoğurt ürünlerini alan bir müşteri %50 olasılıkla Kola da almıştır) söylenebilir.

Yukarıdaki gibi uygulanan market sepeti uygulaması müşterilerin yaptıkları alışverişlerdeki ürünler arasındaki birliktelikleri bularak müşterilerin satın alma alışkanlıklarını analiz eder. Bu tip birlikteliklerin keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve market yöneticileri de bu bilgi ışığında daha etkili satış stratejileri geliştirebilirler [17]. Elde edilen kurallar ışığında rafları düzenleyen market yöneticileri ürünlerindeki satış oranını arttırabilirler. Örneğin bir marketin müşterilerinin bisküvi, yoğurt ve kola ürünlerini birlikte satın alma oranı yüksek ise, market yöneticileri bu ürünlerin raflarını yan yana koyarak bu ürünlerin satışlarını arttırabilirler.

II.7.4.3. Birliktelik Kurallarının Başarılı Ve Başarısız Olduğu Durumlar

Birliktelik kurallarının sağlamış olduğu en büyük avantajlardan birisi, veri madenciliği sonunda ortaya çıkan sonuçların kolayca anlaşılabilir olmasıdır.

Her ne kadar kayıtların sayısı ve kombinasyon seçimine göre işlem adedi artsa da birliktelik kurallarının her adımda gerekli olan hesaplamalar diğer yöntemlere göre (genetik algoritmalar, yapay sinir ağları vb.) çok daha basittir.

Veri madenciliği aşamasında yapılması gereken hesaplamaların karmaşık olmaması bu metodun diğer bir üstünlüğüdür.

Diğer bir yandan birliktelik kurallarının kullanılması ile ilgili zorluklar da vardır. Sorunun boyutu büyüdükçe, gerekli hesaplamalar üstel olarak artmaktadır.

Destek ve güven eşik değerleri üretilen kural sayısında sınırlama getirirler fakat eşik değerlerinin çok düşük belirlendiği durumda kullanıcı gerçekten ilgilendiği kuralları kaybetme tehlikesi ile karşı karşıya kalır.

II.7.4.4. Birliktelik Kurallarının Belirlenmesinde Kullanılan Temel Algoritmalar

Birliktelik kurallarının belirlenmesinde kullanılan bazı algoritmalar şunlardır; AIS, Apriori, DHP, Partition. Bu algoritmalardan en çok bilineni Apriori algoritmasıdır.

- AIS Algoritması

AIS algoritması, Agrawal, Imielinski ve Swami tarafından 1993 yılında veritabanındaki tüm yaygın nesne kümelerini oluşturmak için geliştirilmiş ve yayınlanmış ilk algoritmadır. Karar destek sorgulamaları yapmak için veri tabanlarının işlevlerini artırmaya odaklanmıştır. Bu algoritmanın hedefi nitelikli kuralları bulmaktır. Bu teknik, sonuçta sadece bir nesne ile kısıtlanmıştır. Birliktelik kuralları $X \Rightarrow Ij \mid \alpha$ biçiminde tanımlanır. Burada X nesneler kümesi, Ij, I alanındaki tek bir nesne ve α da kuralın güven değeridir [15].

- Apriori Algoritması

Agrawal ve diğerleri tarafından 1994 yılında geliştirilen Apriori algoritması veri madenciliğinde kullanılmakta olan en yaygın tekniklerden biridir. Algoritmanın ismi, bilgileri bir önceki adımdan aldığı için "prior" anlamında Apriori'dir.

Bu algoritmada veri tabanı birden fazla taranarak sık geçen öğe kümeleri bulunur. İlk taramada bir elemanlı minimum destek ölçütünü sağlayan sık geçen öğe kümeleri bulunur. Daha sonraki taramalarda ise bir önceki taramada bulunan sık geçen öğe kümeleri aday kümeler adı verilen yeni potansiyel sık geçen öğe kümelerini üretmek için kullanılır. Aday kümelerin destek değerleri tarama sırasında hesaplanır ve aday kümelerinden minimum destek ölçütü sağlayan kümeler o geçişte üretilen sık geçen öğe kümeleri olur. Sık geçen öğe kümeleri bir sonraki geçiş için aday küme olurlar. Bu süreç yeni bir sık geçen öğe kümesi bulunmayana kadar devam eder [4].

Bu tez çalışmasında geliştirilen yazılımda Apriori algoritması kullanılmıştır. Bölüm IV'te Apriori algoritması daha geniş olarak ele alınacaktır.

- DHP Algoritması

DHP (Direct Hashing and Pruning) algoritması da k-öğe kümesi adaylarını k-1 elemanlı sık geçen öğe kümelerinden elde eder. Apriori algoritmasından farklı olarak sık geçen küme adaylarını (arama uzayını) azaltır. DHP algoritması da veritabanının birçok kere taranmasını gerektirir [11,18].

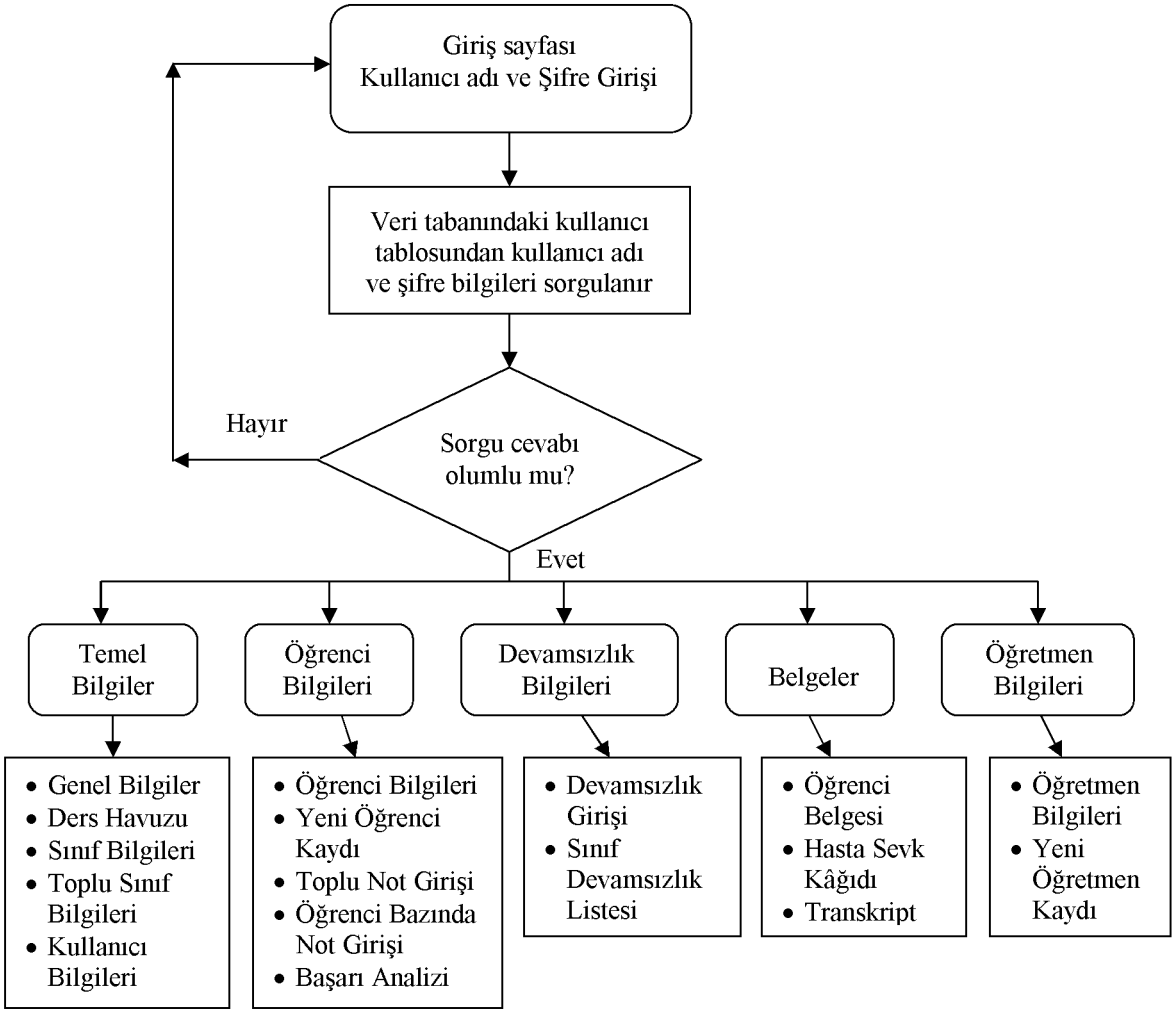
- Partition Algoritması

Partition algoritması veri tabanlarındaki tarama sayısını ikiye indirir. Veri tabanını küçük bölmelere böler ve her bir bölmenin ana bellekte tutulabileceğini varsayar. İlk taramada potansiyel olarak sık geçen öğe kümelerini bulur, ikinci taramada ise öğe kümelerinin destek değerleri hesaplanır[15].

BÖLÜM III

OKUL OTOMASYON YAZILIMININ GELİŞTİRİLMESİ

Burada geliştirilen otomasyon yazılımının amacı insan müdahalesini kısmen ortadan kaldırarak, işlemlerin otomatik olarak yapılmasını sağlamak, okul yöneticilerinin işlerini kolaylaştırmak ve öğrenci verileri üzerine veri madenciliği uygulayarak, veri tabanı içinde tutulan verilerden anlamlı, faydalı daha önceden bilinmeyen bilgilerin ortaya çıkarılması ve bu bilgilerden faydalanılarak başarı analizleri yapmaktır. Şekil III.1'de uygulama yazılımının aşamaları görülmektedir.



Şekil III.1 Okul otomasyon yazılımının aşamaları

III.1. ARAŞTIRMA YÖNTEMİ

Bu tez kapsamında okul yöneticileriyle görüşülmüş ve okul otomasyonun da olması gerekenler hakkında fikir alış verişinde bulunulmuştur. İstanbul Eyüp İ.M.K.B Ticaret Meslek Lisesi öğrenci verileri incelenmiş ve yapılan otomasyon yazılımı üzerinde bu veriler ile veri madenciliği kullanılarak birliktelik kuralı algoritmalarından Apriori algoritmasının uygulanmasına karar verilmiştir.

III.2. ARAŞTIRMA ARAÇLARI

III.2.1. Donanım

- Bilgisayar, yazılımı geliştirmek için kullanıldı.
- Yazıcı, uygulama yazılımında raporlama işlemleri için kullanıldı.

III.2.2. Yazılım

- Uygulama yazılımının geliştirileceği bilgisayarda işletim sistemi olarak Microsoft Windows XP Professionel Edition
- Yazılımın geliştirilmesi için Microsoft VB.NET programlama dili
- Veri tabanı programı olarak Microsoft Access 2003 kullanılmıştır.

III.3. OTOMASYON YAZILIMININ BÖLÜMLERİ

Otomasyon yazılımı beş ayrı bölümden oluşmaktadır. Bunlar:

Temel Bilgiler

- Genel Bilgiler
- Ders Havuzu
- Sınıf Bilgileri
- Toplu Sınıf Bilgileri
- Kullanıcı Bilgileri
- Sınıf Tanımlamaları
- 2. Döneme Geçiş
- Yeni Eğitim Öğretim Dönemi

Öğrenci Bilgileri

- Öğrenci Bilgileri
- Yeni Öğrenci Kaydı
- Toplu Not Girişi

- Öğrenci Bazında Not Girişi
- Başarı Analizi (Veri madenciliği işlemlerinin yapıldığı bölüm)

Devamsızlık Bilgileri

- Devamsızlık Girişi
- Sınıf Devamsızlık Listesi

Belgeler

- Öğrenci Belgesi
- Hasta Sevk Kağıdı
- Transkript

Öğretmen Bilgileri

- Öğretmen Bilgileri
- Yeni Öğretmen Kaydı

III.4. VERİ TABANINDA TABLO YAPILARININ OLUŞTURULMASI

Uygulamada kullanılacak olan veri tabanının tek bir veri tabanı olması ve tüm tabloların bu veri tabanı içerisinde oluşturulması uygun görülmüştür. Bu doğrultuda Microsoft Access 2003 programında "Otomasyon" adında bir veri tabanı oluşturulmuştur. Otomasyon veri tabanının içerisinde ise tablo yapıları şu şekilde oluşturulmuştur:

Ders_havuzu Tablosu: Bu tabloda okulda gösterilen derslere ait bilgiler bulunmaktadır. Ders_havuzu tablosundaki bilgiler, aşağıda belirtilen veri alanlarında tutulmaktadır. Şekil III.2'de ders_havuzu tablosunun veri yapısı yer almaktadır.

- Ders_kodu: Derslere ait ders kodları,
- Ders_adı: Derslerin isimleri,
- Hds: Haftalık ders saati
- Sınıf: Derslerin hangi sınıfta okutulacağı
- Öğretmen: Dersin öğretmeni

ders_havuzu : Tablo

	Alan Adı	Veri Türü
	id	Otomatik Sayı
🔑	ders_kodu	Metin
	ders_adı	Metin
	hds	Sayı
	sınıf	Metin
	ogretmen	Metin

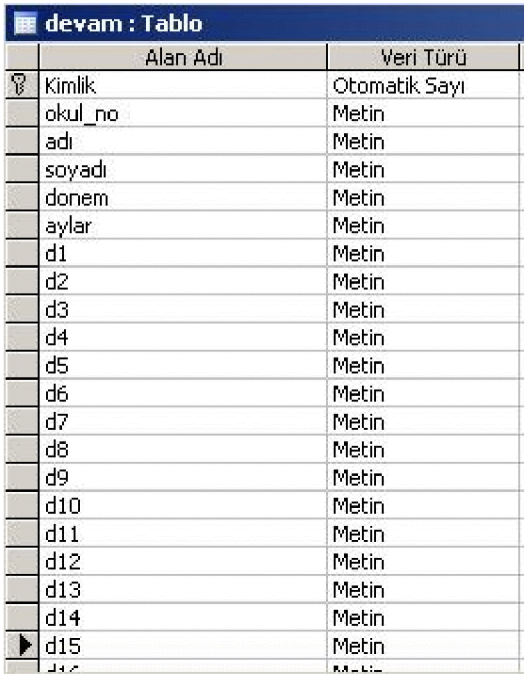
Şekil III.2 Ders_havuzu tablosunun veri yapısı

Devam Tablosu: Devam tablosunda öğrencilerin devamsızlık bilgileri tutulmaktadır. Aşağıdaki alanlardan oluşmaktadır. Şekil III.3'te devam tablosunun veri yapısı görülmektedir.

- Okul_no: Öğrencilerin okul numaraları
- Adı: Öğrencilerin adları
- Soyadı: Öğrencilerin soyadları
- Donem: Eğitim öğretim yılı
- Aylar: Devam durumunu gösteren Eylül-Haziran arası aylar
- d1: Ayın birinci günü

(Bu araya d1'den d31'e kadar ayın günlerini tutan alan isimleri gelmektedir)

- d31: Ayın otuz birinci günü

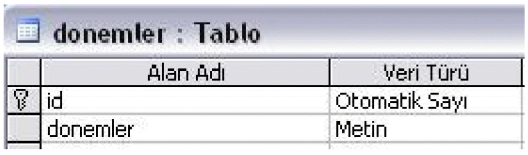

devam : Tablo

Alan Adı	Veri Türü
Kimlik	Otomatik Sayı
okul_no	Metin
adı	Metin
soyadı	Metin
donem	Metin
aylar	Metin
d1	Metin
d2	Metin
d3	Metin
d4	Metin
d5	Metin
d6	Metin
d7	Metin
d8	Metin
d9	Metin
d10	Metin
d11	Metin
d12	Metin
d13	Metin
d14	Metin
d15	Metin

Şekil III.3 Devam tablosunun veri yapısı

Dönemler Tablosu: Dönem isimlerinin tutulduğu tablodur. Şekil III.4'de dönemler tablosunun veri yapısı görülmektedir.

- Dönemler: Eğitim-Öğretim dönemlerinin isimleri


donemler : Tablo

Alan Adı	Veri Türü
id	Otomatik Sayı
donemler	Metin

Şekil III.4 Dönemler tablosunun veri yapısı

Genel_bilgiler Tablosu: Okulun genel bilgilerinin tutulduğu tablodur (okul adı, müdür, müdür yardımcısı, okulun bulunduğu il, okulun bulunduğu ilçe vb.).

- Ad: Tutulacak bilgilerin başlıkları (Ör: Okul Adı)
- Değer: Başlıkların içerikleri (Ör: Eyüp İ.M.K.B Ticaret Meslek Lisesi)



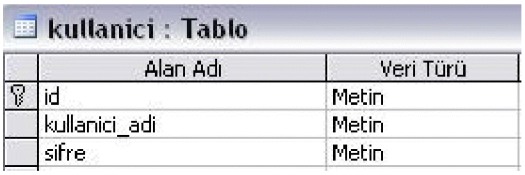
genel_bilgiler : Tablo

	Alan Adı	Veri Türü
🔑	id	Otomatik Sayı
	ad	Metin
	deger	Metin

Şekil III.5 Genel_bilgiler tablosunun veri yapısı

Kullanıcı Tablosu: Bu tabloda kullanıcılara ait kullanıcı adları ve parolalar bulunmaktadır. Şekil III.6'da kullanıcı tablosunun veri yapısı görülmektedir.

- Kullanıcı_adi: Kullanıcı adları
- Şifre: Kullanıcı parolaları



kullanici : Tablo

	Alan Adı	Veri Türü
🔑	id	Metin
	kullanici_adi	Metin
	sifre	Metin

Şekil III.6 Kullanıcı tablosunun veri yapısı

Notlar Tablosu: Öğrencilerin yazılı sınav, sözlü ve ödevlerden almış olduğu notların tutulduğu tablodur. Aşağıdaki alanlardan oluşmaktadır. Şekil III.7'de notlar tablosunun veri yapısı görülmektedir.

- Ders_kodu: Derslerin kodları
- Ders_adı: Derslerin adı
- Okul_no: Öğrencilerin okul numaraları
- Dönem: Eğitim-Öğretim dönemi
- Adı: Öğrencilerin adı
- Soyadı: Öğrencilerin soyadı
- Y1: Birinci yazılı sınavı notu
- Y2: İkinci yazılı sınavı notu
- Y3: Üçüncü yazılı sınavı notu
- Y4: Dördüncü yazılı sınavı notu

- S1: Birinci sözlü sınavı notu
- S2: İkinci sözlü sınavı notu
- S3: Üçüncü sözlü sınavı notu
- S4: Dördüncü sözlü sınavı notu
- Ödev: Ödev notu
- Ortalama: Yazılı sınavlarının, sözlülerin ve ödev notlarının ortalaması

notlar : Tablo

	Alan Adı	Veri Türü
🔑	kimlik	Otomatik Sayı
	ders_kodu	Metin
	ders_adı	Metin
	okul_no	Metin
	donem	Metin
	adı	Metin
	soyadı	Metin
	y1	Metin
	y2	Metin
	y3	Metin
	y4	Metin
	s1	Metin
	s2	Metin
	s3	Metin
	s4	Metin
	odev	Metin
	ortalama	Sayı

Şekil III.7 Notlar tablosunun veri yapısı

Öğrenci_bilgileri Tablosu: Örencilere ait bilgilerin tutulduğu tablodur. Aşağıdaki alanlardan oluşmaktadır. Şekil III.8'de öğrenci_bilgileri tablosunun veri yapısı görülmektedir.

- Okul_no: Öğrencilerin okul numarası
- Sınıf: Öğrencilerin öğrenim gördüğü sınıf
- Adres: Öğrencilerin ev adresi
- Tel: Öğrencilerin velilerinin telefon numarası
- Veli: Öğrenci velisi
- Veli_iş: Öğrenci velisinin işi
- TC_kimlik: Öğrenci T.C. kimlik numarası
- Ad: Öğrencilerin adı
- Soyad: Öğrencilerin soyadı
- Baba: Öğrencilerin baba adı
- Anne: Öğrencilerin anne adı

- Doğum_yeri: Öğrencilerin doğum yeri
- Doğum_tarihi: Öğrencilerin doğum tarihi

ogrenci_bilgileri : Tablo

Alan Adı	Veri Türü
id	Otomatik Sayı
okul_no	Metin
sınıf	Metin
adres	Metin
tel	Metin
veli	Metin
veli_iş	Metin
tc_kimlik	Metin
ad	Metin
soyad	Metin
baba	Metin
anne	Metin
dogum_yeri	Metin
dogum_tarihi	Metin

Şekil III.8 Öğrenci_bilgileri tablosunun veri yapısı

Öğretmen_bilgileri Tablosu: Örencilere ait bilgilerin tutulduğu tablodur. Aşağıdaki alanlardan oluşmaktadır. Şekil III.9'da öğretmen_bilgileri tablosunun veri yapısı görülmektedir.

- Mezun_okul: Öğretmenlerin mezun olduğu üniversiteler
- Branş: Öğretmenlerin branşları
- Adres: Öğretmenlerin adresleri
- Tel_ev: Öğretmenlerin ev telefonu numaraları
- Tel_cep: Öğretmenlerin cep telefonu numaraları
- Medeni_hal: Öğretmenlerin medeni halleri
- TC_kimlik: Öğretmenlerin T.C. kimlik numaraları
- Ad: Öğretmenlerin adları
- Baba: Öğretmenlerin baba adları
- Anne: Öğretmenlerin anne adları
- Doğum_yeri: Öğretmenlerin doğum yeri
- Doğum_tarihi: Öğretmenlerin doğum tarihi

ogretmen_bilgileri : Tablo

Alan Adı	Veri Türü
id	Otomatik Sayı
mezun_okul	Metin
brans	Metin
adres	Metin
tel_ev	Metin
tel_cep	Metin
medeni_hal	Metin
tc_kimlik	Metin
ad	Metin
baba	Metin
anne	Metin
dogum_yeri	Metin
dogum_tarihi	Metin

Şekil III.9 Öğretmen_bilgileri tablosunun veri yapısı

Resmi_adres Tablosu: Okulun iletişim bilgilerinin tutulduğu tablodur. Şekil III.10'da resmi_adres tablosunun veri yapısı görülmektedir.

- Ad: Tutulacak bilgilerin başlıkları (Ör: Adres)
- Değer: Başlıkların içerikleri (Ör: Karadolap Mah. Gülbahar Sok. Eyüp)

resmi_adres : Tablo

Alan Adı	Veri Türü
id	Otomatik Sayı
ad	Metin
deger	Metin

Şekil III.10 Resmi_adres tablosunun veri yapısı

Sınıflar Tablosu: Sınıf isimlerinin tutulduğu tablodur. Şekil III.11'de sınıflar tablosunun veri yapısı görülmektedir.

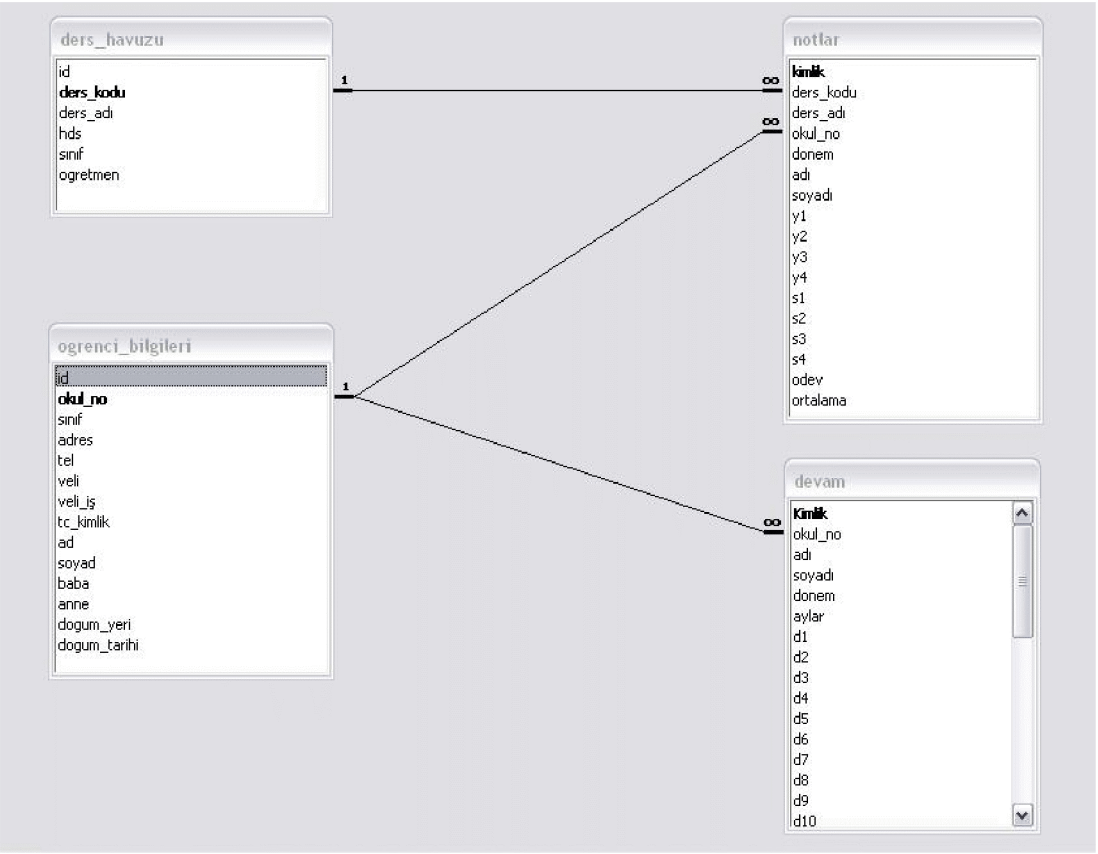
- Sınıf: Sınıf simleri
- Hds: haftalık ders saatleri

sınıflar : Tablo

Alan Adı	Veri Türü
id	Otomatik Sayı
sınıf	Metin
hds	Sayı

Şekil III.11 Sınıflar tablosunun veri yapısı

Şekil III.12'de veri tabanındaki bazı tablolar arasındaki ilişkiler görülmektedir.



Şekil III.12 Uygulamada kullanılan veri tabanındaki bazı tablolar arasındaki ilişkiler

III.5. OTOMASYON YAZILIMININ OLUŞTURULMASI

III.5.1. Yazılımı Geliştirme Ortamı

Uygulama yazılımı Microsoft Visual Studio .NET platformunda Microsoft Visual Basic.NET programlama dili kullanılarak geliştirilmiştir.

Microsoft Visual Studio.NET kendi içinde bulunan dilleri aynı ortamda kullanarak kolaylıkla tasarım yapmayı, hataları tespit etmeyi ve oluşturulan programları başka bir ortama aktarmayı sağlayan program topluluğudur.

Veriler Access 2003 veri tabanına yazılıp okunmakta ve uygulamanın işleyişine göre Access 2003 veri tabanında geçici tablolar oluşturulmaktadır.

Uygulama yazılımı ile veri tabanı arasındaki iletişimi ADO.NET nesneleri sağlamaktadır. ADO.NET, veri kaynakları ile bağlantı kurarak, veriler üzerinde işlemler yapmak için kullanılır. Kod ve veri kaynağı arasındaki ilişkiyi en yüksek performansta sağlar.

İki tür .NET Data Provider vardır:

SQL Server .NET: SQL Server 7.0 ve sonrası için kullanılır.

OLE DB.NET: SQL Server 6.5 ve öncesi ile Oracle, Sybase ve Access gibi veri tabanlarında kullanılır.

Bu çalışmada OLE DB.NET provider'ı ve buna bağlı ADO.NET nesneleri kullanılmıştır. Bu nesneler:

- Connection: Veritabanı ile bağlantı sağlar.
- Command: Veri tabanına SQL komutları gönderip çalıştırılmasını sağlar. Dört tane metodu vardır. Bunlar;
  ExecuteReader
  ExecuteNonQuery
  ExecuteScalar
  ExecuteXmlReader
- DataReader : Command nesnesi ile gelen verilerin biriktirilmesini sağlar.
- DataAdapter: Veritabanından gelen verilerin dataset'e aktarılmasını ve güncelleme yapılmasını sağlar.
- Dataset: Verilerin, ilişkilerin (relationship) ve constraint'lerin tutulmasını sağlar.

III.5.2. Otomasyon Yazılımının Aşamaları

Bu bölümde otomasyon yazılımının özellikleri, yazılımın işleyişi ve bazı bölümlerin, program kod parçaları verilmektedir.

Otomasyon yazılımını işleyişi şu şekildedir;

Yazlım açıldığında ilk olarak kişilerin rasgele girişlerini engellemek amacıyla Şekil III.13'teki kullanıcı adı ve parolanın sorulduğu giriş ekranı gelir. Yazılımla ilgili bütün veriler "otomasyon" adlı veri tabanında tutulmaktadır. Veri tabanıyla bağlantı sağlandıktan sonra kullanıcı adı ve parola bilgileri doğrulandığında kullanıcı otomasyon yazılımına giriş hakkı kazanmaktadır.



Şekil III.13 Otomasyon yazılımına giriş ekranı

Yazılımın bulunduğu klasörün altında bulunan “otomasyon.mdb” veri tabanı dosyası ile ilişkiye geçmek için yazılması gereken kod aşağıdaki şekildedir.

```
Dim conn As New OleDb.OleDbConnection("Provider=
Microsoft.Jet.Oledb.4.0;Data Source=" & Application.StartupPath &
"\Otomasyon.mdb")
```

Oledbconnection ile bağlanacağımız veri tabanının tipi ve yeri belirlendikten sonra, veri tabanında sorgulama yapmak için oledbcommand nesnesi tanımlanır. Open metodu veri kaynağına ulaşıp açmak için kullanılır.

```
Dim co As New OleDb.OleDbCommand
co.Connection = conn
conn.Open()
```

Bundan sonraki yapılan bütün veri tabanı bağlantı işlemleri yukarıdaki kodlar ile yapılmaktadır. Bundan dolayı bu kodlar ileride tekrar yazılıp açıklanmayacaktır. Bağlantı kurulduktan ve gerekli tanımlamalar yapıldıktan sonra oledbcommand nesnesinin commandtext özelliği ile sorgulama yapmak için gerekli SQL ifadesi aşağıdaki gibi tanımlanır.

```
co.CommandText = "select * from kullanici"
```

Command objesi için özellikler tanımlandıktan sonra komutun çalıştırılması için bir metodun çağırılması gerekir. Bu metotlar çevrimiçi olarak çalışır. Yani bu komutların çalışması için mutlaka veri tabanı ile bağlantının aktif olması gerekir, yoksa işlemler gerçekleşmez. Burada Executereader metodu kullanılmaktadır. Bu metot, belirlenen sütun değerlerinin tüm satırlarını döndürür. Bunun çalışabilmesi için mutlaka bir oledbdatareader nesnesi tanımlanması gerekir. Oledbdatareader nesnesi bir dizi gibi davranır ve executereader ile çalıştırılan oledbcommand nesnesinin getirdiği bilgileri bu diziye aktararak, kullanılmasını sağlar.

Kullanici tablosundaki tüm kayıtlar bir nevi dizi olan oledbdatareader içine yüklenir ve burada hepsi tek tek taratılarak kullanici_ad ve parola verileri ile karşılaştırılır. Eğer kayıtlardan herhangi biri uyuşursa kullanıcı doğru giriş yapmış demektir ve yazılımın bütün menüleri kullanılabilir hale getirilir, uyuşmaz ise

kullanıcıdan bilgileri tekrar girmesi istenir. Aşağıda bu işlemi gerçekleştiren kodlar görülmektedir.

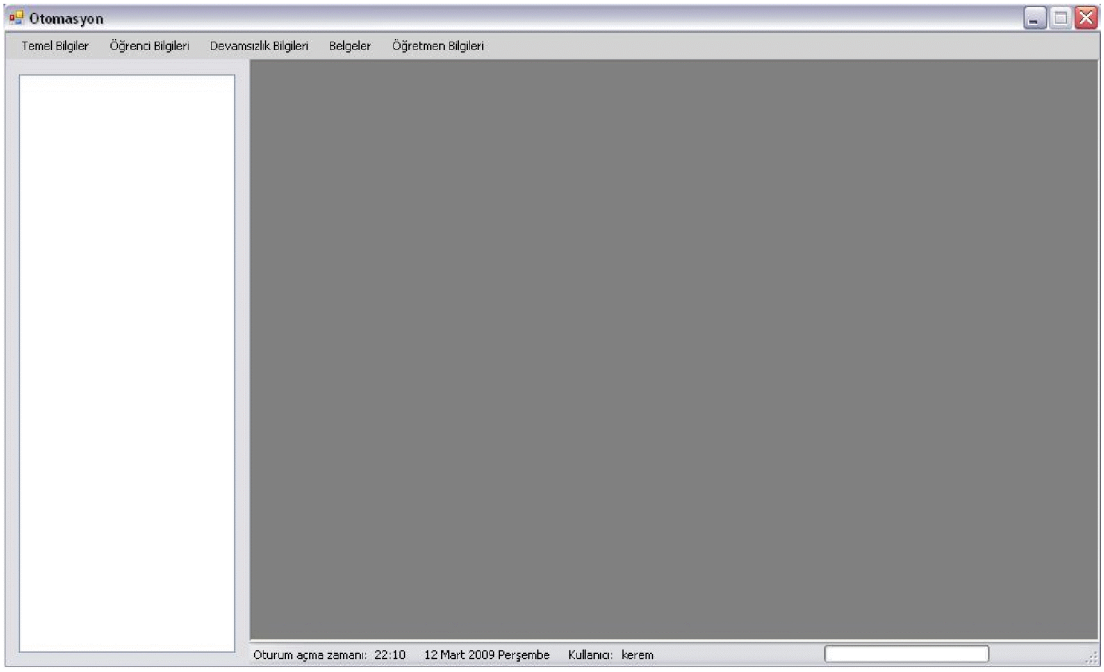
```
Dim yaz As OleDb.OleDbDataReader
yaz = co.ExecuteReader(CommandBehavior.CloseConnection)
Dim say = 0
Dim kullanici_ad As String = TextBox1.Text
Dim parola As String = TextBox2.Text
Do While yaz.Read
   If kullanici_ad = yaz.GetString(1) And parola = yaz.GetString(2) Then
      otomasyon.DosyaToolStripMenuItem.Enabled = True
      otomasyon.ÖğrenciBilgileriToolStripMenuItem.Enabled = True
      otomasyon.DevamsızlıkBilgileriToolStripMenuItem.Enabled = True
      otomasyon.BelgelerToolStripMenuItem.Enabled = True
      otomasyon.ÖğretmenBilgileriToolStripMenuItem.Enabled = True
      otomasyon.ComboBox1.Enabled = True
      Me.Close()
      Exit Do
   Else
      MsgBox("Yanlış giriş yaptınız. Lütfen tekrar deneyin.")
      TextBox1.Focus()
      Exit Do
   End If
Loop
```

Eğer kullanıcı şifresini hatırlamıyor ise "Parolamı Unuttum" butonunu tıkladığında Şekil III.14'teki kimlik doğrulama ekranıyla karşılaşmaktadır. Burada, kayıt olduğu esnada yazmış olduğu soruya gizli yanıtı verdiği takdirde şifresini öğrenebilmekte ve programa giriş yapabilmektedir.



Şekil III.14 Kimlik doğrulama ekranı

Giriş hakkını elde eden kullanıcıya yazılımın bütün menüleri aktif olur ve Şekil III.15'teki gibi kullanıma hazır hale gelir.

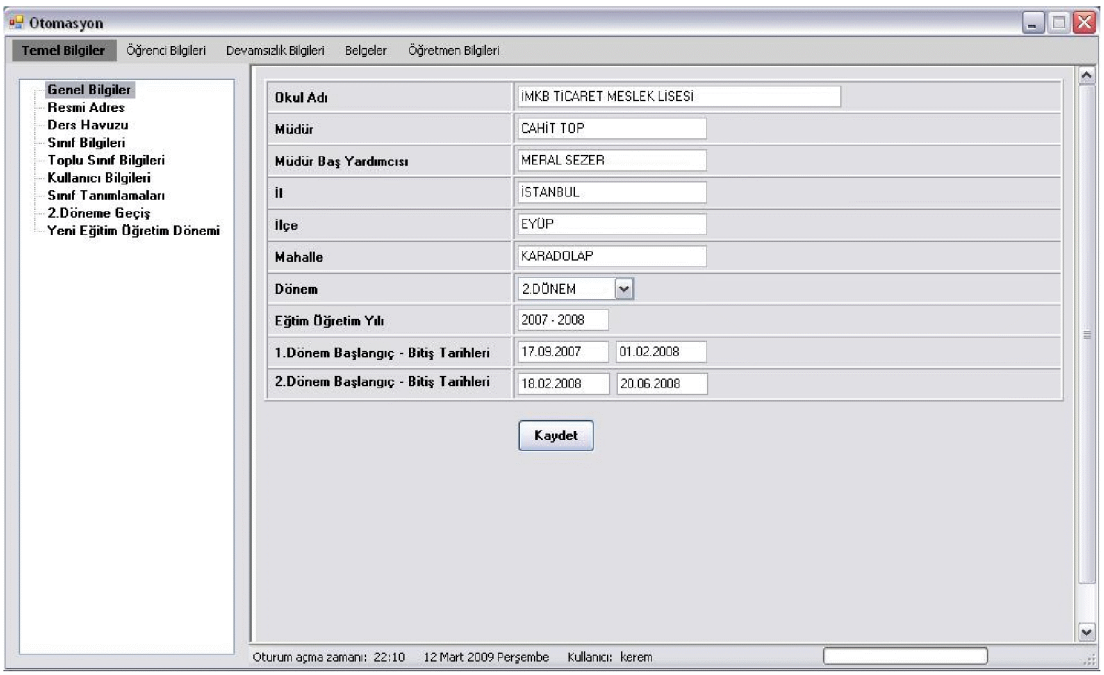


Şekil III.15 Yazılımın ana penceresinin görünümü

III.5.2.1. Temel Bilgiler Menüsü

Temel Bilgiler menüsü altında dokuz adet seçenek vardır. Bunlar genel bilgiler, resmi adres, ders havuzu, sınıf bilgileri, toplu sınıf bilgileri, kullanıcı bilgileri, sınıf tanımlamaları, 2. döneme geçiş, yeni eğitim öğretim dönemidir.

Genel Bilgiler: *Genel Bilgiler* seçeneği seçildiğinde Şekil III.16'daki *Genel Bilgiler* ekranı gelir. Bu ekran okul adı, müdür adı, müdür başyardımcısı adı gibi genel bilgilerin kayıtlarının tutulduğu yerdir. Kullanıcı yeni verileri girip kaydet butonuna bastığı anda yeni bilgiler veri tabanındaki genel_bilgiler tablosuna kaydedilir.



Şekil III.16 Temel Bilgiler menüsü altındaki Genel Bilgiler penceresinin görünümü

Aşağıdaki kod ile genel_bilgiler tablosundaki ad alanı okul_adı olan kaydın değeri, Textbox2' ye (form üzerindeki Okul Adı alanı) yazılan veri ile değiştirilir.

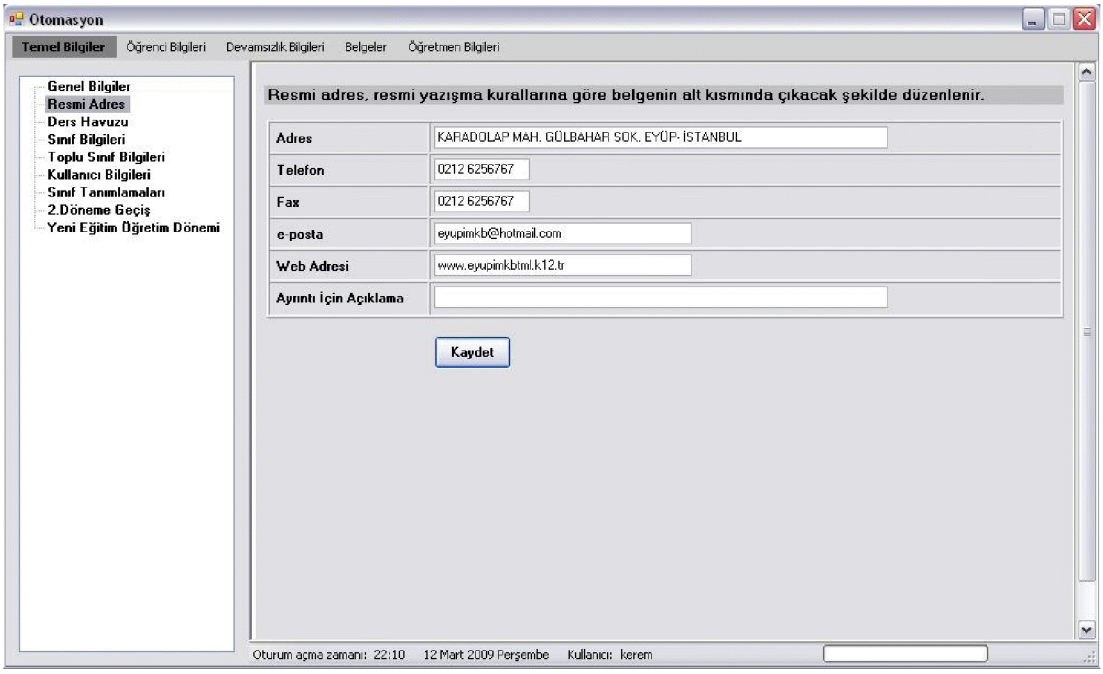
```
co.CommandText = "update genel_bilgiler set deger='" & TextBox2.Text & "'
where (ad='okul_adı' )"
Dim sonuc1 = co.ExecuteNonQuery
```

Aşağıdaki kod ile genel_bilgiler tablosundaki ad alanı müdür_adı olan kaydın değeri Textbox3' e (form üzerindeki Müdür alanı) yazılan veri ile değiştirilir.

```
co.CommandText = "update genel_bilgiler set deger='" & TextBox3.Text & "'
where (ad='müdür_adı' )"
Dim sonuc2 = co.ExecuteNonQuery
```

Genel Bilgiler penceresindeki diğer alanları veri tabanına kaydetmek için de benzer kodlar yazılmaktadır.

Resmi Adres: *Resmi Adres* seçeneği seçildiğinde Şekil III.17'deki *Resmi Adres* ekranı gelir. Bu ekran okulun adresi, telefon numarası gibi iletişim bilgilerin kayıtlarının tutulduğu yerdir. Kullanıcı yeni verileri girip kaydet butonuna bastığı anda yeni bilgiler veri tabanındaki resmi_adres tablosuna kaydedilir.



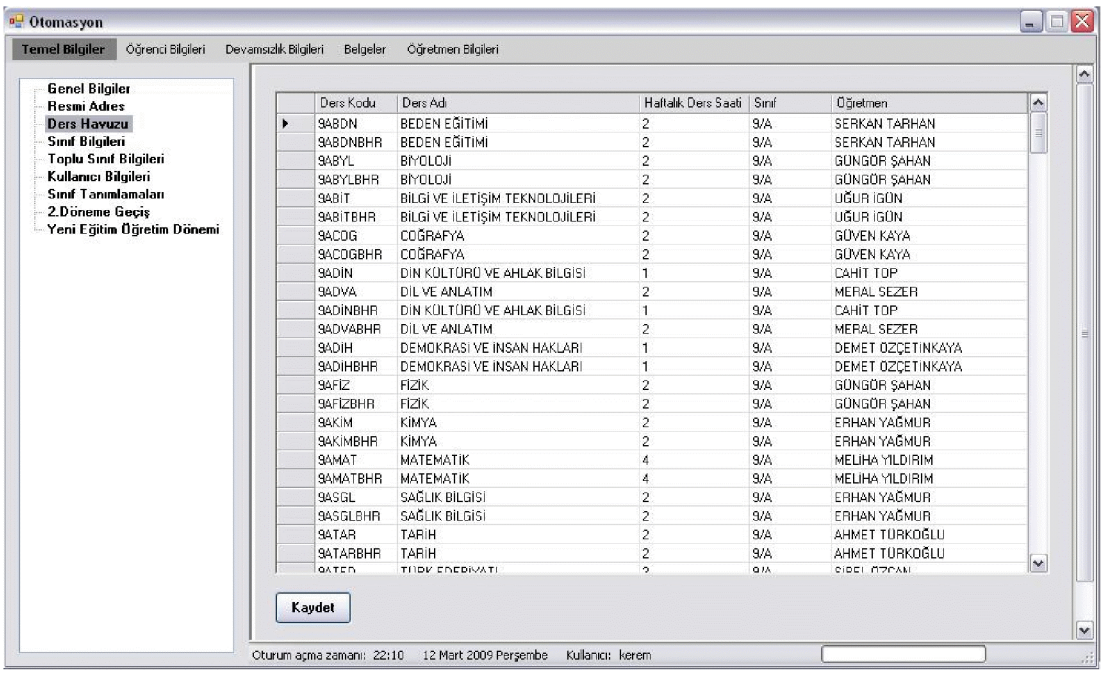
Şekil III.17 Temel Bilgiler menüsü altındaki Resmi Adres penceresinin görünümü

Aşağıdaki kod ile resmi_adres tablosundaki ad alanı adres olan kaydın değeri Textbox44' e (form üzerindeki Adres alanı) yazılan veri ile değiştirilir.

```
co.CommandText = "update resmi_adres set deger='" & TextBox44.Text & "'
where (ad='adres' )"
Dim sonuc1 = co.ExecuteNonQuery
```

Resmi Adres penceresindeki diğer alanları veri tabanına kaydetmek için de benzer kodlar yazılmaktadır.

Ders Havuzu: *Ders Havuzu* seçeneği seçildiğinde Şekil III.18'deki *Ders Havuzu* ekranı gelir. Burada okulda gösterilecek olan dersler oluşturulmaktadır. Dersler oluşturulurken ders kodu, ders adı, dersin haftada kaç saat gösterileceği, o dersin gösterileceği sınıf ve o dersin öğretmeni bilgileri girilir. Kullanıcı yeni verileri girip kaydet butonuna bastığı anda yeni bilgiler veri tabanındaki ders_havuzu tablosuna kaydedilir.



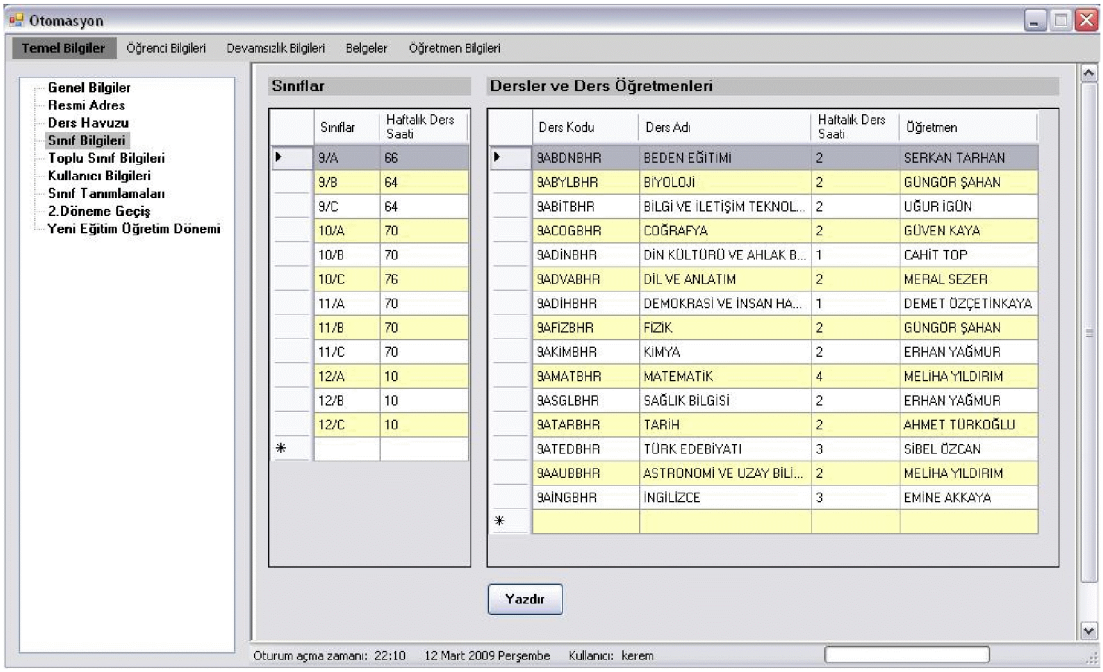
Şekil III.18 Temel Bilgiler menüsü altındaki Ders Havuzu penceresinin görünümü

Aşağıdaki kod ile dataset nesnesine bağlanmış olan datagrid içindeki bilgiler ders_havuzu tablosuna kaydedilmektedir.

```
OleDbDataAdapter1.Update(DataSet11.Tables("ders_havuzu"))
MsgBox("Kaydedildi")
```

Aynı zamanda kaydet butonuna basıldığında ders_havuzu tablosu, notlar tablosu ile ilişkili olduğundan, yeni eklenmiş olan bir ders hangi sınıf için eklenmiş ise o sınıftaki bütün öğrencilerin notlar tablosundaki ders notları bilgilerine de eklenmiş olur.

Sınıf Bilgileri: *Sınıf Bilgileri* seçeneği seçildiğinde Şekil IV.18'deki *Sınıf Bilgileri* ekranı gelir. Burada sınıf adları, sınıflarda bir haftada okutulan toplam ders saati, derslerin ayrı ayrı adları, ders kodları, bir haftada kaç saat okutuldukları ve derse giren öğretmen adı bilgileri yer almaktadır. Ayrıca yazdır butonuna basıldığında Şekil III.19'daki dersler ve ders öğretmenleri bölümünün yazıcı çıktısı alınmaktadır.



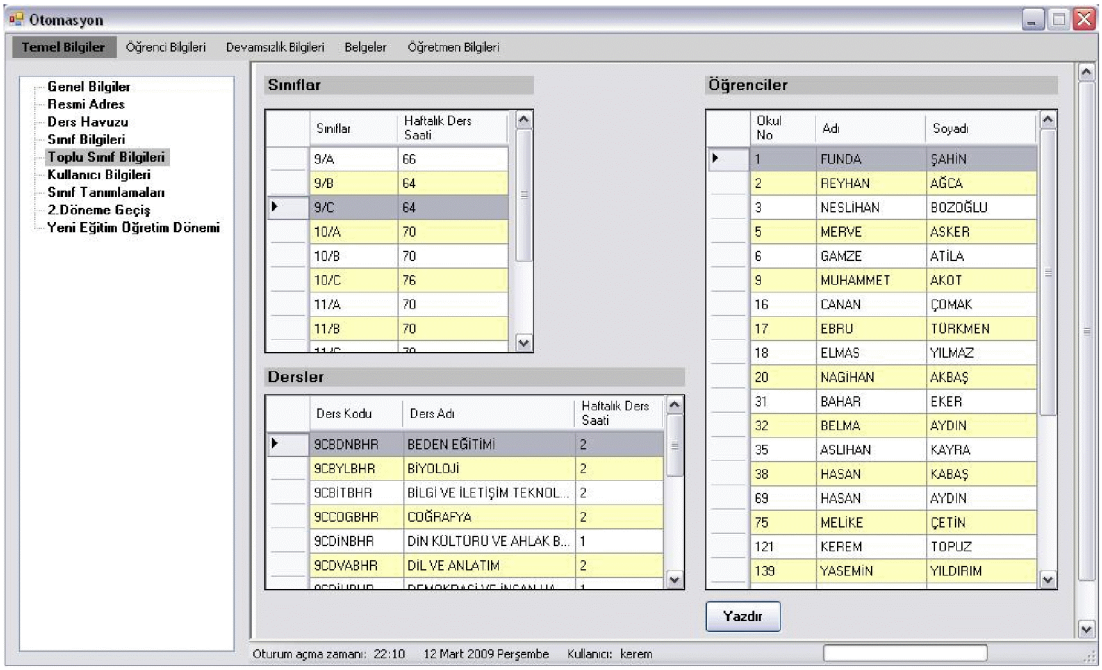
Şekil III.19 Temel Bilgiler menüsü altındaki Sınıf Bilgileri penceresinin görünümü

Şekil III.19'daki sınıflar bölümünden istenilen sınıf seçildiğinde dersler ve ders öğretmenleri bölümünde o sınıfla ilgili bilgiler anında görüntülenmektedir.

Aşağıdaki kodda sınıflar tablosundaki bütün kayıtlar seçilmekte, bilgileri tutacak bir dataset tanımlanmakta ve dataadapter' i kullanarak verileri dataset içine getrimek için fill metodu kullanılmaktadır. Datagridview kontrolüne dataset bağlanarak kayıtların görüntülenmesi sağlanmaktadır. En sonunda ise datagridview kontrolünün sütun ayarları yapılmaktadır.

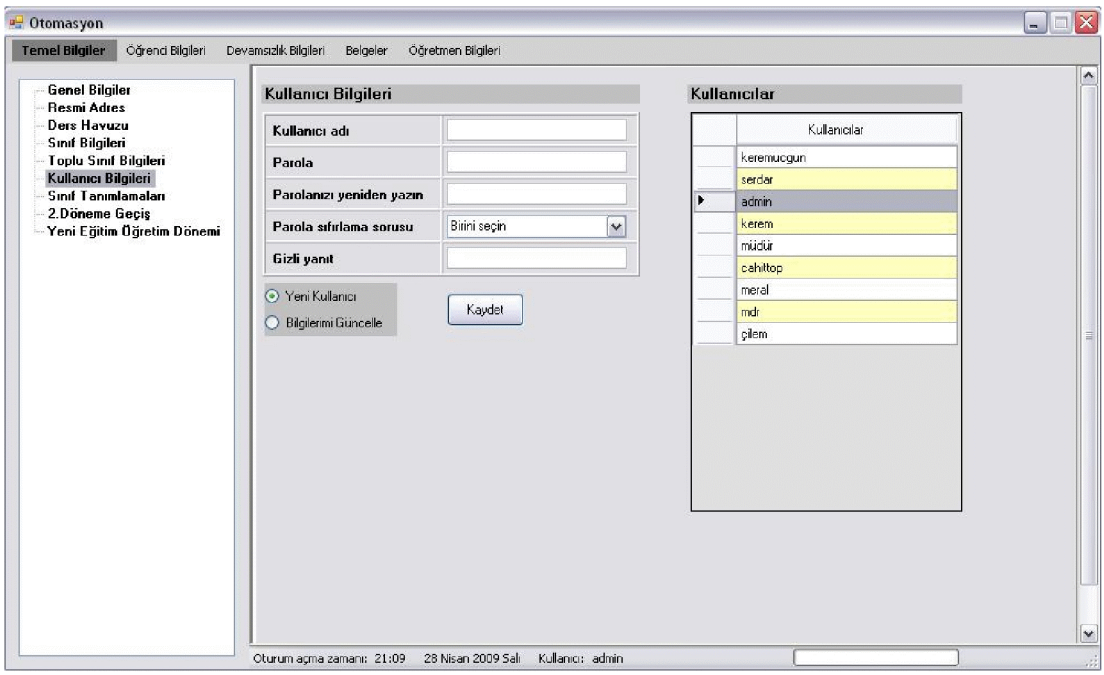
```
co.CommandText = "select * from sınıflar"
Dim adap As New OleDb.OleDbDataAdapter
adap.SelectCommand = co
Dim ds As New DataSet
adap.Fill(ds, "sınıflar")
DataGridView2.DataSource = ds.Tables("sınıflar")
DataGridView2.Columns(0).Visible = False
DataGridView2.Columns(1).Width = 60
DataGridView2.Columns(1).HeaderText = "Sınıflar"
DataGridView2.Columns(2).HeaderText = "Haftalık Ders Saati"
DataGridView2.Columns(2).Width = 80
```

Toplu Sınıf Bilgileri: *Toplu Sınıf Bilgileri* seçeneği seçildiğinde Şekil III.20'deki *Toplu Sınıf Bilgileri* ekranı gelir. Burada sınıf adları, sınıflarda bir haftada okutulan toplam ders saati, derslerin ayrı ayrı adları, ders kodları, bir haftada kaç saat okutuldukları ve sınıflarda okuyan öğrencilerin adı, soyadı, okul numarası bilgileri yer almaktadır. Ayrıca yazdır butonuna basıldığında Şekil III.20'deki öğrenciler bölümünün yazıcı çıktısı alınmaktadır.



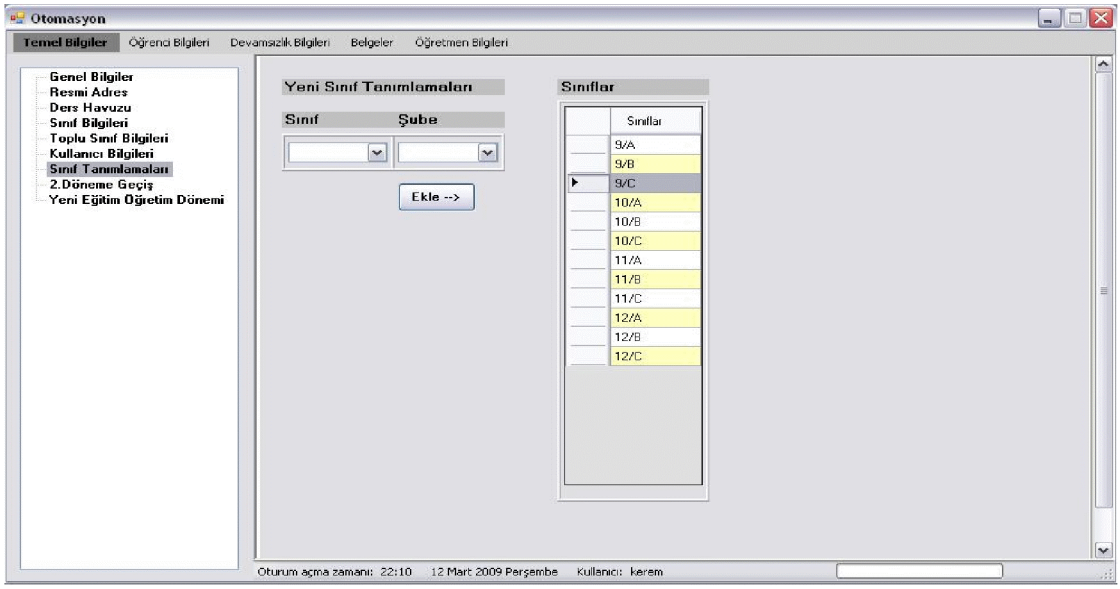
Şekil III.20 Temel Bilgiler menüsü altındaki Toplu Sınıf Bilgileri penceresinin görünümü

Kullanıcı Bilgileri: *Kullanıcı Bilgileri* seçeneği seçildiğinde Şekil III.21'deki *Kullanıcı Bilgileri ekranı* gelir. Burada yazılıma erişim izni verilecek kullanıcıların tanımlamaları yapılmakta ve mevcut kullanıcıların bilgilerinde değişiklik yapılabilmektedir. Gerekli bilgiler girildikten sonra kaydet butonuna basıldığında yeni kullanıcı bilgileri veri tabanında kullanıcılar tablosuna kaydedilmektedir.



Şekil III.21 Temel Bilgiler menüsü altındaki Kullanıcı Bilgileri penceresinin görünümü

Sınıf Tanımlamaları: *Sınıf Tanımlamaları* seçeneği seçildiğinde Şekil III.22'deki *Sınıf Tanımlamaları* ekranı gelir. Burada okuldaki bütün sınıf ve şube tanımlamaları yapılmaktadır. Sınıflar ve şubeleri seçildikten sonra kaydet butonuna basıldığında yeni sınıf isimleri veri tabanında sınıflar tablosuna kaydedilmektedir. Bundan sonra otomasyon yazılımındaki sınıf isimlerinin geçtiği bütün yerlerde sınıf isimleri sınıflar tablosundan alınmaktadır.



Şekil III.22 Temel Bilgiler menüsü altındaki Sınıf Tanımlamaları penceresinin görünümü

Aşağıdaki kodda, girilen sınıf ve şube ismi sınıflar tablosundaki sınıf alanına kaydedilmektedir.

```
Dim a1 As String
a1 = ComboBox2.Text & "/" & ComboBox3.Text
co.CommandText = "insert into sınıflar(sınıf)" & _
"values('" & a1 & "')"
Dim sonuc = co.ExecuteNonQuery
```

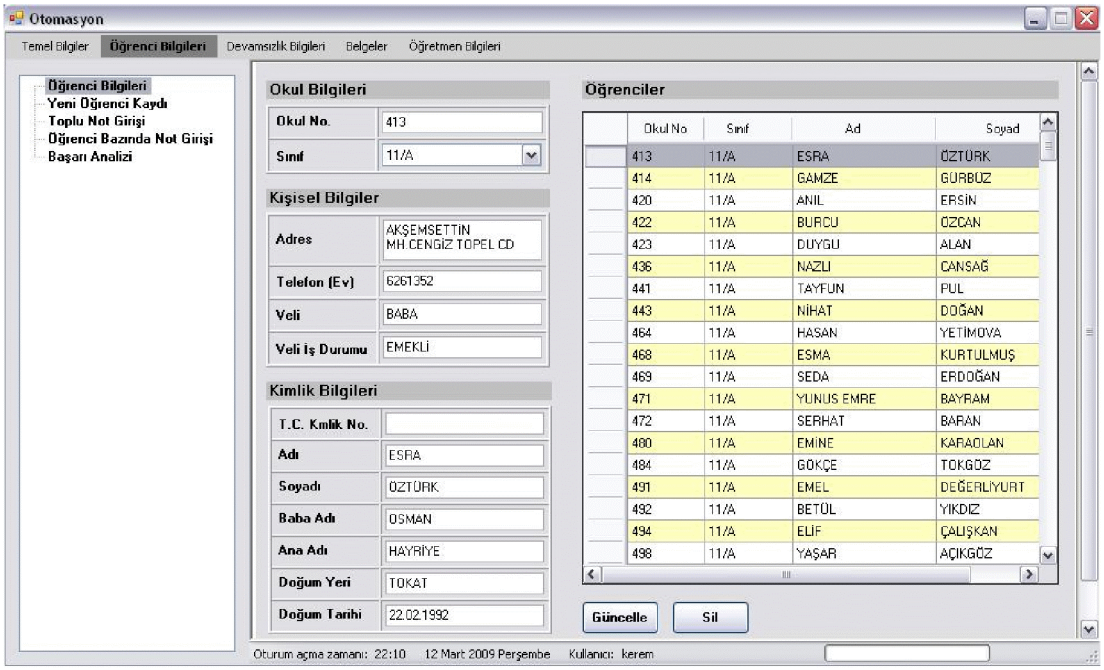
2.Döneme Geçiş: *2. Döneme Geçiş* seçeneği seçildiğinde eğitim öğretim dönemi ikinci dönem olarak ayarlanmaktadır. Böylece dersler, öğrenci notları ve devamsızlık bilgileri gibi bilgiler otomatik olarak 2.Dönem için yeniden düzenlenmektedir.

Yeni Eğitim Öğretim Dönemi: Bu seçenek seçildiğinde otomasyon programında yeni eğitim öğretim dönemi oluşturulmaktadır. Yeni girilen dönem ismi veri tabanındaki dönemler tablosuna kaydedilmektedir. Yeni eğitim öğretim dönemine geçiş ile dersler, öğrenci notları ve devamsızlık bilgileri gibi bilgiler otomatik olarak yeni dönem için yeniden düzenlenmektedir. Ayrıca öğrencilerin sınıf bilgileri değişmektedir ve öğrenciler bir üst sınıfa atlatılmaktadır. Öğrencilere ait ders bilgileri de değişip her öğrenciye bir üst sınıfta okutulan dersler atanmaktadır. Son sınıfta okuyan öğrenciler veri tabanına mezun adı altında kaydedilmektedir.

III.5.2.2. Öğrenci Bilgileri Menüsü

Öğrenci Bilgileri menüsü altında beş adet seçenek vardır. Bunlar öğrenci bilgileri, yeni öğrenci kaydı, toplu not girişi, öğrenci bazında not girişi ve başarı analizidir.

Öğrenci Bilgileri: *Öğrenci bilgileri* seçeneği seçildiğinde Şekil III.23'teki Öğrenci Bilgileri ekranı gelir. Bu ekran öğrencilerin okul bilgileri, kişisel bilgileri ve kimlik bilgileri gibi bilgilerin bulunduğu yerdir. Kullanıcı bu pencereden bilgileri güncelleyebilmekte veya silebilmektedir.

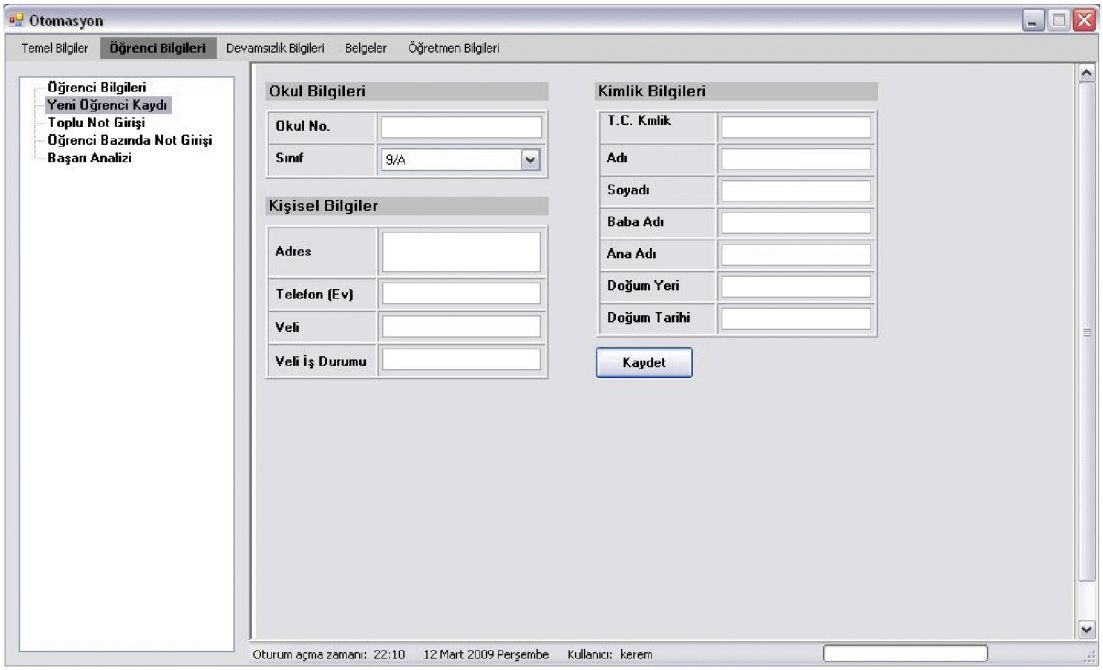


Şekil III.23 Öğrenci Bilgileri menüsü altındaki Öğrenci Bilgileri penceresinin görünümü

Güncelle butonuna yazılan kodlar genel bilgiler bölümündeki kaydetme kodları ile benzerlik gösterdiğinden burada tekrar yazılmamıştır. Aşağıdaki kodlar ise seçilen öğrencinin tüm bilgilerini ögrenci_bilgileri tablosundan silme işlemini gerçekleştirmektedir.

```
Dim gec As Integer
Dim x As Integer = DataGrid8.CurrentRowIndex
Dim sil As Integer = DataGrid8.Item(x, 0)
co.CommandText = "delete from ogrenci_bilgileri where id = " & sil
Dim sonuc = co.ExecuteNonQuery
If sonuc >= 0 Then MsgBox("Silme İşlemi Gerçekleşti")
```

Yeni Öğrenci Kaydı: *Yeni Öğrenci Kaydı* seçeneği seçildiğinde Şekil III.24'teki *Yeni Öğrenci Kaydı* ekranı gelir. Okula yeni öğrenci kaydı yapmak için kullanılır. Öğrencilerin okul bilgileri, kişisel bilgileri ve kimlik bilgileri gibi bilgilerin veri tabanındaki ögrenci_bilgileri tablosuna kaydedildiği penceredir.



Şekil III.24 Öğrenci Bilgileri menüsü altındaki Yeni Öğrenci Kaydı penceresinin görünümü

Yeni öğrenci kaydı yapıldığında bu öğrenci için devam tablosunda öğrencinin devamsızlık bilgilerini tutmak için yeni bir kayıt oluşturulur. Ayrıca notlar tablosunda öğrencinin ders notlarını tutmak amacıyla o öğrenciye ait yeni bir kayıt oluşturulur.

Aşağıdaki kodda devam tablosunda sadece bir ay için devamsızlık bilgileri kaydı oluşturulmaktadır. Okullar bir eğitim öğretim döneminde on ay açık kaldığı için bu kodlar on ay için ayrı ayrı yazılmıştır.

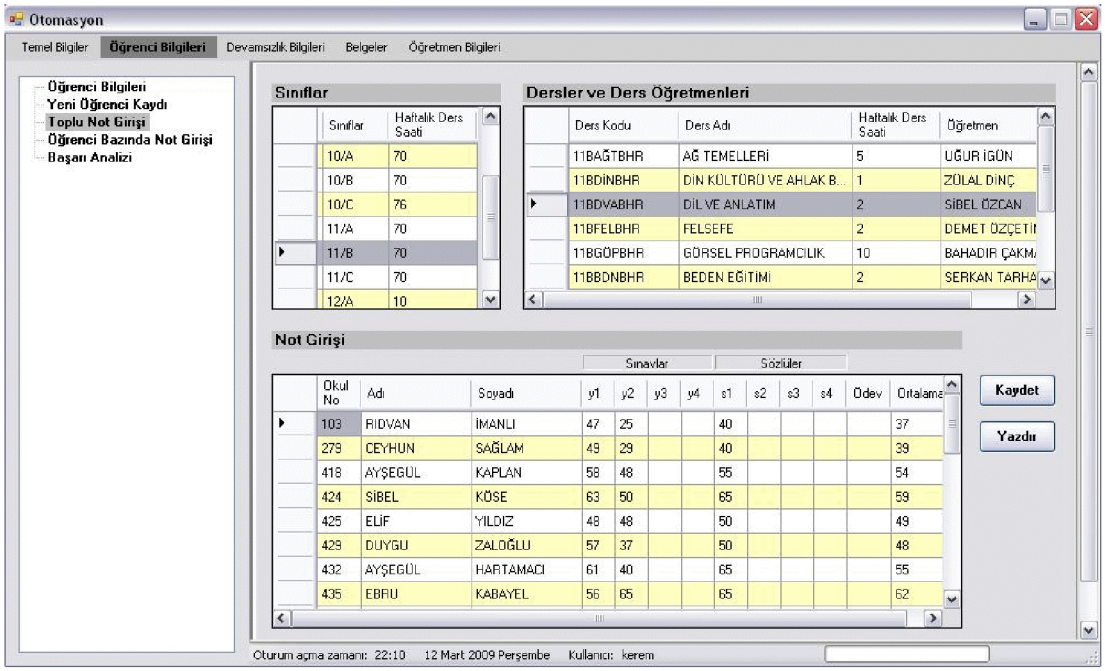
```
co2.CommandText = "insert into devam(okul_no, aylar,adı,soyadı,donem)
values ('" & a1 & "','" & ay1 & "','" & a8 & "','" & a9 & "','" &donem_adı& "')"
Dim sonuc2 = co2.ExecuteNonQuery
```

Aşağıdaki kod ise notlar tablosuna öğrencinin ders notları bilgileri için yeni bir kayıt oluşturulmaktadır.

```
For i = 0 To ders_sayısı - 1
    conot.CommandText = "insert into
    notlar(ders_kodu,okul_no,adı,soyadı,ders_adı,donem)" & _"values ('" &
    DataGrid3.Item(i, 1) & "','" & TextBox19.Text & "','" & TextBox26.Text &
    "','" & TextBox25.Text & "','" & DataGrid3.Item(i, 2) & "','" & donem& "')"
    Dim sonucnot = conot.ExecuteNonQuery
```

```
Next
If sonuc >= 0 Then MsgBox("Kaydetme İşleminiz Gerçekleşti")
```

Toplu Not Girişi: *Toplu Not Girişi* seçeneği seçildiğinde Şekil III.25'teki *Toplu Not Girişi* ekranı gelir. Bu ekranda, seçilen derse ait sınav, sözlü ve ödev notları, tüm öğrencilere toplu bir şekilde girilmektedir. Ders bazında not girme işlemi gerçekleşmektedir.



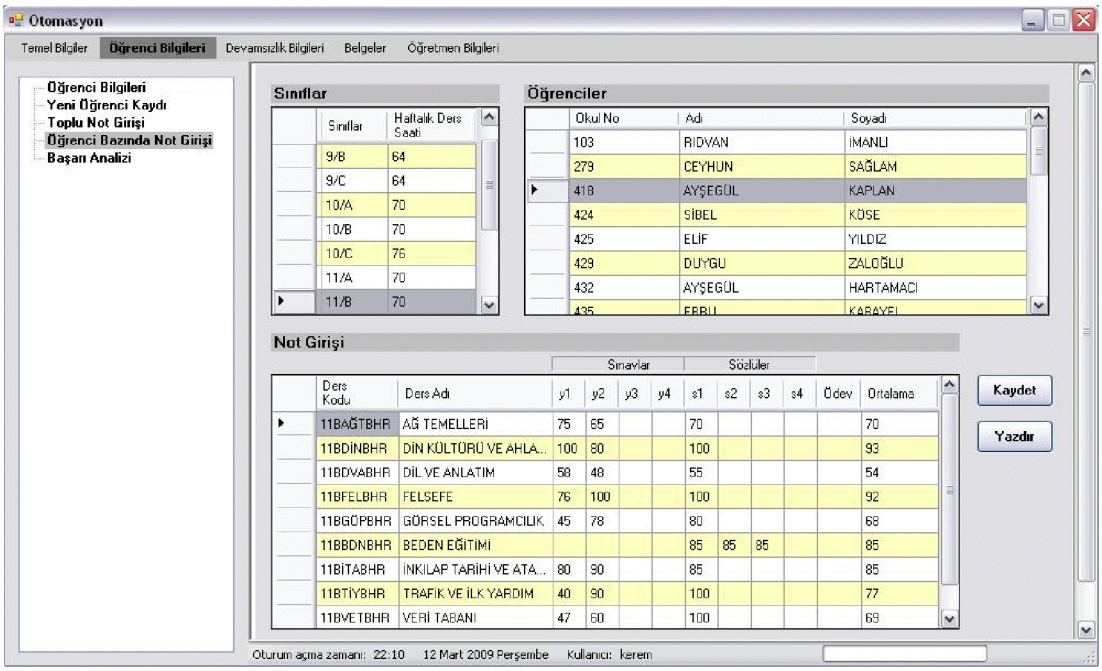
Şekil III.25 Öğrenci Bilgileri menüsü altındaki Toplu Not Girişi penceresinin görünümü

Sınıflar bölümünden not giriş yapılmak istenen sınıf seçildikten sonra *Dersler ve Ders Öğretmenleri* bölümüne seçilen sınıfta okutulan dersler gelmektedir. Buradan da hangi derse ait not girişi yapılacak ise o ders seçildiğinde, *Not Girişi* bölümüne seçilen sınıfta okuyan öğrenciler gelmektedir. Her öğrenciye dört adet yazılı notu, dört adet sözlü notu ve bir adet ödev notu girilebilmektedir. Not girişi yapıldığı esnada girilen notların ortalaması aynı anda ortalama alanında gösterilmektedir. Not girme işleminden sonra kaydet butonuna basıldığında tüm öğrencilerin notları veri tabanındaki notlar tablosuna kaydedilmektedir. Yazdır butonuna basıldığında tüm öğrencilerin seçilen dersten aldıkları tüm notların, yazıcıdan çıktısı alınmaktadır.

Aşağıdaki kod, not girişi alanına notlar girildiği anda, girilen notların ortalamasının ortalama alanında gösterilmesi için yazılmıştır.

```
Dim say, top, sayac As Integer
For say = 3 To 11
    If DataGridView5.Item(say,
    DataGridView5.CurrentRow.Index).Value.ToString <> "" Then
        sayac = sayac + 1
        top = top + DataGridView5.Item(say,
        DataGridView5.CurrentRow.Index).Value
    End If
Next
If sayac <> 0 Then
    DataGridView5.Item(12, DataGridView5.CurrentRow.Index).Value =
    top / sayac
End If
```

Öğrenci Bazında Not Girişi: *Öğrenci Bazında Not Girişi* seçeneği seçildiğinde Şekil III.26'daki *Öğrenci Bazında Not Girişi* ekranı gelir. Bu ekranda seçilen öğrenciye ait tüm derslerin sınav, sözlü ve ödev notları girilmektedir. Öğrenci bazında not girme işlemi gerçekleşmektedir.



Şekil III.26 Öğrenci Bilgileri menüsü altındaki Öğrenci Bazında Not Girişi penceresinin görünümü

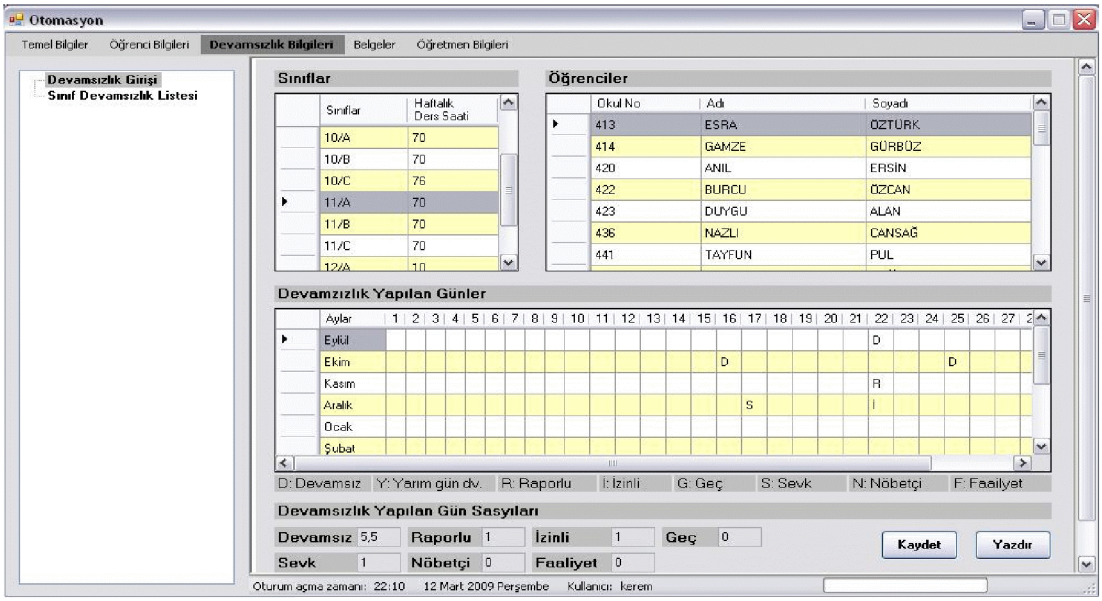
Sınıflar bölümünden not giriş yapılmak istenen sınıf seçildikten sonra *Öğrenciler* bölümüne seçilen sınıftaki öğrenciler gelmektedir. Buradan da hangi öğrencinin not girişi yapılacak ise o öğrenci seçildiğinde, *Not Girişi* bölümüne seçilen öğrencinin dersleri gelmektedir. Her derse ait dört adet yazılı notu, dört adet sözlü notu ve bir adet ödev notu girilebilmektedir. Not girişi yapıldığı esnada girilen notların ortalaması aynı anda ortalama alanında gösterilmektedir. Not girme işleminden sonra kaydet butonuna basıldığında öğrencinin notları veri tabanındaki notlar tablosuna kaydedilmektedir. Yazdır butonuna basıldığında seçilen öğrencinin tüm derslerden aldığı notların yazıcıdan çıktısı alınmaktadır.

Başarı Analizi: Bu bölüm öğrenci verileri üzerine veri madenciliğinin uygulandığı aşamadır. Bölüm IV'te detaylı olarak anlatılacaktır.

III.5.2.3. Devamsızlık Bilgileri Menüsü

Devamsızlık Bilgileri menüsü altında iki adet seçenek vardır. Bunlar devamsızlık girişi ve sınıf devamsızlık listesidir.

Devamsızlık Girişi: *Devamsızlık Girişi* seçeneği seçildiğinde Şekil III.27'deki *Devamsızlık Girişi* ekranı gelir. Öğrencilerin okula devam durumlarının kayıtlarını tutmak için kullanılır. Öğrencilerin devamsızlık durumları eğitim-öğretim döneminin başladığı Eylül ayından eğitim-öğretim döneminin sona erdiği Haziran ayına kadar her gün için ayrı ayrı tutulup veri tabanındaki devam tablosuna kaydedilmektedir.



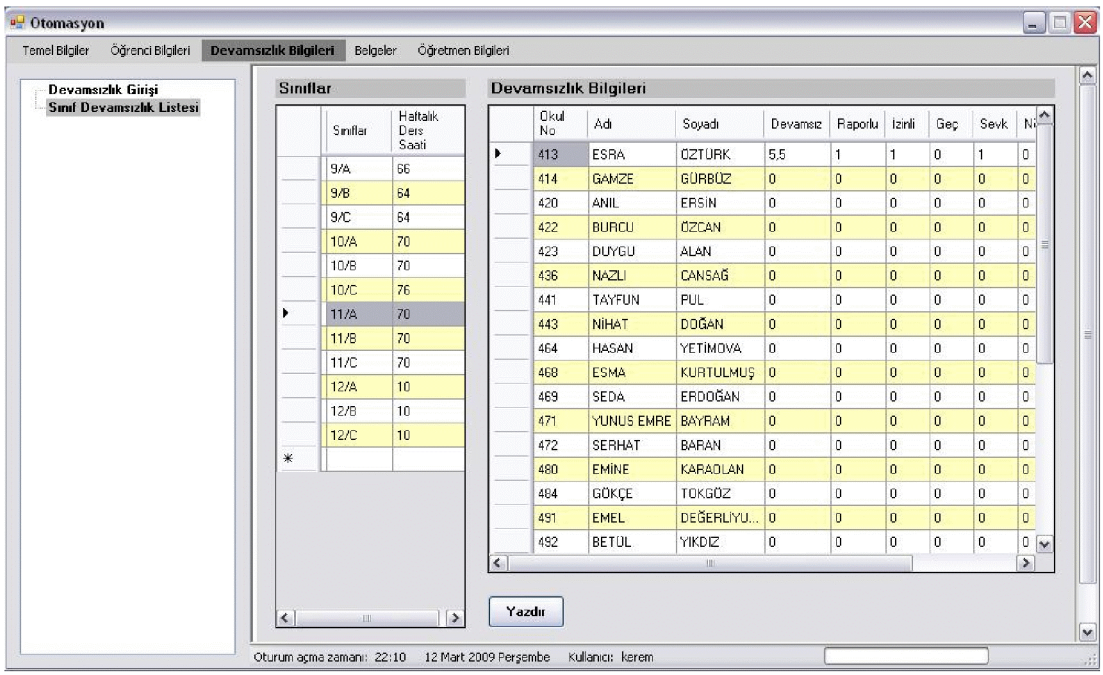
Şekil III.27 Devamsızlık Bilgileri menüsü altındaki Devamsızlık Girişi penceresinin görünümü

Sınıflar bölümünden devamsızlık bilgileri girilmek istenen sınıf seçildikten sonra *Öğrenciler* bölümüne seçilen sınıftaki öğrenciler gelmektedir. Buradan da hangi öğrencinin devam-devamsızlık bilgisi girilecek ise o öğrenci seçildiğinde, D*evamsızlık Yapılan Günler* bölümüne seçilen öğrencinin Eylül-Haziran ayları arasındaki günlük devamsızlık tablosu gelmektedir. Bu tablo üzerine, devam durumuna göre, devam durumunu belirten harfler yazılarak veri tabanındaki devam tablosuna kaydedilmektedir. Ayrıca öğrencinin toplam yapmış olduğu devamsızlık bilgileri de gösterilmektedir. Yazdır butonuna basıldığında seçilen öğrencinin tüm devamsızlık bilgilerinin yazıcı çıktısı alınmaktadır.

Aşağıdaki kodlar ile datagridview nesnesi taranarak öğrencinin toplam devamsızlık durumu hesaplanmakta ve ilgili alanlara yazılmaktadır.

```
Dim sayac1, sayac2 As Integer
For sayac1 = 0 To 9
   For sayac2 = 0 To 30
   Select Case DataGridView3.Item(sayac2, sayac1).Value.ToString
   Case "d", "D" : TextBox2.Text = TextBox2.Text + 1
   Case "y", "Y" : TextBox2.Text = TextBox2.Text + 0.5
   Case "r", "R" : TextBox4.Text = TextBox4.Text + 1
   Case "s", "S" : TextBox3.Text = TextBox3.Text + 1
   Case "n", "N" : TextBox5.Text = TextBox5.Text + 1
   Case "i", "İ" : TextBox6.Text = TextBox6.Text + 1
   Case "f", "F" : TextBox7.Text = TextBox7.Text + 1
   Case "g", "G" : TextBox8.Text = TextBox8.Text + 1
   End Select
   Next
Next
```

Sınıf Devamsızlık Listesi: *Sınıf Devamsızlık Listesi* seçeneği seçildiğinde Şekil III.28'deki *Sınıf Devamsızlık Listesi* ekranı gelir. Seçilen sınıftaki öğrencilerin toplam devamsızlık durumlarının bir listesi ekrana çıkar. Yazdır butonuna basıldığında seçilen sınıftaki tüm öğrencilerin toplam devamsızlık durumlarının ayrıntılı olarak yazıcı çıktısı alınmaktadır.

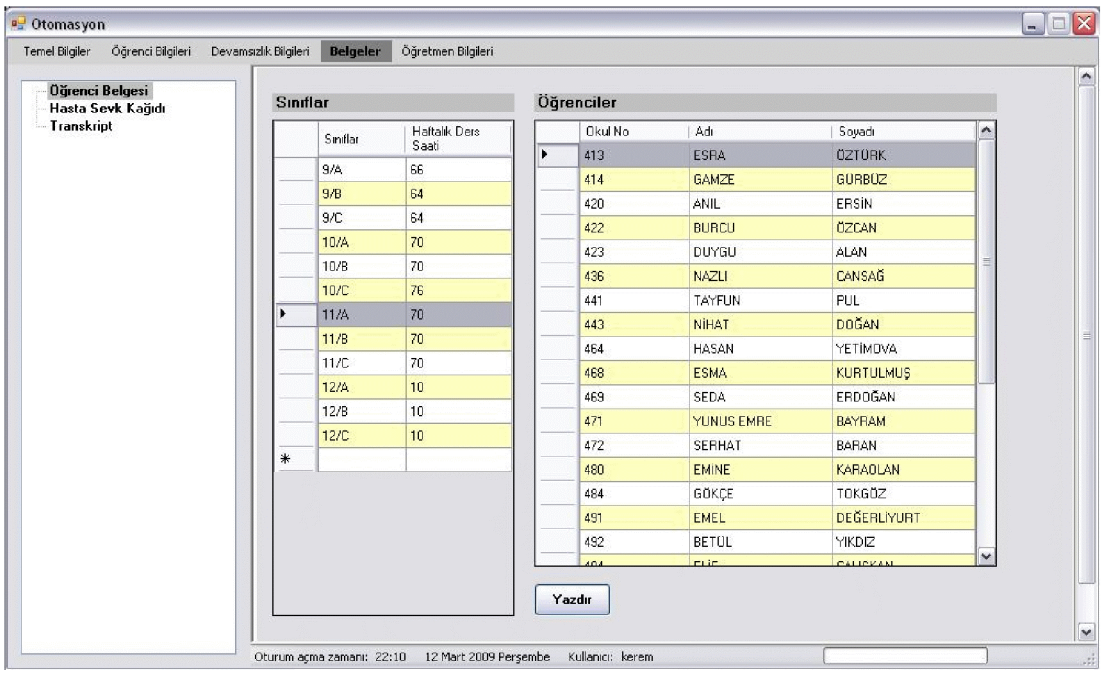


Şekil III.28 Devamsızlık Bilgileri menüsü altındaki Sınıf Devamsızlık Listesi penceresinin görünümü

III.5.2.4. Belgeler Menüsü

Belgeler menüsü altında üç adet seçenek vardır. Bunlar öğrenci belgesi, hasta sevk kağıdı ve transkripttir.

Öğrenci Belgesi: *Öğrenci Belgesi* seçeneği seçildiğinde Şekil III.29'daki *Öğrenci Belgesi* ekranı gelir. Seçilen sınıftaki öğrencilerin listesi ekrana çıkar. Yazdır butonuna basıldığında seçili öğrenciye ait bilgiler veri tabanından alınarak öğrenci belgesinin yazıcı çıktısı alınmaktadır.



Şekil III.29 Belgeler menüsü altındaki Öğrenci Belgesi, Hasta Sevk Kağıdı, Transkript penceresinin görünümü

Aşağıdaki kod yazdır butonuna basıldığında öğrenci belgesinin yazıcı çıktısını almak için kullanılmaktadır. Öncelikli olarak yeni yazı fontları tanımlanmakta ve daha sonra yazdırılmak istenen bilgiler kağıt üzerindeki koordinatları belirtilerek istenilen yere yazdırılmaktadır.

```
Dim yazi1 As Font = New Font("tahoma", 14, FontStyle.Bold)
Dim yazi2 As Font = New Font("tahoma", 14, FontStyle.Regular)
Dim trans As Font = New Font("tahoma", 10, FontStyle.Regular)
Dim trans2 As Font = New Font("tahoma", 10, FontStyle.Bold)
If TextBox1.Text = "Öğrenci Belgesi" Then
e.Graphics.DrawString("ÖĞRENCİ BELGESİ", yazi1, Brushes.Black, 320,
200)
e.Graphics.DrawString(DateTime.Today, yazi1, Brushes.Black, 600, 300)
e.Graphics.DrawString("T.C. Kimlik No", yazi2, Brushes.Black, 100, 400)
e.Graphics.DrawString("Öğrencinin Adı Soyadı", yazi2, Brushes.Black, 100,
430)
e.Graphics.DrawString("Baba Adı", yazi2, Brushes.Black, 100, 460)
e.Graphics.DrawString("Anne Adı", yazi2, Brushes.Black, 100, 490)
e.Graphics.DrawString("Doğum Yeri", yazi2, Brushes.Black, 100, 520)
```

```
e.Graphics.DrawString("Doğum Tarihi", yazi2, Brushes.Black, 100, 550)
e.Graphics.DrawString("Okulu", yazi2, Brushes.Black, 100, 580)
e.Graphics.DrawString("Sınıfı", yazi2, Brushes.Black, 100, 610)
e.Graphics.DrawString("Numarası", yazi2, Brushes.Black, 100, 640)
e.Graphics.DrawString(tc, yazi2, Brushes.Black, 330, 400)
e.Graphics.DrawString(ad & " " & soyad, yazi2, Brushes.Black, 330, 430)
e.Graphics.DrawString(baba, yazi2, Brushes.Black, 330, 460)
e.Graphics.DrawString(anne, yazi2, Brushes.Black, 330, 490)
e.Graphics.DrawString(dogum_yeri, yazi2, Brushes.Black, 330, 520)
e.Graphics.DrawString(dogum_tarihi, yazi2, Brushes.Black, 330, 550)
e.Graphics.DrawString(okul, yazi2, Brushes.Black, 330, 580)
e.Graphics.DrawString(sınıf, yazi2, Brushes.Black, 330, 610)
e.Graphics.DrawString(okul_no, yazi2, Brushes.Black, 330, 640)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 400)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 430)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 460)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 490)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 520)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 550)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 580)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 610)
e.Graphics.DrawString(":", yazi2, Brushes.Black, 320, 640)
e.Graphics.DrawString("Yukarıda açık kimliği yazılı olan " & ad & " " &
soyad & " okulumuzun " & sınıf & " sınıfında", yazi2, Brushes.Black, 130,
700)
e.Graphics.DrawString(okul_no & " numara ile kayıtlı olup halen okulumuzun
devamlı öğrencisidir.", yazi2, Brushes.Black, 100, 730)
e.Graphics.DrawString("İş bu belge kendisinin isteği üzerine düzenlenerek
kendisine verilmiştir.", yazi2, Brushes.Black, 100, 780)
e.Graphics.DrawString(mudur, yazi2, Brushes.Black, 550, 900)
```

Hasta Sevk Kağıdı: *Hasta Sevk Kağıdı* seçeneği seçildiğinde (bkz: Şekil III.29) *Hasta Sevk Kağıdı* ekranı gelir. Seçilen sınıftaki öğrencilerin listesi ekrana çıkar. Yazdır butonuna basıldığında seçili öğrenciye ait bilgiler veri tabanından alınarak

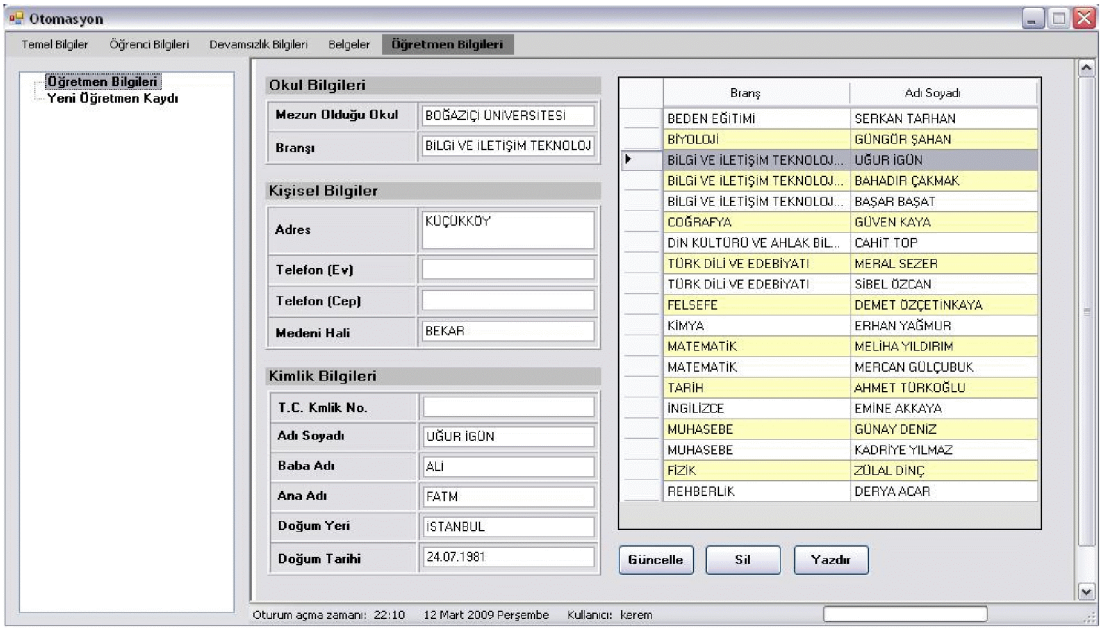
Hasta Sevk Kağıdı'nın yazıcı çıktısı alınmaktadır. Yazdır butonuna yazılan kodlar Öğrenci Belgesindeki kodlarla benzerlik göstermektedir.

Transkript: *Transkript* seçeneği seçildiğinde (bkz: Şekil III.29) *Transkript* ekranı gelir. Seçilen sınıftaki öğrencilerin listesi ekrana çıkar. Yazdır butonuna basıldığında seçili öğrenciye ait bilgiler veri tabanından alınarak Transkriptin yazıcı çıktısı alınmaktadır. Yazdır butonuna yazılan kodlar *Öğrenci Belgesi*'ndeki kodlarla benzerlik göstermektedir.

III.5.2.5. Öğretmen Bilgileri Menüsü

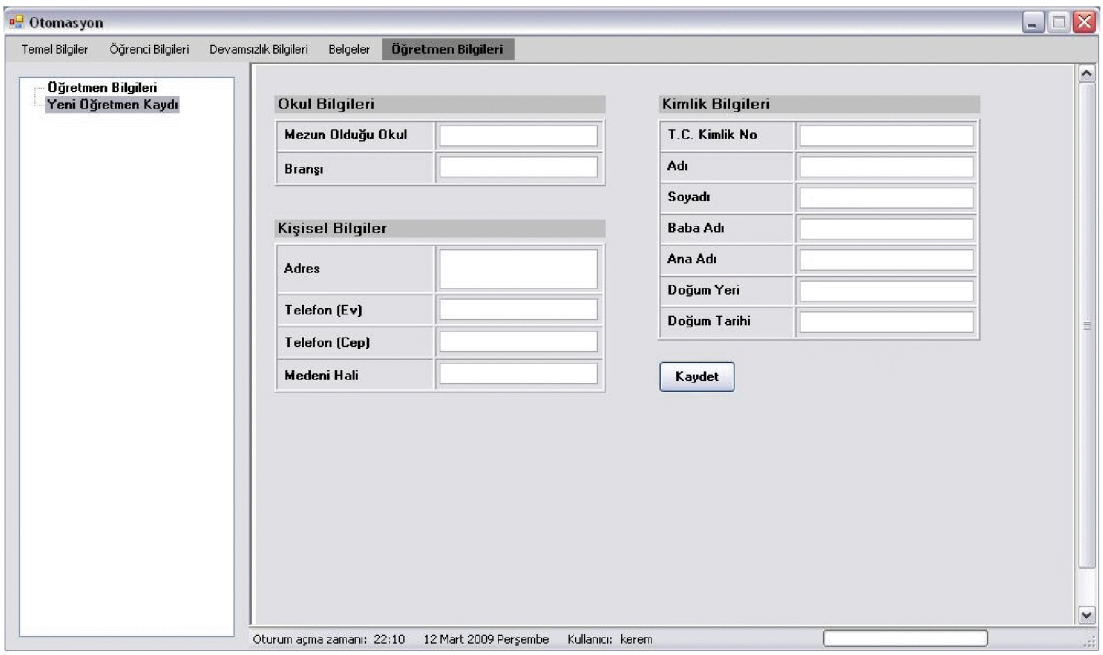
Öğretmen Bilgileri menüsü altında iki adet seçenek vardır. Bunlar öğretmen bilgileri ve yeni öğretmen kaydıdır.

Öğretmen Bilgileri: *Öğretmen bilgileri* seçeneği seçildiğinde Şekil III.30'daki *Öğretmen Bilgileri* ekranı gelir. Bu ekran öğretmenlerin okul bilgileri, kişisel bilgileri ve kimlik bilgileri gibi bilgilerin bulunduğu bölümdür. Kullanıcı bu pencereden bilgileri güncelleyebilmekte, silebilmekte ve yazıcı çıktısı alabilmektedir.



Şekil III.30 Öğretmen Bilgileri menüsü altındaki Öğretmen Bilgileri penceresinin görünümü

Yeni Öğretmen Kaydı: *Yeni Öğretmen Kaydı* seçeneği seçildiğinde Şekil III.31'deki *Yeni Öğretmen Kaydı* ekranı gelir. Okula yeni atanan öğretmenlerin kaydını yapmak için kullanılır. Öğretmenlerin okul bilgileri, kişisel bilgileri ve kimlik bilgileri gibi bilgilerin veri tabanındaki öğretmen_bilgileri tablosuna kaydedildiği penceredir.



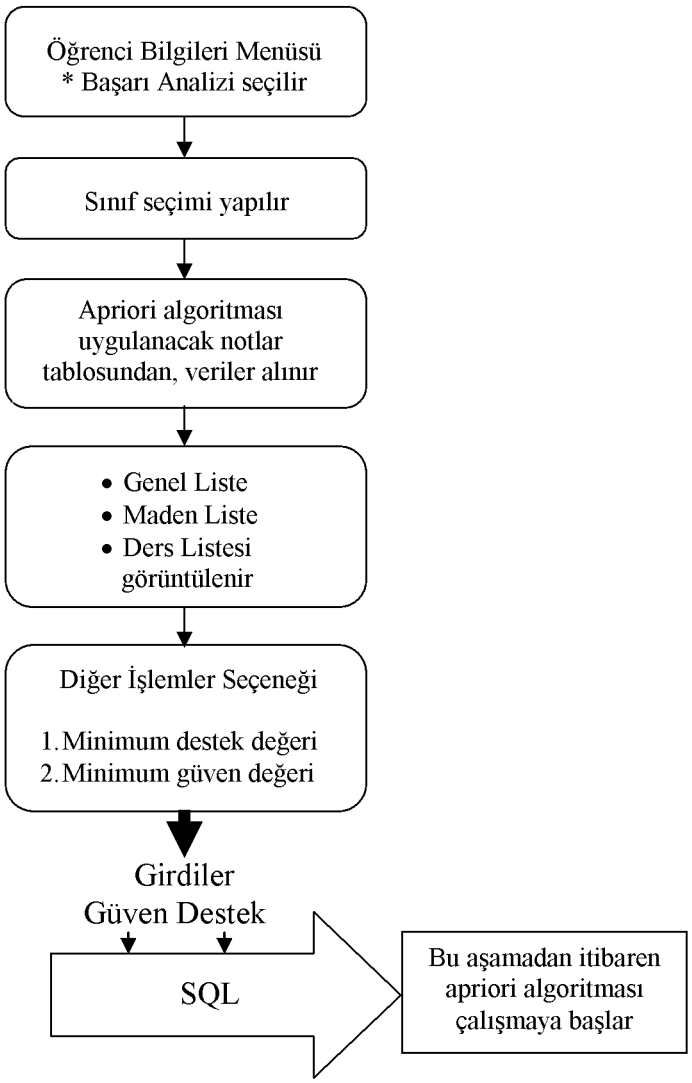
Şekil III.31 Öğretmen Bilgileri menüsü altındaki Yeni Öğretmen Kaydı penceresinin görünümü

BÖLÜM IV

UYGULAMA ve SONUÇLAR

Bu bölümün amacı, birliktelik-ilişki kuralı algoritmalarından Apriori algoritması hakkında genel bilgi vermek, uygulama yazılımının başarı analizi bölümünün çalışma mantığını incelemek ve bir orta öğretim kurumunun verileri üzerine Apriori algoritması uygulayarak elde edilen sonuçları değerlendirmektir.

Şekil IV.1'de uygulama yazılımın başarı analizi bölümünün genel işleyişi görülmektedir.



Şekil IV.1 Başarı Analizi bölümünün aşamaları

IV.1. APRIORI ALGORİTMASI

Apriori algoritması birliktelik kuralları çıkarımında en çok bilinen algoritmadır. Algoritmanın ismi, bilgileri bir önceki adımdan aldığı için "prior" anlamında Apriori'dir. Apriori algoritmasındaki temel yaklaşım, eğer k-öğe kümesi (k adet elemana sahip öğe kümesi) minimum destek ölçütünü sağlıyor ise, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar.

Birliktelik kuralı madenciliği, tüm sık geçen öğelerin bulunması ve sık geçen bu öğelerden güçlü birliktelik kurallarının üretilmesi olmak üzere iki aşamalıdır. Kullanılan pazar sepeti verisinde her harekette yer alan ürün kodları sayısaldır ve ürün kodları küçükten büyüğe doğru sıralıdır. Öğe kümeleri eleman sayıları ile birlikte anılır ve k adet ürüne sahip bir öğe kümesi, k-öğe kümesi olarak isimlendirilir. k-öğe kümesi c ile ifade edilirse, öğeleri (ürünler) c[1], c[2], c[3],...,c[k] şeklinde gösterilir ve c[1] < c[2] < c[3]< ... <c[k] olacak şekilde küçükten büyüğe doğru sıralıdır [19]. Her öğe kümesine destek ölçütünü tutmak üzere bir sayaç değişkeni eklenmiştir ve sayaç değişkeni öğe kümesi ilk kez oluşturulduğunda sıfırlanır. Sık geçen (geniş) öğe kümeleri L karakteri ile gösterilir, aday öğe kümeleri ise C karakteri ile gösterilir [4].

Algoritmada kullanılan değişkenler Tablo IV.1'de özetlenmiştir.

Tablo IV.1 Algoritmada kullanılan değişkenler [11]

k-öğe kümesi (k-itemset)	**k adet öğe içeren öğe kümesi**
$\mathbf{L_k}$	Minimum destek şartını sağlayan sık geçen k- öğe küme setleri. Bu setlerin her bir elemanının iki alanı vardır: i) öğe kümesi ve ii) destek sayacı
$\mathbf{C_k}$	Aday k-öğe küme setleri. Bu setlerin her bir elemanının iki alanı vardır: i) öğe kümesi ve ii) destek sayacı

Şekil IV.2'de Apriori algoritmasının klasik özet kodu görülmektedir. Bu şekilde yer alan apriori-gen işlevi, (k-1) adet öğeye sahip L_{k-1} öğeler kümesini kullanarak k adet öğeye sahip aday kümeleri üretir.

L_1= {sık geçen 1-öğe kümesi};
for (k=2; $L_{k-1} \neq \emptyset$; k++) **do begin**
C_k=**apriori-gen** (L_{k-1}); // Yeni adaylar
forall transactions-hareketler $t \in D$ **do begin**
C_t = **subset** (C_k, t); // Adaylar t içindedir
forall candidates – adaylar $c \in C_t$
c.**count**++;
end
L_k = {$c \in C_k$ | c.**count** ≥ **minsup**}
end
Answer = $\cup_k L_k$;

Şekil IV.2 Klasik Apriori algoritması özet kodu [11]

Bu işlevin çalışma şekli şöyledir; ilk önce, L_{k-1} sık geçen öğe kümesi ile L_{k-1} *birleştirme (join)* işlemine tabi tutulur. Birleştirme işleminde L_{k-1} sık geçen öğe kümesinin her satırında yer alan son öğe haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öğe haricinde diğer öğelerle yakalanan benzerliklerden yeni aday öğe kümeleri oluşturulur. Oluşan kümeler *budama (prune)* adımı ile budanır. Budama işleminde ise; c aday kümesinin (k-1) öğeye sahip alt kümelerinden L_{k-1} 'de yer almayan kümeler silinir. Apriori-gen işlevinin algoritma kesiti Şekil IV.3'te gösterilmektedir [19].

insert into C_k
select p.**items**$_1$, p.**items**$_2$,.....,p.**items**$_{k-1}$, q.**item**$_{k-1}$
from L_{k-1} p, L_{k-1} q
where p.**item**$_1$=q.**item**$_1$, ..., p.**item**$_{k-2}$=q.**item**$_{k-2}$,
p.**item**$_{k-1}$ < q.**item**$_{k-1}$;
forall itemsets $c \in C_k$ **do**
forall (k-1)-**subsets** s **of** c **do**
if ($s \notin L_{k-1}$) **then**
delete c **from** C_k

Şekil IV.3 Apriori-Gen işlevinin algoritma kesiti [19]

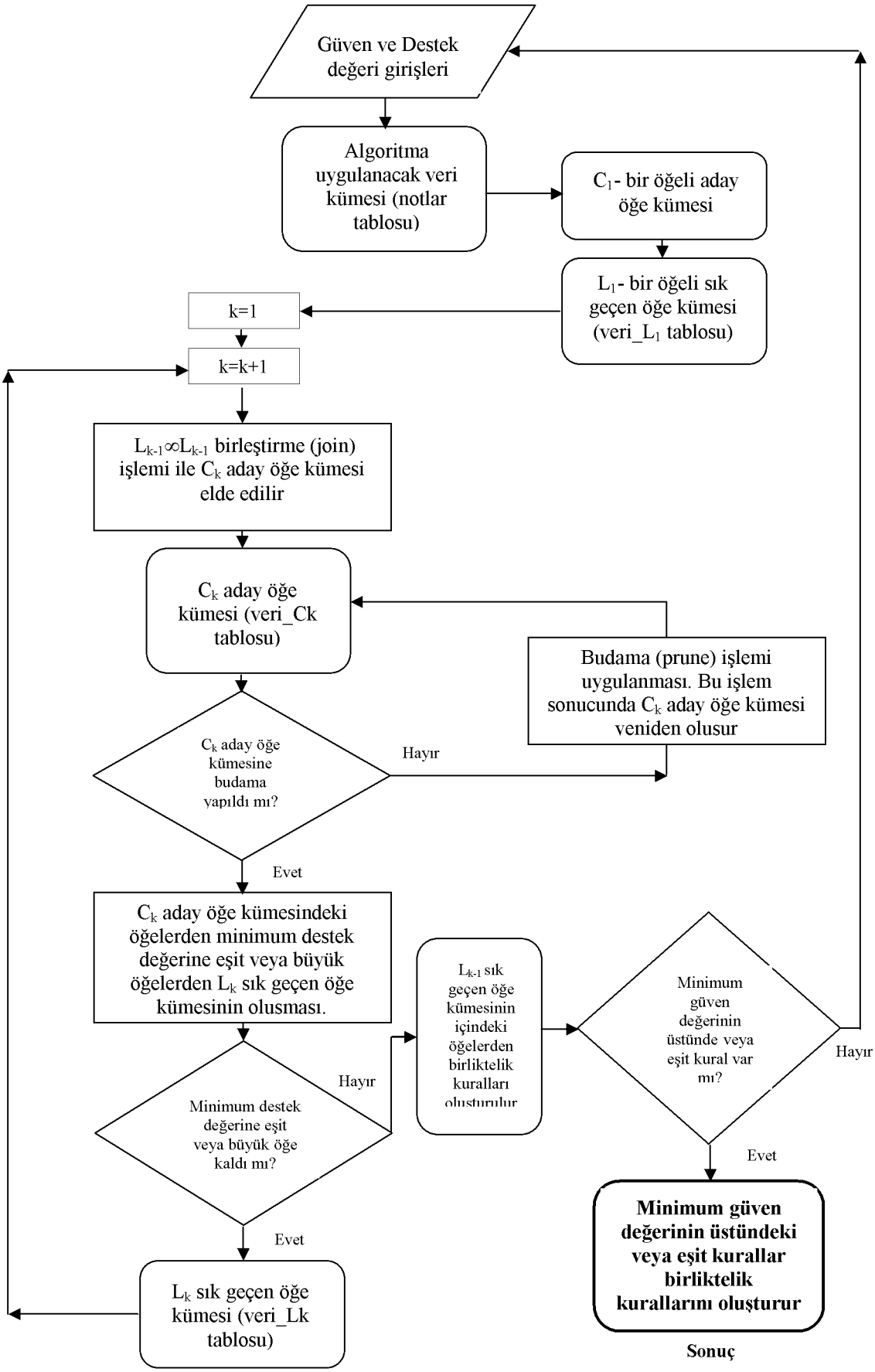
Budama işleminde; c aday kümesinin (k-1) öğeye sahip alt kümelerinden L_{k-1} sık geçen öğe kümesinde yer almayan tüm alt kümeler silinir. Diğer bir deyiş ile budama, C_k aday öğe kümesindeki öğelerin alt kümelerinin L_{k-1} sık geçen öğe kümesinde yer alıp almadığına bakılır, eğer bir öğenin alt kümelerinden biri, L_{k-1} sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_k aday öğe kümesinden silinir [19].

Apriori algoritması özet kodu incelendiğinde sık geçen öğe kümelerini bulmak için birçok kez veritabanının tarandığı görülmektedir. İlk aşamadan önce, veri madenciliği uygulanacak veri topluluğunun taranarak öğelerin kaç adet hareket kaydı içinde yer aldığı tespit edildiği (her öğe için tespit edilen bu değere destek sayacı adı verilir) ve destek sayacı minimum destek değerine eşit veya büyük olan öğelerin L_1 sık geçen 1-öğe kümesi olarak belirlendiği varsayılarak işleme başlanır [11].

Kod içinde kurulan döngü yapısı ile ilk aşamada L_1 sık geçen öğe kümesinin öğelerinin ikili kombinasyonuna benzer bir şekilde ($L_1 \infty L_1$) yeni bir küme oluşur, bu işleme *birleştirme (join)* adı verilir. Bu işlem ile oluşan kümelere de aday öğe kümeler adı verilir ve C harfi ile simgelenir. Oluşan bu aday öğe kümesinin her elemanı iki adet öğeden oluştuğu için C_2 ifadesi ile isimlendirilir. Bu aday küme apriori-gen işlevi ile budama işlemine tabi tutulur ve C_2 kümesinin elemanlarına ait alt kümelerinin L_1 öğe kümesinde olup olmadığına bakılır, alt kümelerden L_1 içinde yer almayan küme elemanları C_2 aday kümesinden silinir. Apriori algoritması uygulanan veri topluluğu tekrar taranarak budama işleminden geçen C_2 aday kümesi elemanlarının kaç adet hareket kaydı içinden geçtiği (destek sayacı) bulunur. Bulunan destek sayaç bilgileri doğrultusunda C_2 aday kümesi elemanlarının destek sayacı minimum destek değerine eşit veya büyük destek değerine sahip olan elemanları L_2 sık geçen öğe kümesini oluşturur [11].

Döngü bir sonraki aşamada L_2 kümesi öğelerinin üçlü kombinasyonu ile yeni bir aday öğe kümesi oluşturur ve bu küme C_3 ifadesi ile simgelenir. İlk aşamada olduğu gibi bu küme de budama işleminden geçer ve budama işleminden sonra minimum destek seviyesinin üstünde kalan elemanları ile L_3 sık geçen öğe kümesi oluşturulur. Döngü her dönüşünde öğe sayısını artırarak devam eder. Bu süreç yeni bir sık geçen öğe kümesi bulunamayana kadar sürer [11].

Bu çalışmada apriori algoritması ile veri madenciliği işlemini gerçekleştirmek için hazırlanan programın, başarı analizi bölümünün (veri madenciliği işlemlerinin uygulandığı bölüm) çalışma aşamaları Şekil IV.4'te görülmektedir.



Şekil IV.4 Yazılımda kullanılan Apriori algoritmasının aşamaları

IV.2. OTOMASYON YAZILIMININ BAŞARI ANALİZİ BÖLÜMÜNÜN TASARIMI

Yazılımın başarı analizi bölümünde öğrenci verileri üzerine veri madenciliği işlemleri uygulanmaktadır. Bu bölümde sınıflar bazında işlemler yapılmaktadır. Veri madenciliği yapılacak olan sınıf seçildikten sonra veri madenciliği aşamaları başlamaktadır. Bu aşamalar ve gerçekleşen işlemlerin açıklamaları aşağıda verilmektedir.

IV.2.1. Genel Liste

Seçilen sınıfa ait öğrencilerin okumuş oldukları süre içinde görmüş oldukları derslerden almış oldukları tüm notlar gösterilmektedir. Verilerin eleme süreci öncesindeki ham veri hali gösterilmektedir. Şekil IV.5'te genel listenin bir kesiti verilmektedir. Burada öğrencilerin okul numaraları, okumuş oldukları süre içindeki gördükleri tüm derslerin kodları, ders adları ve bu derslerden almış oldukları notlar görülmektedir.

OKUL NO	DERS KODU	DERS ADI	SENE SONU NOTU
413	11AİTA	İNKILAP TARİHİ VE ATATÜRKÇÜLÜK	56
413	11AİTABHR	İNKILAP TARİHİ VE ATATÜRKÇÜLÜK	88
413	11ATİY	TRAFİK VE İLK YARDIM	85
413	11ATİYBHR	TRAFİK VE İLK YARDIM	100
413	11AVET	VERİ TABANI	86
413	11AVETBHR	VERİ TABANI	98
414	9ABDN	BEDEN EĞİTİMİ	83
414	9ABDNBHR	BEDEN EĞİTİMİ	77
414	9ABYL	BİYOLOJİ	81
414	9ABYLBHR	BİYOLOJİ	66
414	9ABİT	BİLGİ VE İLETİŞİM TEKNOLOJİLERİ	47
414	9ABİTBHR	BİLGİ VE İLETİŞİM TEKNOLOJİLERİ	75

Şekil IV.5 Genel Liste ekranından bir kesit

IV.2.2. Maden Liste

Seçilen sınıfa ait öğrencilerin okumuş oldukları süre içinde görmüş oldukları derslerden, sadece başarısız oldukları derslerin notları gösterilmektedir. Öğrenciler 45 'ten daha düşük bir puan aldıklarında başarısız sayılmaktadırlar. Maden liste ekranında veri tabanında bulunmakta olan veriler bir süzgeçten geçirilerek ön eleme işlemi yapılmaktadır. Bu ön eleme işlemi sonucunda, birliktelik kurallarını oluşturma esnasında kullanılacak veriler elde edilmektedir. Bu verilerin elde edilme

aşamasında öğrencinin ortalama yükseltme ve sorumluluk sınavlarından almış olduğu notlar verinin güvenilirliğini azaltacağından dikkate alınamamıştır. Şekil IV.6'da maden listenin bir kesiti verilmektedir. Burada öğrencilerin okul numaraları, okumuş oldukları süre içindeki başarısız oldukları derslerin kodları, ders adları ve bu derslerden almış oldukları notlar görülmektedir.

OKUL NO	DERS KODU	DERS ADI	SENE SONU NOTU
441	10ATEÖBHR	TEMEL ELEKTRONİK VE ÖLÇME	29
441	10ABTTBHR	BİLİŞİM TEKNOLOJİLERİ TEMELLERİ	43
441	11AGÖPBHR	GÖRSEL PROGRAMCILIK	23
443	9ADVABHR	DİL VE ANLATIM	37
443	9AFİZ	FİZİK	28
443	9AKİM	KİMYA	38
443	9AMATBHR	MATEMATİK	44
443	10AMAT	MATEMATİK	42
443	10ATEÖ	TEMEL ELEKTRONİK VE ÖLÇME	37
464	9AKİM	KİMYA	35
464	9AAUB	ASTRONOMİ VE UZAY BİLİMLERİ	40
468	9AKİM	KİMYA	36

Şekil IV.6 Maden Liste ekranından bir kesit

IV.2.3. Ders Listesi

Ders Listesi ekranında maden listede elde edilen sonuçlar kullanılarak her bir öğrencinin başarısız olduğu tüm dersler, ders kodları ile birlikte listelenmektedir. Böylece ders listesi ekranında aday öğe kümelerinin elde edileceği veri tablosu oluşturulmaktadır. Şekil IV.7'de Ders listesinin bir kesiti verilmektedir. Burada öğrencilerin okul numaraları ile birlikte okumuş oldukları süre içindeki başarısız oldukları tüm derslerin kodları görülmektedir.

OKUL NO	KALDIĞI DERSLER
413	9AFİZ - 9AKİM - 9AMAT - 9AMATBHR - 10AMAT
414	9AFİZ - 9AKİM - 9AMATBHR - 9AAUB - 10AMAT
420	9AMAT - 10AMAT
423	9AMAT - 9AFİZ
436	9ABİT - 9ADİH - 9AFİZ - 9AKİM - 9AAUB - 9AİNGBHR - 10AMAT - 10AMATBHR - 10ATARBHR - 10ATEÖ - 10...
441	9AMAT - 9ATEDBHR - 9AAUB - 10AMATBHR - 10ATEÖ - 10ATEÖBHR - 10ABTTBHR - 11AGÖPBHR
443	9ADVABHR - 9AFİZ - 9AKİM - 9AMATBHR - 10AMAT - 10ATEÖ
464	9AKİM - 9AAUB
468	9AKİM
469	9ACOGBHR - 9ADVABHR - 9AKİM - 9AKİMBHR - 9AMATBHR - 10AMAT - 10ATAR - 10ATEÖ
471	9AFİZ - 9AKİM - 10AMAT
472	9AFİZ - 9AMAT - 9ATARBHR - 9ATED - 9AAUB - 9AİNGBHR - 9ABİT - 9ABİTBHR - 9ADVA - 10ACOG - 10ADV...

Şekil IV.7 Ders Listesi ekranından bir kesit

IV.2.4. Diğer İşlemler

IV.2.4.1. 1.Adım

Algoritmanın birinci adımında, algoritma Şekil IV.7'deki ders listesi tablosunu tarar ve her nesnenin kaç işlemde geçtiğinin bilgisine erişilir. Elde edilen bu veriler Apriori algoritmasının C_1 aday öğe kümesini temsil eder. Bu veriler ekranda ders kodu, ders adı ve destek bilgisi şeklinde görüntülenir. Daha sonraki adımlarda kullanılmak üzere minimum destek ve minimum güven değerleri belirlendikten sonra bir sonraki aşamaya geçilmektedir.

IV.2.4.2. 2.Adım

İkinci adımda, bir önceki adımda belirlenen minimum destek (support) değeri yardımı ile apriori algoritmasının L_1 bir öğeli sık geçen öğe kümesi elde edilir, yani minimum destek değerine eşit veya daha yüksek destek değerine sahip öğeler L_1 sık geçen öğe kümesini oluştururlar. Elde edilen bu veriler veri tabanında veri_L1 tablosunda tutulmaktadır. Tablonun oluşturulmasında ve verilerin tabloya doldurulmasında aşağıdaki kodlar kullanılmaktadır.

```
tabloadi = "veri_L1"
cmd.CommandText = "create table " & tabloadi & "(derskod text(15),dersadi
text(50),destek number)"
cmd.ExecuteScalar()
Dim a1, a2 As String
Dim a3 As Byte
mindestek = TextBox30.Text
For s = 0 To DataGridView14.RowCount - 1
    If DataGridView14.Item(2, x).Value >= mindestek Then
        a1 = DataGridView14.Item(0, s).Value
        a2 = DataGridView14.Item(1, s).Value
        a3 = DataGridView14.Item(2, s).Value
        cmd.CommandText = "insert into veri_L1(derskod, dersadi, destek)" & _
        "values('" & a1 & "','" & a2 & "','" & a3 & "')"
        Dim sonuc = cmd.ExecuteNonQuery
    End if
Next
```

Şekil IV.8'de L_1 sık geçen öğe kümesinin oluşumu görülmektedir.

OKUL	KALDIĞI DERSLER
413	9AFİZ - 9AKİM - 9AMAT - 9AMATBHR - 10AMAT
414	9AFİZ - 9AKİM - 9AMATBHR - 9AAUB - 10AMAT
420	9AMAT - 10AMAT
423	9AMAT - 9AFİZ
436	9ABİT- 9ADİH - 9AFİZ - 9AKİM - 9AAUB - 9AİNGBHR - 10AMAT - 10AMATBHR - 10ATARBHR - 10ATEÖ - 10ATEÖBHR - 10ABTT - 11AİTA
441	9AMAT - 9ATEDBHR - 9AAUB - 10AMATBHR - 10ATEÖ - 10ATEÖBHR - 10ABTTBHR - 11AGÖPBHR
443	9ADVABHR - 9AFİZ - 9AKİM - 9AMATBHR - 10AMAT - 10ATEÖ
464	9AKİM - 9AAUB - 10AMAT
468	9AKİM
469	9ACOGBHR - 9ADVABHR - 9AKİM - 9AKİMBHR - 9AMATBHR - 9AAUB - 10AMAT - 10ATAR - 10ATEÖ
471	9AFİZ - 9AKİM - 10AMAT
472	9AFİZ - 9AMAT - 9ATARBHR - 9ATED - 9AAUB - 9İNGBHR - 9ABİT - 9ABİTBHR - 9ADVA - 10ACOG - 10ADVA - 10AMAT - 10AMES - 10APPR - 10ATARBHR - 10ATED - 10ATEÖ – 11AFEL - 11AGÖPBHR - 11AİTA
480	9AFİZ - 9AKİM - 9AMAT
491	11AGÖPBHR
494	9ABİT - 9AFİZ - 9AKİM - 9AAUB - 9AİNGBHR - 10AMAT - 10ATEÖBHR - 11AGÖPBHR - 11AİTA
498	9ABYLBHR - 9AFİZ - 9AKİM - 9AAUB - 9AİNG - 10AMAT - 10AMATBHR - 10ATEÖ - 10ATEÖBHR - 10ABTT
500	10AMAT - 10AİNGBHR
506	9ADVA - 9AFİZ - 9AKİM - 9AMAT - 9AMATBHR - 9ATED - 9AAUB - 9AİNG - 9AİNGBHR - 10AMAT - 10AMATBHR - 10ATEÖ - 10ATEÖBHR - 10ABTT - 11AGÖPBHR
516	9ABİT - 9AFİZ - 9AMAT - 9AAUB - 10AMAT - 10ATED - 10ATEÖ - 11AİTA
520	9ABİT - 9ACOG - 9ACOGBHR - 9ADVA - 9AFİZ - 9AAUB - 10AMATBHR - 10APPRBHR - 10ATARBHR - 10ATED - 10ATEÖ - 11AİTA
522	9ABİT - 10ABTT - 10ABTTBHR
529	9ADVABHR - 9AFİZ - 9AMAT - 10AMAT - 10ATAR - 10ATED - 10ATEÖ - 10ATEÖBHR - 11AFEL - 11AFELBHR - 11AGÖPBHR
539	9ABİT - 9AFİZ - 9AKİM - 9AMAT - 9AMATBHR - 9AAUB - 9AAUBBHR - 10AMAT - 10ATED - 10ATEÖ - 10ATEÖBHR - 10ABTT -
541	9ABYLBHR - 9ACOG - 9ADVA - 9ADVABHR - 9ADİH - 10ADVA - 9AFİZ - 9AKİM - 9AMAT - 9AMATBHR - 9ATEDBHR - 9AAUB - 9AİNGBHR - 10AMAT - 10APPR - 10ATAR - 10ATEÖ - 10AİNG - 11AİTA
553	9ABİT - 9ABİTBHR - 9AAUB - 10AMAT - 10ATEÖ

C_1 aday öğe kümesi

DERS KODU	DERS ADI	DESTEK
10AMAT	MATEMATİK	18
9AFİZ	FİZİK	16
9AAUB	ASTRONOMİVE UZAY BİLİMLERİ	14
9AKİM	KİMYA	14
10ATEÖ	TEMEL ELEKTRONİK VE ÖLÇME	13
9AMAT	MATEMATİK	11
9ABİT	BİGİVE İLETİŞİM TEKNOLOJİLERİ	8
10ATEÖBHR	TEMEL ELEKTRONİK VE ÖLÇME	7
11AGÖPBHR	GÖRSEL PROGRAMCILIK	7
9AMATBHR	MATEMATİK	7
11AİTA	İKILAP TARİHİ VE ATATÜRKÇÜLÜK	7
10AMATBHR	MATEMATİK	5
9AİNGBHR	İNGİLİZCE	5
10ABTT	BİLİŞİM TEKNOLOJİERİ TEMELLERİ	5
10ATED	TÜRK EDEBİYATI	5
9ADVA	DİL VE ANLATIM	4
9ADVABHR	DİL VE ANLATIM	4
10ATARBHR	TARİH	3
10ATAR	TARİH	3
10APPR	PAKET PROGRAMLAR	2
9ATEDBHR	TÜRK EDEBİYATI	2
11AFEL	FELSEFE	2
10ABTTBHR	BİLİŞİM TEKNOLOJİLERİ TEMELLERİ	2
9ABYLBHR	BİYOLOJİ	2
9ABİBHR	BİGİ VE İLETİŞİM TEKNOLOJİLERİ	2
9AİG	İGİİCE	2
9ADİH	DEMOKRASİ VE İNSAN HAKLARI	2
9ACOG	COĞRAFYA	2
10ADVA	DİL VE ANLATIM	2
9ATED	TÜRK EDEBİYATI	2
9ACOGBHR	COBAFYA	2
10AMES	MESLEKİ GELİŞİM	1
10İNGBHR	İNGİLİZCE	1
9AAUBBHR	ASTRONOMİVE UZAY BİLİMLERİ	1
10AİNG	İNGİLİZCE	1
10APPRBHR	PAKET PROGRAMLAR	1
9ATARBHR	TARİH	1
11AFELBHR	FELSEFE	1
10ACOG	COĞRAFYA	1
9AKİMBHR	KİMYA	1

Destek değeri 9'un altında olanlar listeden silinir. Destek değeri 9 ve üstünde olanlar L_1 sık geçen öğe kümesini oluşturur.

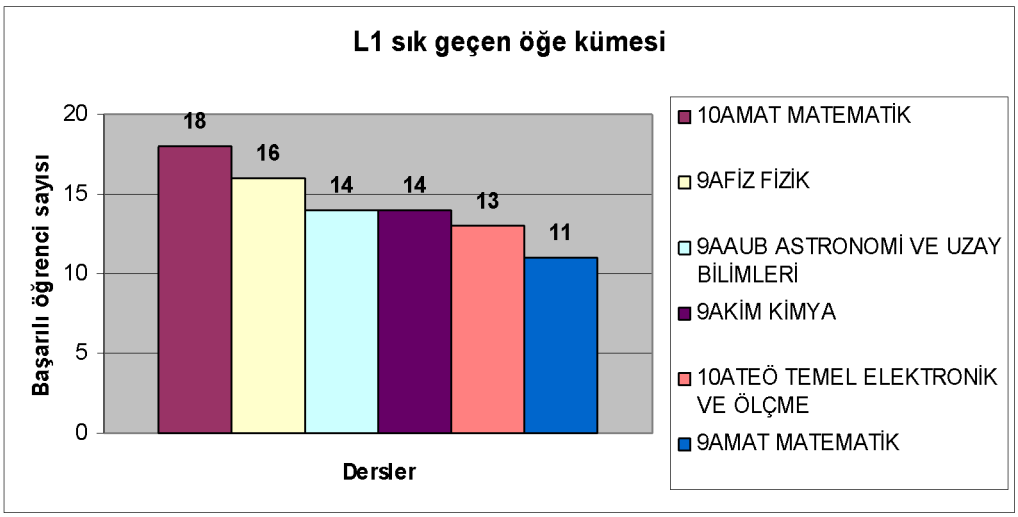
=> Minimum destek (9)

L_1 sık geçen öğe kümesi

DERS KODU	DERS ADI	DESTEK
10AMAT	MATEMATİK	18
9AFİZ	FİZİK	16
9AAUB	ASTRONOMİ VE UZAY BİLİMLERİ	14
9AKİM	KİMYA	14
10ATEÖ	TEMEL ELEKTRONİK VE ÖLÇME	13
9AMAT	MATEMATİK	11

Destek değeri, o dertsen kaç kişinin başarılı olduğunu gösterir.

Şekil IV.8 L_1 sık geçen öğe kümesinin oluşumu



Şekil IV.9 L_1 Sık geçen öğe kümesi

IV.2.4.3. 3. Adım

Bu adımda L_1 kümesinin öğelerinin ikili kombinasyonundan ($L_1 \infty L_1$ – birleştirme işlemi) C_2 aday öğe kümesi oluşturulur. C_2 aday öğe kümesi verileri veritabanında veri_C2 tablosunda tutulur. C_2 aday öğe kümesinin oluşturulmasında aşağıdaki kodlar kullanılmaktadır.

```
Dim cmd As New OleDb.OleDbCommand
Dim tabloadi
cnn.Open()
cmd.Connection = cnn
tabloadi = "veri_C2"
cmd.CommandText = "create table " & tabloadi & "(L1XL1 text(50),L1_1
text(15), L1_2 text(15))"
cmd.ExecuteScalar()
cnn.Close()
'Join işlemi çin L1 tablosunun aynısı "veri_L1_G" oluşturuluyor
tabloadi = "veri_L1_G"
cnn.Open()
cmd.CommandText = "select * into " & tabloadi & " from veri_L1"
cmd.ExecuteScalar()
cnn.Close()
```

'Birleştirme(join) işlemi uygulanarak C_2 aday öğe kümesi oluşturuluyor

```
cnn.Open()
cmd.CommandText = "insert into veri_C2 (L1_1, L1_2, L1XL1)" _
& "select veri_L1.derskod, veri_L1_G.derskod, rtrim( veri_L1.derskod )" _
& "+ ' X ' + rtrim (veri_L1_G.derskod) as derskod " _
& " from veri_L1 inner join " _
& " veri_L1_G on veri_L1.derskod < veri_L1_G.derskod " _
& "order by veri_L1.derskod"
cmd.ExecuteScalar()
cnn.Close()
```

Şekil IV.10'da C_2 aday öğe kümesinin oluşumu görülmektedir.

L_1 sık geçen öğe kümesi

DERS KODU	DERS ADI	DESTEK
10AMAT	MATEMATİK	18
9AFİZ	FİZİK	16
9AAUB	ASTRONOMİ VE UZAY BİLİMLERİ	14
9AKİM	KİMYA	14
10ATEÖ	TEMEL ELEKTRONİK VE ÖLÇME	13
9AMAT	MATEMATİK	11

→ $L_1 \infty L_1$

Birleştirme (join) işlemi

↓

C_2 aday öğe kümesi

L_1 kümesinin öğelerinin ikili kombinasyonundan ($L_1 \infty L_1$ – birleştirme işlemi) yeni bir aday öğe kümesi olan C_2 oluşturulur.

L1 X L1	L1_1	L1_2
10AMAT X 10ATEÖ	10AMAT	10ATEÖ
10AMAT X 9AAUB	10AMAT	9AAUB
10AMAT X 9AMAT	10AMAT	9AMAT
10AMAT X 9AKİM	10AMAT	9AKİM
10AMAT X 9AFİZ	10AMAT	9AFİZ
10ATEÖ X 9AAUB	10ATEÖ	9AAUB
10ATEÖ X 9AMAT	10ATEÖ	9AMAT
10ATEÖ X 9AKİM	10ATEÖ	9AKİM
10ATEÖ X 9AFİZ	10ATEÖ	9AFİZ
9AAUB X 9AMAT	9AAUB	9AMAT
9AAUB X 9AKİM	9AAUB	9AKİM
9AAUB X 9AFİZ	9AAUB	9AFİZ
9AFİZ X 9AMAT	9AFİZ	9AMAT
9AFİZ X 9AKİM	9AFİZ	9AKİM
9AKİM X 9AMAT	9AKİM	9AMAT

Şekil IV.10 C_2 aday öğe kümesinin $L_1 \infty L_1$ birleştirme işlemi ile oluşumu

IV.2.4.4. 4. Adım

Bu adımda C_2 aday öğe kümesi oluştuktan sonra budama (prune) işleminin yapılması gerekmektedir. Budama işleminde C_2 aday öğe kümesindeki öğelerin alt kümelerinin L_1 sık geçen öğe kümesinde olup olmadığına bakılır, eğer herhangi bir

öğenin alt kümelerinden biri L_1 sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_2 aday öğe kümesinden silinir. C_2 ve sonraki aday öğe kümeleri için bu işlem yapılması gerekmekte iken, C_2 aday öğe kümesinin tüm öğelerinin alt kümeleri tek öğeden oluştuğu için ve L_1 sık geçen öğe kümesinden birleştirme işlemine uğrayarak C_2 aday öğe kümesine geldiği için, C_2 aday öğe kümesi için bu işlem gerçekleştirilmez. Yani C_2 aday öğe kümesinin bütün alt kümeleri L_1 sık geçen öğe kümesinde yer almaktadır.

Bundan dolayı bu adımda, C_2 aday öğe kümesine bakılır ve minimum destek değerine eşit veya daha yüksek olan değerlerden L_2 sık geçen öğe kümesi elde edilir. Oluşan bu sık geçen öğe küme verileri veritabanında veri_L2 tablosunda tutulur. L_2 sık geçen öğe kümesinin oluşturulmasında aşağıdaki kodlar kullanılmaktadır.

```
'L2 için yeni bir tablo oluşturuluyor
tabloadi = "veri_L2"
cnn.Open()
cmd.CommandText = "create table " & tabloadi & "(derskod text(15),destek
number)"
cmd.ExecuteScalar()
cnn.Close()
'Veri_C2 tablosunun kayıt sayısı bulunuyor
cnn.Open()
tabloadi = "veri_C2"
cmd.CommandText = "select count(*) from " & tabloadi & ""
c = cmd.ExecuteScalar
'veri_C2 tablosundaki kayıtar datareader'a aktarılıyor
Dim dizi(c, 2) As String
tabloadi = "veri_C2"
cmd.CommandText = " select * from " & tabloadi & ""
Dim dtr As OleDb.OleDbDataReader
dtr = cmd.ExecuteReader
Dim byy As Integer = 1
    Do While dtr.Read = True
    dizi(byy, 1) = dtr(1)
    dizi(byy, 2) = dtr(2)
```

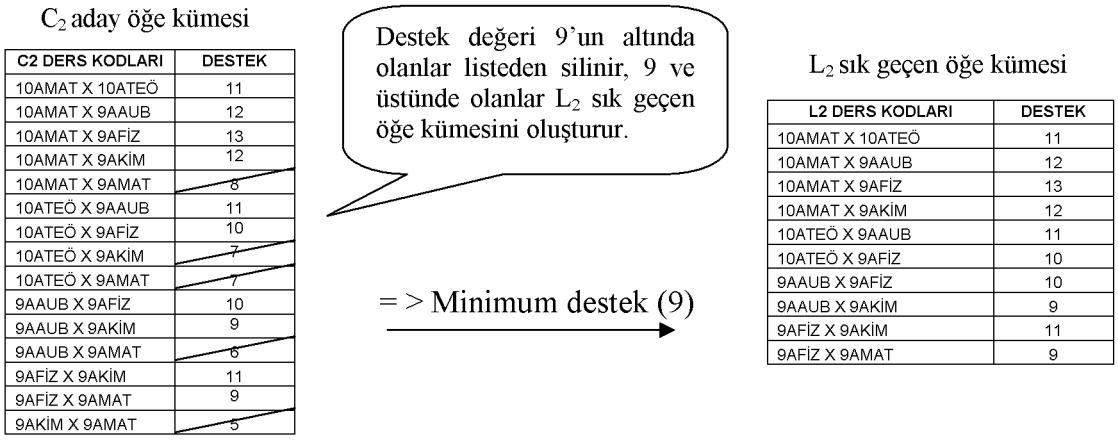
```
            byy = byy + 1
        Loop
        dtr.Close()
        Dim sn As Integer = 1
        'veri_L2_1 ve veri_L2_2 tabloları oluşturuluyor
        tabloadi2 = "veri_L2_1"
        cnn.Open()
        cmd.CommandText = "create table " & tabloadi2 & "(skod_L2 text(15), belge
text(15))"
        cmd.ExecuteScalar()
        cnn.Close()
        cnn.Open()
        tabloadi2 = "veri_L2_2"
        cmd.CommandText = "create table " & tabloadi2 & "(skod_L2 text(15), belge
        text(15))"
        cmd.ExecuteScalar()
        cnn.Close()
        'Verilerin  veri_L2_1 ve veri_L2_2 tablolarına aktarılması
        Dim xl As Integer
        For xl = 1 To c
        'Verilerin  veri_L2_1 tablosuna aktarımı
            cnn.Open()
            cmd.CommandText = "insert into veri_L2_1 (skod_L2,belge)" _
            & "select '" & RTrim(dizi(xl, 1)) & " X " & RTrim(dizi(xl, 2)) & "'," _
            & "okulno as belge " _
            & "from kural" _
            & " where derskod= '" & dizi(xl, 1) & "'"
            cmd.ExecuteScalar()
            cnn.Close()
        ' Verilerin  veri_L2_2 tablosuna aktarımı
            cnn.Open()
            cmd.CommandText = "insert into veri_L2_2 (skod_L2,belge)" _
            & "select '" & RTrim(dizi(xl, 1)) & " X " & RTrim(dizi(xl, 2)) & "'," _
            & "okulno as belge " _
```

```
    & "from kural" _
    & " where derskod= '" & dizi(xl, 2) & "'"
    cmd.ExecuteScalar()
    cnn.Close()
Next
```

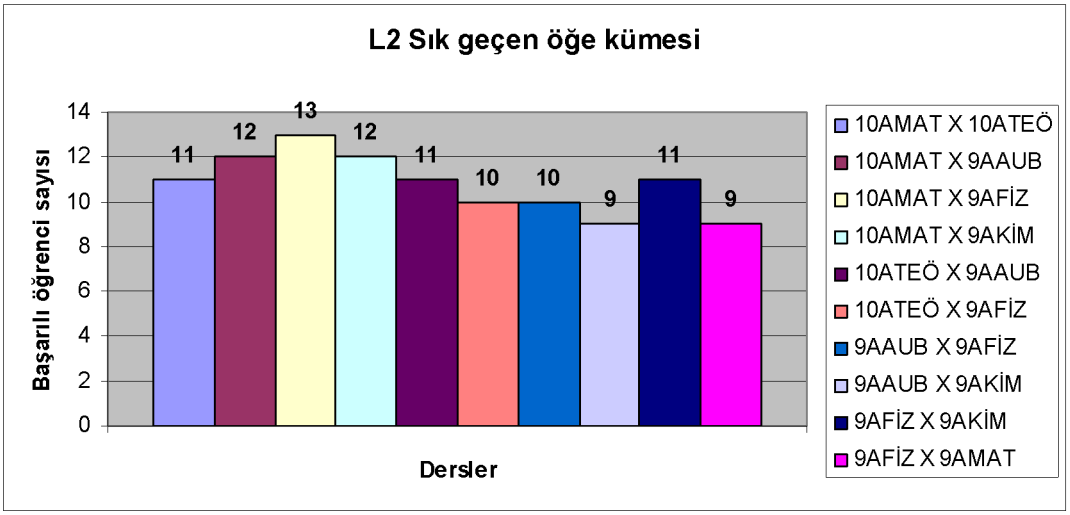
'veri_L2_1 ve veri_L2_2 tablolarının birleştirme (join) işlemine uğrayarak destek seviyelerinin veri_L2 tablosuna aktarılması

```
cnn.Open()
cmd.CommandText = " insert into veri_L2 (derskod, destek)" _
& "select veri_L2_1.skod_L2, count(veri_L2_1.belge) as destek " _
& "from veri_L2_1 inner join " _
& "veri_L2_2 on veri_L2_1.skod_L2 = veri_L2_2.skod_L2 and " _
& "veri_L2_1.belge = veri_L2_2.belge " _
& "group by veri_L2_1.skod_L2 "
cmd.ExecuteScalar()
cmd.CommandText = "delete from veri_L2 " _
& " where destek < " & mindestek & " "
cmd.ExecuteScalar()
cnn.Close()
```

Şekil IV.11'de L_2 sık geçen öğe kümesinin oluşumu görülmektedir.



Şekil IV.11 L_2 sık geçen öğe kümesinin oluşumu



Şekil IV.12 L_2 sık geçen öğe kümesi

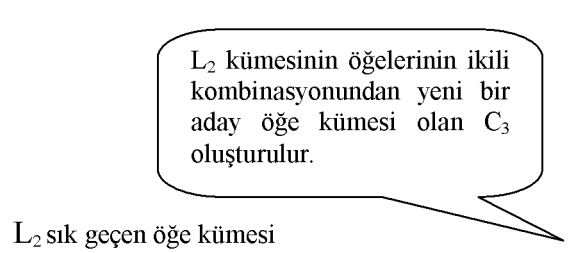
IV.2.4.5. 5. Adım

Uygulamanın beşinci adımında L_2 kümesinin öğelerinin ikili kombinasyonundan ($L_2 \infty L_2$ – birleştirme işlemi) yeni bir aday öğe kümesi olan C_3 aday öğe kümesi oluşturulur. Oluşan bu aday öğe küme verileri veritabanında veri_C3 tablosunda tutulur. C_3 aday öğe kümesinin oluşturulmasında aşağıdaki kodlar kullanılmaktadır.

```
'Veri_C3 tablosu oluşturuluyor
tabloadi = "veri_C3"
cnn.Open()
cmd.CommandText = "create table " & tabloadi & "(L2XL2 text(50),L2_1
text(15), L2_2 text(15),L2_3 text(15))"
cmd.ExecuteScalar()
cnn.Close()
'Join işlemi çin L2 tablosunun kopyası oluşturuluyor
tabloadi = "veri_C2_G"
cnn.Open()
cmd.CommandText = "select * into " & tabloadi & " from veri_C2"
cmd.ExecuteScalar()
cnn.Close()
'Birleştirme (join) işlemi ile C3 aday öğe kümesi oluşturuluyor
cnn.Open()
```

```
cmd.CommandText = "insert into veri_C3 (L2_1, L2_2, L2_3, L2XL2)" _
& "select veri_C2.L1_1, veri_C2.L1_2," _
& "veri_C2_G.L1_2, rtrim (veri_C2.L1_1)" _
& " + ' X ' + rtrim (veri_C2.L1_2) + ' X ' + rtrim (" _
& " veri_C2_G.L1_2) as L2XL2 " _
& " from veri_C2 inner join " _
& " veri_C2_G on veri_C2.L1_1 = veri_C2_G.L1_1 and " _
& " veri_C2.L1_2 < veri_C2_G.L1_2 " _
'& " order by veri_C2.L1_1 , veri_C2.L1_2, " _
'& " veri_C2_G.L1_2"
cmd.ExecuteScalar()
cnn.Close()
```

Şekil IV.13'te C_3 aday öğe kümesinin oluşumu görülmektedir.



L_2 sık geçen öğe kümesi

L2 DERS KODLARI	DESTEK
10AMAT X 10ATEÖ	11
10AMAT X 9AAUB	12
10AMAT X 9AFİZ	13
10AMAT X 9AKİM	12
10ATEÖ X 9AAUB	11
10ATEÖ X 9AFİZ	10
9AAUB X 9AFİZ	10
9AAUB X 9AKİM	9
9AFİZ X 9AKİM	11
9AFİZ X 9AMAT	9

$L_2 \infty L_2$

Birleştirme (join) işlemi

C_3 aday öğe kümesi

L2 X L2	L2_1	L2_2	L2_3
10AMAT X 10ATEÖ X 9AAUB	10AMAT	10ATEÖ	9AAUB
10AMAT X 9AFİZ X 9AMAT	10AMAT	9AFİZ	9AMAT
10AMAT X 9AKİM X 9AMAT	10AMAT	9AKİM	9AMAT
10AMAT X 9AAUB X 9AMAT	10AMAT	9AAUB	9AMAT
10AMAT X 10ATEÖ X 9AMAT	10AMAT	10ATEÖ	9AMAT
10AMAT X 9AFİZ X 9AKİM	10AMAT	9AFİZ	9AKİM
10AMAT X 9AAUB X 9AKİM	10AMAT	9AAUB	9AKİM
10AMAT X 10ATEÖ X 9AKİM	10AMAT	10ATEÖ	9AKİM
10AMAT X 9AAUB X 9AFİZ	10AMAT	9AAUB	9AFİZ
10AMAT X 10ATEÖ X 9AFİZ	10AMAT	10ATEÖ	9AFİZ
10ATEÖ X 9AFİZ X 9AMAT	10ATEÖ	9AFİZ	9AMAT
10ATEÖ X 9AKİM X 9AMAT	10ATEÖ	9AKİM	9AMAT
10ATEÖ X 9AAUB X 9AMAT	10ATEÖ	9AAUB	9AMAT
10ATEÖ X 9AFİZ X 9AKİM	10ATEÖ	9AFİZ	9AKİM
10ATEÖ X 9AAUB X 9AKİM	10ATEÖ	9AAUB	9AKİM
10ATEÖ X 9AAUB X 9AFİZ	10ATEÖ	9AAUB	9AFİZ
9AAUB X 9AFİZ X 9AMAT	9AAUB	9AFİZ	9AMAT
9AAUB X 9AKİM X 9AMAT	9AAUB	9AKİM	9AMAT
9AAUB X 9AFİZ X 9AKİM	9AAUB	9AFİZ	9AKİM
9AFİZ X 9AKİM X 9AMAT	9AFİZ	9AKİM	9AMAT

Şekil IV.13 C_3 aday öğe kümesinin $L_2 \infty L_2$ dönüşümü ile oluşumu

IV.2.4.6. 6. Adım

Apriori algoritmasının özet koduna göre C_3 aday öğe kümesi oluştuktan sonra budama işleminin yapılması gerekmektedir. Budama işleminde C_3 aday öğe kümesindeki öğelerin alt kümelerinin L_2 sık geçen öğe kümesinde bulunup bulunmadığına bakılır, eğer herhangi bir öğenin alt kümelerinden biri L_2 sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_3 aday öğe kümesinden silinir. Budama sonucunda tüm öğeler silinebilir ve algoritma

sonlanabilir. Eğer C_3 aday öğe kümesinde yapılan budama sonucunda algoritma sonlanır ise L_2 sık geçen öğeler kümesindeki öğeler birliktelik kurallarını oluşturmak için kullanılır. C_3 aday öğe kümesi üzerinde budama işlemi için aşağıdaki kodlar kullanılmaktadır.

```
'budama sonrası silinen öğeleri tutmak için veri_C3G tablosu oluşturulur
tabloadi = "veri_C3_G"
cmd.CommandText = "drop table " & tabloadi & ""
cmd.ExecuteScalar()
cmd.CommandText = "create table " & tabloadi & "(L2XL2 text(50))"
cmd.ExecuteScalar()
cnn.Close()
Dim sayisi As Integer
Dim c, xn As Integer
c = 3
Dim kk As New Integer, kk1, kk2
kk1 = 1
kk2 = 1
For xn = 1 To c
    kk1 = kk1 * xn
Next
For xn = 1 To c - 2
    kk2 = kk2 * xn
Next
kk = kk1 / (2 * 1 * kk2)
Dim mm As Integer, cc, g, say11
Dim kkk(kk, 2) As String
Dim dersler(c), metin As String
cmd.CommandText = "select count(*) from veri_C3"
sayisi = cmd.ExecuteScalar
cmd.CommandText = "select * from veri_C3"
Dim dtr As OleDb.OleDbDataReader
dtr = cmd.ExecuteReader
cc = 1
```

```
' C3 kümesi  dizi içine doldurulur
Dim dizis(sayisi, c) As String
Do While dtr.Read = True
    For g = 1 To c
        dizis(cc, g) = dtr(g)
    Next g
    cc = cc + 1
Loop
dtr.Close()
say11 = 1
For say11 = 1 To sayisi
    metin = ""
'C3 kümesi'nin her satırındaki veriler tek boyutlu bir dizinin  içine
aktarılır
For ik = 1 To c Step 1
    dersler(ik) = dizis(say11, ik)
    If ik = 1 Then
        metin = RTrim(dizis(say11, ik))
    Else
        metin = metin & " X " & RTrim(dizis(say11, ik))
    End If
Next ik
Dim byy As New Integer, nn,, z, op, rt, p, s, ikili, sd
nn = 1
byy = 1
op = 0
rt = 1
'2 boyutlu dizinin 1 ve 2. kolonu tek boyutlu indisten gelen 3 öğenin
' ikili kombinasyonları ile doldurulur
For ikili = 1 To c Step 1
    If c - nn <> 0 Then
'2 boyutlu dizinin 1. kolonu doldurulur
        For sd = 1 To c - nn
            kkk(byy, 1) = dersler(ikili)
```

```
                byy = byy + 1
            Next sd
        End If
        nn = nn + 1
    Next ikili
    rt = 1
    ' 2 boyutlu dizinin 2. kolonu doldurulur
    For xl = 1 To kk
        For rt = 1 To c
            If dersler(rt) = kkk(xl, 1) Then GoTo rtfor
        Next rt
    rtfor:
        For s = rt + 1 To c
            kkk(xl, 2) = dersler(s)
            xl = xl + 1
        Next s
        xl = xl - 1
    Next xl
```

' 2 boyutlu dizinin her satırındaki verilerin L_2 tablosunda bulunup bulunmadıkları kontrol edilir

```
    Dim kon As New Integer, kay
    For kon = 1 To kk
        cmd.CommandText = "select count(*) from veri_L2" _
        & " where derskod = '" & RTrim(kkk(kon, 1)) & " X " _
        & RTrim(kkk(kon, 2)) & "'"
        kay = cmd.ExecuteScalar
```

'kontrol sonucunda C_3'ün her bir satırındaki öğenilerin ikili alt kümelerinden herhangi birii L_2'de 'bulunmazsa C_3'deki ilgili öğe bir geçici tabloya aktarılıp C_3'ten silinir.

```
        If kay = 0 Then
            cmd.CommandText = "insert into veri_C3_G(L2XL2) values " _
            & "( '" & metin & "')"
            cmd.ExecuteScalar()
            cmd.CommandText = "delete from veri_C3 " _
```

```
        & "where L2XL2 = '" & metin & "'"
        cmd.ExecuteScalar()
        Exit For
      End If
    Next kon
    Next say11
    cnn.Close()
```

Şekil IV.14'te C_3 aday öğe kümesinde budama işlemi görülmektedir.

Budama öncesi C_3 aday öğe kümesi

L2 X L2	L2_1	L2_2	L2_3
10AMAT X 10ATEÖ X 9AAUB	10AMAT	10ATEÖ	9AAUB
~~10AMAT X 9AFİZ X 9AMAT~~	10AMAT	9AFİZ	9AMAT
~~10AMAT X 9AKİM X 9AMAT~~	10AMAT	9AKİM	9AMAT
~~10AMAT X 9AAUB X 9AMAT~~	10AMAT	9AAUB	9AMAT
~~10AMAT X 10ATEÖ X 9AMAT~~	10AMAT	10ATEÖ	9AMAT
10AMAT X 9AFİZ X 9AKİM	10AMAT	9AFİZ	9AKİM
10AMAT X 9AAUB X 9AKİM	10AMAT	9AAUB	9AKİM
~~10AMAT X 10ATEÖ X 9AKİM~~	10AMAT	10ATEÖ	9AKİM
10AMAT X 9AAUB X 9AFİZ	10AMAT	9AAUB	9AFİZ
10AMAT X 10ATEÖ X 9AFİZ	10AMAT	10ATEÖ	9AFİZ
~~10ATEÖ X 9AFİZ X 9AMAT~~	10ATEÖ	9AFİZ	9AMAT
~~10ATEÖ X 9AKİM X 9AMAT~~	10ATEÖ	9AKİM	9AMAT
~~10ATEÖ X 9AAUB X 9AMAT~~	10ATEÖ	9AAUB	9AMAT
~~10ATEÖ X 9AFİZ X 9AKİM~~	10ATEÖ	9AFİZ	9AKİM
~~10ATEÖ X 9AAUB X 9AKİM~~	10ATEÖ	9AAUB	9AKİM
10ATEÖ X 9AAUB X 9AFİZ	10ATEÖ	9AAUB	9AFİZ
~~9AAUB X 9AFİZ X 9AMAT~~	9AAUB	9AFİZ	9AMAT
~~9AAUB X 9AKİM X 9AMAT~~	9AAUB	9AKİM	9AMAT
9AAUB X 9AFİZ X 9AKİM	9AAUB	9AFİZ	9AKİM
~~9AFİZ X 9AKİM X 9AMAT~~	9AFİZ	9AKİM	9AMAT

→ 2' li alt küme oluşturma

↓

C_3 aday öğe kümesinin ikili alt kümeleri

DERS KODLARI
(10AMAT X 10ATEÖ) (10AMAT X 9AAUB) (10ATEÖ X 9AAUB)
(10AMAT X 9AFİZ) (~~10AMAT X 9AMAT~~) (9AFİZ X 9AMAT)
(10AMAT X 9AKİM) (~~10AMAT X 9AMAT~~) (~~9AKİM X 9AMAT~~)
(10AMAT X 9AAUB) (~~10AMAT X 9AMAT~~) (~~9AAUB X 9AMAT~~)
(10AMAT X 10ATEÖ) (~~10AMAT X 9AMAT~~) (~~10ATEÖ X 9AMAT~~)
(10AMAT X 9AFİZ) (10AMAT X 9AKİM) (9AFİZ X 9AKİM)
(10AMAT X 9AAUB) (10AMAT X 9AKİM) (9AAUB X 9AKİM)
(10AMAT X 10ATEÖ) (10AMAT X 9AKİM) (~~10ATEÖ X 9AKİM~~)
(10AMAT X 9AAUB) (10AMAT X 9AFİZ) (9AAUB X 9AFİZ)
(10AMAT X 10ATEÖ) (10AMAT X 9AFİZ) (10ATEÖ X 9AFİZ)
(10ATEÖ X 9AFİZ) (~~10ATEÖ X 9AMAT~~) (9AFİZ X 9AMAT)
(~~10ATEÖ X 9AKİM~~) (~~10ATEÖ X 9AMAT~~) (9AKİM X 9AMAT)
(10ATEÖ X 9AAUB) (~~10ATEÖ X 9AMAT~~) (~~9AAUB X 9AMAT~~)
(10ATEÖ X 9AFİZ) (~~10ATEÖ X 9AKİM~~) (9AFİZ X 9AKİM)
(10ATEÖ X 9AAUB) (~~10ATEÖ X 9AKİM~~) (9AAUB X 9AKİM)
(10ATEÖ X 9AAUB) (10ATEÖ X 9AFİZ) (9AAUB X 9AFİZ)
(9AAUB X 9AFİZ) (~~9AAUB X 9AMAT~~) (9AFİZ X 9AMAT)
(9AAUB X 9AKİM) (~~9AAUB X 9AMAT~~) (~~9AKİM X 9AMAT~~)
(9AAUB X 9AFİZ) (9AAUB X 9AKİM) (9AFİZ X 9AKİM)
(9AFİZ X 9AKİM) (9AFİZ X 9AMAT) (~~9AKİM X 9AMAT~~)

↓

Budama işlemi ile C_3 kümesinin 2'li alt kümelerinin L_2 kümesinde bulunup bulunmadığı kontrol ediliyor

↓

L2 DERS KODLARI
10AMAT X 10ATEÖ
10AMAT X 9AAUB
10AMAT X 9AFİZ
10AMAT X 9AKİM
10ATEÖ X 9AAUB
10ATEÖ X 9AFİZ
9AAUB X 9AFİZ
9AAUB X 9AKİM
9AFİZ X 9AKİM
9AFİZ X 9AMAT

↓

Budama sonrası C_3 aday öğe kümesi

L2 X L2	L2_1	L2_2	L2_3
10AMAT X 10ATEÖ X 9AAUB	10AMAT	10ATEÖ	9AAUB
10AMAT X 9AFİZ X 9AKİM	10AMAT	9AFİZ	9AKİM
10AMAT X 9AAUB X 9AKİM	10AMAT	9AAUB	9AKİM
10AMAT X 9AAUB X 9AFİZ	10AMAT	9AAUB	9AFİZ
10AMAT X 10ATEÖ X 9AFİZ	10AMAT	10ATEÖ	9AFİZ
10ATEÖ X 9AAUB X 9AFİZ	10ATEÖ	9AAUB	9AFİZ
9AAUB X 9AFİZ X 9AKİM	9AAUB	9AFİZ	9AKİM

C_3 aday öğe kümesindeki öğelerin alt kümelerinin L_2 kümesindeki varlığı kontrol edilir, herhangi bir öğenin alt kümelerinden biri L_2 kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_3 aday öğe kümesinden silinir.

Şekil IV.14 C_3 aday öğe kümesinde budama işlemi

IV.2.4.7. 7. Adım

Yedinci adımda C_3 aday öğe kümesindeki minimum destek değerine eşit veya yüksek olan değerlerden L_3 sık geçen öğe kümesi elde edilir ve veriler veri_L3 tablosunda saklanır. L_3 sık geçen öğe kümensin oluşturulması için kullanılan kodlar aşağıdaki gibidir.

```
'veri_L3, veri_L3_1 ve veri_L3_2 veri_L3_3 tabloları oluşturuluyor
tabloadi = "veri_L3"
cnn.Open()
cmd.CommandText = "create table " & tabloadi & "(derskod text(50), destek
number)"
cmd.ExecuteScalar()
tabloadi = "veri_L3_1"
cmd.CommandText = "create table " & tabloadi & "(skod_L3 text(50), belge
text(15))"
cmd.ExecuteScalar()
tabloadi = "veri_L3_2"
cmd.CommandText = "create table " & tabloadi & "(skod_L3 text(50), belge
text(15))"
cmd.ExecuteScalar()
tabloadi = "veri_L3_3"
cmd.CommandText = "create table " & tabloadi & "(skod_L3 text(50), belge
text(15))"
cmd.ExecuteScalar()
cnn.Close()
'veri_C3 tablosundaki kayıt sayısı hesaplanıyor
Dim c As Integer
cnn.Open()
cmd.CommandText = "select count (*) from veri_C3"
c = cmd.ExecuteScalar
Dim kkk(c, 3) As String
cmd.CommandText = "select * from veri_C3"
Dim dtr As OleDb.OleDbDataReader
dtr = cmd.ExecuteReader
```

```
Dim byy As Integer = 1
'Dizinin 1. 2. ve 3. kolonu C3 tablosu ile doldurulur
Do While dtr.Read = True
kkk(byy, 1) = dtr(1)
kkk(byy, 2) = dtr(2)
kkk(byy, 3) = dtr(3)
byy = byy + 1
Loop
dtr.Close()
cnn.Close()
'Dizideki verilerin veri_L3_1, veri_L3_2, 'veri_L3_3  tablolarına aktarılması
Dim xl As Integer
Dim sn As Integer = 1
For xl = 1 To c
'Veriler  veri_L3_1 tablosuna aktarılır
cnn.Open()
cmd.CommandText = "insert into veri_L3_1 (skod_L3, belge)" _
& " select '" & RTrim(kkk(xl, 1)) & " X " & RTrim(kkk(xl, 2)) & " X " _
& RTrim(kkk(xl, 3)) & " ' , okulno as belge" _
& " from kural" _
& " where derskod= '" & kkk(xl, 1) & "'"
cmd.ExecuteScalar()
cnn.Close()
'Veriler veri_L3_2 tablosuna aktarılır
cnn.Open()
cmd.CommandText = "insert into veri_L3_2 (skod_L3, belge)" _
& " select '" & RTrim(kkk(xl, 1)) & " X " & RTrim(kkk(xl, 2)) & " X " _
& RTrim(kkk(xl, 3)) & " ' , okulno as belge" _
& " from kural" _
& " where derskod= '" & kkk(xl, 2) & "'"
cmd.ExecuteScalar()
cnn.Close()
'Veriler veri_L3_3 tablosuna aktarılır
cnn.Open()
```

```
cmd.CommandText = "insert into veri_L3_3 (skod_L3, belge)" _
& " select '" & RTrim(kkk(xl, 1)) & " X " & RTrim(kkk(xl, 2)) & " X " _
& RTrim(kkk(xl, 3)) & " ' , okulno as belge" _
& " from kural" _
& " where derskod= '" & kkk(xl, 3) & "'"
cmd.ExecuteScalar()
Next xl
```

'veri_L3_1, veri_L3_2, veri_L3_3 tablolarına birleştirme işlemi uygulanarak verilerin veri_L3 tablosuna aktarılması

```
cmd.CommandText = " insert into veri_L3 ( derskod, destek ) " _
& "select veri_L3_1.skod_L3, count(veri_L3_1.belge) as destek " _
& "from (veri_L3_1 inner join " _
& "veri_L3_2 on veri_L3_1.skod_L3 = veri_L3_2.skod_L3 AND " _
& "veri_L3_1.belge = veri_L3_2.belge) inner join " _
& "veri_L3_3 on veri_L3_1.skod_L3 = veri_L3_3.skod_L3 AND " _
& "veri_L3_1.belge = veri_L3_3.belge " _
& "group by veri_L3_1.skod_L3 "
'& "order by destek desc"
cmd.ExecuteScalar()
```

'minimum destek seviyesinin altındaki değerler silinir

```
cmd.CommandText = "delete from veri_L3 " _
& " where destek < " & mindestek & " "
cmd.ExecuteScalar()
cnn.Close()
```

Şekil IV.15'te L_3 sık geçen öğe kümesinin oluşumu görülmektedir.

C_3 aday öğe kümesi

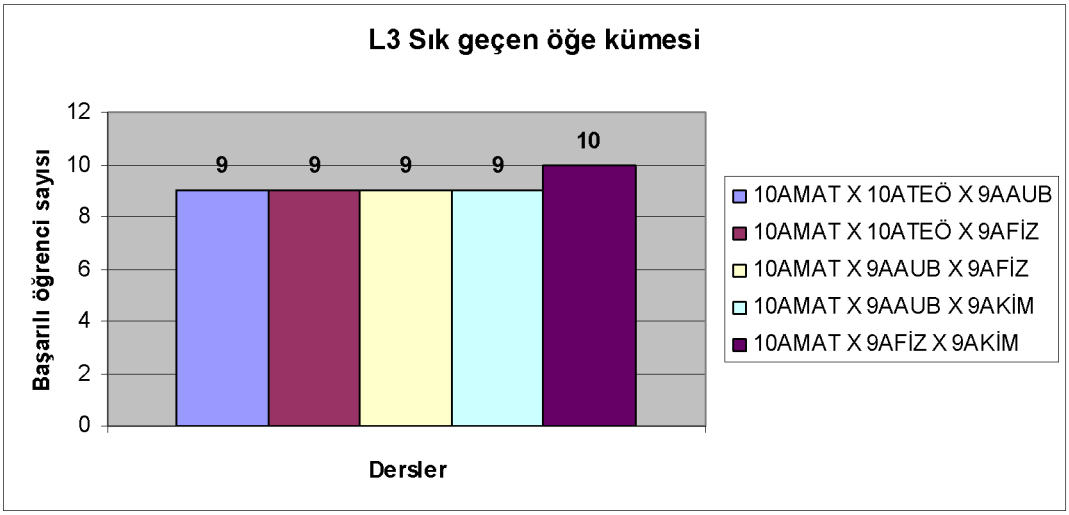
C3 Ders Kodları	Destek
10AMAT X 10ATEÖ X 9AAUB	9
10AMAT X 10ATEÖ X 9AFİZ	9
10AMAT X 9AAUB X 9AFİZ	9
10AMAT X 9AAUB X 9AKİM	9
10AMAT X 9AF□ X 9AKİM	10
10ATEÖ X 9AAUB X 9AFİZ	~~8~~
9AAUB X 9AFİZ X 9AKİM	~~7~~

=> Minimum destek (9)

L_3 sık geçen öğe kümesi

L3 Ders Kodları	Destek
10AMAT X 10ATEÖ X 9AAUB	9
10AMAT X 10ATEÖ X 9AFİZ	9
10AMAT X 9AAUB X 9AFİZ	9
10AMAT X 9AAUB X 9AKİM	9
10AMAT X 9AFİZ X 9AKİM	10

Şekil IV.15 L_3 sık geçen öğe kümesinin oluşumu



Şekil IV.16 L_3 sık geçen öğe kümesi

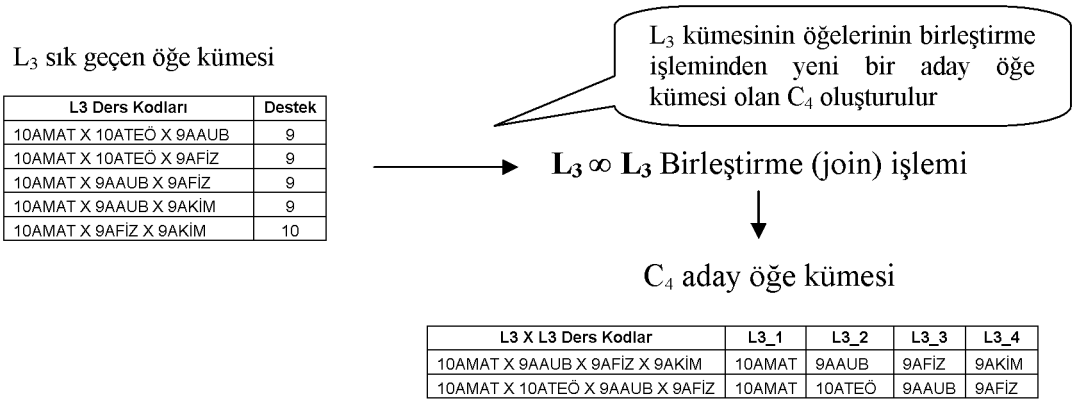
IV.2.4.8. 8. Adım

Bu adımda L_3 sık geçen öğe kümesinin öğelerinin birleştirme işleminden ($L_3 \infty L_3$ – birleştirme işlemi) yeni bir aday öğe kümesi olan C_4 aday öğe kümesi oluşturulur. Oluşan bu aday öğe küme verileri veritabanında veri_C4 tablosunda tutulur. C_4 aday öğe kümesinin oluşturulmasında aşağıdaki kodlar kullanılmaktadır.

```
'Veri_C4 tablosu oluşturulur
tabloadi = "veri_C4"
cnn.Open()
cmd.CommandText = "create table " & tabloadi & "(L3XL3 text(60),L3_1
text(15), L3_2 text(15),L3_3 text(15),L3_4 text(15))"
cmd.ExecuteScalar()
cnn.Close()
'Birleştirme (join) işlemi çin L3 tablosunun kopyası oluşturulur
tabloadi = "veri_L3_G"
cnn.Open()
cmd.CommandText = "select * into " & tabloadi & " from veri_C3"
cmd.ExecuteScalar()
'Birleştirme işlemi uygulandıktan sonra veri_C4 tablosuna veriler aktarılır
cmd.CommandText = "insert into veri_C4 (L3_1, L3_2, L3_3, L3_4, L3XL3)"
& "select veri_C3.L2_1, veri_C3.L2_2," _
& "veri_C3.L2_3,veri_L3_G.L2_3, rtrim (veri_C3.L2_1)" _
```

```
& " + ' X ' + rtrim (veri_C3.L2_2) + ' X ' + rtrim (" _
& " veri_C3.L2_3)" _
& " + ' X ' + rtrim (veri_L3_G.L2_3) as L3XL3 " _
& " from veri_C3 inner join " _
& " veri_L3_G on veri_C3.L2_1 = veri_L3_G.L2_1 and " _
& " veri_C3.L2_2 = veri_L3_G.L2_2 and " _
& " veri_C3.L2_3 < veri_L3_G.L2_3 " _
'& " order by veri_C3.L2_1 , veri_C3.L2_2, " _
'& " veri_C3.L_3"
cmd.ExecuteScalar()
cnn.Close()
```

Şekil IV.17'de C_4 aday öğe kümesinin oluşumu görülmektedir.



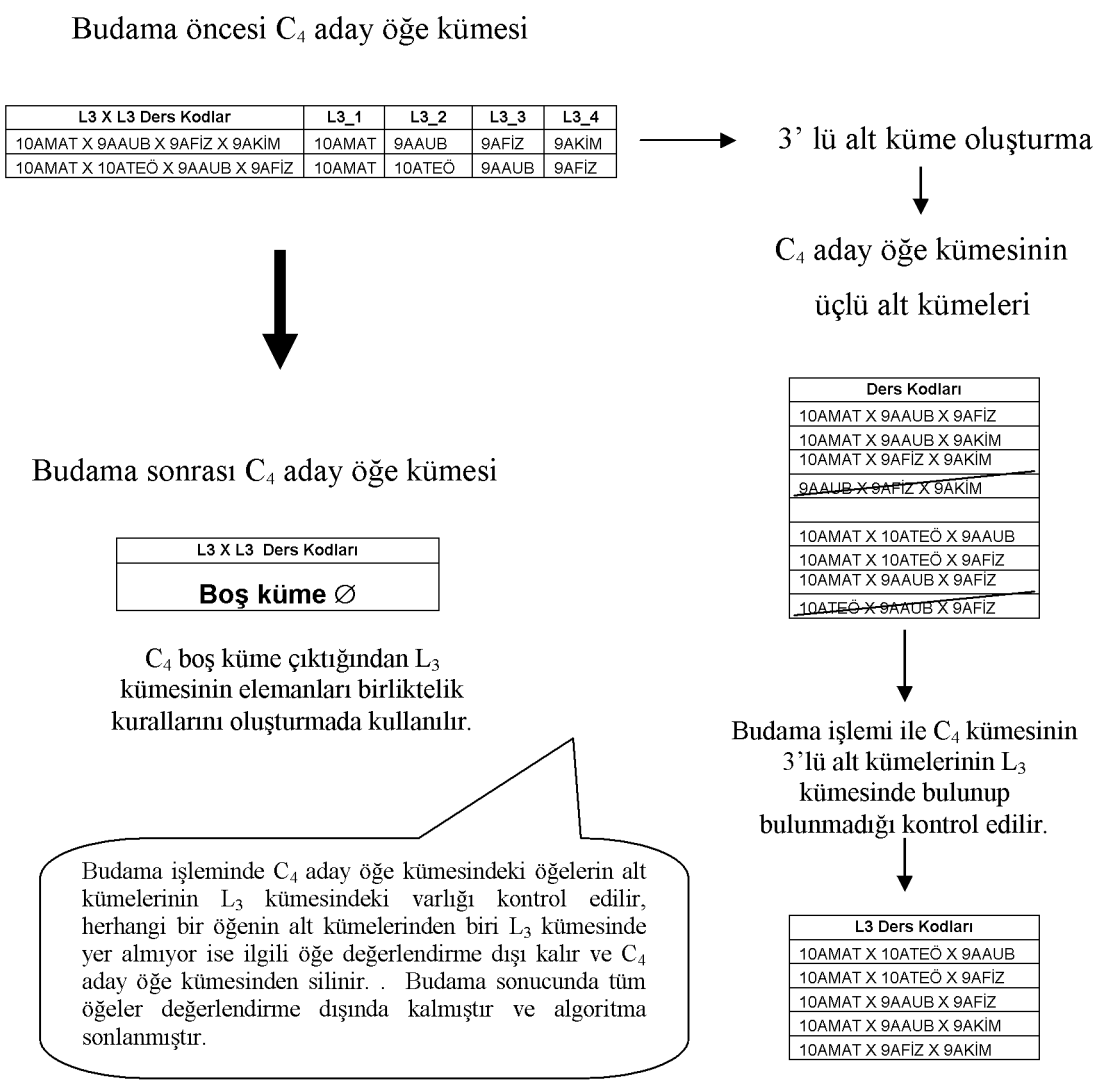
Şekil IV.17 C_4 aday öğe kümesinin oluşumu

IV.2.4.9. 9. Adım

Apriori algoritmasının özet koduna göre C_4 aday öğe kümesi oluştuktan sonra budama işleminin yapılması gerekmektedir. Budama işleminde C_4 aday öğe kümesindeki öğelerin alt kümelerinin L_3 kümesinde bulunup bulunmadığına bakılır. Eğer herhangi bir öğenin alt kümelerinden biri L_3 sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_4 aday öğe kümesinden silinir. Budama işlemi sonucunda tüm öğeler silinebilir ve algoritma sonlanabilir. C_4 aday öğe kümesinde yapılan budama sonucunda algoritma sonlandığında L_3 sık geçen öğe kümesindeki öğeler, birliktelik kurallarını oluşturmak için kullanılır. C_4 aday öğe kümesi üzerinde budama işleminin yapılması için gerekli kodlar C_3 aday öğe

kümesinde budama işlemindeki kodların benzeridir, sadece öğe alt kümelerinin aktarıldığı dizi değişkeni +1 fazladır.

Şekil IV.18'de C_4 aday öğe kümesinde budama işlemi yapılması görülmektedir.



Şekil IV.18 C_4 aday öğe kümesinde budama işlemi

IV.2.4.10. 10. Adım

Onuncu adımda C_4 aday öğe kümesindeki verilerden minimum destek değerine eşit veya yüksek olan değerlerden L_4 sık geçen öğe kümesi elde edilir ve veriler veri_L4 tablosunda saklanır. C_4 aday öğe kümesinden L_4 sık geçen öğe kümesi elde edilirken C_4 aday öğe kümesindeki hiçbir öğe minimum destek değerini geçemez ise algoritma sonlanır ve belirlenen minimum destek değerine eşit veya yüksek değerdeki L_3 kümesinin öğeleri birliktelik kurallarını oluşturmak için kullanılır.

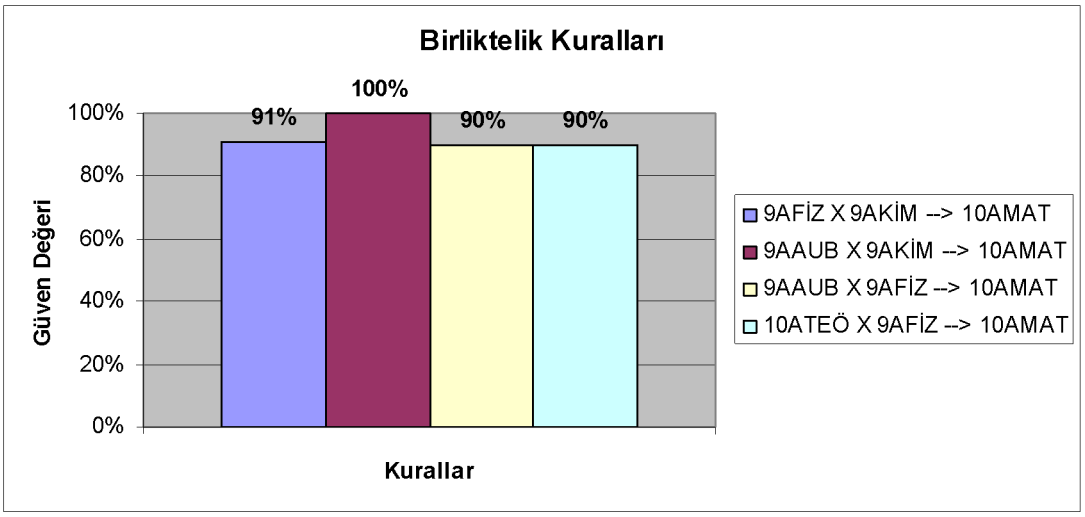
Bu adımlar veri kümesinin boyutuna göre uzatılabilir ya da daha kısa tutulabilir.

IV.2.4.11. Kurallar

Bu bölümde en son elde edilen en sık geçen öğe kümesinin elemanlarından birliktelik kuralları oluşturulur ve birinci adımda belirlenen güven değerine eşit veya daha yüksek olanlar ekranda görüntülenir. Oluşan kurallar, veriler ile Şekil IV.19'da görülmektedir.

KURALLAR	GÜVEN
9AFİZ X 9AKİM --> 10AMAT	91%
9AAUB X 9AKİM --> 10AMAT	100%
9AAUB X 9AFİZ --> 10AMAT	90%
10ATEÖ X 9AFİZ --> 10AMAT	90%

Şekil IV.19 Algoritma sonunda oluşan birliktelik kuralları

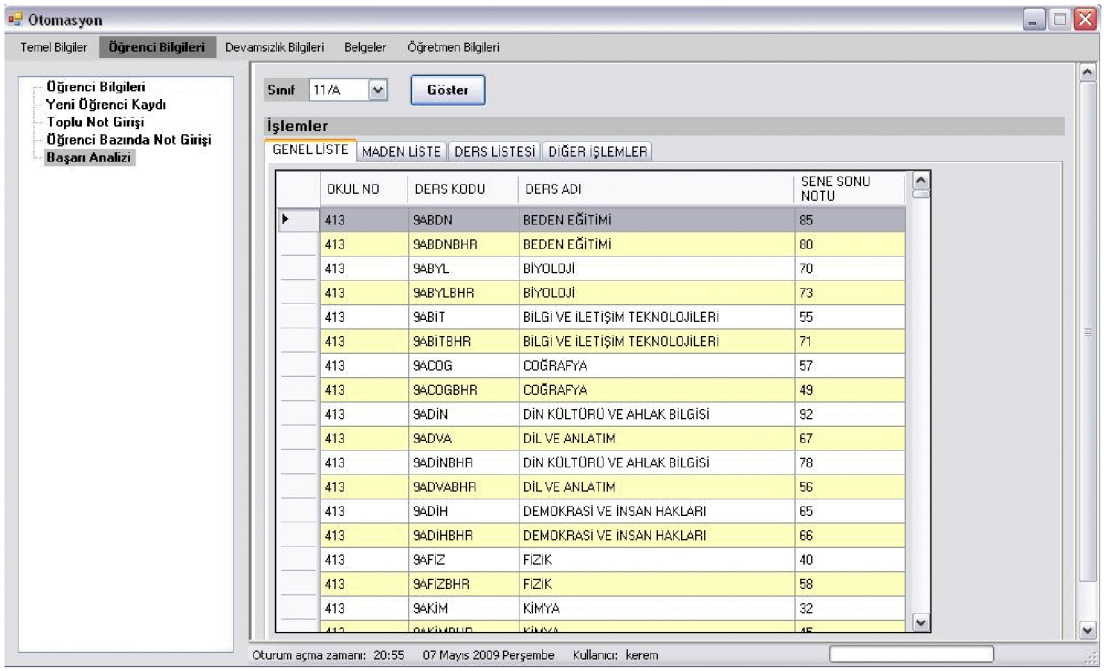


Şekil IV.20 Birliktelik kuralları

IV.3. VERİ MADENCİLİĞİ UYGULAMASININ ÇALIŞTIRILMASI

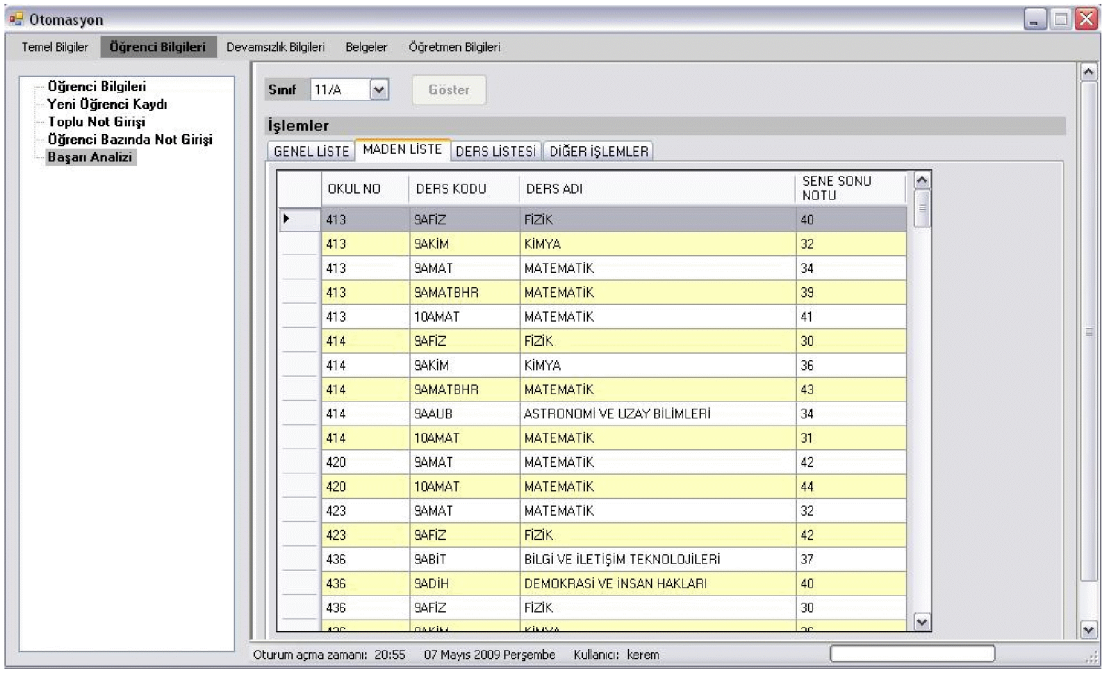
Bu bölümde apriori algoritması gerçek veriler üzerinde uygulanarak yazılımın her aşamasının ekran görüntüleri verilmektedir. Apriori algoritmasının uygulanacağı veri seti olarak Eyüp İ.M.K.B Ticaret Meslek Lisesi 11-A sınıfı öğrencilerinin verileri kullanılmaktadır. 11-A sınıfında 28 öğrenci bulunmaktadır. Birinci dönem ve ikinci dönem olarak dersler ayrı ayrı ele alındığında, öğrencilerin görmüş oldukları 9. sınıfta 30 ders, 10. sınıfta 26 ders ve 11. sınıfta ise 18 ders olmak üzere toplam 74 dersten aldıkları notlar üzerinde apriori algoritması uygulanmaktadır.

Veri madenciliği uygulanacak sınıf seçildikten sonra verilerin eleme süreci öncesindeki ham veri hali Şekil IV.21' deki ekran görüntüsüyle kullanıcıya inceleme amaçlı gösterilir.



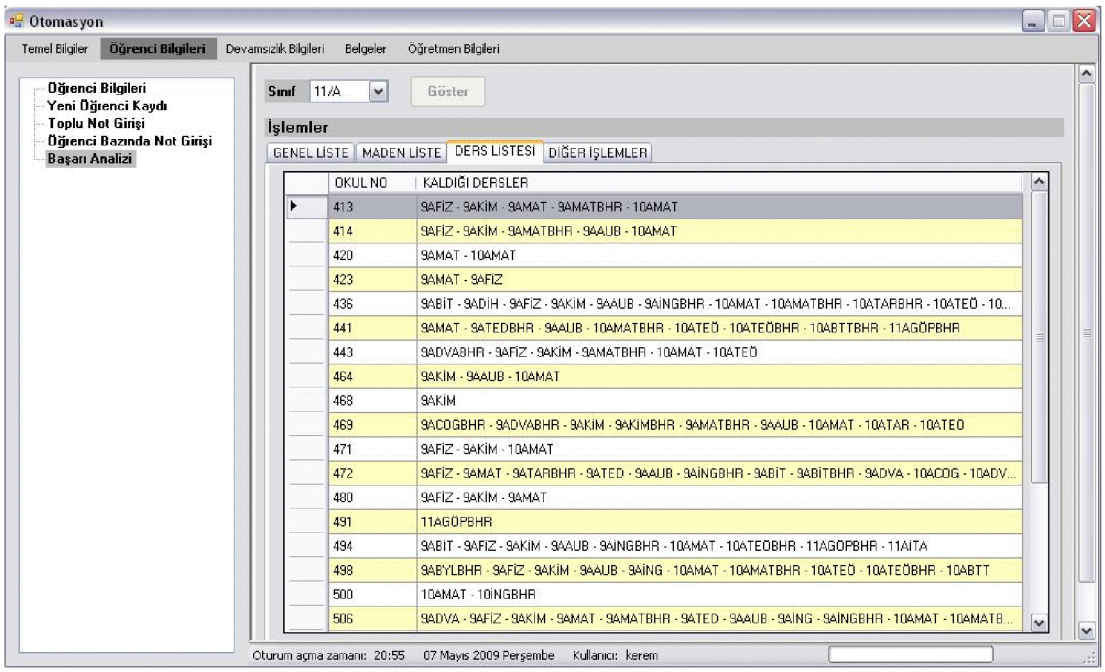
Şekil IV.21 Genel Liste ekranı

Şekil IV.22'deki maden liste ekranında öğrencilerin okumuş oldukları süre içinde görmüş oldukları derslerden sadece başarısız oldukları derslerin notları gösterilmektedir. Burada veriler bir süzgeçten geçirilerek ön eleme işlemi yapılmaktadır.



Şekil IV.22 Maden Liste ekranı

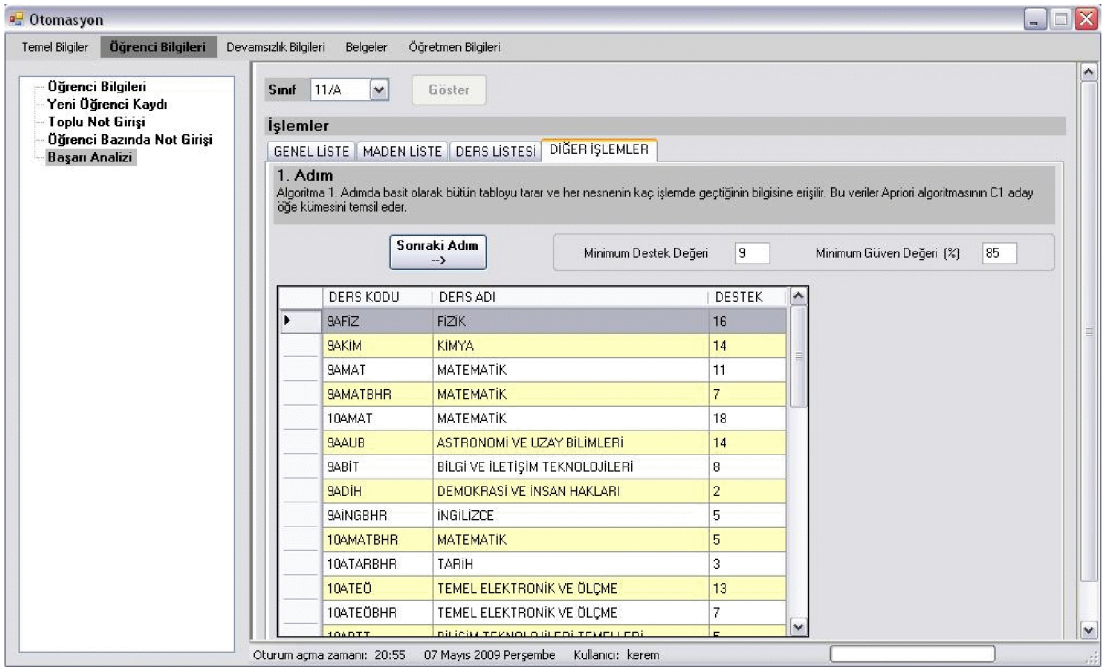
Şekil IV.23'teki ders listesi ekranında maden listeden alınan veriler doğrultusunda aday öğe kümelerinin elde edileceği veri tablosu oluşturulmaktadır.



Şekil IV.23 Ders Listesi ekranı

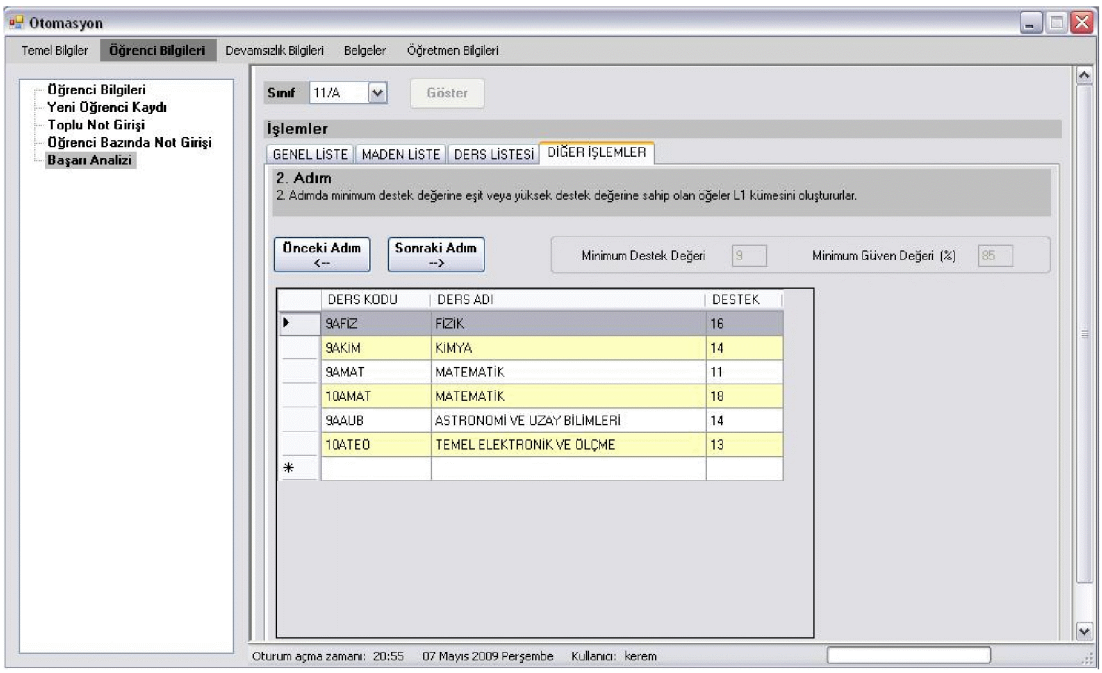
Algoritmanın birinci adımında, algoritma tüm veri setini tarar ve her nesnenin kaç işlemde geçtiğinin bilgisine erişilir ve C_1 aday öğe kümesi içinde bu veriyi tutar. Her nesnenin kaç adet hareket kaydı içinde geçtiği bilgisi nesnenin destek değeri

olarak belirlenir. Şekil IV.24'te veri setinde hareket gören öğe sayısının 40 olduğu görülmektedir. Bu aşamada minimum destek değeri ve minimum güven değeri belirlenir. Minimum destek değeri olarak 9, minimum güven değeri olarak (%) 85 girildi.



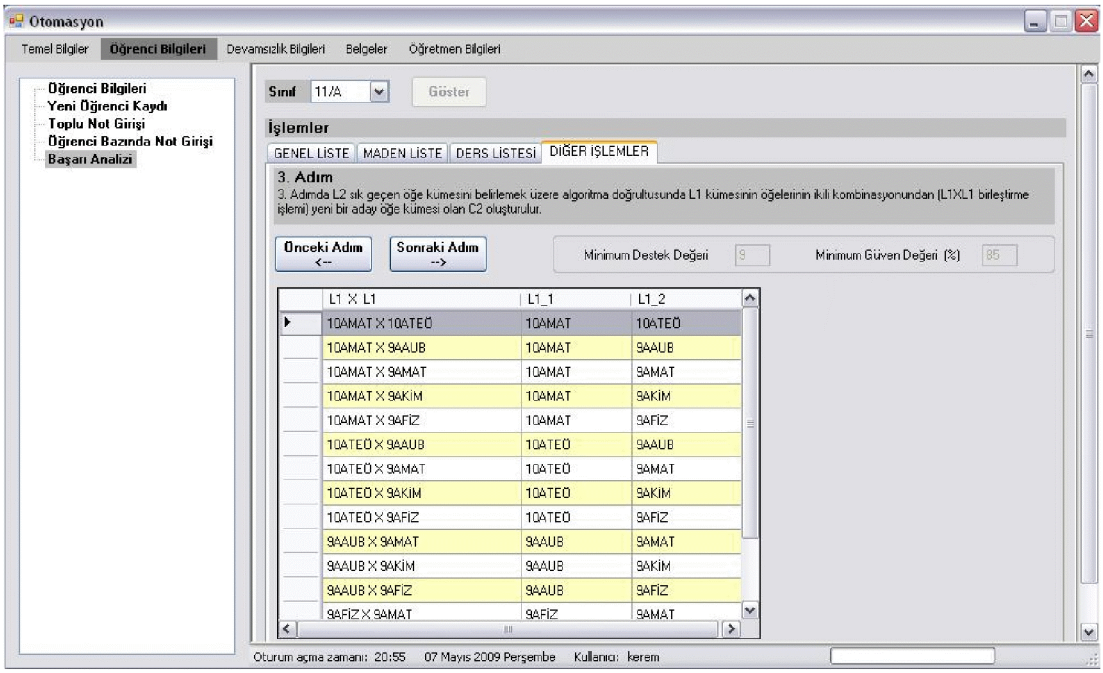
Şekil IV.24 C_1 aday öğe kümesi, destek ve güven değerlerini giriş ekranı

Minimum destek değerine eşit veya daha yüksek destek değerine sahip C_1 aday öğe kümesinin öğeleri L_1 sık geçen öğe kümesini oluşturur. Şekil IV.25'te minimum destek değerine eşit veya daha yüksek öğelerin oluşturduğu L_1 sık geçen öğe kümesinin 6 adet öğeden oluştuğu görülmektedir. Minimum destek değerinden küçük 34 öğe değerlendirme dışı kalmıştır. Burada fizik dersinden kalanların sayının 16, kimya dersinden kalanların sayısının 14, 9.sınıf matematik dersinden klanların sayısının 11, 10. sınıf matematik dersinden klanların sayısının 18, astronomi ve uzay bilimleri dersinden kalanların sayısının 14, temel elektronik ve ölçme dersinden kalanların sayısının 13 olduğu görülmektedir.



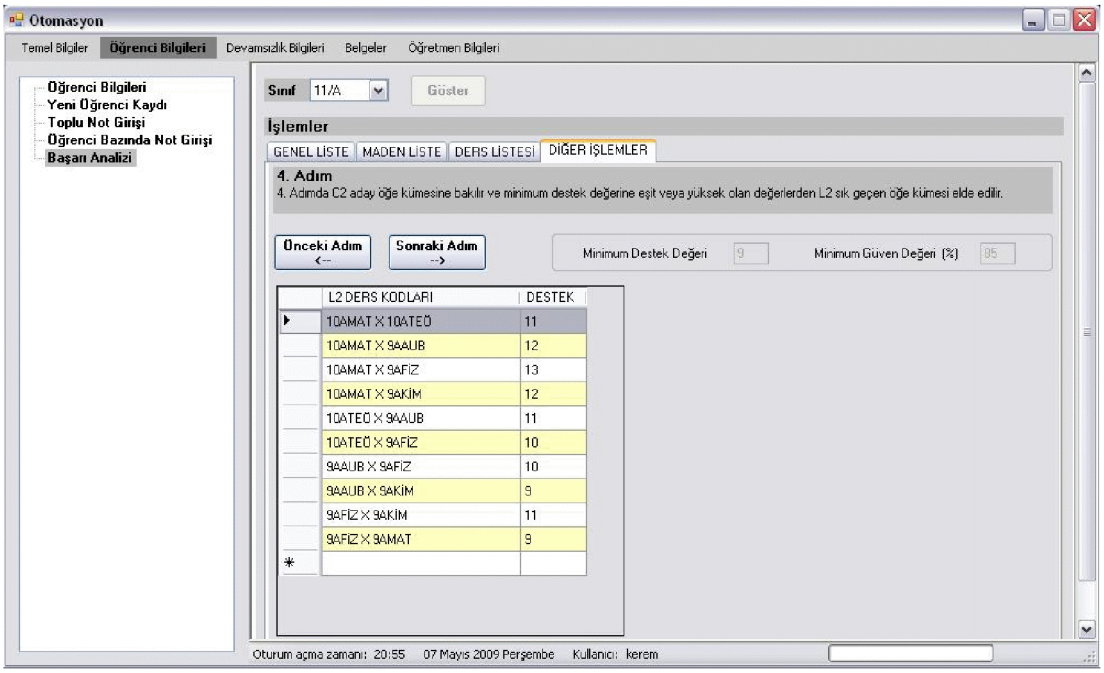
Şekil IV.25 L_1 sık geçen öğe kümesi ekranı

Üçüncü adımda L_1 kümesinin öğeleri üzerinde birleştirme (join) işlemi uygulanır ve ($L_1 \infty L_1$) C_2 aday öğe kümesi oluşur. Şekil IV.26'da C_2 aday öğe kümesinin 15 öğeden oluştuğu görülmektedir.



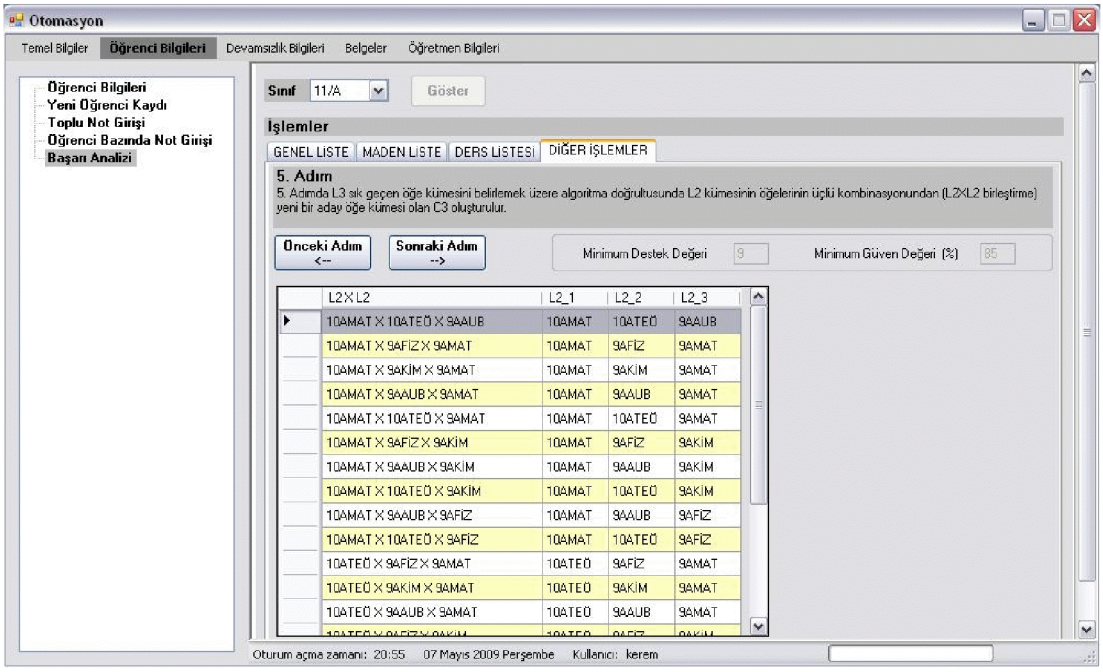
Şekil IV.26 C_2 aday öğe kümesi ekranı

Dördüncü adımda C_2 aday kümesi içinde yer alan ikili öğe setlerinin veri tabanında taraması yapılarak ikili öğelerin birlikte yer aldıkları kayıt sayısı destek değeri olarak belirlenir ve minimum destek değerine eşit veya yüksek olan C_2 aday öğe kümesi öğeleri L_2 sık geçen öğe kümesini oluşturur. Şekil IV.27'de dördüncü adımın sonucunda ortaya çıkan L_2 sık geçen öğe kümesinin 10 adet öğeden oluştuğu görülmektedir.



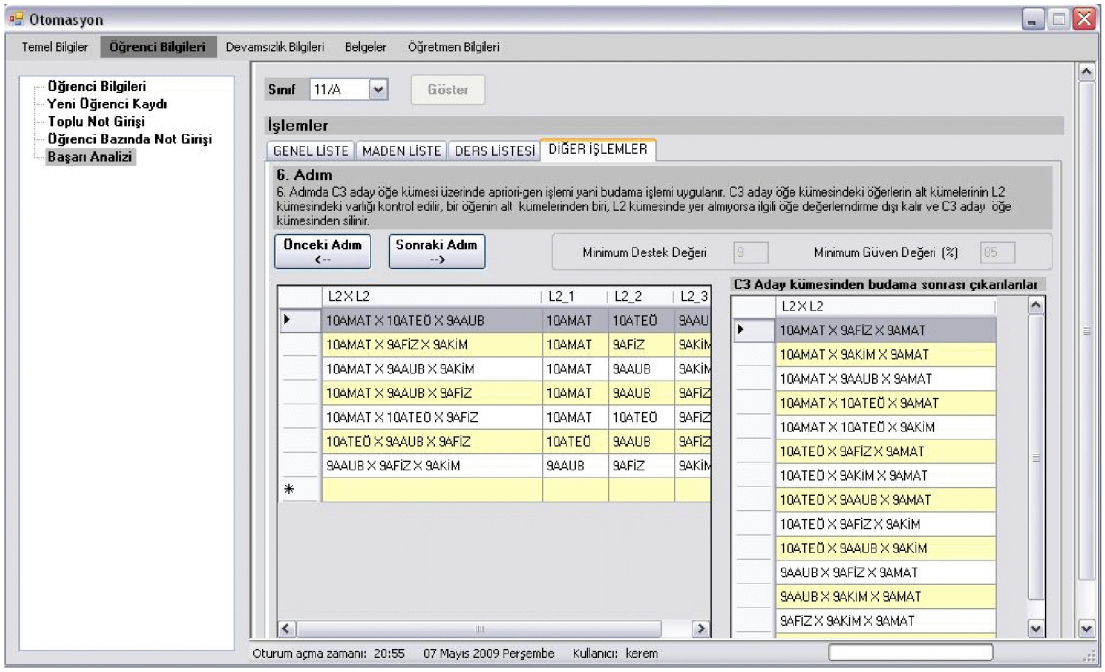
Şekil IV.27 L_2 sık geçen öğe kümesi ekranı

Beşinci adımda L_2 sık geçen öğe kümesinin öğeleri üzerinde birleştirme (join) işlemi uygulandığında ($L_2 \infty L_2$) C_3 aday öğe kümesi oluşur. Şekil IV.28'de C_3 aday öğe kümesinin 20 öğeden oluştuğu görülmektedir.



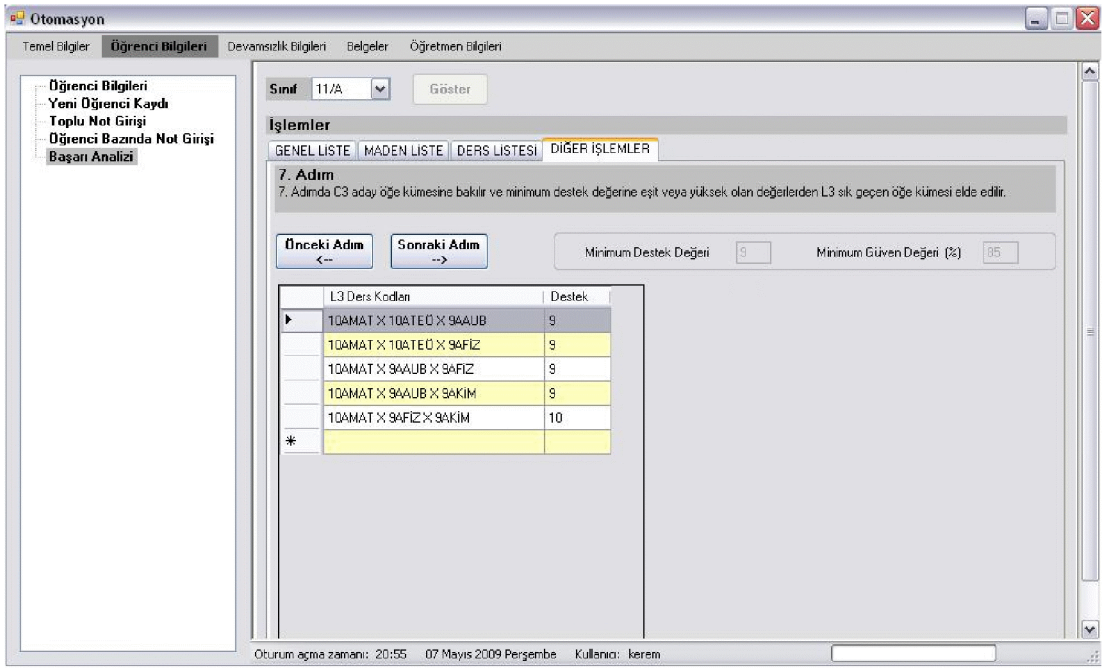
Şekil IV.28 C_3 aday öğe kümesi ekranı

Altıncı adımda C_3 aday kümesi üzerinde budama (prune) işlemi uygulanır. C_3 aday öğe kümesindeki öğelerin alt kümelerinin L_2 sık geçen öğe kümesindeki varlığı kontrol edilir, herhangi bir öğenin alt kümelerinden biri L_2 sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_3 aday öğe kümesinden silinir. Budama işlemi sonrasında elenen öğeler de ekranda kullanıcıya gösterilir. Şekil IV.29'daki budama işlemi sonucunda oluşan C_3 aday öğe kümesinin öğe sayınsın 7 olduğu, budama sonrası çıkarılan öğe sayısının ise 13 olduğu görülmektedir.



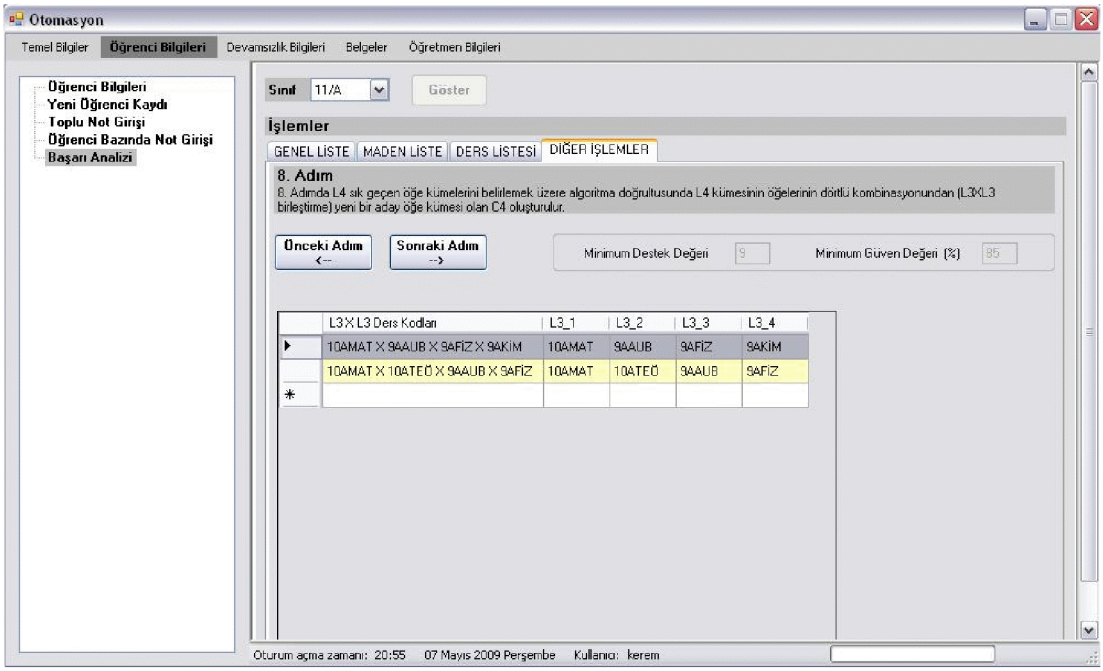
Şekil IV.29 C_3 aday öğe kümesi (budama sonrası) ekranı

Yedinci adımda C_3 aday kümesi içinde yer alan üçlü öğe setlerinin veri tabanında taraması yapılarak üçlü öğelerin birlikte yer aldıkları kayıt sayısı destek değeri olarak belirlenir ve minimum destek değerine eşit veya yüksek olan C_3 aday öğe kümesi öğeleri L_3 sık geçen öğe kümesini oluşturur. Şekil IV.30'da L_3 sık geçen öğe kümesinin 5 adet öğeden oluştuğu görülmektedir.



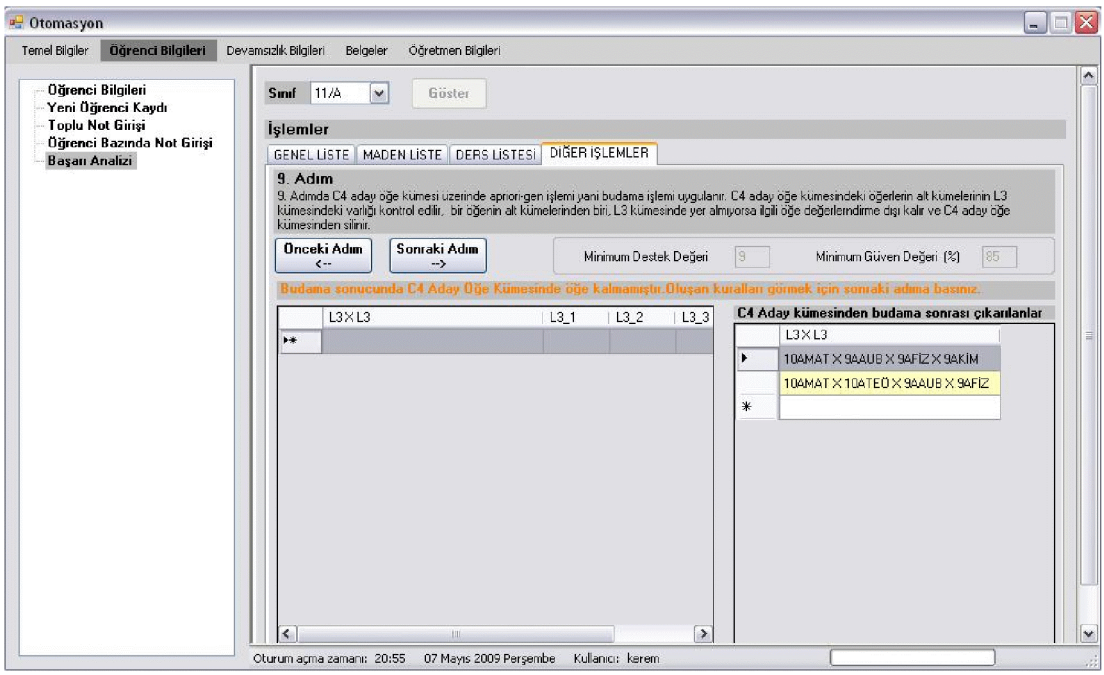
Şekil IV.30 L_3 sık geçen öğe kümesi ekranı

Sekizinci adımda L_3 sık geçen öğe kümesinin öğeleri üzerinde birleştirme (join) işlemi uygulandığında ($L_3 \infty L_3$) C_4 aday öğe kümesi oluşur. Şekil IV.31'de C_4 aday öğe kümesinin 2 öğeden oluştuğu görülmektedir.



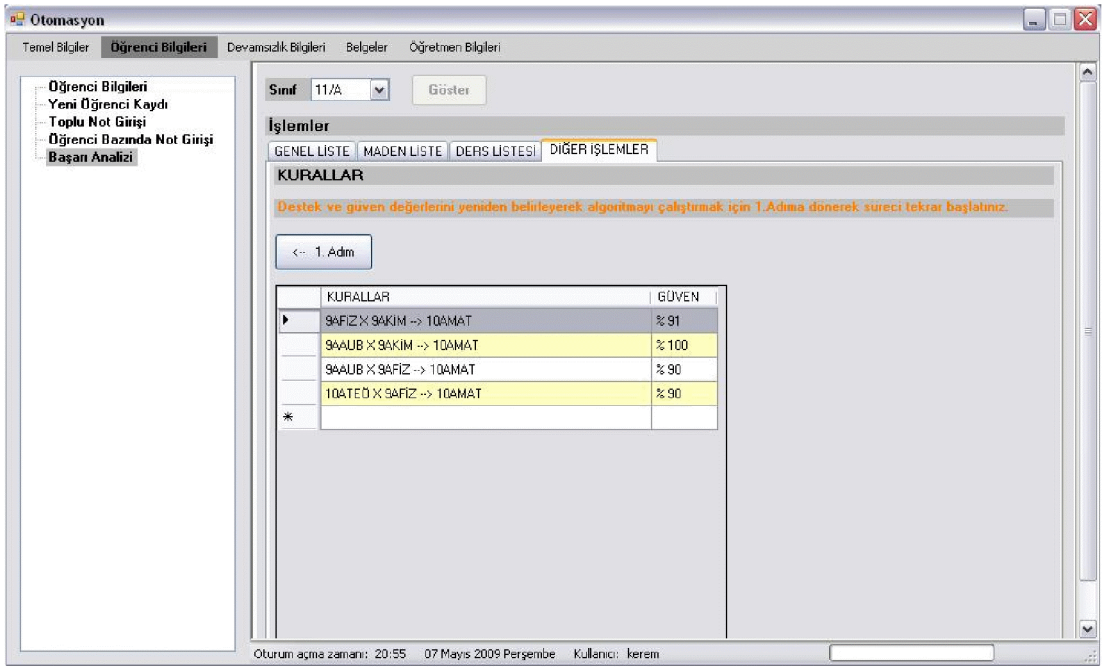
Şekil IV.31 C_4 aday öğe kümesi ekranı

Dokuzuncu adımda C_4 aday kümesi üzerinde budama (prune) işlemi uygulanır. C_4 aday öğe kümesindeki öğelerin alt kümelerinin L_3 sık geçen öğe kümesindeki varlığı kontrol edilir, herhangi bir öğenin alt kümelerinden biri L_3 sık geçen öğe kümesinde yer almıyor ise ilgili öğe değerlendirme dışı kalır ve C_4 aday öğe kümesinden silinir. Budama işlemi sonrasında elenen öğeler de ekranda kullanıcıya gösterilir. Şekil IV.32'de budama işlemi sonucunda oluşan C_4 aday öğe kümesinin tüm öğelerinin budandığı ve C_4 aday öğe kümesinde hiç öğe kalmadığı görülmektedir.



Şekil IV.32 C_4 aday öğe kümesi (budama sonrası) ekranı

Budama sonrasında C_4 aday öğe kümesinde hiç öğe kalmadığından, bu aşamada sonraki adım butonuna basıldığında Şekil IV.33'teki birliktelik kuralları görüntülenmektedir.



Şekil IV.33 Birliktelik kuralları ekranı

Bu sonuca göre C_4 aday öğe kümesinde budama sonrasında öğe kalmamıştır. Yedinci adımda elde edilen L_3 sık geçen öğe kümesi en sık kullanılan öğeler olarak belirlenir. L3 sık geçen öğe kümesinin 5 elemanı birlikte en sık başarısız olunan dersleri göstermektedir.

Bu kurallar incelendiğinde sayısal derslerde başarısız olan öğrencilerin bir sonraki sene sayısal derslerde yine başarısız olduğu sonucuna ulaşabiliriz. Özellikle dokuzuncu sınıftaki sayısal derslerde başarısız olan öğrencilerin onuncu sınıftaki matematik dersinde de başarısız oldukları gözlenmektedir.

(bkz:Şekil IV.33'teki) kurallara göre 11- A sınıfı öğrencilerinden;

- 9. sınıftaki kimya ve fizik derslerinden başarısız olanların % 91'inin 10. sınıf matematik dersinden de başarısız oldukları,
- 9. sınıf astronomi ve uzay bilimleri dersinden ve kimya dersinden başarısız olanların % 100' ünün 10. sınıf matematik dersinden başarısız oldukları,
- 9. sınıf astronomi ve uzay bilimleri dersinden ve fizik dersinden başarısız olanların % 90' ının 10. sınıf matematik dersinden başarısız oldukları,
- 9. sınıf temel elektronik ve ölçme dersinden ve fizik dersinden başarısız olanların % 90' ının 10. sınıf matematik dersinden başarısız oldukları görülmektedir

Program tekrar çalıştırılıp minimum destek değeri 7 ve minimum güven değeri (%) 90 olarak girildiğinde algoritma bu sefer hiç kesilmeden L_4 seviyesine kadar ulaşmıştır. Elde edilen L_4 sık geçen öğe kümesi en sık kullanılan öğeler olarak belirlenir. Bunun sonucunda Tablo IV.2'deki kural elde edilmiştir.

Tablo IV.2 Kurallar

KURALLAR	GÜVEN
9AAUB X 9AFİZ X 9AKİM --> 10AMAT	100%

Elde edilen bu kurala göre 9. sınıftaki astronomi ve uzay bilimleri, fizik ve kimya derslerinden başarısız olanların tamamının 10. sınıf matematik dersinden de başarısız oldukları görülmektedir.

BÖLÜM V

DEĞERLENDİRME ve ÖNERİLER

Veri Madenciliği, eğitim alanındaki elde edilen verilerden değerli bilgileri ortaya çıkartarak, eğitim öğretim kalitesinin artırılması açısından büyük katkılar sağlar.

Bu çalışmada veri madenciliği algoritmalarından Apriori algoritmasının eğitim alanında nasıl kullanılabileceği konusunda örnek bir çalışma yapılmış ve sonuçları gözlenmiştir. Apriori algoritması kullanılarak öğrenci ders notları ile ilgili bazı kurallar çıkarılmıştır. Apriori algoritması, veri madenciliğinde sık geçen öğelerin keşfedilmesi için kullanılan en çok bilinen algoritmadır. Bu algoritmada sık geçen öğe kümelerini bulmak için birçok kez veritabanını taramak gerekir. Özellikle Pazar sepeti analizinde yoğun olarak kullanılmaktadır.

Algoritmanın veriler üzerinde çalışması için bir okul otomasyonu yazılımı hazırlanmıştır. Bu yazılım genel amaçlı olup veri tabanından birliktelik kurallarını çıkarmaya yönelik hazırlanmıştır. Bu nedenle genel amaçlı bir veri tabanı hazırlanmıştır. Yazılım sayesinde veri tabanına öğrencilerin ve okulun tüm bilgileri aktarılmıştır. Daha sonra bu veriler kullanılarak birliktelik kuralları elde edilmiştir. Yazılım, algoritma süreçlerinin çalıştırılmasına imkan tanıyacak şekilde gerçekleştirilmiştir. Bu sayede algoritmanın her bir aşamasında kullanılan tekniklerin ve her bir aşamanın rahatça izlenmesine olanak sağlanmıştır.

Ortaya çıkan birliktelik kurallarının güvenilirliği, destek ve güven değerlerinin arttırılması ile artmaktadır. Tüm birliktelik kuralları kesinlikle kullanılabilir veya kullanılamaz diye bir yargıya varılması doğru değildir. Bu kurallar değerlendirilerek, kullanılabilir olanlar ayıklanmalıdır.

Bu çalışmanın önemli özelliklerinden bir tanesi de okuldaki öğrencilerin genelinde değil, her sınıf için ayrı ayrı uygulanabilir olmasıdır. Tüm sınıflardaki öğrencilerin aynı seviyede olmaları beklenemez. Farklı sınıflardaki öğrencilerin aynı dersten başarı durumlarının farklılık göstermesi olağandır. Bu çalışmada ders

birliktelikleri sınıf bazında yapılarak her bir sınıf için ayrı birliktelik kuralları ortaya çıkarılması esnekliği sağlanmıştır.

Veri madenciliği, eğitim kurumlarında eğitim-öğretim faaliyetlerinin etkili bir şekilde yürütülebilmesi, öğrenci performanslarının sürekli olarak izlenmesi, öğrencilerin yeteneklerine ve özelliklerine göre ders ve meslek seçiminde yönlendirilmesi için etkili bir şekilde kullanılabilir. Böylece öğrenci başarı düzeyi artırılabilir. Öğrencilerin başarısız oldukları derslerdeki birliktelikler değerlendirilerek bu durumun ortadan kaldırılmasına yönelik değişik stratejiler belirlenebilir. Öğrencilerin başarısız oldukları dersler günün hangi saatinde işleniyor, hangi dersten önce işleniyor ya da hangi dersten sonra işleniyor, o dersin öğretmenleri kimlerdir, dersin işleniş yöntemleri nelerdir gibi sorularla başarısızlığın nedenleri araştırılabilir. Bu nedenler sonucunda dersin işlenme gününde, saatinde, sınav tarihlerinde, ders işleme materyallerinde değişikliğe gidilebilir. Okulda seçmeli dersler okutuluyor ise elde edilen kurallar değerlendirilerek öğrencilerin gelecekte ders seçimi aşamasında bir kıstas olarak kullanılabilir. Rehber öğretmen tarafından öğrenci başarısız olabileceği bir ders yerine başka bir derse yönlendirilebilir. Böylece öğrencilerin daha iyi bir eğitim alması ve derslerde daha başarılı olması sağlanabilir.

Bu çalışmada hazırlanan yazılım geliştirilmeye açıktır. Daha gelişmiş, daha çok özelliklere sahip bir otomasyon yazılımı farklı yazılım dilleri ile de geliştirilebilir. Veri sayısının fazla olması, verilerin yazılıma yüklenme süresinin ve sorgu süresinin uzamasına neden olmuştur. Daha yüksek teknik özelliklere sahip bir bilgisayar ile veri madenciliği işlemleri daha hızlı bir şekilde yürütülebilir.

KAYNAKLAR

[1] Gürsakal, N.: "*Sosyal Bilimlerde Araştırma Yöntemleri*", VİPAŞ, Bursa, **(2001)** 189.

[2] Oğuzlar, A.: *"Veri Ön İşleme"*, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21, Temmuz – Aralık, **(2003)**

[3] Alataş, B.; Akın, E.: *"Veri Madenciliğinde Yeni Yaklaşımlar"*, YA/EM'2004 – Yöneylem Araştırması/Endüstri Mühendisliği – XXIV Ulusal Kongresi, Gaziantep – Adana, 15-18 Haziran, **(2004).**

[4] Sever, H.; Oğuz, B.: "Veri Tabanlarında Bilgi Keşfine Formal Bir Yaklaşım, Kısım1: Eşleştirme Sorguları ve Algoritmalar", *Bilgi Dünyası*, 3(2), Ekim, **(2002)** 173-204.

[5] Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: "The KDD Process for Extracting Useful Knowledge From Volumes of Data", *Communications of ACM*, 39(11), **(1996)** 27-34.

[6] Alpaydın E.: "Zeki Veri Madenciliği: Ham Veriden Altın Veriye Ulaşma Yöntemleri", *Bilişim 2000 Eğitim Semineri.*

[7] Frawley, W.J.; Piatetsky-Shapiro, G.; Matheus, C.J.: "Knowledge discovery databases: An overview. G. Piatetsky-Shapiro, W.J.Frawley 29(eds.). Knowledge discovery in databases", MA, AAAI/MIT, Cambridge, **(1991)** 1-27.

[8] Agrawal, R.; Srikant, R.: "Mining sequential Patterns.", *11th International Conference on Data Engineering,* Taipei, Taiwan, **(1995).**

[9] Holsheimer, M.; Siebes, A.P J.M.: " Data Mining: the search for knowledge in databases", *Technical Report CS-R9406*, Amsterdam, The Natherlands, **(1994).**

[10] Vahaplar, A.; İnceoğlu, M.M.: "Veri Madenciliği ve Elektronik Ticaret", *Ege Üniversitesi Bilgisayar Müh. Böl.*, **(2001)** 2-6.

[11] Özçakır, F.C.: "Müşteri işlemlerindeki birlikteliklerin belirlenmesinde veri madenciliği uygulaması", *Yüksek Lisans Tezi*, Marmara Üniversitesi. Fen Bilimleri Enstitüsü, İstanbul, Türkiye, **(2006)**

[12] EKER, H.: "Veri Madenciliği Veya Bilgi Keşfi", **(2007)** http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=538. (17.02.2009).

[13] Akpınar H.: *"Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği"*, İ.Ü. İşletme Fakültesi Dergisi, Nisan 2000 sayısı, 29, Nisan **(2000)** 1.

[14] Han, J.; Kamber,M.: "Data Mining Concepts And Tecniques", *Morgan Kaufmann Publishers*, San Francisco, USA, **(2000)**

[15] Döşlü, A.: "Veri Madenciliğinde Market Sepet Analiziz Birliktelik Kurallarının Belirlenmesi", *Yüksek Lisans Tezi*, Yıldız Teknik Üniversitesi. Fen Bilimleri Enstitüsü, İstanbul, Türkiye, **(2008)**

[16] Ding, Q.: "Association Rule Mining on Remotely Sensed Imagery Using P-Trees", *PhD Thesis*, Fargo, North Dakota, April **(2002)**

[17] Özekes, S.: *"Veri Madenciliği Modelleri ve Uygulama Alanları"*, İstanbul Ticaret Üniversitesi Dergisi, 3, **(2003)** 65-82.

[18] Zaki, M.J.; Ogihara, M.: "Theorical foundations of association rules.", *3rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD),* Seattle, WA, June **(1998)**

[19] Agrawal, R.; Srikant, R.: "Fast Algorithms for Mining Association Rules", *Proceedings of the VLDB,* Santiago de Chile, September **(1994)**

ÖZGEÇMİŞ

KEREM ÜÇGÜN

1983 yılında İzmit'te doğdum. İlkokula İzmit Abidin Pak İlkokulu'nda başladım. Ortaokulu İzmit Seka Çocuk Dostları Ortaokulunda tamamladım. Liseyi Kocaeli Sabancı Anadolu Teknik Lisesi Bilgisayar Bölümü'nde tamamladım. 2002 yılında girdiğim Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar ve Kontrol Öğretmenliği Bölümü'nden 2006 yılında mezun oldum. 2006 yılında Marmara Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Anabilim Dalı, Bilgisayar Kontrol Eğitimi Programı'nda yüksek lisans eğitimime başladım.

2007 yılında İstanbul Eyüp İ.M.K.B Ticaret Meslek Lisesinde Bilişim Teknolojileri öğretmeni olarak öğretmenlik hayatıma başladım. Halen aynı okulda Bilişim Teknolojileri Öğretmeni olarak görevime devam etmekteyim.