

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ZEKİ ÖĞRETİM SİSTEMLERİNDE VERİ
MADENCİLİĞİ KULLANILMASI

Buket DOĞAN
(Bilgisayar Teknik Öğretmeni, MSc.)

DOKTORA TEZİ
ELEKTRONİK – BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR – KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Prof.Dr. A.Yılmaz ÇAMURCU

İSTANBUL 2006

T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ZEKİ ÖĞRETİM SİSTEMLERİNDE VERİ
MADENCİLİĞİ KULLANILMASI

Buket DOĞAN, MSc.
(Bilgisayar Teknik Öğretmeni)
(141200420010272)

DOKTORA TEZİ
ELEKTRONİK – BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
BİLGİSAYAR – KONTROL EĞİTİMİ PROGRAMI

DANIŞMAN
Prof.Dr. A.Yılmaz ÇAMURCU

İSTANBUL 2006

ÖNSÖZ

Bu tez çalışmasında; sabrını, desteğini, rehberliğini ve zamanını esirgemeyen, her zaman bana yol gösteren tez danışmanım Prof. Dr. Yılmaz Çamurcu'ya, önerileriyle yol gösteren tez izleme komitesi üyeleri Yrd.Doç.Dr. Erbil Akbay'a ve Yrd. Doç.Dr. Melih İnal'a,

Bu tez Marmara Üniversitesi Bilimsel Araştırma Projeleri Komisyonu (BAPKO) tarafından FEN-DKR 250405-0112 nolu proje ile desteklenmiştir. Desteğinden dolayı, Marmara Üniversitesi Bilimsel Araştırma Projeleri Komisyonu'na,

Yardımlarından dolayı, Maltepe Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği öğretim görevlisi T. Tugay Bilgin'e,

Çalışmalarım sırasında bana olan inancı ve güveni ile destek olan eşime ve aileme,

Yardımlarını ve desteğini esirgemeyen değerli iş arkadaşlarıma,

teşekkürlerimi sunarım.

Ekim, 2006

Buket DOĞAN

İÇİNDEKİLER

	SAYFA
ÖNSÖZ.....	I
ÖZET.....	VII
ABSTRACT.....	IX
YENİLİK BEYANI.....	XI
SEMBOL LİSTESİ.....	XIII
KISALTMALAR.....	XIV
ŞEKİL LİSTESİ.....	XV
TABLO LİSTESİ.....	XIX
BÖLÜM I. GİRİŞ ve TEZİN AMACI	1
I.1. GİRİŞ	1
I.2. TEZ ÇALIŞMASININ AMACI.....	2
I.3. TEZ BÖLÜMLERİNİN KISA ÖZETİ	4
BÖLÜM II. ZEKİ ÖĞRETİM SİSTEMİ MİMARİSİ	6
II.1. GİRİŞ	6
II.2. ZEKİ ÖĞRETİM SİSTEMİNİN TANIMI	6
II.2.1. Alan Modeli	8
II.2.2. Öğrenci Modeli.....	8
II.2.2.1. Model İzleme	9

II.2.2.2. Kaplama Öğrenci Modeli.....	11
II.2.2.3. Kısıt Tabanlı Modelleme.....	13
II.2.2.4. Durum Tabanlı Sorgulama.....	15
II.2.2.5. Bayes Öğrenci Modeli	16
II.2.2.6. Streotip Öğrenci Modeli.....	17
II.2.2.7. Etmen Tabanlı Modelleme	18
II.2.3. Öğretim Modeli	19
II.2.4. Kullanıcı Arabirimi.....	20
II.2.5. MYCIN Uzman Sistemi Güven Faktörü Hesaplama Yöntemini Kullanan ZÖS Örnekleri.....	20

BÖLÜM III. VERİ MADENCİLİĞİ, BİRLİKTELİK

KURALLARI ve KÜMELEME ANALİZİ.....	22
III.1.GİRİŞ.....	22
III.2.VERİ MADENCİLİĞİ.....	22
III.2.1.Veritabanlarında Bilgi Keşfinin Aşamaları.....	24
III.2.2.Veri Madenciliğinde Teknikler	25
III.3.BİRLİKTELİK KURALLARI	27
III.4.KÜMELEME ANALİZİ.....	29
III.4.1.Kümeleme Analizinin Tanımı	29
III.4.2.K-Means Kümeleme Algoritması	30
III.5.ÇOK BOYUTLU VERİLERİN GÖRSELLEŞTİRİLMESİNDE PARALEL KOORDİNAT SİSTEMİNİN KULLANILMASI.....	33
III.6.PARVİS YAZILIMI İLE PARALEL KOORDİNATLARDA ÇOK BOYUTLU VERİLERİN GÖRÜNTÜLENMESİ	34
III.7.VERİ MADENCİLİĞİNİN EĞİTİM ALANINDA KULLANILMASINA İLİŞKİN LİTERATÜRDEKİ ÇALIŞMALAR	35

BÖLÜM IV. GERÇEKLEŞTİRİLEN ZEKİ ÖĞRETİM

SİSTEMİNİN TASARIMI	39
IV.1.AMAÇ	39
IV.2.ALAN MODELİ	40
IV.3.ÖĞRENCİ MODELİ	43
IV.4.ÖĞRETİM MODELİ.....	47

IV.5.KULLANICI ARABİRİMİ	51
IV.5.1.Kullanıcı Arabirimi Tasarımında Öğretim Açısından Dikkat Edilmesi Gereken İlkeler	51
 BÖLÜM V. VERİ MADENCİLİĞİ ANALİZ YAZILIMLARININ TASARIMI	54
V.1.GİRİŞ	54
V.2.VERİ MADENCİLİĞİNİ BİRLİKTELİK KURALI ANALİZİ İLE YAPAN YAZILIMLARIN TASARIMLARI	55
V.2.1. Amaç.....	55
V.2.2. ZÖS ve Birliktelik Kuralı Analizinin Genel Yapısı	57
V.2.3. Veri Madenciliğinde Birliktelik Kuralı Analizi için Kullanılan Apriori Algoritması.....	59
V.2.4. Kullanılan Veri Yapısı	66
V.2.5. Birliktelik Kuralı Bulan ZEBİKA Yazılımının Tasarımı	69
V.2.5.1.ZEBİKA Programının Yüklenme Aşamasında Yapılan İşlemler	71
V.2.5.2. Kural_say Alt Programı	72
V.2.5.3. Birleştir Alt Programı.....	72
V.2.5.4. Kural_Kontrol Alt Programı	73
V.2.5.5. Guven_Say Alt Programı	74
V.2.6. Çok Boyutlu Birliktelik Kuralı Bulan ZECOKA Yazılımının Tasarımı	75
V.2.6.1. Programın Yüklenme Aşamasında Yapılan İşlemler.....	78
V.2.6.2. Kural_say1 Alt Programı	79
V.2.6.3. Alt_Küme_Kontrol Alt Programı	79
V.2.6.4. Birleştir_say Alt Programı	80
V.2.6.5. Guven_Hesapla Alt Programı	81
V.3.KONU BAŞARI PUANLARININ KÜMELEME ANALİZİ	82
V.3.1. Yazılımda Kullanılan K-Means Kümeleme Algoritmasına Ait Ölçüt ve Hata Fonksiyonu	83
V.3.2. Kullanılan Veri Yapısı.....	84
V.3.3. Tekrarlı K-means Algoritmasına Göre Kümeleme Yapan Yazılım Tasarımı	85

V.4.VERİ MADENCİLİĞİ ANALİZ YAZILIMLARININ TEST SONUÇLARI.....	87
V.4.1. ZEBİKA'ya Ait Test Sonuçları.....	87
V.4.2. ZECOKA'ya Ait Test Sonuçları.....	88
V.4.3. Kümeleme Analizine Ait Test Sonuçları.....	91
BÖLÜM VI. SONUÇLAR.....	94
VI.1.AMAÇ	94
VI.2.TÜM ÖĞRENCİLERE AİT BULUNAN BİRLİKTELİK KURALLARI	95
VI.2.1. Tüm Öğrenci Verilerinde Aynı Konuda Yer Alan Sorulara Ait Bulunan Birliktelik Kuralları	95
VI.2.1.1.Giriş Konusu İçin Bulunan Birliktelik Kuralları	95
VI.2.1.2.Veriyolu Yapıları Konusu İçin Bulunan Birliktelik Kuralları ...	98
VI.2.1.3.Bellek Yapıları Konusu İçin Bulunan Birliktelik Kuralları.....	99
VI.2.1.4. İşlemci Yapısı ve Fonksiyonları Konusu İçin Bulunan Birliktelik Kuralları	100
VI.2.1.5.Komut Düzeyinde Paralellik Konusu İçin Bulunan Birliktelik Kuralları	101
VI.2.1.6.Komut Takımı ve Adresleme Modları Konusu İçin Bulunan Birliktelik Kuralları.....	103
VI.2.2. Konulardaki Başarı Durumlarına İlişkin Bulunan Birliktelik Kuralları	104
VI.2.3. Tüm Konularda Yer Alan Sorular Arasında Bulunan Çok Boyutlu Birliktelik Kuralları	105
VI.3.TÜM ÖĞRENCİLERE AİT VERİDE KÜMELEME ANALİZİ SONUÇLARI.....	108
BÖLÜM VII. VERİ MADENCİLİĞİ ANALİZ SONUÇLARININ EĞİTİM AÇISINDAN DEĞERLENDİRİLMESİ	112
VII.1.AMAÇ	112
VII.2.TÜM ÖĞRENCİLERİN BİRİNCİ ve İKİNCİ YANITLARINA AİT BİRLİKTELİK KURALLARININ KARŞILAŞTIRILMASI.....	113
VII.2.1.1.Giriş Konusu İçin Bulunan Birliktelik Kuralları	113

VII.2.1.2.Veriyolu Yapıları Konusu İçin Bulunan Birliktelik Kuralları	114
VII.2.1.3.Bellek Yapıları Konusu İçin Bulunan Birliktelik Kuralları....	115
VII.2.1.4.İşlemci Yapısı ve Fonksiyonları Konusu İçin Bulunan Birliktelik Kuralları.....	116
VII.2.1.5.Komut Düzeyinde Paralellik Konusu İçin Bulunan Birliktelik Kuralları	118
VII.2.1.6.Komut Takımı ve Adresleme Modları Konusu İçin Bulunan Birliktelik Kuralları.....	119
VII.3.TÜM ÖĞRENCİLERİN BİRİNCİ ve İKİNCİ YANITLARINA AİT ÇOK BOYUTLU BİRLİKTELİK KURALLARININ KARŞILAŞTIRILMASI	120
VII.4.EĞİTİM AÇISINDAN KÜMELEME ANALİZİNE AİT SONUÇLARIN DEĞERLENDİRİLMESİ.....	122
VII.5.EĞİTİM AÇISINDAN ÇOK BOYUTLU KÜMELEME SONUÇLARININ GÖRÜNTÜLENMESİ	123
BÖLÜM VIII. DEĞERLENDİRME VE ÖNERİLER.....	128
VIII.1.DEĞERLENDİRME	128
VIII.2.ÖNERİLER.....	130
KAYNAKLAR.....	131

ÖZET

ZEKİ ÖĞRETİM SİSTEMLERİNDE VERİ MADENCİLİĞİ KULLANILMASI

Eğitim-öğretim faaliyetlerinde yardımcı ders yazılımı olarak kullanılan, zeki öğretim sistemleri (ZÖS), öğrencinin seviyesine göre bireysel davranması özelliği ile eğitim-öğretim faaliyetlerine olumlu katkı sağlamaktadır. Öğrenciyle ilgili farklı verileri içeren bu tür sistemlerde yer alan verilerin analiz edilmesi; bu sistemlerden üst düzeyde yararlanmayı, bulunan analiz sonuçlarına göre sistemin geliştirilmesine devam edilmesini, öğrencilere sınıf ortamında bu sonuçlara göre kılavuzluk yapılmasını sağlayacaktır.

Veri analizinde ise, veriden kullanışlı, önceden bilinmeyen ve anlaşılır bilgiler elde etmek için veri madenciliği kullanılmaktadır. Veri madenciliği yöntemlerinin, zeki öğretim sistemlerindeki öğrenci verisi üzerinde kullanılması ile, bu sistemlerdeki verilerin analiz edilmesi mümkündür.

Bu tez çalışmasında, zeki öğretim sisteminde veri madenciliği uygulamaları için dört ayrı yazılım tasarlanmıştır.

Bu yazılımlardan birincisi, kaplama öğrenci modeli kullanan zeki öğretim sistemidir. Bu sistemde, konu alanı olarak Marmara Üniversitesi, Teknik Eğitim Fakültesi, Elektronik-Bilgisayar Eğitimi Bölümü'nde 8.dönemde yer alan “Bilgisayar Sistemleri” dersi belirlenmiştir. Sistemi, örgün ve ikinci öğrenci öğretimde toplam 103 öğrenci kullanmıştır. Zeki öğretim sisteminden alınan verilerin analiz edilmesi için ise, veri madenciliği tekniklerinden “birliktelik kuralları” ve “kümeleme analizi” kullanılmıştır.

İkinci yazılımda, veri madenciliğini tek boyutlu birliktelik kuralı analizi ile gerçekleştiren program tasarlanmıştır. Bu program, ZÖS'den alınan verilerde, aynı konu içerisinde birlikte yanlış yapılan veya doğru yapılan soruları tek boyutlu birliktelik kuralı analizi ile bulmaktadır. Bu program ile, öğrencilerin birlikte başarılı veya başarısız olduğu konuların da bulunması mümkündür.

Üçüncü yazılım ise, herhangi bir sorudaki yanlış veya doğru cevabın farklı konulardaki diğer soruları ne şekilde etkilediğini bulmak için analiz yapmaktadır. Bu program, ZÖS'den alınan öğrenci verilerinde farklı konulardaki sorular arasında, birlikte yanlış veya doğru yapılan soru dizilerini çok boyutlu birliktelik kuralı analizi ile bulmaktadır. Geliştirilen programlarda farklı destek değerleri ile analiz yapılabilmektedir ve analiz sonucunda çıkan kuralların güven değerleri programda hesaplanmaktadır.

Dördüncü yazılımda, kümeleme analizi için, k-means algoritmasını kullanan program tasarımı gerçekleştirilmiştir. Bu program ile, öğrencilerin ZÖS'de konulardan aldıkları başarı puanlarına göre, kümeleme işlemi gerçekleştirilmektedir. Kümeleme analizi sonucunda, öğretmen istediği konulardaki başarı puanlarına ait öğrenci dağılımını görerek, sınıfta oluşan öğrenci kümelerinin ne şekilde yoğunlaştığını belirler ve bu sonuçlara göre öğretim etkinliklerini düzenler. Kümeleme işleminde, k-means algoritmasının rasgele başlangıç merkezi seçiminden etkilenmesini engellemek için, tekrarlı yapıda çalışan k-means algoritması tasarlanmış ve programda kullanılmıştır. Kümeleme işlemi, öğretmen tarafından belirlenen konulardaki başarı puanları için yapılmaktadır. Belirlenen herhangi iki konuya ait başarı puanlarına göre kümeleme yapıldıktan sonra, sonuçların görselleştirilmesi için tasarlanan kümeleme programı kullanılmaktadır. Kümeleme işlemi için belirlenen konu sayısı üç veya daha fazla ise, k-means programından alınan kümeleme analizi sonuçlarının görselleştirilmesi için, paralel koordinat sistemi kullanılmıştır. Bu yöntem ile çok boyutlu veriler, anlaşılır, kullanılabilir ve kolayca sonuç çıkarılabilecek şekilde, boyut indirgemesi ve veri kaybı olmadan görselleştirilmiştir.

Veri madenciliğinin ZÖS'de kullanılmasına ilişkin bu tezin, öğrenciler üstündeki uygulama sonuçları eğitim açısından değerlendirilmiştir.

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.

ABSTRACT

USING DATA MINING ON INTELLIGENT TUTORING SYSTEMS

Intelligent tutoring systems(ITS) have been used in teaching and learning activities as a supplementary software. These systems make contributions in education activities, because they can behave adaptively to the students' learning level and they are capable of collecting vast amounts of data about students. Analyzing data which resides in ITS provides more benefit from these systems. The teacher can improve ITS and guide students in the classroom according to the analyze results.

For analyzing data and getting useful, extracting unknown knowledge and understandable information, data mining is used. It is also possible to analyze data in ITS applying by data mining techniques.

In this thesis for we designed four different software to apply data mining techniques on ITS.

The first designed software is ITS which uses overlay student model. The domain model of ITS is "Computer Systems" course. It is 8th term course at Marmara University at Technical Education Faculty in Electronic-Computer Education Department. ITS is used by 103 students and students' data in this system is analyzed by association rule mining and clustering techniques.

The second software analyses and finds interesting association rules about students' correct or incorrect answers by single dimensional association rule mining. This program also finds interesting association rules about concepts which are students successful or unsuccessful together.

The third software finds interesting association rules about correct or incorrect answers in different concepts by multidimensional association rule mining. Designed software can make analysis with different support values and also confidence values are calculated in the program.

The fourth software makes data mining application using clustering analysis. This program makes clustering analysis with k-means algorithm on the students' concept success value. Teacher can perceive distributions of students' concept success values and student clusters with this software. To prevent k-means algorithm initial randomly selected cluster center effect, recursive k-means algorithm is designed and used in the clustering software. Cluster analysis is made for concepts which are defined by the teacher. For visualizing any two concept success values' clustering result, designed program is used. If defined concepts are more than three, clustering analysis result is visualized on parallel coordinate system. This visualization method is understandable, usable and it doesn't require dimension reduction.

This thesis propose an application of data mining on ITS. The application results which were taken from students are demonstrated and evaluated for educational aspect.

The contribution of this study is using ITS data for data mining analysis . Data mining analysis in this study include single and multi dimensional association rule analysis, clustering analysis and visualization clustering results.

YENİLİK BEYANI

ZEKİ ÖĞRETİM SİSTEMLERİNDE VERİ MADENCİLİĞİ KULLANILMASI

Bu tez çalışması, hiçbir lisansüstü tez çalışmasının kopyası değildir. Tüm literatür araştırması ve geliştirilen yazılımların tamamı tarafımızca hazırlanmıştır.

II. ve III. Bölümde ve diğer kısımlarda yer alan açıklamalarda gerekli referanslar verilmiştir.

Zeki öğretim sistemlerinde, öğrencilerin bilgi düzeyini belirlemek için yapılan sınavlar çeşitli tekniklerle değerlendirilmekte ve öğrenciye ait yanıtlar saklanmaktadır. Fakat, aynı konu içerisinde ya da farklı konular içerisinde bir soruyu yanlış yapan öğrencinin başka hangi soruları yanlış yanıtladığı veya bir soruyu doğru yanıtlayan öğrencinin başka hangi soruları doğru yanıtladığı belli değildir. Aynı şekilde, bir konudan başarısız olan öğrencilerin bu konuyla birlikte başka hangi konularda başarısız olduğunun da analiz edilmesi gerekmektedir. Özellikle konu tekrarının yapıldığı zeki öğretim sistemlerinde, her tekrardaki yanıtların benzer şekilde analizde edilmesi, öğretmene ve öğrenciye rehberlik edecek ve sınıftaki başarı veya başarısızlığın daha çok hangi soru gruplarında olduğunun belirlemesini sağlayacaktır. Bu tip bir analizi, istatistiksel yöntemlerle elde etmek mümkün değildir. Bu sorunu çözüme ulaştırmak için; bu tez çalışmasında, veri madenciliği tekniklerinden tek boyutlu ve çok boyutlu birliktelik kuralı analizi yapılmıştır. Tarafımızca geliştirilen ve kaplama öğrenci modeli kullanan zeki öğretim sistemindeki veriler, bu tez çalışmasında tasarlanan birliktelik kuralı analiz yazılımlarında kullanılarak, üç farklı analiz sonucu elde edilmektedir.

- Birinci analiz sonucu, aynı konu içerisinde, öğrencilerin yaptıkları yanlış soru dizilerinin hangi sorular olduğuna dair bilgidir. Geliştirilen yazılımla, aynı analizi doğru sorular için de yapmak mümkündür.

- İkinci analiz, bir konudan başarısız olan öğrencilerin başka hangi konulardan başarısız olduğunu bulmaktadır. Aynı analiz, başarılı olunan konu dizilerinin bulunması için de yapılabilir.
- Üçüncü analiz sonucunda ise, öğrencilerin farklı konularda yaptıkları yanlış veya doğru soru dizileri çok boyutlu birliktelik kuralı analizi ile bulunmaktadır.

Ayrıca, konular düzeyinde oluşan öğrenci gruplarının oluşturulması ve konulardaki başarı puanlarının birbirini nasıl etkilediğinin belirlenmesi için öğrencilerin zeki öğretim sistemlerinde elde ettikleri konu başarı puanlarına göre dağılımlarının belirlenmesi gerekmektedir. Bu sorunu çözmek için, veri madenciliği tekniklerinden kümeleme analizi yapan yazılım tasarlanmıştır. Kümeleme analizi yazılımı; zeki öğretim sisteminden alınan verilerde, belirlenen konulardaki, başarı puanları ile, k-means algoritmasına göre kümeleme işlemini gerçekleştirmekte ve sonuçları Parvis isimli yazılımda görselleştirilmektedir. Bu analiz sonuçları, öğretmenin sınıfta oluşan öğrenci kümelerine göre öğretim etkinlikleri düzenleyip, zeki öğretim sisteminde gerekli güncellemeleri yapmasını sağlayacaktır.

Bu çalışmanın bilimsel katkısı yukarıda açıklanan sorunları çözmeye yönelik bir model önererek, geliştirilen veri madenciliği yazılımları ile, zeki öğretim sistemlerinde daha önce yapılmamış bir analizi gerçekleştirmesi ve zeki öğretim sisteminden doğru, anlamlı ve kullanışlı bilgilerin elde edilmesini sağlamasıdır. Bu tez çalışması; zeki öğretim sisteminde, veri madenciliği tekniklerinden, birliktelik kuralı analizi ve kümeleme analizinin yapılmasını sağlayarak, bu alana yenilik getirmiştir.

SEMBOL LİSTESİ

- k : küme sayısı.
- n : veri setindeki toplam nokta sayısı.
- L_k : k uzunluğa sahip sık tekrar eden küme.
- C_k : k uzunluğa sahip aday küme.
- GF_d : Soru doğru yanıtlandığında alınacak güven değeri.
- GF_y : Soru yanlış yanıtlandığında alınacak güven değeri.
- GF_{yeni} : Güncellenen yeni güven değeri.
- M_k : k . kümenin merkezini gösteren ortalama vektör.
- e_k : k . kümenin küme içi varyasyon değeri.
- E_k : k kümeye ayrılmış veri setindeki toplam hata.

KISALTMALAR

ZÖS : Zeki Öğretim Sistemi

DTS : Durum Tabanlı Sorgulama

GF : Güven Faktörü

ZEBİKA : Zeki Öğretim Sistemindeki Verilerle Tek Boyutlu Birliktelik
Kuralı Analizi Yapan Yazılım

ZECOKA : Zeki Öğretim Sistemindeki Verilerle Çok Boyutlu Birliktelik
Kuralı Analizi Yapan Yazılım

ŞEKİL LİSTESİ

SAYFA NO

Şekil I.1	Tasarlanan sistemin blok diyagramı.....	3
Şekil II.1	Zeki öğretim sistemi mimarisi [6,7]	7
Şekil II.2	Kaplama öğrenci modeli örneği [20].....	12
Şekil III.1	Veritabanlarında bilgi keşfi süreci [61]	24
Şekil III.2	K-means algoritması ile kümeleme örneği (küme sayısı=2)	32
Şekil III.3	Altı boyuta sahip verinin paralel koordinatlardaki görüntüsü	34
Şekil III.4	Parvis'e yüklenerek görselleştirilen örnek veri seti [103]	35
Şekil IV.1	Tasarlanan ZÖS'e ait mimari yapı	39
Şekil IV.2	Tasarlanan sistemin alan modelinin akış şeması.....	42
Şekil IV.3	Kaplama Öğrenci Modeli [116]	44
Şekil IV.4	Tasarlanan sisteme ait öğrenci modeli akış şeması.....	47
Şekil IV.5	Öğretim modelinin çalışmasına ait akış şeması	50
Şekil IV.6	Kullanıcı arabirimi örnek ekranı	51
Şekil V.1	ZÖS ve veri madenciliği	54
Şekil V.2	Gerçekleştirilen birliktelik kuralı analizi yazılımları ile bulunabilecek birliktelik kurallarının genel yapısı	57
Şekil V.3	ZÖS ve birliktelik kuralı analizinin genel yapısı ve işlem süreci	58
Şekil V.4	Apriori Algoritmasına ait sözde kod.....	61
Şekil V.5	Bellek yapıları konu sonu sınavındaki öğrenci yanıtlarını içeren metin dosyası.....	68
Şekil V.6	ZEBİKA yazılımına ilişkin tasarımına ait akış şeması.....	69
Şekil V.7	Kural_say alt programının algoritması	72
Şekil V.8	Birleştir alt programının algoritması.....	73
Şekil V.9	Kural_kontrol alt programına ait algoritma	74

Şekil V.10 Guven_say alt programına ait algoritma.....	75
Şekil V.11 Çok boyutlu birliktelik kuralına ait sözde kod	76
Şekil V.12 ZECOKA yazılımına ilişkin akış şeması.....	77
Şekil V.13 Kural_say1 alt programının algoritması	79
Şekil V.14 Alt_küme_kontrol al programına ait algoritma	80
Şekil V.15 Birleştir_say alt programına ait algoritma	81
Şekil V.16 Guven_Hesapla alt programına ait algoritma	81
Şekil V.17 K-means kümeleme algoritması için kullanılan verisetinin ekran görüntüsü.....	85
Şekil V.18 Tekrarlı çalışan K-means algoritmasına göre kümeleme işlemi yapan yazılımın algoritması.....	86
Şekil V.19 ZEBİKA'nın test verisi ile yapılan analiz sonuçları.....	87
Şekil V.20 ZEBİKA'nın test verisi ile yapılan güven değeri analizi.....	88
Şekil V.21 ZECOKA'nın test verisi ile yapılan analiz sonuçları	90
Şekil V.22 ZECOKA'nın test verisi ile yapılan güven değeri analizi	91
Şekil V.23 Standart k-means algoritmasının birinci test verisi ile analiz sonucu.....	92
Şekil V.24 Tekrarlı çalışan k-means algoritmasının birinci test verisi ile analiz sonucu	92
Şekil V.25 Standart k-means algoritmasının ikinci test verisi ile analiz sonucu.....	93
Şekil V.26 Tekrarlı çalışan k-means algoritmasının ikinci test verisi ile analiz sonucu	93
Şekil VI.1 Giriş konusu için bulunan birliktelik kuralları	96
Şekil VI.2 Giriş konusu güven değeri hesaplaması için analiz ekranı	97
Şekil VI.3 Veriyolu yapıları konusu için bulunan birliktelik kuralları.....	98
Şekil VI.4 Veriyolu yapıları konusu için güven değeri analizi	99
Şekil VI.5 Bellek yapıları konusu için bulunan birliktelik kuralları	99
Şekil VI.6 Bellek yapıları konusu için güven değeri analizi	100
Şekil VI.7 İşlemci yapısı ve fonksiyonları konusu için bulunan birliktelik kuralları	100
Şekil VI.8 İşlemci yapısı ve fonksiyonları konusu için güven değeri analizi	101
Şekil VI.9 Komut düzeyinde paralellik konusu için bulunan birliktelik kuralları ..	102
Şekil VI.10 Komut düzeyinde paralellik konusu için güven değeri analizi	102
Şekil VI.11 Komut takımı ve adresleme modları konusu için bulunan birliktelik kuralları	103

Şekil VI.12 Komut takımı ve adresleme modları konusu için yapılan güven değeri analizi	104
Şekil VI.13 Konulardaki başarı durumlarına ait bulunan birliktelik kuralları	105
Şekil VI.14 Konulardaki başarı durumlarına ait güven değeri analizi	105
Şekil VI.15 ZECOKA ile bulunan çok boyutlu birliktelik kuralları	106
Şekil VI.16 Çok boyutlu birliktelik kuralına ait güven değeri analizi.....	107
Şekil VI.17 Kümeleme analizi programına ait ekran görüntüsü	108
Şekil VI.18 Giriş ve veriyolu yapıları konusuna ait kümeleme analizi sonuçları ..	109
Şekil VI.19 Bellek yapıları ile işlemci yapısı ve fonksiyonları konularına ait kümeleme analizi sonuçları.....	110
Şekil VI.20 İşlemci yapısı ile komut düzeyinde paralellik konusu için yapılan kümeleme analizi sonuçları.....	110
Şekil VI.21 İşlemci yapısı ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizi sonuçları	111
Şekil VI.22 Komut düzeyinde paralellik ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizi sonuçları	111
Şekil VII.1 Giriş konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	113
Şekil VII.2 Giriş konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	114
Şekil VII.3 Veriyolu yapıları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	114
Şekil VII.4 Veriyolu yapıları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	115
Şekil VII.5 Bellek yapıları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	116
Şekil VII.6 Bellek yapıları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	116
Şekil VII.7 İşlemci yapısı ve fonksiyonları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi.....	117
Şekil VII.8 İşlemci yapısı ve fonksiyonları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	117
Şekil VII.9 Komut düzeyinde paralellik konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	118

Şekil VII.10 Komut düzeyinde paralellik konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi	119
Şekil VII.11 Komut takımı ve adresleme modları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi.....	119
Şekil VII.12 Komut takımı ve adresleme modları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi.....	120
Şekil VII.13 Tüm öğrenciler için öğrencilerin birinci yanıtlarına ait veri ile yapılan çok boyutlu birliktelik kuralı analizi	121
Şekil VII.14 Tüm öğrenciler için öğrencilerin ikinci yanıtlarına ait veri ile yapılan çok boyutlu birliktelik kuralı analizi	121
Şekil VII.15 Kümeleme analizi sonuçlarının görselleştirilmesi.....	123
Şekil VII.16 Tüm öğrencilere ait başarı puanlarının Parvis'e yüklenmesi ile elde edilen ekran görüntüsü	125
Şekil VII.17 Tüm öğrencilerin başarı puanlarına ait veri üzerinde Parvis'de seçilen beşinci kümenin görüntülenmesi.....	126
Şekil VII.18 Tüm öğrencilerin başarı puanlarına ait veri üzerinde Parvis'de seçilen dördüncü kümenin görüntülenmesi.....	127

TABLO LİSTESİ

	<u>SAYFA NO</u>
Tablo IV.1 Tasarlanan zeki öğretim sisteminde yer alan konular.....	42
Tablo IV.2 TblSorular tablosunun içerdiği alanlar	43
Tablo IV.3 TblKullanıcılar Tablosuna Ait Alanlar	45
Tablo IV.4 TblPuan Tablosuna Ait Alanlar	45
Tablo IV.5. TblCevaplar Tablosuna Ait Alanlar.....	46
Tablo IV.6 Güven Değeri ve Anlamları.....	49
Tablo V.1 Öğrenci yanıtlarını içeren dizi yapısı	63
Tablo V.2 Apriori algoritmasında kullanılmak üzere oluşturulan işlemsel veri	63
Tablo V.3 Destek Değeri=2 ile aday küme ve sık tekrarlanan kümelerin oluşturulması	64
Tablo V.4 Zeki öğretim sisteminde yer alan konular ve konu sonu sınavındaki soru adetleri.....	67
Tablo V.5 Çok boyutlu öğrenci yanıt dizisi	89
Tablo V.6 Test verisi ile bulunan tek boyutlu birliktelik kuralları.....	90
Tablo V.7 Test verisi ile bulunan iki boyutlu birliktelik kuralları	90
Tablo VI.1 Giriş konusunda bulunan birliktelik kurallarına ait sorular	97
Tablo VI.2 Veriyolu Yapıları konusunda bulunan birliktelik kurallarına ait sorular	98
Tablo VI.3 Bellek Yapıları konusunda bulunan birliktelik kurallarına ait sorular ..	99
Tablo VI.4 İşlemci yapısı ve fonksiyonları konusunda bulunan birliktelik kurallarına ait sorular.....	101
Tablo VI.5 Komut düzeyinde paralellik konusunda bulunan birliktelik kurallarına ait sorular.....	102
Tablo VI.6 Komut takımı ve adresleme modları konusunda bulunan birliktelik kurallarına ait sorular	103
Tablo VI.7 Çok boyutlu birliktelik kuralında yer alan sorular.....	107

BÖLÜM I

GİRİŞ ve TEZİN AMACI

I.1. GİRİŞ

Bilgisayar yazılımları 1970’li yıllardan itibaren öğretim faaliyetlerinde kullanılmaktadır [1]. Bu yazılımlar, ilk dönemlerde bilgisayar destekli eğitim adı altında incelenirken, 2000’li yıllarda daha bireysel davranabilen yazılımlar geliştirilmiştir. Araştırmacıların bu yazılımlar üzerinde çalışmasının en önemli nedeni, öğrenci sayılarının fazla ve mevcut eğitim-öğretim imkanlarının kısıtlı olmasıdır. Bu nedenle, gelişen bilgisayar ve İnternet teknolojileri yardımıyla, mümkün olduğunca bilgisayarın öğretimde kullanılması teşvik edilmektedir. Özellikle 1990’ların sonlarında araştırmacılar, öğretim faaliyetlerinde kullanılmak üzere daha etkin, öğrenci seviyesi ve ihtiyaçlarına göre uyarlanabilir yazılımlar geliştirilmeyi başarmışlardır. İçerisinde öğrenciyle ilgili çeşitli bilgileri tutan ve farklı öğrenci modellerine sahip olan bu öğretim yazılımları “zeki öğretim sistemleri” olarak adlandırılmaktadır [2]. Zeki öğretim sistemleri, öğrencinin bilgi seviyesi, tercihleri, cevapları gibi öğrenciye özel verileri içermektedir. Bu verilerin analiz edilmesi ve anlamlı bilgiler elde edilmesi; bu tip sistemlerin etkinliğinin artırılması, öğretime katkısının belirlenmesi ve gelişmesinin sağlanması açısından önemlidir. Sistemdeki verilerin analiz edilmesi; öğrencilerin güçlü veya zayıf oldukları konuların belirlenmesini, öğretmenin öğretim etkinliklerini bu sonuçlara göre sınıf ortamında da devam ettirmesini ve zeki öğretim sisteminde gerekli güncellemelerin yapılmasını sağlar.

Bu analiz işleminde, son yıllarda önemi iyice artan veri madenciliği analiz tekniklerini kullanmak, veriden ilginç ve anlaşılır sonuçların alınması için etkili bir

yöntemdir. Pazarlama, bankacılık, sigortacılık, tıp gibi birçok alandaki verilerin analiz edilmesinde kullanılan veri madenciliğinin, eğitim öğretim alanında da kullanılması bu alanda kullanılan eğitim yazılımlarına dair doğru ve anlamlı bilgilerin oluşturulmasını sağlayacaktır. Özellikle, zeki öğretim sisteminde; alan bilgisinin ve soruların güncellenmesi, birbiriyle ilişkili konu ve soruların belirlenmesi, sorulara ait puanların belirlenmesi gibi işlemlerde veri madenciliğinden faydalanılması mümkündür.

I.2. TEZ ÇALIŞMASININ AMACI

Öğrenciye göre uyarlamalı öğretim gerçekleştiren zeki öğretim sistemlerinde yer alan veriler aracılığı ile daha önce bilinmeyen bilgileri elde etmek, öğrenci ve öğretmen için faydalı olacaktır. Öğretmenin bulunan bilgileri inceleyerek, sistemdeki soruların ve konuların birbiriyle olan ilişkilerini saptaması ve sistemden geribildirim alması sağlanmış olacaktır. Bu analiz, zeki öğretim sisteminden öğrencilerin daha etkin şekilde faydalanmasını sağlayacak ve öğretmenin sistemdeki konuları ve soruları güncellemesinde yardımcı olacaktır. Verilerin analizi için, veritabanlarında yeni, önemli ve faydalı bilgileri bulmayı amaçlayan veri madenciliği analiz yöntemleri, anlaşılır ve kullanışlı bilgileri oluşturacaktır.

Özellikle ZÖS’de bulunan öğrenci yanıtları ve alınan değerlendirme sonuçları ile ilgili analiz yapılması, öğrenci yanıtları ve not dağılımı ile ilgili örüntülerin bulunmasını sağlayacaktır.

Öğrenci yanıtlarında birlikte doğru veya yanlış yapılan soru dizilerinin ve birlikte başarısız veya başarılı olunan konuların bulunması ile, sınıf ortamında ve ZÖS’de daha fazla önem verilmesi gereken, birbirini etkileyen konuların ve kavramların belirlenmesini sağlayacaktır. Böyle bir bilgiyi elde etmek için, öğrenci yanıtlarının veri madenciliği tekniklerinden birliktelik kuralı analizi ile incelenmesi gerekmektedir.

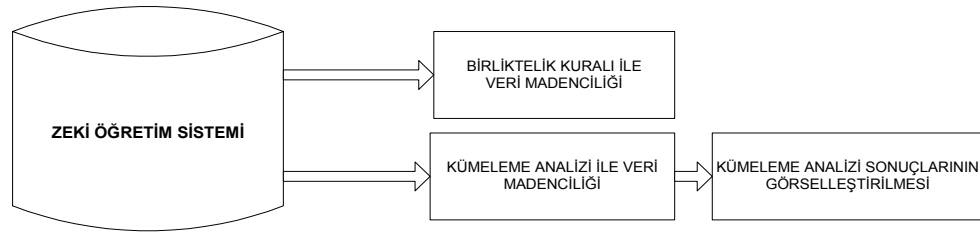
Öğrencilerin konulardaki değerlendirme sonuçlarına dair yapılan analiz ise, sınıfta öğrencilerin daha çok hangi alanlarda kümelandiklerinin belirlenmesini sağlayarak, ders konuları düzeyinde alınan puanlara göre dağılımı gösterecektir. Öğretmenin bu sonuçlara göre, öğrenciler için farklı öğretim stratejileri belirlemesi ve ZÖS’de gerekli güncellemeleri yapması mümkündür. Bu tip bir analizi

gerçekleştirmek için, öğrenci başarı puanlarının veri madenciliği tekniklerinden kümeleme ile analiz edilmesi gerekmektedir.

Fakat, mevcut veri madenciliği yazılımlarının birçoğunu kullanabilmek için; bu alanda uzman olmak, verileri yazılıma uygun hale getirmek ve algoritmayı belirlemek gerekmektedir. Ayrıca bu yazılımlar genel amaçlıdır, sadece belli bir alandaki verilere yönelik değildir. Hangi verilerin yazılımda kullanılacağı ve sonuçların ne şekilde yorumlanacağı da kullanıcıya kalmaktadır.

Bu problemleri çözmek için, bu tez çalışmasında aşağıda belirtilen maddeler amaçlanmıştır:

- Şekil I.1’de blok diyagramı görülen sistemde, farklı dersler için kullanılabilir, kaplama öğrenci modeli kullanan, zeki öğretim sistemi geliştirmek ve bu zeki öğretim sistemindeki verileri veri madenciliği ile analiz ederek veri madenciliğinin zeki öğretim sisteminde kullanılmasına ilişkin bir örnek oluşturmak,



Şekil I.1 Tasarlanan sistemin blok diyagramı

- Analiz işleminde; öğrenci yanıtlarına ait veride, aynı konu içerisinde birlikte yanlış veya doğru yapılarak tekrar eden soruları, veri madenciliği tekniklerinden birliktelik kuralı analizi ile bulan yazılımı tasarlamak,
- Farklı konular içerisinde yer alan öğrenci yanıtlarında birlikte yanlış veya doğru yapılarak tekrar eden soruları, çok boyutlu birliktelik kuralı analizi ile bulan yazılımı tasarlamak,
- Zeki öğretim sisteminde yer alan konulardaki başarı puanlarına göre, belirlenen konular için veri madenciliği tekniklerinden kümeleme analizini gerçekleştirecek yazılımı tasarlamak, bu kümeleme analizi sonuçlarını görselleştirmek,

I.3. TEZ BÖLÜMLERİNİN KISA ÖZETİ

Bu tez çalışması sekiz bölümden oluşmaktadır.

Bölüm II' de; zeki öğretim sisteminin tanımı, bileşenleri, kullanılan öğrenci modellerinden; model izleme, kaplama öğrenci modeli, kısıt tabanlı modelleme, durum tabanlı sorgulama, bayes öğrenci modeli, streotip öğrenci modeli ve etmen tabanlı modelleme açıklanmaktadır. Ayrıca bu bölümde, MYCIN uzman sistemindeki güven faktörü hesaplama yöntemini kullanan zeki öğretim sistemi örnekleri incelenmektedir.

Bölüm III'de veri madenciliğinin tanımı, temel veri madenciliği teknikleri incelendikten sonra birliktelik kurallarının tanımı, kümeleme algoritmalarından k-means algoritması, çok boyutlu verilerin görselleştirilmesinde kullanılan paralel koordinat sistemi açıklanmakta ve veri madenciliğinin eğitim alanından kullanılmasına ilişkin örnekler incelenmektedir.

Bölüm IV'de tasarlanan zeki öğretim sisteminin, alan modeli, öğrenci modeli, öğretim modeli ve kullanıcı arabirimi açıklanmaktadır. Alan modelinde bilginin ne şekilde temsil edildiği ve sunulduğu açıklanmaktadır. Öğrenci modelinde, öğrenci bilgisinin temsil edilmesinde kullanılan kaplama öğrenci modeli ve bu amaçla kullanılan veri tablolarının yapısı açıklanmaktadır. Öğretim modelinde ise, yapılan sınavlarla öğrencinin bilgi düzeyinin belirlenmesinde, MYCIN güven faktörü hesaplama yönteminin kullanımı açıklanmaktadır.

Bölüm V'de, ZÖS'den alınan veriler ile birliktelik kuralı analizi ve kümeleme analizi yapan veri madenciliği yazılımlarının yapısı açıklanmaktadır. Birliktelik kuralı analizi yapan yazılımlar başlığı altında, aynı konu içerisinde birlikte yanlış veya doğru yapılan soru dizilerini bulan ZEBİKA (Zeki Öğretim Sistemindeki Verilerle Birliktelik Kuralı Analizi) yazılımı ve farklı konular içerisinde birlikte yapılan yanlış veya doğru soruları çok boyutlu birliktelik kuralı analizi ile bulan ZECOKA (Zeki Öğretim Sistemindeki Verilerle Çok Boyutlu Birliktelik Kuralı Analizi) yazılımının tasarımı, kullanılan veri yapısı, kullanılan alt programlar incelenmektedir. Başarı puanlarının kümeleme analizi başlığı altında ise, k-means algoritması ile kümeleme analizi gerçekleştiren yazılım tasarımı, kullanılan veri yapısı ve alt programlar açıklanmaktadır.

Bölüm VI’da, birliktelik kuralı analiz yazılımları ve kümeleme analizi yazılımında ZÖS’den elde edilen verilerin kullanılmasıyla elde edilen sonuçlar sunulmaktadır.

Bölüm VII’de, öğrencilerin verdikleri birinci ve ikinci yanıtlara ait verilerin, tasarlanan veri madenciliği analiz yazılımlarında kullanılması ile elde edilen analiz sonuçları, eğitim açısından değerlendirilmektedir. Bu bölümde, çok boyutlu kümeleme analiz sonuçlarının görselleştirilmesi için kullanılan paralel koordinat sistemine ait sonuçlar da yer almaktadır.

Bölüm VIII’de ise, tasarlanan ZÖS ve veri madenciliği analiz yazılımlarına ait değerlendirme ve öneriler yer almaktadır.

BÖLÜM II

ZEKİ ÖĞRETİM SİSTEMİ MİMARİSİ

II.1. GİRİŞ

Bilgisayar teknolojilerindeki son gelişmeler; yapay zeka, sinir ağları, uzman sistemler gibi birçok konunun bilgisayar destekli öğretim sistemlerinde uygulanmasında etkili olmuştur ve öğrencinin öğrenme performansını artırmaya yönelik etkin yazılımlar geliştirilmiştir. Zeki öğretim sistemi (ZÖS) de, eğitim-öğretim alanında kullanılan yardımcı bilgisayar programlarından birisidir [2]. Bu bölümde, ZÖS mimarisi ve kullanılan modellere ilişkin literatürde yer alan çalışmalar açıklanmaktadır.

II.2. ZEKİ ÖĞRETİM SİSTEMİNİN TANIMI

1970’li yılların başından itibaren bilgisayar programlarının öğretim faaliyetlerini daha etkili hale getirmesi için çalışmalar yapılmaktadır. Bu programlardan birisi de, zeki öğretim sistemleridir. ZÖS, ne öğretileceği ve bu bilgilerin nasıl öğretileceği ile ilgili öğretim stratejilerini içeren ve öğrenciye göre sistemin dinamik olarak uyarlanmasını sağlayan bilgisayar tabanlı bir öğretim sistemidir [2].

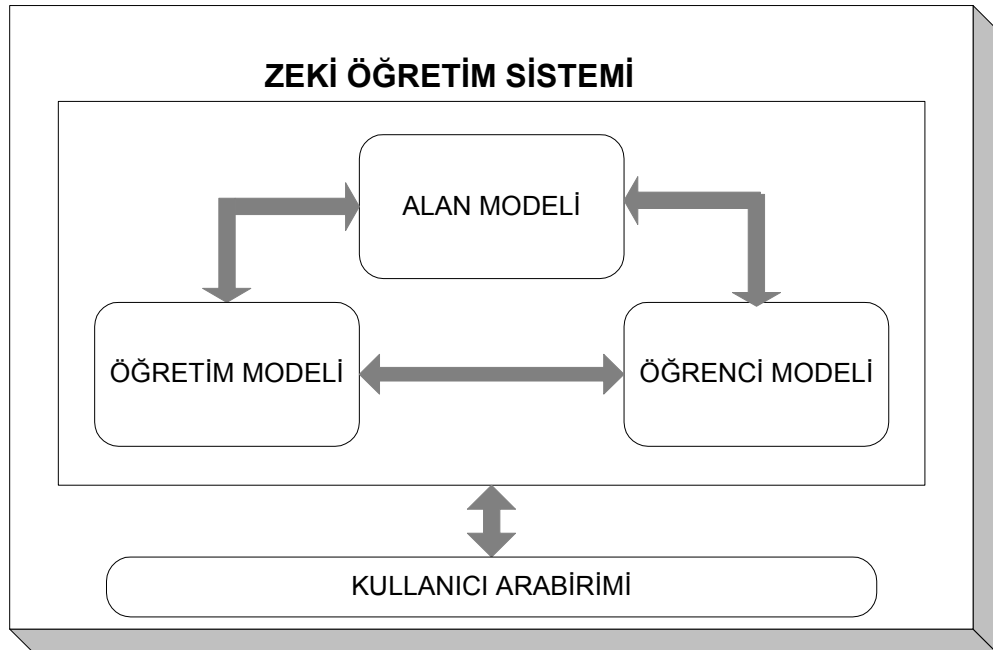
Başka bir tanıma göre ise; zeki öğretim sistemi bilgiyi temsil ederken ve öğrenci ile etkileşim sağlarken yapay zeka tekniklerini kullanan bilgisayar programıdır [3,4].

Zeki öğretim sistemi, belli bir düzeyde uyarlanabilirlik sağlayan bilgisayar tabanlı eğitim sistemlerinin bir sınıfıdır. Öğrenci, sistemi kullandıkça, sistem

öğrencinin ilerleyişine göre öğretim stratejisini değiştirerek uyarlanmış olmaktadır. Örneğin, sistem öğrenciyi izleyerek, öğrencinin o anki anlama durumuna göre dersin seviyesini ayarlamaya çalışmaktadır.

ZÖS, öğrencinin dinamik bir modelini çıkartabilmekte ve bunu ideal olan uzman model ile karşılaştırmaktadır. Bu yönleri ile ZÖS bilgisayar destekli öğretim sistemlerinden ayrılmaktadırlar. Öğrenci sistemde problem çözerken, sistem öğrencinin bilgi tabanını oluşturmakta ve uzman bilgisi ile kıyaslayarak öğrenci eğitimindeki stratejiyi değiştirmektedir [5].

Tipik bir ZÖS, Şekil II.1’de görüldüğü gibi dört bileşenden oluşmaktadır. Bunlar; alan modeli, öğrenci modeli, öğretim modeli ve kullanıcı arabirimidir [6,7]. Alan modeli, uzmana özel bilgileri bildiren bilgi veya işlemsel bilgi biçiminde içeren kısımdır. Öğrenci modeli, öğrencinin öğrenme düzeyini temsil eden ve öğrenciye özel bilgileri saklayan kısımdır. ZÖS’de bulunması gereken zeki özelliği, bireysel uyarlamanın sağlanmasıyla mümkün olmaktadır. Öğrenci süreçlerinin izlenmesi ve öğrenci modelinin uzman model ile karşılaştırılması ile; öğrencinin hangi konuda zorluk yaşadığı, nerede yardıma ihtiyacı olduğu ve hangi aşamada yeni bir konuya geçeceği belirlenebilir. Öğretim modeli ise, çeşitli öğretim stratejilerini içeren bileşendir. Bu model aracılığı ile öğrenciye geribildirim sağlanır ve en uygun konuya yönlendirilir. Kullanıcı arabirimi ise, öğrenci ve sistem arasındaki etkileşimi sağlamaktadır.



Şekil II.1 Zeki öğretim sistemi mimarisi [6,7]

II.2.1.Alan Modeli

Alan modeli, uzman bilgisindeki kuralları kullanarak problemleri çözmeye çalışmaktadır. Alan bileşeni, problemleri çözmeyi sağlayacak ve öğretim amaçlarını gerçekleştirecek nitelikte olmalıdır. Uzman bilgisi; semantik ağlar, çerçeveler, üretim sistemleri, konu içeriğinin bulunduğu örün sayfaları veya kısıtlamalar yoluyla sunulabilir. Alanın ne şekilde bilgi sunacağına karar vermek önemlidir, çünkü buna göre problem çözülmekte ve gerekli açıklama yapılmaktadır. Sunum, hem problemi çözecek kadar yeterli, hem de açıklama yapmak için kullanışlı olmalıdır [8, 9].

II.2.2.Öğrenci Modeli

Öğrenci modeli, her öğrenci için kişisel bilgiler içermekte ve böylece sistem kişisel ihtiyaçlara cevap vermektedir [10]. Öğrenci modeli; öğrencinin sistem ile olan etkileşiminden elde ettiği bilgilerle, öğrenci bilgisini temsil ederek, bu modeli geliştirmektedir. Öğrenci modeli dinamik olarak, sistem açısından öğrencinin güçlü ve zayıf olduğu yönleri gösterebilmelidir. Kusursuz bir öğrenci modeli; öğrenci için öğrenmeyi ve verimini etkileyecek tüm öğrenci davranışı ve bilgisine ait ayrıntıları içermelidir [7]. Pratikte, böyle bir modeli oluşturmak bilgisayar ortamında çok karmaşık olacağından dolayı, kısmi ve basitleştirilmiş modeller kullanılmaktadır.

Aynı zamanda öğrenci modeli, öğrencinin hangi konularda daha fazla yardım ve desteğe ihtiyacı olduğuyla ilgili özellikleri içermelidir. Öğrenci modeli ZÖS'ün en önemli parçasıdır. Eğer öğrenci için doğru bir modelleme yapılmaz ise, öğretim modelinde doğru bir karar verilemeyecektir. Öğrenci modelleme yöntemine ilişkin araştırmalar devam etmektedir [3, 8].

Öğrenci modelleri, öğrenci bilgisinin devamı için ne tür bilgi kullandıklarına bağlı olarak sınıflandırılabilir. Bazı öğrenci modelleri, yakın zaman bilgisini kullanırken, diğerleri daha uzun vadeli bilgileri kullanabilmektedirler. Her iki tür bilgiyi kullanan sistemler de bulunmaktadır. Yakın zaman bilgisi, öğrenciye verilecek anlık yardımlar için kullanışlı olabilmektedir. Uzun vadeli bilgiler ise, yeni bir problem seçimi veya öğrenci için en uygun konunun belirlenmesi gibi bazı pedagojik kararlar için kullanılabilir [8, 11].

Öğrenci modelleme teknikleri:

- Model izleme,
- Kaplama öğrenci modeli,

- Kısıt tabanlı modelleme,
- Durum tabanlı modelleme,
- Bayes öğrenci modeli,
- Streotip öğrenci modeli,
- Etmen tabanlı modelleme

şeklinde listelenebilir [11].

Model izleme ve kısıt tabanlı modelleme arasındaki fark; model izlemenin işlemsel ve bildirici bilgiyi temsil edebilmesine karşın, kısıt tabanlı modellemenin sadece bildirici bilgiyi sunmasıdır. Streotip ve kaplama modelleri öğrenci bilgisini farklı şekillerde temsil etmektedir. Streotip modelde, öğrenciler bazı özet sınıflar içinde gruplandırılmaktadır. Kapsama öğrenci modelinde ise; öğrenci bilgisi, uzman bilgisinin bir alt kümesi olacak biçimde temsil edilmektedir [11]. Literatürde geçen, öğrenci modellemeye ilişkin yapılmış çalışmalar aşağıdaki başlıklar altında açıklanmaktadır.

II.2.2.1. Model İzleme

Model izleme, Anderson ve arkadaşları tarafından geliştirilen ACT-R (Adaptive Control of Thought-Rational) teorisi üzerine kurulmuştur. ACT-R bilişsel teorisi, belleğin bildirici ve işlemsel olmak üzere iki farklı depolama yapısı olduğunu iddia etmektedir. Bildirici bilgi, olgulara ve gerçeklere dayanan daha çok teorik olarak ifade edilen bilgileri temsil etmektedir. Örneğin, matematik alanındaki teoremler buna bir örnektir. İşlemsel bilgi ise, amaç tabanlı bilgileri içermektedir. Örneğin, bir matematik teoreminin nasıl uygulanacağı işlemsel bir bilgi türüdür ve üretim kuralları ile temsil edilmektedir [12, 13].

ACT üretim sistemi mimarisi, üretim kuralını durum-hareket çiftleri şeklinde tanımlar. Yani, durumla karşılaşıldığında yapılacak olan problem çözme hareketi bellidir. Öğrenci modelinin karşılaştırılacağı bilişsel model önceden kurulmalıdır ve bu model bir üretim kuralı ağı şeklinde tanımlanmalıdır. Bilgi tabanı ve beceriler de benzer ağ yapıları ile tanımlanabilir.

Anderson, bu yöntemi kullanılarak ZÖS geliştirilmesinde aşağıdaki işlemlerin izlenmesi gerektiğini belirtmektedir:

a. Yapılandırılmış bir arayüz geliştirilmesi

Anderson, problem çözmede kullanılacak planlanmış bir arayüz tasarımının, ZÖS geliştirme işleminin birinci adımı olduğunu belirtmektedir.

b. Çözüm sözdizimi belirlenmesi

ZÖS, problem çözme oturumu sırasında, çözümün durumunu değerlendirebilmeli ve değiştirebilmelidir. Verilen problemin çözümü, alt parçaları ZÖS'te olan bilgi tabanı içinde temsil edilmelidir.

c. Problem temsilde sözdiziminin belirlenmesi

ZÖS'ün bilgi tabanı içinde yer alan tüm problemlerin, söz diziminin de belirlenmesi gerekmektedir. Verilen problemin yapısı ve sistemde ne şekilde temsil edileceği; öğrenciyi süreç içinde izlerken veya eğitimsel amaçlarla problemi bir durum çalışması şeklinde sunarken oldukça önemlidir.

d. Üretim kurallarının yazılması

Üretim kuralı herhangi bir durumla karşılaşıldığında kullanılacak durum-hareket çiftini belirlemektedir. Belli bir alandaki bilgi, üretim kuralı şeklinde temsil edilebilir. Anderson'a göre bunun asıl amacı; problem çözümü sırasında, problem çözümündeki o anki durum ile problem temsili arasında bağlantı kurmaktır. İlişki, yapılandırılmış arabirim ile sağlanmalıdır.

e. Üretim kurallarına bildirici bilginin eklenmesi

Her üretim kuralı öğrenci tarafından yapılması gereken bir bilgi parçasıdır. ACT teorisine göre, üretim kuralı ile ilgili bilgi parçası, üretim kuralını uygulayan örnek çözümlerle elde edilebilir. Örnek içerisinde problem çözme amaçları ve bu örneğin konuyla nasıl bağlantı kurduğunu açıklayan bilgi ile üretim kuralları ile birlikte sunulmalıdır.

f. İçeriğin belirlenmesi

İçeriğin belirlenmesi için en kolay yol, ders kitapların organizasyonunu takip etmektir. Bu yöntemde, kitaplar farkı parçalara ve bölümlere ayrılmışlardır. Her bölüm, belli bir konuyla ilgili kısmı tanıtmakta ve bu konuyla ilgili işlemsel problemleri çözümleriyle birlikte içermektedir. Daha sonra ise, öğrenci tarafından çözülmek üzere, öğrencinin bilgi düzeyini kontrol edecek farklı problemler sunmaktadır.

Bu yöntemle bağlantılı olarak, önceden belirlenen üretim kuralları, küçük kümeler halinde, örnek ve problemlere uyacak şekilde, bölümler içinde

gruplandırılabilir. Alan uzmanı, problemlerin sunum sırasını ve her bölümdeki alıştırmaları belirleyebilir veya öğrenci istediği alıştırmayı seçebilir [5, 14].

Model izleme tekniği içerisinde yer alan bilişsel öğretim sistemleri farklı alanlar için kullanılmıştır. Örneğin, cebir için PACT Algebra Tutor [15], geometri için PACT Geometry Tutor [16] sistemleri bulunmaktadır. PACT Algebra Tutor ve PACT Geometry Tutor sistemleri üretim kümelerini kullanmaktadırlar ve öğrencinin çözüm yolunu takip ederek, verimlerine göre geribildirim sağlamaktadırlar. Sistemler, öğrenci bilgisinin temsilini sağlamak için uzun vadeli öğrenci bilgilerine bağlıdır.

Heffernan III N.T [17] tarafından geliştirilen Ms. Lindquist cebir öğreticisi de model izleme tekniğini kullanmaktadır. Ms. Lindquist, tecrübeli gerçek öğretmen gözlemlerine dayanarak çalışmaktadır ve sembolizasyon alanında zengin stratejiler içermektedir. Bu çalışma ile, tecrübeli gerçek öğretmen analizlerine göre bilişsel model ortaya konmuştur ve gerçek öğretmen davranışlarının kopya edilmesi sağlanmıştır. Alana özgü pedagojik stratejiler kullanılarak, öğretim optimize edilmektedir. Geleneksel model izleme yapısında hatayı yapan öğrenci, o hatayı kaç kez yaptığına bakılmaksızın aynı mesajı almaktadır.

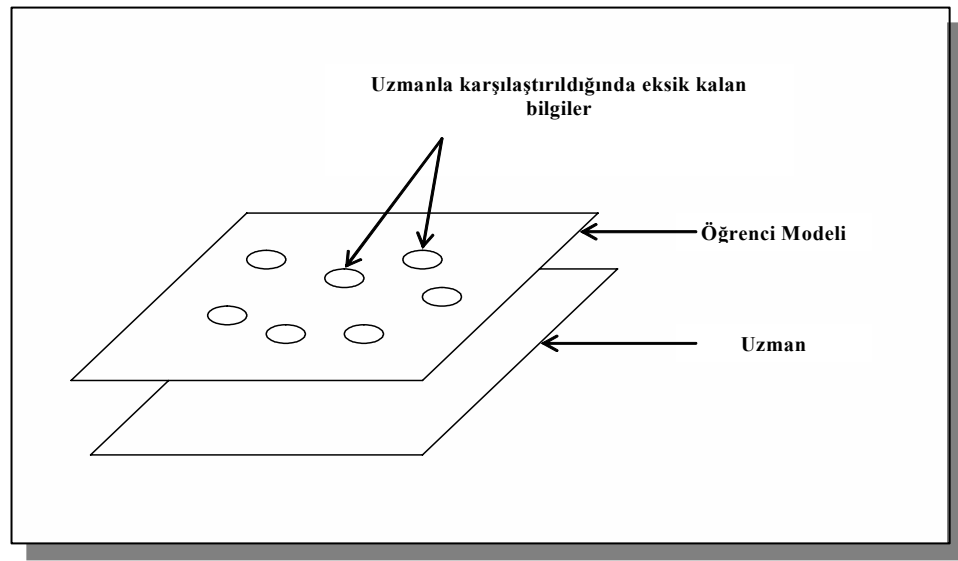
Conceptual Helper [18] sistemi ise, öğrencilere ev ödevlerini çözerken yardım eden sistemdir. Fizik öğretiminde bilişsel tabanlı bir yaklaşım kullanılmıştır. Geliştirilen sistemde, kavramsal anlama için gerekli olan konulara önem veren öğretim stratejileri, etkili öğretim teknikleri ve pedagojik yöntemler kullanılmıştır. Sistem, öğrencinin bir konuyu yanlış anlaması durumunda bunun nedenini sorgulayacak yapıya sahiptir.

II.2.2.2. Kaplama Öğrenci Modeli

Kaplama öğrenci modelinde, öğrenci bilgisi, alan bilgisinin bir alt kümesi olarak görülmektedir. Bu modeli Goldstein ilk defa, ITS WUMPUS sisteminde kullanmıştır [19]. Model, uzmanın bilmesi gereken her bir konuyu veya kavramın düzeyini tahmin etmektedir. Kaplama öğrenci modelinde, önkoşul olarak, alan bilgisi; kurallar, kavramlar ve olaylar şeklinde genel maddelere bölünebilir olmalıdır. Her bir konunun bilinme düzeyi için sistem tasarımcısı tarafından belirlenen değer aralığı kullanılmaktadır. Örneğin, 0 ve 100 aralığı bir konunun bilinme düzeyi için kullanıldığında, 0 değeri konunun bilinmediğini, 100 değeri ise konunun en üst düzeyde bilindiğini gösterecektir. Her madde için öğrenci düzeyi bir başlangıç değeri

ile başlamakta ve öğrenci davranışına göre bu düzey dinamik olarak değişmektedir [8].

Kaplama öğrenci modeli, uzman ve öğrenci arasındaki farkları içermektedir. İki farklı bilginin olması söz konusudur. Bunlar; bilinmeyen konular ve yanlış anlaşılan konulardır. Bu modelde, öğrencinin bilgisi uzman bilgisi ile karşılaştırılarak, öğrenci bilgisinin uzman modeli içine alınması sağlanmaktadır. Böylece öğrencinin neleri bildiği ve neleri bilmesi gerektiği belirlenmektedir. Şekil II.2’de öğrenci modeliyle uzman bilgisini kıyaslayarak, öğrencinin eksik kalan bilgilerini de gösteren kaplama öğrenci modeline ait bir örnek görülmektedir [20].



Şekil II.2 Kaplama öğrenci modeli örneği [20]

SQL(Structured Query Language)-Tutor’da [21, 22], yakın zaman öğrenci modelleme için kısıt tabanlı modelleme uygulanırken, uzun vadeli öğrenci bilgisi için kaplama öğrenci modeli kullanılmakta, bilgiler ise kısıtlar şeklinde temsil edilmektedir. SQL-Tutor, SQL veritabanı dilini öğretmeyi amaçlayan bir sistemdir [23, 24].

Weber E.M [25], tarafından geliştirilmiş GILES sistemi, PSPICE programı ile elektronik devre tasarımını konu almaktadır. Her alt konu bitiminden sonra sınav yapılmaktadır. Sistemin amacı, öğrencinin verilen problemin doğru sonucunu hesaplamasını sağlamaktır. Sistemin uzman modülü üretim kuralı olarak IF-THEN ifadelerinden oluşmaktadır. Öğrenci modeli ise, acemiden uzmana doğru bir yaklaşım sunan kaplama öğrenci modeli ile temsil edilmiştir.

Programlama konusunu alan olarak seçen Pl@tos [26] sistemi ise, uyarlamalı örün (web) tabanlı bir sistemdir. Programlama bilgisi, bildiren ve işlemsel olarak sınıflandırıldıktan sonra, programlama bilgisinin tipine göre, öğretim stratejileri ve metotları ile eşleştirilmiştir. Pl@tos’da, CLIPS (C Language Integrated Production System) ile geliştirilen uzman sistem örün ortamına entegre edilmiştir. Konular alt ünitelere bölünmüştür ve her ünite başlangıçta “öğrenilmedi” olarak işaretlenmektedir. Ünite sonundaki test, öğrencinin durumunu belirlemektedir.

WEST sistemi, oyun tabanlı bir ZÖS’tür. Sistemde öğrenciler, temel aritmetik becerileri kazanmak için “West nasıl kazandı” isimli bir oyun oynamaktadırlar ve gerektiğinde sistem öğrenciye yönlendirme ve yardım sağlamaktadır. [27]. Oyun sırasında öğrencilerin hataları hemen düzeltilmemektedir. Öğrenciler, kişisel olarak sistem hatalarını keşfetmekte ve düzeltmektedirler. WEST, uzman sistemindeki modülde bulunan örneği, öğrencinininki ile karşılaştırılarak, öğrenci becerilerini değerlendirmektedir. Öğrenci modeli, “tanıyıcı” ve “değerlendirici” olarak iki farklı öge içermektedir. Tanıyıcılar, öğrencinin bilgi ve yeteneklerini değerlendirerek bir model kurmaktadır. Değerlendiriciler ise, bu modeli yorumlayarak öğrencinin güçlü ve zayıf olduğu tarafları ortaya çıkarmaya çalışmaktadırlar. Öğrenci modelinin, öğrencinin kullandığı becerileri izleme ile ilgili bazı sorunları bulunmaktadır. Çünkü, sistem birçok beceriyi kapsamaktadır ve bazen öğrencinin oyunu sorgulama şeklini anlamak güçleşmektedir [28].

II.2.2.3. Kısıt Tabanlı Modelleme

Kısıt tabanlı modelleme, yapılan hatalardan öğrenme teorisi üzerine kurulmuş bir modeldir [29]. Bu modelde, model izlemede olduğu gibi öğrenci bilgisini tanımlamak yerine, hatalı bilgilerin üzerine yoğunlaşılır. Kısıt tabanlı modellemede, öğrencinin problem aşamasındaki bilgileri kullanılarak işlem yapılmakta, işlem sırası dikkate alınmamaktadır. Ayrıca, konunun temel prensiplerini bozacak şekilde probleme yaklaşıldığında doğru sonuca ulaşamayacağı düşünülmektedir [30].

Yanlış bilgiler, doğru bilgilerden çok daha fazla olacağından bu modellemede doğru çözümler kısıtlar ile temsil edilmektedir. Kısıt kümeleri, tüm olası doğru çözümleri tanımlamaktadır. Bu model, teoremler gibi, sadece bildiren bilgileri (declarative knowledge) temsil edebilmektedir.

Kısıt tabanlı modellemedeki bilgi birimine “durum kısıtı” denmektedir. Her kısıt, <Cr, Cs> şeklinde belli bir sıralı çift şeklindedir. Cr, uygunluk durumunu, Cs

ise yeterlilik durumunu göstermektedir. Örneğin, geometride üçgenin bir iç açısının hesaplanması konusu ele alındığında, Cr , iki açısı bilinen bir üçgendeki varsayım kısıtının, uygulanabilirlik durumunu göstermektedir. Cr , iki açısı bilinen herhangi bir üçgene ait probleme denk düşecektir. Cs ise, $\theta_3 = 180 - (\theta_1 + \theta_2)$ şeklinde çözümün doğruluğunu temsil edecektir.

Kısıtın sağlandığının veya bozulduğunun testi ileriye doğru yapılan bir araştırma ile belirlenmektedir. İlk önce Cr 'nin, öğrenci çözümü ile eşleşip eşleşmediği kontrol edilmektedir. Cr eşleşmiyorsa, kısıt atlanmaktadır. Cr eşleşiyorsa, öğrenci çözümü ile Cs 'nin değerlendirilmesi yapılmaktadır. Öğrenci çözümü, Cs 'yi sağlıyorsa, "sağlandı" olarak işaret konmakta, sağlamıyorsa "uygun değil" olarak işaret konmaktadır.

SQL-Tutor, veritabanı modelleme ve noktalama işaretleri için geliştirilmiş ve kısıt tabanlı modellemeyi kullanan bir öğretim sistemidir. SQL-Tutor, üniversite öğrencilerinin veritabanı kursunda kullandıkları bir sistemdir. Solaris, Windows ve örün ortamında çalışan versiyonları bulunmaktadır. Sistemde, normal konuşma cümlesi şeklinde sunulan problemin, SQL (Structured Query Language) eşitliğinin veya komutunun oluşturulması istenmektedir. Öğrenci bu sistem ile, SQL dilinin temellerini öğrenmektedir. Çünkü sistem, kaçırılan bir kısım olduğunda öğrenciye sonucu sunmaktadır. Bir problemin birden çok çözüm yolu olsa bile, SQL-Tutor'da tek ideal çözüm bulunmaktadır. İdeal çözüm, öğrenci çözümünün doğruluğunu saptamak için kullanılmaktadır. Arabirim, öğrencinin çözümündeki her tümce için ayrı bir metin alanı içermektedir. SQL cümlesini bu şekilde parçalayarak çözümün karmaşıklığı giderilmiştir. Öğrenci, yardıma ihtiyacı olduğunda, sistemden yardım alabilir ve sistem o andaki çözümü kısıtlarla eşleştirerek geribildirimini verir. Geribildirim seviyesi, sistem tarafından öğrenciye göre ayarlanabilir. Çok özet bilgiden, her kısıt ile ilgili tam çözüm içeren geribildirime kadar farklı düzeylerde geribildirim verilebilir. [31, 32].

KERMIT (A Knowledge-based Entity Relationship Modelling Intelligent Tutor) ise veritabanı modelleme için geliştirilen bir zeki öğretim sistemidir. Bu sistem, üniversite seviyesindeki öğrenciler için kullanılmıştır ve varlık ilişkisi modellemeyi (Entity Relationship-ER Modelling) konu almıştır. Sistemde öğrenciler için kişisel yardım verilebilmektedir. Öğrencilere ER modelleme kullanarak gerçekleştirmeleri için bir veritabanı senaryosu verilmektedir. Öğrenciler, problem çözme sürecinin herhangi bir aşamasında yardım alabilmektedirler. KERMIT

çözümü değerlendirip, hatalı bir durumla karşılaşıldığında geribildirim sunmaktadır [33].

CAPIT (An Intelligent Tutoring System for Capitalization and Punctuation) ise noktalama işaretlerini konu alan ve 10-11 yaşındaki öğrencileri hedefleyen bir zeki öğretim sistemidir [34]. CAPIT öğrenciden, verilen yazı parçasındaki imla hatalarını kontrol etmesini ve düzeltilmesini istemektedir. Eğer öğrenci hata yaparsa, hata mesajı görünmektedir. Birden fazla hata olsa dahi, tek bir hata mesajı görüntülenmektedir. Öğrenci tekrar çözümü gönderdiğinde, çözüm değerlendirilerek uygun geribildirim mesajı verilmektedir. Hata mesajları kısadır ve tek bir hata ile ilgilidir. Eğer öğrenci daha ayrıntılı bir açıklama isterse, “why” (neden) düğmesini kullanarak açıklamayı görmektedir.

II.2.2.4. Durum Tabanlı Sorgulama

ZÖS için geliştirilmiş durum tabanlı sorgulama oldukça önemli bir model türüdür. Bu modelde, öğrencinin problem çözme yeteneğini değerlendirmek için, öğrencinin geçmişte soruları nasıl çözdüğüne bakılmakta ve benzer bir durumla karşılaşıldığında da bu bilgilerden yararlanılmaktadır.

DTS (Durum Tabanlı Sorgulama), dört aşamadan oluşmaktadır. Öncelikle, probleme en uygun olan çözüm sistem tarafından öğrenciye gönderilmektedir. İkinci aşama, öğrenci çözümü tekrar kullanmaktadır. Üçüncü adımda ise, istenilen çözüm öğrenci tarafından düzenlenmekte veya uyarlanmaktadır. Son adımda ise öğrenci çözümü değerlendirilerek, öğrenci modelinin elde edilmesi sağlanmaktadır [35].

Bu modeli kullanarak geliştirilen SARA sistemi iki temel bilgi tabanı kullanılarak organize edilmiştir. Birinci bilgi tabanında, sistemin kullandığı problemler saklanmaktadır. İkinci bilgi tabanında ise, sistemde saklanan durumlar kümesi bulunmaktadır. Durum, önceki tecrübelerin (çözülmüş problemlerin) temsilidir. Sistem; öğrencinin çözümüne en uygun durumu, benzer çözüm kümesi içinden, sezgisel (heuristic) bir fonksiyonla seçerek belirlemektedir [35,36].

Durum tabanlı sorgulama modelini kullanan bir başka sistem ise, CBR (Case Based Reasoning) - Tutor sistemidir [37]. CBR-Tutor, etmen tabanlı ve Internet ortamında çalışabilen bir öğretim sistemidir. Öğretim stratejisine göre, verilen durumun uygun olup olmadığı sistem tarafından fark edilebilmektedir. Durum tabanlı öğretim etmeni, doğrudan öğrenci ile bağlantı kurar ve kendisine ait yerel bir

durum kümesi oluşturur. Sistemdeki algoritma ile, her yeni durum için en uygun olan durum seçilerek öğrenciye gösterilmesi sağlanır [36].

II.2.2.5. Bayes Öğrenci Modeli

ZÖS alanındaki araştırmacıların çoğunun ilgisi, öğrenci modelleme konusunda Bayes ağları üzerinde yoğunlaşmıştır. Bayes ağları ile koşullu olasılıklar ortaya konarak, bir ağ yapısının ortaya çıkması ve öğrenci modellemesinin yapılması sağlanmaktadır.

Bayes öğrenci modeli, ANDES [38], HYDRIVE [39], DT-Tutor [40] ve CAPIT [41]'de kullanılmıştır. Bu yöntemin olası bir dezavantajı, çok fazla değişkene sahip olabilmesi ve bunun sonucu olarak da ağ yapısının etkili bir şekilde hayatta kalamamasıdır.

ANDES koşullu olasılık problemini çözmek için OR ve AND kurallarının tanımlamaları yapmaktadır. OR, herhangi bir üyenin birisi doğru olduğunda doğru, aksi durumda yanlış bir yapıya sahiptir. AND ise üyelerinin tümünün doğru olması durumunda doğru değerini alan bir yapıdır. Pratikte koşullu olasılıkları, OR ve AND ile sınırlandırmak gereken olasılık sayısını belli ölçüde indirgemekte ve gözlenemeyen değişkenlerin modellenmesini kolaylaştırmaktadır.

ANDES'in temel öğrenci modelleme bileşeni iki kısımdan oluşmaktadır. Statik kısım problemde problemde geçişlerde aynen kalmaktadır. Dinamik kısım ise, her yeni problem için yeniden kurulmakta, problemin çözümünden sonra ağ yapısındaki bağlantılar ortadan kaldırılmaktadır. Statik kısım, öğrencinin belli konudaki kuralları uygulayabilme yeteneğini modellemektedir. Dinamik kısım ise, o anki problem için öğrencinin problem çözme adımlarını, stratejisini modellemektedir. Dinamik kısım, belli bir problem için, olası tüm çözümleri içeren problem çözme grafiğinden eş zamanlı alınmaktadır. Çözüm grafiği ise, problem çözücüsünden alınmaktadır.

HYDRIVE ise uçak hidrolik teknisyenlerinin, arıza düzeltme eğitiminde kullanılan bir sistemdir. Bayes ağı ile öğrenci modellemesi gerçekleştirilmiştir ve sistemde belli hata bulma gözlemlerinden, daha genel strateji ve işlemler ele alınmaktadır. Uzmanı tamamen modellemeye çalışan bilişsel sistemlerden farklı olarak, HYDRIVE' daki öğrenci modeli, usta ve acemi öğrencileri ayırmak için önemli olan faktörleri yakalamaya çalışmaktadır.

DT-Tutor genel amaçlı kullanılabilecek, alandan bağımsız mimaride öğrenci modelleme ve pedagojik hareket seçimi sunmaktadır. ANDES ve HYDRIVE gibi

öğrencinin bilgisini bayes ağları ile modellenmekle birlikte, diğer sistemlerden farklı olarak öğrencinin morali, bağımsızlığı ve dikkat odağı gibi saklı durumları, öğrencinin sistemle olan etkileşimine bağlı olarak belirlediği değişkenlerle modellemektedir.

CAPIT ise noktalama işaretlerini konu alan bir ZÖS'tür ve öğrenci için yeni problemleri seçerken, uzun vadeli öğrenci modeli için, Bayes ağlarını kullanmaktadır. Bu modelde, ağdaki koşullu olasılıklar öncelikle, verilerden öğrenilmektedir. CAPIT öğrenci modeli, gözlenen değişkenler arasındaki ilişkiler ile öğrenci verimini bulmaya çalışmakta ve gözlenmemiş öğrenci durumlarını modellemek için hazırlık yapmaktadır [41, 42].

II.2.2.6. Streotip Öğrenci Modeli

Streotip modellemede, öğrenci belli bir kalıba ait olarak görülerek basit bir öğrenci modeli oluşturulmaktadır. İki çeşit streotip bulunmaktadır. Bunlar: sabit [43] ve varsayılan streotip modelleridir [44]. Sabit streotip modelde, öğrenci performansına göre öğrenciye bir seviye verilmektedir. Bu modelde, aynı streotipde bulunan öğrencilerin aynı bilgi düzeyinde oldukları ve problem çözme davranışlarına sahip oldukları varsayılmaktadır. Sistem öğrenciyi bir streotipten diğerine taşıyabilir, fakat streotipler kendi kendilerine değişmezler. Bu model, karmaşık analizlerde kullanışlı bir yöntem değildir. Fakat bilginin küçük parçalara bölünemediği alanlarda geçerliliği olabilecek bir yöntemdir [8]. Varsayılan streotip daha esnek bir yaklaşım sunmaktadır. Başlangıçta öğrenciye bir streotip atanmaktadır. Zamanla öğrenci yanıtlarına göre, bu streotip daha bireyselleştirilmiş ayarlara göre değiştirilmektedir. Kaplama model gibi karmaşık öğrenci modelleri için, bu model bir başlangıç noktası olabilir.

Bilimsel terminoloji öğretimi için tasarlanan StyLE-OLM [45], bu tür bir yaklaşım içermektedir. Sistem doğal diyalogları ile öğrenci ile iletişim kurarak; kendi kuralları ile, öğrenci cevaplarından öğrenci bilgi tabanını oluşturmaya çalışmaktadır [8].

Cebir ve geometri öğretmek için tasarlanmış WPS-Tutor [46] sisteminde problemler seviyelere göre yerleştirilmiştir. Bir seviyedeki problem, bir öncekinden sadece biraz daha zordur. Bir öğrenci aynı seviyedeki birkaç problemi yardım almadan çözdüğünde, seviye yükselmektedir. WPS-Tutor'da seviye, öğrenci modelindeki ana tanımlamadır.

ATULA [47] sistemi sabit streotip model kullanarak geliştirilmiştir. Sistem kullanılmaya başlanmadan önce 134 öğrenciden deneysel veri toplanmıştır. Kişisel ve geçmiş bilgileri kapsayan bir test öğrencilere sunulmuştur. Kişisel ve geçmiş bilgiler kullanılarak, istatistiksel kümeleme analizi gerçekleştirilmiştir. Kümeleme işlemindeki hassas ayarlar ise sonraki test sonuçlarına göre yapılmıştır. Sonuçta, altı kız ve altı erkek öğrenci streotipine ulaşılmıştır. Yeni bir öğrenci sisteme girdiğinde hangi streotipe üye olduğuna, sisteme ilk bağlandığında yaptığı test sonucuna göre karar verilmektedir. Sistemdeki test sonuçlarına göre öğrenciler, olabilecek en uygun streotipe yerleştirilmektedirler. Fakat, oturum süresince öğrenciler farklı streotiplere taşınabilirler. Örneğin, öğrencinin bulunduğu streotip bir yeteneğin öğrencide bulunduğunu gösterirken, sistemdeki test sonucu o yeteneğin bulunmadığı sonucuna varırsa, öğrenci daha uygun streotipe alınmaktadır.

II.2.2.7. Etmen Tabanlı Modelleme

Basit sistem işlemlerinden yüksek seviyedeki yazılım-donanım işlemleri arasında değişebilen, belli bir görevi veya görevler kümesini yerine getirebilen programlara verilen genel isim “etmen”dir [48].

Başka bir tanıma göre etmen, belli bir ölçüde bağımsız olarak kullanıcı veya başka programlar için çeşitli işlem kümelerini yerine getiren programlardır [49].

Etmenler zeki öğretim sistemlerinde, pedagojik etmen olarak görev yapmaktadırlar. Pedagojik etmenlerin, kurallara bağlı öğretim amaç kümesi ve bu amaçlara ulaşmak için planları bulunmaktadır [50].

ZÖS’te kullanılan etmenler, sistemin öğretim fonksiyonunu yerine getirmek için öğrencilerle iletişim kuran varlıklardır. ZÖS’te öğretim bileşeni olarak etmenlerin kullanılmasının nedeni, etmenlerin iletişim ve etkileşim yapabilme kapasiteleridir. Etmenler; eğitim oturumları içinde de, öğrenebilme, uyarılma yapabilme yeteneği olan varlıklardır [35]. Pedagojik etmenler, sanal bir öğretici veya sanal bir öğrenme katılımcısı olarak öğrenme sürecinde, öğrenciye yardımcı olmaktadır [48].

Etmen kullanılarak tasarlanan sistemlerden birisi de STEVE (Soar Training Expert for Virtual Environments) [51]’dir. Bu sistem, sanal bir ortam içinde laboratuvar durumunu öğrenciye sunmaktadır. Öğrenci, bu ortam içinde çeşitli laboratuvar donanımlarının tamir ve çalışması ile ilgili işlemsel ve fiziksel görevleri yerine getirmektedir.

Tıp alanında travma tedavisi ve onkoloji için geliştirilen ADELE (Agent for Distance Learning Environments) [52] ve hemşirelik eğitiminde kullanılan INES (Intelligent Nursing Education System) [53] sistemlerinde de etmenlerden yararlanılmıştır.

II.2.3.Öğretim Modeli

Bir eğitim sisteminin çalışmasını sağlayan temel kısım öğretim modelidir ve öğrenci modeli ile sıkı bir bağ içindedir. Bu bileşen, öğrenciyle yapılan eğitimsel etkileşimlerin tasarlanmasını ve kontrol edilmesini sağlamakta ve kararları verirken öğrenci bileşeni ve alan bilgisinden yararlanmaktadır. Bu bileşen yardımıyla, öğrenci bileşeninin oluşturulması ve güncellenmesi sağlanmaktadır.

Öğretim modeli, öğrenci modelinden öğrenci yetenekleri ile ilgili analizi alarak; sistemin öğretim sürecini belirlemekte ve sistemin ne zaman konu tekrarı yapması gerektiğini, sonraki adımda hangi konunun gösterilmesi gerektiğini belirlemektedir [13].

Aynı zamanda öğretim modeli, öğrenci modeline uyacak; yol gösterme, öneride bulunma, açıklama yapma, başka bir konuya geçme gibi öğretim etkinliklerinin yapılmasını sağlamaktadır [8].

Farklı öğretim stilleri ve modelleri bulunmaktadır. İyi bir öğretici, öğrencinin öğrenme stiline uymadığı noktalarda kendi stilini değiştirebilecek özellikte olmalıdır [54]. Benzer şekilde, ZÖS'te farklı pedagojik kararlar alabilen öğretim modeline sahip olmalıdır. Pedagojik stratejiler öğrenmenin iki ucu olan öğretici ve keşif tabanlı öğrenme arasında değişmektedir. Öğretici yaklaşım, biçimsel ve daha geleneksel bir yapıya sahiptir. Bu öğrenme bileşeninde, öğrenci bilgilendirilir ve görevler ağırlıklı olarak hedefi açıklanan amacı gerçekleştirmek içindir. Öğretici yöntemi benimseyen öğretim sistemlerinde, öğrenci aktiviteleri kontrol edilmektedir. Bütün aktiviteler sistemin hedeflediği eğitim hedeflerine odaklanmıştır. Bir konuyu yeni öğrenmeye başlayan öğrenciler için etkili olabilecek bir öğretim yöntemidir. Çünkü, bu tip öğrenciler, daha fazla yardıma ihtiyaç duymaktadırlar. Fakat, daha bilgili öğrenciler için bu yöntem fazla sınırlı ve kolay olabilir.

Keşif tabanlı öğrenme ise, deneyimlerden öğrenmeyi amaçlamaktadır. Keşif tabanlı öğrenmenin, önceden öğrenilen bilgilerle öğrenilmiş bilgiler arasında bağlantı kurmayı sağlayabilmesi, öğretici öğrenmeye karşı bir üstünlüktür. Fakat, bir konuyu

yeni öğrenmeye başlayan öğrenciler için, sistemin hedeflerini keşfetmek uzun bir süre alabilir. Yakın tarihte yapılan çalışmalar, keşif süresince yapılacak kılavuzluğun öğrenme sürecinin etkinliğini artırdığını ortaya koymuştur. [8] .

II.2.4.Kullanıcı Arabirimi

Kullanıcı arabirimi, öğrenci ve öğretim sistemi arasında aracı görevindedir ve sistem ile öğrencinin iletişim kurabilmesini sağlamaktadır. Sistem, alan bilgisini öğretebilmek için kullanıcı arabirimini kullanmaktadır [7, 55].

Kullanıcı arabirimi ile öğrenci, gerçek uygulamaları keşfetme imkanına sahip olur ve öğrenme çevresi içinde öğrencinin eğitim süreçlerini daha iyi kavraması sağlanır [56].

Öğretim sistemi arabiriminin sahip olması gereken bazı önemli özellikler bulunmaktadır. Öncelikle, arabirim öğrenciye devam etmesi için motivasyon sağlamalıdır. Etkili bir öğretim için, motivasyon çok önemlidir. Motivasyon sağlandığında, sistem ile daha fazla zaman harcayan öğrenci de, sistemden daha fazla bilgi edinir.

Kullanıcı arabirimi, öğrencinin bellek yükünü azaltarak öğrenmeyi artırabilir. Bu nedenle, sadece öğretilmek istenen belli bir kısma odaklanma sağlanmalıdır. Böylece, öğrencinin probleme yoğunlaşması mümkün olur.

Amacı iyi görüntüleyebilen bir kullanıcı arabirimi, öğrencinin hedefe daha kolay ulaşmasına yardımcı olacaktır. Örneğin, sorular çoktan seçmeli veya boşluk doldurma şekline getirilerek daha kolay anlaşılması sağlanabilir. Ayrıca kullanıcı arabirimi, öğrencinin düşüncesini ve yanıtını yapılandırmasını sağlayacak şekilde tasarlanmalı, istenilen cevap şekli açıkça ortaya konmalıdır [8].

II.2.5.MYCIN Uzman Sistemi Güven Faktörü Hesaplama Yöntemini Kullanan ZÖS Örnekleri

MYCIN uzman sistemi (Computer- Based Medical Consultation), 1976 yılında Stanford üniversitesinde Edward Feingbaum başkanlığında bir grup uzman hekim tarafından geliştirilmiş bir uzman sistemdir [57]. Bu sistemin amacı menenjit ve bakteriyel enfeksiyonların teşhis ve tedavisinde doktorlara yardımcı olmaktır. Sistem; hastanın geçmiş bilgileri, laboratuvar sonuçları ve semptomlardan yola çıkarak tedavi yöntemlerini belirlemektedir. Bu uzman sistemde, belirsizliği

gidermek ve sistemde bulunan bilgilerle bir sonuca ulaşmak için güven faktörü hesaplaması yapılmaktadır. Bu hesaplama yönteminden, aşağıda belirtilen zeki öğretim sistemlerinde de yararlanılmıştır.

GUIDON, kural tabanlı, tıbbi teşhis konusunu ele alan, 1970 yılında William Clancey tarafından Stanford Üniversitesinde geliştirilmiş bir sistemdir. Sistemi tıp öğrencileri kullanmaktadır. Uzman modülü, kural tabanlı MYCIN teşhis uzmanından oluşmaktadır. Öncelikle hastanın geçmişi, fiziksel muayene sonuçları ve test sonuçları öğrenciye verilmektedir. Daha sonra ise, uygun klinik verinin nasıl alınacağını ve hastalığa neden olan organizmanın nasıl teşhis edileceği öğretilmektedir [58, 28].

LeCo-EAD (Learning Companion System), örün tabanlı zeki öğretim sistemi, sahip olduğu sanal öğrenci yapısı ile sisteme bağlanan öğrenciye yardımcı olmaktadır. Alan modeli, örün sayfaları aracılığı ile öğrenciye sunulmuştur. Öğrencinin kavramlardaki başarı puanına göre, sistemde ilerlemesi sağlanmaktadır. Her konu ve soru sistemde farklı ağırlıklara sahiptir. Kaplama öğrenci modeli kullanan sistemde, öğrenci modelinin güncellenmesi için MYCIN uzman sistemi güven değeri hesaplama yöntemi kullanılmaktadır. Öğrencinin sorulara verdiği yanıtı göre, MYCIN güven faktörü ile kavram başarı puanı hesaplanmaktadır. Bu başarı puanına göre, kavramdaki bilgi düzeyi ölçülmektedir [59].

Anjaneyulu tarafından geliştirilen ve alan modelini kavramlar seviyesinde modelleyen ve kaplama öğrenci modeli kullanan örün tabanlı öğretim sisteminde, kavramlardaki başarının hesaplanması için MYCIN güven faktörü hesaplaması kullanılmıştır. Sistemde, kavramlar arası önşart ilişkisine bağlı kurs içeriği oluşturulmuştur. Daha sonra ise, öğrencinin kavramlarda yer alan sorulara verdiği cevaba göre, kavram başarısını hesaplamak için MYCIN uzman sistemindeki güven faktörü hesaplama yönteminden yararlanılmıştır [60].

BÖLÜM III

VERİ MADENCİLİĞİ, BİRLİKTELİK KURALLARI ve KÜMELEME ANALİZİ

III.1. GİRİŞ

Bu bölümde;

- Veri madenciliğinin tanımı, veri madenciliğinde kullanılan teknikler açıklanmakta ve bu tez çalışmasında tasarımı gerçekleştirilen yazılımlarda kullanılan birliktelik kuralı analizi algoritmalarından Apriori ve kümeleme analizi algoritmalarından k-means algoritması,
- Çok boyutlu verilerin görselleştirilmesinde kullanılan paralel koordinat sistemi ve paralel koordinat sistemini kullanarak görselleştirme işlemini gerçekleştiren Parvis yazılımı,
- Veri madenciliğinin eğitim alanında kullanılmasına ilişkin gerçekleştirilmiş çalışmalar,

açıklanmakta ve incelenmektedir.

III.2. VERİ MADENCİLİĞİ

Veri elde etme ve saklama teknolojilerinin hızlı ilerlemesi veritabanlarında ve veri ambarlarında saklanan verilerin çok fazla büyümesine neden olmuştur. Çok büyük ölçekli veriler ve büyük ölçekli veri tabanları, değerli verileri bulunduran bir veri madeni gibi düşünülebilir. Bu büyüklükteki verilerin analizi, bu analiz sonucunda anlamlı bilgi elde etme ve elde edilen bilgiyi yorumlamak insan

yeteneğinin ve ilişkisel veri tabanlarının yapabileceklerinden fazlasını gerektirmektedir [61].

Bu tür bilgileri ortaya çıkarmak için, 1980'lerin sonlarında veri madenciliği yeni bir disiplin olarak ortaya çıkmıştır [62]. Veri madenciliği, veriden anlamlı bilgi etmek için araştırma ve analiz yapmaktır [63]. Veri madenciliği, veri ambarlarındaki büyük veri setlerinin yapay zeka ve çeşitli matematiksel tekniklerle işlenerek yeni korelasyonlar ve örüntülerin bulunması işlemidir [64]. Başka bir tanıma göre; veri madenciliği, veride bulunan anlamlı, yeni, kullanışlı ve anlaşılabilir örüntüleri tanımlama sürecidir [65]. Keşfedilen bu örüntüler için önemli olan, gerçek dünya için de anlamlı olmalarıdır [66].

Bilgi keşfi de veri madenciliği ile birlikte sıkça kullanılan bir terimdir. Bilgi keşfi; literatürde, “veri tabanlarındaki ilginç örneklerin, kısa ve anlamlı bir şekilde tanımlamasını yapmak” olarak tanımlanmaktadır [67, 68].

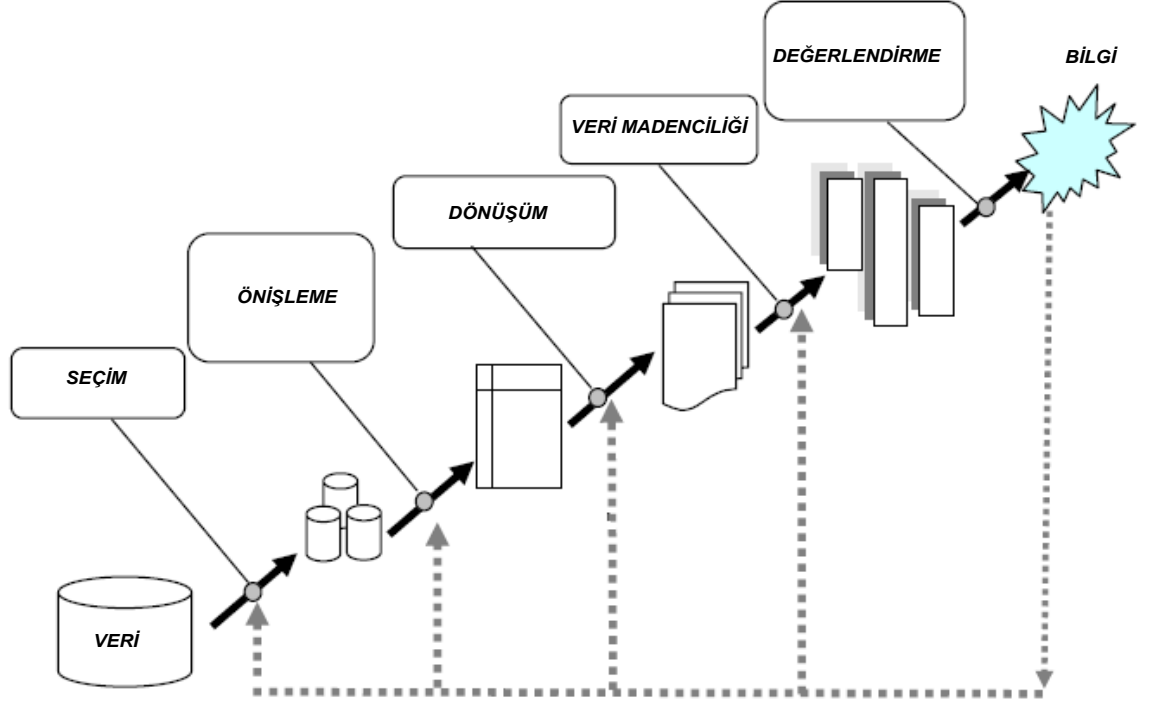
Disiplinler arası bir yapısı olduğu için, veri madenciliği; veritabanları, makine öğrenimi, istatistik ve veri görüntülenmesi gibi birçok farklı disiplinden katkı almaktadır. Veritabanlarında güçlü bir veri yönetimi olmadan ve makine öğreniminin sağladığı pratik veri analiz yöntemleri olmadan veri madenciliği etkili sonuçlar üretememektedir [62].

Veri madenciliğinde zor olan kısım, geçmişte ne olduğunu tanımlayacak bilgileri içeren tahmin edici model kurmaktır. Bu amaç için geliştirilmiş bazı araçlar yardımıyla geçmiş başarı ve eksiklikler göz önünde tutularak, daha sonra ne olacağı bulunmaya çalışılmaktadır. Veri madenciliği araçları ve insan arasındaki en büyük fark; veri madenciliği araçlarının çok büyük veri tabanlarında hızlı bilgi bulabilmesi ve çok küçük bir modelden yola çıkarak iyi bir tahmin yapabilmesidir [68].

Veri madenciliğinin pratikte iki önemli görevi bulunmaktadır. Bunlar, tahmin etme ve tanımlamadır. Tahmin etme, veritabanlarında ilgilenilen değişkenlerin bilinmeyen gelecek durumlarını, yine veritabanında bulunan bazı alan ve değişkenler yardımıyla bulmayı içermektedir. Böylece, sonuçları bilinen verilerden yola çıkılarak bir model geliştirilmekte ve bu model yardımıyla bilinmeyen veri kümelerinin bulunması sağlanmaktadır. Tanımlama ise, insanın yorumlayabileceği örnekleri kullanarak veriyi tanımlamak üzerine yoğunlaşmıştır ve karar vermeye yardımcı olacak örüntülerin bulunmasını amaçlamaktadır [61, 69].

III.2.1. Veritabanlarında Bilgi Keşfinin Aşamaları

Bilgi keşfi, seçilen veri madenciliği tekniği ve verinin yapısına göre birçok aşama içermektedir. Şekil III.1.'de görülen bu aşamalarda yapılan işlemler:



Şekil III.1 Veritabanlarında bilgi keşfi süreci [61]

1. Seçim İşlemi: Genelde birçok kaynaktan gelen veriler, tek bir alanda toplanmaktadır. İlk aşamada, analiz edilecek veriler veritabanından seçilip alınmaktadır.

2. Önileme: Birçok kaynaktan alınan verilerin düzenlenmesi, gürültülü, tutarsız verilerin işlenmesi ve eksik verilerin veri setinden çıkarılması işlemidir.

3. Dönüşüm İşlemi: Algoritmanın uygulanabilmesi için verinin kategorik değerlere dönüştürülmesi, özetleme yapılması gibi veriye uygulanacak algoritma için daha uygun hale getirme işlemlerini kapsamaktadır.

4. Veri Madenciliği: Veri madenciliği, önceden hazırlanmış ve genellikle dönüşüm yapılmış veriye, gerçek algoritmanın uygulanması ile gerçekleştirilmektedir. Analiz gerçekleştirilirken, algoritma seçimi, parametrelerin belirlenmesi sonucun kalitesini etkileyen faktörlerdir.

5. Değerlendirme İşlemi Farklı parametrelerle algoritmanın denenmesi veriden elde edilen analiz sonuçlarının incelenmesi işlemidir [61, 70].

III.2.2. Veri Madenciliğinde Teknikler

Temel veri madenciliği teknikleri, buldukları örüntü yapıları, sonuçların kullanım amaçları, veri yapısı ve kullandıkları yöntemle ilgili olarak farklılık göstermektedirler [71].

Bölüm III.2’de belirtildiği gibi, veri madenciliğinin temelde tahmin etme ve tanımlama olmak üzere iki görevi vardır ve veri madenciliği teknikleri de, tahmin edici ve tanımlayıcı olmak üzere iki ana başlıkta incelenmektedir [61]. Tahmin edici teknikler, sonuçları bilinmeyen veri kümeleri için, sonuçları bilinen verilerdeki değerleri kullanmaktadırlar. Örneğin, bir bankanın kredilerle ilgili elinde bulunan verilerden hareketle, daha sonraki kredi taleplerinde müşteri özelliklerine göre, verilecek kredinin geri ödenip ödenmeyeceğinin tahmini bu tip bir tekniğe örnektir. Tanımlayıcı teknikler ise, mevcut örüntülerin tanımlanmasını ve karar vermeye rehberlik etmesini sağlamaya çalışmaktadır. Geliri X-Y aralığında olan, iki veya daha fazla arabası olan çocuklu aileler ile çocuğu olmayan ve geliri X-Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerliğinin belirlenmesi tanımlayıcı tekniklere örnektir [72].

Tahmin edici tekniklerde amaç; sınıflama ve tahmin gibi işlemleri gerçekleştirebilecek bir sistem üretmektir. Tanımlayıcı teknikler ise, büyük veri kümelerindeki ilişkilerin ve örüntülerin elde edilmesini sağlamaktadır. Tanımlayıcı ve tahmin edici tekniklerdeki amaçlar, aşağıda açıklanan veri madenciliği teknikleri ile elde edilmektedir. Literatürde ortak bir sınıflama olmamakla birlikte, veri madenciliğinde kullanılan teknikler aşağıdaki gibi sınıflandırılabilir:

1. Sınıflama: Bir veri elemanının, daha önceden tanımlanmış farklı sınıflardan hangisine ait olduğunu keşfeden, tahmin edici fonksiyonun bulunması işlemidir. Elde edilen modeller, karar ağaçları, matematiksel formüller ve sinir ağları gibi çeşitli yöntemlerle gösterilebilir. Sınıflama, kategorik değerlerin tahmin edilmesinde kullanılmaktadır. Örneğin, sınıflama analizi ile banka kredi uygulamalarının güvenli veya riskli olmaları sınıflandırılabilir [73] [74].

2. Birliktelik Kuralı Analizi: Genelde pazarlama alanında müşterilere ait veriler üzerinde kullanılan bir tekniktir. Bu teknik, insanın fark edemeyeceği ilişki

örüntülerini otomatik olarak araştırmaktadır. Birliktelik kuralı ile, bir arada normalden fazla tekrarlanan veri parçaları bulunmaktadır. X ve Y bir birliktelik kuralına sahipse; X'in var olması, Y'nin de belli bir tekrar değeri ile var olması anlamına gelecektir. Birliktelik kuralı analizi, farklı uzunluktaki veritabanları üzerinde çalışabilen öğreticisiz bir tekniktir ve eş zamanlı olarak gerçekleşen ilişkilerin tanımlanmasında kullanılmaktadır [61, 75, 76].

3. Regresyon: Bir veri elemanının olabilecek gerçek bir değer ile eşleştirilmesi işlemi için, tahmin edici fonksiyonun bulunması işlemidir. Regresyon, süreklilik gösteren değerleri tahmin edilmesinde kullanılmaktadır. Örneğin, teşhis ve test sonuçlarına göre hastanın yaşama olasılığının belirlenmesi veya yeni bir ürüne olacak talebin reklam giderlerine göre belirlenmesi regresyon örnekleridir [69]. Tahmin edici regresyon analizinde, seçilen veri kümesinde değişkenlerin sayısal değerleri ile ilgili bazı tahminler yapılmaktadır. Bu yaklaşımla, veri noktalarına uyan en iyi eğri belirlenmektedir. Daha sonra ise, bu eğrinin matematiksel fonksiyonu çıkartılarak, korelasyon ölçüsü bulunmaktadır [77].

4. Kümeleme: Bir veri kümesini tanımlamak için sonlu sayıdaki kategori veya kümenin bulunmasını amaçlayan tanımlayıcı bir işlemdir. Kümeleme analizi, öğreticisiz bir yöntemdir ve küme içi benzerliklerin fazla, küme dışındaki benzerliklerin az olma prensibine göre işlem yapmaktadır. Kümeleme tekniğinde, sınıflama analizinde olduğu gibi veri sınıfları mevcut değildir. Sınıflama analizinde, verilerin sınıfları bilinerek, yeni gelen verinin hangi sınıfa ait olabileceği tahmin edilmektedir. Kümeleme analizinde, henüz sınıfları bulunmayan veriler kümeler halinde sınıflara ayrılırlar [61].

5. İstisna (Outlier) Analizi: Veri kümesi içerisinde verinin genel yapısı veya veri modeline uymayan veriler bulunabilir. Birçok veri madenciliği yöntemleri bu sıradışı verileri gürültü veya istisna olarak belirler. Özellikle dolandırıcılık veya seyrek geçen olayların belirlenmesi gibi uygulamalar, daha düzenli gerçekleşen işlemlere göre ilgi çekici olmaktadır. Bu istisna durumundaki verilerin analizi, istisna madenciliği olarak adlandırılmaktadır. Bu analiz, istatistik tabanlı yöntemlerle tespit edilebileceği gibi, veri noktaları arasındaki uzaklığın dikkate alındığı yoğunluk tabanlı yöntemler de kullanılmaktadır [78].

6. Gelişimsel Analiz: Bu analiz, bir zaman içinde değişen veri nesnesi için düzenlilik veya eğilim modelinin tanımlanması işlemidir. Bu analiz zaman serisi

şeklinde olan veri içerisinde birliktelik, sınıflama veya kümeleme analizi olabileceği gibi, periyodik örüntü eşleştirme ve benzerlik tabanlı veri analizini de içermektedir.

Bu tez çalışmasında; veri madenciliği analizini gerçekleştirmek için kümeleme ve birliktelik kuralı teknikleri kullanılmıştır. Kümeleme analizi k-means algoritması ile, birliktelik kuralı analizi ise Apriori algoritması ile gerçekleştirilmiştir. [61, 78].

III.3. BİRLİKTELİK KURALLARI

Bu tezde, veri madenciliği analizinde birliktelik kuralları kullanıldığından, birliktelik kuralı analizinin ayrıntısı burada açıklanmaktadır.

Birliktelik kuralları verilen veri seti içindeki elemanlar arasındaki ilginç(interesting) kuralları ve bir zaman aralığında sıklıkla gerçekleşen olayları bulmaya çalışmaktadır [61, 79]. Birliktelik kuralları, çapraz satış (cross-marketing), posta gönderiminde hedef kitlenin bulunması, katalog tasarımı, mağaza ürün yerleştirme tasarımı, müşterilerin satın alma örüntülerinin bulunması ve buna göre gruplama yapılması gibi işlemlerde kullanılmaktadır [61, 68]. Birliktelik kuralı, öğreticisiz öğrenme sistemlerinde yerel örüntü bulma işleminin en yaygın şeklidir. Bu yöntem ile, veritabanındaki ilginç örüntüler bulunmaktadır [78]. Birliktelik kuralı analizinde, kural indirgemesi yapılırken, verilen destek ve güven değerlerine göre, analiz yapılmaktadır [68].

Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti uygulamasıdır. Market sepeti verisi üzerinde birliktelik kurallarının kullanılması, ilk olarak 1993 yılında ortaya çıkartılmıştır [80,81]. Bu çalışmada, marketlerdeki büyük veritabanları kullanılarak, birliktelik kuralları uygulandığında, bir arada alınma olasılığı yüksek olan ürünlerin ortaya çıkarılabileceği gösterilmiştir. Örneğin, bilgisayar alan kişilerin %60'ının, yazıcı da alması birliktelik kuralı ile bulunabilecek bir bilgidir. Bu örnekte, bilgisayar kuralın ön kısmı, yazıcı ise sonuç kısmıdır ve bu kural Denklem III.1'deki gibi gösterilir. Bu kurallara göre, pazarlama politikaları geliştirilebilir, mağazalardaki raflar veya ürün sayfaları bu bilgilere göre düzenlenir [80] .

$$\text{Bilgisayar} \Rightarrow \text{Yazıcı} \quad [\text{Destek}=\%5, \text{Güven}=\%60] \quad (\text{III.1})$$

Denklem III.1'de görülen destek ve güven ifadeleri, birliktelik kurallarında kullanılan ölçütlerdir. Denklem III.1'deki kuralda; destek değeri, veri setinde tüm

kayıtlar içerisinde bilgisayar ve yazıcı alan müşterilerin birlikte bulunma oranının %5 olduğunu göstermektedir. Başka bir ifadeyle, bir veri kümesinin tüm veri setinde bulunma yüzdesi, veri kümesinin destek değeri olarak adlandırılmaktadır. $X \Rightarrow Y$ gibi bir kuralda %c oranındaki güven değeri ise; tüm veri setinde X'in var olduğu kayıtların %c'sinde Y'nin de var olduğunu göstermektedir. Denklem III.1 için, güven değerinin %60 olması, tüm veri setinde bilgisayar alan müşterilerin %60'ının yazıcı da aldığını göstermektedir. Güven değeri, bir kuralın ne kadar doğru olduğunu gösteren bir ölçüttür [81, 82]. Kullanıcı tarafından belirlenen minimum güven ve minimum destek eşik değerlerini aşan kurallar ilginç sayılmaktadır. [61, 81-83].

$X \Rightarrow Y$ gibi bir birliktelik kuralında destek değeri Denklem III.2'deki gibi hesaplanmaktadır

$$\text{Destek}(X \Rightarrow Y) = \frac{\text{X ile Y'nin bir arada yer aldığı kayıtların sayısı}}{\text{Toplam kayıt sayısı}} \quad (\text{III.2})$$

$X \Rightarrow Y$ gibi bir birliktelik kuralında güven değeri ise, Denklem III.3'deki gibi hesaplanmaktadır

$$\text{Güven}(X \Rightarrow Y) = \frac{\text{X ile Y'nin bir arada yer aldığı kayıtların sayısı}}{\text{X'nin yer aldığı kayıtların sayısı}} \quad (\text{III.3})$$

Başka bir ifadeyle; $X \Rightarrow Y$ kuralındaki destek değeri, veri kümesindeki $P(XUY)$ olasılığı yani X ve Y elemanının bir veri kümesinde birlikte bulunma olasılığı, güven değeri ise, $P(Y|X)$ olasılığı yani veri kümesinde X'in bulunma durumunda Y'nin bulunma olasılığı olarak gösterilmektedir [61, 83].

Birliktelik kuralı bulmak için en yaygın olarak kullanılan algoritma Apriori algoritmasıdır [61]. Apriori algoritması Bölüm V.2.3'de açıklanmaktadır.

III.4. KÜMELEME ANALİZİ

III.4.1. Kümeleme Analizinin Tanımı

Kümeleme, veri grubunu birbirinden farklı özelliklere sahip parçalara veya gruplara ayırma işlemini gerçekleştirmektedir. Aynı özelliklere sahip veriler, aynı kümede bulunmaktadır. Kümeleme işlemi sonucunda elde edilen kümeler, örneklerin ve veri noktalarının çizildiği alanda bir mekanizma oluşturur. Bu mekanizma, bazı nesnelerin kendi aralarında diğer nesnelerden daha güçlü bir benzerlik oluşturmasını sağlamaktadır [68]. Kümeleme analizi ile, yapay sinir ağları, şekil tanıma ve istatistik gibi birçok alanda çalışılmaktadır. Marketlerde farklı müşteri gruplarının keşfedilmesi ve bunların alışveriş örüntülerinin belirlenmesi, örüntü tanıma, veri analizi, resim tanıma, metin madenciliği, pazarlama, uzaysal veritabanı uygulamaları, coğrafik analizler birer kümeleme uygulamasıdır [74, 84,85].

Kümeleme analizinde kullanılan algoritmalar aşağıdaki gibi gruplandırılmaktadır.

Bölümlendirme Algoritmaları: n tane nesnenin olduğu veritabanında, nesneler mantıksal gruplara ayrılarak analiz edilir. Küçük ve orta boyutlu veritabanlarında birkaç grup olabilirken, veritabanı büyüdüğünde daha çok grup oluşabilir. Gruplandırma yapılırken değişik ölçütler değerlendirilebilir. Yapılan gruplandırma, analizin kalitesine etki eder. Bölümleyici algoritmalar, çok büyük olmayan veri setlerinde küresel kümelerin bulunmasında en iyi sonucu vermektedir. En çok olarak kullanılan bölümlendirme algoritmaları; k-means, k-medoids, EM (Expectation-Maximization) ve CLARA (Clustering Large Applications)-CLARANS (Clustering Algorithm based on Randomized Search)'tır [61,86].

Yoğunluk Tabanlı Algoritmalar: Birçok kümeleme yöntemi, nesnelerin birbirleri arasındaki farklılıklarına göre kümeleme yaparken, bu yöntem nesnelerin yoğunluğuna göre gruplama yapmaktadır. Yoğunluk, analiz edilen nesnelerin sayısını belirtmektedir. Bu algoritmalarda, yoğun veri bölgeleri veri kümesi olarak kabul edilmektedir. Yoğunluk tabanlı yöntemlere örnek olarak DBSCAN (Density Based Spatial Clustering of Applications with Noise) ,OPTICS (Ordering Points To Identify the Clustering Structure) ve DENCLUE (Density based Clustering) algoritmaları verilebilir [61].

Hiyerarşik Algoritmalar: Analiz yapılmadan önce nesnelerin hiyerarşik bir yapıya göre düzenlenmesi gerekmektedir. Veriyi hiyerarşik bir yapıya çevirmek için değişik yöntemler kullanılır. Veri nesneleri arasındaki hiyerarşik ilişkiyi göstermek için genelde “dendrogram” yapısı kullanılır. Bu yapı, veri nesnelerini veya küçük kümeleri birleştirerek ya da büyük kümeleri bölerek oluşturulur. Hiyerarşik kümeleme metotları, dendrogramın oluşturulma yöntemine bağlı olarak iki bölümde incelenir. Bunlar, birleştirici ve ayırıcı hiyerarşik kümeleme yöntemleridir [87].

Izgara Tabanlı Yöntemler : Nesneleri izgara yapısı oluşturacak şekilde sayılarına göre sınıflandırır. Temel avantajı hızlı tamamlanması ve nesnelerin sayısından bağımsız olmasıdır [78].

Model Tabanlı Yöntemler: Her küme için bir matematiksel model belirlenmesi ve bu modele uyan verilerin uygun kümeye yerleştirilmesi işlemidir. İstatistiksel veya yapay sinir ağları yaklaşımlarından birisi kullanılarak işlem yapılabilmektedir [73, 88].

III.4.2. K-Means Kümeleme Algoritması

Bu çalışmada kümeleme analizi için, k-means algoritması kullanıldığından, bu algoritmanın ayrıntıları bu kısımda verilmektedir.

K-means kümeleme algoritması, kümeleme problemini çözmek için MacQueen tarafından 1967 yılında sunulmuş öğreticisiz bir algoritmadır [89]. Bu algoritmadan, bilimsel ve endüstriyel alanda sıkça yararlanılmaktadır [90].

Kümeleme işlemi öncelikle k adet kümenin önceden belirmesi işlemi ile başlamaktadır. Veri kümesi içersinden seçilen k adet küme ise, her bir kümenin merkezini (centroid) veya orta noktasını temsil etmektedir. Geriye kalan nesnelerden her biri kendisine en yakın küme merkezine göre, kümelere dağıtılmaktadır. Kümeye yerleştirme işleminde, veri noktası ile kümenin merkezi arasındaki uzaklık ölçüt olarak kullanılmaktadır. Bir veri noktası hangi kümeye yakın ise o kümeye ait olmaktadır. Bu işlem sonucunda her bir veri noktası k adet kümeye yerleştirilmiş olacaktır. Oluşturulmuş yeni kümeler için merkez değeri hesaplanır ve bu değer kümenin yeni merkezi olur. Merkez noktası değiştiği için, veri kümesinin merkez noktalarına uzaklıkları da değişecektir. Bu nedenle, her bir veri değerinin kümelere yerleştirilmesi ve bu yerleşime göre oluşan kümelerin merkez noktalarının bulunması işlemi tekrarlanır. Bu işlem, bir kümedeki veri noktalarının kümenin merkez

noktasına olan uzaklıkları toplamının değerindeki düşüş en az oluncaya kadar sürmektedir. İşlem sonucunda, küme merkezleri birbirinden mümkün olduğunca uzak, küme içersindeki veri noktaları ise birbirine yakın olmaktadır [61, 91].

K-means algoritmasının çalışmasına ait adımlar aşağıda belirtilmektedir.

1. İlk küme merkezi olarak, veri kümesi içersinden k adet rasgele nokta belirlenir.

2. Her veri noktasının belirlenen merkezlere olan uzaklıkları hesaplanır ve veri noktası kendisine en yakın kümeye ait olacak şekilde işaretlenir.

3. Kümelerin yeni merkez noktaları ve o kümedeki tüm veri noktalarının uzaklıkları yeniden hesaplanır.

4. Merkez noktalar değişmeyene kadar 2. ve 3. adımlar tekrarlanır

Kümeleme işleminde kümelerin merkez noktaları ve küme içindeki nesneler arası uzaklık için kullanılan hata parametresini hesaplamak gerekmektedir. Hata parametresi hesaplamak için, üç farklı hesaplama yöntemi kullanılmaktadır [92-94].

Öklit Uzaklığı ve Öklit Uzaklığının Karesi : Öklit uzaklığı ve öklit uzaklığının karesi formülleri k-means algoritmasında sıkça kullanılan hata parametresi (uzaklık) hesaplama yöntemleridir. Öklit uzaklığı ve Öklit Uzaklığının Karesi formülleri Denklem III.4 ve Denklem III.5’de görülmektedir. Bu denklemlerde, x_i ve y_i , n boyutlu verideki i. boyuta ait veri değerini temsil etmektedir.

Öklit Uzaklık Formülü:

$$uzaklık(x, y) = \left\{ \sum_{i=1}^n (x_i - y_i)^2 \right\}^{1/2} \quad (III.4)$$

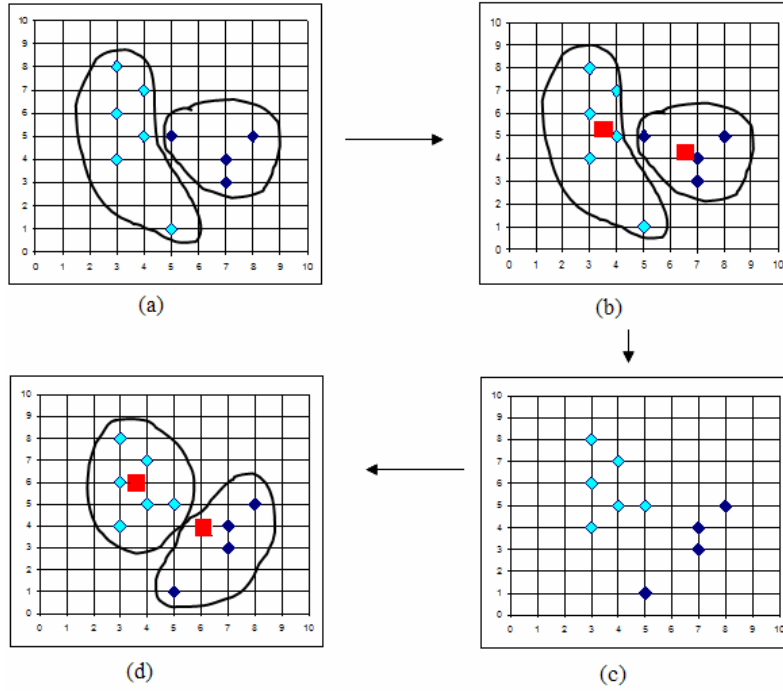
Öklit Uzaklığının Karesi Formülü:

$$uzaklık(x, y) = \sum_{i=1}^n (x_i - y_i)^2 \quad (III.5)$$

Manhattan uzaklık Formülü: Manhattan uzaklık hesaplaması için, boyutlar arasındaki ortalama fark işlemi yapılmaktadır. Bu yöntem sıra dışı verilerden az etkilenmekte ve daha kararlı sonuçlar vermektedir. Manhattan uzaklık hesaplaması Denklem III.6’da görülmektedir [61].

$$uzaklık(x,y) = \sum_{i=1}^n |x_i - y_i| \quad (III.6)$$

Şekil III.2’de k-means algoritmasının çalışmasına ilişkin bir örnek görülmektedir. Şekil III.2’de yer alan noktaların iki kümeye ayrılmak istendiği varsayılırsa, oluşturulacak küme sayısı $k=2$ olmaktadır [61,94].



Şekil III.2 K-means algoritması ile kümeleme örneği (küme sayısı=2)

Şekil III.2 (a)’da, birinci adımda iki nokta rasgele seçilerek, iki kümenin merkezi olarak belirlenmiştir. Geriye kalan noktalar ise, merkeze olan uzaklıklarına göre kümelenmiştir. Ayrılmış noktalara göre, her iki kümenin yeni merkez noktası hesaplanmıştır. Yeni merkezler, Şekil III.2 (b)’de kare ile gösterilmiştir.

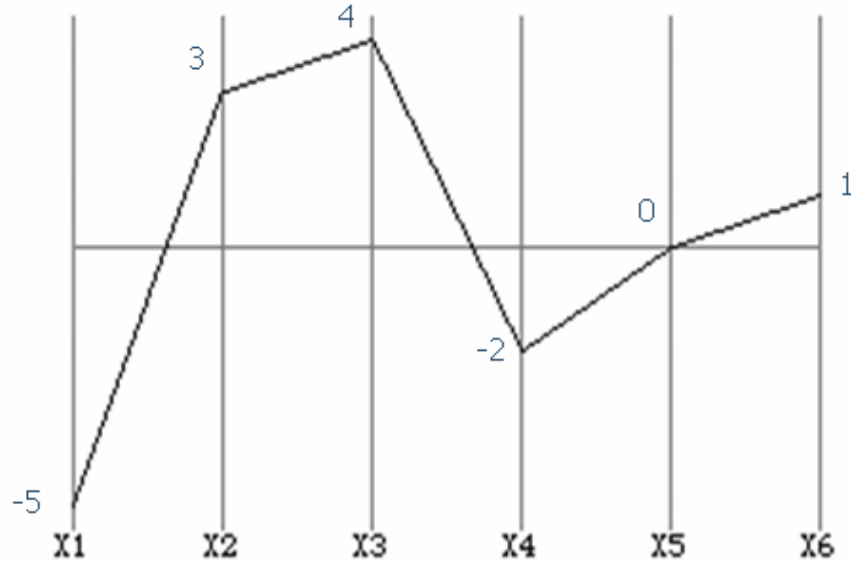
Yeni merkez noktalarına göre, veri noktaları kendilerine en yakın merkeze ait olacak şekilde tekrar kümelenir. Şekil III.2 (c)’de, yeni küme dağılımı görülmektedir. Bu aşamada, (5,1) ve (5,5) koordinatlarına sahip iki veri noktasının ait olduğu küme değişmiştir. Veri noktalarının dağılımı değiştiği için, küme merkezleri de değişmiştir. Yeni küme merkezleri, Şekil III. 2 (d)’de yine kare ile temsil edilmiştir.

Artık merkez noktalarda herhangi bir değişiklik olmadığı için, algoritma son bulmuş ve Şekil III. 2 (d)’ de görülen iki adet küme ortaya çıkmıştır [61, 95].

III.5. ÇOK BOYUTLU VERİLERİN GÖRSELLEŞTİRİLMESİNDE PARALEL KOORDİNAT SİSTEMİNİN KULLANILMASI

Veri madenciliği sonuçlarını, basit, anlaşılır, kullanılabilir biçime dönüştürmek için sonuçların görselleştirilmesi gerekmektedir. Görsel veri madenciliği ile sonuçların hızlı bir şekilde anlaşılması, sonuçlar çıkartılması ve araştırmacıya kılavuzluk yapması sağlanmaktadır. Veri madenciliği sonuçlarını görselleştirme aşamasında, özellikle üç boyuttan daha fazla boyut içeren verilerin bilgisayar ortamında görüntülenmesi ve algılanmasında bilgi kaybı ve anlaşılabilirliği konularında çeşitli zorluklar yaşanmaktadır [96]. Ekran, yazıcı ve çizici (plotter) gibi çıkış aygıtları da iki boyutlu işlemler yapmaktadır. Üç boyutlu veya daha fazla verinin iki boyutlu düzlem üzerine taşınması ise, verinin gerçek yapısının bozulmasına ve karmaşaya neden olmaktadır. Bu problemi çözmek için, kullanılan yöntemlerden birisi de paralel koordinat sistemidir [97].

Paralel koordinat sistemi ilk defa Inselberg tarafından tanımlanmıştır [98]. Bu sistemde, n -boyutlu veriyi iki veya üç boyutlu alan üzerine taşımak gerekmemektedir. Bunun yerine, x eksenine paralel olacak şekilde tüm boyutlar için birer eksen oluşturulmaktadır [99]. Şekil III.3'de altı boyuta sahip $(-5,3,4,-2,0,1)$ değerleri paralel koordinat sisteminde görselleştirilmiştir. Şekil III.3'de Paralel koordinat sisteminde, her çok boyutlu veri yatay eksenini kesecek şekilde çoklu dikey eksenlerde görüntülenmektedir. Verinin değeri ise, ilgili boyutu temsil eden eksen üzerinde işaretlenerek veriyi görselleştirmek için bir hattın işaretlenen tüm eksenlerden geçmesi sağlanmaktadır [100].



Şekil III.3 Altı boyuta sahip verinin paralel koordinatlardaki görüntüsü

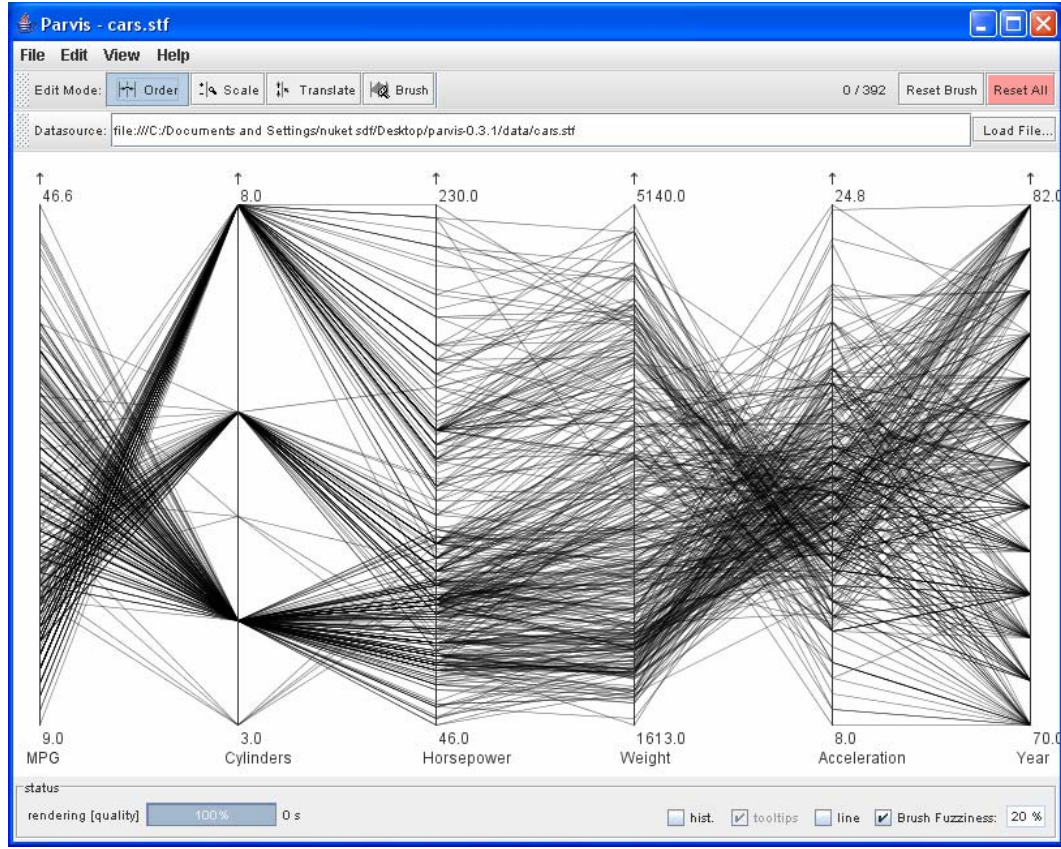
Paralel koordinat sisteminin temel avantajı, boyut sayısının yatay eksen alanı ile temsil edilmesi, bu nedenle de fazla boyutlu verilerinin karmaşaya neden olmamasıdır. Paralel koordinat sistemi ile karmaşık veri setlerinin görüntülenmesi sağlanır. Bu görüntülenme sistemi ile başka görüntüleme tekniklerinde elde edilemeyecek şekilde, verinin sezgisel genel yapısı ile ilgili görüntüleme yapılmış olur [101].

III.6. PARVİS YAZILIMI İLE PARALEL KOORDİNATLARDA ÇOK BOYUTLU VERİLERİN GÖRÜNTÜLENMESİ

Parvis (Parallel Coordinates Visualization), çok boyutlu veri setlerinin görüntülenmesini paralel koordinat sistemi ile yapan açık kaynaklı bir yazılımdır [102, 103]. Parvis yazılımındaki temel amaç, esnek, yeniden kullanılabilir kullanıcı arayüzü oluşturmaktır. Parvis aracılığı ile bu konudaki en son teknolojiler kullanılarak, paralel koordinat sistemiyle, veri setinin görüntülenmesi ve kullanıcının veriyle görsel etkileşim içinde olması sağlanır. Özellikle büyük veri setlerinde eş zamanlı yüksek kalitede görselleştirme işlemi çok zaman gerektirmektedir. Parvis uygulaması ile hızlı bir şekilde verinin görüntülenmesi sağlanmaktadır.

Sınırsız sayıda boyuta sahip, çok boyutlu veri setleri Parvis aracılığı ile görüntülenebilir. Şekil III.4’de örnek bir veri setinin Parvis’e yüklenmesi ile yapılan

görselleştirme görülmektedir. Şekil III.4'deki örnekte, veri setinde yer alan altı boyut ve veri setinde yer alan 392 veri örneği görülmektedir [103].



Şekil III.4 Parvis'e yüklenerek görselleştirilen örnek veri seti [103]

Parvis'de yüklenen veri setindeki her bir boyut bir eksen ile temsil edilir ve verinin değeri bu eksen üzerinde işaretlenir. Her boyut için veri değeri eksen üzerinde belirlendikten sonra eksenlerdeki bu noktalar bir hat ile çizdirilerek veri değeri görsel hale getirilir. Parvis yazılımında, görselleştirilmiş veri kümeleri seçilebilmekte, bu veri öğelerine ait hatların farklı renklerde görüntülenmesi ve seçime dahil olan veri öğelerinin kaydedilmesi sağlamaktadır.

III.7. VERİ MADENCİLİĞİNİN EĞİTİM ALANINDA KULLANILMASINA İLİŞKİN LİTERATÜRDEKİ ÇALIŞMALAR

Öğrenciyle ilgili bilgileri içeren örün tabanlı öğretim sistemleri, yardımcı ders yazılımları gibi bilgisayar programlarında ve eğitim-öğretim faaliyetlerine ait verilerde, veri madenciliği kullanımına ilişkin literatürde yer alan örnekler bu başlık altında incelenmektedir.

Zeki öğretim sistemi içindeki öğrencilerin sınıflanması ve öğrencilerin gelecekte gösterecekleri başarının tahmin edilmesi için Bidgoli ve arkadaşları tarafından yapılan çalışmada, LON-CAPA (Learning Online Network with Computer-Assisted Personalized Approach) örün tabanlı eğitim sisteminde veri madenciliği kullanılmıştır. Bu çalışmada amaç, öğrencilerin sistemi kullanmasından yola çıkarak, yıl sonu notlarını tahmin etmektir. Çoklu sınıflayıcı kullanımı sınıflamanın başarısını büyük ölçüde artırmıştır. Genetik Algoritmadaki ağırlıklı vektör yapısı kullanılarak doğruluğun optimizasyonu sağlanabilmektedir. Sistemde çoklu sınıflayıcı kullanılırken Genetik Algoritma kullanılmıştır ve %10 oranında doğruluk artışı sağlanmıştır [104].

Öğretmenler için öğrettikleri ders ile ilgili pedagojik örüntülerin bulunması da, derste yapacakları planlama, takip edecekleri yöntemi belirlemeleri açısından önemlidir. Merceron ve Yacef tarafından yapılan araştırmada, öğrenciler tarafından bilgisayar ortamında yapılan alıştırma sonuçlarını, öğretmenlerin görüntülemesi ve inceleyebilmesi için veri madenciliği işlemini gerçekleştiren veri analiz yazılımı geliştirilmiştir. Geliştirilen yazılım ile öğretmen tarafından belirlenen ölçütlere göre, karar ağaçları kullanılarak benzer öğrenci sınıfları oluşturulmaktadır. Böylece öğrencilerin öğrenme güçlükleri belirlenir ve yazılım sonucunda bulunan gruplara öğretmen gerekli yardımı sağlar [105].

Romero ve arkadaşları tarafından gerçekleştirilen çalışma ile, örün tabanlı bir eğitim sisteminde, öğrencinin sistemi kullanması sonucu oluşan bilgiler yardımıyla, uyarlamalı içeriğe sahip örün kurslarını geliştirmek için ilginç(interesting) kuralların nasıl bulunabileceği ele alınmıştır. Kurs içeriğinin tasarımcılarına geri bildirim sağlamak için, veri madenciliği kullanılmıştır. Bulunan birliktelik kuralları, “IF THEN” şeklinde sunulmuş ve kursun gelişimi için kullanılmıştır. Örneğin farklı konular arasında, IF Konu1’deki seviyesi uzman THEN Konu2’de de seviyesi uzmandır şeklinde, konular arasındaki seviye, başarı ve sistemi kullanma zamanı gibi parametreler arasındaki ilişkilere ait kurallar bulunmuştur [106].

Yu ve arkadaşları tarafından gerçekleştirilen çalışmada veri madenciliği teknolojileri kullanılarak hatalı öğrenci davranışları ortaya çıkartılmıştır. Örün kayıtları (log) kullanılarak bulanık birliktelik kuralları ile sistemde kayıt edilen öğrenci hareketleri (harcanan zaman, okunan makale, sorulan soru gibi) arasındaki ilişkiler bulunmuştur [107].

Shen tarafından yapılan çalışmada örün tabanlı bir eğitim sisteminde yer alan veriler üzerinde veri madenciliği ve durum tabanlı sorgulama yöntemi kullanılmıştır. Veri madenciliği ile kullanıcı örüntüleri ve davranışları üzerinde çalışılmış ve öğrencilerin öğrenme davranışlarına göre sınıflandırılması sağlanmıştır. Durum tabanlı sorgulama ile soru-cevap sisteminin davranışlarının düzenlenmesi ve öğrencinin sanal öğretmene aktif olarak soru sorması sağlanmıştır [108].

Slovenya lisesindeki 889 öğrenciye ait veri üzerinde yapılan veri madenciliği analizi ile, gelecekteki başarılarının tahmin işlemi gerçekleştirilmiştir. Kümeleme, karar ağacı ve hiyerarşik yöntemlerle öğrenci verilerindeki örüntüler bulunmuştur. Öğrencilerin başarı durumu dikkate alınarak, yapacakları kolej seçiminde yardımcı olabilmek için bu analiz gerçekleştirilmiştir. Çoklu nitelik modelleme ile öğrenci hakkında geleceğe yönelik tahminler yapılmaktadır [109].

Tang ve Chan tarafından gerçekleştirilen çalışmada öğrenci gruplarının oluşturulması için e-öğrenme sisteminde bulunan öğrencilerin sistemdeki gezinti (browsing) verileri üzerinde ağırlıklı öklit uzaklığı kullanılarak kümeleme işlemi yapılmıştır [110].

Chen ve arkadaşları tarafından yapılan çalışmada, grupla öğretim yapan örün tabanlı eğitim sisteminde yer alan veriler üzerinde DBMiner yazılımı ile birliktelik kuralları oluşturulmaktadır. Bu kurallar, grupların sisteme kaç kez bağlandığı, ödev notu, cinsiyet, örün üzerindeki bir tartışma konusunda cevap verme veya okuma sayısı gibi bilgileri içermektedir. Çıkan kurallara göre, grupları yönlendirme ve öğrencilerin başarısını artırmak amaçlanmaktadır [111].

Zaiane ise, örün tabanlı eğitim sistemlerindeki kayıtlarda (log) WebSIFT, WebLogMiner gibi yazılımların kullanılmasını önermektedir. Zaiane' göre, örün tabanlı eğitim sisteminde yer alan sayfalarda sık ziyaret edilen veya hiç ziyaret edilmeyen sayfaların belirlenmesinde, sayfa ziyareti ve öğrenci aktiviteleri arasındaki ilişkilerin bulunması veya başarılı öğrencilerin ziyaret ettikleri sayfaların belirlenmesinde veri madenciliği analizi sonuçları kullanılmalıdır. Bu bilgiler ile, eğitim sisteminin daha etkin hale getirilmesi ve başarısız öğrencilere rehberlik edilmesi sağlanır [112].

Myller ve arkadaşları tarafından yapılan çalışmada, bilgisayar programlama dersinden öğrencilerin sınıfta yapılan alıştırılardan ve sınavlardan aldıkları puanlar kullanılarak kümeleme yapılması sağlanmıştır. Kümeleme işleminde amaç, farklı becerilere sahip heterojen grupların oluşturulmasıdır [113].

TechBC ve Virtual-U örün tabanlı öğretim sistemlerinin kayıt (log) dosyaları üzerinde Zaiane ve Luo tarafından veri madenciği analizi gerçekleştirilmiştir. Bu sistemdeki örün kayıtları ile birlikte sık ziyaret edilen sayfalar birliktelik kuralı ile bulunmuştur [114].

Bu çalışmalarda görüldüğü gibi, eğitim alanında veri madenciliğinden örün tabanlı eğitim sistemlerinde yararlanılmaktadır.

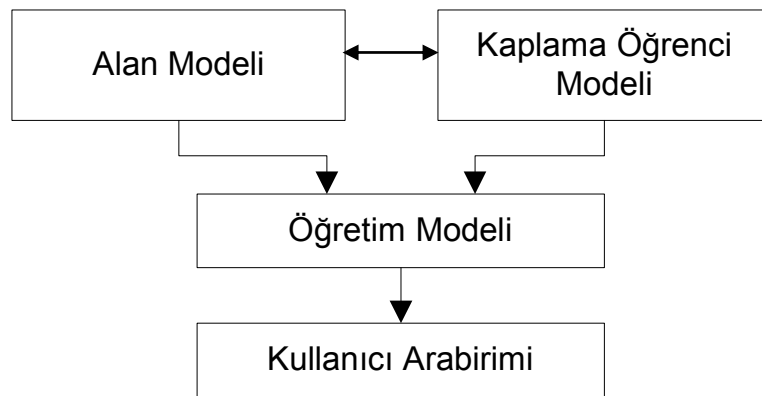
BÖLÜM IV

GERÇEKLEŞTİRİLEN ZEKİ ÖĞRETİM SİSTEMİNİN TASARIMI

IV.1. AMAÇ

Bu bölümde, istenilen herhangi bir alan için kullanılmak üzere tasarlanan örün tabanlı zeki öğretim sisteminin yapısı, bileşenleri ve bu bileşenlerde kullanılan modeller açıklanmaktadır.

Tasarlanan zeki öğretim sistemi (ZÖS) Şekil IV.1’de görüldüğü gibi, öğrenci bilgisinin düzeyini temsil eden öğrenci modeli, öğrencinin bilmesi gereken konuları içeren alan modeli, uygun pedagojik kararların alınmasını sağlayan öğretim modeli ve öğrenciyle iletişim kurulmasını sağlayan kullanıcı arabiriminden oluşmaktadır.



Şekil IV.1 Tasarlanan ZÖS’e ait mimari yapı

Zeki öğretim sistemlerinin en temel amacı, öğrenciye göre uyarlamalı bir ortam sağlayabilmektir. Bunu gerçekleştirebilmek için sistem, öğrencinin hangi konuyu bildiğini, hangi konuyu bilmediğini anlayabilmeli ve o anki öğrenci durumuna bağlı

olarak karar verebilmelidir. Öğrenci bilgi durumu, öğrenci modeli ile temsil edilmektedir ve sistem herhangi bir öğretim etkinliği gerçekleştirmek için öğrenci modeli ile etkileşim içerisinde olmalıdır. Öğrenci sistemi kullandıkça sistem, öğrenci modelini güncellemeli ve her öğrenci için kişisel bilgileri kontrol ederek o doğrultuda öğretim hizmeti sunabilmelidir.

Bu tezde kaplama öğrenci modeli [19] kullanan bir ZÖS tasarlanmıştır. Kaplama öğrenci modeli ile, öğrenci bilgisinin düzeyi belirlenmektedir. Alan modelinde konuların birbirleriyle ilgili önşart ilişkisi bulunmaktadır. Böylece öğrenci, tasarımcı tarafından belirlenen konular dizisi içerisinde hareket ederek sistemde ilerlemektedir. Öğrencinin herhangi bir konudan aldığı başarı durumunun saptanmasında ise, MYCIN uzman sisteminde kullanılan güven faktörü hesaplaması kullanılmaktadır. Bu yöntem ile, öğrencinin başarı durumu ile birlikte, bilgi eksikliğinin de hangi düzeyde olduğu konusunda bir sonuç oluşturulmaktadır [115].

Bu başarı puanına göre, öğretim modeli öğrenciyi uygun konuya yönlendirerek sistemin hangi konudan devam edeceğine karar vermektedir. Öğrenciyle iletişim kurmak için ise, tasarlanan kullanıcı arabirimi ile öğrenci sisteme giriş yapmakta, konu alanı içerisinde ilerlemekte, sistemdeki soruları yanıtlanmaktadır.

Gerçekleştirilen ZÖS’de konu alanı olarak, Marmara Üniversitesi, Teknik Eğitim Fakültesi, Elektronik-Bilgisayar Eğitimi Bölümü’nde 8.dönemde yer alan “Bilgisayar Sistemleri” dersi belirlenmiştir. Gerçekleştirilen ZÖS, örün tabanlı olarak veya yerel bir Internet bilgi sunucusu (Internet Information Server) üzerine yüklenerek yerel bir ağ üzerinde de çalışmaktadır. Sistemdeki bileşenlerin gerçekleştirilmesinde uygulama aracı olarak Visual Studio.NET içerisinde yer alan ASP.NET örün uygulamaları geliştirme programı kullanılmıştır.

Tasarlanan sistemde bulunan alan modeli, kaplama öğrenci modeli ve öğretim modeli; kullanıcı arabirimini kullanarak yerel bir ağ veya Internet üzerinden öğrenci ile iletişim kurmaktadır.

IV.2. ALAN MODELİ

Zeki öğretim sistemlerinde alan modeli, öğrenciye sunulacak olan konu ile ilgili kuralları, bilgileri, soruları ve çözümleri içeren bölümdür. Sistemde alan modelinin tasarlanması için, öncelikle alanda yer alan her bir konunun ve konular

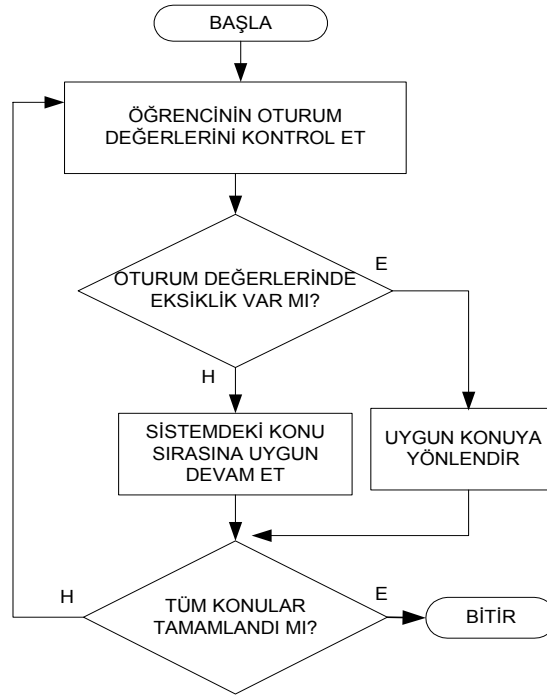
arası ilişkilerin belirlenmesi gerekmektedir. Alandaki her konu ASP.NET tasarım ortamında hazırlanan aspx uzantısına sahip dosyalar içerisinde metin ve şekiller yardımıyla açıklanmıştır. Her bir konu içerisindeki dosyalar, sistem tasarımcısının belirlediği sıra ile öğrencinin karşısına gelmektedir. Öğrenci sisteme bağlandığında, öncelikle sahip olduğu öğrenci numarası ile veritabanı kontrol edilerek, en son hangi konuyu öğrenmeye çalıştığı belirlenmekte ve bu konuyu takip eden, öğrencinin henüz çalışmadığı ilk konu karşısına çıkmaktadır. Her bir konunun sonunda, konuyla ilgili çoktan seçmeli soruların öğrenci tarafından yanıtlanması gerekmektedir. Soruların yanıtlanması sırasında, güven değeri, yani öğrencinin konuyu ne düzeyde bildiğini gösteren sayısal değer Bölüm IV.4’de açıklandığı gibi MYCIN güven faktörü hesaplama yöntemi ile bulunmaktadır.

Sistemde bir konu içerisinde yer alan kavramların anlatımı da sistem tasarımcısının belirlediği sıralama ile öğrencinin karşısına gelmektedir. Zeki öğretim sistemlerinde, öğrencinin o anki bilişsel yükünü azaltmak ve gerçekte odaklanması gereken konudan öğrencinin uzaklaşmaması için birden fazla bağlantıyı öğrencinin karşısına çıkarmak çok tercih edilmemekte ve sistemin yönlendirme yapması sağlanmaktadır.

Tasarlanan sistemde, konulara ait sayfaların öğrenci tarafından ziyaret edilip edilmediğinin belirlenmesi ve doğru bir şekilde öğrencinin karşısına gelmesi için, öğrenci sisteme bağlanarak oturum açtıktan sonra, ASP.NET nesnelerinden Session (Oturum) nesnesi kullanılarak kontrol yapılmaktadır. Oturum nesnesi, verilerin oturum boyunca saklanmasını sağlamaktadır. Oturum sayesinde kullanıcıya ait veriler sayfalar arasında kullanılmaktadır. Bir oturum, örün tarayıcısının açılıp kapanmasına kadar geçen süredir. Öğrenci sisteme bağlandığı andan itibaren, sadece ona ait özel değişkenler kullanılarak, bulunduğu konu içerisindeki ziyaret sırasını ve hangi sayfaları ziyaret ettiğini belirlemek mümkün olmaktadır. Bu şekilde, öğrenciye tasarımcının belirlediği sayfa yönlendirmesi yapılmaktadır. Bu işlem için öğrenci herhangi bir konuyla ilgili sayfayı ziyaret ettiğinde o konuya ilişkin Oturum değeri 1 değerine eşitlenmektedir. Örneğin, Konu 1.1’ e ait sayfanın ziyaret edildiğini belirtmek için, *Session(“Konu 1.1”) = 1* komutu kullanılmaktadır. Konu 1.1.’i takip eden sayfa, örneğin Konu 1.2’ye ait sayfa yükleneceğinde, bu sayfanın yüklenmesi sırasında *Page_Load()* olayı içerisinde Konu 1.1’ e ait oturum değeri kontrol edilmektedir. Yapılan kontrolde, Session(Konu 1.1)’in değeri 1’den farklıysa, öğrencinin Konu 1.1’e yönlendirilmesi sağlanmaktadır. Diğer sayfa geçişleri

sırasında da benzer kontroller yapılarak, konu içerisindeki anlatım sırasının kontrol edilmesi ve öğrencinin bağlantılar arasında kaybolmadan konuyu tamamlaması amaçlanmaktadır.

Tasarlanan sistemde, alan modelinin takibine ilişkin akış şeması Şekil IV.2’de görülmektedir. Oturum değerinin kontrolü konular tamamlanıncaya kadar devam etmekte ve yapılan kontroller sonucunda öğrenci uygun konuya yönlendirilmektedir.



Şekil IV.2 Tasarlanan sistemin alan modelinin akış şeması

Konu alanı olarak seçilen “Bilgisayar Sistemleri” dersi Tablo IV.1’de görüldüğü gibi toplamda altı konuya ayrılmıştır.

Tablo IV.1 Tasarlanan zeki öğretim sisteminde yer alan konular

Konu Adı
1) Giriş
2) Veriyolu Yapıları
3) Bellek Yapıları
4) İşlemci Yapısı ve Fonksiyonları
5) Komut Düzeyinde Paralellik
6) Komut Takımı ve Adresleme Modları

Her konu anlatımının sonunda, konuyla ilgili hazırlanmış olan çoktan seçmeli sorular öğrencinin karşına gelmektedir. Bu soruların her biri zorluk ve önem

derecesine göre, tasarımcının belirlediği farklı ağırlık değerlerine sahiptir. Öğrenci konuyla ilgili soruları yanıtladıktan sonra, öğrencinin sistemdeki başarısına göre, diğer konulara ilerlemesi sağlanmaktadır. Eğer öğrenci için hesaplanan güven değeri, o konu için belirlenmiş güven değerine eşit veya büyükse, öğrenci bir sonraki konuya geçmektedir. Eğer öğrenci, belirlenen sınır değerinden daha düşük bir puan alırsa, konuyu tekrar etmesi için aynı konuya yönlendirilmektedir.

Konuların sonunda yer alan sorular sistemde Microsoft Access veritabanında “TblSorular” isimli tabloda tutulmakta ve Tablo IV.2’deki alanları içermektedir:

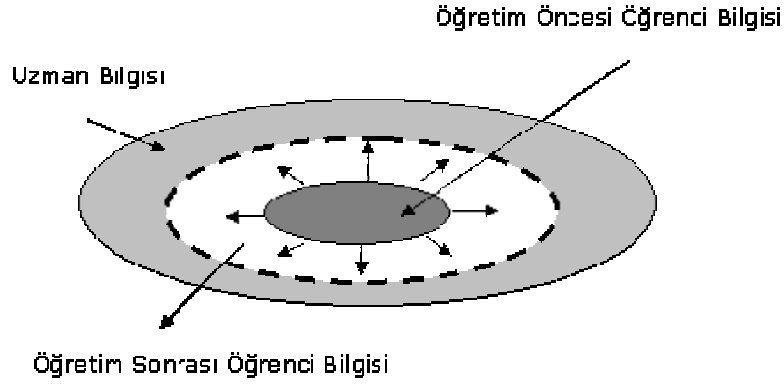
Tablo IV.2 TblSorular tablosunun içerdiği alanlar

Alan Adı	Açıklama
SoruNo	Soru Numarası
SınavKodu	Soru grubunun hangi konuyla ilgili olduğuna bağlı olarak değişen sınav kodu değeri
Soru	Soru metni
Cevap1	A seçeneğinin değeri
Cevap2	B seçeneğinin değeri
Cevap3	C seçeneğinin değeri
Cevap4	D seçeneğinin değeri
Cevap5	E seçeneğinin değeri
Doğru Cevap	Sorunun doğru yanıt seçeneğinin değeri
Ağırlık	Sorunun 0 ile 1 arasında değişen ağırlık değeri
Konu	Sorunun hangi konu ile ilgili olduğunu gösteren değer

IV.3. ÖĞRENCİ MODELİ

Tasarlanan sistemde, öğrenci bilgisinin modellenmesi için kaplama öğrenci modeli kullanılmaktadır. Kaplama öğrenci modelinde Şekil IV.3’de görüldüğü gibi, öğrenci bilgisi uzman bilgisinin bir alt kümesi olarak görülmektedir. Bu modelde, konu içinde öğrencinin, uzmana göre ne kadar ilerleyebildiği gösterilmekte ve alanda uzmanın bilmesi gereken her öğenin düzeyi belirlenmektedir. [8,19]. Ayrıca, kaplama öğrenci modelinde öğrencinin bilgisi, uzman bilgisi ile karşılaştırılarak öğrenci bilgisinin uzman modeli içindeki düzeyi de belirlenmektedir. Böylece öğrencinin neleri bildiği ve neleri bilmesi gerektiğine karar verilir. [116, 117].

Kaplama öğrenci modeli; konu alanının kurallar ve konular şeklinde genel parçalara ayrılabilirdiği alanlarda daha başarılı olmaktadır. Bu nedenle, alt konulara parçalanabilen Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Bilgisayar-Kontrol programı 8. dönem derslerinden “Bilgisayar Sistemleri” alan olarak seçilmiştir.



Şekil IV.3 Kaplama Öğrenci Modeli [116]

Tasarlanan ZÖS’de, öğrenci sisteme bağlandığında, öncelikle alanla ilgili hazırlanan konu alanı sayfalarını ziyaret etmektedir. Her konu sonunda, konuyla ilgili hazırlanan çoktan seçmeli sorulardan oluşan sınav, öğrenci tarafından yanıtlanmaktadır ve bu yanıtlara göre konuyla ilgili, başarı puanını temsil etmek için kullanılan güven değeri hesaplanmaktadır. Sistemde her konu için öğrencinin o konuda aldığı güven değeri kaydedilmektedir. Güven değeri hesaplaması için, öğretim modelinde ayrıntılı olarak açıklanan MYCIN güven faktörü hesaplama yöntemi kullanılmaktadır. Öğrencinin konuyla ilgili aldığı güven değeri, konunun öğrenci tarafından ne kadar bilindiğini ve bilgi eksikliği düzeyinin hangi durumda olduğu ile ilgili sistem inancını göstermektedir. Bu güven değeri, öğrenci sistemde sorulara yanıt verdikçe değişmektedir. Bu değer en yüksek ve en düşük değeri tasarımcı tarafından belirlenmektedir.

Tasarlanan sistemde, her öğrenci sisteme bir kullanıcı adı ve şifreyle bağlanmaktadır. Sisteme bağlanan öğrenciler için Tablo IV.3’de görüldüğü gibi “TblKullanıcılar” isimli tabloda; öğrencinin elektronik posta, okul numarası, isim, şifre ve sınıfına ait bilgiler tutulmaktadır. Öğrenciyle ilgili tüm tablolar Microsoft Access veritabanı dosyasında yer almaktadır.

Tablo IV.3 TblKullanıcılar Tablosuna Ait Alanlar

Alan Adı	Açıklama
E-Posta	Öğrenciye ait elektronik posta adresi
Numara	Öğrencinin okul numarası
İsim	Öğrencinin isim ve soyadı bilgileri
Şifre	Öğrencinin sisteme bağlanırken kullanacağı şifre
Sınıf	Öğrencinin devam ettiği sınıfın adı

Öğrencilerle ilgili tutulan diğer bir tablo ise, “TblPuan” tablosudur. Bu tablo içerisinde her öğrencinin konulardan aldığı güven değeri ve her konuya ait tekrar değeri bulunmaktadır. Tablo IV.4’de görüldüğü gibi, “TblPuan” tablosu içerisinde; öğrencinin sistemi kullandığı tarih, Konu 1’den Konu N’e kadar olan alanlar içerisinde çalıştığı konular, Puan1’den Puan N’e kadar olan alanlar içerisinde çalıştığı konulardan aldığı güven değerleri tutulmaktadır. Öğrenci başarısız olduğunda konuyu tekrar etmektedir ve bu tekrarın kaç kez yapıldığı her konu için, konu numarasına göre Sayaç 1’den Sayaç N’e kadar olan alanlar içerisinde tutulmaktadır. Örneğin, sistemde Konu 1’in kaç kez tekrar edildiği Sayaç 1 alanına, Konu 1’den alınan güven değeri ise Puan 1 alanına kaydedilmektedir.

Tablo IV.4 TblPuan Tablosuna Ait Alanlar

Alan Adı	Açıklama
Numara	Sistemi kullanan öğrencinin okul numarası
Tarih	Öğrencinin sistemi kullandığı en son tarih
Konu 1 : Konu N	Alana ait konuların sayısı sistem tasarımcısı tarafından belirlenmektedir. Bu alanlarda konuların adı tutulmaktadır.
Puan 1 : Puan N	Öğrencinin her konudan aldığı güven değeri konuya ait Puan alanında saklanmaktadır. Örneğin Konu 1’ e ait güven değeri Puan 1 alanında saklanmaktadır.
Sayaç 1 : Sayaç N	Öğrencinin her bir konuyu kaç kez kullandığına dair tekrar değeri bu alanlarda tutulmaktadır. Örneğin Konu 1’in kaç kez tekrar edildiği Sayaç 1 değişkeninde tutulmakta ve her tekrar edildiğinde bir artırılması sağlanmaktadır.

Ayrıca sistemde, öğrencinin konularla ilgili her bir soruya verdiği yanıt ve yanıtın doğru veya yanlış olma durumuna ait durumun saklandığı “TblCevaplar” isimli tablo bulunmaktadır. Bu tablo, tüm öğrencilerin sorulara verdiği yanıtlardan oluşmakta ve Tablo IV.5’de belirtilen alanları içermektedir. Bu tablo içinde öğrencinin okul numarası Numara alanında, öğrencinin soruyu yanıtladığı andaki tarih ve saat bilgisi Tarih alanında tutulmaktadır. Öğrencinin hangi yanıt seçeneğini işaretlediğini gösteren sayısal değer ise, Ogr_Cevap alanına kaydedilmektedir. Ogr_Cevap alanında 1 değeri A seçeneğini, 2 değeri B seçeneğini, 3 değeri C seçeneğini, 4 değeri D seçeneğini ve 5 değeri E seçeneğini temsil etmektedir. Sorunun doğru yanıtlanıp yanıtlanmadığını gösteren değer ise, Durum isimli alanda saklanmaktadır. Durum alanında, öğrenci soruyu doğru yanıtlamışsa 1 değeri, yanlış yanıtlamışsa 0 değeri bulunmaktadır. Sorunun hangi konuya ait olduğunu gösteren değer ise, Konu Adı alanında tutulmaktadır.

Tablo IV.5. TblCevaplar Tablosuna Ait Alanlar

Alan Adı	Açıklama
Numara	Soruyu yanıtlayan öğrencinin okul numarası
Tarih	Öğrencinin soruyu yanıtlama tarih saati
Soru No	Soru Numarası
Ogr_Cevap	Öğrencinin hangi yanıt seçeneğini işaretlediğine dair sayısal değer (Bu alanda 1 değeri A seçeneğini, 2 değeri B seçeneğini, 3 değeri C seçeneğini, 4 değeri D seçeneğini ve 5 değeri E seçeneğini temsil etmek için kullanılmaktadır.)
Durum	Öğrencinin yanıtının durumunu gösteren sayısal değer tutulmaktadır. Bu alan, eğer öğrenci soruyu doğru yanıtlamışsa 1 değeri, yanlış yanıtlamışsa 0 değerini almaktadır.
Konu Adı	Sorunun hangi konuya ait olduğu

Öğrenci modeli, Şekil IV.4’de belirtilen akış şemasında, belirtilen aşamaları içermektedir. Öğrenci, öncelikle TblKullanıcılar tablosunda yer alan ve Tablo IV.3’de görülen alanlara ait bilgilerle sisteme kayıt olmaktadır. Bu bilgiler elektronik posta, okul numarası, isim, şifre ve sınıf alanlarından oluşmaktadır. Bu bilgiler kayıt edildikten sonra, ikinci adım olarak öğrenci kendine ait kullanıcı adı ve şifreyle sisteme bağlanabilmektedir. Daha sonra ise, tasarımcı tarafından belirlenen konu sırası ile öğrenci öncelikle konu alanında ilerlemektedir. Her konunun sonunda ise,

konuyla ilgili sınavın yer aldığı bölüm bulunmaktadır. Öğrencinin burada sorulara verdiği yanıtlara göre, her konu için güven değeri hesaplanmakta ve bu değer kaydedilmektedir. Öğrencinin aldığı güven değerine göre, öğretim modeli gerekli pedagojik kararı vermektedir.



Şekil IV.4 Tasarlanan sisteme ait öğrenci modeli akış şeması

IV.4. ÖĞRETİM MODELİ

Bu kısımda, öğrencinin herhangi bir konudan hangi durumlarda başarılı ya da başarısız olacağının, yapılacak konu yönlendirmesinin ve ne zaman konu tekrarı yapacağının kararları verilmektedir. Öğrencinin sistemde bir konuyu en fazla kaç kez tekrar edeceği sistem tasarımcısı tarafından belirlenir. Sistemde bilgi düzeyini hesaplama işlemi için, öğrencinin verdiği yanıtlardan ilgili konudan aldığı puan hesaplanmaktadır. Bu puanın hesaplanması işleminde MYCIN uzman sisteminde kullanılan GF (güven faktörü) hesaplaması kullanılmaktadır. Güven değeri, belli bir konudaki yargının ne kadar doğru, gerçek ve güvenilir olduğunu göstermektedir. Güven faktörünün amacı, herhangi bir kurala güven faktörlerinin eklenmesini sağlamaktır. Bu değer ile elde bulunan bilgilere bağlı olarak yargı ortaya çıkartılmaktadır [115].

Herhangi bir konudan öğrencinin başarılı sayılabilmesi için önceden belirlenmiş güven değerine ulaşması gerekmektedir. Bir konuyla ilgili güven değeri -1 ile +1 arasında değişmektedir. -1 değeri konunun bilinmediğini, +1 değeri ise

konunun bilindiğini temsil etmektedir. Öğrenci teste başlamadan önce tüm konular için güven değeri sıfırdır.

Bir konunun öğrenilip öğrenilmediğini test eden her soru için, iki değer bulunmaktadır. Değerlerden bir tanesi öğrencinin konuyu bildiği varsayım durumunu gösterirken, diğeri yanlış yanıtlaması durumunda konuyu bilmediği varsayım durumunu göstermektedir. Bir soruyla ilgili güven değerlerini temsil etmek için GF_d ve GF_y simgeleri kullanılmıştır. GF_d simgesi, soru doğru yanıtladığında alınacak puanı, GF_y ise soru yanlış yanıtladığında alınacak puanı temsil etmektedir. Bu değerler sistem tasarımcısı tarafından belirlenir.

Sistemde, bir konuyla ilgili öğrencinin önceki bilgisi (GF_0) ile birlikte soruya verdiği cevaba göre, yeni güven değerinin hesaplanmasında, MYCIN uzman sisteminde kullanılan hesaplama yöntemi kullanılmaktadır [60, 118].

Öğrenci soruyu doğru yanıtlarsa; güven değeri GF_d değerine, yanlış yanıtlarsa, GF_y değerine eşit olacaktır. Öğrencinin herhangi bir anda üzerinde çalıştığı sorudan alacağı güven değeri, Denklem IV.1, IV.2 ve IV. 3’de GF_a ile gösterilmiştir. Eğer öğrenci soruyu doğru yanıtlamışsa, GF_a değeri GF_d ’ye eşit olacaktır. Fakat öğrenci soruyu yanlış yanıtlamışsa, GF_a değeri, GF_y ’ye eşit olacaktır [60, 115, 118].

$$GF_{Yeni} = GF_0 + GF_a - GF_0 * GF_a \quad \text{Eğer } GF_0 \text{ ve } GF_a > 0 \quad (IV.1)$$

$$GF_{Yeni} = GF_0 + GF_a + GF_0 * GF_a \quad \text{Eğer } GF_0 \text{ ve } GF_a < 0 \quad (IV.2)$$

$$GF_{Yeni} = \frac{GF_0 + GF_a}{(1 - MIN(|GF_0|, |GF_a|))} \quad \text{Eğer } GF_0 \text{ veya } GF_a < 0 \quad (IV.3)$$

GF_d değeri, öğrenci soruyu doğru yanıtladığında alacağı puan olduğu için, bu değer sorunun o konudaki ağırlığına bağlı olarak değişmektedir. Konunun içerisinde daha büyük öneme sahip sorular için, GF_d değeri diğer sorulara göre daha büyük bir değer olarak belirlenmiştir. GF_y değeri ise, öğrenci soruyu yanlış yanıtladığında konuyu bilmediği yönünde güven değerinde azalma olmasına neden olacak değerdir. Bu değer sistemde her bir soru için $-GF_d$ olarak belirlenmiştir.

Örneğin, herhangi bir konu için sistemde sırasıyla 0.35, 0.4 ve 0.3 GF_d değerlerine sahip üç soru olsun. GF_y değerleri ise, bu değerlerin -1 ile çarpılması ile elde edilmektedir. Öğrenci ilk soruyu doğru yanıtladığında, bu konu için güven değeri Denklem IV.1'deki gibi $0+0.35-0*0.35=0.35$ olacaktır. Öğrenci ikinci soruyu da doğru yanıtladığında yine aynı formül kullanılarak, $0.35+0.4-0.4*0.35=0.61$ değeri bulunacaktır. Fakat, öğrenci üçüncü soruyu yanlış yanıtladığında ise $(0.61-0.3)/(1-\min(0.61,0.3))=0.44$ değeri bulunacaktır.

Eğer öğrenci, tasarımcı tarafından belirlenen güven değerine ulaşamamışsa, o konudan başarısız kabul edilmekte ve sistemde tekrar aynı konu içeriğine yönlendirilmektedir. Öğrenci sistemi kullanmak için her bağlandığında, öğrenci modelindeki güven değerleri kontrol edilerek, hangi konuya yönlendirilmesi gerektiğine karar verilmektedir.

Sistemdeki herhangi bir öğrenci için her bir konu ile ilgili, konuyu bilip bilmediği ve konuyu hangi düzeyde bildiği ile ilgili ölçüm yapılmaktadır. Güven değerinin, -1'e yaklaşması, öğrencinin o konu için bilgi düzeyindeki eksikliği göstermektedir. Sistem öğrencinin konuyu hangi düzeyde bildiğini gösterdiği gibi, bilgi eksikliğinin de düzeyini göstermektedir. Herhangi bir konudaki güven değerinin matematiksel değeri, Tablo IV.6'da belirtilen anlamlara karşılık gelmektedir:

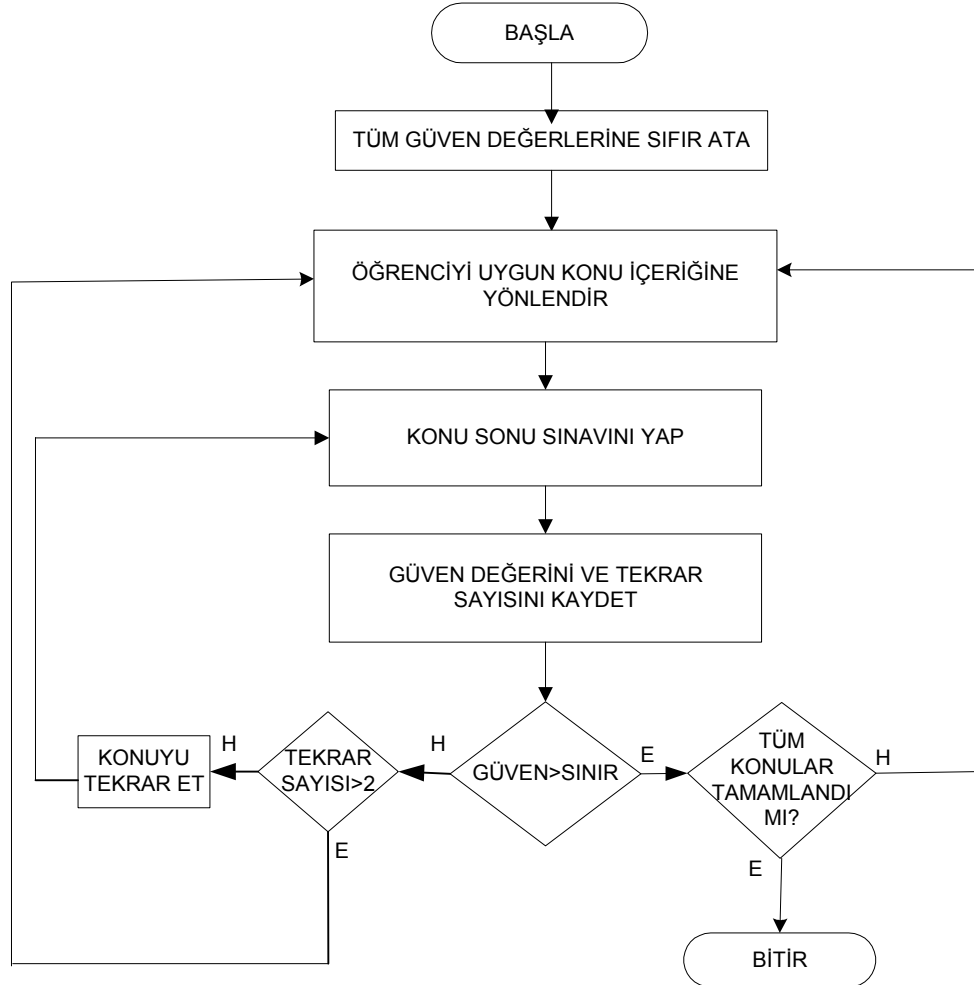
Tablo IV.6 Güven Değeri ve Anlamları

Anlam	Değer
Kesinlikle bilmiyor	$-1 \leq \text{Değer} \leq -0,8$
Çok büyük olasılıkla bilmiyor	$-0,8 < \text{Değer} \leq -0,6$
Muhtemelen bilmiyor	$-0,6 < \text{Değer} \leq -0,4$
Bilmiyor olabilir	$-0,4 < \text{Değer} \leq -0,2$
Bilgi yok	$-0,2 < \text{Değer} < 0,2$
Biliyor olabilir	$0,2 \leq \text{Değer} < 0,4$
Muhtemelen biliyor	$0,4 \leq \text{Değer} < 0,6$
Çok büyük ihtimalle biliyor	$0,6 \leq \text{Değer} < 0,8$
Kesinlikle biliyor	$0,8 \leq \text{Değer} \leq 1$

Sistemde, her konu anlatımının sonunda yer alan sorulardan öğrencinin başarılı kabul edilebilmesi için, alması gereken güven değeri belirlenmiştir. Eğer öğrenci, o değerden daha düşük bir güven değerine sahipse, sistemde o konuyu tekrar

etmesi gerekmektedir. Konuyu kaç kez tekrar edeceği ise, yine konuya göre değişebilen ve tasarımcı tarafından belirlenen bir değerdir.

Öğretim modelinde Şekil IV.5’de belirtilen akış şemasındaki sıra izlenerek işlemler gerçekleştirilir. Öğrenci sisteme ilk defa bağlanıyorsa, tüm konular için güven değeri sıfır olarak atanmaktadır . Daha sonra ise, konu içeriğinin sonunda yer alan sınavdaki yanıtlarına göre Denklem IV.1, IV.2 ve IV.3’den yararlanılarak konuya ait güven değeri hesaplanmaktadır.

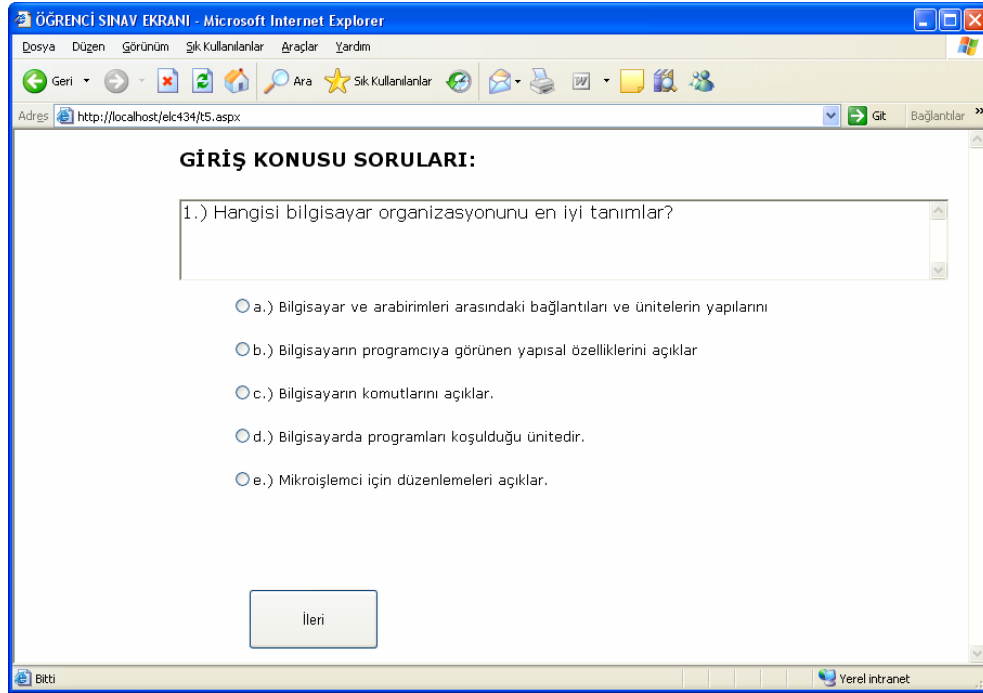


Şekil IV.5 Öğretim modelinin çalışmasına ait akış şeması

Güven değeri ve tekrar sayısı değerleri veritabanına kaydedilmektedir. Bu güven değeri, belirlenen sınır değerinden büyük veya eşitse öğrenci bir sonraki konuya geçmektedir. Alınan güven değeri, sınır değerinden daha küçükse öğrenci konuyu tekrar etmektedir. Tekrar durumunun kaç kez yapılacağı tasarımcı tarafından belirlenir. Tasarlanan sistemde, bir öğrenci, herhangi bir konuyu en fazla iki kez tekrar edebilmektedir. İki kez tekrarın sonunda, öğrencinin güven değeri kaydedilerek bir sonraki konuya yönlendirilmektedir.

IV.5. KULLANICI ARABİRİMİ

ZÖS’deki arabirim tasarımı Microsoft ASP.NET örün programlama teknolojileri ile gerçekleştirilmiştir. Sisteme kayıt olan öğrenci, kullanıcı adı ve şifresi ile giriş yaptıktan sonra, alan modelindeki konu sıralamasına uygun şekilde sistemdeki konuları takip edip, Şekil IV.6’da görüldüğü gibi, konu sonu sınavına ait arabirim aracılığı ile soruları yanıtlar. Öğrenci konuyla ilgili soruya yanıt vermek için, seçeneklerin solunda yer alan radyo düğmelerini kullanmaktadır. “İleri” komut düğmesi ise sonraki sorunun ekrana gelmesini sağlamaktadır. Öğrencinin soruya odaklanmasını sağlamak ve bellek yükünü artırmamak için, sadece bir sorunun ekranda görüntülenmesi sağlanmıştır. Sorular bittikten sonra ise, öğrencinin konudan başarılı olup olmadığına dair bilgi verilir ve öğrenci uygun konu içeriğine yönlendirilir.



Şekil IV.6 Kullanıcı arabirimi örnek ekranı

IV.5.1. Kullanıcı Arabirimi Tasarımında Öğretim Açısından Dikkat Edilmesi Gereken İlkeler

İçerisinde öğretilmek istenen konu alanı bilgisinin sunumunu içeren ve öğrenciyle iletişim kuran, kullanıcı arabiriminde, öğretim kalitesini ve zeki öğretim

sisteminin etkinliğini artırmak için aşağıda yer alan temel tasarım ilkelerine uyulmuştur.

Satır Uzunluğu: Schwier & Misanchuk’a göre öğretim yapmayı amaçlayan ürün sayfalarında satır uzunluğu 35-75 karakter arasında olmalıdır [119].

Metin Boyutu ve Hizalama: Öğretim amaçlı tasarımlarda, başlık dışındaki yazılar için 12-14 punto büyüklüğündeki boyutlar önerilmiştir. Başlıklar için ise, gövde metni yazı boyutu 12-14 arası olarak varsayıldığında, 18-16 punto daha uygun olmaktadır [120].

Öğretim materyallerinin birçoğunda her iki yönden hizalama kullanılmasına rağmen, araştırmalar sola doğru hizalanan metinlerin daha kolay okunduğunu göstermektedir [121]. Her iki yönden hizalama yapıldığında kelimeler veya harfler arasındaki boşluklar artmakta veya kelimelerde kesme kullanmak gerekmektedir. Bu olay da okurken, metni takip etmeyi zorlaştırmaktadır.

Renkler: Materyaller öncelikle gri, siyah veya beyazın tonları şeklinde tasarlanmalı, daha sonra belli eğitim amaçlarına göre renklendirilmelidir. Çok fazla rengin dökümünde kullanılması karmaşaya neden olacağı için yapılan tasarımda da bundan kaçınılmıştır [122].

Eğer renkler doğru bir şekilde kullanılırsa, iyi bir iletişim aracı olmaktadır. Bu amaçla, renklere bazı fonksiyonel amaçlar verilerek daha yapıcı olmalarını sağlamak mümkündür [123].

Önemli kavramlar üzerine dikkat çekilmek istendiğinde parlak renkler kullanılmalıdır. Metin öğrencinin okumasını veya görmesini kolaylaştırıcı zemin rengi uygun zıtlık oluşturacak renklerden seçilmelidir. Fakat, aynı ekranda gereğinden fazla renk kullanımı kullanıcıyı ilgisini azaltmaktadır. Bu nedenle alan tasarımında farklı renkte art alan ve metin rengi kullanımından kaçınılmıştır.

Ekran tasarımlarında, sarı zemin üzerine siyah metin rengi, beyaz zemin üzerine yeşil, kırmızı, siyah ve mavi metin rengi, mavi zemin üzerine beyaz metin rengi ve siyah zemin üzerine sarı metin rengi kolay takip edilebilir bir tasarım oluşturmaktadır [124].

Ruffini’ ye göre beyaz veya açık pastel artalanlar herhangi bir site için kullanmada en iyi renklerdir [125]. Ayrıca, Lauren Scharff tarafından yapılan araştırmaya göre, açık zemin üzerinde koyu renk metnin okunabilirliği daha fazla olmaktadır [126].

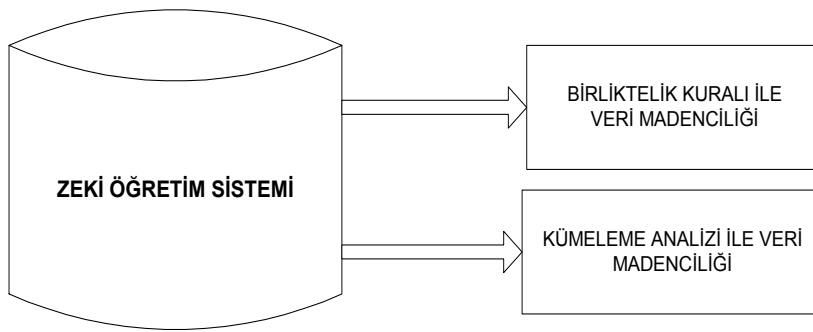
Gerçekleştirilen alan tasarımında beyaz zemin üzerinde koyu mavi ve siyah metin kullanılarak, metinlerin kolay takip edilebilmesi sağlanmıştır.

BÖLÜM V

VERİ MADENCİLİĞİ ANALİZ YAZILIMLARININ TASARIMI

V.1. GİRİŞ

Bu tezde tasarlanan sistemin genel yapısı Şekil V.1’de görülmektedir. Bu sistemde, ZÖS’den alınan verilerde veri madenciliği tekniklerinden, birliktelik kuralı analizi ve kümeleme analizi yapılmaktadır. Bu bölümde; içerisinde öğrencilerin konu sonlarında yer alan sınavlara verdikleri yanıtlar ve konu başarı puanlarını içeren, ZÖS’deki verilerle veri madenciliği analizi yapmak için gerçekleştirilen birliktelik kuralı ve kümeleme analizi yapan yazılımların amacı ve tasarımları açıklanmaktadır.



Şekil V.1 ZÖS ve veri madenciliği

V.2. VERİ MADENCİLİĞİNİ BİRLİKTELİK KURALI ANALİZİ İLE YAPAN YAZILIMLARIN TASARIMLARI

V.2.1. Amaç

Bir önceki bölümde açıklanan ZÖS’den alınan öğrenci yanıtlarından oluşan veriye, birliktelik kuralı analizinin uygulanması için tasarlanan programların amaçları ve işlevleri şunlardır:

1. ZÖS’de her konunun içerisinde yer alan sorulardan, herhangi birisinden başarısız olan öğrencilerin başka hangi sorularda başarısız olduğuna ilişkin bilginin elde edilmesi, o konuda birbirini etkileyen alt kavramların öğretmen tarafından fark edilmesi açısından önemlidir. Aynı şekilde, bir konunun içerisinde yer alan sorularda, herhangi bir soruda başarılı olan öğrencilerin, başka hangi sorularda başarılı olduğunu gösterecek bir analiz de, birbirini destekleyici nitelikteki alt kavramların ortaya çıkarılmasını sağlar. Bu amaçla, öğrenci yanıtları üzerinde birliktelik kuralı analizi yapan ve sonuçta sık tekrar eden doğru yanıt dizilerini veya yanlış yanıt dizilerini bulan yazılımın tasarımı gerçekleştirilmiştir.

2. ZÖS’de öğrencinin konu sonu sınavından aldığı başarı puanının, o konudan alınması gereken minimum puan ile karşılaştırılması sonucunda öğrenci, konudan başarılı veya başarısız olmaktadır. Bu bilgi kullanılarak, bir konudan başarısız olan öğrencilerin başka hangi konulardan başarısız olduğunun veya bir konudan başarılı olan öğrencilerin başka hangi konulardan başarılı olduğunun belirlenmesi, konuların birbirini etkileme durumunun ortaya çıkarılması bakımından önemlidir. Tasarlanan program ile, öğrencilerin birlikte başarısız olduğu veya başarılı olduğu konu dizilerine ait birliktelik kuralları bulunmaktadır. Öğretmen tarafından bu kurallar kullanılarak, hangi konuya daha fazla önem vermesi gerektiği belirlenir.

3. ZÖS’de bulunan herhangi bir konu içerisindeki soruların, farklı konularda yer alan soruları başarı veya başarısızlık açısından nasıl etkilediğinin belirlenmesi, alt kavramlar düzeyindeki ilişkileri ortaya çıkarmak açısından önemlidir. Bu amaçla, herhangi bir konuda yer alan bir soruyu yanlış yapan öğrencilerin, başka konularda yer alan hangi sorularda da başarısız olduklarının veya herhangi bir konuda yer alan soruyu doğru yanıtlayan öğrencilerin, başka konularda yer alan hangi soruları da doğru yanıtladığının analizinin yapılması ile öğretmene yardımcı olunur. Bu analiz;

konular arası sorulara ilişkin çok boyutlu birliktelik kurallarını bulan ve tasarımı bu tezde gerçekleştirilmiş yazılım ile bulunmaktadır.

Yukarıda tanımlanan sorunları çözmek için veri madenciliği tekniklerinden birliktelik kuralı analizi kullanılmaktadır. Gerçekleştirilen yazılımlar ve ZÖS'den alınan veriseti aşağıda kısaca açıklanmaktadır. Ayrıntılı açıklamalar Bölüm V.2.2'den itibaren verilmektedir.

Geliştirilen yazılımlar veri seti olarak, öğrencilerin her konunun sonunda yer alan çoktan seçmeli sorulara verdikleri yanıtları ve öğrencilerin konu sonunda aldıkları puana göre başarılı olup olmadıkları bilgisini kullanmaktadır. Kullanılan tüm veriler boolean tipindedir.

Şekil V.2'de görüldüğü gibi, ZÖS'de bulunan veriler üzerinde veri madenciliği işlemini tek boyutlu birliktelik kuralı analizi ile yapan yazılım (ZEBİKA), kullanılan veri yapısına bağlı olarak iki farklı yapıda birliktelik kuralı bulmakta ve tek boyutlu birliktelik kuralı analizi yapmaktadır. Bulunan birliktelik kurallarından birincisinde, öğrencilerin her konunun sonunda yer alan çoktan seçmeli sorulara verdikleri yanıtların doğru veya yanlış olma durumu kullanılmaktadır. Bu veri seti ile, birlikte yanlış yapılan veya birlikte doğru yapılan sorulara ait birliktelik kuralları bulunmaktadır.

Bu programda konular arasındaki birliktelik kurallarının bulunması için oluşturulan diğer bir veri setinde ise, öğrencilerin ZÖS'de, her konunun sonunda aldığı başarı puanı ile, alınması gereken minimum başarı puanı karşılaştırılmaktadır. Eğer öğrenci, minimum başarı puanından daha büyük veya eşit not almışsa başarılı, daha küçük bir not almışsa başarısız olarak kabul edilir. Konudan başarılı olma durumu 1 ile, başarısız olma durumu 0 ile temsil edilen bu veri seti, ZEBİKA adı verilen programda kullanılarak, öğrencilerin birlikte başarılı veya başarısız oldukları konu dizileri bulunur.



Şekil V.2 Gerçekleştirilen birliktelik kuralı analizi yazılımları ile bulunabilecek birliktelik kurallarının genel yapısı

Zeki öğretim sisteminden alınan verilerden, veri madenciliği işlemini çok boyutlu birliktelik kuralı analizi ile yapan yazılım (ZECOKA) ise, farklı konuların içerisinde yer alan sorular arasındaki ilişkilerin bulunması için çok boyutlu birliktelik kuralı analizi yapmaktadır. ZECOKA ile farklı konular içerisinde yer alan alt kavramlara ait sorularla ilgili çok boyutlu birliktelik kuralları bulunarak, bu soruların birbirlerini başarı veya başarısızlık açısından etkileme durumu elde edilmektedir.

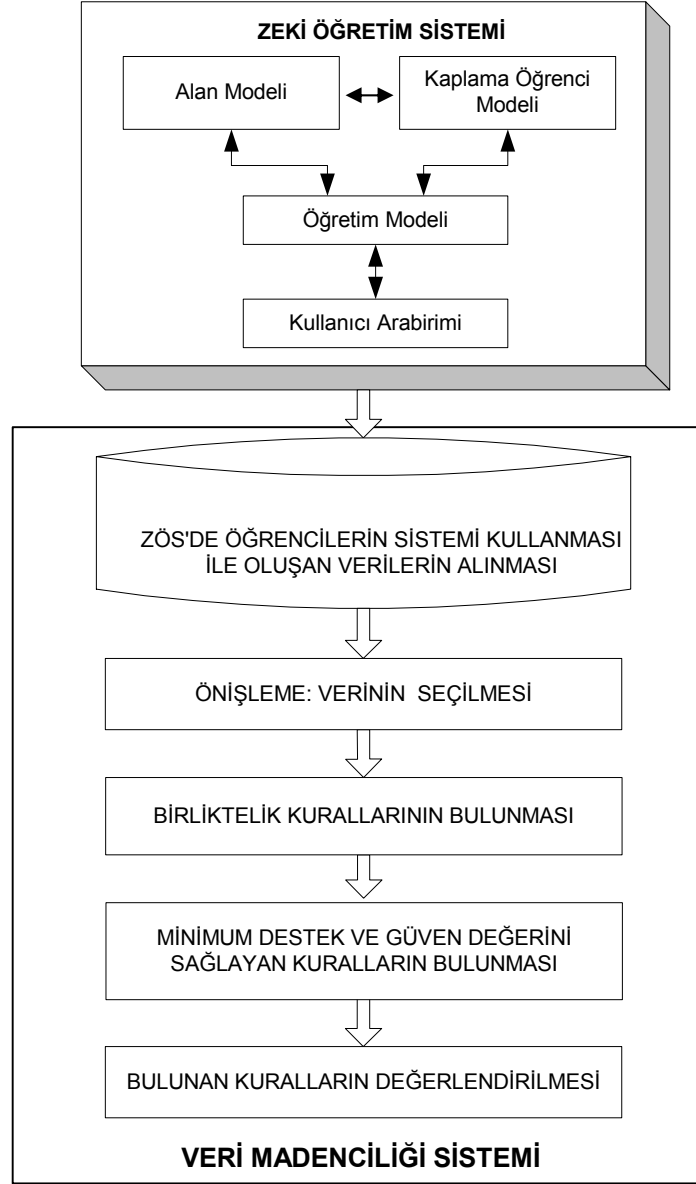
V.2.2. ZÖS ve Birliktelik Kuralı Analizinin Genel Yapısı

ZÖS ve birliktelik kuralı analizi için tasarlanan sistemin genel yapısı ve işlem süreci Şekil V.3’de görülmektedir. Bölüm IV’de açıklandığı gibi, tasarlanan zeki öğretim sistemi dört bileşeni içermektedir. Bunlar; öğrenciye sunulan bilgiyi içeren alan modeli, öğrenciyle ilgili bilgileri tutan öğrenci modeli, hangi pedagojik aktivitenin yapılacağına karar veren öğretim modeli ve öğrenci ile sistem arasındaki etkileşimi kontrol eden kullanıcı arabirimidir.

Tasarlanan zeki öğretim sisteminin öğrenci modelinden alınan veriler ile birliktelik kuralı analizinin yapılabilmesi için şu aşamalar gerekmektedir.

Zeki Öğretim Sisteminin Oluşturulması: Tasarımı, Bölüm IV’de açıklanan ve alan modelinde konu alanıyla ilgili bilgileri, öğretim modelini, öğrenci modelini ve kullanıcı arabirimini içeren ZÖS oluşturulmuştur.

ZÖS’ün Kullanılması: Öğrenciler ürün tarayıcısı aracılığı ile ZÖS’deki alan içeriğini takip etmektedir. Konular içerisinde yer alan sorulara verdikleri yanıtlar ve konulardan aldıkları puanlar sistemde yer alan veritabanına kayıt edilmektedir.



Şekil V.3 ZÖS ve birliktelik kuralı analizinin genel yapısı ve işlem süreci

Önişleme Süreci: ZÖS’den alınan veriler üstünde önişleme süreci ile veri madenciliğinin ilk aşaması gerçekleştirilir. Sistemde, öğrencilerin konu sonlarında yer alan çoktan seçmeli sorulara verdikleri yanıtlar, Microsoft Access veritabanında tutulmaktadır. Öğrenci yanıtlarının yer aldığı Bölüm IV.3’de açıklanan

“TblCevaplar” tablosundan veriler alınmaktadır. Sistemde, öğrenciler bir konuyu en çok iki kez tekrar etmektedirler. Dolayısıyla öğrenci konu sonu sınavını en çok iki kez yanıtlar. Analiz yapmak için, eğer öğrenci konuyu tekrar etmiş ise, verdiği en son yanıtın kullanılması amacıyla TblCevaplar tablosu kontrol edilerek, en son öğrenci yanıtlarını içeren ayrı bir tablo oluşturulmuştur. Bu tablodaki veriler ise, metin dosyasına dönüştürülerek birliktelik analizi için hazır hale getirilmiştir.

Birliktelik Kurallarının Bulunması: Öğrencilerin sistemi kullanması sonucu oluşan veriler arasındaki önemli ilişkiler, birliktelik kuralı analizi ile bulunmaktadır. Bu ilişkiler, öğrencilerin başarısında veya başarısızlığında etkili olan konular arasındaki ilişkiler, aynı konu içerisindeki sorular arasındaki ilişkiler ve farklı konular içerisinde yer alan sorular arasındaki ilişkilerden oluşmaktadır. Öğretmen, birliktelik kurallarını bulan, veri madenciliği yazılımlarını kullanarak bu ilişkileri bulmaktadır. Burada geliştirilen birliktelik kuralı analiz programları, Bölüm III.3’de açıklanan minimum destek ve güven değerlerine göre, Apriori algoritmasını kullanarak birliktelik kurallarını bulmaktadır.

Kural Sonuçlarının Değerlendirilmesi: Zeki öğretim sisteminin performansının artırılması için, birliktelik kuralı analizi sonucu bulunan kurallar öğretmen tarafından kullanılır. Bulunan kurallar incelenerek, bu kurallardan yüksek destek ve güven değerine sahip olanlar uzman tarafından belirlenerek, sistemde gerekli değişiklikler yapılması için geri bildirim bilgisi olarak kullanılır. Örneğin, bulunan birliktelik kurallarından elde edilen bilgi ile, konular arasındaki önşart ilişkisi ve konu tekrarı sayısı yeniden düzenlenerek, ZÖS’ün güncellenmesinde yararlanır.

V.2.3. Veri Madenciliğinde Birliktelik Kuralı Analizi için Kullanılan Apriori Algoritması

Literatürde birliktelik kurallarını bulmak için en çok kullanılan algoritma, Apriori algoritmasıdır [81,127,128]. Birliktelik kuralı bulmak için bölümlleme [129] ve örnekleme [130] algoritmaları da kullanılmaktadır. Fakat özellikle destek değeri %20 ve üzeri olarak belirlendiğinde Apriori daha iyi sonuçlar vermektedir [131]. Bu nedenle, bu çalışmada birliktelik kuralı bulma işlemini gerçekleştirmek için Apriori algoritması esas alınmıştır.

Apriori algoritması tekrarlı çalışan bir algoritmadır. Algoritma, sık tekrar eden eleman kümelerini oluşturabilmek için Bölüm III.3’de açıklanan minimum destek değerini ölçüt olarak kullanmaktadır [81-83]. Apriori algoritmasında, öncelikle verilen eleman kümesi içerisinde, her elemanın kaç kez tekrar edildiği hesaplanır ve tekrar değeri minimum destek değerine eşit olan veya bu değeri aşan elemanlar seçilerek sık tekrarlanan eleman kümesi bulunur. Bu seçilen elemanların tümünden oluşan ilk kümeye, sık tekrarlanan eleman kümesi 1 adı verilir. Sık tekrarlanan eleman kümesi 1’i temsil etmek için L1 simgesi kullanılmaktadır. L1 simgesinde, L harfi, küme elemanlarının sık tekrar edildiğini, 1 değeri ise kümedeki her bir elemanın uzunluğunu belirtmek için kullanılmaktadır. Algoritmada, L1 kümesi kullanılarak, her elemanın uzunluğu iki olan L2 kümesi bulunmaktadır. Bu işlem için öncelikle, L1 kümesinde birleştirme işlemi yapılmaktadır. Birleştirme işlemi, $L1 \times L1$ ile gösterilmektedir ve birleştirme sonucunda oluşan küme, L2 olarak adlandırılan sık tekrarlanan eleman kümesi 2’nin bulunması için kullanılmaktadır. Aynı şekilde, sık tekrarlanan eleman kümesi 2 (L2) de, L3’ün bulunmasında kullanılmaktadır. Bu işlem daha fazla elemandan oluşan küme bulunamayınca kadar devam eder.

k pozitif bir tamsayıyı ifade etmek koşuluyla, sık tekrar edilen (k-1) uzunlukta elemanlara sahip kümeyi ifade eden L_{k-1} ’den L_k ’nın nasıl oluşturulduğu aşağıda açıklanmaktadır. Bunun için iki aşamalı bir çalışma yapılır. Bu aşamalar, birleştirme (join) ve budama (prune) işlemleridir.

Birleştirme İşlemi: L_{k-1} ’yi bulabilmek için, L_{k-1} ’in kendisi üzerine birleştirilmesi ile k eleman uzunluğuna sahip aday küme oluşturulmaktadır. Aday küme, C_k simgesi ile gösterilir. $L_{k-1} \times L_{k-1}$ birleştirme işleminin yapılabilmesi için, bu kümedeki üyelerin ilk (k-2) elemanlarının ortak olması gerekmektedir. Birleştirme işleminde ilk (k-2) elemanı eşit olan elemanlar kullanılarak, k uzunluğa sahip elemanları içeren aday eleman kümesi C_k bulunmaktadır.

Budama İşlemi: C_k simgesi, k uzunluğa sahip aday kümeyi temsil etmektedir. C_k ’da yer alan herhangi bir elemanın sık tekrarlanan eleman olup olmadığı, yani tekrar değerlerinin minimum destek değerine eşit veya büyük olma durumunun incelenmesi gerekmektedir. Tekrar değeri minimum destek değerinden küçük olan elemanlar atılarak, sık tekrarlanan eleman kümesi L_k bulunur. Apriori algoritması, her elemanın (k-1) uzunluğa sahip alt kümelerini oluşturup, bu alt kümelerin sık tekrar eden elemanlar olup olmadığını kontrol eder. Herhangi bir alt kümesinin tekrar

değeri minimum destek değerinden küçük olan eleman, Ck'dan atılmaktadır. Ck'da kalan elemanların ise, tüm veri seti taranarak tekrar değerleri bulunmakta ve minimum destek değerini aşamayan küme elemanlarının da Ck'dan atılması sağlanmaktadır. Sonuçta, tüm (k-1) uzunluğa sahip alt kümeleri sık tekrar edilen ve tüm elemanlarının tekrar değerleri minimum destek değerini sağlayan Lk kümesi oluşturulmaktadır [61, 81-83].

Şekil V.4'de Apriori algoritmasının sözde kodu görülmektedir [61, 82].

Ana Program Apriori

```
L1 = {sık tekrarlanan eleman kümesi 1}
SAYARAK YİNELE ( k:=2, Lk-1 ≠ ∅ , k:=k+1 )
  /* k uzunluğa sahip elemanlardan oluşan aday kümelerin bulunması */
  Ck = Aday_olustur(Lk-1, min_destek)
  Tüm işlemler için
  [Ck'daki tüm elemanların tekrar değerlerini hesapla]
  Lk = {Ck'da minimum destek değerini aşan elemanlar}
]
```

L=Uk Lk DÖNDÜR

Alt program Aday olustur (Lk-1, min_destek)

```
SAYARAK YİNELE Her I1 ∈ Lk-1 için
  SAYARAK YİNELE Her I2 ∈ Lk-1 için
    EĞER (I1[1]=I2[1]) /\ ( I1[2]=I2[2]) /\ ... /\ ( I1[k-2]=I2[k-2]) /\ (I1[k-1] < I2[k-1])
    [C=I1>I2 /* aday kümeler için birleştirme işlemi
    EĞER sık olmayan alt kümeye sahip mi (c, Lk-1)
    C'yi sil
    DİĞER DURUMLAR İÇİN
    c'yi Ck'ya ekle]
  Ck'yı DÖNDÜR
```

Alt Program sık olmayan alt kümeye sahip mi(c, Lk-1)

```
SAYARAK YİNELE c'nin (k-1) alt kümeleri s için
  Eğer s ∉ Lk-1 ise
    Doğru değeri DÖNDÜR
  DİĞER DURUMLAR İÇİN
    Yanlış değeri DÖNDÜR
```

Şekil V.4 Apriori Algoritmasına ait sözde kod

Apriori algoritmasının ilk aşamasında, sık tekrarlanan eleman kümesi 1 olarak adlandırılan L1 kümesi bulunmaktadır. Sık tekrarlan eleman kümesi 1'in bulunması için, tüm veri seti taranarak her elemanın tekrar değeri bulunur ve tekrar değeri minimum destek değerini aşamayan elemanlar kümeden budanır. Bundan sonraki aşamalarda, bir döngü içerisinde, döngü $k=2$ 'den başlamak üzere, k uzunluğa sahip aday kümeyi temsil eden C_k 'yı bulabilmek için **Aday_olustur** alt programında Lk-1 kullanılır. Aday_olustur alt programı ile, aday kümeler oluşturulmakta ve apriori algoritmasının sahip olduğu özellik ile de, alt kümesi sık tekrar edilmeyenler elenmektedir. Aday_olustur alt programı temelde iki görevi yerine getirmektedir. Bunlar, birleştirme ve budama işlemleridir. Lk-1 ile Lk-1 birleştirilerek potansiyel aday küme oluşturulmaktadır. Birleştirme işleminde ilk $(k-2)$ elemanı eşit olan elemanların birleştirilmesi için kontrol yapılması gerekmektedir. Kontrol işleminde, I1 ve I2, Lk-1'in herhangi bir elemanını temsil etmek şartıyla her Lk-1 elemanı ikili döngü içerisinde kontrol edilerek I1 ve I2 elemanlarından sadece ilk $(k-2)$ elemanı eşit olanlar birleştirilmektedir. Daha sonra ise, aday kümeler içinden alt kümesi sık tekrar edilmeyenler atılarak budama işlemi yapılmaktadır. Bu işlemi, **sık_olmayan_alt_küme_sahip_mi** alt programı gerçekleştirmektedir.

Alt küme kontrolü yapılarak, k uzunluğa sahip elemanları içeren C_k aday kümesi oluşturulduktan sonra, tüm veriseti taranarak C_k kümesindeki tüm elemanların tekrar sayıları hesaplanmaktadır. Bu aday küme içinden minimum destek değerini sağlayanlar, sık tekrarlanan eleman kümesi Lk'yı oluşturmaktadır. Ana program içerisinde, yukarıda açıklanan Aday_olustur alt programını çağıran ve alt küme kontrolünü yapan döngü, Lk-1 kümesi boş kümeye eşit oluncaya kadar devam etmektedir.

Apriori algoritmasının, gerçekleştirilen yazılımlardaki çalışmasına bir örnek olması amacıyla, zeki öğretim sisteminin öğrenciler tarafından kullanılması ile elde edilen örnek veri kümesi Tablo V.1'de görülmektedir. Verinin boyutu kaç öğrencinin sistemi kullandığı ve konuda kaç sorunun bulunduğuyla ilgili olarak değişmektedir. Tablo V.1'de sekiz öğrencinin, yedi sorudan oluşan bir konu sonu sınavına verdikleri yanıtlar görülmektedir. Bu yanıtlar, Access veritabanından alınmakta ve ogr() dizisi içerisinde saklanmaktadır. Ogr dizisindeki 1 değeri sorunun doğru yanıtlandığını, 0 değeri ise sorunun yanlış yanıtlandığını belirtmektedir.

Tablo V.1'deki gibi değerlere sahip olan veri kümesinde, örneğin birinci öğrencinin yanıtları ogr(1) dizisinde tutulmaktadır ve bu öğrencinin; dördüncü ve altıncı sorulara yanlış yanıt verdiği $ogr(1) = "1110101"$ verisinden anlaşılmaktadır.

Tablo V.1 Öğrenci yanıtlarını içeren dizi yapısı

Soru No \ Dizi Adı	1	2	3	4	5	6	7	Yanıt Dizisi
ogr(1)	1	1	1	0	1	0	1	"1110101"
ogr(2)	0	1	1	1	0	1	0	"0111010"
ogr(3)	0	1	1	0	0	0	1	"0110001"
ogr(4)	1	1	1	0	0	0	1	"1110001"
ogr(5)	1	0	1	0	0	1	1	"1010011"
ogr(6)	0	1	1	0	0	1	0	"0110010"
ogr(7)	1	0	1	0	0	1	1	"1010011"
ogr(8)	1	1	1	0	1	1	1	"1110111"

Tablo V.1'de yer alan verilerin Apriori algoritmasında kullanmak üzere, işlemsel veri şekline dönüştürülmesi gerekir. Tablo V.2'de, oluşturulan işlemsel veri görülmektedir. Tablo V.2'de, her öğrenciye T1'den başlayarak T8'e kadar işlem numarası verilmiştir. Elemanlar dizisi sütununda ise, öğrencinin yanlış yaptığı soruları temsil etmek için S harfinin yanında, yanlış yapılan soru numarasını belirtilen rakam kullanılmaktadır. Örneğin, ogr(1) dizisi, T1 işlem numarası ile temsil edilmiştir. Bu öğrencinin dördüncü ve altıncı sorulara yanlış yanıt verdiğini ifade etmek için ise S4 ve S6 simgeleri kullanılmıştır. Diğer dizi elemanları da benzer şekilde Tablo V.2'de işlem numarası ve elemanlar dizisi ile gösterilmektedir. Bu tablodaki veriler, Apriori algoritmasında kullanılarak, en çok hata yapılan soru dizilerinin bulunması amaçlanmaktadır.

Tablo V.2 Apriori algoritmasında kullanılmak üzere oluşturulan işlemsel veri

İşlem No	Elemanlar Dizisi
T1	S4, S6
T2	S1, S5, S7
T3	S1, S4, S5, S6
T4	S4, S5, S6
T5	S2, S4, S5
T6	S1, S4, S5, S7
T7	S2, S4, S5
T8	S4

Bu örnekte, minimum destek değeri 2 kabul edilerek, sık tekrarlanan eleman kümeleri bulunacaktır. Minimum destek değeri aynı zamanda $2/8=25\%$ değerine karşılık gelmektedir. Algoritmanın çalışma adımları Tablo V.3’de gösterilmektedir

Tablo V.3 Destek Değeri=2 ile aday küme ve sık tekrarlanan kümelerin oluşturulması

Adım 1: C1 ve L1 kümelerinin oluşturulması

<i>Her aday için tüm veri seti taranır ve tekrar değerleri bulunur</i>		<i>Her adayın tekrar değeri minimum destek değeri ile karşılaştırılır</i>	
C1 kümesi	Tekrar Değeri	L1 kümesi	Tekrar Değeri
S1	3	S1	3
S2	2	S2	2
S3	0	S4	7
S4	7	S5	6
S5	6	S6	3
S6	3	S7	2
S7	2		

Adım 2: C2 ve L2 kümelerinin oluşturulması

<i>L1 kümesi birleştirilerek C2 kümesi oluşturulur</i>	<i>Her adayın kaç kez tekrarlandığını bulmak için tüm veri seti taranır</i>		<i>Her adayın tekrar değeri minimum destek değeri ile karşılaştırılır</i>	
C2 kümesi	C2 kümesi	Tekrar Değeri	L2 kümesi	Tekrar Değeri
S1 S2	S1 S2	0	S1 S4	2
S1 S4	S1 S4	2	S1 S5	3
S1 S5	S1 S5	3	S1 S7	2
S1 S6	S1 S6	1	S2 S4	2
S1 S7	S1 S7	2	S2 S5	2
S2 S4	S2 S4	2	S4 S5	5
S2 S5	S2 S5	2	S4 S6	3
S2 S6	S2 S6	0	S5 S6	2
S2 S7	S2 S7	0	S5 S7	2
S4 S5	S4 S5	5		
S4 S6	S4 S6	3		
S4 S7	S4 S7	1		
S5 S6	S5 S6	2		
S5 S7	S5 S7	2		
S6 S7	S6 S7	0		

Tablo V.3 (Devam) Destek Değeri=2 ile aday küme ve sık tekrarlanan kümelerin oluşturulması

Adım 3: C3 ve L3 kümelerinin oluşturulması

<i>L2 kümesi birleştirilerek C3 kümesi oluşturulur</i>	<i>Her adayın kaç kez tekrarlandığını bulmak için tüm veri seti taranır</i>		<i>Her adayın destek değerini minimum destek değeri ile karşılaştırılır</i>	
C3 kümesi	C3 kümesi	Tekrar Değeri	L3 kümesi	Tekrar Değeri
S1S4S5	S1S4S5	2	S1S4S5	2
S1S5S7	S1S5S7	2	S1S5S7	2
S2S4S5	S2S4S5	2	S2S4S5	2
S4S5S6	S4S5S6	2	S4S5S6	2

Tablo V.2’de yer alan veri seti üzerinde Apriori algoritmasının çalışması, Tablo V.3’de görülen aday küme ve sık tekrarlanan kümeler oluşturularak, aşağıda açıklandığı gibi gerçekleştirilmektedir.

1. Algoritmanın ilk adımında L1, sık tekrarlanan elemanlar 1 kümesinin bulunması gerekmektedir. Bunun için, veri setinde S1(soru 1)’den S7(soru 7)’ye kadar tüm elemanları içeren C1 kümesindeki elemanların tekrar değerleri bulunur. L1 kümesi ise, sadece C1 kümesinde minimum destek değerine eşit veya büyük tekrar değerine sahip elemanlardan oluşmaktadır. Tablo V.3’de Adım 1’de görüldüğü gibi bu kural uygulanarak L1 kümesinde 0 tekrar değerine sahip, S3 sorusu budanmıştır.
2. İkinci adımda, Sık tekrarlanan elemanlar 2 kümesini bulabilmek için $L1 \times L1$ birleştirmesinin yapılması gerekmektedir. Adım 2’de L1 kümesi birleştirilerek C2 kümesi oluşturulur. Daha sonra ise, tüm veri seti taranarak, C2 kümesinde yer alan her elemanın tekrar değeri hesaplanır. L2 kümesinin bulunması için ise, C2 kümesinde minimum destek değerinden küçük tekrar değerine sahip elemanların kümeden budanması sağlanmıştır. Sonuçta dokuz elemana sahip L2 kümesi oluşturulmuştur. Oluşturulan L2 kümesinin elemanları ve tekrar değerleri Tablo V.3’de Adım 2’de görülmektedir.
3. Üçüncü adımda ise, $C3 = L2 \times L2$ birleştirme işlemi yapılır. Birleştirme işleminde Apriori algoritmasına göre L_{k-1} ’in bulunması sırasında, herhangi iki küme arasında birleştirme yapılabilmesi için ilk $(k-2)$ elemanları ortak olması gerekmektedir. Bu adımda $k-1=2$ ve $k=3$ olduğundan, kontrol

edilmesi gereken ilk $k-2$ eleman değeri $3-2=1$ olmaktadır. Başka bir ifadeyle, L_2 kümesinde herhangi iki elemanın birleştirilebilmesi için, ilk elemanlarının aynı olması gerekmektedir. Bu kurala göre birleştirme yapıldığında küme elemanları $\{S1S4S5, S1S4S7, S1S5S7, S2S4S5, S4S5S6, S5S6S7\}$ olmaktadır. Apriori algoritmasına göre, k uzunlukta elemanlara sahip bir aday kümenin sık tekrar eden eleman olabilmesi için, kümede, $(k-1)$ elemanlı alt kümelerinin de sık tekrar etmesi gerekmektedir. Bu aşamada küme elemanlarının uzunluğu $k=3$ olmakta ve her elemanın $3-1=2$ elemanlı alt kümelerinin sık tekrarlanan elemanlar olup olmadığının kontrol edilmesi gerekmektedir. Bu kurala göre $\{S1S4S7, S5S6S7\}$ değerleri, alt kümelerinin tümü L_2 kümesinde yer almadığı için C_3 kümesinde yer alamamıştır. Örneğin, $S1S4S7$ 'nin alt kümeleri $\{S1S4, S4S7, S1S7\}$ 'dir. Bu alt küme elemanlarından $S4S7$, L_2 alt kümesinde yer almadığı için, $S1S4S7$ soru dizisi C_3 kümesine girememiştir. Adım 3'de son olarak tüm veri seti taranarak, C_3 kümesinde minimum destek değerini sağlayan elemanlar bulunmuş ve bu elemanlardan L_3 kümesini oluşturmuştur.

4. $L_3 \times L_3$ birleştirme işleminin yapılabilmesi için, bu kümedeki üyelerin ilk 2 elemanlarının ortak olması gerekmektedir. Bu kurala uyan eleman olmadığı için bu işlem yapılamamakta $C_4 = \emptyset$ ve $L_4 = \emptyset$ olmaktadır. Bu durumda, Apriori algoritması eleman uzunluğu üç olan ve dört adet sık tekrar eden elemana sahip L_3 kümesini bularak sonlanmaktadır. Bulunan L_3 kümesi, Tablo V.3'de Adım 3'de görülmektedir.

V.2.4. Kullanılan Veri Yapısı

ZEBİKA adını verdiğimiz, zeki öğretim sisteminden alınan veriler üzerinde veri madenciliği işlemini, birliktelik kuralı analizi ile yapan yazılımın kullandığı veri yapısına ilişkin açıklamalar bu bölümde yer almaktadır.

ZEBİKA ile zeki öğretim sistemindeki herhangi bir konu sonunda yer alan öğrenci verilerine ait başarı veya başarısızlık durumuna ait birliktelik kuralları ve konu başarı durumlarına ait birliktelik kuralları bulunur. Böylece bir konu içerisinde yer alan, daha alt düzeylerdeki kavramların birbirlerini ne şekilde etkiledikleri, veri madenciliği analiz yöntemlerinden birliktelik kuralı ile keşfedilir.

Bu analiz, Bölüm III’de birliktelik kuralı başlığı altında açıklanan destek ve güven değerlerine göre yapılmaktadır. Çıkan sonuçlar içerisinde, yüksek güven değerine sahip olanlar, öğrenci ve öğretmene rehberlik hizmeti sağlamak için kullanılır. Öğrenci açısından, özellikle başarısızlık durumu ile ilgili bulunan birliktelik kuralları içerisinde yer alan bir soruya yanlış yanıt verdiğinde, ileride karşılaşılabilecek olası başarısızlık durumları hakkında bilgilendirme yapılarak, öğretim sistemi içerisinde uygun konu alanına yönlendirilmesi yapılır. Öğretmen ise, çıkan birliktelik kurallarına göre başarısızlık açısından birbirlerini fazla etkileyen konuların anlatımına daha fazla önem verilmesini sağlar.

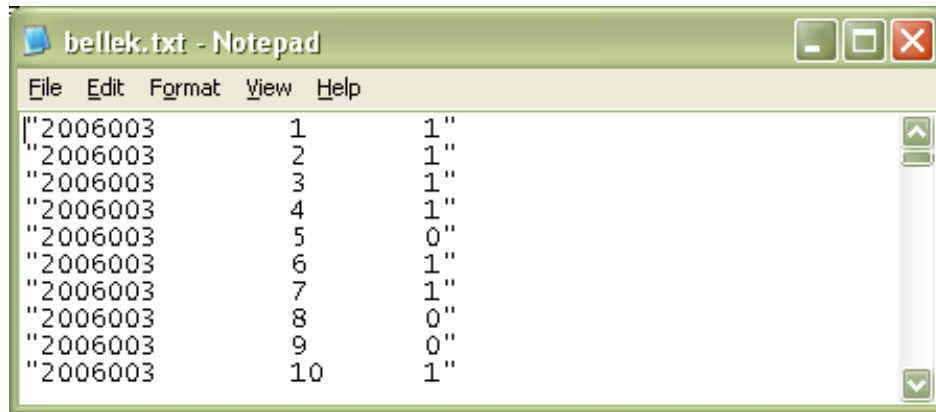
Geliştirilen birliktelik kuralı analiz programı, Microsoft Visual Studio 6.0 içerisinde yer alan Microsoft Visual Basic 6.0 ortamında hazırlanmıştır.

Zeki öğretim sistemi iki öğrenci grubuna uygulanmıştır. Konu alanı “Bilgisayar Sistemleri” dersi olan zeki öğretim sistemi Teknik eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Bilgisayar-Kontrol Eğitimi 8. dönem örgün ve ikinci öğretim öğrencilerine uygulanmıştır. Sistemi; örgün öğretimde 42, ikinci öğretimde 61 olmak üzere toplam 103 öğrenci kullanmıştır. Sistemde her konunun sonunda bir konu sonu sınavı bulunmaktadır. Sistemdeki konular ve içerdiği sorular Tablo V.4.’de gösterilmektedir. Bu soru adetleri; giriş konusunda 7 soru, veriyolu yapıları konusunda 8 soru, bellek yapıları konusunda 10 soru, işlemci yapısı ve fonksiyonları konusunda 9 soru, komut düzeyinde paralellik konusunda 8 soru, komut takımı ve adresleme modları konusunda 11 soru olacak şekilde düzenlenmiştir. Sistemde toplam 53 soru bulunmaktadır.

Tablo V.4 Zeki öğretim sisteminde yer alan konular ve konu sonu sınavındaki soru adetleri

Konu Adı	Soru Sayısı
Giriş	7
Veriyolu Yapıları	8
Bellek Yapıları	10
İşlemci Yapısı ve Fonksiyonları	9
Komut Düzeyinde Paralellik	8
Komut Takımı ve Adresleme Modları	11
TOPLAM SORU SAYISI	53

ZEBİKA yazılımında veri madenciliği analizi için iki farklı veri seti kullanılmaktadır. Kullanılan birinci veri setinde, her konuya ilişkin öğrencilerin sınav sonuçlarının bulunduğu veritabanından öncelikle tüm konulara ait öğrenci yanıtlarının bulunduğu Bölüm IV.3’de açıklanan TblCevaplar süzülerek, her konuya ait yanıtlar, ayrı ayrı veri tablosu halinde tutulmaktadır. Her tablo, metin dosyasına dönüştürülerek veriler analiz için hazır hale getirilmiştir. Hazırlanan metin dosyaları, öğrenci numarası ile birlikte, kural sonucu bulunacak konuyla ilgili soru numarası ve öğrencinin cevabının doğru veya yanlış olduğunu gösteren 1 (bir) veya 0 (sıfır) değerini içermektedir. 0 değeri sorunun yanlış yanıtlandığını, 1 değeri ise doğru yanıtlandığını belirtmektedir. Şekil V.5’de bellek yapıları konusuna ait konu sonu öğrenci yanıtlarını gösteren metin dosyasının içeriğinden bir örnek görülmektedir. Bu örnekte, 2006003 numaralı öğrenci bellek yapıları konusunda 5. 8. ve 9. sorulara yanlış, diğer sorulara doğru yanıt vermiştir. Bu dosyada, sistemi kullanan her öğrencinin benzer yapıdaki yanıtları bulunmaktadır.



File	Edit	Format	View	Help
"2006003	1	1"		
"2006003	2	1"		
"2006003	3	1"		
"2006003	4	1"		
"2006003	5	0"		
"2006003	6	1"		
"2006003	7	1"		
"2006003	8	0"		
"2006003	9	0"		
"2006003	10	1"		

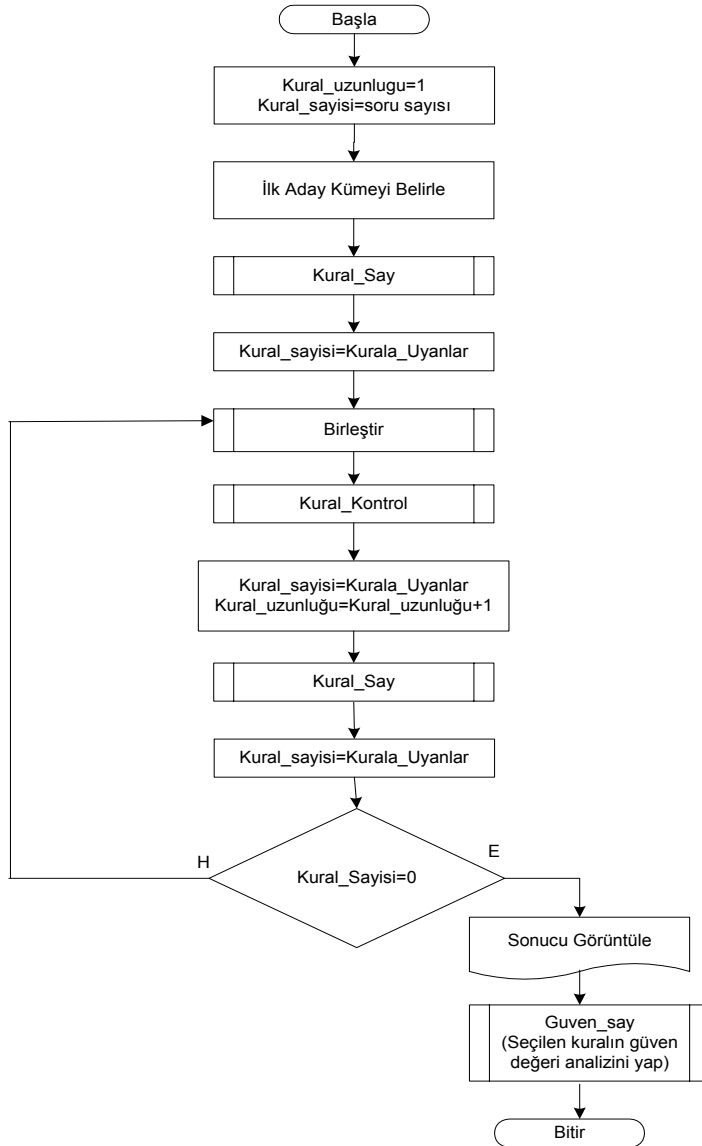
Şekil V.5 Bellek yapıları konu sonu sınavındaki öğrenci yanıtlarını içeren metin dosyası

ZEBİKA programında kullanılmak üzere oluşturulan ikinci veri seti, öğrencilerin konu sonlarından aldıkları başarı puanlarına (MYCIN güven faktörleri) göre, konudan başarılı veya başarısız olduklarını belirten değerleri içermektedir. Bu veri setinde, TblPuan tablosunda bulunan her öğrencinin aldığı puan, konudan alınması gereken minimum başarı değeri ile karşılaştırılır. Karşılaştırma sonucunda, eğer öğrencinin başarı puanı, minimum başarı değerinden büyük veya eşitse öğrenci başarılı olarak kabul edilir. Öğrencinin başarı puanı, minimum başarı değerinden küçükse öğrenci o konudan başarısız olarak kabul edilir. Konudan başarılı olma durumu 1 ile, başarısız olma durumu ise 0 ile temsil edilmiştir. ZÖS’de yer alan toplam altı konu için her öğrencinin başarı puanları yukarıda açıklandığı gibi kontrol

edilerek, her öğrencinin başarı durumunu gösteren veri seti oluşturulur. Bu veri seti, ZEBİKA programında kullanılarak, birlikte başarılı veya başarısız olunan konu dizileri bulunur.

V.2.5. Birliktelik Kuralı Bulan ZEBİKA Yazılımının Tasarımı

Zeki öğretim sisteminden alınan veriler üzerinde veri madenciliği işlemini birliktelik kuralı analizi ile yapan yazılım (ZEBİKA), Apriori algoritmasını esas alarak çalışmakta ve birliktelik kurallarını bulmaktadır. Şekil V.6.'da ZEBİKA yazılımına ait akış şeması görülmektedir.



Şekil V.6 ZEBİKA yazılımına ilişkin tasarımına ait akış şeması

Programda ilk deęer olarak kural uzunluęu bir olarak, kural sayısı ise soru sayısına eřit olarak atanmaktadır. Aday kümedeki elemanların kural_say alt programı ile tüm veri seti taranarak tekrar deęerleri bulunmakta ve minimum destek deęerini aşan deęerler küme içersinde kalmakta, dięerleri atılmaktadır. Kural_kontrol alt programı ise, aday kümedeki her elemanın alt kümelerini oluřturup, bir önceki sık tekrar edenler kümesinde olup olmadıęını kontrol etmektedir. Ancak tüm alt kümeleri, bir önceki sık tekrar edenler kümesinde olan elemanlar kümede kalmakta, dięerleri kümeden atılmaktadır.

Tasarlanan yazılımdaki, Apriori algoritmasının prensiplerini geręekleřtiren alt programlar ařaęıda belirtilmektedir:

1. **Kural_say** (aday_eleman_kümesinin_uzunluęu) : Aday kümenin tüm elemanlarının tekrar sayıları, veritabanında yapılan tarama ile bulunmakta ve sadece minimum destek deęerini aşan küme elemanları aday küme içersinde kalmakta dięerleri atılmaktadır.
2. **Birlestir** (sık_tekrarlan_elemanlar): Kural uzunluęu n olan, sık tekrarlanan eleman kümesinin birleřtirme iřlemini yapmaktadır. Bu alt program, Apriori algoritmasının alıřma prensibine göre birleřtirme iřlemini geręekleřtirmektedir.
3. **Kural_kontrol** (): Birleřtirilmiř yeni aday eleman kümesinin tüm elemanlarının alt kümeleri oluřturulmakta ve bu alt küme elemanlarının, bir önceki iterasyonda bulunan sık tekrar eden eleman kümesinde yer alıp almadıęı belirlenmektedir. Tüm alt kümeleri bir önceki ařamadaki sık eleman kümesinde bulunan adaylar, aday eleman kümesinde yer almakta dięerleri kümeden atılmaktadır.
4. **Guven_say**(): Program sonucunda oluřturulan kurallardan, seilen herhangi bir kurala ait güven deęeri analizini geręekleřtirmektedir. Seilen kurala ait oluřturulabilecek kurallar listelenir ve bu kuralların güven deęerleri hesaplanarak görüntülenir.

Daha sonra ise, kural sayısının yeni deęeri atanmakta ve kural_sayısının deęerinin sıfıra eřit olup olmadıęı kontrol edilmektedir. Eęer kural_sayısı sıfırdan farklı ise, aday kümenin elemanları için birleřtir alt programı ile yeni aday kümeler oluřturulmakta ve kural uzunluęu bir artırılmaktadır Kural sayısı sıfır ise, kural sonuçları ekranda görüntülenmekte ve seilen kurala ait güven deęeri analizi Guven_say alt programı ile hesaplanmaktadır. ZEBİKA yazılımı kural sayısı

sıfırdan farklı oluncaya kadar, yani sık tekrarlanan eleman kümesi boş kümeye eşit oluncaya kadar, Şekil V.6'da akış şemasında belirtildiği gibi tekrarlı bir şekilde çalışmaktadır.

V.2.5.1. ZEBİKA Programının Yüklenme Aşamasında Yapılan İşlemler

ZEBİKA yazılımı ilk çalıştırıldığında, ana formun yüklenme aşamasında, aynı konuda yer alan sorular arasındaki birliktelik kurallarının bulunması için, (Form_Load) öncelikle öğrenci yanıtları ve yanıtın durumunu gösteren verilerin bulunduğu metin dosyası açılmakta ve dosya sonuna kadar okunarak, her öğrenciye ait durum verisi bir dizi içerisinde tutulmaktadır. Örneğin, konu sonu sınavında 8 soru olduğu varsayıldığında, her öğrenci için $ogr(1) = "1100111"$, $ogr(2) = "0011111"$ şeklinde öğrenci cevap durumunu gösteren bir dizi yapısı oluşturulmaktadır. $ogr()$ dizisi öğrenci sayısı kadar ilerlemektedir.

ZEBİKA yazılımı ile, konular arası başarı ilişkilerini bulabilmek için ise, öğrencilerin konu başarı durumlarını gösteren verilerin bulunduğu metin dosyasının okunması gerekmektedir. Zeki öğretim sisteminde altı konu bulunduğu için, her öğrenci için konu başarı durumunu gösteren veri altı elemanlı olacaktır. Örneğin, $ogr(1) = "110110"$ şeklindeki bir dizi, öğrencinin üçüncü ve altıncı konudan başarısız olduğunu göstermektedir. Her öğrenci için benzer yapıda oluşturulan veri setinin, program ilk yüklendiğinde $ogr()$ dizisi içersine alınması gerekmektedir.

Programda ilk aday kümenin oluşturulması için, başlangıç değerlerinin atanması gerekmektedir. İlk değerler, $aday(1,1)=1$, $aday(2,1)=2$ şeklinde veritabanında her öğrenci için kaç adet soru varsa o kadar ilerlemektedir. Örneğin, konuda 8 soru olduğu kabul edildiğinde $aday(8,1)=8$ dizinin son elemanı olacaktır. Konular arası birliktelik kuralı bulunurken, benzer şekilde $aday()$ dizisi ZÖS'de altı konu bulunduğu için, $aday(1,1)=1$ 'den başlayıp $aday(6,1)=6$ değerine kadar ilerleyecektir. İki boyutlu aday dizisindeki, birinci boyuttaki değer dizinin kaçınıcı elemanı olduğu göstermekte, ikinci boyut ise kuralın uzunluğu kadar ilerlemektedir. Program ilk çalıştırıldığında, kuraldaki eleman sayısı bir olduğu için, iki boyutlu aday dizisinin ikinci boyutundaki değer bir olarak atanmaktadır. Son olarak ise, oluşturulan kural sayısının belirlenmesi gerekmektedir. Çünkü program, kural sayısı sıfıra eşit oluncaya kadar çalışmaktadır. Başlangıçta, $kural_sayısı$ sorular arası birliktelik kuralları bulunurken, hangi konuyla ilgili birliktelik kuralı bulunuyorsa o

konuya ait toplam soru sayısına eşit olarak atanmaktadır. Konular arası birliktelik kuralı için, başlangıçtaki kural_sayısı toplam konu sayısı olan altı değerine eşitlenmektedir. Daha sonra kural_say alt programı ile, her aday elemanın tekrar sayısı belirlenmekte ve destek değerini aşamayanlar aday kümeden atılmaktadır.

Programda minimum destek değeri form üzerindeki metin kutusuna girilen sayısal değerle kullanıcı tarafından belirlenmektedir. Bulunacak kural sayısı ve uzunluğu bu parametreye bağlı olarak değişmektedir.

V.2.5.2. Kural_say Alt Programı

Kural_say alt programı, oluşturulan aday kümede yer alan elemanların tekrar sayılarının bulunması ve minimum destek değerini aşan küme elemanlarının belirlenmesi için kullanılmaktadır. Bu alt program, parametre olarak “aday_eleman_kümesinin_uzunluğu” nu almaktadır. Aday eleman kümesinin uzunluğu, ilk adımda bir olarak atanmaktadır. Programda, her kural birleştirme işleminden sonra, kural uzunluğu bir artırılmaktadır. Kural_say alt programının algoritması Şekil V.7’de açıklanmaktadır. İlk adımda, aday küme elemanlarının tekrar sayıları hesaplanmaktadır. Daha sonra ise, aday küme elemanlarının, tekrar sayısı minimum destek değeri ile karşılaştırılmakta ve bu sınır değerini aşan elemanlardan, yeni sık_tekrar eden_eleman kümesi oluşturulmaktadır. Son adımda ise, kural sayısı olarak, destek değerini aşan eleman sayısı atanmaktadır.

1.Adım : Her öğrenci verisini kontrol ederek, her aday küme elemanı için, aday küme elemanın uzunluğu kadar ilerle ve her aday elemanın tekrar sayısını bul

2.Adım : Her aday küme elemanının tekrar sayısını, minimum destek değeri ile karşılaştır.

3.Adım : Minimum destek değerini aşan elemanlardan yeni sık_tekrarlanan_eleman kümesini oluştur ve kurala uyan eleman sayısını bul

4.Adım : Kural sayısına, kurala uyan eleman sayısı değerini ata

Şekil V.7 Kural_say alt programının algoritması

V.2.5.3. Birleştir Alt Programı

Birleştir alt programı, sık tekrar edilen kümenin kendi üzerine birleştirme işlemini gerçekleştirmektedir. Birleştirme işleminde, Lk, k uzunlukta sık tekrar eden

eleman kümesini temsil etmekte ve Lk'ya ulaşabilmek için Lk-1'in kendi üzerine birleştirilmesi gerekmektedir. Apriori algoritmasında, $L_{k-1} \times L_{k-1}$ birleştirme işleminin gerçekleştirilebilmesi için, elemanların ilk (k-2) elemanının ortak olması gerekmektedir. Bu kural, birleştirme sonucunda tekrar eden kuralların ortaya çıkmasını engellemek için yapılmaktadır.

Birleştir alt programı da, Apriori algoritmasındaki birleştirme kuralına uyarak işlemini gerçekleştirmektedir. Örneğin, uzunluğu iki olan ve beş elemana sahip sık_tekrarlan_elemanlar() dizisinden, uzunluğu üç olan aday dizisine ulaşabilmek için yapılacak birleştirme işleminde örnek veri olarak sık_tekrarlanan_elemanlar[]={ [1,2],[1,4],[1,5],[2,3],[2,6]} değerleri kabul edilsin. Aday dizisindeki her küme elemanı bir soru numarasını temsil etmektedir. Örneğin, [1,2] değeri, aynı konuda yer alan birinci ve ikinci sorunun aynı anda bilinme veya bilinmeme durumunu temsil etmektedir, diğer elemanlar da benzer şekilde yine soru numaralarını temsil etmektedirler. Bu kümeden, uzunluğu üç olan aday dizisine ulaşabilmek için Apriori algoritmasında kural gereği ilk (k-2) elemanı eşit olan elemanlar birleştirilebilmektedir. Bu örnekte, k=3 ve (k-2)=1 olmaktadır ve ancak birinci elemanları ortak elemanlar birleştirilir. Bu kural doğrultusunda, birleştir alt programı uzunluğu üç olan aday dizisini {[1,2,4], [1,2,5], [1,4,5], [2,3,6]} şeklinde oluşturur.

Şekil V.8'de birleştir alt programının algoritması görülmektedir. Yukarıda da açıklandığı gibi, alt program ilk (k-2). değerleri ortak olan dizi elemanlarını birleştirerek yeni aday diziyi oluşturur.

1.Adım : k=kural_uzunlugu+1 olarak ata 2.Adım : Her aday küme elemanı için ilk (k-2) elemanı eşit olanları birleştir 3.Adım : Yeni kümenin eleman sayısını hesapla ve say değişkenine ata

Şekil V.8 Birleştir alt programının algoritması

V.2.5.4. Kural_Kontrol Alt Programı

Kural_kontrol alt programı, birleştir alt programı tarafından oluşturulmuş aday eleman kümesinin alt kümelerini kontrol ederek, bu alt kümelerin bir önceki sık tekrarlanan küme içerisinde olup olmadığını kontrol etmektedir. Kural_kontrol alt

programına ait algoritma, Şekil V.9’da görülmektedir. Öncelikle kural uzunluğu bir artırılarak işleme başlanmaktadır. Kontrol yapılmadan önce, aday dizinin her elemanının, (Kural_uzunluğu-1) uzunluğuna sahip alt kümelerinin oluşturulması gerekmektedir. Eğer, oluşturulmuş alt kümelerin tümü, bir önceki sık_tekrar_eden_elemanlar dizisinde yer alıyorsa, dizi elemanı ancak o zaman aday küme içerisinde kalmaktadır. Herhangi bir alt kümesi, bir önceki adımdaki sık_tekrar_eden_elemanlar kümesinde yer almayan dizi elemanı, diziden budanmaktadır. Kural_kontrol alt programı içerisinde tüm elemanların alt küme kontrolü yapılarak, sadece kurala uyan elemanlardan oluşan aday dizisi oluşturulmakta ve bu dizinin eleman sayısı hesaplanmaktadır.

```
1.Adım : Kural_uzunluğunu bir artır
2.Adım : Aday dizinin her elemanı için uzunluğu
(kural_uzunluğu-1) olan alt kümeleri oluştur
3.Adım : Aday dizisindeki tüm elemanların alt kümelerinin
bir önceki adımdaki sık_tekrarlanan_elemanlar kümesinde var
olup olmadığını kontrol et
3.Adım : Alt kümesi sık_tekrarlanan_elemanlar kümesinde yer
almayan aday elemanları kümeden buda
4.Adım : Kurala uyan eleman sayısını bul ve kural_sayisi
değişkenine ata
```

Şekil V.9 Kural_kontrol alt programına ait algoritma

V.2.5.5. Guven_Say Alt Programı

Verilen minimum destek değerine göre, birliktelik kuralları oluşturulup görüntülendikten sonra, seçilen bir kurala ait güven değeri analizini guven_say alt programı gerçekleştirmektedir. Şekil V.10’daki algoritmanın birinci adımında görüldüğü gibi, öncelikle seçilen kurala ait, tekli ve çoklu alt kümeler oluşturulur. Daha sonra ise, bu kurallardan üretilebilecek birliktelik kuralları ve bu birliktelik kurallarının güven değerleri hesaplanır. Örneğin, Soru 1, Soru 2 ve Soru 5 arasındaki birliktelik kuralını gösteren {1 2 5} kuralı seçilerek, analiz yapılmak istendiğinde oluşturulabilecek tekli kurallar {1,2,5} çoklu kurallar {2 5, 1 5, 1 2} olacaktır. Bu kuraldan, oluşturulabilecek birliktelik kuralları ise ; 1=>2 5 , 2=>1 5, 5=> 1 2, 2 5=>1, 1 5=>2, 1 2=>5 şeklindedir. Guven_say alt programı, öncelikle bu kural

kombinasyonlarını oluşturur, daha sonra ise, Bölüm III.'de açıklanan güven değeri hesabı ile, oluşturulan kuralların güven değerlerini ekranda görüntüler.

1.Adım : Seçilen birliktelik kuralına ait tekli ve çoklu kuralları oluştur.

2.Adım : Oluşturulan kuralların tekrar değerlerini hesapla.

3.Adım : oluşturulan tüm kurallar için;

$$(A \Rightarrow B) = \frac{A \text{ ve } B' \text{nin bir arada bulunduğu kayıtların sayısı}}{A' \text{nin bulunduğu kayıtların sayısı}}$$

Formülü ile güven değerlerin hesapla ve ekranda görüntüle.

Şekil V.10 Guven_say alt programına ait algoritma

V.2.6. Çok Boyutlu Birliktelik Kuralı Bulan ZECOKA

Yazılımının Tasarımı

Birden fazla boyut içeren birliktelik kuralları, çok boyutlu birliktelik kuralı olarak adlandırılmaktadır. Örneğin, iki faktöre bağlı olarak bilgisayar alımını gösteren Denklem V.1'deki kural, geliri yüksek ve yaşı 30 ile 35 arasında olan kişilerin dizüstü bilgisayar alma durumunu göstermektedir [61, 132].

$$\text{Yaş}(X, "30..35") \wedge \text{Gelir}(X, "yüksek") \rightarrow \text{Alıyor}(X, "dizüstü bilgisayar") \quad (\text{V.1})$$

Yukarıdaki örnekte verildiği gibi, ZÖS'de yer alan farklı konular içerisindeki sorular arasında çok boyutlu birliktelik kurallarının analizinde kullanmak için ZECOKA yazılımı tasarlandı. Böylece, farklı konuların içerisinde yer alan alt kavramların birbirlerini ne şekilde etkiledikleri bulunmaktadır. Denklem V.2'de ZECOKA programı ile bulunabilen kurala ait bir örnek görülmektedir.

$$\text{Konu2} (\text{Soru2}=\text{Yanlış}) \wedge \text{Konu3} (\text{Soru4}=\text{Yanlış}) \rightarrow \text{Konu5} (\text{Soru3}=\text{Yanlış}) \quad (\text{V.2})$$

Denklem V.2'deki kural sonucunda, Konu2'de Soru 2 yanlış yanıtlanmış ve Konu3'de Soru4 yanlış yanıtlanmış ise, Konu5'de Soru3'ün yanlış yanıtlanma durumuna ait çok boyutlu birliktelik kuralı görülmektedir. Tasarlanan program ile, verilen minimum destek değerine bağlı olarak farklı sonuçlar ortaya çıkabilmektedir.

Tek boyutlu birliktelik kuralında sık tekrarlanan elemanlar bulunurken, çok boyutlu birliktelik kuralında ise, tekrarlanan belirteç kümesi bulunmaktadır. Denklem V.2'deki Konu2, Konu3 ve Konu5 kuraldaki belirteçlerdir.

Apriori algoritmasının çalışma mantığına uygun şekilde, ZECOKA yazılımının tasarımında Şekil V.11'de görülen sözde koddan yararlanıldı [61, 132].

Çok boyutlu birliktelik kuralı çıkarımında, öncelikle Şekil V.11'de, belirtildiği gibi, tüm boyutların bir elemanlı sık tekrarlanan kümelerin oluşturulması gerekmektedir. Bu işlem için, her boyut içindeki her elemanın tekrar sayısı hesaplanmakta ve tekrar sayısı minimum destek değerini aşan elemanlar ile, her boyut için S1 kümesi (sık tekrarlanan eleman kümesi 1) oluşturulmaktadır.

Adım 1: Veritabanındaki her boyut (konu) için , boyutta yer alan her elemanın (sorunun) bir elemanlı tekrar değerini bul, minimum destek değeri ile karşılaştır.

Adım 2: Minimum destek değerini aşan bir elemanlı küme elemanlarından her boyut için S1'i oluştur.

Adım 3: Tüm boyutlardaki S1 kümesini birleştir ve L1'i(cok boyutlu küme) oluştur.

Adım 4: For(k=2;k<=boyut && Lk-1 $\neq\emptyset$; k++)

Adım 4.1: Lk-1 birleştir ve Lk kümesini oluştur,

Adım 4.2: Lk kümesinin (k-1) elemanlı alt kümelerinin Lk'da varlığını kontrol et,

Adım 4.3: Alt kümelerinden herhangi biri Lk-1'de olmayan elemanı kümeden buda,

Adım 4.4: Lk'daki elemanların tekrar değerlerini tüm veritabanını tarayarak bul,

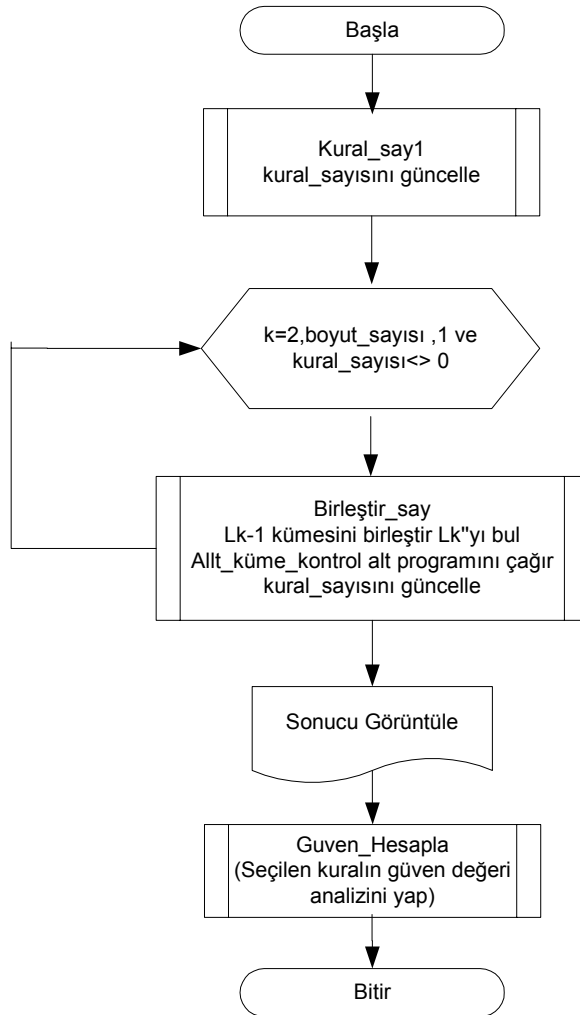
Adım 4.5: Minimum destek değerini aşmayan elemanları kümeden buda.

Şekil V.11 Çok boyutlu birliktelik kuralına ait sözde kod

Tüm boyutlara ait S1 kümeleri birleştirilerek, L1 kümesinin oluşturulması sağlanmaktadır. Oluşturulan L1 kümesi bir döngü içerisine alınarak, döngüdeki başlangıç değeri olarak kullanılmaktadır. Bu döngü içerisinde, Lk-1 kümesinin birleştirme işlemi, birleştirilmiş kümenin alt küme kontrolü ve birleştirilmiş yeni kümenin tekrar değerlerinin bulunması işlemleri yapılmaktadır. Tekrar değerleri, minimum destek değeri ile karşılaştırılarak, minimum destek değerini aşan

elemanların Lk kümesinde yer alması sağlanmaktadır. Bu döngü, k değişkeni boyut değerine eşit oluncaya kadar, Lk-1 kümesinin boş kümeden farklı olması şartıyla devam etmektedir.

Şekil V.12’de ZECOKA programına ait akış şeması görülmektedir. Programda öncelikle kural_say1 alt programı ile tüm boyutlara ait L1 kümesi bulunmaktadır. Daha sonra ise, birleştir_say alt programı ile bir döngü içerisinde Lk-1’in birleştirme ve alt küme kontrol işlemleri yapılmaktadır. Program, $k \leq \text{boyut_sayısı}$ ve $Lk-1 \neq \emptyset$ olduğu sürece devam etmektedir. Sonuçta oluşturulan çok boyutlu birliktelik kuralları görüntülenmekte ve seçilen kurala ait güven değeri hesaplaması Guven_Hesapla alt programı ile gerçekleştirilmektedir.



Şekil V.12 ZECOKA yazılımına ilişkin akış şeması

ZECOKA yazılımında kullanılan alt programlar aşağıda açıklanmaktadır:

1. **Kural_say1()**: Her konuya ait bir elemanlı sık tekrarlanan eleman kümesini bulma işlemi için kullanılan alt programdır. Alt program

- sonunda tüm konular içinde minimum destek değerini aşan elemanlar birleştirilerek, L1 kümesi oluşturulur.
2. **Alt_küme_kontrol():** Lk kümesinin (k-1) elemanlı alt kümelerinin tümünün sık tekrar edilip edilmediğini kontrol etmekte ve sık tekrar edilmeyen bir alt kümeye sahip olan elemanı Lk'dan budanmaktadır.
 3. **Birleştir_say():** Bir döngü içerisinde k=2'den başlayarak, konu sayısına kadar, Lk-1 kümesinin boş kümeden farklı olması şartını da kontrol ederek devam etmektedir. Döngü içerisinde ise, Lk-1 kümesinin kendi üzerine birleştirme işlemini gerçekleştirip Lk'yı elde ettikten sonra, Alt_küme_kontrol alt programını çağırarak Lk'nın (k-1) elemanlı alt kümelerinin sık tekrar edilip edilmediğini kontrol etmekte ve tüm veri setini tarayarak Lk'daki elemanların tekrar sayılarını bulmaktadır.
 4. **Guven_Hesapla():** Program sonucunda bulunan sık tekrarlanan eleman kümesinden istenilen kurala ait, güven değeri analizinin yapılmasını sağlamaktadır.

V.2.6.1. Programın Yüklenme Aşamasında Yapılan İşlemler

Tasarlanan yazılımda veri seti olarak, Bölüm V.2.4'de açıklanan veriler kullanıldı. Program ilk yüklendiğinde, her konuya ait oluşturulmuş altı adet metin dosyasının okunması ve program içerisinde kullanılmak üzere bir dizi yapısı içerisinde saklanması gerekmektedir. Bu amaçla, ogr(6,120) uzunluğunda ve String tipinde iki boyutlu bir dizi tanımlanmakta ve altı adet metin dosyasının içerisindeki değerler sırayla bu diziye yüklenmektedir. Öncelikle giriş konusunda yer alan öğrenci yanıt durumunun bulunduğu giris.txt dosyasında, her öğrenci için bulunan yedi adet yanıt durumu sırasıyla okunup, ogr dizisi içerisine alınmaktadır. ogr dizisinde ilk boyut konu numarasını, ikinci boyut öğrencinin veri setindeki sırasını temsil etmektedir. Giriş konusu, ilk sırada yer aldığı için iki boyutlu ogr dizisinde, örneğin ogr(1,1) birinci konuda, ilk sırada yer alan öğrenci verisini, ogr(1,2) birinci konuda ikinci sırada yer alan öğrenci verisini saklamak için kullanılmaktadır. Diğer konulara ait yanıtlar da benzer şekilde iki boyutlu dizi yapısı içerisinde saklanmaktadır.

Örnek öğrenci verisi, ogr(1,5)="0011111" olarak ele alındığında; ogr dizisinin ilk boyutunda yer alan 1 değeri, bu dizideki değerlerin ilk konu olan giriş konusuna ait

olduğunu, ikinci boyutta yer alan beş değeri ise, bu dizideki verinin veri setindeki beşinci öğrenciye ait olduğunu temsil etmek için kullanılmaktadır. Sonuç olarak ogr(1,5) verisi giriş konusuna ait, veri setinde beşinci sırada yer alan öğrenciye ait, konu sonu sınavına verilen yanıtları saklamaktadır. Bu dizideki değerden, öğrencinin birinci ve ikinci soruya yanlış, diğer sorulara doğru yanıt verdiği anlaşılmaktadır.

V.2.6.2. Kural_say1 Alt Programı

Çok boyutlu birliktelik kurallarında, öncelikle her boyuta ait bir elemanlı S1 (sık tekrar eden elemanlar 1) kümesi bulunmaktadır. Bu amaçla her boyutta yer alan sorulara ait, tekrar sayılarının hesaplanması ve minimum destek değeri ile karşılaştırılması gerekmektedir. Şekil V.13’de kural_say1 alt programının algoritması görülmektedir. Kural_say1 alt programı ile öncelikle, hangi boyutla ilgili işlem yapılıyorsa, boyutta yer alan her elemanın veri setindeki tekrar değeri hesaplanmaktadır. Daha sonra ise, hesaplanan tekrar değeri minimum destek değeri ile karşılaştırılarak, sadece minimum destek değerini aşan elemanların kümede kalması sağlanmaktadır. Tüm boyutlarda minimum destek değerini aşan elemanlar birleştirilerek, L1 kümesi oluşturulmakta ve oluşturulan kural sayısı bulunmaktadır.

Adım 1: Her boyut için , boyut içerisindeki her elemanın tekrar değerini bul,
Adım 2: Tüm boyutlardaki, eleman tekrar değerini minimum destek değeri ile karşılaştır, bu değeri aşan elemanlardan her boyut için S1’i oluştur
Adım 3: Tüm boyutlar için bulunan, S1 kümesini birleştir ve L1’i (çok boyutlu küme) oluştur ve kural sayısını bul

Şekil V.13 Kural_say1 alt programının algoritması

V.2.6.3. Alt_Küme_Kontrol Alt Programı

Çok boyutlu birliktelik kuralı çıkarımında, Apriori algoritmasına benzer şekilde, sık tekrarlanan kümeler arasında birleştirme işlemi yapılırken, yeni oluşturulan k boyutlu, k uzunluğa sahip bir kümenin (k-1) uzunluğa sahip alt kümelerinin, bir önceki adımdaki L_{k-1}’de bulunuyor olması gerekmektedir. Bu amaçla, oluşturulan her yeni aday kümenin (k-1) elemanlı alt kümeleri oluşturulup kontrol edilerek, bir önceki adımda oluşturulmuş L_{k-1}’de yer almayanlar L_k

kümesinden budanmaktadır. Alt_küme_kontrol alt programı oluşturulan, her yeni Lk kümesi için yukarıda belirtilen alt küme kontrolünü gerçekleştirmekte ve kurala uymayan herhangi bir eleman varsa, onu kümeden atmaktadır. Şekil V.14’de alt_küme_kontrol alt programına ait algoritma görülmektedir. Alt program çalıştırıldıktan sonra, (k-1) elemanlı alt kümelerinin tümü, Lk-1’de bulunan yeni Lk kümesi oluşturulacak ve Lk’da yer alan kural sayısı bulunacaktır

Adım 1: Oluşturulan Lk kümesinin, her elemanı için (k-1) elemanlı alt kümelerini oluştur
Adım 2: Her elemanın oluşturulmuş alt kümelerinin Lk-1’de varlığını kontrol et
Adım 3: Tüm alt kümeleri Lk-1’de yer almayan elemanı kümeden buda ve oluşturulan kural sayısını bul

Şekil V.14 Alt_küme_kontrol al programına ait algoritma

V.2.6.4. Birleştir_say Alt Programı

Kural_say1 alt programı ile tüm boyutlara ait S1(sık tekrarlanan eleman 1) kümesi bulunup, tüm boyutlardaki S1 birleştirilerek, L1 oluşturulduktan sonra, L1 kümesinin birleştirilmesiyle çok boyutlu birliktelik kurallarının oluşturulması ve bu kuralların tekrar sayılarının hesaplanması gerekmektedir. Birleştir_say alt programına ait algoritma Şekil V.15’de görülmektedir. Şekil V.15’deki ilk adımda görüldüğü gibi, birleştirme ve kural sayma işlemi, $k=2$ ’den başlamakta ve boyut sayısına kadar, Lk-1’in eleman sayısı sıfırdan farklı olmak şartıyla ilerleyen bir döngü içerisinde devam etmektedir. Birleştir_say alt programında ilk önce, L(k-1) sık tekrar eden küme elemanlarını birleştirilerek, Lk’yı bulmaktadır. Birleştirme işleminde, kural olarak ancak ilk (k-2). değeri ortak olan elemanlar birleştirilmektedir. Bu kural için kontrol işlemi yapılarak, Adım 1.1’de birleştirme işlemi yapılmaktadır. Daha sonra ise, alt_kume_kontrol alt programı çağrılarak oluşturulan Lk kümesinin (k-1) elemanlı alt kümelerinin sık tekrar eden elemanlar olup olmadığı, Adım 1.2’de kontrol edilmektedir. Ancak, tüm alt kümeleri sık tekrar eden elemanlar kalabilmekte, diğerleri Lk kümesinden budanmaktadır. Adım 1.3’de ise, tüm elemanları kontrol edilmiş, Lk kümesinin her elemanın tekrar değeri bulunmaktadır. Tekrar değerleri Adım1.4’de minimum destek değeri ile karşılaştırılarak, sadece minimum destek değerini aşan elemanların Lk’da kalması,

diğerlerinin kümeden budanması sağlanmaktadır. Bu işleme, $k \leq \text{boyut sayısı}$ ve L_k-1 boş kümeden farklı olduğu sürece devam edilmektedir.

Adım 1: For($k=2; k \leq \text{boyut} \ \&\& \ L_{k-1} \neq \emptyset; k++$)

Adım 1.1: L_{k-1} kümesinde ilk $(k-2)$ elemanı ortak olan elemanları birleştir ve L_k kümesini oluştur,

Adım 1.2: Alt_kume_kontrol alt programını çağır (L_k 'nin $(k-1)$ elemanlı alt kümelerinin L_{k-1} 'deki varlığını kontrol et)

Adım 1.3: L_k 'daki elemanların tekrar değerlerini tüm veritabanını tarayarak bul

Adım 1.4: Minimum destek değerini aşamayan elemanları kümeden buda ve oluşturulan kural sayısını bul

Şekil V.15 Birleştir_say alt programına ait algoritma

V.2.6.5. Guven_Hesapla Alt Programı

Oluşturulan çok boyutlu birliktelik kurallarına ait, güven değeri analizi, istenilen bir kural seçilerek, guven_hesapla alt programı ile yapılmaktadır. Bu işlem için, Şekil V.16'daki algoritmada görüldüğü gibi öncelikle seçilen çok boyutlu birliktelik kuralına ait tekli ve çoklu kurallardan oluşan alt kümeler oluşturulur. İkinci adımda, oluşturulan bu kuralların, tüm veri setindeki tekrar değerleri hesaplanır. Üçüncü adımda ise, bu alt kümelerle, güven değeri analizinde incelenecek kural kombinasyonları oluşturulur ve Bölüm III'de açıklanan güven değeri hesabı ile, oluşturulan kuralların güven değerleri ekranda görüntülenir.

1.Adım : Seçilen çok boyutlu birliktelik kuralına ait tekli ve çoklu kural kümelerini oluştur

2.Adım : Oluşturulan kuralların tekrar değerlerini hesapla

3.Adım : Kural kombinasyonlarını oluştur ve tüm kurallar için güven değerlerini hesapla

Şekil V.16 Guven_Hesapla alt programına ait algoritma

V.3. KONU BAŞARI PUANLARININ KÜMELEME ANALİZİ

Bu bölümde, Bölüm IV’de tasarımı açıklanan zeki öğretim sisteminin öğrenciler tarafından kullanılması ile elde edilen MYCIN güven faktörü hesaplama yöntemi ile hesaplanmış, konu sonu başarı puanlarının (güven değerleri) kümeleme analizi için yazılım tasarımı gerçekleştirilmiştir.

Öğrencilerin konular düzeyinde başarı dağılımlarının görüntülenmesi, sınıftaki not dağılımının ne şekilde olduğunu görmek ve oluşan öğrenci kümelerini belirlemek için gereklidir. Standart istatistiksel yöntemlerle, kümeleme dağılımının elde edilmesi mümkün değildir. Bu dağılım, öğretmenin sınıftaki ders işleyiş seviyesini, pedagojik kararlarını düzenlenmesine yardımcı olur. Ayrıca sınıfta grup çalışması amacıyla heterojen grupların oluşturulması aşamasında da kümeleme analizinden faydalanılır.

Tasarlanan yazılımda, öğrencilerin belirlenen konulardan aldıkları başarı puanları, k-means kümeleme algoritması ile bulunmakta ve oluşan öğrenci kümelerinin görüntülenmesi sağlanmaktadır.

Kümeleme analizi için ise, öğreticisiz kümeleme algoritmalarından k-means [89, 90] algoritması kullanılmıştır. K-means algoritması, bölümleyici (partitional) kümeleme algoritmalarındandır [78, 133]. Gerçekleştirilen uygulamada, öğrencilerin belirlenen konulardan aldıkları başarı puanları, k-means algoritmasına göre kümelenecek, bu algoritmanın eğitim alanında bir veri üzerinde uygulanması sağlanmıştır.

K-means algoritması, başlangıçta belirlenen rasgele merkez atamasından sonuçları etkilenen bir algoritmadır. Özellikle, ilk atamadaki merkez noktası birbirine çok yakın değerlerden oluşuyor ise, k-means algoritmasının performansı azalmaktadır. Bu negatif etkiyi ortadan kaldırmak için, bu tezde her veri elemanını sırayla işleme dahil eden ve her yeni eleman için k-means algoritmasını tekrar çalıştıran bir yazılım tasarımı gerçekleştirilmiştir.

V.3.1. Yazılımda Kullanılan K-Means Kümeleme Algoritmasına Ait Ölçüt ve Hata Fonksiyonu

Bölümleyici bir algoritma olan k-means algoritması, belirlenen ölçüt fonksiyonuna göre sonucu optimize etmektedir. Ölçüt tanımlaması yerel veya genel olabilmektedir. Genel ölçüte örnek olarak, öklit uzaklığının karesi verilebilir. Bu ölçüt kullanıldığında, her kümenin merkezi hesaplanır ve veri örneği en yakın kümeye dahil edilir. Minimum karşılıklı komşu uzaklığı (MND, Minimal Mutual Neighbor Distance) yerel ölçüte örnek olarak verilebilir. Bu tür ölçütlerde, yüksek yoğunluklu veri aralığı bulunarak kümeleme yapılmaktadır.

K-means algoritması, kullanılan veri setinin büyük olmaması ve istisna değerlerin veride bulunmaması sebebiyle tasarlanan kümeleme programında esas alınmıştır. K-means algoritmasının sonuç alınması büyük veri setlerinde fazla zaman gerektirdiği için tercih edilmemektedir. Analiz edilen öğrenci verilerinde, öğrenci başarı puanlarının aralığı bellidir, veri setindeki değerler (-100,+100) aralığında değişmektedir ve toplam 103 öğrenciye ait veri incelenmiştir.

K-means algoritmasında, kümele işlemi için en çok kullanılan hesaplama yöntemi hatanın karesi ölçütüdür, tasarlanan programda da bu ölçüt kullanılmıştır. Bu ölçüt ile, veri kümesinde belli bir sayıda kümeye ulaşılır ve toplam hata karesini minimum yapan veriler kümelere dahil edilir. Örneğin, N elemanlı, n boyutlu bir veri seti K kümeye ayrılmak istendiğinde küme isimleri $\{C_1, C_2, \dots, C_k\}$ olarak gösterilebilir. Her C_k kümesi n_k kadar veri örneğine sahiptir ve tüm kümelerdeki verilerin toplamı ($\sum n_k = N$, $k=1, \dots, K$) eleman sayısı N'ye eşit olmaktadır. C_k kümesinin merkezini ifade eden, ortalama vektör M_k ise, merkez(centroid) olarak da adlandırılmakta ve Denklem V.3'deki gibi hesaplanmaktadır [78].

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} \quad (V.3)$$

Burada; x_{ik} , C_k kümesine ait i. veri örneğini temsil etmektedir. C_k kümesi için hatanın karesi, C_k kümesine ait tüm veri örnekleri ile C_k 'nin merkezi arasındaki öklit uzaklıklarının karesi ile hesaplanmaktadır. Bu hata fonksiyonu, küme içi varyasyon olarak da adlandırılmakta ve Denklem V.4'deki gibi hesaplanmaktadır.

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2 \quad (V.4)$$

K adet kümede yer alan toplam hata ise, küme içindeki varyasyonların toplamına eşit olmakta ve Denklem V.5'deki gibi hesaplanmaktadır.

$$E_k^2 = \sum_{k=1}^K e_k^2 \quad (V.5)$$

K-means bölümleyici kümeleme algoritması, ilk bölümleme işlemini rasgele yaparak işleme başlamakta ve kalan veri örneklerinin kümelerini belirlemektedir. Küme belirleme işleminde, veri örneği ve küme arasındaki benzerlik dikkate alınmaktadır. Artık hiçbir küme elemanının yerinde değişiklik meydana gelmeyinceye kadar, işleme devam edilmektedir. K-means algoritması yaygın ve kolay uygulanabilir bir algoritmadır, fakat başlangıç noktası seçiminden etkilenmektedir.

K-means algoritması hesaplama yöntemi ile özellikle sıkışık ve küresel verilerde etkili sonuçlar vermektedir. K-means algoritmasının temel çalışma yöntemi aşağıda görülmektedir:

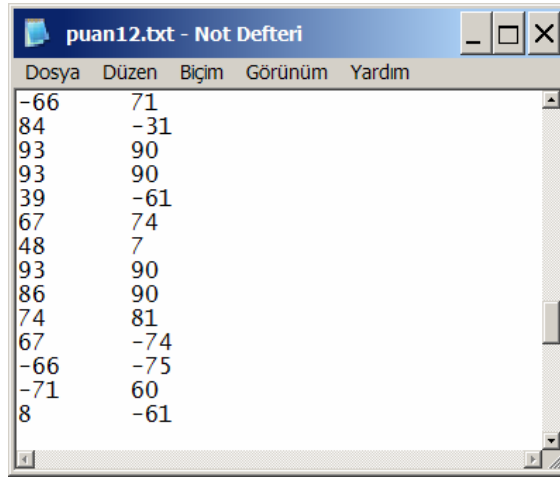
1. İlk bölümleme için K adet veri örneği belirle, ve buna göre merkezleri hesapla,
2. Her veri örneğini kendisine en yakın küme merkezine ait olacak şekilde ata,
3. Küme merkezlerini yeniden hesapla,
4. İkinci ve üçüncü adımı, hiçbir küme elemanında yer değişikliği olmayıncaya kadar tekrar et [61, 78].

V.3.2. Kullanılan Veri Yapısı

Tasarlanan k-means kümeleme analizi yazılımında, zeki öğretim sisteminin öğrenciler tarafından kullanılması ile oluşan, gerçek bir veri seti kullanılmıştır. Veri setinde, öğrencinin belirlenen konulardan aldığı MYCIN güven faktörü değerleri, yani konu başarı puanları bulunmaktadır. Bölüm IV.4'de öğretim modelinde de belirtildiği gibi, MYCIN güven faktörü hesaplama tekniği sonucunda, öğrencinin herhangi bir konudan aldığı güven değeri -1 ile +1 aralığındadır. Güven değerinin,

-1'e yaklaşması, öğrencinin o konu için bilgi düzeyindeki eksikliği, +1'e yaklaşması ise konuyu bilindiği varsayım durumunu göstermektedir.

Öğrencilerin aldıkları başarı puanları, Bölüm IV.3'de açıklanan TblPuan tablosundan alınarak uygulama kolaylığı açısından, her değer öncelikle 100 ile çarpılır. Yeni oluşan değerler tablosundan, belirlenen konulara ait başarı puanları alınarak, metin dosyasına dönüştürülür ve analiz için hazır hale getirilir. Örneğin, Şekil V.17'de ZÖS'de yer alan giriş ve veriyolu yapıları konularına ait, iki boyutlu ve değerleri (-100, +100) aralığında değişen, analiz için uygun biçime dönüştürülmüş verisetine ait ekran görülmektedir. Şekil V.17'de ilk sütun giriş konusuna ait başarı puanlarını, ikinci sütun ise veriyolu yapılarına ait başarı puanlarını göstermektedir. Tasarlanan yazılımla, ikiden daha fazla boyuta sahip başarı puanlarının kümeleme analizinin yapılması da mümkündür.



-66	71
84	-31
93	90
93	90
39	-61
67	74
48	7
93	90
86	90
74	81
67	-74
-66	-75
-71	60
8	-61

Şekil V.17 K-means kümeleme algoritması için kullanılan verisetinin ekran görüntüsü

V.3.3. Tekrarlı K-means Algoritmasına Göre Kümeleme Yapan Yazılım Tasarımı

K-means algoritmasının rasgele yapılan başlangıç merkezi seçiminden negatif etkilenmesini ortadan kaldırmak için, algoritmanın her adımda bir veri ögesi işleme dahil edilerek, tüm veri seti tamamlanıncaya kadar her adımda yeniden çalıştırılmasını temel alan bir yaklaşım esas alınmıştır. Standart K-means algoritmasına göre kümeleme yapan yazılımdan farklı olarak, tekrarlı çalışan K-means algoritmasında, ilk veriden başlayarak, her eleman sırayla alınmakta ve her yeni eleman geldiğinde, yeni küme merkezleri belirlenerek, her elemanın hangi kümede yer alacağı belirlenmektedir.

Tekrarlı K-means algoritmasında, programın yükleme aşamasında, veri seti metin dosyasından okunmakta ve oluşturulan veri seti, veri() dizisi içerisine alınmaktadır. Tekrarlı K-means algoritmasında, kmeans() ve uzaklik() alt programlarından oluşan iki alt program kullanılmaktadır.

Kmeans() alt programı: Bu alt program her çağrıldığında; yani yeni bir veri örneği işleme dahil edildiğinde, öncelikle kaç adet veri örneği için kümeleme işleminin yapılacağı kontrol edilmektedir. Eğer veri sayısı küme sayısından küçük ise, merkez nokta olarak veri örneğinin değeri atanmaktadır. Eleman sayısı küme sayısından fazla olduğunda ise, öncelikle yeni gelen veri örneğinin mevcut kümeler içinde en yakın kümeye dahil edilmesi sağlanmaktadır. Daha sonra ise, oluşan yeni durum için merkez noktaları tekrar hesaplanmakta ve o anda üzerinde çalışılan tüm veri örneği için, kümelere olan uzaklıklar hesaplanarak her veri örneğinin en yakın kümeye dahil edilmesi sağlanmaktadır. Bu işleme hiçbir veri örneğinin kümesinde değişiklik durumu ortaya çıkmayana kadar devam edilmektedir.

Uzaklık fonksiyonu: Bu fonksiyon ile belirlenen iki nokta arasındaki uzaklık, Denklem III.5’de tanımlanan öklit uzaklığının karesi formülü ile hesaplanmaktadır.

Şekil V.18’de tekrarlı çalışan K-means algoritmasına göre kümeleme işlemi yapan yazılımın algoritması görülmektedir. Yazılım, a=1’den başlayarak tüm veri kümesi işleme dahil edilinceye kadar devam eden bir döngü üzerinde işlem yapmakta ve her seferinde kmeans alt programını çağırılmaktadır.

```
Adım 1: For(a=1;a<=eleman sayısı a++){  
Adım 2: kmeans (a)alt programını çağır  
    Adım 2.1: a<küme sayısı ise küme merkezi koordinatları  
        olarak k veri noktasının koordinatlarını ata  
    Adım 2.2: a>küme sayısı ise, yeni gelen a noktasını  
        en yakın kümeye ata  
    Adım 2.3: Tüm veri kümesindeki elemanların merkez  
        noktalara uzaklığını uzaklik() fonksiyonu ile hesapla  
        ve veri elemanını minimum uzaklığa sahip kümeye dahil  
        et  
    Adım 2.4: Hiçbir veri örneğinin yeri değişmeyene kadar  
        Adım 2.3’e git ve işlemleri tekrarla.}
```

Şekil V.18 Tekrarlı çalışan K-means algoritmasına göre kümeleme işlemi yapan yazılımın algoritması

Kmeans alt programında, adım 2.1’de görüldüğü gibi küme sayısı ile a değeri karşılaştırılmaktadır. a değeri küme değerinden küçükse, merkez koordinat olarak a veri örneğinin tüm boyutlarındaki değerler atanmaktadır. a değeri küme sayısından büyük olduğunda ise, öncelikle yeni gelen a değeri en yakın kümeye dahil edilmektedir. Daha sonra ise, işleme dahil edilen veri örneğinin tümünün (birinci veri örneğinden a. veri örneğine kadar) küme merkezlerine olan uzaklıkları hesaplanmakta ve en yakın kümeye ait olmaları sağlanmaktadır. Artık küme elemanlarında yer değişikliği olmayana kadar işleme devam edilmektedir. Programın başlangıcında oluşturulan döngü ise, a=eleman sayısı oluncaya kadar devam etmekte ve her seferinde kmeans alt programını çağırılmaktadır.

V.4. VERİ MADENCİLİĞİ ANALİZ YAZILIMLARININ TEST SONUÇLARI

V.4.1. ZEBİKA’ya Ait Test Sonuçları

ZEBİKA yazılımının test edilmesi için, Bölüm V.2.3, Tablo V.1’de yer alan örnek öğrenci verileri kullanılmıştır. Bu veri seti üzerinde Apriori algoritmasının ne şekilde çalıştığı adım adım Tablo V.3’de belirtilmiştir. Tablo V.3’de minimum destek değeri 2 ile yapılan analizde, kural uzunluğu üç olan dört adet kural bulunmuştur. ZEBİKA yazılımında, minimum destek değeri 2 ile yapılan analize ait sonuçlar Tablo V.19’da görülmektedir. Bu sonuçlar, Tablo V.3’de yer alan sonuçların aynısıdır.

TEST

Minimum Destek=

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 6	Kurala uyanlar = 9	Kurala uyanlar = 4
1 Tekrar= 3	1 4 Tekrar= 2	1 4 5 Tekrar= 2
2 Tekrar= 2	1 5 Tekrar= 3	1 5 7 Tekrar= 2
4 Tekrar= 7	1 7 Tekrar= 2	2 4 5 Tekrar= 2
5 Tekrar= 6	2 4 Tekrar= 2	4 5 6 Tekrar= 2
6 Tekrar= 3	2 5 Tekrar= 2	
7 Tekrar= 2	4 5 Tekrar= 5	
	4 6 Tekrar= 3	
	5 6 Tekrar= 2	
	5 7 Tekrar= 2	

ANALİZİ YAPILABİLECEK KURALLAR

1 4 5
1 5 7
2 4 5

Şekil V.19 ZEBİKA’nın test verisi ile yapılan analiz sonuçları

{145} kuralının güven değeri analizi, Bölüm 3’de, Denklem III.3’deki gibi hesaplanmaktadır. Bu denkleme göre, 45=>1 kuralına ait güven değeri analizi Denklem V.6’daki gibi %40 olarak hesaplanmıştır.

$$4\ 5 \Rightarrow 1 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{45\text{'in bulunduğu kayıtların toplamı}} = \frac{2}{5} = 0.4 = \%40 \quad (\text{V.6})$$

{145} ile oluşturulan diğer kurallar da benzer şekilde aşağıdaki gibi hesaplanmaktadır.

$$1\ 5 \Rightarrow 4 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{15\text{'in bulunduğu kayıtların toplamı}} = \frac{2}{3} = 0.666 = \%67 \quad (\text{V.7})$$

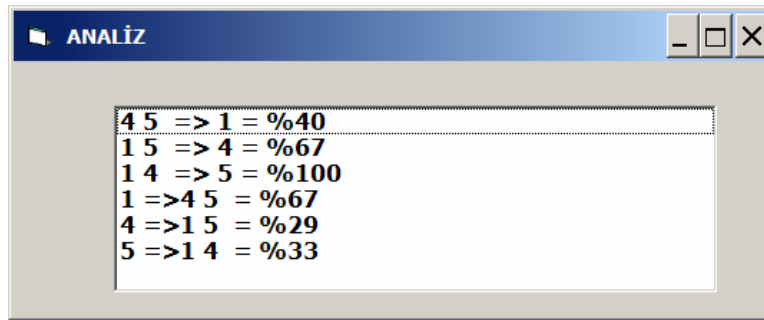
$$1\ 4 \Rightarrow 5 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{14\text{'ün bulunduğu kayıtların toplamı}} = \frac{2}{2} = 1 = \%100 \quad (\text{V.8})$$

$$1 \Rightarrow 4\ 5 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{1\text{'in bulunduğu kayıtların toplamı}} = \frac{2}{3} = 0.666 = \%67 \quad (\text{V.9})$$

$$4 \Rightarrow 1\ 5 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{4\text{'ün bulunduğu kayıtların toplamı}} = \frac{2}{7} = 0.285 = \%29 \quad (\text{V.10})$$

$$5 \Rightarrow 1\ 4 = \frac{145\text{'in bulunduğu kayıtların toplamı}}{5\text{'in bulunduğu kayıtların toplamı}} = \frac{2}{6} = 0.333 = \%33 \quad (\text{V.11})$$

ZEBİKA’da elde edilen güven değeri analizi ise, Şekil V.20’deki görülmektedir. Bu sonuçlar Denklem V.6, Denklem V.7, Denklem V.8, Denklem V.9, Denklem V.10 ve Denklem V.11’deki sonuçların aynısıdır.



Şekil V.20 ZEBİKA’nın test verisi ile yapılan güven değeri analizi

Test verisi sonuçlarına göre, ZEBİKA yazılımı birliktelik kuralı analiz sonuçlarını doğru hesaplamaktadır.

V.4.2. ZECOKA’ya Ait Test Sonuçları

Tablo V.5’de görülen örnek veri üzerinde, çok boyutlu birliktelik kuralı analizi gerçekleştirilmektedir. Ogr dizisi iki boyutlu bir dizidir ve ilk boyut konu numarasını,

ikinci boyut ise öğrencinin veri setindeki numarasını ifade etmektedir. Örneğin, bu dizi yapısında $ogr(1,3)$ birinci öğrencinin üçüncü konuya ait yanıtlarını göstermektedir. Tablo V.5’de alt öğrencinin üç boyuttan oluşan veriseti görülmektedir.

Tablo V.5 Çok boyutlu öğrenci yanıt dizisi

Boyut	Soru No	1	2	3	4	Yanıt Dizisi
	Dizi					
1. Boyut Konu 1	$ogr(1,1)$	1	1	1	0	"1110"
	$ogr(1,2)$	0	1	1	1	"0111"
	$ogr(1,3)$	0	1	1	0	"0110"
	$ogr(1,4)$	1	1	1	0	"1110"
	$ogr(1,5)$	1	0	1	0	"1010"
	$ogr(1,6)$	0	1	1	0	"0110"
2. Boyut Konu 2	$ogr(2,1)$	1	0	1	1	"1011"
	$ogr(2,2)$	1	0	0	1	"1001"
	$ogr(2,3)$	0	0	1	1	"0011"
	$ogr(2,4)$	1	1	1	0	"1110"
	$ogr(2,5)$	1	1	1	0	"1110"
	$ogr(2,6)$	0	0	1	1	"0011"
3. Boyut Konu 3	$ogr(3,1)$	1	1	1	0	"1110"
	$ogr(3,2)$	1	1	1	0	"1110"
	$ogr(3,3)$	0	1	0	1	"0101"
	$ogr(3,4)$	0	1	0	1	"0101"
	$ogr(3,5)$	1	0	0	1	"1001"
	$ogr(3,6)$	1	1	1	0	"1110"

Çok boyutlu birliktelik kuralı analizinde öncelikle her boyuta ait sık tekrarlanan eleman kümesi $L1$ ’in bulunması gerekmektedir. Minimum destek değeri 3 ile yapılan analizde her boyuta ait tek boyutlu birliktelik kuralları Tablo V.6’da görülmektedir. Bir sonraki adımda ise, iki boyutlu kuralların oluşturulması ve tekrar değerlerinin bulunarak sadece minimum destek değerini aşanların bu kümede yer almasının sağlanması gerekmektedir. Bulunan iki boyutlu birliktelik kuralları Tablo V.7’de görülmektedir. Son adımda ise, üç boyutlu kurallar oluşturularak tekrar değerlerinin bulunması gerekmektedir. Fakat oluşturulan üç boyutlu kurallardan, hiç birisi minimum destek değeri 3’ü aşmadığı için algoritma son bulmaktadır.

Tablo V.6 Test verisi ile bulunan tek boyutlu birliktelik kuralları

Tek Boyutlu Kurallar		
Konu No	Soru No	Tekrar
1	1	3
1	4	5
2	2	4
3	3	3
3	4	3

Tablo V.7 Test verisi ile bulunan iki boyutlu birliktelik kuralları

İki Boyutlu Kurallar		
Konu No	Soru No	Tekrar
12	1, 2	3
12	4, 2	3
13	4, 3	3
23	2, 4	3

Şekil V.21’de, sonuç ekranı görülen ZECOKA yazılımı, Tablo V.5’deki veri üzerinde, minimum destek değeri 3 ile yaptığı analizde Tablo V.6’daki sonuçların aynısını bularak son bulmuştur. Sonuçta iki boyutlu dört kural bulunmuştur.

ÇOK BOYUTLU BİRLİKTELİK KURALLI ANALİZİ

Konu1: 1 Tekrar=3, 4 Tekrar=5
Konu2: 2 Tekrar=4
Konu3: 3 Tekrar=3, 4 Tekrar=3

Destek Değeri: 3 Kuralları Bul

Veriler: Öğrenci Sayısı: 6

ANALİZİ YAPILABİLECEK KURALLAR

K= 12 S 12 T:3
K= 12 S 42 T:3
K= 13 S 43 T:3
K= 23 S 24 T:3

Güven Değeri Analizi

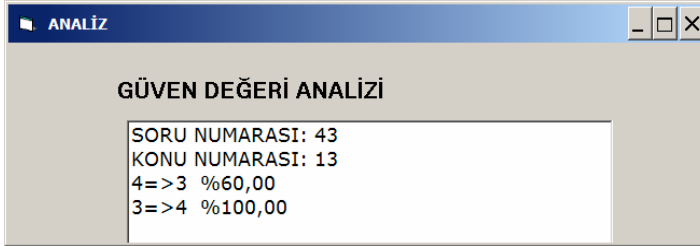
Şekil V.21 ZECOKA’nın test verisi ile yapılan analiz sonuçları

ZECOKA’nın Tablo V.5’deki veri ile yaptığı analize ait güven değeri sonuçları ise Şekil V.22’de görülmektedir. Güven değeri analizine ise, Konu1’deki dördüncü soru ve Konu3’deki üçüncü soru arasındaki iki boyutlu kuralı gösteren kurala ait

güven değeri analizi Denklem V.12 ve Denklem V.13'deki gibi hesaplanmaktadır. ZECOKA'nın aynı kural için hesapladığı güven değerleri ise Şekil V.22'de görülmektedir.

$$4 \Rightarrow 3 = \frac{\text{Konu1'deki 4.soru ve Konu3'deki 3. sorunun tekrar değeri}}{\text{Konu1'deki 4.sorunu tekrar değeri}} = \frac{3}{5} = 0.6 = \%60 \quad (\text{V.12})$$

$$3 \Rightarrow 4 = \frac{\text{Konu1'deki 4.soru ve Konu3'deki 3. sorunun tekrar değeri}}{\text{Konu3'deki 3.sorunu tekrar değeri}} = \frac{3}{3} = 1 = \%100 \quad (\text{V.13})$$



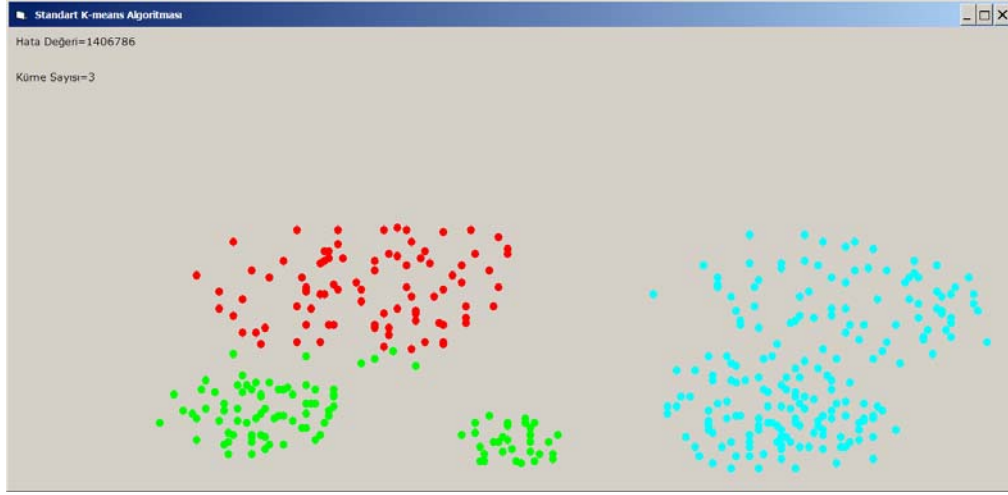
Şekil V.22 ZECOKA'nın test verisi ile yapılan güven değeri analizi

Test verisi sonuçlarına göre, ZECOKA yazılımı çok boyutlu birliktelik kuralı analiz sonuçlarını doğru hesaplamaktadır.

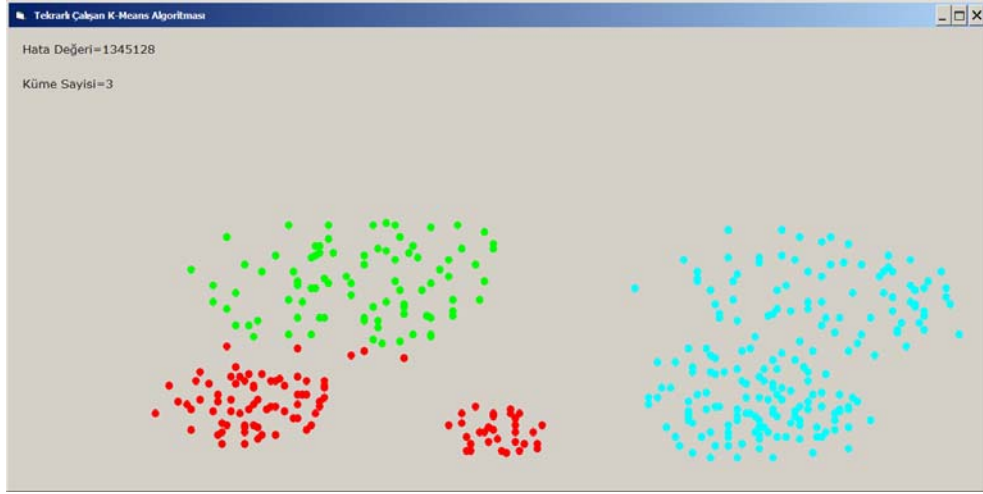
V.4.3. Kümeleme Analizine Ait Test Sonuçları

İki farklı veriseti kullanılarak, standart ve tekrarlı k-means algoritmasına göre çalışan kümeleme analiz yazılımlarının ortaya çıkardıkları toplam hata değeri bulunarak, kümeleme başarıları test edilmektedir. Toplam hata için, Denklem V.5'deki formül kullanılmaktadır.

Birinci veriseti ile, Şekil V.23'de standart k-means algoritmasına göre analiz yapan yazılımda küme sayısı üç olarak belirlendiğinde, toplam hata değeri 1406786 olarak hesaplanırken, Şekil V.24'de görülen tekrarlı k-means algoritmasını toplam hata değeri 1345128'dir. Bu sonuca göre, bu veriseti için, tekrarlı çalışan kümeleme analiz yazılımı daha az hata üretmektedir.

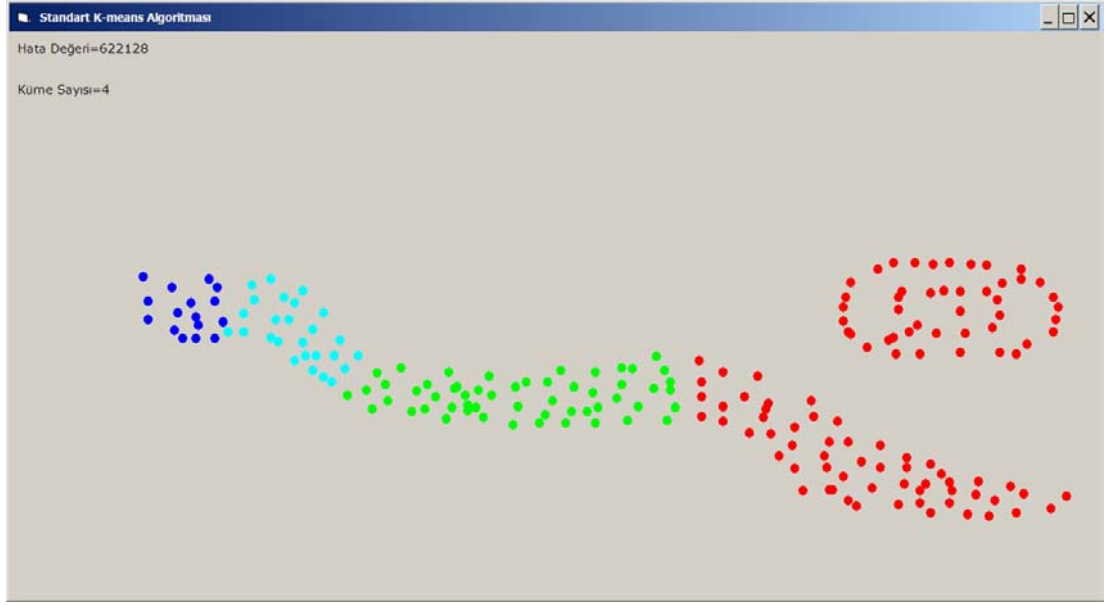


Şekil V.23 Standart k-means algoritmasının birinci test verisi ile analiz sonucu

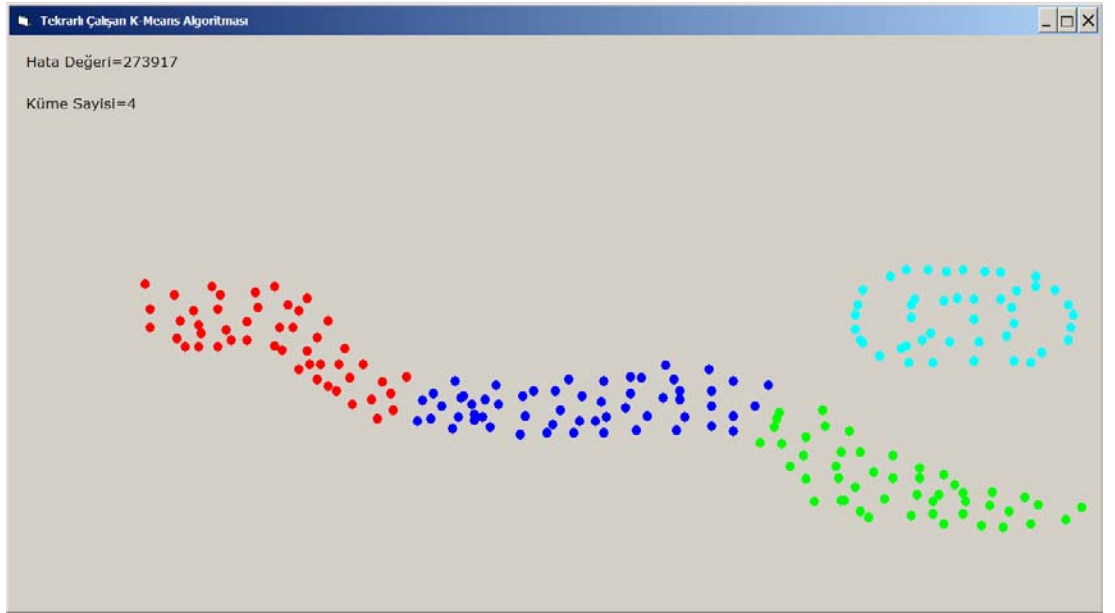


Şekil V.24 Tekrarlı çalışan k-means algoritmasının birinci test verisi ile analiz sonucu

Şekil V.25’de; ikinci veriseti ile küme sayısı dört olarak belirlendiğinde, standart k-means algoritmasına göre çalışan programda, toplam hata değerinin 622128 olduğu görülmektedir. Aynı veri seti için tekrarlı çalışan k-means algoritmasında, küme sayısı dört ile analiz yaptığında, Şekil V.26’da görüldüğü gibi, 273917 hata değeri bulunmuştur. İkinci verisetinde, tekrarlı çalışan k-means algoritmasının daha az hata sonucu ile kümeleme analizi gerçekleştirdiği görülmektedir.



Şekil V.25 Standart k-means algoritmasının ikinci test verisi ile analiz sonucu



Şekil V.26 Tekrarlı çalışan k-means algoritmasının ikinci test verisi ile analiz sonucu

Her iki veriseti ile yapılan denemeler sonucunda tekrarlı çalışan k-means algoritmasının standart k-means algoritmasına göre, daha az toplam hata oluşturarak kümeleme işleminde daha başarılı olduğu görülmüştür.

BÖLÜM VI

SONUÇLAR

VI.1. AMAÇ

Bölüm V’de kullandıkları veri seti ve tasarımları açıklanan birliktelik kuralı analiz yazılımları ve kümeleme yazılımına ait sonuçlar, bu bölümde açıklanmakta ve değerlendirilmektedir.

Bu çalışmada oluşturulan ZÖS, Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü’nde 8.dönem örgün ve ikinci öğretim öğrencilerine uygulanmıştır. ZÖS’ü, örgün öğretimde 42 öğrenci, ikinci öğretimde 61 öğrenci olmak üzere, toplam 103 öğrencinin kullanması sağlanmıştır. Birliktelik kuralı analiz yazılımları, bu öğrencilerin ZÖS’de bulunan verilerini analiz etmektedir. Bu yazılımlar ile, öğrencilerin doğru ve yanlış yanıtlarına ait birliktelik kurallarının oluşturulması mümkündür. Birliktelik kuralı yazılımları; örgün ve ikinci öğretim öğrencilerinin verilerinin birleştirilmesi ile oluşan veri üzerinde, öğrencilerin yanlış yanıtları üzerinde analiz yapmıştır.

Birliktelik kuralı analiz yazılımları ile bu bölümde, ZÖS’de bulunan öğrenci verileri kullanılarak aşağıda belirtilen üç farklı birliktelik kuralı bulunmaktadır. Bunlar;

- Aynı konu içerisindeki sorulardaki başarısızlık ilişkisine ait birliktelik kurallarının ZEBİKA ile bulunması,
- ZÖS’de bulunan altı adet konu arasındaki başarısızlık ilişkisini gösteren birliktelik kurallarının ZEBİKA ile bulunması,
- Konular arası sorulardaki başarısızlık ilişkisini gösteren çok boyutlu birliktelik kurallarının ZECOKA ile bulunması .

Kümeleme analizi yapan yazılım ise; öğrenci verileri üzerinde, ZÖS’de bulunan altı farklı konu içerisinde, istenilen konulara ait başarı puanına göre k-means algoritması ile kümeleme işlemini gerçekleştirmektedir.

Yukarıdaki üç maddede açıklanan birliktelik kurallarına ait sonuçlar ve kümeleme analizi sonuçları tüm öğrenciler açısından, bu bölümde açıklanarak değerlendirilecektir.

VI.2. TÜM ÖĞRENCİLERE AİT BULUNAN BİRLİKTELİK KURALLARI

VI.2.1. Tüm Öğrenci Verilerinde Aynı Konuda Yer Alan Sorulara Ait Bulunan Birliktelik Kuralları

Örgün ve ikinci öğretiminden, toplam 103 öğrenci ve altı konuya ait veri ile bulunan birliktelik kuralları sonuçları bu kısımda açıklanmaktadır. ZÖS’de bulunan konu isimleri ve kaç adet soru içerdikleri, Bölüm V.2.4’de açıklanmıştır. Sonuçlar sistemde izlenen konu sırasına uygun olarak açıklanmaktadır. Birliktelik kurallarının bulunmasında kullanılan minimum destek değeri, bu başlık altındaki giriş, işlemci yapısı ve fonksiyonları konularında 12 değeri (%12), bellek yapıları konusunda ise, minimum destek değeri 11 (%11) olarak belirlenmiştir. Veriyolu yapıları, komut düzeyinde paralellik, komut takımı ve adresleme modları konularında ise, öğrencilerin yanlış yaptıkları soru sayısı daha fazla olduğu için, destek değeri yükselterek daha güçlü kurallar oluşturulmuştur. Veriyolu yapıları ve komut düzeyinde paralellik konusunda, destek değeri 21 (%20), komut takımı ve adresleme modları konularında ise destek değeri 30 (%29) olarak belirlenmiştir.

VI.2.1.1. Giriş Konusu İçin Bulunan Birliktelik Kuralları

ZÖS’de ilk konu olan, giriş konusundaki sınavda toplam yedi soru bulunmaktadır ve bulunan birliktelik kuralları, Şekil VI.1’de görülmektedir. Bu ekranda, bir metin kutusu içerisinde istenilen minimum destek değeri yazılmakta ve “Kuralları Bul” komut düğmesi kullanılarak, birliktelik kurallarının bulunması sağlanmaktadır. Veri inceleme komut düğmesi ise, yazılımda kullanılan ve metin dosyasına dönüştürülmüş veri setinin, dosyadan okunup ekranda görüntülenmesi için kullanılmaktadır.

Birliktelik kuralları bulunan konuya ait, tüm soruların incelenmesi için, “Soruları İncele” komut düğmesi kullanılır. Bu şekilde, öğretmenin karşısına konuyla ilgili tüm sorular ayrı bir form ekranında gösterilir. Sadece analiz sonucunda çıkan soruların incelenmesi için; kural, “analizi yapılabilecek kurallar” kısmından seçildikten sonra, “kurala ait soruları incele” komut düğmesinin kullanılması gerekmektedir.

The screenshot shows a software window titled "GİRİŞ KONUSU". It features a "Minimum Destek=" input field with the value "12". Below this are three columns of results for different rule lengths: "Kural Uzunluğu=1", "Kural Uzunluğu=2", and "Kural Uzunluğu=3". Each column lists rules and their corresponding "Tekerar" (repetition) values. On the right, there is a section titled "ANALİZİ YAPILABİLECEK KURALLAR" containing a list of rules: "1 2 6" and "1 2 7". At the bottom right, there are three buttons: "Güven Değeri Hesapla", "Soruları İncele", and "Kurala Ait Soruları İncele".

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 5	Kurala uyanlar = 6	Kurala uyanlar = 2
1 Tekerar= 20	1 2 Tekerar= 18	1 2 6 Tekerar= 13
2 Tekerar= 29	1 6 Tekerar= 13	1 2 7 Tekerar= 12
4 Tekerar= 15	1 7 Tekerar= 13	
6 Tekerar= 43	2 6 Tekerar= 16	
7 Tekerar= 42	2 7 Tekerar= 18	
	6 7 Tekerar= 20	

Şekil VI.1 Giriş konusu için bulunan birliktelik kuralları

Şekil VI.1’de görüldüğü gibi, bulunan birliktelik kuralları kural uzunluğuna bağlı olarak farklı alanlarda bulunmaktadır. Örneğin, ilk sütunda, kural uzunluğu bir olan kurallar görülmektedir. Her kural alanının sol tarafında, minimum destek değerini sağlayan soru numaraları görülmektedir. Soru numarasının yanında ise, veri setinde bu soruyu yanlış yanıtlayan öğrenci sayısı, yani kuralın tekrar değeri bulunmaktadır. Örneğin, kural uzunluğu bir olan ilk sütunda, altıncı sorunun en fazla yanlış yanıtlanan soru olduğu ve bu sorunun toplamda 43 öğrenci tarafından yanlış yanıtladığı görülmektedir. ZEBİKA yazılımı giriş konusu için, minimum destek değeri 12 ile, kural uzunluğu üç olan iki adet kural bulmuştur. Bu kurallar; {1 2 6} ve {1 2 7}’dir. Bu kurallardaki her rakam soru numarasını temsil etmektedir. Örneğin, {1 2 6} kuralı Soru 1, Soru 2 ve Soru 6 arasındaki kuralı temsil etmek için kullanılmaktadır. Ekranda çok fazla yer gerektireceği için, kurallarda Soru kelimesi kullanılmadan doğrudan soru numarası yazılmıştır. Verilen minimum destek değerine göre; giriş konusunda, en fazla uzunluğu üç olan birliktelik kuralları bulunmuştur. Tablo VI.1’de bulunan birliktelik kurallarında yer alan 1, 2, 6 ve 7 numaralı çoktan seçmeli soruların içeriği ve doğru yanıt seçenekleri görülmektedir.

Tablo VI.1 Giriş konusunda bulunan birliktelik kurallarına ait sorular

1) Hangisi bilgisayar organizasyonunu en iyi tanımlar?
Doğru Yanıt) Bilgisayar ve arabirimleri arasındaki bağlantıları ve ünitelerin yapılarını
2.) Hangisi bilgisayar yapısını en iyi tanımlar?
Doğru Yanıt) Bilgisayarın programcıya görünen yapısal özelliklerini açıklar.
6.) Operandları ve ALU da elde edilen sonuçları geçici olarak hangisi tutar?
Doğru Yanıt) Akümülatör
7.) Bilgisayar performansını artırmada hangisi çok etkili değildir?
Doğru Yanıt) İşlemci ve ana bellek arasındaki veriyolu hızını azaltmak

Oluşturulan bu kuralların, güven değerlerinin analiz edilmesi mümkündür. Hangi kural analiz edilmek isteniyorsa, Şekil VI.1’de “Analizi Yapılabilecek Kurallar” kısmından seçilmesi gerekmektedir. Örneğin, Soru 1, Soru 2 ve Soru 6 arasındaki birliktelik kuralının güven değerini bulmak için, Şekil VI.1’deki ZEBİKA ekranında sağ tarafta bulunan “Analizi Yapılabilecek Kurallar” kısmından “1 2 6” seçilip, “Güven Değeri Hesapla” komut düğmesine basılır. Bu işlem sonucunda, Şekil VI.2’de görülen, Soru 1, Soru 2 ve Soru 6 arasındaki birliktelik kuralına ait, hesaplanan güven değerleri görüntülenir. Güven değeri analizi, bir kuralın ne kadar güçlü ve güvenilir olduğunun belirlenmesi için yapılan analizdir. Yüksek güven değerine sahip kurallar, “güçlü kural” olarak adlandırılır [61].



Şekil VI.2 Giriş konusu güven değeri hesaplaması için analiz ekranı

Şekil VI.2’ye göre, 2 6=>1=%81 satırı, ikinci soruyu ve altıncı soruyu yanlış yanıtlayan öğrencinin %81’lik bir güven değeri ile, yani %81 olasılıkla Soru 1’i de yanlış yanıtladığını ifade etmektedir. İkinci satırda bulunan 1 6=>2=%100 ise, birinci ve altıncı soruyu yanlış yanıtlayan bir öğrencinin %100’lük bir güven değeri ile, ikinci soruyu da yanlış yanıtladığını göstermektedir.

VI.2.1.2. Veriyolu Yapıları Konusu İçin Bulunan Birliktelik Kuralları

Tasarlanan ZÖS’de, veriyolu yapıları konusuna ait, toplam sekiz adet soru bulunmaktadır. ZEBİKA yazılımı ile, veriyolu yapıları konusunda, öğrencilerin yanlış yaptıkları sorulara ait bulunan birliktelik kuralı sonuçları, Şekil VI.3’de görülmektedir. Bu analizde, minimum destek değeri 21 ile, kural uzunluğu üç olan {2 3 7}, {3 5 7} kurallarını bulunmuştur.

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 6	Kurala uyanlar = 11	Kurala uyanlar = 2
2 Tekrar= 48	2 3 Tekrar= 34	2 3 7 Tekrar= 21
3 Tekrar= 68	2 5 Tekrar= 23	3 5 7 Tekrar= 22
4 Tekrar= 28	2 7 Tekrar= 25	
5 Tekrar= 49	2 8 Tekrar= 23	
7 Tekrar= 47	3 4 Tekrar= 25	
8 Tekrar= 45	3 5 Tekrar= 39	
	3 7 Tekrar= 38	
	3 8 Tekrar= 34	
	5 7 Tekrar= 26	
	5 8 Tekrar= 27	
	7 8 Tekrar= 25	

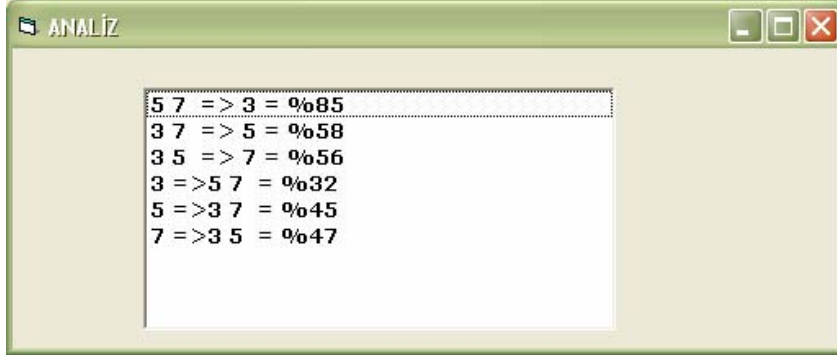
Şekil VI.3 Veriyolu yapıları konusu için bulunan birliktelik kuralları

Tablo VI.2’de 2, 3, 5 ve 7 numaralı çoktan seçmeli soruların içeriği ve doğru yanıt seçenekleri görülmektedir.

Programda çıkan {3,5,7} kuralı için, “Güven Değeri Hesapla” komut düğmesine basılarak yapılan güven değeri analizinde ise, Şekil VI.4’deki sonuç görülür. Örneğin; ilk satırda görüldüğü gibi, Soru 5 ve Soru 7’yi yanlış yanıtlayan öğrenci, %85’lik bir güven değeri ile Soru 3’ü de yanlış yanıtlamıştır.

Tablo VI.2 Veriyolu Yapıları konusunda bulunan birliktelik kurallarına ait sorular

2.) PCI express veriyolu yapısı için hangisi yanlıştır?
Doğru Yanıt) Veri hattı sayısı 32 bittir,
3.) PCI express veriyolu yapısında veri hattı sayısı kaçtır?
Doğru Yanıt) 128 Bit
5.) ISA veriyolu yapısında kaç bitlik veriler taşınabilir.
Doğru Yanıt) 8 Bit
7.) Bilgisayar veriyolu sisteminde aşağıdaki hatlardan hangisi bulunmaz?
Doğru Yanıt) Paralel Hatlar

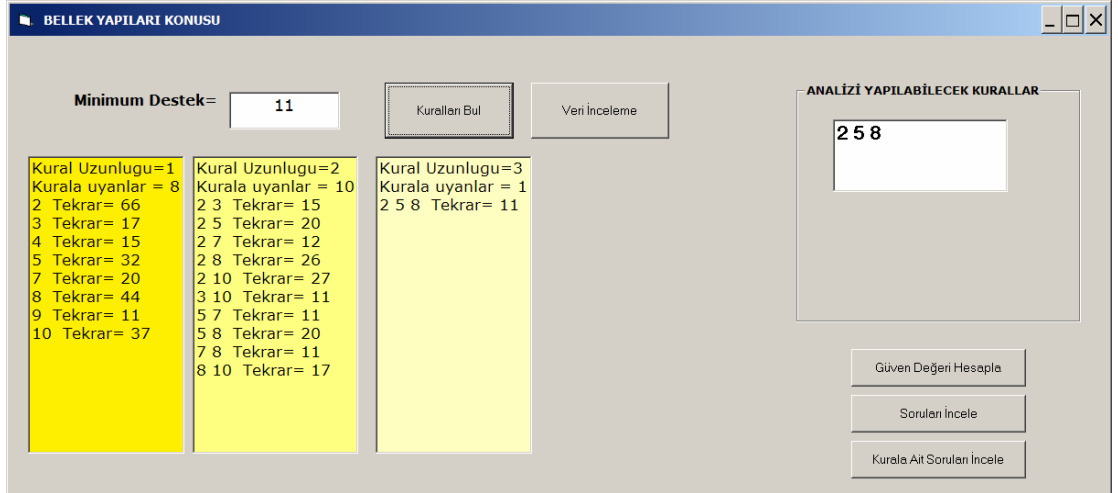


Şekil VI.4 Veriyolu yapıları konusu için güven değeri analizi

VI.2.1.3. Bellek Yapıları Konusu İçin Bulunan Birliktelik Kuralları

Toplamda on adet soru içeren, bellek yapıları konusuna ait birliktelik kuralı sonuçları Şekil VI.5’de görülmektedir. ZEBİKA yazılımı bellek yapıları konusu için, minimum destek değeri 11 ile, uzunluğu üç olan bir adet kural bulmuştur. Bulunan {2,5,8} kuralına ait tekrar değeri 11’dir.

Tablo VI.3’de 2, 5 ve 8 numaralı çoktan seçmeli soruların içeriği ve doğru yanıt seçenekleri görülmektedir.

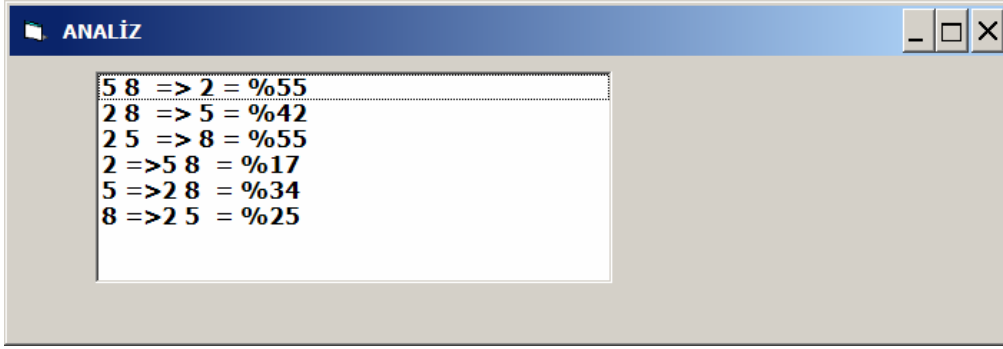


Şekil VI.5 Bellek yapıları konusu için bulunan birliktelik kuralları

Tablo VI.3 Bellek Yapıları konusunda bulunan birliktelik kurallarına ait sorular

2.) Ana bellekte okuma ya da yazma süresi hangisidir?
Doğru Yanıt) Erişim süresi
5.) Ön belleklerde satır sayısı çok büyükse etkisi ne olur?
Doğru Yanıt) c.) Aranan bilgiye ulaşma süresi artar, bu performansı azaltır,
8.) DDR-RAM’ in hangi özelliği yoktur?
Doğru Yanıt) Kendi içindeki sinyal üreticindeki saat sinyalini kullanır,

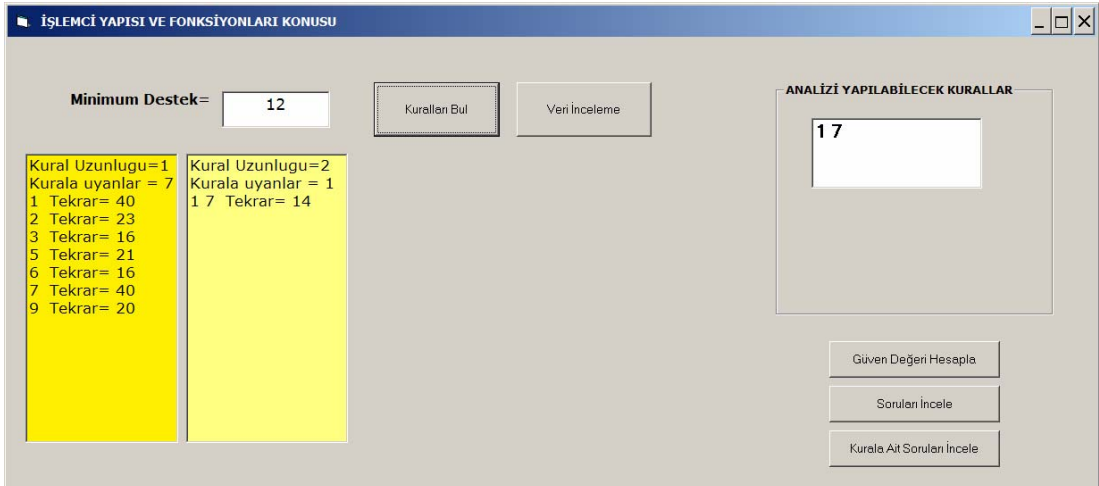
{2,5,8} kuralına ait güven değeri analizi, Şekil VI.6'da görülmektedir. Bu güven değerleri içerisinde, en yüksek değere sahip olan güçlü kurallardan 2 5 => 8 kuralı, %55'lik bir güven değerine sahiptir ve Soru 2 ve Soru 5'i yanlış yapan bir öğrencinin %55'lik bir güven değeri ile, Soru 8' yi de yanlış yanıtladığını göstermektedir.



Şekil VI.6 Bellek yapıları konusu için güven değeri analizi

VI.2.1.4. İşlemci Yapısı ve Fonksiyonları Konusu İçin Bulunan Birliktelik Kuralları

Tasarlanan ZÖS'de işlemci yapısı ve fonksiyonları konusu için toplam dokuz adet soru bulunmaktadır. ZEBİKA yazılımı bu konu için Şekil VI.7'de görüldüğü gibi, minimum destek değeri 12 ile, kural uzunluğu iki olan bir tane birliktelik kuralı bulmuştur.



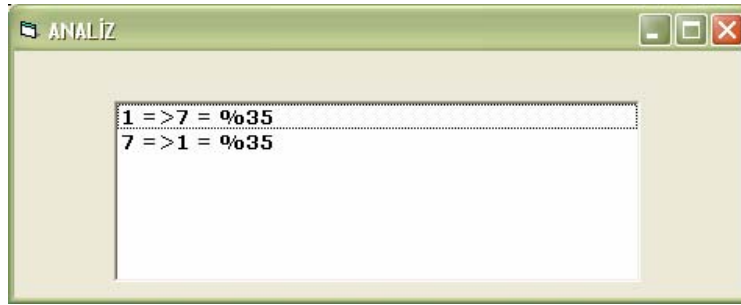
Şekil VI.7 İşlemci yapısı ve fonksiyonları konusu için bulunan birliktelik kuralları

Bulunan {1,7} kuralına ait, birinci ve yedinci soru içerikleri ve doğru yanıtları Tablo VI.4'de görülmektedir.

Bu kuralın güven değeri analizine ilişkin sonuçlar, Şekil VI.8’de görülmektedir. Bu analize göre; birinci soruyu yanlış yanıtlayan bir öğrenci %35’lik bir güven değeri ile, yedinci soruyu da yanlış yanıtlamıştır. Yedinci soruyu yanlış yanıtlayan bir öğrenci ise, yine %35’lik bir güven değeri ile birinci soruyu yanlış yanıtlamıştır sonucu ortaya çıkmıştır.

Tablo VI.4 İşlemci yapısı ve fonksiyonları konusunda bulunan birliktelik kurallarına ait sorular

1.) İşlemcide kullanıcı tarafından görülmeyen register hangisidir?
Doğru Yanıt)Yukarıdakilerin hiç biri
7.) İş hattı(Pipeline) yapısına sahip bir işlemcide potansiyel olarak etkinliği artırmada hangi faktör önemli rol oynar?
Doğru Yanıt) İş hattındaki ünite sayısı artırılmalıdır.



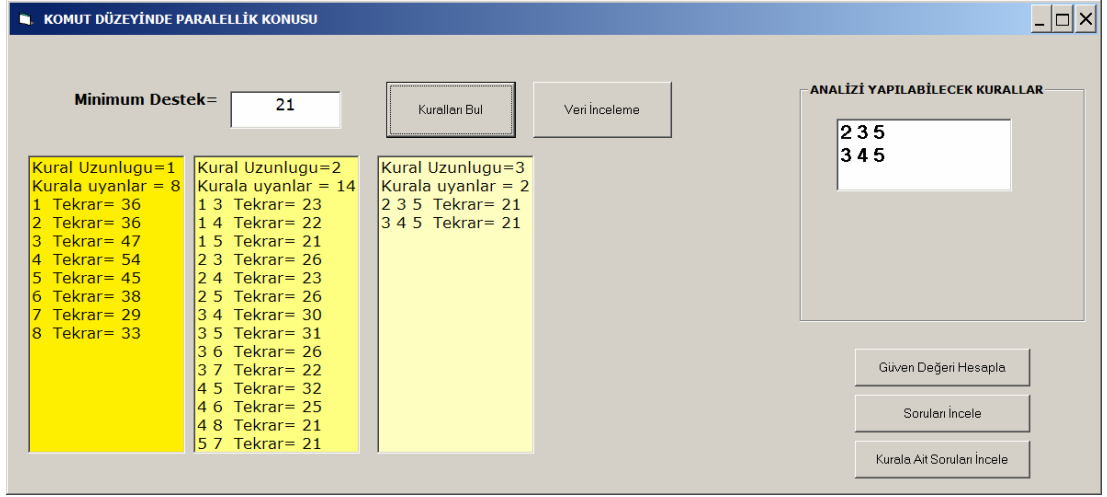
1 =>7 = %035
7 =>1 = %035

Şekil VI.8 İşlemci yapısı ve fonksiyonları konusu için güven değeri analizi

VI.2.1.5. Komut Düzeyinde Paralellik Konusu İçin Bulunan Birliktelik Kuralları

Tasarlanan ZÖS’de komut düzeyinde paralellik konusu için, toplam sekiz soru bulunmaktadır. Bu konu içerisinde, minimum destek değeri 21 ile bulunan kurallar Şekil VI.9’da görülmektedir. ZEBİKA yazılımı komut düzeyinde paralellik konusu için, kural uzunluğu üç olan {2 3 5} ve { 3 4 5} kullarını bulmuştur.

Tablo VI.5’de komut düzeyinde paralellik konusunda bulunan 2, 3, 4 ve 5 numaralı çoktan seçmeli soruların içeriği ve doğru yanıt seçenekleri görülmektedir.

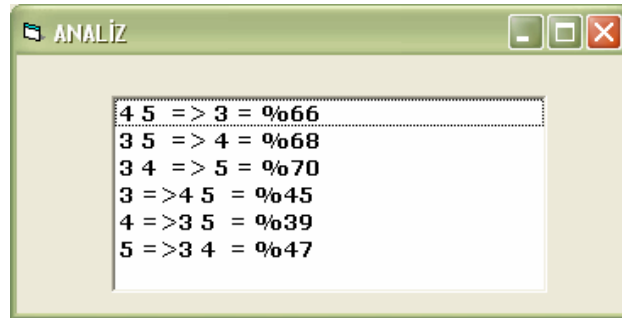


Şekil VI.9 Komut düzeyinde paralellik konusu için bulunan birliktelik kuralları

Tablo VI.5 Komut düzeyinde paralellik konusunda bulunan birliktelik kurallarına ait sorular

2.) Super işhatlı(super pipelined) işlemciadaki en önemli özellik hangisidir?
Doğru Yanıt) Komutlar iş hattında her yarım saat saykılında bir diğer üniteye geçer.
3.)Komut düzeyinde paralellikte aşağıdaki komutlarda hangi sınırlık görülür? Add r1, r2 Move r3,r1
Doğru Yanıt) Gerçek veri bağımlılığı
4.) Komut düzeyinde paralellikte aşağıdaki komutlarda hangi sınırlık görülür? Add r1, r2 Move r3,r2
Doğru Yanıt) Kaynak Çakışması.
5.) Komut düzeyinde paralellik de dallanma komutu ile hangi sınırlık görülür?
Doğru Yanıt) İşlemsel bağımlılık.

Bulunan kurallardan {3 4 5} ile yapılan güven değeri analizinde ise, Şekil VI.10'daki sonuçlar elde edilmiştir. Örneğin, analiz sonucunda ortaya çıkan 3 4=>5 kuralı, üçüncü ve dördüncü soruyu yanlış yapan bir öğrencinin, %70'lik bir güven değeri ile beşinci soruyu da yanlış yaptığını göstermektedir.



Şekil VI.10 Komut düzeyinde paralellik konusu için güven değeri analizi

VI.2.1.6. Komut Takımı ve Adresleme Modları Konusu İçin Bulunan Birliktelik Kuralları

Komut takımı ve adresleme modları için, ZÖS’de 11 adet soru bulunmaktadır. Bu konudaki sorularla ilgili yapılan analizde, öğrenciler tarafından yapılan hatalar daha fazladır ve bu nedenle %12 gibi düşük minimum destek değeri kullanıldığında çok fazla kural ortaya çıkmaktadır. ZEBİKA yazılımında oluşturulan kural sayısını indirmek ve daha güçlü kurallar oluşturmak amacıyla, bu konuda minimum destek değeri, 30 (%29) olarak kullanıldı. Şekil VI.11’de görüldüğü gibi, bu destek değeri ile kural uzunluğu üç olan {4 6 9} kuralı bulunmuştur.

Bu konu için bulunan kuralda yer alan dördüncü, altıncı ve dokuzuncu sorular ve doğru yanıtları Tablo VI.6’da görülmektedir.

KOMUT TAKIMI VE ADRESLEME MODLARI KONUSU

Minimum Destek= 30

Kuralları Bul

Veri İnceleme

ANALİZİ YAPILABİLECEK KURALLAR

4 6 9

Güven Değeri Hesapla

Soruları İncele

Kuralları Ait Soruları İncele

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 8	Kurala uyanlar = 5	Kurala uyanlar = 1
2 Tekrar= 31	4 6 Tekrar= 40	4 6 9 Tekrar= 31
3 Tekrar= 32	4 9 Tekrar= 34	
4 Tekrar= 44	6 9 Tekrar= 54	
5 Tekrar= 32	6 11 Tekrar= 36	
6 Tekrar= 72	9 11 Tekrar= 32	
9 Tekrar= 65		
10 Tekrar= 32		
11 Tekrar= 40		

Şekil VI.11 Komut takımı ve adresleme modları konusu için bulunan birliktelik kuralları

Tablo VI.6 Komut takımı ve adresleme modları konusunda bulunan birliktelik kurallarına ait sorular

4.) “ADD C” komutunda kaç adet adres kullanılmaktadır?
Doğru Yanıt) Bir adres vardır
6.) Komut takımında hangisi operand olarak kullanılmaz?
Doğru Yanıt) Register
9.) Kontrol registerlerindeki bilgiyi okuyan ya da değiştiren komutlar hangisidir?
Doğru Yanıt) Sistem kontrol

{4 6 9} kuralına ait güven değeri analizi Şekil VI.12’de görülmektedir. Yapılan güven değeri analizine göre; dördüncü ve altıncı soruları yanlış yanıtlayan bir öğrenci %78’lik bir güven değeri ile, dokuzuncu soruyu da yanlış yanıtlamıştır.

Konu Numarası	Güven Değeri
69 => 4	%57
49 => 6	%91
46 => 9	%78
4 => 69	%70
6 => 49	%43
9 => 46	%48

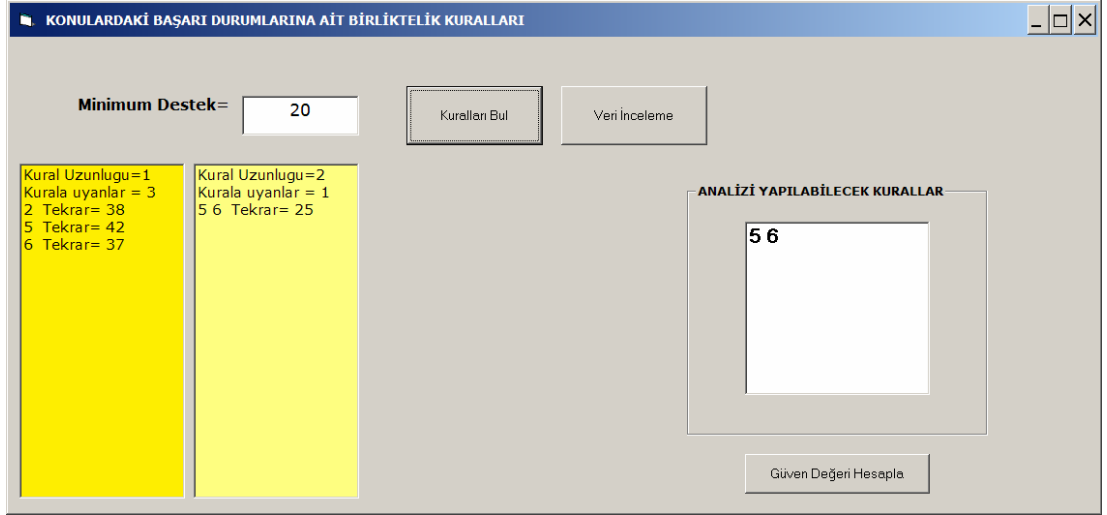
Şekil VI.12 Komut takımı ve adresleme modları konusu için yapılan güven değeri analizi

VI.2.2. Konulardaki Başarı Durumlarına İlişkin Bulunan Birliktelik Kuralları

Bir konudan başarısız olan öğrencilerin başka hangi konulardan başarısız olduklarının bulunması amaçlanmıştır. Bu amaçla, öğrencinin bir konudaki başarı durumunu belirleyen ve nasıl elde edildiği Bölüm V.2.4’de açıklanan “durum” bilgisi kullanılmıştır. Tüm öğrenci verileri dikkate alınarak yapılan analizde, Şekil VI.13’de görülen birliktelik kuralları bulunmuştur. Bu ekranda, her sütunda yer alan ilk rakam, konu numarasını göstermektedir. Konu numaraları ise, aşağıdaki listede belirtilen konuları temsil etmektedir.

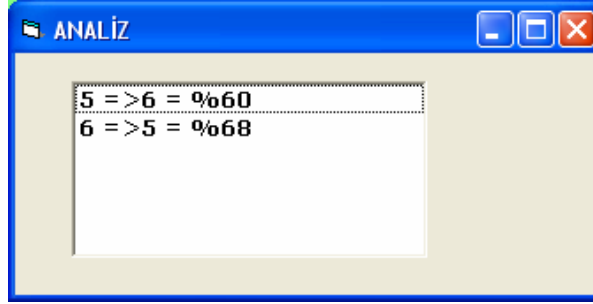
- 1) Giriş
- 2) Veriyolu yapıları
- 3) Bellek yapıları
- 4) İşlemci yapısı ve fonksiyonları
- 5) Komut düzeyinde paralellik
- 6) Komut takımı ve adresleme modları

Konu numarasının yanındaki tekrar değeri ise, konudan kaç öğrencinin başarısız olduğunu göstermektedir. Örneğin, Şekil VI.13’de ilk sütunda, öğrencilerin en fazla başarısız oldukları konunun, 5 sayısı ile temsil edilen komut düzeyinde paralellik konusu olduğu görülmektedir, bu konudan toplam 42 öğrenci başarısız olmuştur. ZEBİKA yazılımı, minimum destek değeri 20 ile, uzunluğu iki olan {5 6} kuralını bularak işlemini tamamlamıştır. Bu kural, beş ve altıncı konular olan, komut düzeyinde paralellik ve komut takımı-adresleme modları konularında birliktelik olduğunu göstermektedir. Yani, komut düzeyinde paralellik konusunda başarısız olan öğrenciler, komut takımı-adresleme modları konusunda da başarısız olmaktadır.



Şekil VI.13 Konulardaki başarı durumlarına ait bulunan birliktelik kuralları

Bulunan birliktelik kuralına ait yapılan güven değeri analizinde ise, Şekil VI.14’de görülen sonuçlar ortaya çıkmıştır. Buna göre, komut düzeyinde paralellik konusunda başarısız olan bir öğrenci, %60’lık bir güven değeri ile komut-takımı-adresleme modları konusundan da başarısız olmuştur.



Şekil VI.14 Konulardaki başarı durumlarına ait güven değeri analizi

VI.2.3. Tüm Konularda Yer Alan Sorular Arasında Bulunan Çok Boyutlu Birliktelik Kuralları

ZÖS’de yer alan altı adet konu içerisinde yer alan sorular arasındaki birliktelik kurallarını bulmak için, ZECOKA yazılımı kullanılmıştır. Bu program ile, tüm konularda yer alan sorularla ilgili, başarı veya başarısızlık ilişkisini gösteren çok boyutlu birliktelik kuralları bulunmaktadır.

ZECOKA’da minimum destek değeri 30 (%29) olarak belirlendiğinde, bulunan kurallar ve tekrar değerleri Şekil VI.15’de görülmektedir. Şekil VI.15’de, ekranın üst kısmında yer alan altı adet sütunda, ZÖS’de yer alan konu isimleri ve verilen minimum destek değerine göre, her konudan minimum destek değerini aşan soru

numaraları ile birlikte tekrar değerleri görülmektedir. Örneğin, giriş konusundan, Soru 6 minimum destek değerini aşmıştır ve toplamda 43 öğrenci tarafından yanlış yanıtlanmıştır.

Ekranın alt kısmında ise, ikinci boyuttan başlayarak altıncı boyuta kadar devam eden ve bulunan çok boyutlu birliktelik kurallarını içeren alanlar bulunmaktadır. Bu alanlarda, öncelikle bulunan kuralın hangi konuya ait olduğunu belirten sayısal değer, daha sonra soru numarası, son olarak da kuralın tekrar değeri bulunmaktadır. Örneğin, iki boyutlu kuralların bulunduğu alandaki ikinci kural K:13 S:62 T=31 olarak görülmektedir. Kuralda, K harfinin yanında yer alan rakamlar ile kuralın ait olduğu konu numaraları belirtilmektedir. K:13 ile bulunan kuralın 1 ve 3 numaralı konulara, yani giriş konusu ile bellek konusuna ait olduğu belirtilmiştir. S:62 değeri, kuraldaki soru numaralarını, T=31 değeri ise, belirtilen kuralın tekrar değerini temsil etmektedir. K:13 S:62 T=31 kuralı, giriş konusundaki (1 numaralı konu) Soru 6 ile, bellek yapıları konusundaki (3 numaralı konu) Soru 2 arasındaki birliktelik kuralını göstermektedir ve bu kuralın tekrar değeri 31’dir.

Şekil VI.15 ZECOKA ile bulunan çok boyutlu birliktelik kuralları

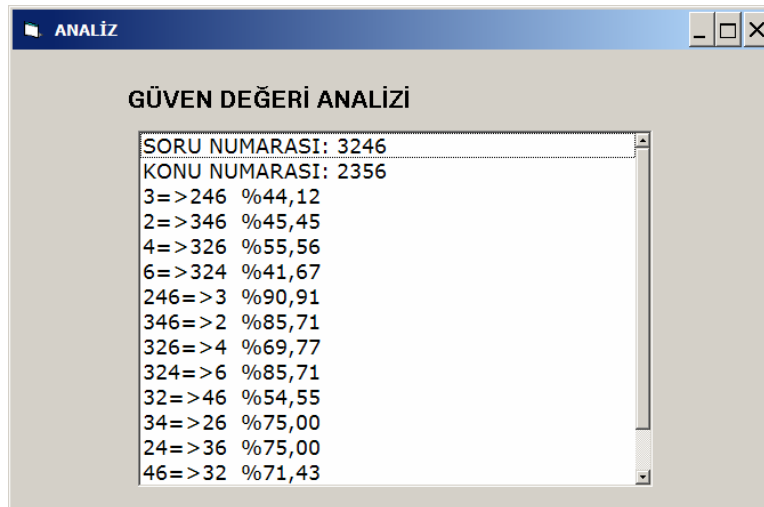
Program, verilen minimum destek değeri ile, 2356 numaralı konular içinde yer alan, 3246 numaralı sorulara ait, dört boyutlu kuralı bularak son bulmuştur. Kuralda yer alan sorular ve doğru yanıtları Tablo VI.7’de görülmektedir.

Tablo VI.7 Çok boyutlu birliktelik kuralında yer alan sorular

SORU	AİT OLDUĞU KONU ve KONU NUMARASI
3.) PCI express veriyolu yapısında veri hattı sayısı kaçtır?	Veriyolu yapıları konusu
Doğru Yanıt)128 bit	Konu No: 2
2.) Ana bellekte okuma ya da yazma süresi hangisidir?	Bellek yapıları konusu
Doğru Yanıt)Erişim süresi	Konu No: 3
4.) Komut düzeyinde paralellikte aşağıdaki komutlarda hangi sınırlık görülür? Add r1, r2 Move r3,r2	Komut düzeyinde paralellik konusu
Doğru Yanıt) Kaynak Çakışması.	Konu No: 5
6.) Komut takımında hangisi operand olarak kullanılmaz?	Komut takımı ve adresleme modları konusu
Doğru Yanıt) Register	Konu No: 6

Yapılan güven değeri analizine ait sonuçlar ise, Şekil VI.16’da görülmektedir. Bu sonuçlara göre; örneğin 324=>6 kuralı, veriyolu yapıları konusundaki 3. soruya, bellek yapıları konusundaki 2.soruya ve komut düzeyinde paralellik konusundaki 4. soruya yanlış yanıt veren öğrencinin %85,71’lik bir güven değeri ile, komut takımı ve adresleme modları konusundaki 6. soruya da yanlış yanıt verdiğini göstermektedir.

34=>26 kuralı ise, veriyolu yapıları konusundaki 3. soruya ve komut düzeyinde paralellik konusundaki 4. soruya yanlış yanıt veren bir öğrencinin %75’lik bir güven değeri ile, bellek yapıları konusundaki 2.soruya ve komut takımı ve adresleme modları konusundaki 6. soruya da yanlış yanıt verdiğini göstermektedir.



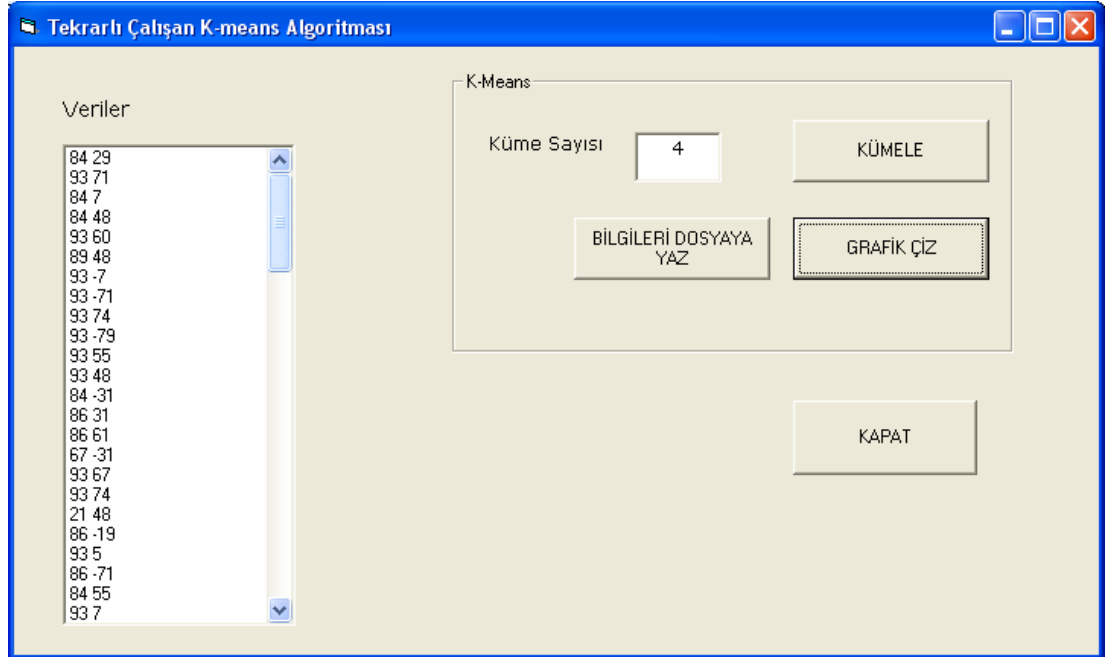
GÜVEN DEĞERİ ANALİZİ	
SORU NUMARASI:	3246
KONU NUMARASI:	2356
3=>246	%44,12
2=>346	%45,45
4=>326	%55,56
6=>324	%41,67
246=>3	%90,91
346=>2	%85,71
326=>4	%69,77
324=>6	%85,71
32=>46	%54,55
34=>26	%75,00
24=>36	%75,00
46=>32	%71,43

Şekil VI.16 Çok boyutlu birliktelik kuralına ait güven değeri analizi

VI.3. TM đRENCİLERE AİT VERİDE KMELEME ANALİZİ SONUÇLARI

Kmeleme analizinde, đrencilerin belirlenen konulardan aldıkları başarı puanları (MYCIN gven faktr) kullanılarak, kmeleme analizi yapılması amaçlanmıştır. Kmeleme analizi sonucunda, belirlenen kme sayısına gre, đrencilerin başarı puanlarıyla benzer kmelerinin oluřması sađlanır ve sınıftaki başarı dađılımının ne řekilde olduđuna dair bir analiz yapılır.

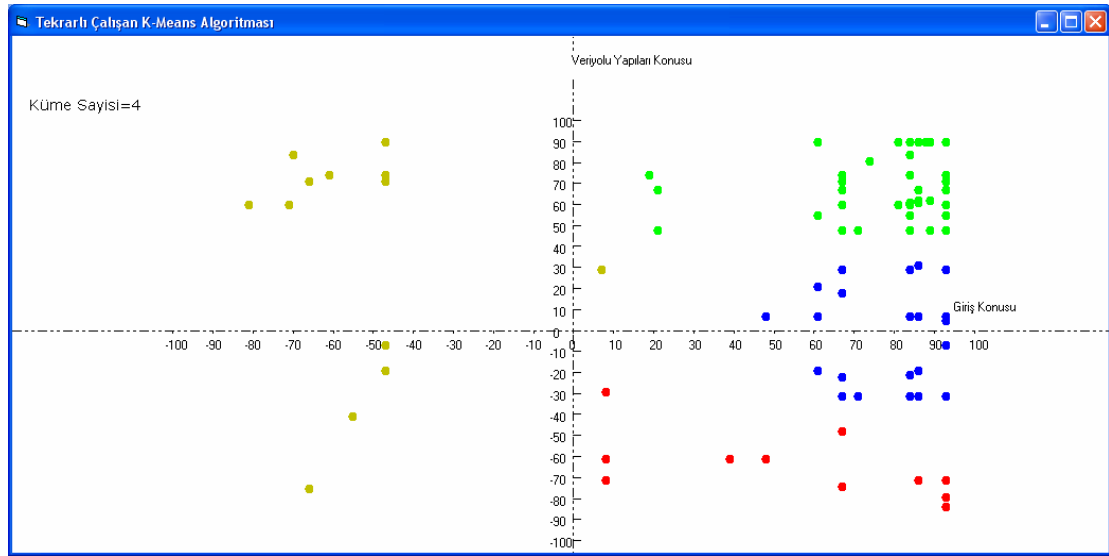
Kmeleme iřlemine bařlamadan nce, hangi konular iin kmeleme yapılacađı belirlenerek, đrencilerin başarı puanlarına ait veri setinin oluřturulması gerekmektedir. Daha sonra ise, řekil VI.17’de grlen kmeleme programı alıřtırılarak, program sonucunda oluřturulması istenen kme sayısı deđerinin ‘‘Kme Sayısı’’ alanına girilmesi gerekmektedir. K-means algoritmasına gre, kmeleme iřleminin yapılması iin ise, ‘‘Kmele’’ komut dđmesi kullanılır. Kmeleme iřleminde kullanılan veri deđerleri ise, ‘‘Veriler’’ alanında grlmektedir. Kmeleme analizi yapılan veri ile birlikte, verinin hangi kmeye ait olduđunu belirten kme etiketinden oluřan deđerlerin ayrı bir dosyaya kaydedilmesi iřlemi iin, ‘‘Bilgileri Dosyaya Yaz’’ komut dđmesi kullanılmaktadır. Bu komut dđmesi, veriyi ‘‘sonuc.txt’’ adında yeni bir dosyaya kaydetmektedir.



řekil VI.17 Kmeleme analizi programına ait ekran grnts

Kümeleme analizi yapılacak konu sayısı iki ise, program içerisinde “Grafik Çiz” komut düğmesi kullanılarak, analiz sonucu iki boyutlu eksen üzerine görüntülenir. Analizi yapılacak konu sayısı ikiden fazla olduğunda ise, kümeleme sonuçlarının paralel koordinatlar yardımıyla görüntülenmesi için Bölüm VII.5’de açıklanan Parvis yazılımı kullanılmaktadır.

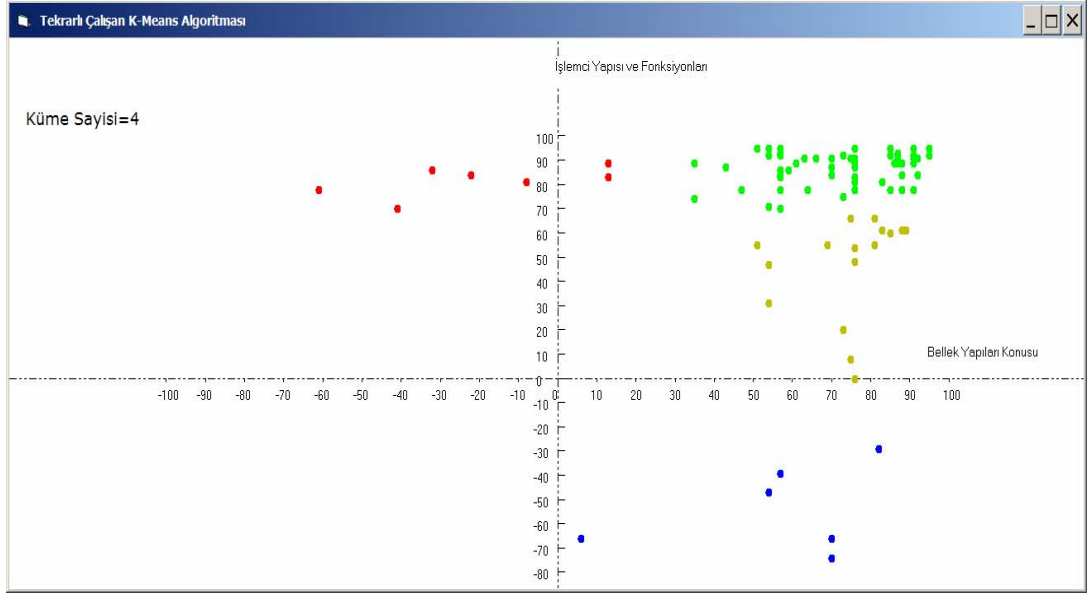
“Grafik Çiz” komut düğmesi kullanıldığında, Şekil VI.18’deki ekranda görüldüğü gibi, yatay eksen giriş konusu, dikey eksen veriyolu konusu için kullanılan iki boyutlu düzlemde, her veri ait olduğu kümeye bağlı olarak farklı renkte bir nokta ile işaretlenir. Şekil VI.18’deki ekranda, öğrencilerin giriş ve veriyolu yapıları konusundan aldıkları başarı puanlarından oluşturulan, dört küme farklı renklerle temsil edilmiştir.



Şekil VI.18 Giriş ve veriyolu yapıları konusuna ait kümeleme analizi sonuçları

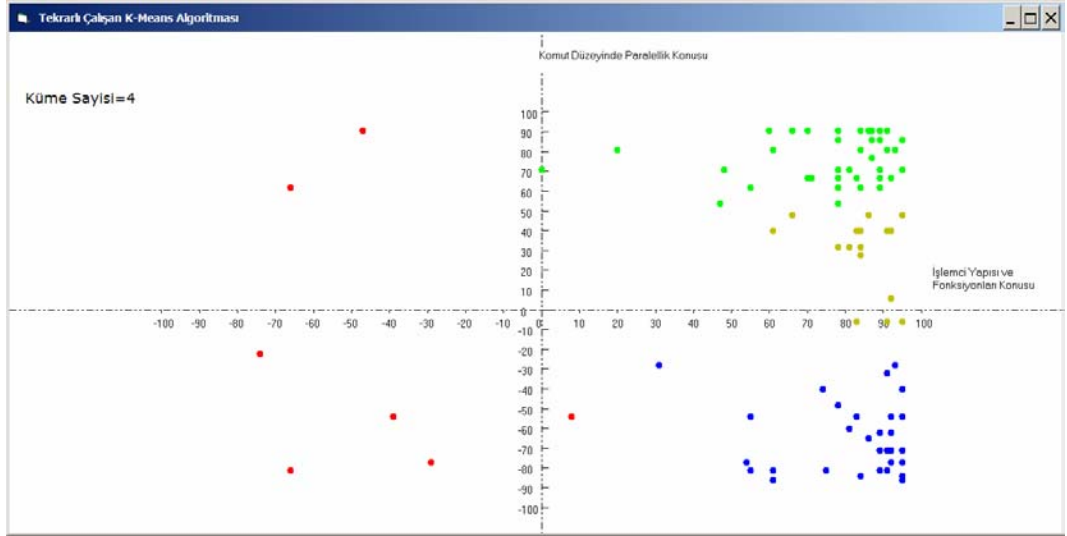
Şekil VI.18’de görülen kümeleme analizine göre, en büyük küme yeşil renk ile temsil edilen kümedir. Bu kümede, yoğunlukla giriş ve veriyolu yapıları konusundan başarılı olan öğrenciler bulunmaktadır.

Bellek yapıları konusu ile işlemci yapısı ve fonksiyonları konusuna ait veri ile, yapılan analizde, küme sayısı 4 olarak belirlendiğinde, alınan sonuçlar Şekil VI.19’da görülmektedir.



Şekil VI.19 Bellek yapıları ile işlemci yapısı ve fonksiyonları konularına ait kümeleme analizi sonuçları

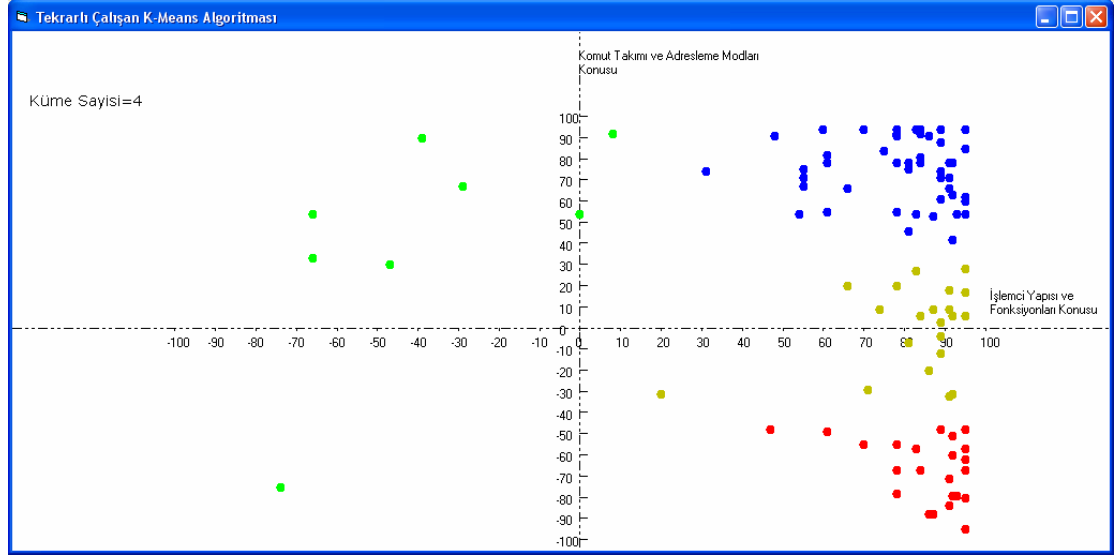
İşlemci yapısı ve komut düzeyinde paralellik konusu için küme sayısı 4 olarak belirlendiğinde, alınan sonuçlar Şekil VI.20’de görülmektedir. En yoğun küme, genelde öğrencilerin iki konudan da başarılı olduklarını belirten yeşil renkli kümedir.



Şekil VI.20 İşlemci yapısı ile komut düzeyinde paralellik konusu için yapılan kümeleme analizi sonuçları

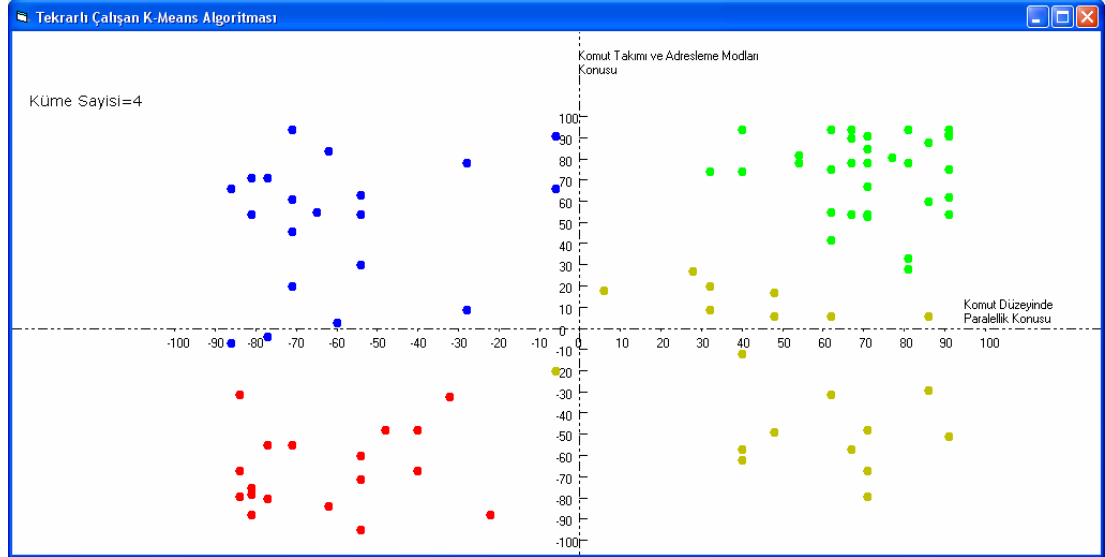
İşlemci yapısı ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizinde küme sayısı 4 olarak belirlendiğinde, bulunan kümeler Şekil VI.21’de görülmektedir. Bu analizde, en büyük kümeyi genelde her iki konudan da başarılı olan öğrenciler oluşturmuştur. Fakat, işlemci yapısı konusundan başarılı olan birçok öğrencinin komut takımı ve adresleme modları konusundan başarısız olduğu

da görülmekte ve oluşturulan sarı ve kırmızı kümeler bu öğrencileri temsil etmektedir.



Şekil VI.21 İşlemci yapısı ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizi sonuçları

Komut düzeyinde paralellik ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizi sonucu Şekil VI.22’de görülmektedir. Bu analizde küme sayısı 4 olarak belirlenmiştir.



Şekil VI.22 Komut düzeyinde paralellik ile komut takımı ve adresleme modları konusu için yapılan kümeleme analizi sonuçları

BÖLÜM VII

VERİ MADENCİLİĞİ ANALİZ SONUÇLARININ EĞİTİM AÇISINDAN DEĞERLENDİRİLMESİ

VII.1. AMAÇ

Bölüm VI’da öğrencilerin ZÖS’deki konu sonu sınavlarına verdikleri en son yanıtlara ait verinin tasarlanan yazılımlarda kullanılması ile bulunan birliktelik kuralları ve kümeleme analizi sonuçları açıklanmaktadır. Bu bölümde ise, bulunan birliktelik kuralları ve kümeleme analizi sonuçları değerlendirilmektedir. Birliktelik kuralı analizinde, öğrencilerin ZÖS’de konu sonu sınavına verdikleri ilk yanıtlara ait tek boyutlu birliktelik kuralı analizine ait sonuçlar ZEBİKA ile, çok boyutlu birliktelik kuralına ait sonuçlar ise ZECOKA ile bulunmaktadır. Bu sonuçlar, öğrencilerin en son yanıtlarına ait birliktelik kuralı sonuçlarıyla karşılaştırılmakta ve geliştirilen sistem öğretime katkısı açısından değerlendirilmektedir. Bu bölümde, konu sonu sınavında öğrencilerin verdikleri ilk yanıtları ifade etmek için, “birinci yanıtlar”, son yanıtlarını ifade etmek için ise, “ikinci yanıtlar” terimleri kullanılacaktır.

Kümeleme analizinde, tasarlanan kümeleme analizi yazılımı ile ZÖS’den öğrencilerin aldığı başarı puanlarından, belirlenen iki konuya ait başarı puanları ile yapılan analiz sonuçları açıklanmaktadır. Tasarlanan kümeleme analizi yazılımı ile, belirlenen üç veya daha fazla konuya ait başarı puanlarının çok boyutlu kümeleme analizi sonuçları ise, Bölüm III.4’de açıklanan Parvis yazılımı ile görselleştirilmiştir ve bu analize ait sonuçlar Bölüm VII.5’de açıklanmaktadır.

VII.2. TÜM ÖĞRENCİLERİN BİRİNCİ ve İKİNCİ YANITLARINA AİT BİRLİKTELİK KURALLARININ KARŞILAŞTIRILMASI

VII.2.1.1.Giriş Konusu İçin Bulunan Birliktelik Kuralları

ZÖS’de yer alan ilk konu olan giriş konusuna ait, toplamda 103 öğrencinin birinci yanıtlarına ait, minimum destek değeri 12 alındığında, uzunluğu üç olan {2, 6, 7} sorularına ait birliktelik kuralları Şekil VII.1’de, öğrencilerin ikinci yanıtlarına ait birliktelik kuralı analiz sonucu ise, Şekil VII.2’de görülmektedir. 31 öğrenci bu konuda başarısız olup, konuyu ikinci kez tekrar etmiştir. Bölüm VI.2.1.1’de açıklanan öğrencilerin ikinci yanıtlarında, aynı destek değeri ile yapılan birliktelik analizinde {1, 2, 6} ve {1, 2, 7} soruları arasında birliktelik kuralları bulunmuştur. Bu sonuçlara göre; kural uzunluğu bir olan ilk sütuna göre; sistemde yapılan konu tekrarı ikinci, altıncı ve yedinci soruların tekrar yanlış yapılmasında azalma sağlamıştır. Öğrencilerin verdiği birinci ve ikinci yanıtlarda, uzunluğu bir olan kurallar içerisinde, en fazla yanlış yanıtlanan soru altıncı soru, uzunluğu iki olan kurallarda birlikte en fazla yanlış yanıtlanan sorular ise altıncı ve yedinci sorulardır. Altıncı ve yedinci soru arasındaki birliktelik kuralı, her iki analizde de elde edildiği için dikkate alınması gereken bir kuraldır.

Genel olarak; birinci ve ikinci yanıtlardaki yanlış yapılma tekrar değerlerine göre, konu tekrarı sonucunda öğrencilerin başarısında bir artış olduğu görülmektedir.

The screenshot shows a software window titled "GİRİŞ KONUSU (BİRİNCİ YANITLAR)". It features a "Minimum Destek=" input field with the value "12". To the right are buttons for "Kuralları Bul" and "Veri İnceleme". On the right side, there is a section titled "ANALİZİ YAPILABİLECEK KURALLAR" with a text box containing "2 6 7". Below this are three buttons: "Güven Değeri Hesapla", "Soruları İncele", and "Kurallara Ait Soruları İncele". The main area displays three columns of results for different rule lengths:

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurallara uyanlar = 5	Kurallara uyanlar = 5	Kurallara uyanlar = 1
1 Tekrar= 18	1 2 Tekrar= 15	2 6 7 Tekrar= 13
2 Tekrar= 34	1 6 Tekrar= 12	
4 Tekrar= 13	2 6 Tekrar= 22	
6 Tekrar= 46	2 7 Tekrar= 20	
7 Tekrar= 44	6 7 Tekrar= 24	

Şekil VII.1 Giriş konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

GİRİŞ KONUSU

Minimum Destek=

Kural Uzunluğu=1 Kurala uyanlar = 5 1 Tekrar= 20 2 Tekrar= 29 4 Tekrar= 15 6 Tekrar= 43 7 Tekrar= 42	Kural Uzunluğu=2 Kurala uyanlar = 6 1 2 Tekrar= 18 1 6 Tekrar= 13 1 7 Tekrar= 13 2 6 Tekrar= 16 2 7 Tekrar= 18 6 7 Tekrar= 20	Kural Uzunluğu=3 Kurala uyanlar = 2 1 2 6 Tekrar= 13 1 2 7 Tekrar= 12
---	---	---

ANALİZİ YAPILABİLECEK KURALLAR

1 2 6
1 2 7

Şekil VII.2 Giriş konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.2.1.2. Veriyolu Yapıları Konusu İçin Bulunan Birliktelik Kuralları

Tüm öğrencilerin, ZÖS’deki veriyolu yapıları konusunda birinci yanıtları için, minimum destek değeri 21 alınarak yapılan birliktelik kuralı analizi Şekil VII.3’de, ikinci öğrenci yanıtları için yapılan birliktelik kuralı analizi ise, Şekil VII.4’de görülmektedir. Bu konuyu tekrar eden öğrenci sayısı 59’dur. Bu sonuçlar içerisinde uzunluğu bir olan ilk sütundaki sonuçlara göre, ikinci, üçüncü, dördüncü, yedinci ve sekizinci soruların tekrar yanlış yanıtlanma durumunda azalma olurken, beşinci sorunun tekrar yanlış yanıtlanma değeri değişmemiştir. Veriyolu yapıları konusunda, konu tekrarı öğrenci başarısını pozitif yönde etkilemiştir. Uzunluğu bir olan kurallara göre, her iki analiz sonucunda en fazla yanlış yanıtlanan sorunun üçüncü soru olduğu görülmektedir.

VERİYOLU YAPILARI KONUSU (BİRİNCİ YANITLAR)

Minimum Destek=

Kural Uzunluğu=1 Kurala uyanlar = 6 2 Tekrar= 50 3 Tekrar= 72 4 Tekrar= 30 5 Tekrar= 49 7 Tekrar= 55 8 Tekrar= 50	Kural Uzunluğu=2 Kurala uyanlar = 1: 2 3 Tekrar= 36 2 5 Tekrar= 23 2 7 Tekrar= 32 2 8 Tekrar= 27 3 4 Tekrar= 26 3 5 Tekrar= 38 3 7 Tekrar= 45 3 8 Tekrar= 40 4 7 Tekrar= 21 5 7 Tekrar= 29 5 8 Tekrar= 28 7 8 Tekrar= 35	Kural Uzunluğu=3 Kurala uyanlar = 6 2 3 7 Tekrar= 25 2 3 8 Tekrar= 22 2 7 8 Tekrar= 23 3 5 7 Tekrar= 23 3 5 8 Tekrar= 21 3 7 8 Tekrar= 28	Kural Uzunluğu=4 Kurala uyanlar = 0
---	--	---	---

ANALİZİ YAPILABİLECEK KURALLAR

2 3 7
2 3 8
2 7 8
3 5 7
3 5 8
3 7 8

Şekil VII.3 Veriyolu yapıları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

Öğrencilerin birinci yanıtlarında, uzunluğu iki olan kurallarda $\{3, 7\}$ kuralı en yüksek destek değerine sahiptir. $\{3, 7\}$ kuralı ikinci öğrenci yanıtlarında da 38 destek değeri ile ikinci en yüksek değere sahiptir. Her iki analizde de yüksek destek değeri ile ortaya çıkan bu kural dikkate alınır niteliktedir. Uzunluğu üç olan kurallarda ikinci öğrenci yanıtlarına ait veride bulunan $\{2, 3, 7\}$ ve $\{3, 5, 7\}$ kuralı ilk öğrenci yanıtlarına ait veride de bulunmuştur. Genel olarak her iki analizdeki sonuçlara bakıldığında, bulunan birliktelik kuralları benzerlik göstermektedir.

VERİYOLU YAPILARI KONUSU

Minimum Destek=

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurallara uyanlar = 6	Kurallara uyanlar = 11	Kurallara uyanlar = 2
2 Tekrar= 48	2 3 Tekrar= 34	2 3 7 Tekrar= 21
3 Tekrar= 68	2 5 Tekrar= 23	3 5 7 Tekrar= 22
4 Tekrar= 28	2 7 Tekrar= 25	
5 Tekrar= 49	2 8 Tekrar= 23	
7 Tekrar= 47	3 4 Tekrar= 25	
8 Tekrar= 45	3 5 Tekrar= 39	
	3 7 Tekrar= 38	
	3 8 Tekrar= 34	
	5 7 Tekrar= 26	
	5 8 Tekrar= 27	
	7 8 Tekrar= 25	

ANALİZİ YAPILABİLECEK KURALLAR

Şekil VII.4 Veriyolu yapıları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.2.1.3. Bellek Yapıları Konusu İçin Bulunan Birliktelik Kuralları

Bellek yapıları konusu için, öğrencilerin konu sonu sınavına verdikleri birinci yanıtlara ait, minimum destek değeri 11 alınarak yapılan birliktelik kuralı analizi Şekil VII.5’de, ikinci yanıtlara ait birliktelik kuralı analizi ise, Şekil VII.6’da görülmektedir. Bu konuyu 25 öğrenci tekrar etmiştir. Şekil VII.5’deki ve Şekil VII.6’daki analizlere göre, uzunluğu bir olan kurallar içerisinde en fazla yanlış yapılan soru ikinci sorudur. İkinci soru her iki analizdeki yüksek destek değeri ile, dikkate alınması gereken bir sorudur. Her iki analizdeki, uzunluğu bir olan kurallar karşılaştırıldığında, ikinci soru dışındaki diğer tüm sorularda yapılan yanlış sayısında azalma sağlandığı görülmektedir. Bu sonuca göre, veriyolu yapıları konusundaki, konu tekrarı başarıyı pozitif yönde etkilemiştir. Her iki analizde de, uzunluğu iki olan kurallar içerisinde en yüksek destek değerine sahip ilk iki kural aynıdır ve bu kurallar $\{2, 8\}$ ve $\{2, 10\}$ olarak görülmektedir.

BELLEK YAPILARI KONUSU (BİRİNCİ CEVAPLAR)

Minimum Destek=

Kural Uzunluğu=1
Kurala uyanlar = 8
2 Tekrar= 64
3 Tekrar= 21
4 Tekrar= 15
5 Tekrar= 33
7 Tekrar= 23
8 Tekrar= 49
9 Tekrar= 13
10 Tekrar= 40

Kural Uzunluğu=2
Kurala uyanlar = 12
2 3 Tekrar= 15
2 5 Tekrar= 20
2 7 Tekrar= 12
2 8 Tekrar= 28
2 10 Tekrar= 26
3 8 Tekrar= 11
3 10 Tekrar= 15
5 7 Tekrar= 12
5 8 Tekrar= 22
5 10 Tekrar= 11
7 8 Tekrar= 14
8 10 Tekrar= 23

Kural Uzunluğu=3
Kurala uyanlar = 3
2 3 10 Tekrar= 11
2 5 8 Tekrar= 12
2 8 10 Tekrar= 12

ANALİZİ YAPILABİLECEK KURALLAR

2 3 10
2 5 8
2 8 10

Şekil VII.5 Bellek yapıları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

Uzunluğu üç olan kurallar içerisinde, Şekil VII.6'da en son öğrenci yanıtlarında bulunan {2, 5, 8} kuralı, ilk öğrenci yanıtlarında da, 12 destek değeri ile bulunmuştur. Bu kural, her iki analizde ortak olduğu için dikkate alınması gereken bir kuraldır.

BELLEK YAPILARI KONUSU

Minimum Destek=

Kural Uzunluğu=1
Kurala uyanlar = 8
2 Tekrar= 66
3 Tekrar= 17
4 Tekrar= 15
5 Tekrar= 32
7 Tekrar= 20
8 Tekrar= 44
9 Tekrar= 11
10 Tekrar= 37

Kural Uzunluğu=2
Kurala uyanlar = 10
2 3 Tekrar= 15
2 5 Tekrar= 20
2 7 Tekrar= 12
2 8 Tekrar= 26
2 10 Tekrar= 27
3 10 Tekrar= 11
5 7 Tekrar= 11
5 8 Tekrar= 20
7 8 Tekrar= 11
8 10 Tekrar= 17

Kural Uzunluğu=3
Kurala uyanlar = 1
2 5 8 Tekrar= 11

ANALİZİ YAPILABİLECEK KURALLAR

2 5 8

Şekil VII.6 Bellek yapıları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.2.1.4.İşlemci Yapısı ve Fonksiyonları Konusu İçin Bulunan Birliktelik Kuralları

İşlemci yapısı ve fonksiyonları konusu için, öğrencilerin verdiği birinci yanıtlara ilişkin minimum destek değeri 12 alınarak yapılan birliktelik kuralı analiz

sonuçları, Şekil VII.7’de, ikinci yanıtlar ise, Şekil VII.8’de görülmektedir. Bu konuyu tekrar eden öğrenci sayısı 27’dir.

Şekil VII.7 İşlemci yapısı ve fonksiyonları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

Her iki analizde de en yüksek destek değerlerine birinci ve yedinci soru sahiptir. Şekil VII.8’deki öğrencilerin ikinci yanıtlarına ait analizde, tüm soruların tekrar yanlış yanıtlanma durumunda azalma görülmektedir. Konunun tekrar edilmesi, öğrenci başarısını olumlu yönde etkilemiştir.

Uzunluğu iki olan kurallar içinde Şekil VII.7’de en yüksek destek değeri {1, 7} kuralına aittir ve aynı kural Şekil VII.8’deki analiz sonucunda da bulunmuştur. {1, 7} kuralının, her iki analiz sonucunda bulunması bu kuralın dikkat çekici ve tutarlı bir kural olduğunu göstermektedir.

Şekil VII.8 İşlemci yapısı ve fonksiyonları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.2.1.5. Komut Düzeyinde Paralellik Konusu İçin Bulunan Birliktelik Kuralları

Komut düzeyinde paralellik konusu için, birinci yanıtlara ait, minimum destek değeri 21 alındığında yapılan birliktelik kuralı analizi Şekil VII.9'da, ikinci yanıtlara ait birliktelik kuralı analizi ise, Şekil VII.10'da görülmektedir. Bu konuyu 56 öğrenci tekrar etmiştir. Şekil VII.9'daki ve Şekil VII.10'daki analizlere göre, uzunluğu bir olan kurallar içerisinde en fazla yanlış yapılan soru, her iki analizde de dördüncü soru olarak görülmektedir. Her iki analizdeki, uzunluğu bir olan kurallar kıyaslandığında, birinci, dördüncü, altıncı, yedinci ve sekizinci sorularda yapılan yanlış sayısında azalma sağlandığı görülmektedir. Bu sonuca göre, komut düzeyinde paralellik konusundaki konu tekrarı başarıyı olumlu yönde etkilemiştir. Her iki analizde de, uzunluğu iki olan kurallara içerisinde en yüksek destek değerine sahip kural aynıdır ve bu kurallar {4,5} olarak görülmektedir.

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 8	Kurala uyanlar = 1	Kurala uyanlar = 3
1 Tekrar= 45	1 2 Tekrar= 23	3 4 5 Tekrar= 23
2 Tekrar= 34	1 3 Tekrar= 25	3 4 6 Tekrar= 21
3 Tekrar= 46	1 4 Tekrar= 31	4 6 8 Tekrar= 21
4 Tekrar= 65	1 5 Tekrar= 22	
5 Tekrar= 41	1 6 Tekrar= 27	
6 Tekrar= 46	1 8 Tekrar= 24	
7 Tekrar= 32	2 4 Tekrar= 25	
8 Tekrar= 39	3 4 Tekrar= 36	
	3 5 Tekrar= 25	
	3 6 Tekrar= 28	
	3 7 Tekrar= 22	
	3 8 Tekrar= 22	
	4 5 Tekrar= 36	
	4 6 Tekrar= 32	
	4 7 Tekrar= 27	
	4 8 Tekrar= 31	

Şekil VII.9 Komut düzeyinde paralellik konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

Şekil VII.9' da öğrencilerin birinci yanıtlarına ait birliktelik kuralı analizinde, en yüksek destek değerine sahip ve kural uzunluğu üç olan {3, 4, 5} kuralı, Şekil VII.10'daki kural sonuçlarında da elde edilmiştir. Her iki analiz sonucunda da yüksek destek değeri ile bulunan {3, 4, 5} kuralı ilginç bir kuraldır.

KOMUT DÜZEYİNDE PARALELİK KONUSU

Minimum Destek=

Kural Uzunluğu=1
Kurala uyanlar = 8

1 Tekrar= 36
2 Tekrar= 36
3 Tekrar= 47
4 Tekrar= 54
5 Tekrar= 45
6 Tekrar= 38
7 Tekrar= 29
8 Tekrar= 33

Kural Uzunluğu=2
Kurala uyanlar = 14

1 3 Tekrar= 23
1 4 Tekrar= 22
1 5 Tekrar= 21
2 3 Tekrar= 26
2 4 Tekrar= 23
2 5 Tekrar= 26
3 4 Tekrar= 30
3 5 Tekrar= 31
3 6 Tekrar= 26
3 7 Tekrar= 22
4 5 Tekrar= 32
4 6 Tekrar= 25
4 8 Tekrar= 21
5 7 Tekrar= 21

Kural Uzunluğu=3
Kurala uyanlar = 2

2 3 5 Tekrar= 21
3 4 5 Tekrar= 21

ANALİZİ YAPILABİLECEK KURALLAR

2 3 5
3 4 5

Şekil VII.10 Komut düzeyinde paralellik konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.2.1.6. Komut Takımı ve Adresleme Modları Konusu İçin Bulunan Birliktelik Kuralları

Komut takımı ve adresleme modları konusu için, öğrencilerin konu sonu sınavına verdikleri birinci yanıtlara ait, minimum destek değeri 30 alınarak yapılan birliktelik kuralı analizi Şekil VII.11’de, ikinci yanıtlara ait birliktelik kuralı analizi ise, Şekil VII.12’de görülmektedir. Konuyu tekrar eden öğrenci sayısı 54’dür. Analiz sonuçlarına göre, birinci, ikinci, beşinci, sekizinci, onuncu, onbirinci sorularda yapılan yanlış sayısı azalırken, diğer sorulardaki yanlış yanıtlanma sayısı artmıştır. Yapılan analizlerde ortak ikili ve üçlü kural bulunmamaktadır. İkinci tekrar sonunda, üçüncü, dördüncü altıncı, dokuzuncu sorularda yapılan yanlış yanıtlanma sayılarındaki artış bu konunun öğrenciler tarafından tam olarak anlaşılmadığını göstermektedir.

KOMUT TAKIMI VE ADRESLEME MODLARI KONUSU (BİRİNCİ YANITLAR)

Minimum Destek=

Kural Uzunluğu=1
Kurala uyanlar = 8

1 Tekrar= 30
2 Tekrar= 41
4 Tekrar= 30
5 Tekrar= 44
6 Tekrar= 53
8 Tekrar= 75
10 Tekrar= 37
11 Tekrar= 61

Kural Uzunluğu=2
Kurala uyanlar = 7

2 8 Tekrar= 32
5 6 Tekrar= 32
5 8 Tekrar= 37
6 8 Tekrar= 49
6 11 Tekrar= 36
8 10 Tekrar= 32
8 11 Tekrar= 53

Kural Uzunluğu=3
Kurala uyanlar = 1

6 8 11 Tekrar= 34

ANALİZİ YAPILABİLECEK KURALLAR

6 8 11

Şekil VII.11 Komut takımı ve adresleme modları konusu için öğrencilerin birinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

KOMUT TAKIMI VE ADRESLEME MODLARI KONUSU

Minimum Destek= 30

Kuralları Bul Veri İnceleme

ANALİZİ YAPILABİLECEK KURALLAR

4 6 9

Güven Değeri Hesapla

Soruları İncele

Kurala Ait Soruları İncele

Kural Uzunluğu=1	Kural Uzunluğu=2	Kural Uzunluğu=3
Kurala uyanlar = 8	Kurala uyanlar = 5	Kurala uyanlar = 1
2 Tekrar= 31	4 6 Tekrar= 40	4 6 9 Tekrar= 31
3 Tekrar= 32	4 9 Tekrar= 34	
4 Tekrar= 44	6 9 Tekrar= 54	
5 Tekrar= 32	6 11 Tekrar= 36	
6 Tekrar= 72	9 11 Tekrar= 32	
9 Tekrar= 65		
10 Tekrar= 32		
11 Tekrar= 40		

Şekil VII.12 Komut takımı ve adresleme modları konusu için öğrencilerin ikinci yanıtlarına ait veri ile yapılan birliktelik kuralı analizi

VII.3. TÜM ÖĞRENCİLERİN BİRİNCİ ve İKİNCİ YANITLARINA AİT ÇOK BOYUTLU BİRLİKTELİK KURALLARININ KARŞILAŞTIRILMASI

Tüm öğrencilerin konu sonu sınavına verdikleri birinci yanıtlara ait, minimum destek değeri 40 alınarak yapılan çok boyutlu birliktelik kuralı analizi Şekil VII.13’de, ikinci yanıtlara ait birliktelik kuralı analizi ise, Şekil VII.14’de görülmektedir. Şekil VII.13’deki ve Şekil VII.14’deki analiz sonucunda bulunan iki boyutlu birliktelik kuralları içerisinde üç tane ortak kural bulunmuştur. Bunlar;

1) Konu 2’ye (veriyolu yapıları konusu) ait üçüncü soru ve konu 3’e (bellek yapıları konusu) ait ikinci soru arasında bulunan birliktelik kuralı,

2) Konu 2’ye (veriyolu yapıları konusu) ait üçüncü soru ve konu 5’e (komut düzeyinde paralellik konusu) ait dördüncü soru arasında bulunan birliktelik kuralı,

3) Konu 3’e (bellek yapıları konusu) ait ikinci soru ve konu 5’e ait (komut düzeyinde paralellik konusu) dördüncü soru arasında bulunan birliktelik kurallarıdır.

Bu birliktelik kuralları her iki analizde de yüksek güven değeri ile bulunan kurallardır olduğu için ilginç kurallardır.

TÜM VERİLER İÇİN BULUNAN ÇOK BOYUTLU BİRLİKTELİK KURALLARI (BİRİNCİ YANITLAR)

Giriş Konusu	Veriyolu Konusu	Bellek Konusu	İşlemci Konusu	Paralellik Konusu	Komut Takımı Konusu
6 Tekrar=46 7 Tekrar=44	2 Tekrar=50 3 Tekrar=72 5 Tekrar=49 7 Tekrar=55 8 Tekrar=50	2 Tekrar=64 8 Tekrar=49 10 Tekrar=40	1 Tekrar=44 7 Tekrar=44	1 Tekrar=45 3 Tekrar=46 4 Tekrar=65 5 Tekrar=41 6 Tekrar=46	2 Tekrar=41 5 Tekrar=44 6 Tekrar=53 8 Tekrar=75 11 Tekrar=61

Destek Değeri: 40 Kuralları Bul

Veriler: Öğrenci Sayısı: 103

ANALİZİ YAPILABİLECEK KURALLAR

İki Boyutlu Kurallar	Üç Boyutlu Kurallar	Dört Boyutlu Kurallar	Beş Boyutlu Kurallar	Altı Boyutlu Kurallar
K= 23 S 32 T:56 K= 25 S 34 T:45 K= 26 S 38 T:55 K= 26 S 3 11 T:47 K= 26 S 78 T:43 K= 35 S 24 T:42 K= 36 S 28 T:46 K= 36 S 2 11 T:42 K= 56 S 48 T:52 K= 56 S 4 11 T:42	K= 236 S 328 T:42			K= 236 S 328 T:42

Güven Değeri Analizi

Şekil VII.13 Tüm öğrenciler için öğrencilerin birinci yanıtlarına ait veri ile yapılan çok boyutlu birliktelik kuralı analizi

TÜM VERİLER İÇİN BULUNAN ÇOK BOYUTLU BİRLİKTELİK KURALLARI

Giriş Konusu	Veriyolu Konusu	Bellek Konusu	İşlemci Konusu	Paralellik Konusu	Komut Takımı Konusu
6 Tekrar=43 7 Tekrar=42	2 Tekrar=48 3 Tekrar=68 5 Tekrar=49 7 Tekrar=47 8 Tekrar=45	2 Tekrar=66 8 Tekrar=44	1 Tekrar=40 7 Tekrar=40	3 Tekrar=47 4 Tekrar=54 5 Tekrar=45	4 Tekrar=44 6 Tekrar=72 9 Tekrar=65 11 Tekrar=40

Destek Değeri: 40 Kuralları Bul

Veriler: Öğrenci Sayısı: 103

ANALİZİ YAPILABİLECEK KURALLAR

İki Boyutlu Kurallar	Üç Boyutlu Kurallar	Dört Boyutlu Kurallar	Beş Boyutlu Kurallar	Altı Boyutlu Kurallar
K= 23 S 32 T:55 K= 25 S 34 T:40 K= 26 S 36 T:55 K= 26 S 39 T:51 K= 35 S 24 T:40 K= 36 S 26 T:50 K= 36 S 29 T:47 K= 56 S 46 T:42	K= 236 S 326 T:43 K= 236 S 329 T:41			K= 236 S 326 T:43 K= 236 S 329 T:41

Güven Değeri Analizi

Şekil VII.14 Tüm öğrenciler için öğrencilerin ikinci yanıtlarına ait veri ile yapılan çok boyutlu birliktelik kuralı analizi

Üç boyutlu kurallar içerisinde, her iki analizde ortak olan kural bulunmamaktadır. Fakat ikinci yanıtlar içerisinde, komut takımı ve adresleme modları konusuna ait Soru 4, Soru 6 ve Soru 9'un yanlış yanıtlanmasındaki artış bu konunun öğrenciler tarafından iyi anlaşılmadığını göstermekte ve bulunan üçlü kurallar içerisinde Soru 6 ve Soru 9'un yer almasına neden olmaktadır.

VII.4.EĞİTİM AÇISINDAN KÜMELEME ANALİZİNE AİT SONUÇLARIN DEĞERLENDİRİLMESİ

Kümeleme analizine ait sonuçlar, Bölüm VI.3’de açıklanmaktadır. Analiz sonucu Bölüm VI.3’de görülen ve öğrencilerin giriş ve veriyolu konularından aldıkları başarı puanlarının küme sayısı dört ile analiz edildiği kümeleme işleminde, en büyük küme 51 öğrenci ile genelde her iki konudan da başarılı olan öğrencilerin oluşturduğu kümedir. Bu konular arasında, başarı açısından bir ilişki olduğu görülmektedir. Giriş konusundan başarılı olan öğrenciler, veriyolu konusundan da başarılı olmuşlardır.

Bellek yapıları ile işlemci yapısı ve fonksiyonları konuları arasında, öğrenci başarı puanlarına ait veri ile küme sayısı dört alınarak yapılan analizde, en büyük küme her iki konudan da başarılı olan ve toplamda 72 üyeye sahip kümedir. Bu kümede, bellek yapıları konusundan 40-100 arasında puan alan öğrenciler, işlemci yapısı ve fonksiyonları konusundan 80-100 arasında puan almışlardır. Bellek yapıları konusundan orta düzeyde puan alan öğrencilerin büyük bir kısmının işlemci yapısı konusundan yüksek puan aldığı görülmüştür. Bellek yapıları konusu, işlemci yapısı ve fonksiyonları konusu için destekleyici bir konu olmuştur.

İşlemci yapısı ve fonksiyonları konusu ile komut düzeyinde paralellik konusu için küme sayısı dört ile yapılan analizde, en büyük küme, ağırlıklı olarak her iki konudan başarılı olan öğrencilerden oluşan 47 üyeye sahiptir. 32 üyeye sahip diğer bir kümede ise, işlemci yapısı konusundan başarılı olan öğrencilerin komut düzeyinde paralellik konusundan oldukça başarısız olduğu görülmektedir. Bu nedenle bu iki konu arasında başarı açısından nasıl bir ilişki olduğunu belirlemek mümkün olmamaktadır.

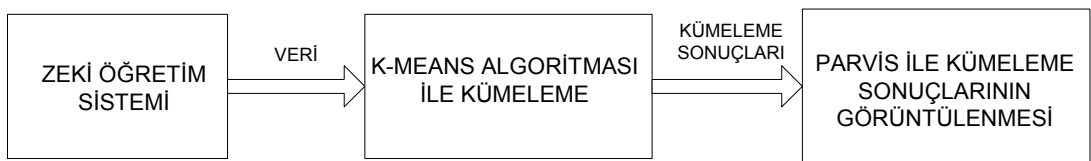
İşlemci yapısı ve fonksiyonları konusu ile komut takımı ve adresleme modları konusu için küme sayısı dört olarak belirlendiğinde, en büyük küme 51 öğrenci ile genelde her iki konudan da başarılı olan öğrencilerin oluşturduğu kümedir. Bu analizde, işlemci yapısı konusundan başarılı olan öğrencilerin komut takımı ve adresleme modları konusundan da başarılı olduğu görülmüştür. Bu analizde bulunan ikinci büyük kümenin eleman sayısı ise 23’dür ve bu kümede yer alan öğrenciler işlemci yapısı ve fonksiyonları konusundan başarılı olmalarına rağmen, komut takımı ve adresleme modları konusundan başarısız olmuştur. İşlemci yapısı ve fonksiyonları

dersinden öğrencilerin başarılı olmasının bir nedeni de, öğrencilerin daha önceki dönemlerde aldıkları mikroişlemciler dersinin bu konu için temel oluşturması olabilir.

Komut düzeyinde paralellik konusu ile komut takımı ve adresleme modları konuları için küme sayısı dört alınarak yapılan analizde, elde edilen en büyük küme 43 üyeye sahip ve genelde her iki konudan da başarılı olmuş öğrencilerin oluşturduğu kümedir. Bu sonuç doğrultusunda, bu iki konu arasında başarı açısından bir ilişki olduğunu belirtmek mümkündür.

VII.5. EĞİTİM AÇISINDAN ÇOK BOYUTLU KÜMELEME SONUÇLARININ GÖRÜNTÜLENMESİ

Bölüm VI.3’de belirlenen iki konuya ait başarı puanlarına göre kümeleme analizi ve bu analiz sonuçlarının görüntülenmesine ait sonuçlar açıklanmıştır. Tasarlanan kümeleme analizi programı ile, çok boyutlu verilerin kümelenmesi de mümkündür. Fakat çok boyutlu verilerin kümeleme sonuçlarının, iki boyutlu eksen üzerinde görüntülenmesi mümkün değildir. Bu sorunu çözmek için, Bölüm III.4’de açıklanan paralel koordinat sistemi ve Parvis [103] programından yararlanılmıştır. Şekil VII.15’deki aşamalar takip edilerek, ZÖS’den alınan öğrenci başarı puanları, tasarımı bu tez çalışmasında gerçekleştirilen kümeleme analizi programı ile kümelendikten sonra, bu analize ait sonuçlar Parvis ile görselleştirilmiştir.

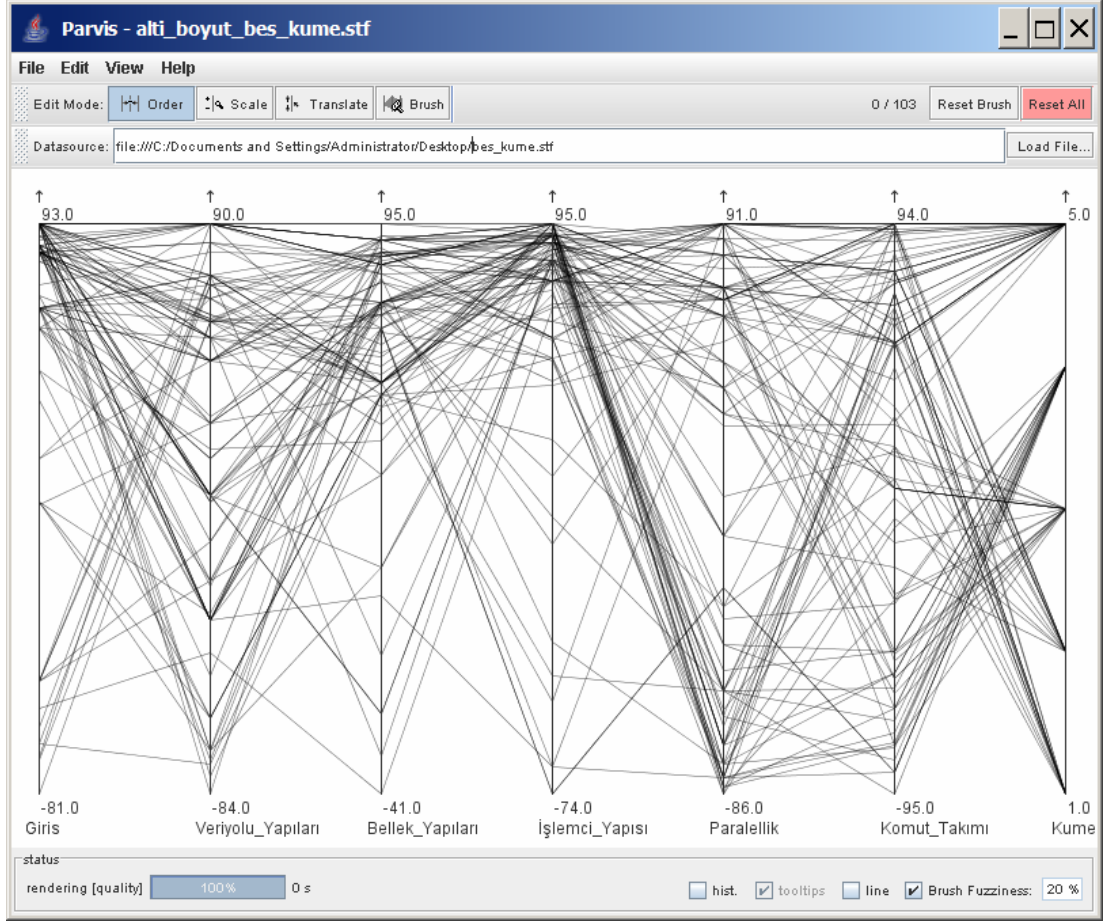


Şekil VII.15 Kümeleme analizi sonuçlarının görselleştirilmesi

Öğrencilerin tüm konulardan aldıkları başarı puanlarına ait 103 elemanlı veri seti, küme sayısı beş olacak biçimde, tasarlanan kümeleme yazılımı ile k-means algoritmasına göre kümelendirilmiştir. Analiz sonuçlarının, Parvis’e yüklenmesi ile elde edilen çok boyutlu kümeleme sonuçları Şekil VII.16’da görülmektedir. Şekil VII.16’daki ilk altı eksen öğrenci başarı puanlarına ait değerleri temsil etmek için, ekranın en sonunda yer alan eksen ise kümeleme analizine göre, oluşturulan beş küme içerisinde, verinin hangi kümede yer aldığını belirtmek için kullanılmaktadır. Her eksenin altında, hangi konuya ait başarı puanının temsil edildiği belirtilmektedir.

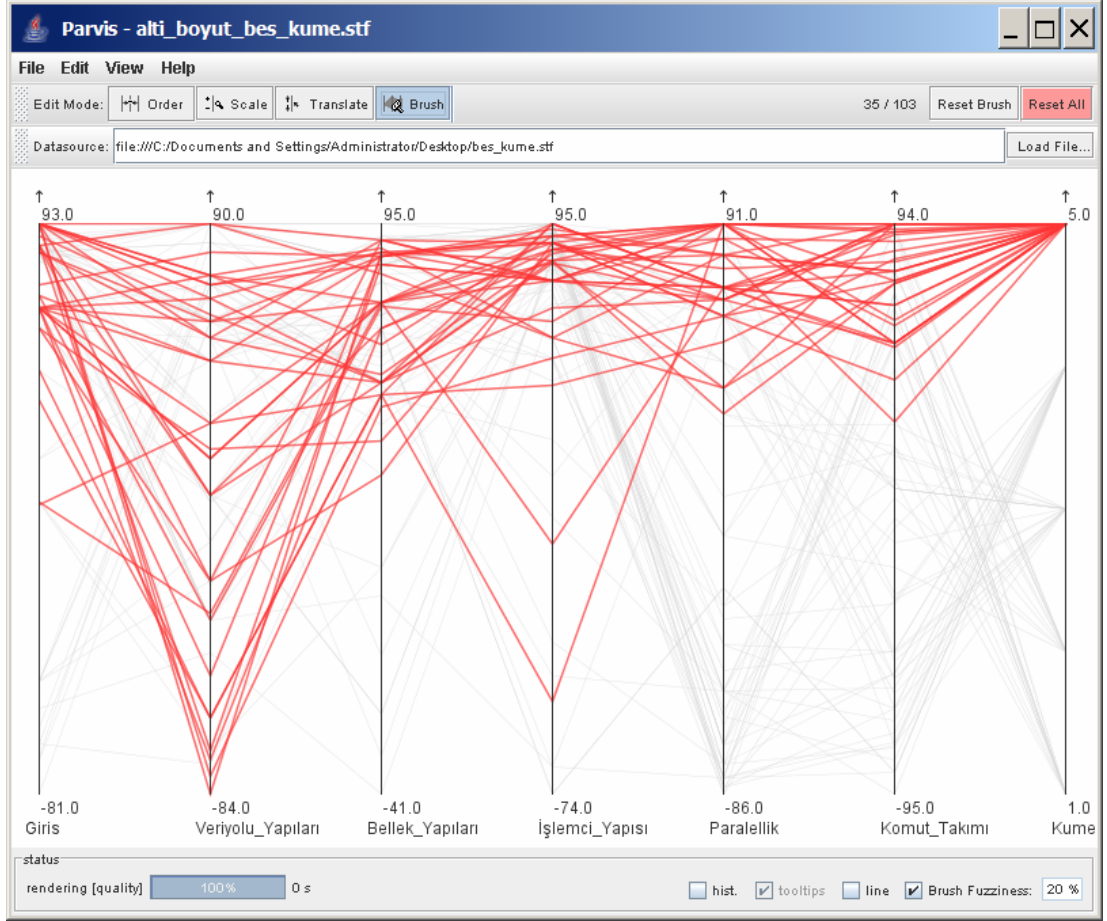
Şekil VII.16'da, öğrencinin her konudan aldığı puan, ilgili konuya ait eksen üzerinde işaretlenmekte ve öğrenciye ait hattın eksenler üzerinde belirlenen noktalardan geçmesi ile öğrencinin puanı temsil edilmektedir. MYCIN güven faktörü kullanılarak hesaplanan öğrenci başarı puanlarının tümü 100 ile çarpılarak, (-100, +100) aralığında değişmesi sağlanmıştır. Başarı puanının -100'e yaklaşması öğrencinin konuyu bilmediğini ve başarısız olduğunu, +100'e yaklaşması ise, konuyu iyi düzeyde bildiğini ve başarılı olduğunu göstermektedir. Parvis veri setindeki her boyut için, boyutta yer alan en büyük ve en küçük değeri kullanarak eksen ölçekleme işlemi otomatik yapmaktadır. Örneğin, Şekil VII.16'da giriş konusu eksen -81'den başlayıp, 93'de son bulmaktadır. Bu eksen, giriş konusundaki en büyük başarı puanının 93, en düşük başarı puanının ise -81 olduğu anlaşılmaktadır.

Şekil VII.16 incelenerek, genel başarı puanlarının durumu ve oluşturulan kümelerdeki başarı durumları konusunda fikir edinmek mümkündür. Örneğin; öğrencilerin giriş konusundan genelde başarılı olduğu, bu konuya ait eksen, eksenin üst kısımlarında yani yaklaşık 80-93 aralığındaki hatların yoğunluğunun fazla olmasından anlaşılmaktadır. Fakat bu konudan başarılı olan birçok öğrencinin, veriyolu yapıları konusundan başarısız olduğu, giriş konusunda 80-93 aralığında yer alan birçok hattın, veriyolu yapılarına ait eksen, eksenin alt kısımlarına, -84'e yakın noktalarda bu eksen kesmesinden anlaşılmaktadır. Ayrıca Şekil VII.16'da, bellek yapıları ve işlemci yapıları konularında başarılı öğrenci yoğunluğunun fazla olduğu görülmektedir. Fakat işlemci yapısı konusunda başarılı olan birçok öğrenci, paralellik konusundan başarısız, komut takımı konusunda ise başarılı olmuştur.



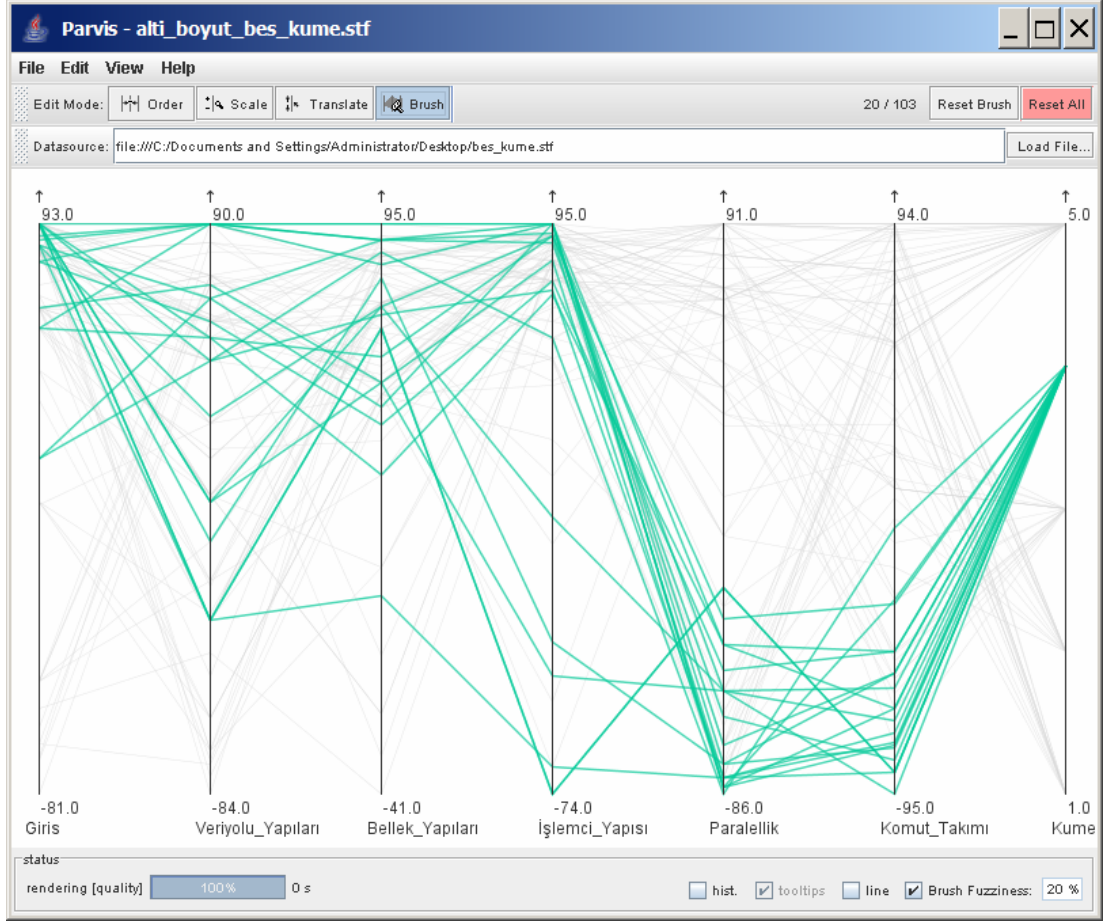
Şekil VII.16 Tüm öğrencilere ait başarı puanlarının Parvis’e yüklenmesi ile elde edilen ekran görüntüsü

Herhangi bir eksende istenilen veri grubunun Parvis’de ile seçildikten sonra, kaydedilmesi ve sadece o kümenin elemanlarının değer dağılımlarının görüntülenerek, kümede var olan eleman sayısı ile ilgili bilgi alınması mümkündür. Şekil VII.17’de bu amaçla seçilen küme görülmektedir. Bu kümeye, toplam 35 adet veri dahil olmuştur ve analizde toplam 103 veri bulunmaktadır. Kümedeki eleman sayısı, ekranın sağ üst köşesinde görülen 35/103 ifadesinden anlaşılmaktadır. Bu ifadede, “/” karakterinden önceki değer kümedeki eleman sayısını, sonraki değer ise analizdeki toplam veri adetini göstermektedir. Oluşturulan kümede, giriş konusundan başarılı olan birçok öğrencinin veriyolu yapıları konusundan başarısız olduğu görülmektedir. Aynı öğrenciler bellek yapıları, işlemci yapısı, paralellik ve komut takımı konularından ise başarılı olmuşlardır.



Şekil VII.17 Tüm öğrencilerin başarı puanlarına ait veri üzerinde Parvis’de seçilen beşinci kümenin görüntülenmesi

Şekil VII.18’de ise, gerçekleştirilen kümeleme analizi sonucunda oluşan, 20 elemanlı dördüncü küme görülmektedir. Bu kümedeki öğrenciler, giriş ve işlemci yapısı konusundan genelde başarılı olurken, paralellik ve komut takımı konusundan başarısız olmuşlardır.



Şekil VII.18 Tüm öğrencilerin başarı puanlarına ait veri üzerinde Parvis’de seçilen dördüncü kümenin görüntülenmesi

Parvis’de istenilen herhangi bir ekseninde birbirine benzer yapıda görülen veri elemanlarının seçilerek kaydedilmesi ve kaydedilen dosyanın metin dosyasına dönüştürülmesi mümkündür. Metin dosyasında, veri seti içinde, herhangi bir veri, seçim işlemine dahilse, verinin bulunduğu satırda “1” değeri yer almakta, veri seçim işleminde yer almıyorsa “0” değeri yer almaktadır. Böylece seçilen verilerin gerçekte veri setinde hangi değere karşılık geldiği belirlenmektedir.

BÖLÜM VIII

DEĞERLENDİRME VE ÖNERİLER

VIII.1. DEĞERLENDİRME

Gelişen bilgisayar teknolojileri ile birlikte, zeki öğretim sistemlerinde öğrenciyle ilgili bilgi toplama imkanları artmıştır. Öğrencilerin verdiği yanıtlar, başarı puanları, öğrencilerin ziyaret ettikleri ürün sayfaları gibi farklı yapıdaki veriler bu tür sistemlerde toplanmaktadır. Bu verilerden anlamlı bilgiler elde edilmesi, geliştirilen öğretim sistemlerinin takibinin, gelişmesinin, güncelleştirilmesinin yapılması, öğrencilerin öğrenme düzeyi ve farklılıkları konusunda bilgi sağlanması için büyük önem taşımaktadır. Veriden anlamlı bilgi elde etmek için veri madenciliği disiplini kullanılmaktadır. Veri madenciliğinin ZÖS gibi öğrenciye ait veriler içeren sistemlerde kullanılması ile sistemdeki verilerin analiz edilmesi mümkündür.

Gerçekleştirilen bu tez çalışması ile kaplama öğrenci modeli kullanan ürün tabanlı bir ZÖS geliştirildi ve bu sistemde yer alan verilerin tasarlanan veri madenciliği programları ile analiz edilmesi sağlandı. Veri madenciliği analizi, birliktelik kuralı ve kümeleme teknikleri kullanılarak yapılmıştır. Bu analizin içeriği ve bilimsel katkısı aşağıda belirtilmektedir:

- ZÖS’den alınan veriler ile, aynı konu içerisinde öğrencilerin birlikte başarısız olduğu soru dizilerinin ve aldıkları başarı puanlarına göre birlikte başarısız oldukları konu dizilerinin, birliktelik kuralı analizi ile bulunması için ZEBİKA yazılımı tasarlanmıştır. Bu yazılım ile; öğretmen, alt kavramlar düzeyinde birbirlerini etkileyen soruları saptamaktadır.

- ZÖS'deki veriler ile, farklı konular içerisinde birlikte yanlış yapılan sorular ise, tasarlanan ZECOKA yazılımı ile, çok boyutlu birliktelik kuralı analizi yapılarak bulunmaktadır. Bu analiz ile, farklı konuların içerisindeki alt kavramların ilişkileri belirlenir. Bulunan birliktelik kurallarına göre, ZÖS'de gerekli güncellemelerin, örneğin başarısız olunan soruların anlatıldığı kısmın basitleştirilmesi, konular arası ilişkilerin düzenlenmesi, soru puanlarının öğrenci başarısına göre belirlenmesi işlemleri yapılır. Tüm birliktelik kuralı analiz sonuçları dikkate alındığında altıncı konu dışında, öğrencilerin konu sonu sınavına verdikleri birinci ve ikinci yanıtlara ait birliktelik kuralları genelde benzerlik göstermektedir. Bu da, bulunan birliktelik kurallarının ilginç ve tutarlı olduğunu, öğretici tarafından dikkate alınabilir nitelikte olduğunu göstermektedir. Analiz sonuçlarında da görüldüğü gibi; ZÖS'deki konu tekrarı, öğrencilerin soruları yanlış yapma durumunda azalma sağlayarak, öğrenci başarısına katkıda bulunmaktadır.
- Gerçekleştirilen kümeleme analizi yazılımı ise, öğrencilerin ZÖS'deki başarı puanlarına göre, kümeleme işlemini gerçekleştirmektedir. Kümeleme analizinde kullanılan k-means algoritmasının başlangıç değerleri seçiminden etkilenmesini engellemek için, tekrarlı çalışan k-means algoritması tasarlanmış ve yazılımda kullanılmıştır. Yapılan denemeler sonucunda; tasarlanan tekrarlı k-means algoritmasının standart k-means algoritmasına göre, daha az küme içi varyasyonlar toplamı, yani daha az toplam hata ürettiği görülmüştür. Tasarlanan kümeleme yazılımı ile; öğretmen, belirlediği konulardaki, konu başarı puanlarına göre istediği sayıda öğrenci kümesi oluşturmaktadır.
- İki boyutlu kümeleme sonuçları tasarlanan programda görüntülenirken, daha fazla boyuta sahip kümeleme analiz sonuçlarının görüntülenmesi için paralel koordinat sistemi kullanılmıştır. Öğrenci başarı puanlarında kümeleme analizinin yapılması, konulardaki başarı durumlarına göre öğrenci dağılımlarının belirlenmesini sağlar. Bu sonuçlar ile, öğretmen, konular düzeyindeki öğrenci kümelerini görerek, öğrencilere rehberlik yapar. Ayrıca bu sonuçlardan yapılacak grup çalışmalarında homojen veya heterojen öğrenci kümelerinin oluşturulmasında faydalanılır.

- Çok boyutlu kümeleme analizi sonuçlarının, paralel koordinat sisteminde görüntülenmesi ise, bu sonuçların anlaşılabilirliğini artırmış ve paralel koordinat sisteminden eğitim alanında da yararlanılması sağlamıştır.
- Tüm yapılan birliktelik kuralı analizleri, öğrencilerin doğru yanıtları için de yapılarak öğrenciye ve öğretmene bilgi sağlanabilir.
- Bu çalışma; zeki öğretim sistemlerinde veri madenciliği tekniklerinden, tek ve çok boyutlu birliktelik kuralı analizi ve kümeleme analizinin yapılmasını sağladığından bu alana yenilik getirmiştir.

VIII.2. ÖNERİLER

Gerçekleştirilen çalışmada, öğrenciler, ZÖS’de başarısız oldukları konuları en fazla iki kez tekrar etmişlerdir. Fakat iki kez tekrar sonucunda başarısız olan öğrenciler bulunmaktadır. Bu nedenle, daha fazla sayıda tekrar imkanı verilerek sistemin kullanılması gerekmektedir. Öğrenci başarısızlıklarının giderilmesi için, özellikle başarısız öğrenci sayısının fazla olduğu konulardaki alan modelinin geliştirilmesi, eksiklerinin giderilmesi gerekmektedir. Analiz sonuçlarına göre, ZÖS’ün gelişiminin sağlanması ve gerekli güncellemelerin yapılarak yeni öğrenci gruplarında sistemin denenmesi ve sonuçların karşılaştırılması sistemin gelişiminin sağlanması açısından gereklidir. Ayrıca farklı öğrenci modellerine sahip olan sistemler tasarlanarak, bu sistemlerden elde edilen verilerin analiz edilmesi, bu tür sistemlerde veri madenciliğinin kullanılmasına ilişkin örnek oluşturacaktır.

Birliktelik kuralı analizi için Apriori algoritması dışında, farklı algoritmalar kullanılarak sonuçlarının karşılaştırılması daha etkili birliktelik kurallarına ulaşmak açısından yararlı olacaktır.

Kümeleme sonuçlarının kullanılması ile, var olan sonuçlardan faydalanılarak öğrencinin gelecekteki başarı durumunun tahmin edilmesi ve farklı kümeleme algoritmalarının uygulanması önerilmektedir.

KAYNAKLAR

- [1] Beck, J.; Stern, M.; Haugsjaa, E.: “Applications of AI in Education”, *In Proceedings of the Third International Conference on Intelligent Tutoring Systems (1996)*.
- [2] Murray, T.: “Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy”, Student Module and Interface Design. *Journal of the Learning Sciences*, Vol 7, No 1 (1998) 5-64.
- [3] Clancey, W.: “Methodology for Building Intelligent Tutoring Systems”, In Greg P. Kearsley (Ed.), *Artificial Intelligence & Instruction Application and Methods*, MA: Addison-Wesley (1987).
- [4] VanLehn, K.: “Student Modelling”, In M. Polson (Ed.), *Foundations of Intelligent Tutoring Systems*, Hillsdale, Lawrence Erlbaum Associates (1988).
- [5] Kimiavi, S.: “ITS-Author:A Framework for Building Hypermedia-Based Intelligent Tutoring Systems for the WWW”, *PhD Thesis*, The School of Engineering and Applied Science of The George Washington University (1998) 15-23.
- [6] Burns, H.; Capps, C. : “Foundations of intelligent tutoring systems: An introduction”, In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems*. Hilldsale, NJ: Erlbaum (1988) 1-20.
- [7] Wenger, E.: “Artificial Intelligence and Tutoring Systems:Computational and Cognitive Approaches to the Communication of Knowledge”, Los Altos: Morgan Kaufmann Publishers (1987).
- [8] Suraweera, P. : “An Intelligent Teaching System for Database Modelling”, *Msc Thesis* , Computer Science at the University of Canterbury (2001) 5-60.
- [9] Anderson, J.R.: “The Expert Module”, In M.C. Polson and J.J Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*, New Jersey: Lawrence Erlbaum Associates, Inc (1988) 21-53.

- [10] Martin, B: "Intelligent Tutoring Systems: The practical implementation of constraint-based modelling", *PhD Thesis*, University of Canterbury **(2003)** 12-36.
- [11] Doğan, B.; Çamurcu Y., "Student Modelling Approaches in Intelligent Tutoring Systems", *New Information Technologies in Education (NITE2004)*, İzmir, Ekim **(2004)**.
- [12] Anderson, J. R.; Corbett, A.; Koedinger, K.; Pelletier, R.: "Cognitive Tutors: Lessons Learned, *Journal of Learning Sciences*, 4 (2) **(1996)** 167-207.
- [13] Anderson, J.R.; Boyle, C.F.; Farrell, R.; Reiser, B.J.: "Cognitive Principles in the Design of Computer Tutors" In P. Morris (Ed.), *Modelling Cognition*, Wiley **(1987)**.
- [14] Anderson, J. R.; Pelletier, R., "A development system for model-tracing tutors", *Proceedings of the International Conference of the Learning Sciences*, **(1991)** 1-8
- [15] Corbett, A. T.; Trask, H. J.; Scarpinato, K. C.; Hadley, W. S: "A Formative Evaluation of the PACT Algebra II Tutor: Support for Simple Hierarchical Reasoning", In Goettl, B. P., Halff, H. M., Redfield, C. L. and Shute, V. J. (eds.). *Proc. of 4th International Conference on Intelligent Tutoring Systems*, San Antonio, Texas **(1998)**.
- [16] Aleven, V.; Koedinger, K. : "Limitations of Student Control: Do students know when they need help? In Gauthier", G., Frasson, C. and VanLehn, K. (eds.). *Proc. of 5th International Conference on Intelligent Tutoring Systems*, Montreal , Springer **(2000)**.
- [17] Heffernan III, N.T.: "Intelligent Tutoring System have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors", *PhD Thesis*, The Carnegie Mellon University , Pittsburg, **(2001)**.
- [18] Albacete, L.P.: "An Intelligent Tutoring System for Teaching Fundamentals of Physics", *PhD Thesis*, University of Pittsburg Graduate Faculty of Art and Sciences **(1999)**.
- [19] Goldstein, P.: "Overlays : A Theory of modelling for computer - aided instructions", AI Memo 406, MIT, Cambridge, MA **(1977)**.
- [20] Intelligent Tutoring Systems, <http://www.cs.mdx.ac.uk/staffpages/serengul/Overlay.student.models.htm>, (Erişim Tarihi:Nisan **2003**).

- [21] Mitrovic, A.: "A Knowledge-Based Teaching System for SQL", In *Proc. of EDMEDIA/ED-TELECOM'98*, Freiburg (1998).
- [22] Mitrovic, A.: "Learning SQL with a Computerised Tutor", In *Proc. of 29th ACM SIGCSE Technical Symposium*, Atlanta (1998).
- [23] Ohlsson, S.: "Constraint-based Student Modelling", In: Greer, J.E., McCalla, G.I.(Eds.): *Student Modelling: the Key to Individualized Knowledge-based Instruction* (1994).
- [24] Mayo, J.M.: "Bayesian Student Modelling and decision-Theoretic Selection of Tutorial Actions in Intelligent Tutoring Systems", *PhD Thesis*, Computer Science In The University Of Canterbury (2001).
- [25] Weber, E.M.: "Architecture for a Web Based Intelligent Tutoring System For Pspice", *MSc Thesis*, Electrical Engineering Department of Electrical and Computer Engineering in the Mississippi State University (1998).
- [26] Liegle, O.L.: "Development and Evaluation of an Adaptive Web-Based Intelligent Tutoring System", *PhD Thesis*, Kent State University Graduate School of Management (1999).
- [27] Burton, R.R; Brown, J.S.: "An investigation of computer coaching for informal learning activities. Intelligent Tutoring Systems", Sleeman D. and Brown J.(Eds), Orlando, FL: Academic Press. (1982).
- [28] Jushua, R.O.: "A Hypermedia-based intelligent computer-assisted instruction model", *PhD Thesis*, University of Manitoba in Computer Science, Canada, (1996).
- [29] Ohlsson, S.: "Learning from Performance Errors", *Psychological Review* 3(2) (1996).
- [30] Clancey W., "Methodology for Building Intelligent Tutoring Systems", In Greg P. Kearsley (Ed.), *Artificial Intelligence & Instruction Application and Methods, Readings*, MA: Addison-Wesley, (1987).
- [31] Mitrovic, A.: "Learning SQL with a Computerised Tutor", In *Proc. of 29th ACM SIGCSE Technical Symposium*, Atlanta (1998).
- [32] Mitrovic, A.; Mayo, M.; Suraweera, P.; Martin, B.: "Constraint-based Tutors: a Success Story. In Monostori", L., Vancza, J. and Ali, M. (eds.). *Proc. of 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2001)*, Budapest, Springer-Verlag Berlin Heidelberg LNAI 2070 (2001).

- [33] Suraweera, P.; Mitrovic, A.: “Designing an Intelligent Tutoring System for Database Modelling”. In Smith, M. J. and Salvendy, G. (eds.). *Proc. of 9th International Conference on Human-Computer Interaction (HCII 2001)*, New Orleans, LA, August **(2001)**.
- [34] Mayo, M.; Mitrovic, A.; McKenzie, J.: “CAPIT: An Intelligent Tutoring System for Capitalisation and Punctuation”. In Kinshuk, Jesshope, C. and Okamoto, T. eds.). *Proc. of Advanced Learning Technology: Design and Development Issues*, Los Alamitos, IEEE Computer Society **(2000)**.
- [35] Shiri, M. E.; Aimeur, E.; Frassen, C.: “Student Modelling by Case-Based Reasoning”, In Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.): *Fourth International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science*, Vol. 1452. Springer-Verlag, Berlin **(1998)** 394-404.
- [36] Funk, P.; Conlan, O., “Using Case-Based Reasoning to Support Authors of Adaptive Hypermedia Systems”, <http://citeseer.ist.psu.edu/650369.html>, (Eriřim Tarihi:řubat **2006**).
- [37] Reyes, R.L.; Sison, R.; “Case Retrival in CBR-Tutor”, Proceedings of the International Conference on Computers in Education (ICCE’02) ” **(2002)**.
- [38] Gertner A., VanLehn K., “Andes: A Coached Problem Solving Environment for Physics”, In Gauthier G., Frasson C., and VanLehn K. (Eds.), *Proceedings of Fifth International Conference on Intelligent Tutoring Systems*, Springer-Verlag, **(2000)**.
- [39] Mislevy, R.; Gitomer, D.: “The Role of Probability-Based Inference in an Intelligent Tutoring System”, *CSE Technical Report 413*, CRESST. <http://cresst96.cse.ucla.edu/CRESST/pages/reports.htm>.
- [40] Murray, R.; VanLehn, K.: “DT Tutor: A Decision-Theoretic, Dynamic Approach for Optimal Selection of Tutorial Actions”, In Gauthier G., Frasson C., and VanLehn K. (Eds.), *Proceedings of Fifth International Conference on Intelligent Tutoring Systems*, Springer-Verlag, **(2000)**.
- [41] Mayo, J.M.: “Bayesian Student Modelling and decision-Theoretic Selection of Tutorial Actions in Intelligent Tutoring Systems”, *PhD Thesis*, Computer Science In The University Of Canterbury **(2001)**.
- [42] Mayo, M.; Mitrovic, A., McKenzie, J.: “CAPIT: An Intelligent Tutoring System for Capitalisation and Punctuation”, In Kinshuk, C. Jesshope & T. Okamoto (Eds.) *Proceedings of IEEE International Workshop on Advanced*

- Learning Technologies* (Los Alamitos: IEEE Computer Society Press) **(2000)** 151-154.
- [43] Winter, M.; McCalla, G.: “The emergence of student models from an analysis of ethical decision making in a scenario-based learning environment”, In Kay, J. (ed.), *Proc. of Seventh International Conference on User Modelling*, Springer-Verlag **(1999)**.
- [44] Kay, J.: “Stereotypes, Student Models and Scrutability”, In Gauthier, G., Frasson, C. and VanLehn, K. (eds.). *Proc. of 5th International Conference on Intelligent Tutoring Systems*, Montreal, Springer **(2000)**.
- [45] Dimitrova, V.; Self, J.; Brna, P.: “The interactive maintenance of open learner models”, *Artificial Intelligence in Education*, 50, **(1999)**.
- [46] Wheeler, J.; Regian, J.: “The use of a cognitive tutoring system in the improvement of the abstract reasoning component of word problem solving”, In *Computers in Human Behaviour*, **(1999)**.
- [47] Milne, S.; Shiu, E.; Cook, J. : “Development of a model of user attributes and its implementation within an adaptive tutoring system”, *User modeling and useradapted interaction* 6, **(1996)**.
- [48] “The use of Agents techniques on Intelligent Tutoring Systems”, <http://www.c5.cl/ieinvestiga/actas/ribie98/156.html>, (Eriřim Tarihi:řubat **2005**).
- [49] The IBM Agent, <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>, (Eriřim Tarihi: Ocak **2005**).
- [50] Thalmann, D.;Noser,H.; Huang, Z.: “Autonomous Virtual Actors Based on Virtual Sensors. In: Creating Personalities for Synthetic Actors: towards autonomous personality agents”, Trappl,R.; Petta,P. (Eds.). Berlin: Springer Verlag, **(1997)**.
- [51] Rickel J.; Johnson W.L.: “Intelligent Tutoring in Virtual Reality: A Preliminary report”, AI-ED97 : *Proc. of Eighth World Conference on AI in Education*, Kobe: Japan **(1997)**.
- [52] Lewis, W J.; Shaw, E. ,”Using Agents to Overcome Deficiencies in Web-Based Courseware”, AI-ED97: *Proc. of Eighth World Conference on AI in Education*, Kobe: Japan **(1997)**.
- [53] Hospers, M., E.; Kroezen, A. Nijholt, R.; Akker & D. Heylen.. “Developing a generic agent-based intelligent tutoring system and applying it to nurse

- education”, *Proceedings IEEE International Conference on Advanced Language Technologies (ICALT'03)*, Athens, Greece **(2003)**.
- [54] Ohlsson, S: “Some Principles of Intelligent Tutoring”, In Lawyer & Yazdani (Ed.), *Artificial Intelligence and Education*, Volume 1, Norwood **(1987)**.
- [55] Jumeily, D.AI; Strickland, P: “Designing an interface on the Web for an intelligent tutoring system” , *23rd EUROMICRO Conference '97 New Frontiers of Information Technology* , Budapest, HUNGARY **(1997)**.
- [56] Intelligent Tutoring Systems, <http://home2.pacific.net.sg/~audrick/its.html>, (Erişim Tarihi: Haziran **2005**).
- [57] Shortliffe, E.H.: “Computer-based Medical Consultations: MYCIN”, American Elsevier Publishers, NewYork **(1976)**.
- [58] Clancey, W: “Knowledge-based Tutoring”, *The GUIDON Program*. MIT Pres **(1987)**.
- [59] Rafael, A. F; Marta, C. R; Fernando, A.G.: “An Approach of Student Modelling in a Learning Companion System”,*Lecture Notes in Computer Science: Advances in Artificial Intelligence – IBERAMIA 2004: 9th Ibero-American Conference on AI*, Puebla, Mexico, November 22-25**(2004)**.
- [60] Anjaneyulu, K.: “Concept Level Modelling on the WWW”, *Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web"*, 8th World Conference of the AIED Society, Kobe, Japan, 18-22 (August **1997**).
- [61] Han, J.; Kamber, M.: “Data Mining Concept and Techniques”, Morgan Kaufmann Publishers, 1st Ed., San Fransisco USA, **(2000)**.
- [62] Zhi-Hua, Z.: “Three perspectives of data mining”, *Artificial Intelligence* **143 (2003)** 139–146.
- [63] Hua, Y.; Chena R.; Tzeng G.: “Finding fuzzy classification rules using data mining techniques”, *Pattern Recognition Letters* **24 (2003)** 509–519 .
- [64] Berson, S. : “Data Warehousing, Data Mining, & OLAP”, McGraw-Hill, **(1997)**.
- [65] Fayad, U. M.; Piatetsky-Sjapiro, G.; Smyth, P.: “From Data Mining to Knowledge Discovery: An overview”. In Fayard, Piatetsky-Sjapiro, Smyth, and Uthurusamy eds., *Knowledge Discovery in Databases*, AAAI/MIT Press, **(1996)**.
- [66] Lin, T.Y.: “Issues in Modeling for Data Mining”, *Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02) IEEE* **(2002)**.

- [67] Simoudis, E.; Livezey, B.; Kerber, R.: "Integrating Inductive and Deductive Reasoning for Data Mining", *Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence, Claifornia*,(1996) 353-373.
- [68] Witten, H.A.; Frank, E.: "Data Mining, Practical Machine Learning Tools and Techniques with JAVA Implementations", Morgan Kaufmann Publishers, (2000) .10-54.
- [69] Wu, H.: "Choice of techniques in data mining", *PhD Thesis*, Faculty of Art and Sciences of American University in Information Systems, Washington (1999).
- [70] Fayyad,U.; Shapiro,G.P.; Smyth,P.: "Knowledge Discovery and Data Mining: Towards a Unifying Framework", *Proceedings of the second Conference on Knowledge Discovery and Data Mining(KDD-96)*, Portland, Oregon, (1996)
- [71] Chen,M.; Han,J; Yu,P: "Data Mining: An Overview From Database Perspective.", *IEEE Transactions on Knowledge and Data Eng.*, 8 (6) (1996) 866-883.
- [72] Akpınar, H.: "Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği", *İstanbul Üni. İşletme Fakültesi Dergisi*, C:29 S:1 (Nisan 2000).
- [73] Kahaed, A.A.: "Efficient Algorithms for Data Mining", *PhD Thesis*, Computer and Information Science in Graduate School of Sycracuse University,(1998).
- [74] Özekes, S., "Veri Madenciliği Uygulaması", *Yüksek Lisans Tezi*, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik-Bilgisayar Eğitimi, (2002).
- [75] Agrawal, R.; Srikant R: "Fast Algorithms for Mining Association Rules", "*Proc. 20th Int. Conf. Very Large Data Bases, {VLDB}*", (1994) 487-499.
- [76] Thomas J., "Adaptive Algorithms for Intelligent Association Mining", *PhD Thesis*, University of Lousiana at Lafayette, (2001).
- [77] Iliskovic, S.A.: "Data mining in Databases: An extended Decision Tree Approach and Methodology in Database Environment" , *PhD Thesis*, University of Alberta, in Information Systems, Alberta (2000).
- [78] Kantardzic, M.: "Data Mining - Concepts, Algorithms", Wiley-IEEE. Computer Society Press, (2003) 2-62.
- [79] Agrawal, R.; Srikant, R.: "Mining Sequential Patterns", *11th International Conference on Data Engineering*,Taipei, Taiwan (1995).

- [80] Sever, H.; Oğuz, B.: “Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım: Kısım 2- Eşleştirme Sorgularının Biçimsel Kavram Analizi ile Modellenmesi” *Bilgi Dünyası*, 4(1): **(2003)** 15-44 6.
- [81] Agrawal, R.; Imielinski, T.; Swami, A.: “Mining association rules between sets of items in large databases”, *ACM SIGMOD Conference on Management of Data*, **(1993)**.
- [82] Agrawal, R.; Srikant, R.: “Fast Algorithms for Mining Association Rules”, *Proc. 20th Int. Conf. Very Large Data Bases*, 487-499, **(1994)**.
- [83] Srikant, R.; Agrawal, R.: “Mining Quantitative Association Rules in Large Relational Tables”, *The ACM SIGMOD Conference on Management of Data*, Montreal, Canada. **(1996)**.
- [84] Han J.: “Data Mining with Microsoft SQL Server 2000”, *Microsoft Pres*, 1st Ed. , Washington, USA, **(2001)**.
- [85] Jain, A.K.; Murty, M.N.; Flynn, P.J.: “Data Clustering: A Review”, *ACM Computing Surveys*, Vol. 31, No. 3 (September **1999**).
- [86] Berkhin, P. : “Survey of Clustering Data Mining Techniques.”, *Accrue Software Inc.*, USA (2002).
- [87] Han, J.; Kamber, M.; Tung, A. K. H.: “Spatial Clustering Methods in Data Mining: A Survey”, in H. Miller and J. Han (eds.), *Geographic Data Mining and Knowledge Discovery*, **(2001)**.
- [88] Srikant, R.; Vu, Q.; Agrawal, R.: “Mining Association Rules with Item Constraint”, *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, **(1997)** 67-73.
- [89] MacQueen, J.: “Some Methods for classification and Analysis of Multivariate Observations”, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, **(1967)** 281-297.
- [90] Hartigan, J.: “A k-means clustering algorithm”. *Applied Statistics*, 28, **(1975)** 100-108
- [91] Jain, A.K.; Murty, M.N.; Flynn, P.J.: “Data Clustering: A Review”, *ACM Computing Surveys*, Vol. 31, No. 3., (September **1999**)
- [92] <http://www.ecn.ulaval.ca/~guy/13543/Textbook/stcluan.html#d>, Cluster Analysis)(Erişim Tarihi: Mart **2006**).

- [93] Su, M.; Chou, C.: “A modified version of the K-means algorithm with a distance based on cluster symmetry”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 23*, Issue 6, (June **2001**) 674 – 680.
- [94] Zhong, W.; Altun, G.; Harrison, R.; Tai, P.C.; Yi P.; “Improved K-means clustering algorithm for exploring local protein sequence motifs representing common structural property” *NanoBioscience IEEE Transactions on Volume 4*, Issue 3, (Sept. **2005**) 255 – 265.
- [95] The k-means clustering algorithm, <http://cms1.gre.ac.uk/research/cassm/DMQuick/DCS2003Lect3.ppt> (Eriřim Tarihi: Ekim **2005**).
- [96] Pak, C.H.: “Visual Data Mining”, *IEEE Computer Graphics and Applications*, (September/October **1999**) 2-3.
- [97] Keim, D.A.: “Information visualization and visual data mining”, *IEEE Transactions on Visualization and Computer Graphics*, Volume 8, Issue 1, (Jan.-March **2002**) 1 – 8.
- [98] Inselberg, A: “N-Dimensional Graphics”, Part I - Lines and Hyperplanes in IBM LASC Tech. Rep. G320-2711. IBM LA Scientific Center, (1981).
- [99] Inselberg, A; Dimsdale, B.: “Parallel coordinates: a tool for visualizing multidimensional geometry”, *IEEE Visualization 1990 Proceedings*, (October **1990**) 361-378.
- [100] Visualization and Optimization, <http://catt.bus.okstate.edu/jones98/parallel.html>, (Eriřim Tarihi:Eylöl **2006**).
- [101] An Investigation of Methods for Visualising Highly Multivariate Datasets, <http://www.agocg.ac.uk/reports/visual/casestud/brunsdon/abstract.htm>, (Eriřim Tarihi: Eylöl **2006**).
- [102] Hauser, H.; Ledermann, F.; Doleisch, H.: “Angular brushing of extended parallel coordinates” *IEEE Visualization 2002 Proceedings*, (October **2002**).
- [103] Parvis Parallel Coordinates Visualisation, <http://home.subnet.at/flo/mv/parvis>, (Eriřim Tarihi: Aęustos **2006**).
- [104] Minaei-Bigdoli, B.; Kortemeyer, G.; Punch, W.F.: “Enhancing Online Learning Performance: An Application of Data Mining Methods”, *The 7th IASTED International Conference on Computers and Advanced Technology in Education (CATE 2004)*, , Kauai, Hawaii, USA, (August **2004**) 173-178.

- [105] Merceron, A.; Yacef, K.: “ TADA-Ed for Educational Data Mining”, *Interactive Multimedia Electronic Journal of Computer Enhanced Learning*, Volume 7, Number 1, (May **2005**).
- [106] Romero, C.; Ventura S.; Hervás, C.; González, P.: “Rule Mining with GBGP to improve web-based adaptive educational systems” In: C. Romero and S. Ventura (Eds.), *Data mining in e-learning. Advances in Management Information Series 4*, WitPress, (**2005**).
- [107] Yu, P.; Own, C.; Lin, L.: “On learning behavior analysis of web based interactive environment” , *International Conference ICCEE*, Oslo, (**2001**).
- [108] Shen R.; Han P., Yang F.: “Data Mining and Case-based Reasoning for Distance Learning”, *International Journal of Distance Education Technologies*, Vol. 1(3), (July-Sept **2003**) 46-58.
- [109] Gasar, S.; Bohanec, M.; Rajkovic, V.: “Combined data mining and decision support approach to the prediction of academic achievement”, *ECML/PKDD'02 workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*,. University of Helsinki, (August **2002**) 41—52.
- [110] Tang, T.Y.; Chan, K.: “Feature Construction for Student Group Forming Based on Their Browsing Behaviors in an E-learning System”, *Lecture Notes In Computer Science; Vol. 2417 archive Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence* (**2002**) 512-521.
- [111] Chen, G-D.; Ou, K-L. ; Liu, C-C.; Liu, B-J: “Intervention and strategy analysis for web group-learning”, *Journal of Computer Assisted Learning, Special Issue: Advanced information technologies for learning in the Asia-Pacific Region* Vol. 17, No. 1, (March **2001**).
- [112] Zaïane, O. R.: “Web Usage Mining for a Better Web-Based Learning Environment”, *Technical Report*. (**2001**).
- [113] Myller, N.; Suhonen J.; Sutinen, E.: “Using Data Mining for Improving Web-Based Course Design”, *IEEE Proceedings of the International Conference on Computers in Education (ICCE'02)*, (**2002**).
- [114] Zaïane, O. R; Luo J.: “Towards Evaluating Learners' Behaviour in a Web-Based Distance Learning Environment”, *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, (**2001**).

- [115] Shortliffe, E.H.; Buchanan, B.G.: “Summary on Certainty Theory, A Model of Inexact Reasoning in Medicine”, *Mathematical Biosciences*, vol. 23 (1975) 351-379.
- [116] Stankov, S.: “Student Model Developing for Intelligent Tutoring Systems , “*International Journal for ENGINEERING MODELLING* “, Vol. 9, No. 1-4, (1996).
- [117] Castillo, G.; Breda, A.M.; Bajuelos, A.L.: “Towards a Prototype of a User Model for an Adaptive Courseware in Geometry”, *Proceedings of the International Conference CINTEC*, Aveiro, (2001) 299-309.
- [118] <http://www.dmi.usherb.ca/~amayers/Cours/ift724/Durkinch12.pdf> (Eriřim Tarihi: Mart 2005).
- [119] Schwier, R.; Misanchuk, E.: “Interactive multimedia instruction”,. Englewood Cliffs, NJ: *Educational Technology Publications, Inc*, (1993).
- [120] Misanchuk, E.; Schwier, R.; Boling, E.: “*Visual design for instructional multimedia*”, (2000).
- [121] Muncer, S.J.; Gorman, B.S.; Gorman, S.; Bibel, D.: “Right is wrong: An examination of the effect of right justification on reading”, *British Journal of Educational Technology*, 1 (17), (1986), 5-10.
- [122] Shneiderman, B.: “Designing the user interface: Strategies for effective human-computer interaction” (3rd ed.) . *Reading, MA: Addison-Wesley Publishing.*, (1998).
- [123] Marcus, A.: “Principles of effective visual communication for graphical user interface design”, In Baecker, R. M., Buxton, W., & J. Grudin (Eds.) *Readings in Human-Computer Interaction: Toward the Year 2000*. San Francisco, CA: Morgan Kaufmann, (1995).
- [124] Heinich, R.; Melonda, M.; Russel, J.: “Instructional Media and The New Technologies of Instruction”, *Macmillan Publishing Company inc*, U.S.A, (1993).
- [125] Ruffini, M. F. “Systematic planning in the design of an educational web site”, *Educational Technology*. 40(3), (2000). 58-64.
- [126] Color Test Result, <http://webdesign.about.com/gi/dynamic/offsite.htm?site=http%3A%2F%2Fhubel.sfasu.edu%2Fresearch%2Fsurvreslts.html>, (Eriřim Tarihi: Mayıs 2005).

- [127] Srikant, R.; Agrawal, R. “Mining Generalized Association Rules”, *21st International Conference on Very Large Databases, VLDB'95*, Zurich, Switzerland, **(1995)**.
- [128] Rajkumar, N.; Karthik, M.R.; Sivanandam, S.N.; “Fast algorithm for mining multilevel association rules”, *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region Volume 2*, (15-17 Oct. **2003**), 688 - 692
- [129] Dunham, M.H. : “Data Mining and Advanced Topics”, *PrenticeHall*, **(2003)**.
- [130] Sarasere, A.; Omiecinsky, E.; Navathe, S.: “An efficient algorithm for mining association rules in large databases”, *In 21st Int'l Conf. on Very Large Databases (VLDB)*, Zurich, Switzerland, (Sept. **1995**).
- [131] Györödi, R. S.; “Acomparative Study of Iterative Algorithms in Association Rule Mining”, *Studies in Informatics and Control, Vol 12, No:3*, (September **2003**).
- [132] Kamber, M.; Han, J.; Chiang, J. Y.: “Metarule-guided mining of multidimensional association rules using data cubes”, *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD'97)*, Newport Beach, California, (August **1997**) 207—210.
- [133] Cheung, Y.M.: “k-Means: A New Generalized k-Means Clustering Algorithm”, *Pattern Recognition Letters, Vol. 24, Issue 15*, **(2003)** 2883—2893.