

154589

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

SQL TABANLI VERİ ANALİZİ

Matematik Müh. Ayşe Öncü UYSAL

F.B.E. Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Ayla ŞAYLI



Prof. Dr. Mehmet Ahlatcıoğlu Doç. Dr. Fatma Tiryaki



İSTANBUL, 2004

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	vii
ÖNSÖZ	viii
ÖZET	ix
ABSTRACT	x
1 GİRİŞ	1
1.1 İstatistik	1
1.2 SQL Veri Tabanı Genel Özellikleri	2
1.3 İlişkisel Veri Tabanlarında Programlama Dillerinin Kullanımı	4
1.3.1 Gömülmüş SQL (Embedded SQL)	4
1.3.2 Aktif Veri Nesneleri (ADO)	4
1.4 Visual Basic Programlama Dili	6
2 REGRESYON VE KORELASYON ANALİZİ	8
2.1 Korelasyon Analizi	8
2.1.1 Serpme Diyagramı	8
2.1.2 Korelasyon Katsayısının Hesabı	9
2.1.3 Korelasyon Katsayısını Hesaplayan Fonksiyon	10
2.2 Regresyon Analizi	11
2.2.1 Regresyon Denkleminin Hesaplanması	12
2.2.2 Regresyon Denklemini Hesaplayan Fonksiyon	12
2.3 Trend Analizi(Zaman Serileri Analizi)	14
2.3.1 Doğrusal Trend	14
2.3.2 Trend Doğrusun Çizimi	14
2.4 Çoklu Regresyon Analizi	17
2.4.1 Regresyonun Anlamlılığının Sınanması	18
2.4.2 Çoklu Belirginlik Katsayısı	19
2.4.3 Tekbiçimleştirilmiş Regresyon Katsayıları	20
2.5 DMRA Programı	21
3 SQL SORGULAMA DİLİ	25
3.1 SQL Server Tablosu	25
3.1.1 Tablo Oluşturulması(CREATE TABLE)	26
3.2 Sorgulama	27
3.2.1 SELECT Deyimi	28
4 ÖĞRENEN SİSTEMLER	36
4.1 Bilgiden Bilgi Öğrenme	36
4.2 Öğrenme Metotları	37
4.2.1 Siegel Yöntemi	37
4.2.2 Knoblock Yöntemi	39

4.2.3	Öğrenme Metotlarının Avantajları ve Dezavantajları	44
5	ELİMİNASYON METODUYLA KURAL ÖĞRENME.....	47
5.1	Tekbiçimleştirilmiş Regresyon Katsayıları Yardımıyla Veritabanı Eliminasyonu.....	47
5.2	Eliminasyon Metodu İle Kural Öğrenen Dinamik Sistem.....	49
5.3	Eliminasyon Metodu Kullanmaksızın Kural Öğrenen Dinamik Sistem	49
5.4	Eliminasyon Metodu İle Kural Öğrenen Dinamik Sistem.....	50
5.5	Eliminasyon Metodu Kullanmaksızın Kural Öğrenen Dinamik Sistem	53
5.6	Uygulama Sonuçları	56
5.6.1	1000 Datalık Tabloda Anahtar Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının ve Kural Sayılarının Kıyaslanması	56
5.6.2	10000 Datalık Tabloda Dizin(index) Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının Kıyaslanması	57
5.6.3	41046 Datalık Tabloda Özelliksiz(none) Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının Kıyaslanması	58
5.6.4	Genel Analiz	59
5.7	Kural Oluşturulması ve Kural Seti	60
5.7.1	Kural Tablosu Genel Özellikleri.....	60
6	SORGUNUN ÇALIŞTIRILMASI VE SORGU KALİTESİNİN TESTİ	64
6.1	Uygulama.....	65
6.2	Uygulama Sonuçları(1).....	66
6.3	Uygulama Sonuçları(2).....	75
7	DİĞER UYGULAMALAR.....	80
7.1	Çoklu Regresyon Analizini Kullanarak Kayıp Data Tahmin Yöntemi.....	80
7.1.1	Çoklu Regresyon Analizi ile Kayıp Data Tahmin Yöntemini Kullanan Dinamik Sistem	80
7.2	Çoklu Regresyon Analizi ile 1981-2002 Yılları Arasındaki Sigorta Primlerindeki Değişimlerin İncelenmesi	82
7.3	1981-2002 Yılları Arasındaki Sigorta Primlerindeki Değişimleri Kullanarak Yapılan Trend Analizi	86
8	SONUÇLAR.....	90
	KAYNAKLAR.....	91
	EKLER	92
Ek 1	EPISODE5 Tablo veri bilgileri (100 Kayıtlık).....	93
Ek 2	EPISODE5 Tablosu için eliminasyon metoduyla öğrenilen kurallar (İlk 10 sorgu için).....	96
Ek 3	EPISODE5 Tablosu için eliminasyonsuz öğrenilen kurallar (İlk 10 sorgu için)...	99
Ek 4	İnsan2 Tablosu veri bilgileri (İlk 30 kayıt).....	105
	ÖZGEÇMİŞ.....	107

SİMGE LİSTESİ

- @ Karşılaştırma operatör göstergesi
- > Kurallarda sol ve sağ tarafların ayrıldığı gösterge



KISALTMA LİSTESİ

ADO	Aktif Veri Nesnesi (Activex Data Objects)
ODBC	Açık Veritabanı Bağlantısı (Open Database Connectivity)
SQL	Yapısal Sorgu Dili (Structured Query Language)
SQO	Anlamsal Sorgu İyileştirmesi (Semantic Query Optimization)
DMRA	Dinamik Çoklu Regresyon Analizi (Dynamic Multiple Regression Analysis)
LWE	Eliminasyon Metoduyla Öğrenme (Learning With Elimination)
QE	Sorgu Çalıştırma (Query Execute)
FDM	Kayıp Data Bulma (Finding Missing Data)



ŞEKİL LİSTESİ

Şekil 1.1 ADO genel yapısı	5
Şekil 2.1 Korelasyon sayısı pozitifken serpm diyagramı	9
Şekil 2.2 Korelasyon sayısı negatifken serpm diyagramı.....	9
Şekil 2.3 Nüfus için trend doğrusu	16
Şekil 2.4 Trend doğrusu ve gözlenen değerler doğrusu	17
Şekil 2.5 DMRA programı	22
Şekil 2.6 Çalıştırıldıktan sonra DMRA programı.....	23
Şekil 5.1 Eliminasyon sistemiyle kural öğrenen sistemin çalışma metodolojisi.....	49
Şekil 5.2 Eliminasyon metodu kullanmaksızın kural öğrenen sistemin çalışma metodolojisi.	49
Şekil 5.3 LWE programı, veritabanı seçim ekranı	50
Şekil 5.4 LWE programı, tablo ve metot seçim ekranı	51
Şekil 5.5 LWE programı, eliminasyon ve kural öğrenilecek değişkenin seçim ekranı.....	52
Şekil 5.6 LWE programı, kural öğrenilmesi ve kuralların veritabanına yazılması ekranı	53
Şekil 5.7 LWE programı tablo ve metot seçimi(eliminasyonsuz).....	54
Şekil 5.8 LWE programı kural öğrenilecek değişkenin seçim ekranı	54
Şekil 5.9 LWE programı,kural öğrenilmesi ve kuralların veritabanına yazılması ekranı(eliminasyonsuz).....	55
Şekil 5.10 1000 data için öğrenme zamanları	56
Şekil 5.11 1000 data için kural sayıları	56
Şekil 5.12 10000 data için öğrenme zamanları	57
Şekil 5.13 10000 data için kural sayıları	57
Şekil 5.14 41046 data için öğrenme zamanları	58
Şekil 5.15 41046 data için öğrenme zamanları	58
Şekil 6.1 Sorgu kalitesinin testi, çalışma metodolojisi.....	64
Şekil 6.2 QE programı, sorgunun çalıştırılması ekranı	65
Şekil 6.3 EPISODE5 tablosu için özelliiksiz(none) değerler üzerinde çalıştırılan 118 sorgunun zamanı.....	74
Şekil 6.4 Kural tablosunun sol tarafının anahtar(key) olma durumunda çalışma zamanları....	77
Şekil 6.5 Kural tablosunun sol tarafının anahtar(key) olma durumunda çalışma zamanları....	78
Şekil 6.6 Çizelge 6.3'ün grafikte ifadesi	79
Şekil 7.1 FMD programı kayıp data tahmin ekranı	81
Şekil 7.2 DMRA programı,veritabanı seçim ekranı.....	83
Şekil 7.3 DMRA programı tablo seçim ekranı.....	83
Şekil 7.4 DMRA programı, regresyon analizi yapan ekran	84
Şekil 7.5 Üretilen toplam primin gözlenen değerleriyle tahmini değerleri	86
Şekil 7.6 Nüfus değerlerinin gözlenen değerleri ve tahmini değerleri.....	86
Şekil 7.7 Üretilen toplam prim gerçek ve tahmini değerleri	89

ÇİZELGE LİSTESİ

Çizelge 2.1 1981-2001 yılları arasında gözlenen nüfus değerler	15
Çizelge 2.2 1981-2001 yılları arasında hesaplanan tahmini değerler.....	15
Çizelge 3.1 Tabloornek tablosu ve veri bilgileri	29
Çizelge 3.2 Sütündaki tüm kayıtları getiren sorgu sonucu.....	29
Çizelge 3.3 Sütündaki tek kayıtları getiren sorgu sonucu	30
Çizelge 3.4 Sütunda en üstteki iki kaydı getiren sorgu sonucu.....	30
Çizelge 3.5 Seçilen sütunları getiren sorgu sonucu.....	30
Çizelge 3.6 Sütun başlığı değiştirilmiş sorgu sonucu.....	32
Çizelge 3.7 Koşul kullanılmış sorgu sonucu	33
Çizelge 3.8 Grup yapılmış sorgu sonucu.....	34
Çizelge 3.9 Grup koşulu kullanılmış sorgu sonucu.....	34
Çizelge 5.1 Tablo Test.....	48
Çizelge 5.2 Elimine edilmiş Tablo Test	48
Çizelge 5.3 Öğrenmeler için eliminasyonlu ve eliminasyonsuz metodun genel analizi	59
Çizelge 5.4 Kural seti sütunları veri tipleri	60
Çizelge 5.5 “Select * from episode5 where episodes=1” sorgusu için üretilen kurallar.....	61
Çizelge 5.6“Select * from episode5 where episodes=1” sorgusu için eliminasyon metodu uygulanarak üretilen kurallar.....	62
Çizelge 6.1 Eliminasyonlu, eliminasyonsuz ve orijinal metotların yüzde olarak kıyaslanması	74
Çizelge 6.2 Kural tablosu için belirlenen kriterler	76
Çizelge 6.3 Çizelge 6.2’deki kriterler için alınan zaman sonuçları.....	76
Çizelge 6.4 Metotların birbiriyle yüzde olarak kıyaslanması.....	79
Çizelge 7.1 1981-2002 yılları arasındaki sigorta primleri(dolar) değişimi	82
Çizelge 7.2 Çizelge 7.1 için hesaplanan tahmini değerler.....	85
Çizelge 7.3 1981-2001 yılları arasında üretilen toplam primin gözlenen değerleri	87
Çizelge 7.4 Çizelge 6.3’ün gözlenen değerleri için tahmini değerler	88
Çizelge Ek 1.1 EPISODE5 Tablo veri bilgileri (100 Kayıtlık).....	93
Çizelge Ek 2. 1 EPISODE5 Tablosu için eliminasyon metoduyla öğrenilen kurallar (İlk 10 sorgu için).....	96
Çizelge Ek 3. 1 EPISODE5 Tablosu için eliminasyonsuz öğrenilen kurallar (İlk 10 sorgu için)	99
Çizelge Ek 4. 1 İnsan2 Tablosu veri bilgileri (İlk 30 kayıt).....	105

ÖNSÖZ

Bilgisayarların her geçen gün hayatımıza daha çok girmesiyle beraber elimizde verileri saklayabileceğimiz, istediğimizde bu verilere ulaşabileceğimiz, arama yapabileceğimiz bir araç oluşmuştur. Verileri sağlıklı bir şekilde depolayabilmek, verilere ulaşımı kolaylaştırmak amacıyla veritabanı sistemleri kurulmuştur.

Bu tezde, istatistik metotlar kullanarak veritabanı üzerinde analizler yapılmıştır. İstatistik metotların veritabanlarına katkısı araştırılmıştır. Sorgu iyileştirilmesi için kendinden öğrenen dinamik bir sistem kurulmuş ve test edilmiştir.

Bu tezin hazırlanmasında yardımlarını ve desteğini esirgemeyen değerli hocamız Sayın Yrd. Doç. Dr. Ayla Şaylı'ya çok teşekkür ederim.



ÖZET

Bilgisayarların çıkışı ve gelişmesiyle her geçen gün biraz daha değişen ve gelişen bir dünyada yaşamaktayız. Bilgisayarlar yaşantımıza bir çok kolaylık katmakta, yapılan işlerin yükünü hafifletmekte daha iyi sonuçlara daha iyi ve daha kısa yollardan ulaşmamızı sağlamaktadır. Bilgisayarlar aynı işi otomatik olarak ve daha verimli yapacağından insan kaynaklı hatalar en aza indirgenir

Veritabanları çok sayıda veriyi depolayan sistemlerdir. Bu sistemlerde saklanan çok sayıda veri istatistik açısından oldukça büyük bir anlam ifade etmektedir. Çok sayıda veriyi depolayan sistemler sayesinde daha çok veri üzerinde uygulanacak olan istatistik metotlar bizi daha anlamlı bilgilere götürecektir. Daha önceleri az veri üzerinde yapılan istatistiksel analizleri boyutu binlere, milyonlara ulaşan verilerin oluşturduğu veritabanları üzerinde yapmak mümkün hale gelmiştir. Ayrıca bu istatistiksel analizler veritabanlarının daha hızlı çalışabilmesi amacıyla da kullanılabilir. İstatistik metotlar kullanılarak veriden çıkarılan bilgi, veriye daha hızlı ulaşmamızı veritabanının performansını yükseltmesi sağlanabilir.

Bu çalışmada SQL Server üzerinde kendinden öğrenen bir sistem yapısının kurulmasına istatistiksel metotların katkısı araştırılmış ve ispatlanmıştır. Ayrıca istatistiksel metotlar yardımıyla veritabanında kayıp data için tahmin yapılmıştır, ekonomik veriler üzerinde zaman serileri analizi yapılmıştır.

Anahtar Kelimeler: Veritabanı, SQL, kurallar, ADO, istatistik, korelasyon, regresyon, çoklu korelasyon, çoklu regresyon.

ABSTRACT

After the invention of computers our world is getting more different. Computers make our life easier and improve the quality of our life. We can get better results with using shorter time and better ways. This reduces human errors since computers can be used for the same work automatically and effectively.

Databases are systems which contains lots of data in. These data has very important meaning for statistics. Making analysis on very large data with statistics methods is getting easier and results of the analysis are more meaningful with the help of systems which contains lots of data in. Now, in today's world we can make analysis on very large databases which has thousand or million records. Also the statistical analysis can be used to improve the performance of the databases. The information which is the result of analysis can improve the performance of the system and make the system faster.

In this thesis, we study about the help of the statistics methods to make a system, which is learning in its self, on SQL Server. Also we study about finding missing data and make trend analysis on economical data.

Keywords: Database, SQL, rules, ADO, statistics, correlation, regression, multiple correlations, multiple regressions.

1 GİRİŞ

Veritabanları içlerinde bir çok veriyi barındıran sistemlerdir. Bu sistemler teknolojinin ve bilgisayarların gelişmesiyle birlikte gün geçtikçe daha çok veriyi saklayabilir hale gelmiştir.

Bu istatistik açısından oldukça önemlidir çünkü daha çok veriyi saklayan veritabanları sayesinde veriye ulaşmak, analizini yapmak ve bu veriden bilgi çıkarmak daha kolay bir hale gelmiştir. Ancak veri sayısının artması veritabanı üzerinde işlem yapmayı, sorgulamayı yavaşlatan bir unsurdur. Daha çok veriyi saklayabilen büyük veritabanları istatistik metotları daha rahat ve kolay kullanılması açısından önemli bir unsurdur.

Bu çalışmada daha rahat ve kolay kullanılabilen istatistik metotların veritabanlarının performansına katkıda bulunup bulunamayacağı araştırılacaktır. İstatistik metotların veri işleme komutlarının performansını artırma yolları araştırılacaktır. Bunun için istatistik metotlar veriler üzerinde uygulanır, uygulamalardan çıkarılan bilgiler ışığında veritabanı için belli kriterlere uygun olarak kurallar üretilir. Üretilen bu kuralların zaman ve performans açısından karlılığı hesaplanır.

Bu çalışma SQL Server veritabanı kullanılarak, Visual Basic programlama ortamında, ADO veri iletişim protokolü kullanılarak, istatistik metotlar yardımıyla geliştirilmiştir.

1.1 İstatistik

İstatistiğin pek çok bilimsel ve bilimsel olmayan tanımı mevcuttur. Dar anlamda İstatistik, geçmiş ve şimdiki durumla ilgili toplanmış sayısal verileri geliştirilmiş olan bazı tekniklerle analiz ederek gelecek hakkında karar vermemizi kolaylaştıran bir bilim dalıdır. 17. Yüzyıla kadar sadece bilgi kaydetme şeklinde gerçekleşen istatistiki çalışmalar, 18. ve 19. Yüzyıllarda J. Bernoulli (1645-1705) ve K. Gauss'un (1777-1855) katkılarıyla matematik temelleri üzerine oturtulmuş, ihtimal teorisi geliştirilmiştir. Sosyal ve antropolojik olaylara istatistiği kapsamlı bir şekilde uygulayan ilk matematikçi olan Adolphe Quételet (1796-1874) ise modern istatistiğin kurucusu olarak kabul edilmiştir. 20. Yüzyılın başında R. A. Fisher, K.

Pearson ve W. S. Gosset'in katkılarıyla tahmin yapma ve karar verme konuları ön plana çıkarak istatistik artık sayısal verilerin yorum ve değerlendirmesini yapan bir bilimsel metotlar topluluğu haline gelmiştir. (Çil 2002)

İstatistik bireylerin iki özelliğe sahip olması gerekir.

- i. İstatistik bireyleri sayılmaya, tartılmaya ve ölçülmeye elverişli olmalıdır.
- ii. İstatistik bireyleri aynı tür olmalıdır.

Karşılaşılan bazı olayları ve araştırma sonuçlarını değerlendirmede matematik modellerden yararlanmanın yanı sıra, bazı belirsizlik durumlarında sonuca ulaşılmasını kolaylaştıran istatistiği uygularken sahip olunması gereken birtakım bakış açıları mevcuttur.

Hemen hemen her iş kolunda ve kademesinde, çalışanlar ve yöneticiler, üretecekleri ürünün belirlenmesinde, veya çalışmalarını yönlendirecekleri hedef kitlenin seçiminde bazı araştırmalar yapmak zorundadırlar. Bilgi toplama, örnek seçimi, değerlendirme ve sonuca varma aşamalarında karşılaşılan safhalar, en basit anlamda aşağıdaki gibi gruplanabilir.

- i. Rakamsal Bilgileri Anlamlandırmak
- ii. Belirsizlik ile İlgilenmek
- iii. Örneklem
- iv. İlişkileri Analiz Etme
- v. Tahmin

1.2 SQL Veri Tabanı Genel Özellikleri

RDBMS büyük miktarlardaki verilerin güvenli bir şekilde tutulabildiği, bilgilere hızlı erişim imkanlarının sağlandığı, bilgilerin bütünlük içerisinde tutulabildiği ve birden fazla kullanıcıya aynı anda bilgiye erişim imkanının sağlandığı yapıdır. SQL veritabanı da bir RDBMS yapısındadır. Genel özellikleri aşağıdaki gibidir.

- i. Büyük miktarda veri tutabilme ve verilerin depolandığı alanları ayarlayabilme imkanı sağlar.

- ii. Aynı anda çok sayıda kullanıcıya birden verilerin bütünlüğünü bozmadan hizmet verebilmektedir.
- iii. Günün 24 saati ve haftalar boyu hiç kapatılmadan çalışabilmektedir.
- iv. İşletim sistemi, veri erişim dilleri ve ağ iletişim protokolleri standartlarıyla uyumludur.
- v. Yetkisiz erişimleri engelleme ve kontrol edebilme imkanı sağlamaktadır.
- vi. Bütünlüğü veritabanı düzeyinde sağlayabilmektedir, böylece daha az kod yazılmaktadır.
- vii. İstemci/Sunucu mimarisinin bütün avantajlarını kullanabilmektedir.

SQL ile ilk defa karşılaşan kullanıcılar genellikle Delphi, Visual C++, Visual Basic gibi görsel programlama dillerine benzeyen uygulamalarla karşılaşmayı umarlar. RDBMS bir programlama dili değildir. Bunun için SQL tarafından geliştirilen ve SQL'ın kendi uygulama geliştirme araçları içerisinde kullanılan bazı programlama dilleri vardır. SQL ürünleri genellikle büyük çaplı veri kontrolünü gerektiren uygulamalarda kullanılır.

SQL veritabanının, işletim sistemi tarafından bakıldığında, biri fiziksel diğeri mantıksal olmak üzere iki bölümü vardır. Fiziksel bölüm, işletim sisteminden görünen kısımdır. Bunlar Veri(Data) dosyası, Kontrol(Control) Dosyası ve Kütük(Log) dosyalarından oluşmaktadır. Mantıksal Bölüm, bir ya da daha fazla tablo alanı(Tablespace) ve tablolar(table), görüntüler(view), sıralar(sequence), eşanlamlar(synonym), dizinler(index), kümeler(cluster), veritabanı bağlantıları (database link), prosedürler(procedure), fonksiyonlar(function), ve paketlerden(package) oluşan şema nesnelerinden oluşmaktadır. Fiziksel bölüm işletim sistemi tarafından görülebilmesine rağmen, mantıksal bölüm ancak SQL'a bağlanıp, SQL komutları çalıştırılarak görülebilmektedir. Yani, SQL kurulu herhangi bir makinede, SQL bilgisi olmayan bir insan, SQL'in sadece fiziksel bölümünü görebilmektedir.

SQL veritabanındaki her nesnenin bir sahibi vardır. Her kullanıcı bir veya daha fazla tablo uzayına sahip olabilir. Her nesne, ait olduğu kullanıcının herhangi bir nesne uzayında (mantıksal olarak) bulunur. Her nesne uzayı da, kendisine sahip olan kullanıcının nesnelerini tutmak için işletim sisteminde bir veya daha fazla veri dosyasına sahiptir.

Sonuç itibarıyla, veritabanındaki her nesnenin bir kullanıcısı vardır ve bu nesneler mantıksal olarak o kullanıcının sahip olduğu nesne uzaylarının herhangi birinin içerisindedir, fiziksel

olarak da o kullanıcının sahip olduğu nesne herhangi bir veri dosyasında bulunur. Fakat, o veri dosyasının içerisine işletim sistemi üzerinden bulunamaz. Bu nesnenin sahibi ve mantıksal yeri sadece 'DML' komutları ile bulunabilmektedir.

1.3 İlişkisel Veri Tabanlarında Programlama Dillerinin Kullanımı

Programlama dillerinden veritabanlarına erişim için iki ayrı yol bulunmamaktadır. Bunlar Gömülmüş SQL üzerinden (Embedded SQL) ve Aktif Veri Objesi üzerinden (ADO - Activex Data Objects) kullanmaktır. Bu araştırmada ADO yapısı kullanıldığı daha ayrıntılı anlatılacaktır.

1.3.1 Gömülmüş SQL (Embedded SQL)

SQL'in programlama dilleri içerisinde kullanıldığı oluşuma Gömülmüş SQL denir. Bu diller PASCAL, COBOL, C/C++, JAVA..... olabilir. Uygulamalar dillerin özellikleri ile oluşturulur, dosyaları ise veritabanında tutulur.

Bu yapıda SQL ifadeleri diller içinde genellikle EXEC SQL komutunun devamı olarak verilir.

Örneğin: Burada <veritabanı> veritabanı ismi, <:host variable> sunucu değişkeni verilerek veri tabanı bağlantısı sağlanır. Kullanımı aşağıda verilmiştir:

EXEC SQL AT <dbname> <SELECT İfadesi>

veya

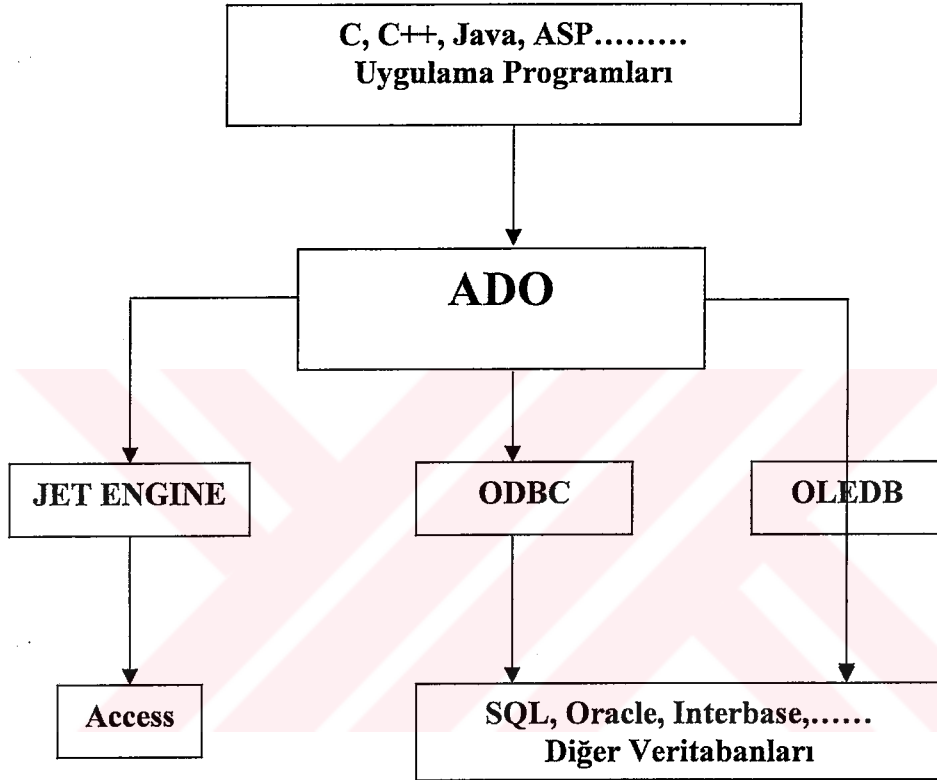
EXEC SQL AT <:host variable> <SELECT İfadesi>

Bu çalışmada SQL komutları Visual Basic 6.0 içinde kullanılmış, bu komutlarla veritabanına bağlantı sağlanmış ve veri analizleri bu komutlar yardımıyla yapılmıştır.

1.3.2 Aktif Veri Nesneleri (ADO)

Microsoft Windows 95 ile standart olarak gelen aktif veri nesnesidir ve veritabanına erişim yapısıdır. ADO'da veritabanına erişim üç yolla gerçekleştirilir:

- i. Microsoft Jet Engine
- ii. Açık Veritabanı bağlantısı (ODBC – Open Database Connectivity),
- iii. Doğrudan Veritabanına erişim (OLEDB).



Şekil 1.1 ADO genel yapısı

Microsoft Jet Engine ya da kısaca jet, Microsoft Access dosyalarına erişmek için kullanılır. Bu dosyanın yerel bilgisayarda olması da zorunlu değildir, ağ üzerindeki başka bir bilgisayarda bulunan bir Microsoft Access dosyasına Jet aracılığıyla erişim yapmak mümkündür.

ODBC, yalnızca Access formundaki değil: dBase, Oracle, Paradox, Excel vs. biçimlerindeki veritabanlarına da ulaşabilmek için kullanılan bir arabirimdir. Bağlantı bu ara birim üzerinden geçerek veri tabanına gider.

OLEDB, veritabanlarına erişim yöntemlerinin üçüncüsü ise doğrudan erişim yöntemidir. Bu yolla, Jet ve ODBC gibi, aslında birer bürokrasi olan arabirimler ortadan kaldırılmakta ve veriye doğrudan erişim sağlanmaktadır. Yetki verilmiş olması halinde, yeryüzünün herhangi bir noktasındaki bir sunucunun veritabanına, OLEDB ile doğrudan erişim mümkündür.

Bu araştırma çalışması, Visual Basic programlama dilinden SQL veritabanına bağlantı ADO-ODBC üzerinden yapılarak gerçekleştirilmiştir

1.4 Visual Basic Programlama Dili

Visual Basic(VB olarak da anılır) bir programlama dilidir. Visual Basic'in bir programın yazılıp çalıştırılmasından daha çok, program geliştirmek için çok sayıda araca sahip olması;kullanıcı arabiriminin tasarlanması, hataların giderilmesi, veritabanı yaratmak, farklı türde programlar üretmek gibi genel özellikleri, onu bir "programlama ortamı" ya da "program geliştirme ortamı" olarak tanımlanmasına sebep olmaktadır.

Visual Basic, çok amaçlı ve kolay bir şekilde program geliştirmemizi sağlayan bir programlama ortamıdır. Visual sözcüğü Windows işletim sisteminin grafik ara biriminin(GUI) özelliklerinden gelir. Visual olmak(görsellik), simgelerle yapılan iletişimi ifade eder. Bu ortamda işlemlerin yapılması için fare(Mouse) işlemleriyle birçok işin yapılması sağlanır. Visual Basic programlarında ortamın tasarımı öncelikli olarak yapılan işlemdir. Kod yazmak ikinci planda kalır.

Visual Basic, sadece program geliştirme işlemi ile sınırlı değildir. Visual Basic, VBA(Visual Basic Application) olarak bilinen ve Microsoft Office içinde;Microsoft Excel, Word ve Access programlarında da kullanılır Ayrıca Internet uygulamalarında kullanılan VBScript de Visual Basic'in bir alt setidir. Visual Basic ile farklı düzeylerde(konularda) programlar geliştirilebilir. (Çubukçu 1998)

Visual Basic'in sahip olduğu araçlar

- i. Kullanıcı ara birimi geliştirmek için çok sayıda kontrol
- ii. Programı derleme ve hata bulmaya yönelik çok sayıda araç
- iii. Değişik veritabanlarına erişim

- iv. SQL Server ve diğer enterprise düzeyindeki veritabanlarına erişim
- v. Diğer uygulamaların fonksiyonlarını kullanmayı sağlayan ActiveX teknolojisi
- vi. Uygulamanın EXE dosya olarak üretilebilmesi ve dağıtılması
- vii. Raporlama araçları
- viii. Veri yapısı araçları
- ix. Web uygulaması geliştirme araçları
- x. Çok katmanlı programlama ve Microsoft Transaction Server

Visual Basic ile neler yapabiliriz

- i. Basit ve genel amaçlı uygulamalar
- ii. Oyunlar
- iii. Şirketler için geliştirilen özel uygulamalar
- iv. Ticari programlar, satış sistemleri
- v. Veritabanı uygulamaları
- vi. Bilimsel uygulamalar
- vii. İstatistik uygulamalar

Bu araştırmada Visual Basic programlama ortamında istatistik metotlar kullanarak veritabanı uygulamaları geliştirilmiştir.

2 REGRESYON VE KORELASYON ANALİZİ

2.1 Korelasyon Analizi

En az iki değişken arasında olan ilişkinin incelenmesine korelasyon denir. Korelasyon iki değişkenli ya da daha çok değişkenli olabilir. İkidenden fazla değişken arasında olan korelasyona çoklu korelasyon adı verilir. Değişkenler arasında olan korelasyon eğrisel ya da doğrusal olabilir. Kelime anlamı olarak korelasyon ilişki, münasebet anlamına gelmektedir. İki değişkenli korelasyona örnek vermek gerekirse boy ve kilo örneğini verebiliriz. Boy ve kiloyu gözlenen değerlerden oluşmuş iki seri olarak düşünersek, kilo bağımlı değişken $Y(i)$, boy ise bağımlı değişken $X(i)$ dir. Çoklu korelasyona örnek vermek gerekir ise kilo, boy, yaş, cinsiyet olarak düşünebiliriz. Burada bağımlı değişken yine kilo $Y(i)$, bağımsız değişkenler ise boy, yaş ve cinsiyet olur.

Değişkenler arasındaki ilişkinin derecesini gösteren katsayıya korelasyon katsayısı adı verilir. Sembolü “r” dir. Korelasyon katsayısı daima +1 ile -1 aralığındadır. Korelasyon katsayısının sıfır çıkması değişkenler arasında herhangi bir ilişkinin olmadığını gösterir. Eğer serilerdeki değişme aynı yönde ise katsayı pozitif çıkar, değişme ters yönde ise negatif çıkar. Katsayı +1’e veya -1’e ne kadar yakın ise değişkenler arasındaki ilişki o kadar güçlüdür. Örneğin kilo ve boy arasındaki ilişki 0.95, kilo ve yaş arasındaki ilişki 0.80. Bu durumda diyebiliriz ki kilo, boya yaşa oranla daha bağımlıdır, kilo ve boy arasındaki ilişki kilo ve yaş arasındaki ilişkiden daha güçlüdür.

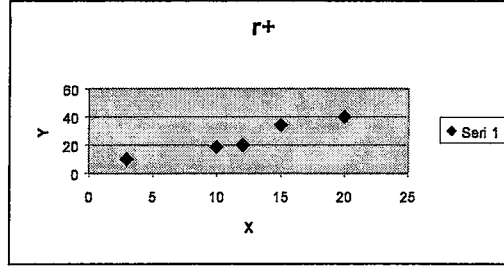
2.1.1 Serpme Diyagramı

Bir korelasyon problemi çözülürken ilk yapılacak işlerden biri serpme diyagramı çizimidir. Serpme diyagramı; bağımlı ve bağımsız değişkenlere ait noktaların serpilmesiyle elde edilir. Serpme diyagramı değişkenler arasındaki ilişkinin ne yönde ve ne derecede olduğunu görmemizi sağlar.

Diyagramda x ekseninde bağımsız değişken X_i değerleri, y ekseninde bağımlı değişken Y_i değerleri gösterilir.

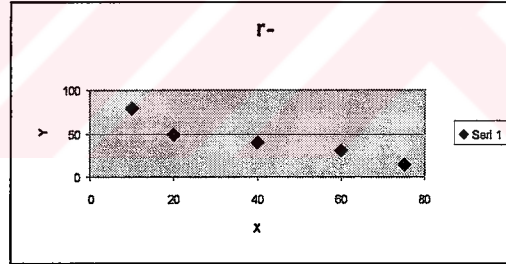
Serpme diyagramındaki noktaların seyrediş şekillerine göre korelasyon katsayısı hakkında yorum yapmak mümkündür. Örneğin eğer noktalar gövdede sol alt köşeden sağ üst köşeye doğru seyrediyor ise korelasyon katsayısı pozitif çıkar, tam tersi durumda negatif çıkar.

Korelasyon katsayısı pozitifken serpme diyagramı şekil 2.1'deki gibidir.



Şekil 2.1 Korelasyon sayısı pozitifken serpme diyagramı

Korelasyon katsayısı negatifken serpme diyagramı şekil 2.2'deki gibidir.



Şekil 2.2 Korelasyon sayısı negatifken serpme diyagramı

2.1.2 Korelasyon Katsayısının Hesabı

En az iki değişken arasındaki ilişkinin derecesini gösteren katsayıya korelasyon katsayısı denir. Standart sapma ve aritmetik ortalama iyi bilindiği takdirde korelasyon katsayısının hesabı kolaylaşır.

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{N \cdot S_x \cdot S_y}$$

(2.1)

Burada

X_i :Bağımsız değişken değerleri

Y_i :Bağımlı değişken değerleri

$\bar{X}$:Bağımsız değişkenin aritmetik ortalaması

$\bar{Y}$:Bağımsız değişkeni aritmetik ortalaması

N :Çiftleşen birey sayısı

S_x :Bağımsız değişkenin standart sapması

S_y :Bağımlı değişkenin standart sapması

Burada S_x ve S_y 'nin hesaplanmış şekilleri ise şu şekildedir.

$$S_x = \sqrt{\frac{\sum (X_i - \bar{X})^2}{N}}$$

(2.2)

$$S_y = \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{N}}$$

(2.3)

Bu durumda korelasyon formülü şu şekle dönüşür

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \cdot \sum (Y_i - \bar{Y})^2}}$$

(2.4)

2.1.3 Korelasyon Katsayısını Hesaplayan Fonksiyon

Kiloboy tablosundaki kilo ile boy arasındaki ilişkiyi hesaplayan fonksiyon

Private Function korelasyon(r, kayitadet)

```

Dim xort, yort, a, b, z, t, k, i, d, e, f, c, g
rst.Open "select count(*), avg(kilo),avg(boy) from kiloboy ", cnn, 3, 3
yort = rst(1)
xort = rst(2)
kayitadet = rst(0)
rst.Close
rst.Open "select boy , kilo from kiloboy", cnn, 3, 3
For i = 1 To kayitadet
d = rst(0)
e = rst(1)
c = c + ((d - xort) * (e - yort))
f = f + (d - xort) ^ 2
g = g + (e - yort) ^ 2
rst.MoveNext
Next
rst.Close
r = c / ((f * g) ^ (1 / 2))
End Function

```

Bu fonksiyonda xor bağımsız değişkenlerin aritmetik ortalamasını, yort bağımlı değişkenlerin aritmetik ortalamasını göstermektedir. Kayitadet ise çiftleşen birey sayısıdır.

2.2 Regresyon Analizi

İki değişken arasındaki ilişkiye korelasyon diyoruz. Regresyon ise iki değişken arasındaki ilişkinin denklem ile ifadesidir. Değişkenler arasındaki ilişki ifade edilebiliyor ise, bilinen değişkenler yardımıyla bilinmeyen değişkenleri bulabiliriz. Burada amaç, bir serpm diyagramındaki noktalara en yakın yerden geçen çizgiyi cebirsel bir fonksiyon ile sağlayan deklami saptamaktır. Bu çizgi regresyon çizgisi, denklem ise regresyon denklemidir. Regresyon denklemini $X(i)$ deki bir birimlik değişmeye karşı bağımlı değişken $Y(i)$ de meydana gelecek ortalama değişikliği açıklar. Eğer değişken sayısı ikiden fazla ise bu regresyona çoklu regresyon denir. (Armutlu 2000)

İki değişken için doğrusal regresyon denklemini

$$Y_i = a + b X_i \quad (2.5)$$

şeklindedir.

Burada Y_i bağımsız değişken ve X_i bağımlı değişkendir. Yukarıdaki denklem doğru denklemdir. a ve b birer sabit katsayıdır. a sabiti doğrunun Y eksenini kestiği notayı, b sabiti ise doğrunun eğimini gösterir. Bu katsayılar hesaplandığı takdirde bağımlı değişkeni bağımsız değişken yardımıyla hesaplamak mümkündür.

Denklemdaki a ve b sabitleri regresyon denklemini en ideal denklem yapacak şekilde hesap edilmelidir. Yani regresyon çizgisi çizildiği zaman serpiye diyagramına en yakın noktalardan geçmelidir. Diyagramdaki noktaların doğruya olan uzaklıklarının kareleri minimum olmalıdır. Bu şart en küçük kareler metodu ile sağlanır.

2.2.1 Regresyon Denkleminin Hesaplanması

Regresyon denklemini hesaplamak için a ve b sabitlerinin hesaplanması gerekir. a ve b istatistiklerinin formülü

$$a = \bar{Y} - b \bar{X}$$

(2.6)

$$b = \frac{\sum X_i Y_i - n \bar{X} \bar{Y}}{\sum X_i^2 - n \bar{X}^2}$$

(2.7)

2.2.2 Regresyon Denklemini Hesaplayan Fonksiyon

Kiloboy tablosunda regresyon denklemini hesaplayan fonksiyon

Private Sub Command1_Click()

Dim a, b, c, e, yc, hatapayi, d, f, g, x, y, n, toplamxy, toplamxkare

```

e = Val(Text1.Text)
rst.Open "select avg(boy),avg(kilo) from kiloboy", cnn, 3, 3
x = rst(0)
y = rst(1)
rst.Close
rst.Open "select count(*) from kiloboy", cnn, 3, 3
n = rst(0)
rst.Close
rst.Open "select * from kiloboy", cnn, 3, 3
For i = 1 To n
c = rst(2)
d = rst(3)
toplamxy = toplamxy + (c * d)
toplamxkare = toplamxkare + c ^ 2
rst.MoveNext
Next
rst.Close
b = (toplamxy - (n * x * y)) / ((toplamxkare) - n * x ^ 2)
a = y - (b * x)
yc = a + (b * e)
rst.Open "select kilo from kiloboy where boy=" & e, cnn, 3, 3
f = rst(0)
rst.Close
hatapayi = f - yc
If hatapayi < 0 Then hatapayi = hatapayi * -1
MsgBox ("belirtilen kilodaki kişinin beklenen ağırlığı=" & yc)
MsgBox ("burada belirlenen hata oranı ise=" & hatapayi)
End Sub

```

2.3 Trend Analizi(Zaman Serileri Analizi)

Zaman serilerinin analizinde yapılacak işlemlerin en önemlisi trendi tespit etmektir. Zaman serisi, aynı olayın belli bir zaman süresi içinde arka arkaya gözlemlerinden meydana gelen seriye denir. Olayın bağlı bulunduğu sebepler olaya bir yön verir. İşte buna trend adı verilir. Bağlı bulunduğu sebeplere göre trenddeki artış ya da azalış bazen hızlanır bazen yavaşlar. Yani trend doğrusal da olabilir, eğrisel de.

2.3.1 Doğrusal Trend

Trendin hesaplanması en küçük kareler metodu yardımıyla yapılabilir.

Doğru denklemi

$$Y=a+bX \quad (2.8)$$

Burada X bağımsız değişken değeri olarak zamanı gösterir. Y ise bağımlı değişken değeri olarak zaman içindeki değişimleri gösterir.

Aslında trend denklemi tek bağımsız değişkenli regresyon denklemidir. Bağımsız değişken olan X zaman değerlerinden meydana gelmektedir. En küçük kareler metodu yardımıyla a ve b sabitleri bulunur böylece trend denklemi belirlenmiş olur.

Belirlenen trend denklemi bize gelecek bir zaman için tahmin yapma olanağı sağlar. Örneğin 1990,2004 yılları arasında gözlenen değerleri kullanarak oluşturulan trend denkleminde X yerine 2008 değerini koyduğumuz zaman belirlenen trende göre 2008 yılı için tahmin yapmak mümkün olmaktadır.

2.3.2 Trend Doğrusun Çizimi

Trend doğrusunun çizimi için denkleme ait iki nokta bulunur ve bu noktalar birleştirildiği takdirde çizim tamamlanmış olur.

Örnek

1981-2001 yılları arasında yılların nüfusa göre dağılımını gösteren gözlenen değerler aşağıdaki gibidir.

Çizelge 2.1 1981-2001 yılları arasında gözlenen nüfus değerler

yıl	nüfus
1981	45,5000
1982	46,7000
1983	47,9000
1984	49,1000
1985	50,3000
1986	51,4000
1987	52,6000
1988	53,7000
1989	54,9000
1990	56,1000
1991	57,3000
1992	58,6000
1993	60,0000
1994	61,4000
1995	63,0000
1996	64,6000
1997	62,6000
1998	64,7860
1999	64,3850
2000	65,3110
2001	66,2290

Bu değerlere göre hesaplanan trend denklemi ise şu şekildedir.

$$Y = -2098,66306189168 + 1,08268961038266 * X \quad (2.9)$$

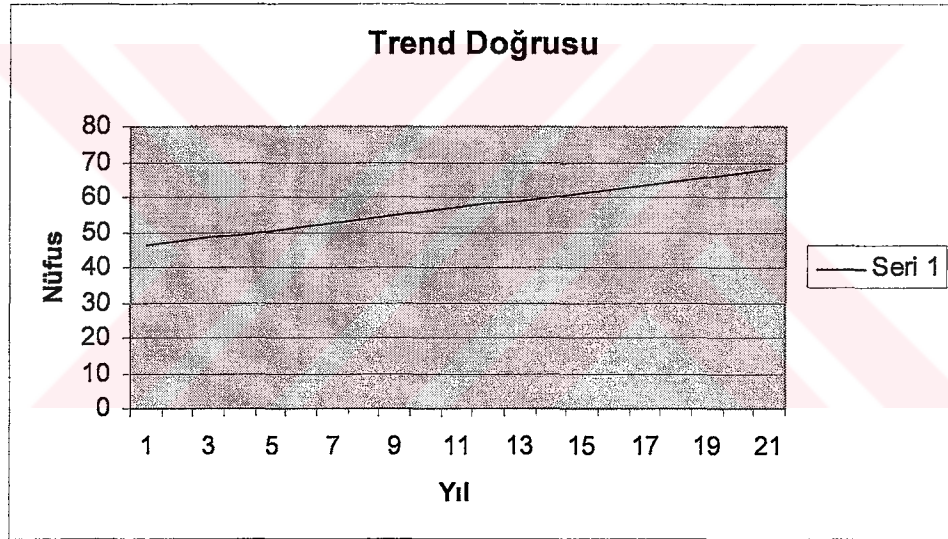
Bu denlemde x yerine 1981,2001 arası yılları koyacak olursak tahmini değerler aşağıdaki gibidir.

Çizelge 2.2 1981-2001 yılları arasında hesaplanan tahmini değerler

yıl	Tahmini Değerler
1981	46,14505628
1982	47,22774589
1983	48,3104355
1984	49,39312511
1985	50,47581472
1986	51,55850433

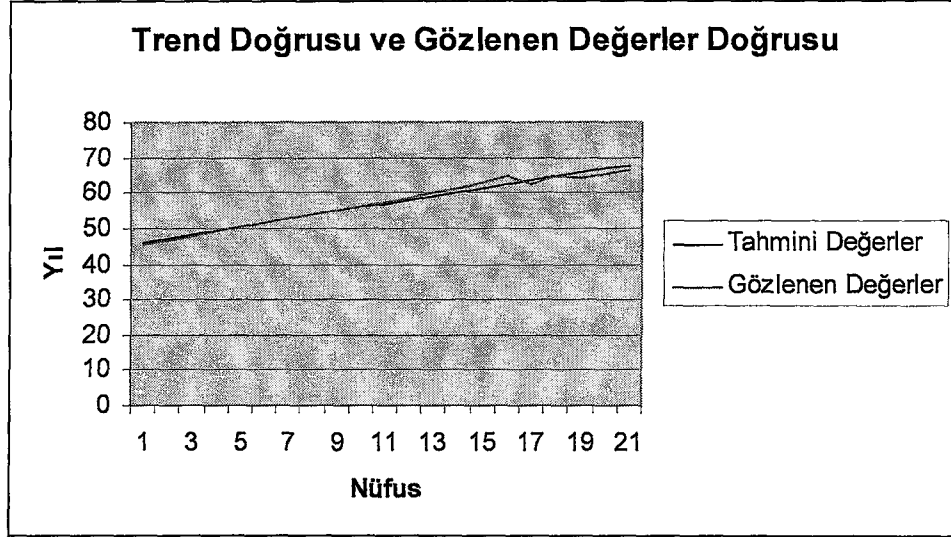
1987	52,64119394
1988	53,72388355
1989	54,80657316
1990	55,88926277
1991	56,97195238
1992	58,05464199
1993	59,1373316
1994	60,22002121
1995	61,30271082
1996	62,38540043
1997	63,46809004
1998	64,55077965
1999	65,63346926
2000	66,71615887
2001	67,79884848

Trend doğrusunu bu değerlere göre çizerek olursak trend doğrusu



Şekil 2.3 Nüfus için trend doğrusu

Gözlenen değerlerin oluşturduğu diyagrama da serpmeye diyagramı diyorduk. Gözlenen değerlerle trend doğrusunu grafikte karşılaştıracak olursak



Şekil 2.4 Trend doğrusu ve gözlenen değerler doğrusu

2.4 Çoklu Regresyon Analizi

Çoklu Regresyon Modeli:

$$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k + e_1 \quad (2.10)$$

Y :Bağımlı değişken

X :Bağımsız değişkenler

$b_0, b_1, b_2, \dots, b_k$:Katsayılar

e_1 :Hata terimi

Bağımsız değişkenler kurulacak modelde bir değişkenli olarak ele alınırsa, basit doğrusal regresyon, birden fazla bağımsız değişkenli olarak alınırsa, çoklu regresyon modeli konusunu oluşturmaktadır. Çoklu regresyon modelinde bağımlı değişkeni açıklayan birden fazla değişken model içinde bulunmaktadır. En ideal modeller genellikle birden fazla sebebin sonucu olarak gelişen olaylardır.

Çoklu regresyon modelleri en küçük kareler yöntemi kullanılarak çözümlenebilmektedir. Bilinmeyen sayısı ve denklem sayısı arttıkça parametrelerin kestirimi zorlaşmaktadır. Parametrelerin kestirimi için matrisleri kullanmak bizi genel bir formüle götürür.

Buna göre bağımlı ve bağımsız değişken için gözlenen değerleri X ve Y matrisleri bilinmeyenleri b matrisi olarak düşünelim. X ve Y ve b matrisleri aşağıdaki gibi olur

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix} \quad (2.11)$$

b matrisini hesaplanışı şu şekilde olacaktır.

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{Y}) \quad (2.12)$$

b matrisinin elemanları çoklu regresyon katsayılarıdır.

Hesaplanan bu katsayılar sayesinde regresyon denklemi elde edilmiş olur.

2.4.1 Regresyonun Anlamlılığının Sınanması

Regresyonun anlamlılığının sınanması bağımlı değişken yani açıklanan değişken ile bağımsız değişken yani açıklayıcı değişken arasında doğrusal bağıntı şeklinde bir ilişkinin olup olmadığının test edilmesidir (Hamburg 1974). Regresyonun anlamlılığını sınamak F testi ile mümkündür.

F testi yaparken 2 hipotez ortaya atılır.

Bunlar

H_0 : Değişkenlerin modelde bulunmaları anlamsız

H_1 : Değişkenlerin modelde bulunmaları anlamlı

Test sonucunda bir güvenirlik sınırı belirlenir, α ve çıkan F değeri F dağılım tablosunda belirlenen α değeri için tablo değeri ile karşılaştırılır. Karşılaştırma sonucuna göre H_0 hipotezi

ret veya kabul edilir. H_0 hipotezinin reddedilmesi bağımsız değişkenlerden en az birinin modelde bulunması gerektiğine işaret eder.

F oranını hesaplamak için MS_R ve MS_E değerleri bilinmelidir.

$$F = MS_E / MS_R \text{ 'dir} \quad (2.13)$$

$$MS_R = SSR / k \quad (2.14)$$

$$MS_E = SSE / (k-1) \quad (2.15)$$

Burada SSE

$$SS_E = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \mathbf{y}^T \mathbf{y} - \mathbf{b}^T \mathbf{X}^T \mathbf{y} \quad (2.16)$$

Burada SSR

$$SS_R = \mathbf{b}^T \mathbf{X}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n} \quad (2.17)$$

2.4.2 Çoklu Belirginlik Katsayısı

Bağımsız değişken sayısı bir ya da birden fazla olma durumunda bağımlı değişkenin bağımsız değişkenler tarafından açıklanmasının yüzdeyle ifade edilmesi modelin çoklu belirginlik katsayısının bulunması ile mümkündür. Basit doğrusal regresyonda olduğu gibi çoklu regresyonda da bağımsız değişkenlerin bağımlı değişkendeki değişkenliği açıklama oranı R^2 ile ölçülür. R^2 , +1 ve -1 aralığındadır. Belirginlik katsayısının pozitif karekökü çoklu korelasyon katsayısıdır. Çoklu korelasyon katsayısı bağımsız değişkenlerle bağımlı değişkenler arasındaki doğrusal ilişkinin derecesini verir.

Çoklu belirginlik katsayısı şu şekilde hesaplanır.

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} \quad (2.18)$$

$$SS_T = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (2.19)$$

$$\underbrace{\sum_{i=1}^n (y_i - \bar{y})^2}_{SS_T} = \underbrace{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}_{SS_R} + \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2}_{SS_E} \quad (2.20)$$

2.4.3 Tekbiçimleştirilmiş Regresyon Katsayıları

Tekbiçimleştirilmiş standart hale getirilmiş anlamına gelmektedir. Regresyon denkleminde değişkenlerin önündeki katsayılar bize o değişkenin bağımlı değişken üzerinde etkisi hakkında bilgi sahibi yapmayabilir. Tekbiçimleştirilmiş katsayılar boyutsuz olup önündeki değişkenin bağımlı değişkenler üzerinde göreceli etkisini gösterir. (Armutlu 2000)

Tekbiçimleşmiş modeli ikiden fazla değişken için genelleştirecek olursak

$$S_{kj} = \sum_{i=1}^n (x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_j) \quad (2.21)$$

olmak üzere x_k ve x_j arasındaki doğrusal korelasyon $r_{kj} = S_{kj} / (S_{kk}S_{jj})^{1/2}$ ve bu korelasyonların oluşturduğu korelasyon matrisi

$$R = \begin{bmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1k} \\ r_{12} & 1 & r_{23} & \cdots & r_{2k} \\ r_{13} & r_{23} & 1 & \cdots & r_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{1k} & r_{2k} & r_{3k} & \cdots & 1 \end{bmatrix} \quad (2.22)$$

olur. Benzer şekilde k adet bağımsız değişkenin her biriyle bağımlı değişken y arasındaki basit doğrusal korelasyonun oluşturduğu sütun matrisi

$$g = \begin{bmatrix} r_{1y} \\ r_{2y} \\ \vdots \\ r_{ky} \end{bmatrix} \quad (2.23)$$

olur. Bu iki matrisin yardımı ile k adet α_j parametresinin EKK yöntemi ile kestirimi

$$a = R^{-1}g \quad (2.24)$$

eşitliğinden bulunur. Bu kestirimin ardından eğer istenirse orijinal modeldeki katsayı kestirimlerine

$$b_j = a_j \left(\frac{S_{yy}}{S_{jj}} \right)^{1/2}, \quad j = 1, 2, \dots, k \quad (2.25)$$

formülü ile ulaşılır.

2.5 DMRA Programı

DMRA programı bir veri tablosu içinde seçilen tablonun sütuna göre (bu sütun bağımlı değişken olacaktır) regresyon katsayıları, çoklu belirginlik katsayısı, çoklu korelasyon katsayısı ve f oranını hesaplayan programdır. Program dinamik olup yapılan işlemler satır ve sütun sayısına göre otomatik olarak değişmektedir. Programın dinamik olmasının sebebi verileri tablodan son güncel haliyle otomatik alıyor olması, kullanıcıya hangi sütunlar arasında çalışacağını seçme şansını veriyor olmasından kaynaklanmaktadır. DMRA programı

sayı değerlerinden oluşmuş her tablo için satır ve sütun sayısı ne olursa olsun, hesaplama yapabilmekte ve sonuçları kullanıcıya anında verebilmektedir. Böylece kullanıcı elindeki her türlü veri için programı çalıştırabilmekte ve sonuç alabilmektedir.

DMRA programı kodu içinde istatistik formüller barındırmaktadır. Program kullanıcı tarafından onaylanan verileri içindeki koda gömülü istatistik formüller sayesinde hesaplayabilmektedir.

DMRA programı Visual Basic programlama ortamında geliştirilmiş SQL server ile uyumlu çalışan ADO veri iletişim protokolünü kullanan bir programdır.

DMRA programı İstatistik hesaplamaları yapan Visual Basic ara yüzü aşağıdaki gibidir.

The screenshot displays the DMRA program's user interface. At the top, there are controls for inserting and deleting columns, and buttons for 'Execute' and 'Back'. Below these is a table with 6 columns: 'sıra', 'kilo', 'boy', 'yas', 'cocuk', and 'mede'. The table contains 30 rows of data. To the right of the table, the text 'SELECT' is visible. Below the table, there are three sections for statistical calculations: 'Calculate Regression Coefficients', 'Calculate Coefficient Of Multiple Determination', and 'Calculate F Statistic'. Each section displays numerical results and execution times. The 'Back' button is located at the bottom center.

sıra	kilo	boy	yas	cocuk	mede
20	69	172	28	3	1
21	57	163	25	0	0
22	66	172	27	1	1
23	70	180	26	0	1
24	52	160	30	0	1
25	61	165	35	2	1
26	47	155	20	0	0
27	70	177	21	0	0
28	81	187	34	0	1
29	66	170	45	2	1
30	74	181	26	0	1

Calculate Regression Coefficients	Calculate Coefficient Of Multiple Determination	Calculate F Statistic
225.252980616858 7.87327952649086E-03 0.960073779527788 0.111631332578981 1.10873561696822	R2= 0.997358186541039 R= 0.998676220720288	F Stat= 1157.75310702382 MSR= 337.343036472566 MSE= 0.291377353622249
Time= 0.6308594 Sn	Time= 0.6416016 Sn	Time= 0.6503906 Sn

Şekil 2.5 DMRA programı

Görüldüğü üzere program üç ana bölümden oluşmaktadır. Bunlardan ilki kullanıcıya seçme olanağı tanıyan bölümdür. Bu bölüm sayesinde kullanıcı isterse çalışmak istemediği verileri eleyebilir. İkinci bölüm programın çalıştırıldığı başka bir deyişle veritabanından dinamik bilgi aktarımı sağlayan bölümdür. Üçüncü bölüm ise bilginin analizini sağlayan bölümdür. Bu bölüm yine kendi içinde üç ayrı kısımdan oluşmaktadır. Birinci kısmı regresyon katsayılarını hesaplayan kısımdır. Hesaplanan katsayıları ekranda görmek mümkündür. Bu kısmı çalıştıran kod içinde istatistik hesaplamalar yapan bir çok fonksiyon barındırmaktadır(bu hesaplamalar en küçük kareler metodunu kullanarak yapılmıştır). İkinci kısım modelin çoklu belirginlik katsayısı ve çoklu korelasyon katsayısını hesaplayan kısımdır. Hesaplanan bu katsayılar bize modeldeki açıklayıcı değişkenlerin açıklanan değişkeni açıklama yüzdesi hakkında bilgi verir. Üçüncü kısım ise modelin anlamlılığını sınamamızı sağlayan F oranının hesaplandığı bölümdür.

The screenshot displays the DMRA program interface. At the top, there is a menu bar with 'DMRA' and standard window controls. Below the menu bar, there are buttons for 'Insert Column', 'Delete', 'Or', and 'And'. A table of data is shown with columns: 'sıra', 'kil', 'boy', 'yas', 'cocuk', and 'medet'. The table contains 30 rows of data. To the right of the table, there is a vertical label 'SELECT'. Below the table, there is a button 'Execute'. Below the 'Execute' button, there is a section for 'Chose Dependent Variable' with a dropdown menu set to 'kil'. Below this, there are three columns of statistical results: 'Calculate Regression Coefficients', 'Calculate Coefficient Of Multiple Determination', and 'Calculate F Statistic'. The results include R-squared, F-ratio, and MSE values. Below these results, there are buttons for 'Write Predictions To Text File' and 'Show Predictions'. A 'Back' button is located at the bottom.

sıra	kil	boy	yas	cocuk	medet
20	69	172	28	3	1
21	57	163	25	0	0
22	55	172	27	1	1
23	70	180	26	0	1
24	62	160	30	0	1
25	61	165	35	2	1
26	47	155	20	0	0
27	70	177	21	0	0
28	61	187	34	0	1
29	68	170	45	2	1
30	74	181	26	0	1

Calculate Regression Coefficients	Calculate Coefficient Of Multiple Determination	Calculate F Statistic
225.252280516888 7.87327952649085E-03 0.980073779527788 0.111631332576981 1.10873581686622	R ² = 0.997358188541039 R= 0.998678220720288	F Oran= 1157.76310702332 MSR= 337.343036472566 MSE= 0.291377353622249
Time= 0.6308594 Sn	Time= 0.6416016 Sn	Time= 0.6503906 Sn

Write Predictions To Text File
Show Predictions

Back

Şekil 2.6 Çalıştırıldıktan sonra DMRA programı

Yapılan tüm bu hesaplamaların ardından program kullanıcıya elde edilen bilgileri ne yapacağını sorar. Bu kullanıcının bilgileri ne amaçla kullanacağına göre yapması gereken bir seçimdir. Kullanıcı bilgileri o anda sadece görüntüleyip yazdırabilir yada bir Excel yada text dosyasına transferini sağlayabilir. DMRA programı kullanıcıya her türlü veri için çok kolay bir şekilde çoklu regresyon analizi yapma olanağı sağlayan bir programdır. Bağımsız değişken sayısının bir olması durumunda program gene aynı şekilde çalışıp basit doğrusal regresyon modelinin kestiriminin yapılmasına yardımcı olur.



3 SQL SORGULAMA DİLİ

Veri erişim yöntemleri sayesinde tablo içindeki bilgilere ulaşılır. Bu çalışma içinde SQL Server içindeki verilere ulaşmak için ADO veri erişim yöntemi kullanılmıştır. ADO veri erişim yöntemiyle erişilen tablo verileri Visual Basic programlama ortamında analiz edilmiş ve sonuçlar alınmıştır. Bu amaçla gömülmüş SQL(Embedded SQL) kullanılmıştır. Gömülmüş SQL, SQL 'in programlama ortamında kullanıldığı oluşumdur. Verileri analiz etmek ve programlama ortamında kullanabilmek için SQL sorgulama dili ve SQL Server tablo yapısı iyi bilinmelidir. Bu bölümün amacı tablo yapısını, SQL sorgulama dilini açıkça ifade etmek ve sorguların anlamlarını anlatmaktır.

3.1 SQL Server Tablosu

SQL Server birçok verinin saklı tutulduğu, depolandığı, birden fazla kullanıcıya aynı anda bilgiye erişim imkanının sağlandığı yapıdır. SQL Server'da depolamanın temel birimi tablolardır. SQL Server tabloları, veritabanında yaratılır ve devamlılıkları burada sağlanır. Bir SQL Server tablosu satır ve sütunlardan oluşur. Bir tablo içindeki satırlar verilerin eksiksiz bir kaydını temsil eder. (Shapiro 2001)

Tabloların Genel Özellikleri

- Bir tablo, SQL Server veritabanında bulunan bir depolama birimidir.
- Tablo bir veritabanı nesnesidir ve başka bir ortama taşınamaz. Ancak verileri bir sonuç kümesi içinde bir kopya olarak taşınabilir.
- Veriler tablodan, bir tablolu datagram olarak taşınabilirler.
- İstemci bileşenleri(Ado gibi) tablolu verileri protokol yığınınından nasıl okuyacaklarını ve onları istemcide nasıl sunacaklarını bilirler.
- Değiştirilmiş olana veriler atılır. Değiştirilmesi gereken veriler sunucuya bir güncelleme olarak geri döndürülür.

3.1.1 Tablo Oluşturulması(CREATE TABLE)

CREATE TABLE komutu veritabanı içerisinde yeni bir tablo oluşmasını sağlar. Tablo oluşturma komutunun kullanımı için kullanıcının tablo oluşturma 'CREATE TABLE' hakkına sahip olması gerekmektedir. Başka bir kullanıcı şemasında tablo oluşturmak için ise diğer şemalarda tablo oluşturma 'CREATE ANY TABLE' hakkına sahip olması gerekmektedir. Tablolar bir veritabanı nesnesidir. Oluşturulan her tablo bir isme sahiptir. Tablolar sütun ve satırlardan meydana gelmiştir ve bir tabloda en fazla 254 sütun olabilir. Sütunlarda veriler saklanır. (Elmasri ve Navathe 1994)

Örnek:

```
CREATE TABLE Tabloornek
```

```
(Numarası numeric,
```

```
adi varchar,
```

```
soyadi varchar)
```

Yukarıdaki ifadede Tabloornek isimli bir tablo oluşturulmuştur. Tabloornek 3 sütundan meydana gelmektedir bu sütunlar, numarası, adı, soyadı sütunlarıdır. Numarası sütunun tipi numeric olup diğer sütunların tipleri varchar'dır.

Sütun Kısıtlama (Column Constraint) : Sütunlara bir kısıtlama koymak için kullanılır. Bu kısıtlama birincil anahtarları, sütunların tekiliğini, sütunlarda boş veri olup olmayacağı veya başka tablo üzerindeki alanlarla verinin bütünlüğünü korumak amacıyla kullanılır. (Codd 1970)

- i. **Birincil Anahtar (Primary Key) :** Tablodaki bir veya birkaç sütunun birleşmesiyle oluşturulan birincil anahtardır. Birincil anahtar bileşkesi bir tabloda ancak bir kere bulunabilir ve tekrarlanamaz. Birincil anahtar bileşkesindeki tüm alanlar "not null" yani içerisinde veri olması zorunlu sütunlar olmak zorundadır. Her tabloda ancak bir tane birincil anahtar bileşkesi olabilir. Fakat tekiliğin sağlanması için bazı durumlarda tablonun birden fazla sütunu birleştirilerek, bileşim birincil anahtar olarak tanımlanır. Bu tip bileşimli anahtarlar ayrıca 'Bileşik Anahtar' olarak da isimlendirilir.

- ii. **Yabancı Anahtar (Foreign Key) :** Tablodaki bir veya birkaç sütunun birleşmesiyle oluşturulan yabancı anahtardır. Yabancı anahtar ana tabloyla ona ait bir detay tablonun ilişkisini yapmak için veya bir tabloyla o tabloda bulunan bazı alanların tanımının yapıldığı tablolar arasındaki ilişkiyi kurmak için yapılır. Bir başka deyişle bir tablonun diğer tabloyu referans etmesidir. Yabancı Anahtarı içeren tablo, referans eden diğer tablo ise referans edilebilir. Böylece tablolar arası ilişkiler sağlanır.
- iii. **Tekil Anahtar (Unique Key) :** Tablodaki bir veya birkaç sütunun birleşmesiyle oluşturulan tekil anahtardır. Tekil anahtar bileşkesi tabloda en az bir kere bulunabilir, buda birincil anahtar üzerindedir. Fakat diğer sütunlarda da tekillik şartı verilebilir. Aynı alanda tekillik bir defa verilir, tekrarlamamayı sağlar.
- iv. **Boş Olmayan (Not Null) :** Tabloda belirtilen alanın boş olmayacağını verilmesidir. Bu alan içerisinde bilgi bulunması zorunludur. Yani bu alan hükümsüz olamaz.
- v. **Veri Sınırı Koymak (Check) :** Tablodaki bir alana verilecek değerlere bir sınır verme işlemidir. Bu değer sınırı, tablodaki diğer alanların ortak kullanımının bir sonucu hesaplanan bir değerde olabilir.

3.2 Sorgulama

Sorgulama, veritabanlarında kullanılan bir veri işleme şeklidir. Sorgulamada SQL sorgulama dili kullanılır Birçok veri erişim erişim yöntemi SQL deyimlerini destekler.

Yaygın Kullanılan SQL deyimleri

SELECT	Bir ya da birden fazla tablodan istenen verilerin alınmasını sağlar.
ALTER	Mevcut bir tabloyu değiştirir
CREATE TABLE	Yeni bir tablo yaratır
DELETE	Kayıtların silinmesini sağlar

INSERT	Mevcut bir tabloya kayıt eklenmesini sağlar
UPDATE	Bir tablodaki kayıtları günceller
EXECUTE	Bir saklı yordam ya da sorguyu çalıştırma

3.2.1 SELECT Deyimi

SELECT deyimi SQL dilinin en yaygın deyimidir. Bu deyim veritabanından veri almayı, veritabanındaki veriler üzerinde sorgu oluşturmayı sağlar. Bu deyim bir veri işleme deyimidir.

SELECT Deyiminin Yapısı:

SELECT (seçim listesi)
 INTO (yeni tablo)
 FROM (tablo)
 WHERE (ifade)
 GROUP BY (ifade)
 HAVING (ifade)
 ORDER BY(ifade)

En basit SELECT deyimi aşağıdaki gibidir.

SELECT 'ONCU'

Bu deyim herhangi bir tablo ya da veriye gerek duymaz. Bu deyim tek bir sonuç kümesi içerisinde tek değer olarak 'ONCU' geri döndürür.

Sonuç kümesinde kaç veri satırının geri döndürülmesi gerektiğini belirlememize yardımcı argümanlar mevcuttur.

ALL: ALL(SELECT ALL) argümanı bütün satırların sonuç listesinde görülmesini sağlar. Bu argüman varsayılan olup ayrıca belirtilmeyebilir. Sorgu kriterini sağlayan bütün satırlar, tekrar edilenler ve özelliiksizler(nullar) dahil olmak üzere geri döndürülür.

Örnek:

Çizelge 3.1 Tabloornek tablosu ve veri bilgileri

Numarası	Adi	Soyadı
1	Öncü	Uysal
2	Esin	Demir
3	Gül	Aydın
4	Öncü	Uysal

Select ALL(Adi)
from Tabloornek

Sorgu çalıştığında döndürülen sonuç çizelge 3.2’de gösterilmiştir.

Çizelge 3.2 Sütundaki tüm kayıtları getiren sorgu sonucu

Adi
Öncü
Esin
Gül
Öncü

DISTINCT: DISTINCT(SELECT DISTINCT) argümanı sadece tek olan kayıtların sonuç listesinde görülmesini sağlar.

Örnek: Çizelge 3.1 için aşağıdaki sorgu çalıştırıldığında sonuç seti çizelge 3.3’deki gibi olmaktadır.

Select DISTINCT(ADI)
From Tabloornek

Çizelge 3.3 Sütundaki tek kayıtları getiren sorgu sonucu

Adi
Öncü
Esin
Gül

Çizelgeden de görüldüğü üzere tablodaki birbirinin aynı olan kayıtlar geri döndürülmemiştir.

TOP: Top(SELECT TOP) argümanı sorguyu, yalnızca onu karşılayan ilk n satırı, tabloya daha önce eklendikleri sırada geri döndürmeye zorlar.

Örnek: Çizelge 3.1 için aşağıdaki sorgu çalıştırıldığında sonuç seti çizelge 3.4'deki gibi olmaktadır.

```
Select top 2 Adi
From Tabloornek
```

Çizelge 3.4 Sütunda en üstteki iki kaydı getiren sorgu sonucu

Adi
Öncü
Esin

3.2.1.1 Seçim Listesi

Seçim listesi veriden seçilecek sütunu belirtir. Virgüllerle ayrılmış olarak seçim listesinde belirtilen öğeler; verilerin sonuç kümesi için seçileceği sütunları temsil eder.

Örnek:Aşağıdaki ifade çizelge 3.1 için çalıştırıldığında, çizelge 3.5 oluşmaktadır.

```
Select adi,soyadi
From Tabloornek
```

Çizelge 3.5 Seçilen sütunları getiren sorgu sonucu

Adı	Soyadı
Öncü	Uysal
Esin	Demir
Gül	Aydın
Öncü	Uysal

Tüm Sütunlar(*): Tüm sütunların getirilmesidir. Yapılan sorguda tablodan veya görüntüden tüm sütunların gelmesi demektir. Sütun ismini vermeden veritabanı üzerinde sorgulanan tablo veya görüntüye ait tüm sütunları getirir.

Örnek: Aşağıdaki ifadede *, Tabloornek tablosundaki tüm sütunların getirilmesini sağlar. Çizelge 3.1'de bu sorgu çalıştırıldığında aynı sonuç bulunmaktadır.

```
Select *
From Tabloornek
```

İsimlendirme (c alias) : Sütunlara yeniden isim verme veya başlık vermektir. Yapılan sorguda gelen verilerin başlığının değiştirilerek istediğimiz bir isimle gösterilmesidir. Takma isim verilirken AS kelimesi isteğe bağlı olarak verilebilir. Verilen takma isim sorguda sadece gelen verilerin sütun sıralanmasında kullanılabilir.

Örnek: Aşağıdaki ifadede “Adı” alanının isimlendirme ile sonucunda başlığın ‘Ad’ olarak gösterilmesini sağlar. Çizelge 3.1’de bu sorgu çalıştırıldığında, çizelge 3.6 oluşmaktadır.

```
Select Adı Ad
From Tabloornek
```

veya

```
Select Adı As Ad
From Tabloornek
```

Çizelge 3.6 Sütun başlığı değiştirilmiş sorgu sonucu

Ad
Öncü
Esin
Gül
Öncü

3.2.1.2 SELECT...INTO

SELECT deyiminin INTO cümlecği, sonuç kümesinin verilerini tutması için yeni bir tablo yaratır. Yaratılan yeni tablo temel tablonun öz niteliklerinden herhangi birini içermez. Örneğin, sonuç kümesindeki sütunlar için takma ad kullanılır ise, takma adlar yeni tabloya taşınır.

3.2.1.3 FROM

FROM cümlecği SELECT sorgusunu karşılayacak olan sütunların hangi tablolardan alınacağını belirler. SELECT deyimi içerisinde tablolara, görünümlere, türetilmiş tablolara ve tablo değişkenlerine her başvurduğumuzda FROM anahtar sözcüğüne ihtiyaç duyulur.

```
SELECT * FROM tablo1
```

Yukarıdaki ifadede sorguya, tablodan bütün satır ve sütunları geri döndürmesi talimatını veren *(yıldız), operatöründen faydalanılmıştır. Ayrıca sorguda aşağıdaki birden fazla tablo adı belirtilebilir.

```
SELECT * FROM tablo1,tablo2,tablo3
```

3.2.1.4 WHERE

WHERE anahtar bir koşul belirterek sadece o koşula uyan kayıtların getirilmesini sağlar. Sorguda kayıtların verilen koşul veya koşullarla sınırlandırılarak getirilmesidir. Verilen koşul doğru ise verileri getirir. Eğer bir koşul verilmezse tablo veya görüntü üzerinde tüm kayıtlar getirilir.

Örnek:Aşağıdaki ifade Tabloornek tablosunda adı 'Öncü' olan kayıtların getirilmesini sağlar.Çizelge 3.1'de bu sorgu çalıştırıldığında çizelge 3.7 oluşmaktadır.

```
SELECT *
FROM Tabloornek
WHERE adi='Öncü'
```

Çizelge 3.7 Koşul kullanılmış sorgu sonucu

Numarası	Adi	Soyadı
1	Öncü	Uysal
4	Öncü	Uysal

3.2.1.5 GROUP BY

GROUP BY kayıtların gruplamasını sağlar. Sonuç kümesini alır ve onu gruplandırır. Gruplama yapılırken grup içerisinde istenilen hesaplamalar yapılabilir.

Örnek: Aşağıdaki ifadede aynı isme sahip olan kayıtları gruplayarak gösterir ve Count(*) komutuyla kaçar tane oldukları gösterilir. Çizelge 3.1'de bu sorgu çalıştırıldığında sonuç çizelge 3.8'de gösterilmiştir.

```
SELECT Adi, Count(*) Adet
FROM Tabloornek
GROUP BY Adi
```

Çizelge 3.8 Grup yapılmış sorgu sonucu

Adi	Adet
Öncü	2
Esin	1
Gül	1

3.2.1.6 HAVING

Sorguda gruplanmış kayıtların istenilen şekilde sınırlandırılmasıdır. İstenen koşul doğru ise verileri getirir. Buradaki amaç gruplamada ki hesaplamalar üzerinde sınırlama yapmaktır. Eğer bir koşul verilmezse tablo veya görüntü üzerinde tüm kayıtlar gruplanarak getirilir.

Örnek:Aşağıdaki sorguda HAVING deyimi GROUP BY ifadesiyle kullanılmaktadır.İfadede Tabloornek tablosunda ismi en az 2 defa geçen öğrenci isimlerini gruplanarak sayıları birlikte getirir. Çizelge 3.1’de bu sorgu çalıştırıldığında sonuç çizelge 3.9’da verilmiştir.

```
SELECT Adi, Count(*)
FROM Tabloornek
GROUP BY Adi
HAVING Count(*) >= 2;
```

Çizelge 3.9 Grup koşulu kullanılmış sorgu sonucu

Adi
Öncü

3.2.1.7 ORDER BY

Yapılan sorguda gelen kayıtların istenilen sütuna göre sıralanarak getirilmesidir. ASC ve DESC sıralama şekillerini belirler. Eğer sıralama şekli belirtilmezse ASC olarak standart şekli otomatik olarak alınır.

Sıralama şekli:

- i. ASC, küçükten büyüğe sıralar,
- ii. DESC, büyükten küçüğe sıralar.

Örnek: Aşağıdaki ifadede Tabloornek tablosu getirilirken, Adi sütununa göre alfabetik olarak sıralayarak getirir. Çizelge 3.1’de hasta adına göre büyükten küçüğe sıralama sorgusu çalıştırıldığında çizelge 3.10 sonucu elde edilir.

Select *

From Prst0100

Order By Hasta_Adi Desc;

Numarası	Adi	Soyadı
2	Esin	Demir
3	Gül	Aydın
1	Öncü	Uysal
4	Öncü	Uysal

4 ÖĞRENEN SİSTEMLER

Bu bölümde SQL tabanlı yapılarda bulunan öğrenme metotları anlatılacaktır. SQO yaklaşımının, daha performanslı çalışması için sorgu iyileştirmesi yapılır. Bu işlem verilen sorguda bulunan koşula göre gerçekleştirilir. Sorgunun koşulu alınır ve sorgu sonuç setiyle öğrenme metotlarından biri kullanılarak sonuç verileriyle kurallar türetilir. Bu öğrenilen kurallar araştırmada DML ifadelerine uygulanır.

4.1 Bilgiden Bilgi Öğrenme

Sorgu iyileştirme yöntemleri, veri modelleri ve dillerinin (Ullman 88) ortaya çıkışından beri üzerinde çalışılan bir konudur. Çünkü genellikle, sorguların verimli uygulanması zordur. Sorgunun yeniden formüle edilmesi yaklaşımı, aynı zamanda anlamsal sorgu iyileştirme yaklaşımı (SQO) olarak da bilinir. İyileştirme yaklaşımına geleneksel sözdizimsel yaklaşımdan daha farklı yaklaşılr, şöyle ki sorguları iyileştirmek için veritabanlarının içerikleri hakkında daha zengin bir bilgi kümesi oluşumunu gerektirir. Anlamsal bilginin kullanımı maliyet indirimi sağlar. Verilerden elde edilen sorgunun sonuç ve koşulu birlikte kullanılarak kurallar üretilir. Üretilen bu kurallar yeni sorgulara uygulanır. Kuralların içerdiği özelliklere göre çalışma performansı artırılır. Bu nedenle SQL, içinde bilgiden bilgi öğrenme sistemini içerir.

Yeniden formüle etmek için otomatik kural türetme fikri (Siegel 1988) tarafından önerilmiştir. Siegel'in yaklaşımında her ne kadar örnek sorgular kullanıyorsa da öğrenme aslında sabit bir Heuristic'ler kümesinden yürütülür. Heuristic'ler veritabanı yapısına ve uygulamasına göre tasarlanırlar. Bizim yaklaşımımız diğerlerinden şu yönden ayrılır; bizim yöntemimiz net Heuristic'lere dayanmaz. Örnek sorgular ve yeniden formüle etmede gerekli kuralları belirlemek için bunlarla elde ettiğimiz verilere odaklanmıştır. Bizim yaklaşımımızın amacı, veritabanı yapısı ve uygulamasına bağımlılığın en aza indirilmesi ve daha çok sorguların anlamsal yönlerini bilgilerle değiştirip yeniden formüle etme gerekliliklerine dayanır.

Sonuç olarak elimizdeki kurallardan yeni kurallar çıkarabilme ve bu kuralları kullanabilme performans için çok etkilidir. Kurallardan kural öğrenme işleminde oluşan kuralın veritabanı üzerinde doğruluğunun kesin olması gerekir, bu işlem için veritabanına erişerek veya

elimizdeki kuralların veritabanı üzerinde kaç tane satırda bulunduğu bilgisini de kullanabiliriz.

Örneğin : Elimizde aşağıdaki gibi iki kural olsun

R1) Hasta_No = '00000053' --> Hasta_Ad = 'Zeki'

R1) Hasta_Tel = '3125245214' --> Hasta_Ad = 'Zeki'

Burada hasta numarası '00000053' olan hastanın adı 'Zeki' ve veritabanında 'Zeki' ismi tek ise hasta numarası '00000053' olan hastanın telefonu '3125245214' olarak bulunur.

R3) Hasta_No = '00000053' --> Hasta_Tel = '3125245214'

Bu şekilde R1 ve R2 kurallarından R3 kuralı elde edilmiş olur.

4.2 Öğrenme Metotları

SQO yaklaşımında, daha iyi bir sorgu uygulanması için, sorgu iyileştirme yapılır ve bunun için kurallar kullanılır. Bununla birlikte kurallar sorgu çalıştırılmadan önce türetilmelidirler. Kural türetmede ki ana düşünce, o anki sorgu ve verilen veritabanına bağlı olarak kuralları otomatik olarak öğrenmektedir. Kural türetmede, SQO yaklaşımının performansını etkileyen bir faktördür. Eğer bir kural türetilmediyse, sorgu iyileştirmede kullanılacak bir kural olmayacaktır ve iyileştirme yapılamayacaktır. Bu nedenle kuralların öğrenilmesi SQO'da çok önemlidir ve bir çok araştırmacı tarafından sorgu iyileştirmesinde kural kullanımının avantajları incelendiği gibi öğrenmede araştırılmıştır. (Hsu ve Knoblock 1993; Sayli ve Lowden 1996, 1997a-b).

4.2.1 Siegel Yöntemi

(Siegel 1988) tarafından, SQL yaklaşımına otomatik kural türetme için temel bir yöntem olarak sunulmuştur. Bu yöntem iki bölümlü bir işlem olarak görülebilir. İlk bölüm kuralların karakteristiklerini belirler. İkinci bölümde ise bu karakteristiklere sahip yeni kuralları türetmek için veritabanından sorgulama yapılır. Bu bölümler SQO yaklaşımında yararlı kurallar bulunmasını sağlar. Bu yöntemle öncelikle bir 'önerilen kural' fikri oluşturulur ve

önerilen kuralın önceki durumu, sorgunun herhangi bir durumunu alarak oluşturulur. Sonuç sütunu, heuristics tarafından bulunur ve sonra yeni bir kural türetmek için veritabanına girilerek uygun bir koşul bulunabilir. Bu yöntem dört modülde açıklanmıştır: kural karakteristiklerini tanımlama, önerilen kuralların seçimi, sorgu üretilmesi ve kural yönetimidir. Bunlar bu bölümde açıklanmaktadır.

4.2.1.1 Kural Karakteristiklerini Tanımlama

Kural karakteristiklerini tanımlama yöntemin ilk modülüdür. Bu modülün amacı, işe yarar kurallar türetmek için sorgu iyileştiricisi tarafından kural türetme işleminin kontrol edilmesidir. Kuralları var olan kurallardan veya doğrudan veritabanından türetebilmek mümkündür. Yinede, eğer türetme işlemi sorgu iyileştiricisine bağlı değilse bu işlem SQO yaklaşımı için gereksiz kurallar bulmamıza neden olabilir. Bu modülde, SQO yaklaşımı beklenen maliyet tasarrufu ile birlikte bütün uyumsuz önerilen kuralları, kural türetme modülüne gönderir ve sonra bu modül tarafından incelenen önerilen kurallar, hesaplanan potansiyel maliyet tasarrufları ile birlikte önerilen kural listesine girer. Eğer bu kural listede mevcutsa birikmiş potansiyel maliyet tasarrufu bu önerilen kural için hesaplanır. Bundan sonra, belli bir eşiğin altındaki birikmiş potansiyel maliyet tasarruflu önerilen kurallar, önerilen kural listesinden silinmelidir. Aksi takdirde, bu önerilen kurallar yeni kurallar gibi algılanır.

4.2.1.2 Önerilen Kuralları Seçme

Yöntemin ikinci modülü önerilen kuralları seçmektir. Bu modül türetilen en iyi kuralları belirlemede kullanılır. Önerilen kuralların belirlenmesi uzmanlar için çok zor bir iştir. Bu sebeple, gittikçe artan beklenen maliyet tasarruflarını hesaplama, önceki tecrübelerle dayandırılır. Potansiyel değeri yeterince yüksek olan uyumlu önerilen kurallardan biri listede yer aldığında, önerilen kurallar işlem için seçilebilir. Bu işlem boyunca, daha düşük potansiyel değeri var olan kurallar, kurallar kümesinden silinecektir. Bu işlem kurallar kümesinin büyüklüğü ve kalitesinin belirlenmesinde kullanılabilir. Aynı amaçla, kural kümesinin en fazla değerine sınır koymak ve bir kuralın en az değerine sınır koymakta yararlı

olabilir. Eğer kurallar bu en az değerin altında bir değere sahipse kurallar kümesinden silinecektir. (Gökçe 2003)

4.2.1.3 Sorgu Üretme

Sorgu üretme yöntemin üçüncü modülüdür. Bu modül, ikinci modül tarafından seçilen bir önerilen kural alır. Temel olarak, modül önerilen kuralı kullanarak bir sorgu şablonu üretir. Bundan sonra veritabanı üzerinde sorgu çalıştırılır. Bu işlem boyunca ortaya çıkabilecek birkaç durum söz konusudur:

- i. Eğer bir önerilen kural bir koşulun çıkarılması ile bulunduysa sonucu ile ilgili bir koşul olacaktır. Bu koşul üretilen sorguda kullanılmaz.
- ii. Eğer önerilen kuralda, iki nesne ilişkisi arasında bir bağlantı her hangi bir sütun ile geliyorsa, bu ilişki koşulu kural oluşturulmasında kullanılmaz.

4.2.1.4 Kural Yönetimi

Kural yönetimi yöntemin son modülüdür. Bu modül, kurallar kümesine yeni bir kuralın eklenip eklenmeyeceğine karar verir. Modülde, türetilmiş kural, üretilen sorgunun cevabından ve önerilen kuraldan oluşturulur. Eğer cevap hükümsüzse, yeni kural olamaz ve önerilen kural çıkarılır. Eğer önerilen kural bir koşul çıkarılması ile oluşuyorsa, önerilen kuralın sonucu sorgunun cevabına göre kontrol edilmelidir. Bu durumda türetilmiş kuralın sonucu sorgunun olabilecek tüm cevaplarını kullanarak bulunabilir. Bu yöntem, olabilecek tüm kuralların türetilmesinde kullanılabilir. Yinede sadece gerekli kuralları bulabilmek, otomatik kural türetmedeki en önemli faktördür.

4.2.2 Knoblock Yöntemi

Öğrenme sistemlerindeki asıl amaç, veritabanındaki veri değerlerinin modellerine ve bunların önemine göre, kurallar türetmek, veritabanında verilen bir sorguda veritabanına mümkünse

giriş yapmadan, mümkün değilse en kısa sürede sorgunun sonucunu bulmaktır. Knoblock yöntemi bu amaçla en çok kullanılan öğrenme yöntemidir.

(Piatetsky ve Saapiro 1991) tarafından yayınlanan, veritabanlarında bilgi keşfine dayalı araştırma, bir yarı uygun algoritma olan KID3 kullanılarak kuralları öğrenmeyi anlatır. Bilgi keşfi, kural-yarar ölçümleri ve tüm veri kümelerindeki örnek türetme kurallarının doğruluğu gibi diğer konuları da içerir. Bu konular, istatistiksel yöntemler kullanılarak tanımlanmışlardır. Kesin kurallar ve sağlam kurallar olarak iki gruba ayrılırlar. Kesin kurallar veri tabanında her zaman doğrudurlar ve SQO yaklaşımı için kullanılırlar, sağlam kurallar hemen hemen her zaman doğrudurlar. Bu yüzden kesin kuralların keşfi için KID3 algoritması kullanılır. Bir kesin kuralı öğrenmek için, önceki duruma uyan kayıtların aynı zamanda sonuç durumuna uyup uymadığı kontrol edilmelidir. KID3 algoritması $A @ a \rightarrow B @ b$ şeklindeki kural için kullanılabilir. Burada 'a' ve 'b' sabitler ve @ karşılaştırma operatörlerinden { <, <=, >, >=, =, != } biridir. Algoritmanın ana fikri, her bir kaydı A ile değiştirmektir. Her bir değişen hücre, tüm kayıtların bir özetini saklamak için kullanılır. Eğer bir kayıt kullanılan bir hücreye dönüştürülürse, hücre özeti, özet ve kayıt arasındaki karşılaştırmaya göre güncellenir. Algoritma aşağıdaki gibidir.

Procedure KID3 (A, file)

```

For Tuple in File
  Get Cell corresponding To Tuple.A
  If empty Cell Then
    Cell.Summary = Tuple, Cell.Count = 1;;Hücre başlangıç durumuna
gelir
  Else
    Cell.Count = Cell.Count + 1;; Kayıt değiştirme
    For field C in Cell.Summary
      When Cell.Summary.C != NIL ;; Hücre özeti hükümsüzleştirilir.
      Do<sütun tipi> Summary-Update(Cell.Summary.C, Tuple.C)
    End for
  End If
End For

```

KID3'deki 'Summary-Update(hücre, kayıt)' prosedürü kayıttaki C değeriyle karşılaştırarak o anki hücre özetini bir C alanı için değiştirir. Eğer alanın özeti çok genelse, özet değeri hükümsüzleştirilir ki bu da alanın öğrenmede kullanılmayacağı anlamına gelir. Dizi alanının tipi için, hücre özeti K değerlerine sahiptir. Burada K, alanın alabileceği değişik değerlerin en fazla sayısıdır. Bu maksimum sayıdan daha fazla K değeri varsa, özet hükümsüzleştirilir. Olağan değerler ve onların frekansları, kompleks özetler için K değerlerinin seçiminde kullanılabilir. 'NUMBER' alan tipi için, hücre, alanın en az ve en fazla değerlerine sahiptir. Kompleks özetler için ortalama değer ve standart sapma gibi değerler elde etmek mümkündür.

Bu aşamada, özetleri kullanarak kuralların nasıl yorumlandığını açıklamak gerekir. Bu hücre sayısı ve her bir hücre için $A = a$ değerli tüm kayıtlarının özetleri kullanılarak yapılır. Her bir geçerli özet alanı C için, alanın tipini bulmak için bir kontrol yapılır. Eğer tip 'CHAR' veya 'VARCHAR2' ise C alanı şöyle yorumlanabilir. $C = c1V.....Vck$, burada $c1,.....,ck$ ise C alanının değerleridir. Eğer tip 'NUMBER' ise, C alanı şöyle yorumlanabilir. $c1 \leq C \leq c2$, burada $c1$ C'nin en az $c2$ ise C'nin en fazla değeridir.

Bu algoritmayı değişik tipteki kurallar için genişletmek mümkündür. Örneğin önceki durum $a1 \leq A \leq a2$ şeklinde ise yine algoritma uygun kayıt sayısı veya veri dizinleme yapısı seçiciliği temeline dayanmaz. Yapılan tek kontrol, A üzerindeki durumla uyumlu tüm kayıtların, C üzerindeki duruma da uyup uymadığının bulunmasıdır. Bu yüzden tüm kuralların veritabanındaki etkilerini kontrol etmek için bir yoldur. Yani bazı kurallar, kurallar kümesinde tutulmaya değer olmayabilir. Bu gelecekte çok büyük bir kurallar kümesine sahip olmamıza ve verimsiz bir sorgulama işlemine yol açar. Her ne kadar bu araştırma, kuralları analiz etme için kural yararı ölçümlerini verse de, öğrenme işleminde kurallar türetme aşamasında daha seçici olması gereklidir.

Knoblock yöntemi ise, veritabanındaki asıl veri örneklerini kontrol ederek, verilen bir sorgunun durumlarını kullanır. Bunu, aday durumlar üzerinde dizinleme gibi özel veri yapılarını içeren en çok istenilen kuralları belirlemek için yapar. Yöntem, verilen bir sorgunun herhangi bir durumu için veritabanından gelen pozitif ve negatif örneklerle, bu durumun SQO yaklaşımında kaç kere oluştuğu temeline dayanır. Yöntem diğerlerine göre daha seçicidir ve buda en büyük avantajıdır.

4.2.2.1 Alternatif Sorgu için Tümevarımla Öğrenme Algoritması

P pozitif veri, N negatif veri, S ise verilen veritabanı şemasıdır.

Adım 1 : q' hükümsüzleştirilir.

Adım 2 : Her bir sütunun (A), değer aralığı R ve P arasında olduğu bulunur.

Adım 3 : Her bir sütunun kazancı: $gain(x)$ ve maliyeti: $cost(x)$ hesaplanır.

Adım 4 : Eğer kazanç < 0 ise kullanılmaz

Adım 5 : Kazanç / Maliyet oranı en yüksek olan sütunu seçilir.

Adım 6 : $q' = q \cup \{x\}$; $A = A - \{x\}$; $N = N - \{n | \text{Her } n, x' \text{ ler hariç } n' \text{ lerin toplamı } N' \text{ den}\}$

Adım 7 : Eğer $N = \text{Hükümsüz}$ ise geriye q' nü döndür.

değilse Adım 3.'e git.

Where $gain(x) = \{n | \text{Her } n, n' \text{ lerin toplamı } N \text{ ve } x' \text{ ler elenir.}\}$

$$cost(x) = E\text{-cost} * C\text{-cost}$$

$$= \text{Değişen } cost(x) * (P + N - gain(x)), \text{ eğer sütun dizin üzerinde ise}$$

$$= \text{Değişen } cost(x) * (P + N), \text{ eğer sütun dizin üzerinde değilse}$$

Bir koşul için kazanç, $gain(x)$ fonksiyonu, durum tarafından elenen örneklerin sayısıdır. Durumun maliyeti, değer maliyeti, E-cost, ile değerlendirilecek örnek sayısının, C-cost, çarpımıyla bulunabilir. E-cost, değer maliyetidir ve bir örneğin duruma uygun olup olmadığını belirler. Bu maliyet, verilen durum için karşılaştırma koşullarının sayısı ile, her bir karşılaştırmaya maliyetinin çarpımıyla bulunur. Karşılaştırma maliyeti, koşul veri tipine bağlıdır. Dizi tipi için, maliyet veri şemasındaki durum sütunun uzunluğunun tanımıdır. Bu sayı tipi için 2'dir. C-cost, koşulun dizinlenmiş bir sütun temelli olup olmadığıyla ilgilidir. Eğer koşul dizinlenmiş bir sütun temelliyse, C-cost koşulu karşılayan örneklerin sayısıdır. Yoksa, veritabanındaki örneklerin toplam sayısıdır. Bunun sebebi, veritabanındaki tüm örnekler koşula değer biçmek için kontrol edilmesindendir.

İkinci bölümde, veri tabanı başlangıç geneli iki aşamada işleme konur. Bu var olan kuralları formüle etmek için yapılır ki böylece kurallar verilen sorgunun yeniden formüle edilmesinde kullanılabilirler, $q \rightarrow q'$: dönüşüm ve yarar. Dönüşüm aşamasında, veritabanı başlangıç geneli bir kriteri karşılamak için dönüştürülür, bu kriter kuralın sonuç tarafı ile orantılı bir aralığa

sahip olmalıdır. Yarar aşamasında dönüştürülen kurallar başka bir kriteri karşılamak için basitleştirilir, bu kriter 'Kuralın önceki durum tarafının olabildiğince kısa olmasıdır.' Daha da fazlası önceki durum tarafını basitleştirmek için, önemsiz koşullar elenir ve bunun için bir algoritma kullanılır. Algoritmayla sonuç tarafındaki negatif örnekler aranır ve sonra eğer koşul sonuç tarafında en çok örneğe sahipse koşulu önceki durum tarafından seçer.

Tümevarımlı öğrenme algoritmasından görüleceği gibi, yöntem sıkça uygulanan ve düşük maliyetli yararlı kuralları öğrenmek için kullanılabilir. Öğrenme işlemi her ne kadar iyi tanımlanmış olsa da, sistem SQO yaklaşımının kural bakımı ve maliyet değeriyle sorgu iyileştirme gibi diğer bileşenlerine daha az hitap eder.

SQO yaklaşımı için Knoblock'un yöntemine benzer olarak (Lowden 1995; Lowden ve Lim 1995) tarafından anlamsal kurallar türetmek için ARDOR ortaya atılmıştır. ARDOR öğrenme zamanını azaltan örnek sorgular temeline dayanır. Bu yöntem sistem kullanıcılarının herhangi bir denetim ve müdahalesine gerek duymaz. Kural türetme işlemine bir ilavesi vardır. Bu ilave, yeni bir kural türetildiğinde, kurallar kümesine eklenmeden önce yeni kuralın aynı önceki durumuna ve sonucuna sahip herhangi başka bir kuralın olup olmadığının kontrol edilmesidir. Eğer böyle bir kural varsa yeni kural ve var olan kural önceki koşul ve sonuç koşulunun sınırlarını değiştirmek için kıyaslanırlar. Bu sonuç kuralının düzgün sınırları olmasını sağlar. Yapılan inceleme aynı sonuçla biterse, fakat önceki koşul farklıysa ve var olan kuralın önceki koşulu yeni kuralın önceki koşulu kullanılarak genişletilebiliyorsa, yeni kural var olan kuralın yerini alır. Diğer bir durumda, eğer inceleme aynı önceki koşullarla fakat farklı sonuçlarla biterse ve var olan kuralın sonucunu yeni kuralın sonucuyla sınırlamak mümkünse, yeni kural var olan kuralın yerini alır. Örneğin iki var olan kural $\{R1, R2\}$ ve iki türetilmiş kural $\{R1a, R1b\}$ olsun.

R1) project $\geq 12 \rightarrow$ dcode = 'SALE', R2) dcode = 'SALE' $\rightarrow$ project ≥ 12
 R1a) project $> 15 \rightarrow$ dcode = 'SALE', R1b) project $\geq 10 \rightarrow$ dcode = 'SALE'

R1a ve R1b, yeni kurallarıyla karşılaştırarak, R1'in sınırını değiştirecek bir kontrolün yapılmasına gerek duyulduğu görülebilir çünkü bu kurallar aynı kurallar sınıfına dahildirler. R1 kuralının önceki koşulunu R1a'ya değiştirirsek, kuralın sınırı daha küçülür. Bu yüzden, R1a kuralı pas geçer. Ama R1'in önceki durumun sınırını R1b'ye değiştirmek sınırı genişletir. Bu yüzden, kural R1b, R1'in yerini alır.

İki türetilmiş yeni kural {R2a, R2b} düşünelim:

$$R2a) dcode = 'SALE' \rightarrow project > 15, \quad R2b) dcode = 'SALE' \rightarrow project \geq 10$$

Bu kuralın R2 kuralıyla aynı olduğu görülebilir. Sonuçlar açısından, R2b kuralı pas geçilir ve R2a kuralı R2'nin yerini almak için seçilir böylece sonucun sınırı en küçük değerine düşürülür. (project $\Rightarrow$ 12'den project $\geq$ 15'ye kadar)

4.2.3 Öğrenme Metotlarının Avantajları ve Dezavantajları

Siegel yönteminin temel yaklaşımı (Siegel 1988) tarafından kural türetme, sorgu dönüşümü ve düşük maliyetli sorgu işlemini bulmaktır. Bu yöntemden, önerilen kuralları kullanarak bir çok dönüşüm türetmek mümkündür ve bunların bir kısmı sorgu iyileştirme işlemi için kullanışlı olmayabilir. Bunun için kurallar kümesinin büyüklüğünü sınırlamak çok önemlidir. Başka bir nokta ise yöntem gerçek verilere değil gerçek verilerden oluşturulan verilere bağlıdır. Bu yüzden, yöntem veri şablonlarını kullanarak geliştirilmelidir. Böylece veritabanındaki gerçek veri değerlerinin üzerinde, kuralların etkileri belirlenebilir. Bu yüzden, bu tarz sorunlarla başa çıkabilmesi için temel yöntemi geliştirmek önemlidir. Bu geliştirmeler Siegel'in aynı araştırmasında üç farklı alanda olabilir. Bunlar, Kullanıcıya özel alan bilgisi, Heuristic bilgi ve istatistiksel artıştır.

4.2.3.1 Kullanıcıya Özel Alan Bilgisi

Otomatik kural türetme sabit bir işlem değildir, başka bir deyişle, çeşitli anlamsal bilgi ve istatistiklere bağlıdır. Temel yöntemde bilginin kullanımı tek bir alan tarafından sınırlandırılmıştır. Yine de, kullanıcıya özel alan bilgisi tarafından analiz edilen ve kullanılan bilgi için KDD ile ortaya atılan birkaç yöntem vardır. Bu alan bilgisi kural sınıflarının bir belirtisi olarak görülebilir ve bu belirtiyeye göre hangi kural sınıfının anlamlı ilişkiler içerdiğini belirlemek mümkündür.

4.2.3.2 Heuristic Bilgi

Temel yöntemi desteklemek amacıyla Heuristic bilgi SQO yaklaşımı için çok önemlidir. Yine de bu bilginin kullanımı da kurallar için sınırlandırılmıştır. Bu kurallar sadece değere bağlıdır. Başka bir değişle bağımlılık yoktur. Eğer ilişkilerin bağımlılıklarını kapsayan yeni bir Heuristic tanımlamak mümkün olursa, temel yöntem gerçekte kullanılan sorgular için daha verimli ve daha uygulanabilir olacaktır. Heuristic, keşfedilmiş Heuristic'lerden (Dizin tanıma, tarama azaltması gibi) türetilmeyen farklı kural tiplerini bulmak için kullanılabilirler. Eklenen Heuristic'ler, referans kural sütunları, dinamik kural sütunları, statik kurallar (Özellikle ilişkinin anahtar sütunu için) ve kategori sütunları gibi yararlı kuralların türetilmesi için ilişkilerde ne tip sütunların olması gerektiğini keşfedecek şekilde geliştirilmelidir.

4.2.3.3 İstatistiksel Artış

İstatistiksel artış, geçmiş sorguların işleme konulması ile toplanabilen istatistikleri (geçmiş bilgileri) kullanan kural türetme işleminin kalitesini artırmak için kullanılır. Siegel yöntemi, istatistiklerin elde edilmesi durumunda kural türetme yöntemindeki önerilen kurallar için potansiyel maliyet kazancının değerlendirilmesinde 3 faktörün belirlenmesi için bu istatistiklerin kullanılabilceğini işaret eder. Bu üç faktör türetilbilme, bakım ve önceki seçicilik faktörleridir. İstatistikleri kullanmadaki sebep eski türetmelerin performansı ile her hangi bir bilgiye sahip olmadan sadece şimdiki sorguyu kullanarak bu faktörlerin yöntemle keşfedilememesidir. Türetilbilme faktörü, kural sınıfındaki sütunların ilişkisine göre yapılır. Bakım faktörü kural kümesindeki bir kuralın bakımının ve bozuklukların kontrol edilmesinin maliyetiyle ilgilenir. Bu faktör sonradan kural türetişinin yararlılığı hakkında bir karar vermek için kullanılabilir. Önceki seçicilik faktörü, önerilen kural için önceki koşulunun seçiciliği ile alakalıdır. Eğer önceki koşul çok genişse, önerilen kuraldan gerçekten yararlı yeni bir kural türetmek pek mümkün değildir. Görüldüğü gibi, bu faktörlerin uzmanlar tarafından belirlenebilmesi çok kolay değildir. Çünkü bu faktörlerde işlenmiş sorguların geçmiş performanslarına ve bunların istatistiklerine bağlıdır.

KDD'de veri bağımlılıkları temelli yöntemleri kullanarak yeni kuralları öğrenmek mümkündür. Bu yöntemlerdeki ana fikir, istatistikler ve alan bilgisi aracılığıyla ilişkiler

arasındaki veri bağımlılığını ölçmektir. Böylece sağlam kural sınıfları bulabilmek için gereksiz sütunların elenmesi sağlanır. Genellikle yöntemler tek bir kuralın yerine, bir kural sınıfında olmasının test edilmesi temeline dayanır. Çünkü bazı durumlarda kurallar aynı kural sınıfında olmalarına rağmen bir kural diğerlerinden daha verimli olabilir. Bu sebeple kural kümesinin verimli verimsiz kurallarla çok fazla büyümesi gerçekleşebilir. Buda SQO'nun performansını azaltır.

(Siegel 1988) Siegel tarafından sunulan, yöntemlere karşılık, (Hsu ve Knoblock 1993) tarafından sunulan, Knoblock temelli yöntem daha verimli kullanılabilir. Çünkü her hangi bir Heuristic ile sınırlandırılmamıştır. Yöntem birkaç yönden çok umut vericidir. Bunlardan biri, verilen sorgu yeniden formüle edilir ve sonra yeni kurallar türetmek için kullanılırlar. Bir başka umut verici yönü ise, veri örnekleri verilen sorunun koşulları için bu yöntemle kontrol edilir ve her bir koşul için koşulla karşılanan pozitif örnekleri bulup bu örnekleri aday koşulların yapımında kullanılır. Sonra negatif örnekleri elemek için en güçlü aday koşulları seçer. Bu eleme, veri tipleri, uzunlukları, dizin yapıları ve bunların oluşum sayılarına bağımlıdır.

(Lowden 1995; Lowden ve Lim 1995) tarafından sunulan SQO sistemi, birkaç ekle benzer bir öğrenme işlemi temeline dayanır.

5 ELİMİNASYON METODUYLA KURAL ÖĞRENME

Kural öğrenmek bir veritabanı üzerinde çalıştırılan sorguda zamandan tasarruf sağlar. Kural öğrenen bir sistemi daha verimli bir hale getirip, zaman kazancını maksimum yapmak, kural öğrenme işini ve sorgunun çalışma zamanını minimize etmek eliminasyon metodu ile mümkündür. Yapılan iş, sistem kural öğrenmeye başlamadan önce sütun sayısını azaltmaktır. Ancak bu anlamlı bir şekilde yapılmalıdır. İstatistik metotlar yardımıyla kural öğrenmek istediğimiz sütun için anlamı olan başka sütunları tespit etmek daha anlamsız olan sütunları ise elemek mümkün olmaktadır. Sütunların azalması yada bir başka deyişle eliminasyon yapılması kural sayısının azalmasına neden olacaktır. Daha az kural üreterek çalışan öğrenen sistem daha hızlı çalışacaktır. Kural seti anlamsız verilerin neden olduğu şişmelerden arınacaktır. Kural setini kullanarak çalışan sorgu daha verimli hale gelecek sorgu çalışma zamanı azalacaktır.

5.1 Tekbiçimleştirilmiş Regresyon Katsayıları Yardımıyla Veritabanı Eliminasyonu

Bir SQL Server tablosunun sütun ve satırlardan oluştuğunu biliyoruz. Sütunları temsil eden değerlerin a,b,c.....z olduğunu ve kural üretmek istediğimiz sütunun a olduğunu varsayar ise a değerinin istatistik karşılığı modelde bağımlı değişken olacaktır. Her a değeri için her b,c.....z değerleriyle çalıştırılan modelde tekbiçimleştirilmiş regresyon katsayılarını hesaplamak mümkün olacaktır. Tekbiçimleştirilmiş katsayılar boyutsuz olup önündeki değişkenin bağımlı değişken üzerindeki görelî etkisini gösterir.(Armutlu 2000)

Bağımlı değişken a ve bağımsız değişkenler b,c,...,z olduğunda çoklu regresyon denklemi

$$a = \beta_0 + \beta_1 b + \beta_2 c + \dots + \beta_{29} z \quad (5.1)$$

Değişken dönüşümünden sonra

$$a = \alpha_1 b + \alpha_2 c + \dots + \alpha_{29} z \quad (5.2)$$

denklemine dönüşecektir.

Denklemdaki α değerleri bize bağımsız değişkenlerin bağımlı değişken üzerindeki görelî etkisini gösterir. Bu değerlerden yola çıkarak a değeri için hangi değerlerin daha anlamlı olduğunu katsayıları yorumlayarak görmemiz mümkün olacaktır. Değerlerin -1,+1 arasında çıkacağını ve -1 veya +1 değerine yaklaşan değerlerin önündeki katsayıların bağımlı değişken

üzerinde daha etkili olduğunu, 0'a yaklaşan değerlerin ise daha anlamsız olduğunu biliyoruz. Buradan bir modellemeye varmak için tüm α değerlerinin aritmetik ortalamasını alıp bulunan ortalamanın üzerindeki α değerlerinin önündeki değişkenler için model için daha anlamlı demek mümkün olacaktır. Daha anlamlıları modelde bırakıp daha anlamsızları modelden çıkarır isek modeldeki değişken sayısı azalacaktır. Model için artık elimine edilmiş model dememiz mümkün olacaktır.

Örnek

Tablo test bir veritabanı tablosu olup y, x1 ve x2 sütun arından oluşmaktadır
Eliminasyon Öncesi Model

Çizelge 5.1 Tablo Test

Y	x1	x2
22	8	6
23	10	7
18	7	5
9	2	2
14	4	3
20	6	4
21	7	4
18	6	3
16	4	3
19	6	3

Eliminasyon sonrası model

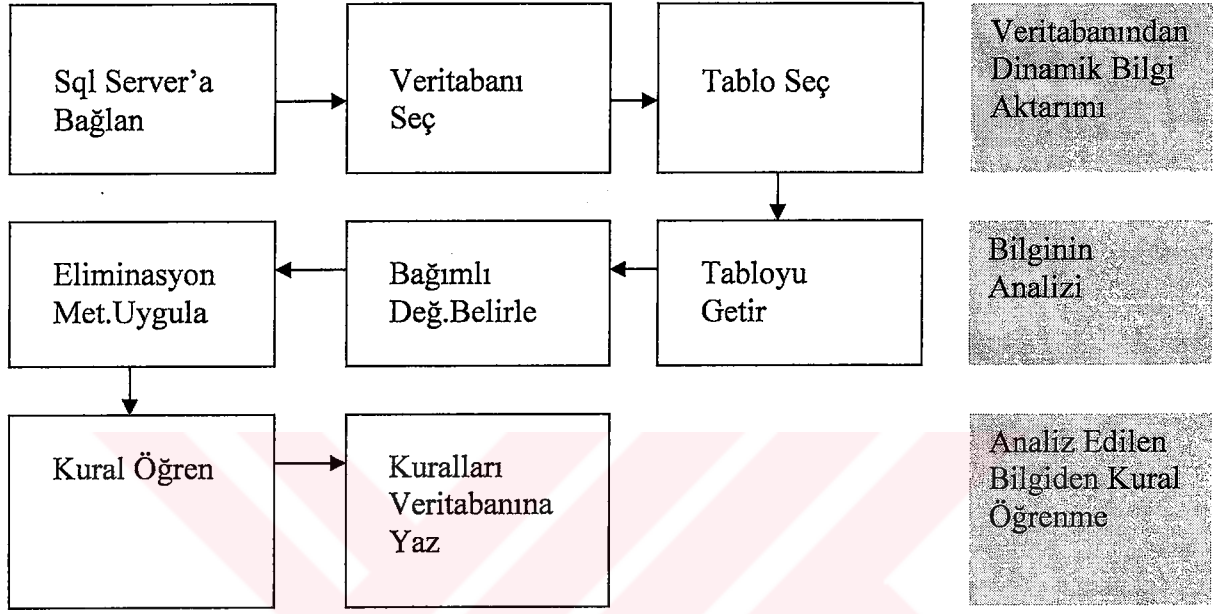
Çizelge 5.2 Elimine edilmiş Tablo Test

Y	x1
22	8
23	10
18	7
9	2
14	4
20	6
21	7
18	6
16	4
19	6

Eliminasyon öncesi ve sonrası modellerden görüldüğü üzere sütunlardan biri elenmiştir. Kural öğrenmek veritabanı üzerinde işlem yaparken zamandan tasarruf sağlar. Üretilen kurallar

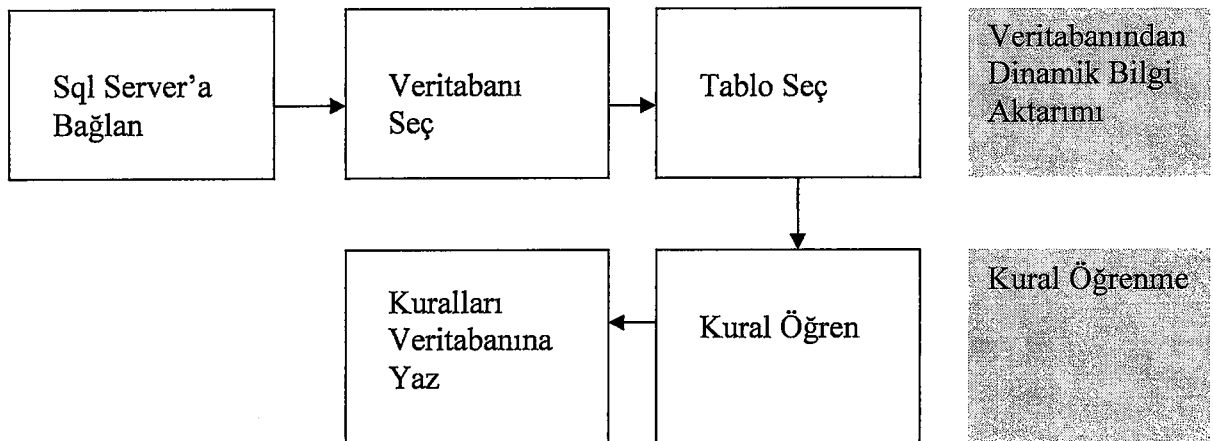
sorgu esnasında zaman maliyetinin azalmasını sağlamaktır. Eliminasyon metodu uygulanarak gerçekleştirilen öğrenme de kazanılan zamanın maksimuma ulaşmasını amaçlamaktadır.

5.2 Eliminasyon Metodu İle Kural Öğrenen Dinamik Sistem



Şekil 5.1 Eliminasyon sistemiyle kural öğrenen sistemin çalışma metodolojisi

5.3 Eliminasyon Metodu Kullanmaksızın Kural Öğrenen Dinamik Sistem



Şekil 5.2 Eliminasyon metodu kullanmaksızın kural öğrenen sistemin çalışma metodolojisi

5.4 Eliminasyon Metodu İle Kural Öğrenen Dinamik Sistem

Uygulama

Tüm uygulamalar için kullanılan:

Program Geliştirme Ortamı: Visual Basic 6.0

Veritabanı: SQL Server 8.0

Veri Erişim Yöntemi:ADO

Bu uygulama için kullanılan:

Veri Tablosu:episode2

Veri Sayısı:1000

Sütun Sayısı:11

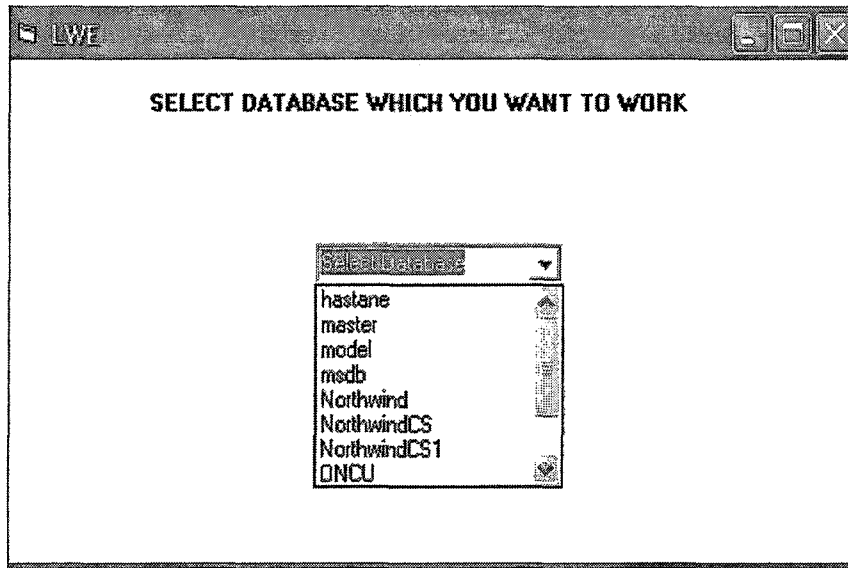
Tablo Anahtar Değeri:id

Tablo dizin(index) değeri:episodes

Veritabanından Dinamik Bilgi Aktarımı

Program her çalıştırıldığında otomatik olarak SQL Server'a bağlanır.

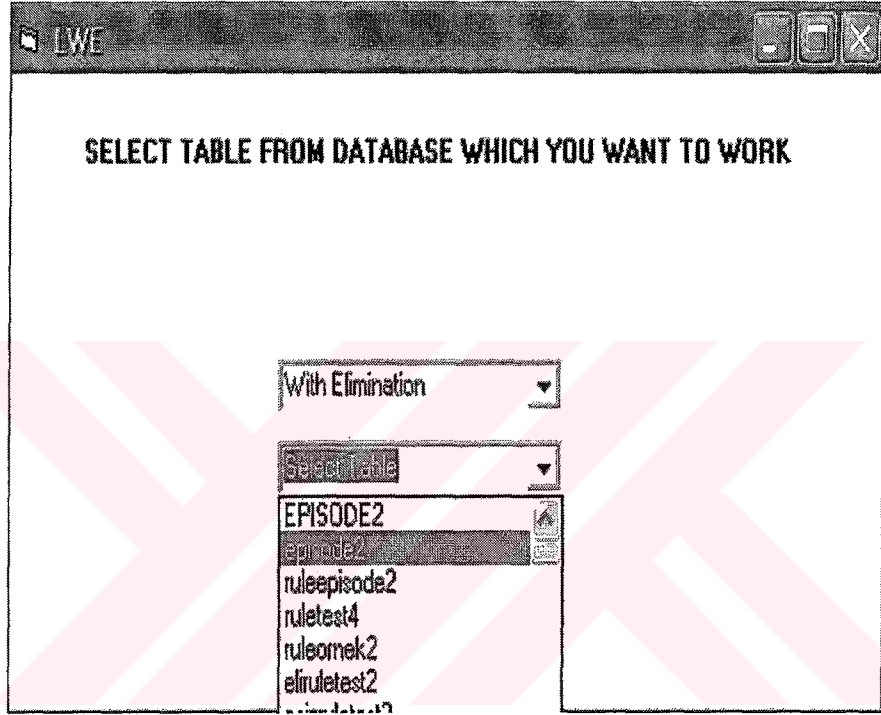
Veritabanı Seçimi



Şekil 5.3 LWE programı, veritabanı seçim ekranı

SQL Server'a bağlanan program kullanıcıdan üzerinde çalışmak istediği veritabanını seçmesini ister. Listelenen veritabanları SQL Server sistem dosyalarından otomatik olarak alınır bu sebeple dinamiktir yani kullanıcıya bir dakika önce bile oluşturulmuş veritabanı üzerinde işlem yapma kolaylığı sağlar

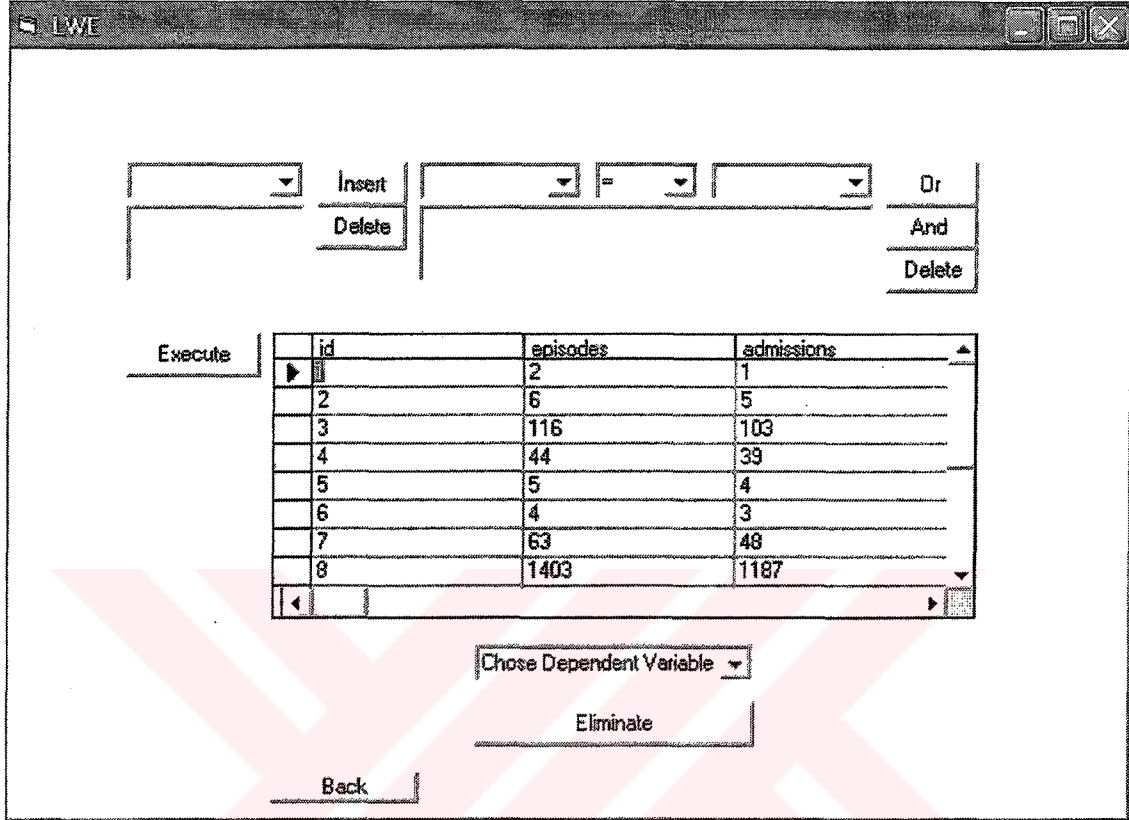
Tablo Seçimi



Şekil 5.4 LWE programı, tablo ve metot seçim ekranı

Üzerinde çalışılacak veritabanının belirlenmesinin ardından program kullanıcıya o veritabanı içindeki tabloların bir listesini görüntüler. Kullanıcı görüntülenen listeden üzerinde analiz yapacağı tabloyu seçer. Burada seçilen kural öğrenme metodu Eliminasyon metoduyla kural öğrenmedir.

Bilginin Analizi, Tablonun getirilmesi, Bağımlı Değişken Seçimi, Eliminasyon Metodunun Uygulanması

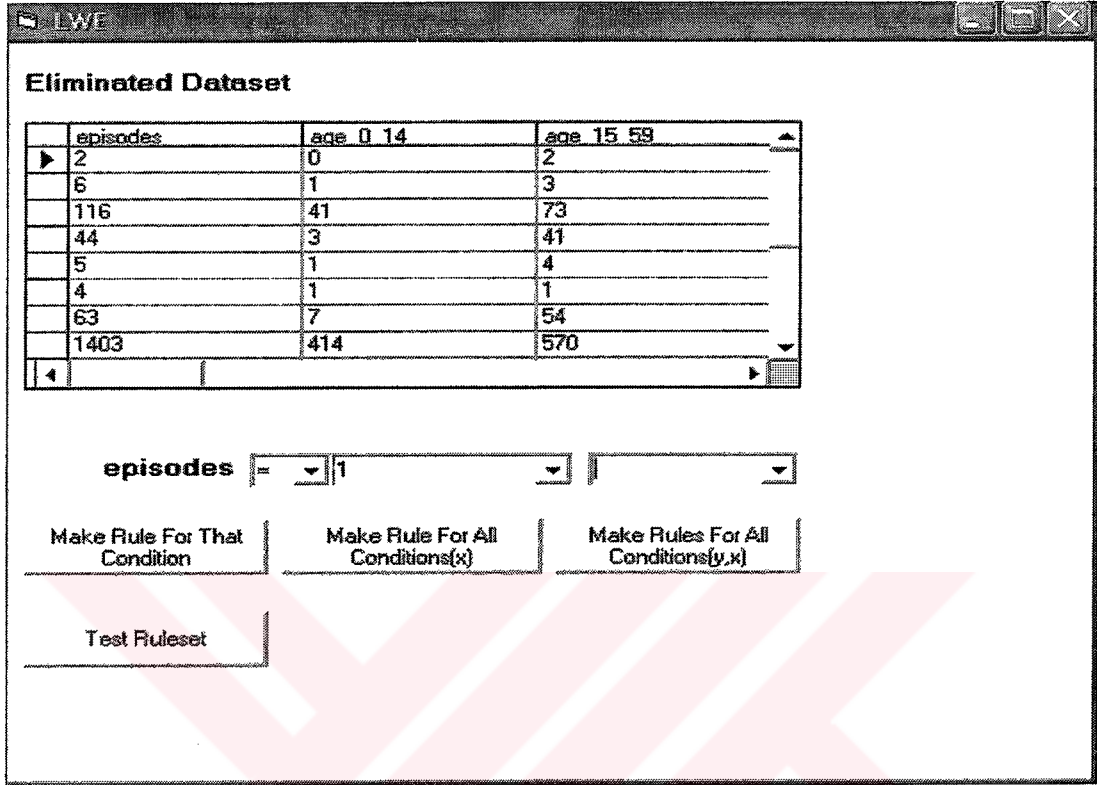


Şekil 5.5 LWE programı, eliminasyon ve kural öğrenilecek değişkenin seçim ekranı

Tablo seçildikten sonra ki ekranda tablonun içindeki bilgiler otomatik olarak program tarafından getirilir. Satır ve sütunlardan oluşmuş tablo değerlerini analiz etmeden önce kullanıcı isteğine göre bu değerleri azaltabilir. Örneğin kullanıcı ben sadece üç sütun için sistemi çalıştırmak istiyorum ya da belli şartları sağlayan verilerle çalışmak istiyorum diye bir seçim yoluna gidebilir.

Bir sonraki adım bağımlı değişken seçme işlemidir. Seçilen bağımlı değişken aynı zamanda üzerinde kural öğrenilecek sütundur. Bu sütun değerleri göz önünde bulundurularak, bu değerlerin diğer sütun değerleriyle olan ilişkisi regresyon metoduyla hesaplanır. İlişkisi daha yüksek olan sütunlar kural öğrenme işlemi için modelde kalacak diğerleri ise elenecektir.

Kuralların Öğrenilmesi, Kuralların Veritabanına Yazılması



Şekil 5.6 LWE programı, kural öğrenilmesi ve kuralların veritabanına yazılması ekranı

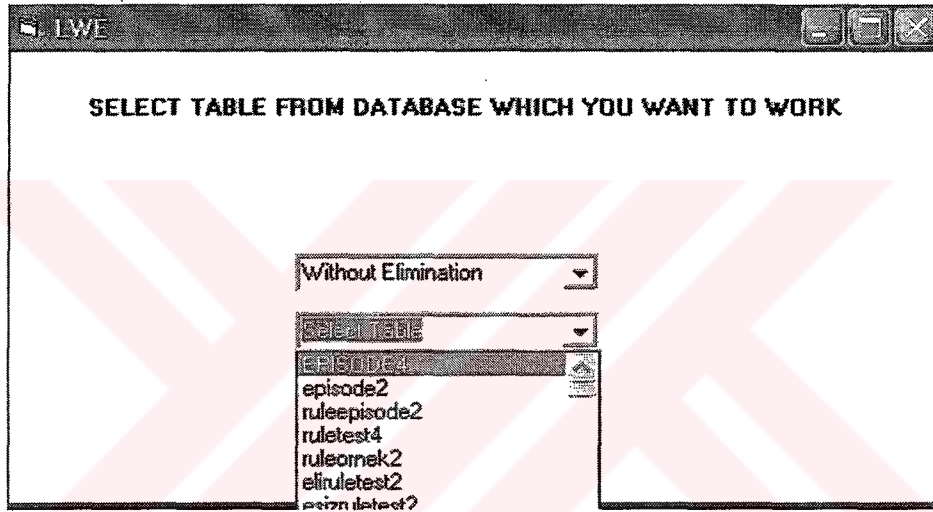
Bağımlı değişken belirlenip sütun elemesi yapıldıktan sonraki işlem kural öğrenme işlemidir. Kural öğrenmek kullanıcı isteğine göre belli şekillerde yapılabilir. Kullanıcı bağımlı değişken dediğimiz yani üzerinde kural öğrenilecek sütun için bir değer belirleyip, ve başka bir sütun belirleyip o değerde o başka sütun için kural üretebilir. Kullanıcı üzerinde kural öğrenilecek sütun için bir değer belirleyip geriye kalan sütunların hepsi için kural öğrenebilir. Kullanıcı üzerinde kural öğrenilecek sütunun her değeri için geriye kalan sütunların hepsi için kural öğrenebilir. Kullanıcı hangi seçimi yapmış olursa olsun programın bir sonraki adımı üretilen kuralları veritabanında bir kural seti tablosu oluşturup o tablonun içine daha sonra kullanılmak üzere yazar. Eğer veri tabanı içerisinde üzerinde çalışılan tablo için oluşturulmuş bir kural seti tablosu mevcut ise yeniden tablo oluşturmaz, varolan tablo içerisine yeni kuralları ekler.

5.5 Eliminasyon Metodu Kullanmaksızın Kural Öğrenen Dinamik Sistem

Veritabanından Dinamik Bilgi Aktarımı

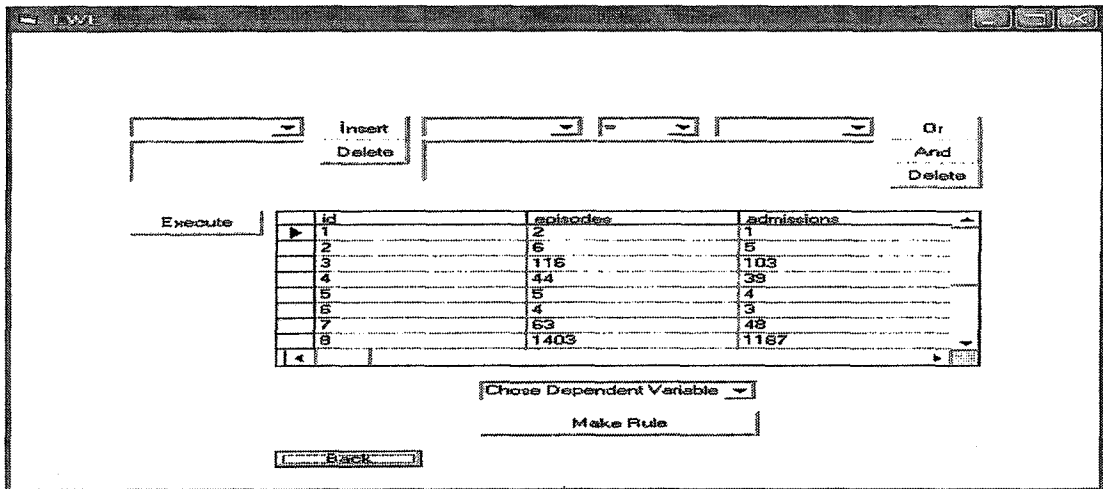
Bu aşamadaki üç işlem SQL Server'a bağlanma, veritabanı seçimi ve tablo seçimi Eliminasyon metodu kullanmaksızın kural öğrenen dinamik sistem için de aynıdır. Ancak kullanıcı tablo seçimi esnasında Eliminasyon Metodu Kullanmaksızın seçeneğini seçmelidir.

Tablo Seçimi



Şekil 5.7 LWE programı tablo ve metod seçimi(eliminasyonsuz)

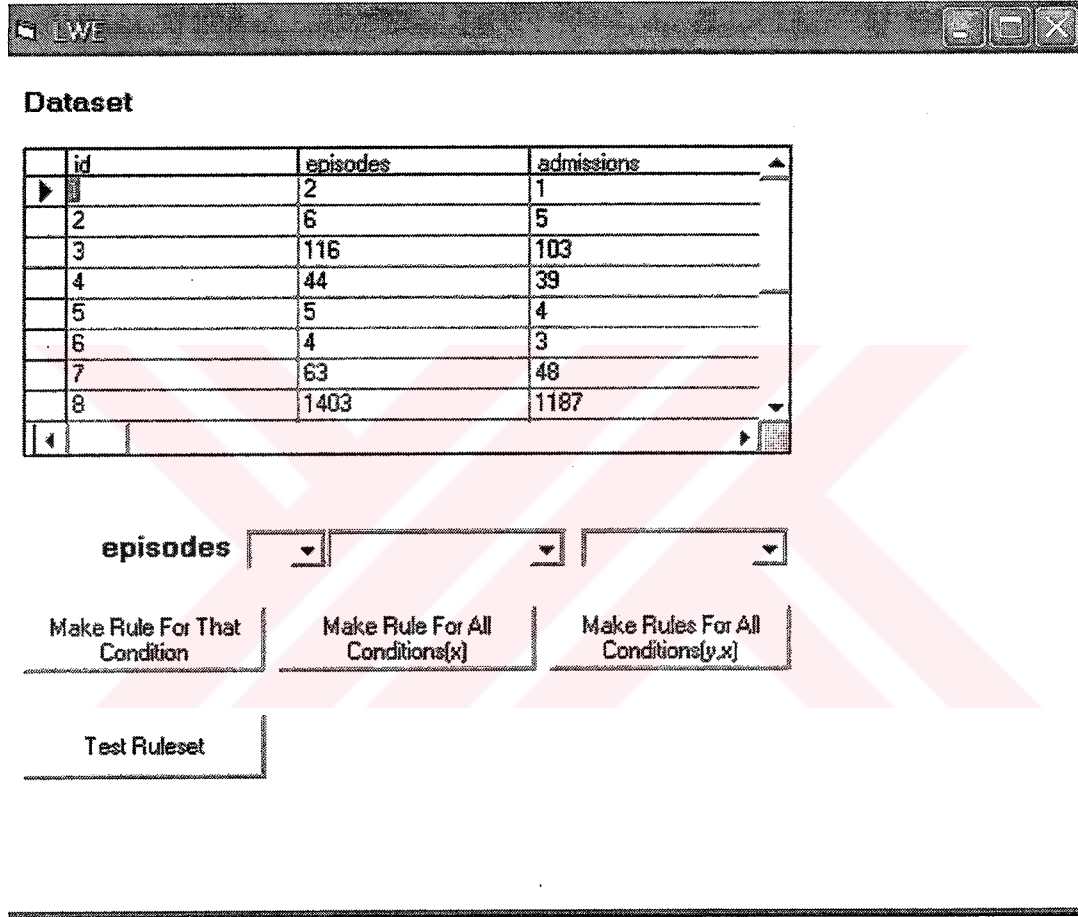
Bu sistemdeki fark bilginin analizi kısmının olmayışdır. Bilgi herhangi bir istatistik metoduna tabi tutulmadan kullanıcının karşısına getirilir.



Şekil 5.8 LWE programı kural öğrenilecek değişkenin seçim ekranı

Kullanıcı üzerinde çalıştığı tablo için üzerinde kural öğreneceği sütunu belirler ve tablo her hangi bir değişikliğe uğramadan, bir başka deyişle sütun eleme işlemi yapılmadan üzerinde kural öğrenilmek için hazır hale gelir.

Kuralların Öğrenilmesi, Kuralların Veritabanına Yazılması



Şekil 5.9 LWE programı,kural öğrenilmesi ve kuralların veritabanına yazılması

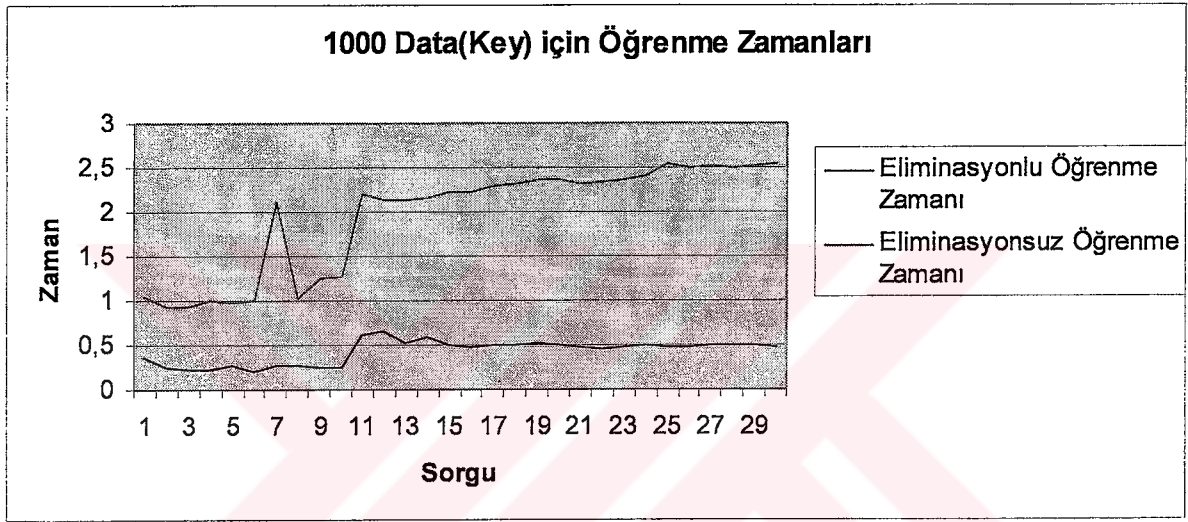
ekranı(eliminasyonsuz)

Programın son adımı olan kural öğrenme işlemi eliminasyon metodu kullanarak kural öğrenen dinamik sisteminkiyle aynıdır. İki sistemin tek farkı birinde tablonun içindeki bilginin analiz edilip sütun eleme işleminin yapılmasının ardından kural öğrenmenin gerçekleştirilmesi,diğerinde ise tablonun orijinal değerleri kullanılarak kural öğrenme işleminin gerçekleştirilmesidir.

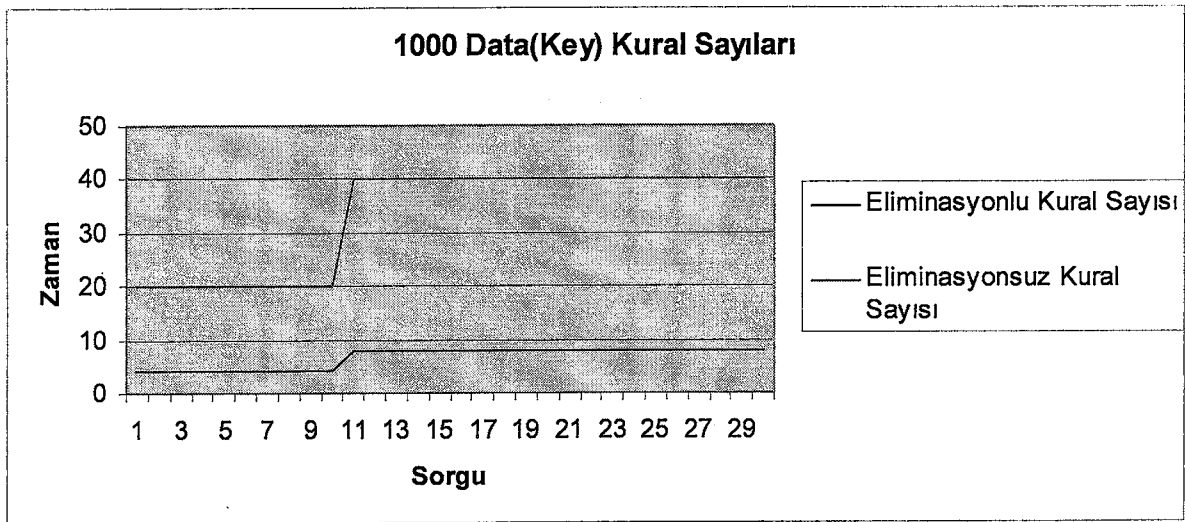
5.6 Uygulama Sonuçları

İki metodu birbiriyle öğrenilen kural sayıları ve kural öğrenme zamanlarına göre kıyaslayacak olur aşağıdaki sonuçları elde ederiz.

5.6.1 1000 Datalık Tabloda Anahtar Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının ve Kural Sayılarının Kıyaslanması

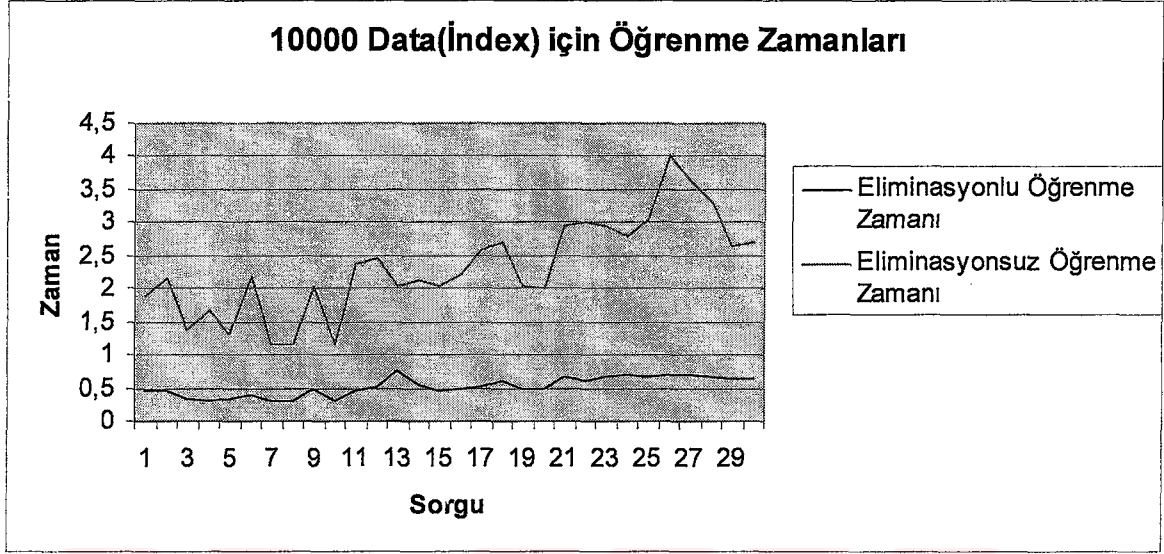


Şekil 5.10 1000 data için öğrenme zamanları

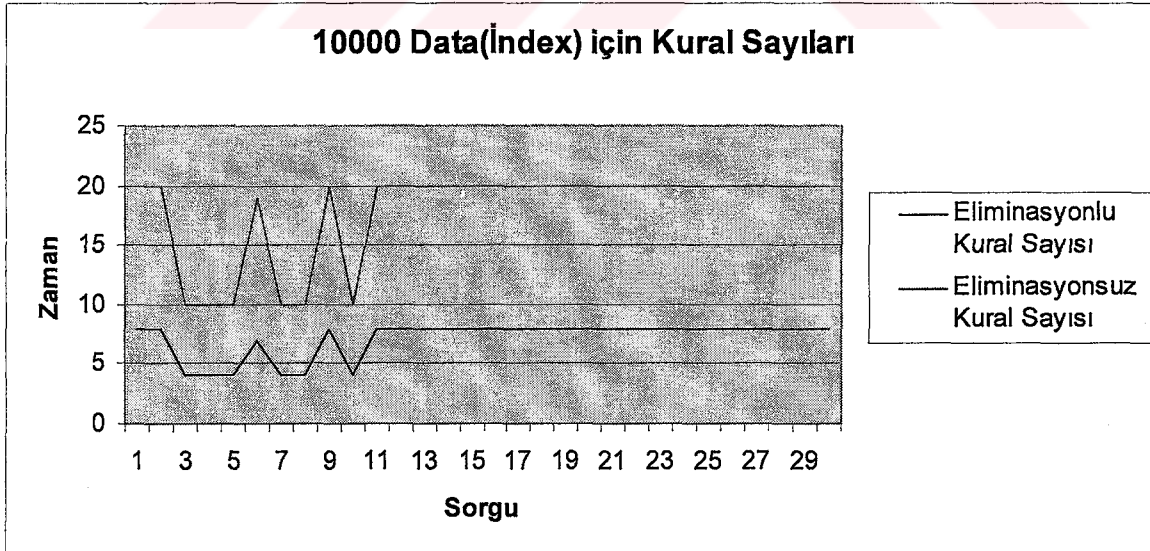


Şekil 5.11 1000 data için kural sayıları

5.6.2 10000 Datalık Tabloda Dizin(index) Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının Kıyaslanması

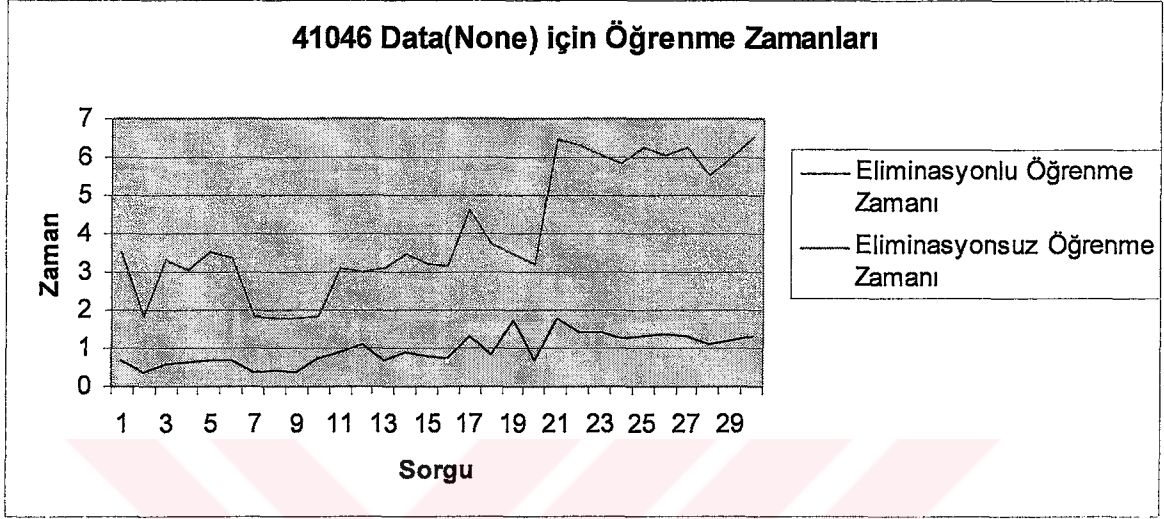


Şekil 5.12 10000 data için öğrenme zamanları

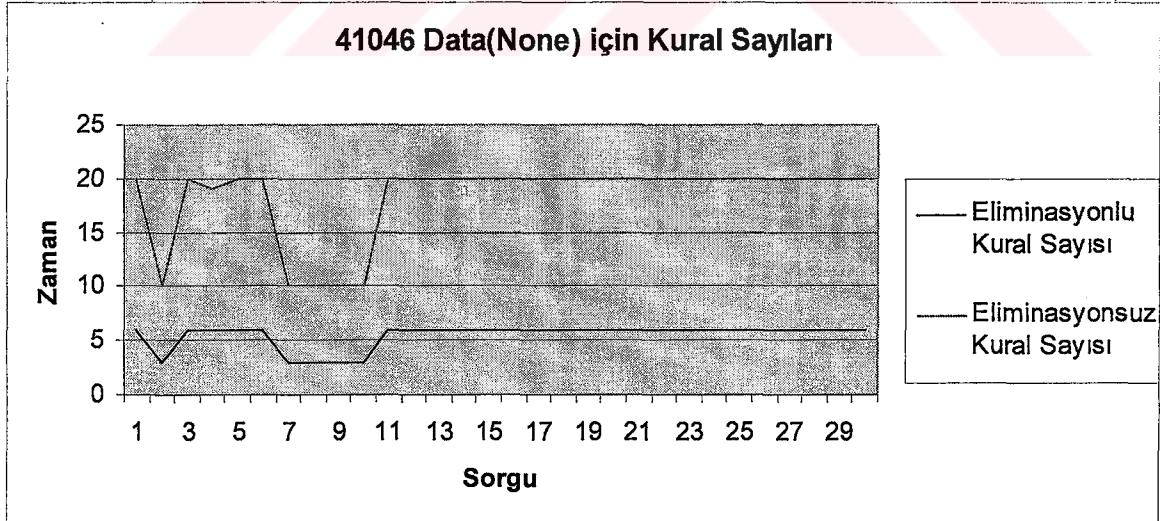


Şekil 5.13 10000 data için kural sayıları

5.6.3 41046 Datalık Tabloda Özelliksiz(None) Değeri İçin Eliminasyonlu ve Eliminasyonsuz Kural Öğrenme Zamanlarının Kıyaslanması



Şekil 5.14 41046 data için öğrenme zamanları



Şekil 5.15 41046 data için öğrenme zamanları

Yukarıdaki grafikler sırasıyla 1000,10000 ve 41046 data için anahtar, dizin(index) ve none değerler için çalıştırılan sorgularda kural öğrenme zamanlarını ve kural sayılarını her iki metot için de göstermektedir. Grafiklerden de görüldüğü üzere eliminasyon metodu

kullanılarak yapılan kural öğrenme zamanları eliminasyon metodu kullanmaksızın yapılan kural öğrenme zamanlarından her noktada yani her sorgu için daha azdır. Aynı zamanda öğrenilen kural sayıları da eliminasyon metodu kullanılarak yapılan öğrenmede her noktada yani her sorgu için daha azdır. Bu noktada diyebiliriz ki veritabanında regresyon metodu kullanılarak yapılan sütun eleme işleme hem öğrenme zamanını düşürür, öğrenme işleminde zamandan tasarruf sağlar hem de kural setine yazılan kural sayısının azalmasına yol açar. Sistem birbiriyle yüksek bağımlılığı bulunmayan sütunları eleyerek daha anlamlı bir kural setini oluşturur.

5.6.4 Genel Analiz

Aşağıdaki tabloda 1000,10000,41046 data için anahtar,dizin(index) ve none değerler için çalıştırılan sorgularda öğrenilen toplam kural sayılarını,toplam öğrenme zamanlarını her iki metot için de görebiliriz

Çizelge 5.3 Öğrenmeler için eliminasyonlu ve eliminasyonsuz metodun genel analizi

<i>Anahtar(key)</i>	Eliminasyon Metodu kullanarak			Eliminasyon Metodu Kullanmaksızın		
	1000 Data	10.000 Data	41.046 Data	1000 Data	10.000 Data	41.046 Data
Kural Sayısı	200	200	100	1000	500	500
Toplam Öğrenme Zamanı	12,809082	17,1425781	14,1688453	58,5048838	59,4863281	102,196289
Ortalama Öğrenme Zamanı	0,4269694	0,57141927	0,472294843	1,95016279	1,982877603	3,406542967
Ortalama Eliminasyon Zamanı	0,0112	0,0114	0,008			
Kazanılan zaman	0,4381694	0,58281927	0,480294843	1,95016279	1,982877603	3,406542967
<i>Dizin(index)</i>	Eliminasyon Metodu kullanarak			Eliminasyon Metodu Kullanmaksızın		
	1000 Data	10.000 Data	41.046 Data	1000 Data	10.000 Data	41.046 Data
Kural Sayısı	240	215	220	600	539	550
Toplam Öğrenme Zamanı	16,8139648	15,6240234	36,0234375	52,8637395	69,7597656	119,639648
Ortalama Öğrenme	0,560465493	0,52080078	1,20078125	1,76212465	2,32532552	3,987988267

Zamanı						
Ortalama Eliminasyon Zamanı	0,0112	0,0114	0,008			
Kazanılan zaman	0,571665493	0,53220078	1,20878125	1,76212465	2,32532552	3,987988267
None	Eliminasyon Metodu kullanarak			Eliminasyon Metodu Kullanmaksızın		
	1000 Data	10.000 Data	41.046 Data	1000 Data	10.000 Data	41.046 Data
Kural Sayısı	298	159	165	597	529	549
Toplam Öğrenme Zamanı	22,8656342	13,8886719	28,7480469	52,4121094	75,7451172	121,44043
Ortalama Öğrenme Zamanı	0,762187807	0,46295573	0,95826823	1,74707031	2,52483724	4,048014333
Ortalama Eliminasyon Zamanı	0,0112	0,0114	0,008			
Kazanılan zaman	0,773387807	0,47435573	0,96626823	1,74707031	2,52483724	4,048014333

5.7 Kural Oluşturulması ve Kural Seti

5.7.1 Kural Tablosu Genel Özellikleri

Kural tablosu 13 sütundan oluşan bir veri tablosudur. Kural seti sütunları sırasıyla Çizelge 5.4'te gösterilmiştir.

Çizelge 5.4 Kural seti sütunları veri tipleri

Adı	Tipi	Açıklaması
Antatt	varchar	Kural tablosunun sol tarafı-adı
Antop	varchar	Kural tablosunun sol tarafı-operatörü
Antvalue	float	Kural tablosunun sol tarafı-değeri
Antindex	float	Kural tablosunun sol tarafı-dizin(index)
Antkey	float	Kural tablosunun sol tarafı-anahtar(key)
Antrn	float	Kural tablosunun sol tarafı-koşulu sağlayan değer sayısı
Consatt	varchar	Kural tablosunun sağ tarafı-adı
Consop	varchar	Kural tablosunun sağ tarafı-operatörü
Consvalue	float	Kural tablosunun sağ tarafı-değeri
Consindex	float	Kural tablosunun sağ tarafı-dizin(index)
Conskey	float	Kural tablosunun sağ tarafı-anahtar(key)
Consrn	float	Kural tablosunun sağ tarafı-koşulu sağlayan değer sayısı
Rlt	float	Kural öğrenme zamanı

Kural oluşturmak demek sorguyu yeniden formüle etmektir. Sorgu kural tablosuyla birlikte başka bir yapı kazanır. Örneğin sorgumuz Select from episode5 where episodes=1 olsun Yukarıda ki SQL komut satırı episode5 tablosunun içinden episodes değerinin bir olduğu satırı veya satırları arayacaktır. Bu sorgu için ya da bir başka deyişle episode5 tablosunda episodes=1 değeri için kural öğrenecek olursak öğrenilen kural, kural tablosu içindeki yerini şu şekilde alır

Çizelge 5.5 “Select * from episode5 where episodes=1” sorgusu için üretilen kurallar

Antatt	Antop	antvalue	antindex	antkey	antm	consatt	consop	consvalue	consindex	conskey	Consm
EPISODES	=	1	1	0	1995	ID	<=	41259	0	1	41046
EPISODES	=	1	1	0	1995	ID	>=	15	0	1	41032
EPISODES	=	1	1	0	1995	ADMISSIONS	<=	1	0	0	2319
EPISODES	=	1	1	0	1995	ADMISSIONS	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	MALE	<=	1	0	0	7514
EPISODES	=	1	1	0	1995	MALE	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	EMERGENCY	<=	1	0	0	7857
EPISODES	=	1	1	0	1995	EMERGENCY	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	WAITING_LIST	<=	1	0	0	10403
EPISODES	=	1	1	0	1995	WAITING_LIST	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE 0 14	<=	1	0	0	16199
EPISODES	=	1	1	0	1995	AGE 0 14	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE 15 59	<=	1	0	0	6408
EPISODES	=	1	1	0	1995	AGE 15 59	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE 60 74	<=	1	0	0	15340
EPISODES	=	1	1	0	1995	AGE 60 74	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE 75	<=	1	0	0	16950
EPISODES	=	1	1	0	1995	AGE 75	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	DAY CASE	<=	1	0	0	14774
EPISODES	=	1	1	0	1995	DAY CASE	>=	0	0	0	41046

“Select * from episode5 where episodes=1” sorgusunun yeniden yapılandırılmış şekli kural tablosunda görülmektedir. Episode2 tablosu için episode=5 değerine sahip iken(bu kural tablosunun sol tarafını oluşturmaktadır) diğer sütunların aldıkları değerler kural tablosunun sağ tarafını oluşturur. Örneğin episode=1 iken episode2 tablosunda age_15_59 <= 1 ve age_15_59>=0 şartını sağlamaktadır. Yani episode=1 değeri için age_15_59 değeri [0,1] aralığındadır.

Bu da aynı sorguyu şu şekilde çalıştırma olanağı sağlamaktadır

Select * from episode5 where age_15_59<=1 and age_15_59>=0 and episodes=1

Böylece sorgu çalıştırılırken episodes=1 değerini aramak için sadece age_15_59 <=1 ve age_15_59>=0 şartını sağlayan satırları göz önünde bulunduracaktır. Bu şartı sağlamayan satırlara bakmayacaktır. Yazılım olarak bu iki sorgu birbirinden farklı ise de anlamsal olarak denktir. Bu şekilde kuralların kullanılarak verilen sorgunun tekrardan yapılandırılmasına ‘Anlamsal Sorgulama Optimizasyonu (Semantic Query Optimisation)’ adı verilir (Sayli ve Lowden 1996, 1997). Anlamsal olarak denklikle bu iki sorguda aynı sonuç değerlerine sahiptir.

Yukarıdaki tablodan da görüldüğü üzere episode=1 değerini için 20 tane kural üretilmiştir. Bu kurallar orijinal tablo içindeki diğer tüm sütunların episode=1 koşulu sağlandığında aldığı değerlerlerdir. Ancak tüm sütun değerleri için üretilen kurallar anlamlı olmayabilir. Anlamsız kuralları kural setinde bulundurmamak sorguyu yavaşlatabilir ve kural setinde gereksiz şişmeye neden olabilir. Bu sebeplerden dolayı sütun eliminasyonu yapmak daha anlamlı kurallar yaratılmasını sağlayacaktır.

Aynı sorgu için sütun eliminasyonu yaparak kural oluşturacak olursak üretilen kurallar kural tablosundaki yerini şu şekilde alır.

Çizelge 5.6“Select * from episode5 where episodes=1” sorgusu için eliminasyon metodu uygulanarak üretilen kurallar

Antatt	antop	antvalue	antindex	antkey	anttrn	consatt	consop	consvalue	consindex	conskey	constrn
EPISODES	=	1	1	0	1995	AGE_0_14	<=	1	0	0	16199
EPISODES	=	1	1	0	1995	AGE_0_14	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_60_74	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	ID	>=	15	0	1	41032
EPISODES	=	1	1	0	1995	ID	<=	41259	0	1	41046
EPISODES	=	1	1	0	1995	AGE_75	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_60_74	<=	1	0	0	15340
EPISODES	=	1	1	0	1995	AGE_75	<=	1	0	0	16950
EPISODES	=	1	1	0	1995	AGE_15_59	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_15_59	<=	1	0	0	6408

Sütun elemesi yapıldıktan sonra oluşturulan kural sayısı on olmuştur. Bu hem öğrenmeyi hızlandıran hem de kural setini anlamsız kuralların yarattığı şişmelerden arındıran bir faktördür. Tablodan da görüldüğü üzere bazı sütun değerleri için kural öğrenilmemiştir. (EPISODE5 tablosunun ilk 100 kayıtlık bölümü Ek 1’de, eliminasyon yapılarak öğrenilen kuralların oluşturduğu tablonun ilk 10 sorgu için olan bölümü Ek 2’de, eliminasyon metodu

kullanılmaksızın öğrenilen kuralların oluşturduğu tablonun ilk 10 sorgu için olan kısmı Ek 3’de mevcuttur)

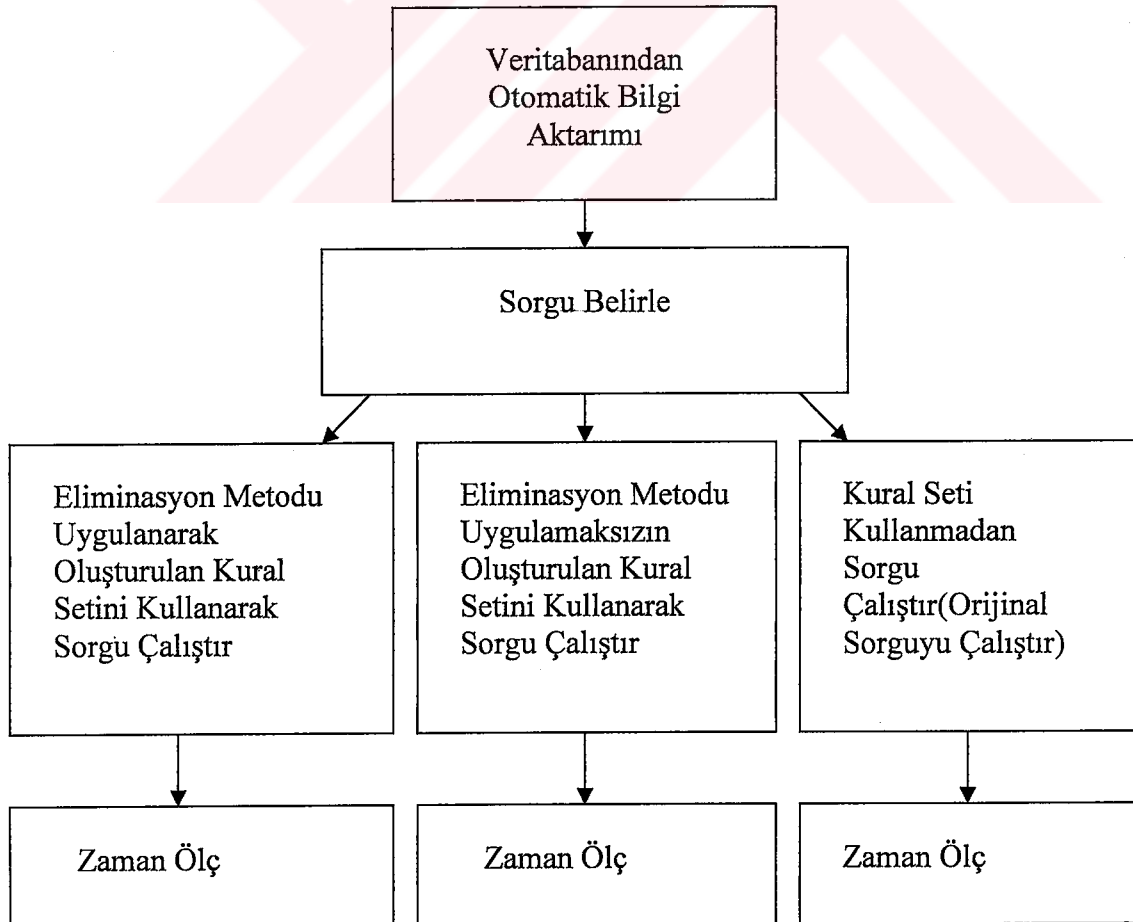


6 SORGUNUN ÇALIŞTIRILMASI VE SORGU KALİTESİNİN TESTİ

Kural öğrenmenin amacı veritabanı üzerinde sorguyu çalıştırırken zamandan tasarruf sağlamaktır. Üretilen kurallar sorgu esnasında zaman maliyetinin azalmasını sağlamaktır. Eliminasyon metodu uygulanarak gerçekleştirilen öğrenme de kazanılan zamanın maksimuma ulaşmasını amaçlamaktadır.

Sorgu kalitesinin testi kural seti kullanılarak yapılan sorguyla kullanmadan yapılan sorgunun zamanını ölçme işlemiyle mümkündür. Zaman ölçme işlemi bize her sorgu için kazanılan ya da kaybedilen zamanı göstermektedir. Ayrıca eliminasyon metodu uygulayarak ve uygulamaksızın oluşturulan kural setlerindeki kuralları kullanarak çalıştırdığımız sorgularda ölçülen zamanlarla hangi metodun zaman açısından daha karlı olduğunu saptamak mümkündür.

Sistemin çalışma metodolojisini şu şekilde gösterebiliriz.



Şekil 6.1 Sorgu kalitesinin testi, çalışma metodolojisi

Sistemin Çalıştırılmasının ardından elimizde her sorgu için üç tane zaman olacaktır.

Bunlar

- Eliminasyon metodu uygulanarak oluşturulan kural setini kullanarak çalıştırılan sorgunun zamanı
- Eliminasyon metodu kullanmaksızın oluşturulan kural setini kullanarak çalıştırılan sorgunun zamanı
- Kural seti kullanmaksızın çalıştırılan sorgunun zamanı

Sorgu kalitesinin testi bu üç zamanın kıyaslanmasıyla mümkündür.

6.1 Uygulama

Tüm uygulamalar için kullanılan:

Program Geliştirme Ortamı: Visual Basic 6.0

Veritabanı: SQL Server 8.0

Veri Erişim Yöntemi:ADO

Bu uygulama için kullanılan:

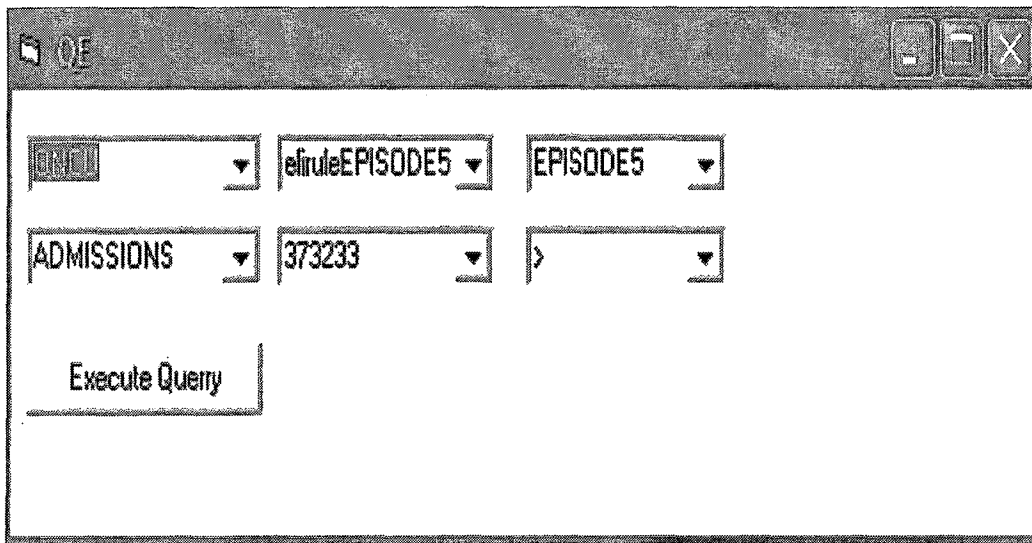
Veri Tablosu:episode5

Veri Sayısı:41046

Sütun Sayısı:11

Tablo Anahtar Değeri:id

Tablo dizin(index) değeri:episodes



Şekil 6.2 QE programı, sorgunun çalıştırılması ekranı

Uygulama veritabanından otomatik bilgi aktarımıyla başlar. Kullanılacak kural tablosu ve orijinal tablo kullanıcı tarafından belirlenir. Bir sonraki adım koşul belirleme işlemidir.

Koşul belirlendikten sonra “execute query” butonuna basılmasının ardından sorgu hem belirlenen orijinal koşul için hem de kural tablosunu kullanarak(belirlenen koşulu kural tablosunda sağlayan koşullar sorguya dahil edilir) çalıştırılır. Her ikisinin de çalışma zamanları kıyaslama kolaylığı sağlamak için bir Excel dosyasına yazılır. Zaman değerlerindeki hata payını gidermek için her sorgu bir döngü içerisinde on defa çalıştırılır. Bulunan on zamanın ortalaması alınır. Excel dosyasına yazılan değerler ortalaması alınmış zaman değerleridir.

6.2 Uygulama Sonuçları(1)

Kural tablosunu sağ ve sol olmak üzere iki kısım olarak düşünebiliriz. Kural tablosunun sol tarafı orijinal sorgu koşulları sağ tarafı ise orijinal sorgu koşullarını sağlayan diğer sütunlar için oluşturulmuş koşullardır.

Verimliliği ölçmek için sol ve sağ taraf için belli kriterler belirlenip sorgular bu kriterlere göre çalıştırılır ise zaman kazancının maksimum olduğu kriterleri ölçmemiz mümkün olacaktır. Bilindiği gibi sorgular anahtar(key) değerler ve dizin(index) değerler üzerinde daha hızlı çalışır. Özelliiksiz(none) değerler için sorgu çalıştırmak zaman maliyetini artıran bir unsurdur. Öyleyse diyebiliriz ki sol tarafı özelliiksiz(none) olan ve sağ tarafında dizin(index) değerine sahip olan bir kural zamandan tasarruf edilmesine neden olacaktır.

Orijinal tablo:episode5

Tablo sütun sayısı:11

Satır sayısı:41046

Tablo anahtar değeri:ID

Tablo dizin(index) değeri:EPISODES

Kural tablosu1:eliruleepisode5

Sütun elemesinden sonra oluşturulan kurallardan meydana gelen tablo

Tablo sütun sayısı:13

Toplam kural sayısı(satır sayısı):2681

Kural tablosu2:esizruleepisode5

Sütun elemesi yapılmadan oluşturulan kurallardan meydana gelen tablo

Tablo sütun sayısı:13

Toplam kural sayısı(satır sayısı):8507

Üzerinde çalışılacak(zaman ölçümü yapılacak) üç tablonun belirlenmesinin ardından sorgu belirleme işlemi gerçekleştirilir. Zaman ölçümü yapmak için özelliiksiz(none) değerler üzerinde arama işlemi yapacak 118 sorgu belirlenir. Belirlenen 118 sorgu aşağıda listelenmiştir.

Belirlenen sorgulardaki kriter aranan değerin none olma özelliğidir.'ADMISSIONS' değeri orijinal tablo EPISODE5'in özelliiksiz (none) olan yani dizin (index) yada anahtar (key) olma özelliğine sahip olmayan sütunudur

SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4274
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	6272
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	54345
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	78300
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	11431
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2771
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	29790
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	34638
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	44717
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13933
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9771
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4861
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	19308
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3272
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4529
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4446
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	12522
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2036
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	16257
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5691
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3770
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	73249
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	7191
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	14514
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3604
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5110
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	28130
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3189
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5946
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8027
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3939
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5697
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5614
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	61484
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2774

SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8348
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8187
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8265
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3020
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4019
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	14929
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1827
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1744
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2356
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3355
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4612
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13352
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	6444
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	7944
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	7861
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1412
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3856
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	7612
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5365
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	21822
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	10779
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5282
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	35018
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5199
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5116
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5033
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	24075
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13945
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	23577
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9700
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2359
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	6782
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4618
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	6699
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13447
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	222241
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4535
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9202
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3275
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2276
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1621
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13032
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	12866
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2193
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	70144
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3109
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8455
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	4120
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	33216
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3026
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	16281
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	16115
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1996
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	28083

SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3942
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5869
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3859
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1913
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	34851
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5620
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3776
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	52046
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	5371
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	2611
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3610
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	14455
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	26303
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	17979
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	238922
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3527
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1747
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	47229
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	3444
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9712
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	1206
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	25141
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9546
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	50315
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	6705
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	13293
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	9048
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	23813
SELECT * FROM EPISODE5 WHERE	ADMISSIONS	=	8882

Elirulepisode5 tablosunun kullanarak yeniden yapılandırılmış sorgular ise aşağıdaki gibi listelenir. Buradaki kriter kura tablosundan elde edilecek olan kuralların(sağ tarafın) dizin (index) olma özelliğidir.

SELECT * FROM EPISODE5	ADMISSIONS=1206 and EPISODES=1377
SELECT * FROM EPISODE5	ADMISSIONS=1412 and EPISODES=1527
SELECT * FROM EPISODE5	ADMISSIONS=1621 and EPISODES=1630
SELECT * FROM EPISODE5	ADMISSIONS=1744 and EPISODES=1799
SELECT * FROM EPISODE5	ADMISSIONS=1747 and EPISODES=2386
SELECT * FROM EPISODE5	ADMISSIONS=1827 and EPISODES=1854
SELECT * FROM EPISODE5	ADMISSIONS=1913 and EPISODES=2074
SELECT * FROM EPISODE5	ADMISSIONS=1996 and EPISODES=2028
SELECT * FROM EPISODE5	ADMISSIONS=2036 and EPISODES=2240
SELECT * FROM EPISODE5	ADMISSIONS=2193 and EPISODES=2374
SELECT * FROM EPISODE5	ADMISSIONS=2276 and EPISODES=2343
SELECT * FROM EPISODE5	ADMISSIONS=2356 and EPISODES=3046
SELECT * FROM EPISODE5	ADMISSIONS=2359 and EPISODES=2394
SELECT * FROM EPISODE5	ADMISSIONS=2611 and EPISODES=3400
SELECT * FROM EPISODE5	ADMISSIONS=2771 and EPISODES=3315
SELECT * FROM EPISODE5	ADMISSIONS=2774 and EPISODES=3583
SELECT * FROM EPISODE5	ADMISSIONS=3020 and EPISODES=3074
SELECT * FROM EPISODE5	ADMISSIONS=3026 and EPISODES=3316

SELECT * FROM EPISODE5	ADMISSIONS=3109 and EPISODES=3225
SELECT * FROM EPISODE5	ADMISSIONS=3189 and EPISODES=3271
SELECT * FROM EPISODE5	ADMISSIONS=3272 and EPISODES=4170
SELECT * FROM EPISODE5	ADMISSIONS=3275 and EPISODES=3613
SELECT * FROM EPISODE5	ADMISSIONS=3355 and EPISODES=3379
SELECT * FROM EPISODE5	ADMISSIONS=3444 and EPISODES=3548
SELECT * FROM EPISODE5	ADMISSIONS=3527 and EPISODES=3795
SELECT * FROM EPISODE5	ADMISSIONS=3604 and EPISODES=3647
SELECT * FROM EPISODE5	ADMISSIONS=3610 and EPISODES=4356
SELECT * FROM EPISODE5	ADMISSIONS=3770 and EPISODES=3846
SELECT * FROM EPISODE5	ADMISSIONS=3776 and EPISODES=4014
SELECT * FROM EPISODE5	ADMISSIONS=3856 and EPISODES=3888
SELECT * FROM EPISODE5	ADMISSIONS=3859 and EPISODES=3927
SELECT * FROM EPISODE5	ADMISSIONS=3939 and EPISODES=4008
SELECT * FROM EPISODE5	ADMISSIONS=3942 and EPISODES=3976
SELECT * FROM EPISODE5	ADMISSIONS=4019 and EPISODES=4137
SELECT * FROM EPISODE5	ADMISSIONS=4120 and EPISODES=4126
SELECT * FROM EPISODE5	ADMISSIONS=4274 and EPISODES=5146
SELECT * FROM EPISODE5	ADMISSIONS=4446 and EPISODES=4457
SELECT * FROM EPISODE5	ADMISSIONS=4529 and EPISODES=4655
SELECT * FROM EPISODE5	ADMISSIONS=4535 and EPISODES=4543
SELECT * FROM EPISODE5	ADMISSIONS=4612 and EPISODES=5140
SELECT * FROM EPISODE5	ADMISSIONS=4618 and EPISODES=4653
SELECT * FROM EPISODE5	ADMISSIONS=4861 and EPISODES=5006
SELECT * FROM EPISODE5	ADMISSIONS=5033 and EPISODES=5055
SELECT * FROM EPISODE5	ADMISSIONS=5110 and EPISODES=5798
SELECT * FROM EPISODE5	ADMISSIONS=5116 and EPISODES=5424
SELECT * FROM EPISODE5	ADMISSIONS=5199 and EPISODES=5718
SELECT * FROM EPISODE5	ADMISSIONS=5282 and EPISODES=5294
SELECT * FROM EPISODE5	ADMISSIONS=5365 and EPISODES=12146
SELECT * FROM EPISODE5	ADMISSIONS=5371 and EPISODES=5470
SELECT * FROM EPISODE5	ADMISSIONS=5614 and EPISODES=5791
SELECT * FROM EPISODE5	ADMISSIONS=5620 and EPISODES=5643
SELECT * FROM EPISODE5	ADMISSIONS=5691 and EPISODES=5991
SELECT * FROM EPISODE5	ADMISSIONS=5697 and EPISODES=6712
SELECT * FROM EPISODE5	ADMISSIONS=5869 and EPISODES=6336
SELECT * FROM EPISODE5	ADMISSIONS=5946 and EPISODES=6293
SELECT * FROM EPISODE5	ADMISSIONS=6272 and EPISODES=6738
SELECT * FROM EPISODE5	ADMISSIONS=6444 and EPISODES=7212
SELECT * FROM EPISODE5	ADMISSIONS=6699 and EPISODES=6768
SELECT * FROM EPISODE5	ADMISSIONS=6705 and EPISODES=7393
SELECT * FROM EPISODE5	ADMISSIONS=6782 and EPISODES=7386
SELECT * FROM EPISODE5	ADMISSIONS=7191 and EPISODES=9838
SELECT * FROM EPISODE5	ADMISSIONS=7612 and EPISODES=12965
SELECT * FROM EPISODE5	ADMISSIONS=7861 and EPISODES=8199
SELECT * FROM EPISODE5	ADMISSIONS=7944 and EPISODES=8055
SELECT * FROM EPISODE5	ADMISSIONS=8027 and EPISODES=8042
SELECT * FROM EPISODE5	ADMISSIONS=8187 and EPISODES=8297
SELECT * FROM EPISODE5	ADMISSIONS=8265 and EPISODES=9705
SELECT * FROM EPISODE5	ADMISSIONS=8348 and EPISODES=9124
SELECT * FROM EPISODE5	ADMISSIONS=8455 and EPISODES=8530
SELECT * FROM EPISODE5	ADMISSIONS=8882 and EPISODES=10878
SELECT * FROM EPISODE5	ADMISSIONS=9048 and EPISODES=10856
SELECT * FROM EPISODE5	ADMISSIONS=9202 and EPISODES=9221
SELECT * FROM EPISODE5	ADMISSIONS=9546 and EPISODES=9569
SELECT * FROM EPISODE5	ADMISSIONS=9700 and EPISODES=9726
SELECT * FROM EPISODE5	ADMISSIONS=9712 and EPISODES=11009
SELECT * FROM EPISODE5	ADMISSIONS=9771 and EPISODES=10281
SELECT * FROM EPISODE5	ADMISSIONS=10779 and EPISODES=10864
SELECT * FROM EPISODE5	ADMISSIONS=11431 and EPISODES=11856
SELECT * FROM EPISODE5	ADMISSIONS=12522 and EPISODES=13706
SELECT * FROM EPISODE5	ADMISSIONS=12866 and EPISODES=13915

SELECT * FROM EPISODE5	ADMISSIONS=13032 and EPISODES=13719
SELECT * FROM EPISODE5	ADMISSIONS=13293 and EPISODES=13324
SELECT * FROM EPISODE5	ADMISSIONS=13352 and EPISODES=13396
SELECT * FROM EPISODE5	ADMISSIONS=13447 and EPISODES=13790
SELECT * FROM EPISODE5	ADMISSIONS=13933 and EPISODES=19035
SELECT * FROM EPISODE5	ADMISSIONS=13945 and EPISODES=16923
SELECT * FROM EPISODE5	ADMISSIONS=14455 and EPISODES=14813
SELECT * FROM EPISODE5	ADMISSIONS=14514 and EPISODES=17225
SELECT * FROM EPISODE5	ADMISSIONS=14929 and EPISODES=15143
SELECT * FROM EPISODE5	ADMISSIONS=16115 and EPISODES=16383
SELECT * FROM EPISODE5	ADMISSIONS=16257 and EPISODES=22194
SELECT * FROM EPISODE5	ADMISSIONS=16281 and EPISODES=16824
SELECT * FROM EPISODE5	ADMISSIONS=17979 and EPISODES=19360
SELECT * FROM EPISODE5	ADMISSIONS=19308 and EPISODES=26027
SELECT * FROM EPISODE5	ADMISSIONS=21822 and EPISODES=23539
SELECT * FROM EPISODE5	ADMISSIONS=23577 and EPISODES=23750
SELECT * FROM EPISODE5	ADMISSIONS=23813 and EPISODES=29482
SELECT * FROM EPISODE5	ADMISSIONS=24075 and EPISODES=33250
SELECT * FROM EPISODE5	ADMISSIONS=25141 and EPISODES=25177
SELECT * FROM EPISODE5	ADMISSIONS=26303 and EPISODES=28355
SELECT * FROM EPISODE5	ADMISSIONS=28083 and EPISODES=29023
SELECT * FROM EPISODE5	ADMISSIONS=28130 and EPISODES=28232
SELECT * FROM EPISODE5	ADMISSIONS=29790 and EPISODES=36923
SELECT * FROM EPISODE5	ADMISSIONS=33216 and EPISODES=33456
SELECT * FROM EPISODE5	ADMISSIONS=34638 and EPISODES=40790
SELECT * FROM EPISODE5	ADMISSIONS=34851 and EPISODES=38242
SELECT * FROM EPISODE5	ADMISSIONS=35018 and EPISODES=35074
SELECT * FROM EPISODE5	ADMISSIONS=44717 and EPISODES=44795
SELECT * FROM EPISODE5	ADMISSIONS=47229 and EPISODES=60252
SELECT * FROM EPISODE5	ADMISSIONS=50315 and EPISODES=52578
SELECT * FROM EPISODE5	ADMISSIONS=52046 and EPISODES=52478
SELECT * FROM EPISODE5	ADMISSIONS=54345 and EPISODES=72426
SELECT * FROM EPISODE5	ADMISSIONS=61484 and EPISODES=75326
SELECT * FROM EPISODE5	ADMISSIONS=70144 and EPISODES=73293
SELECT * FROM EPISODE5	ADMISSIONS=73249 and EPISODES=73924
SELECT * FROM EPISODE5	ADMISSIONS=78300 and EPISODES=102119
SELECT * FROM EPISODE5	ADMISSIONS=222241 and EPISODES=248695
SELECT * FROM EPISODE5	ADMISSIONS=238922 and EPISODES=265713

Elsizrulepisode5 tablosunun kullanarak yeniden yapılandırılmış sorgular ise aşağıdaki gibi listelenir Buradaki kriter kural tablosundan elde edilecek olan kuralların(sağ tarafın) dizin(index) olma özelliğidir.

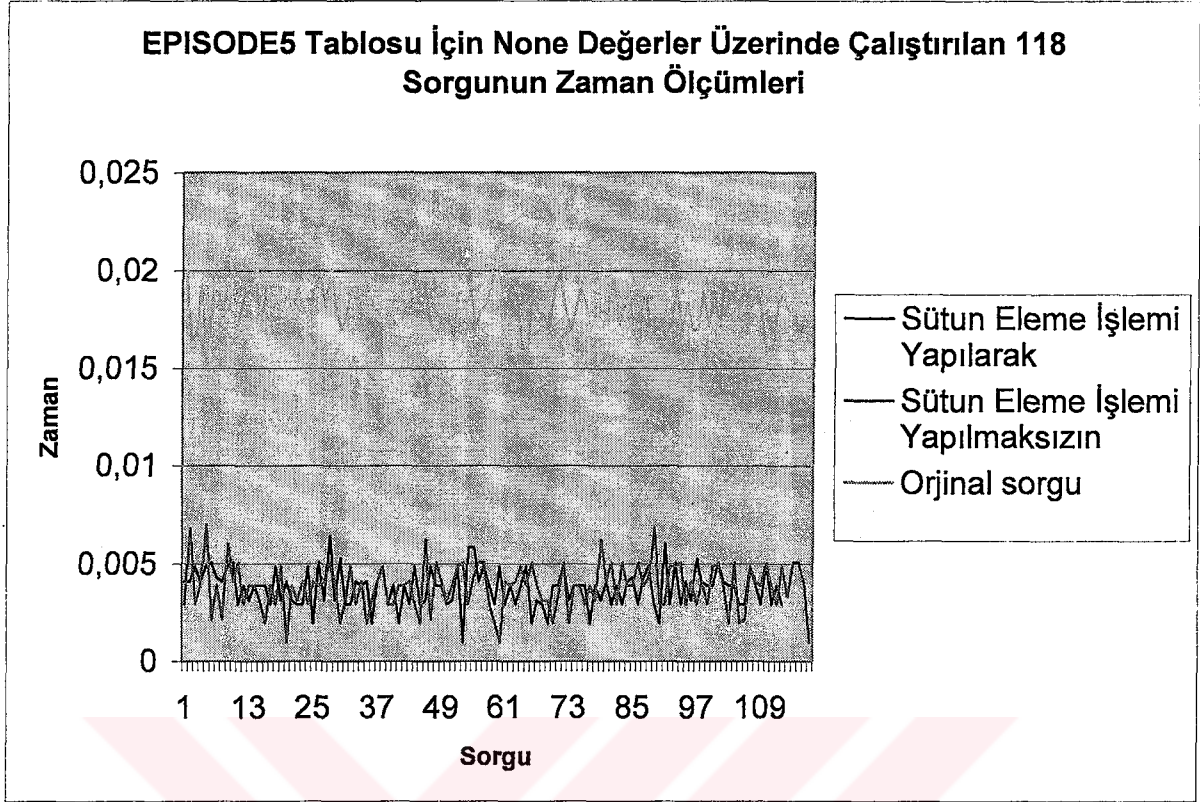
SELECT * FROM EPISODE5	ADMISSIONS=4274 and EPISODES=5146
SELECT * FROM EPISODE5	ADMISSIONS=6272 and EPISODES=6738
SELECT * FROM EPISODE5	ADMISSIONS=8348 and EPISODES=9124
SELECT * FROM EPISODE5	ADMISSIONS=8187 and EPISODES=8297
SELECT * FROM EPISODE5	ADMISSIONS=54345 and EPISODES=72426
SELECT * FROM EPISODE5	ADMISSIONS=8265 and EPISODES=9705
SELECT * FROM EPISODE5	ADMISSIONS=16257 and EPISODES=22194
SELECT * FROM EPISODE5	ADMISSIONS=3020 and EPISODES=3074
SELECT * FROM EPISODE5	ADMISSIONS=4019 and EPISODES=4137
SELECT * FROM EPISODE5	ADMISSIONS=78300 and EPISODES=102119
SELECT * FROM EPISODE5	ADMISSIONS=11431 and EPISODES=11856
SELECT * FROM EPISODE5	ADMISSIONS=5691 and EPISODES=5991
SELECT * FROM EPISODE5	ADMISSIONS=2771 and EPISODES=3315

SELECT * FROM EPISODE5	ADMISSIONS=3770 and EPISODES=3846
SELECT * FROM EPISODE5	ADMISSIONS=14929 and EPISODES=15143
SELECT * FROM EPISODE5	ADMISSIONS=29790 and EPISODES=36923
SELECT * FROM EPISODE5	ADMISSIONS=1827 and EPISODES=1854
SELECT * FROM EPISODE5	ADMISSIONS=73249 and EPISODES=73924
SELECT * FROM EPISODE5	ADMISSIONS=34638 and EPISODES=40790
SELECT * FROM EPISODE5	ADMISSIONS=7191 and EPISODES=9838
SELECT * FROM EPISODE5	ADMISSIONS=14514 and EPISODES=17225
SELECT * FROM EPISODE5	ADMISSIONS=3604 and EPISODES=3647
SELECT * FROM EPISODE5	ADMISSIONS=5110 and EPISODES=5798
SELECT * FROM EPISODE5	ADMISSIONS=44717 and EPISODES=44795
SELECT * FROM EPISODE5	ADMISSIONS=1744 and EPISODES=1799
SELECT * FROM EPISODE5	ADMISSIONS=28130 and EPISODES=28232
SELECT * FROM EPISODE5	ADMISSIONS=13933 and EPISODES=19035
SELECT * FROM EPISODE5	ADMISSIONS=9771 and EPISODES=10281
SELECT * FROM EPISODE5	ADMISSIONS=4861 and EPISODES=5006
SELECT * FROM EPISODE5	ADMISSIONS=19308 and EPISODES=26027
SELECT * FROM EPISODE5	ADMISSIONS=2356 and EPISODES=3046
SELECT * FROM EPISODE5	ADMISSIONS=3355 and EPISODES=3379
SELECT * FROM EPISODE5	ADMISSIONS=4612 and EPISODES=5140
SELECT * FROM EPISODE5	ADMISSIONS=13352 and EPISODES=13396
SELECT * FROM EPISODE5	ADMISSIONS=3272 and EPISODES=4170
SELECT * FROM EPISODE5	ADMISSIONS=4529 and EPISODES=4655
SELECT * FROM EPISODE5	ADMISSIONS=4446 and EPISODES=4457
SELECT * FROM EPISODE5	ADMISSIONS=6444 and EPISODES=7212
SELECT * FROM EPISODE5	ADMISSIONS=3189 and EPISODES=3271
SELECT * FROM EPISODE5	ADMISSIONS=12522 and EPISODES=13706
SELECT * FROM EPISODE5	ADMISSIONS=2036 and EPISODES=2240
SELECT * FROM EPISODE5	ADMISSIONS=5946 and EPISODES=6293
SELECT * FROM EPISODE5	ADMISSIONS=8027 and EPISODES=8042
SELECT * FROM EPISODE5	ADMISSIONS=7944 and EPISODES=8055
SELECT * FROM EPISODE5	ADMISSIONS=7861 and EPISODES=8199
SELECT * FROM EPISODE5	ADMISSIONS=3939 and EPISODES=4008
SELECT * FROM EPISODE5	ADMISSIONS=5697 and EPISODES=6712
SELECT * FROM EPISODE5	ADMISSIONS=1412 and EPISODES=1527
SELECT * FROM EPISODE5	ADMISSIONS=3856 and EPISODES=3888
SELECT * FROM EPISODE5	ADMISSIONS=5614 and EPISODES=5791
SELECT * FROM EPISODE5	ADMISSIONS=61484 and EPISODES=75326
SELECT * FROM EPISODE5	ADMISSIONS=2774 and EPISODES=3583
SELECT * FROM EPISODE5	ADMISSIONS=7612 and EPISODES=12965
SELECT * FROM EPISODE5	ADMISSIONS=5365 and EPISODES=12146
SELECT * FROM EPISODE5	ADMISSIONS=21822 and EPISODES=23539
SELECT * FROM EPISODE5	ADMISSIONS=10779 and EPISODES=10864
SELECT * FROM EPISODE5	ADMISSIONS=5282 and EPISODES=5294
SELECT * FROM EPISODE5	ADMISSIONS=35018 and EPISODES=35074
SELECT * FROM EPISODE5	ADMISSIONS=5199 and EPISODES=5718
SELECT * FROM EPISODE5	ADMISSIONS=5116 and EPISODES=5424
SELECT * FROM EPISODE5	ADMISSIONS=5033 and EPISODES=5055
SELECT * FROM EPISODE5	ADMISSIONS=24075 and EPISODES=33250
SELECT * FROM EPISODE5	ADMISSIONS=13945 and EPISODES=16923
SELECT * FROM EPISODE5	ADMISSIONS=23577 and EPISODES=23750
SELECT * FROM EPISODE5	ADMISSIONS=9700 and EPISODES=9726
SELECT * FROM EPISODE5	ADMISSIONS=2359 and EPISODES=2394
SELECT * FROM EPISODE5	ADMISSIONS=6782 and EPISODES=7386
SELECT * FROM EPISODE5	ADMISSIONS=222241 and EPISODES=248695
SELECT * FROM EPISODE5	ADMISSIONS=4618 and EPISODES=4653
SELECT * FROM EPISODE5	ADMISSIONS=6699 and EPISODES=6768
SELECT * FROM EPISODE5	ADMISSIONS=13447 and EPISODES=13790
SELECT * FROM EPISODE5	ADMISSIONS=2276 and EPISODES=2343
SELECT * FROM EPISODE5	ADMISSIONS=4535 and EPISODES=4543
SELECT * FROM EPISODE5	ADMISSIONS=9202 and EPISODES=9221
SELECT * FROM EPISODE5	ADMISSIONS=3275 and EPISODES=3613

SELECT * FROM EPISODE5	ADMISSIONS=1621 and EPISODES=1630
SELECT * FROM EPISODE5	ADMISSIONS=13032 and EPISODES=13719
SELECT * FROM EPISODE5	ADMISSIONS=12866 and EPISODES=13915
SELECT * FROM EPISODE5	ADMISSIONS=2193 and EPISODES=2374
SELECT * FROM EPISODE5	ADMISSIONS=70144 and EPISODES=73293
SELECT * FROM EPISODE5	ADMISSIONS=3109 and EPISODES=3225
SELECT * FROM EPISODE5	ADMISSIONS=8455 and EPISODES=8530
SELECT * FROM EPISODE5	ADMISSIONS=4120 and EPISODES=4126
SELECT * FROM EPISODE5	ADMISSIONS=33216 and EPISODES=33456
SELECT * FROM EPISODE5	ADMISSIONS=3026 and EPISODES=3316
SELECT * FROM EPISODE5	ADMISSIONS=16281 and EPISODES=16824
SELECT * FROM EPISODE5	ADMISSIONS=16115 and EPISODES=16383
SELECT * FROM EPISODE5	ADMISSIONS=1996 and EPISODES=2028
SELECT * FROM EPISODE5	ADMISSIONS=28083 and EPISODES=29023
SELECT * FROM EPISODE5	ADMISSIONS=3942 and EPISODES=3976
SELECT * FROM EPISODE5	ADMISSIONS=5869 and EPISODES=6336
SELECT * FROM EPISODE5	ADMISSIONS=3859 and EPISODES=3927
SELECT * FROM EPISODE5	ADMISSIONS=5620 and EPISODES=5643
SELECT * FROM EPISODE5	ADMISSIONS=1913 and EPISODES=2074
SELECT * FROM EPISODE5	ADMISSIONS=34851 and EPISODES=38242
SELECT * FROM EPISODE5	ADMISSIONS=3776 and EPISODES=4014
SELECT * FROM EPISODE5	ADMISSIONS=52046 and EPISODES=52478
SELECT * FROM EPISODE5	ADMISSIONS=5371 and EPISODES=5470
SELECT * FROM EPISODE5	ADMISSIONS=2611 and EPISODES=3400
SELECT * FROM EPISODE5	ADMISSIONS=3610 and EPISODES=4356
SELECT * FROM EPISODE5	ADMISSIONS=14455 and EPISODES=14813
SELECT * FROM EPISODE5	ADMISSIONS=26303 and EPISODES=28355
SELECT * FROM EPISODE5	ADMISSIONS=17979 and EPISODES=19360
SELECT * FROM EPISODE5	ADMISSIONS=238922 and EPISODES=265713
SELECT * FROM EPISODE5	ADMISSIONS=3527 and EPISODES=3795
SELECT * FROM EPISODE5	ADMISSIONS=1747 and EPISODES=2386
SELECT * FROM EPISODE5	ADMISSIONS=47229 and EPISODES=60252
SELECT * FROM EPISODE5	ADMISSIONS=25141 and EPISODES=25177
SELECT * FROM EPISODE5	ADMISSIONS=3444 and EPISODES=3548
SELECT * FROM EPISODE5	ADMISSIONS=9712 and EPISODES=11009
SELECT * FROM EPISODE5	ADMISSIONS=1206 and EPISODES=1377
SELECT * FROM EPISODE5	ADMISSIONS=9546 and EPISODES=9569
SELECT * FROM EPISODE5	ADMISSIONS=50315 and EPISODES=52578
SELECT * FROM EPISODE5	ADMISSIONS=6705 and EPISODES=7393
SELECT * FROM EPISODE5	ADMISSIONS=13293 and EPISODES=13324
SELECT * FROM EPISODE5	ADMISSIONS=9048 and EPISODES=10856
SELECT * FROM EPISODE5	ADMISSIONS=23813 and EPISODES=29482
SELECT * FROM EPISODE5	ADMISSIONS=8882 and EPISODES=10878

118 ‘er adet üç grup sorgu bulunmaktadır.

Tüm bu sorguları çalıştırıp zaman ölçümü yapılır ve ölçümün grafik ifadesi şu şekildedir



Şekil 6.3 EPISODE5 tablosu için özelliiksiz(none) değerler üzerinde çalıştırılan 118 sorgunun zamanı

Grafik kural tablosu kullanarak yapılan sorguların 118 sorguda da zaman açısından daha karlı olduğunu göstermektedir. Sorgular özelliiksiz(none) değerler üzerindedir ve özelliiksiz(none) değerler üzerinde sorgu çalıştırmak dizin(index) ve anahtar(key) değerler üzerinde çalıştırmaktan daha yavaştır. Kural tablosundan elde edilen kuralların dizin(index) olma şartını sağlaması yani sağ tarafı dizin(index) olma özelliğine sahip olan kuralların seçilerek kullanılması çalıştırılan her sorgu için zaman tasarrufu sağlamıştır.

Çalıştırılan 118 sorgunun zamanlarının ortalamasını alıp zaman kazancını yüzde olarak ifade edecek olur isek:

Çizelge 6.1 Eliminasyonlu, eliminasyonsuz ve orijinal metotların yüzde olarak kıyaslanması

Eliminasyonlu-orijinal sorgu	Eliminasyonsuz-orijinal sorgu	Eliminasyonlu-eliminasyonsuz
0,789675653	0,789036092	0,00303161

Eliminasyon yöntemiyle yapılan sorgunun orijinal sorguya göre yüzde yetmiş sekiz zaman kazandırdığını görüyoruz. Eliminasyonlu ve eliminasyonsuz zamanlar arasında çok büyük bir fark yoktur ancak eliminasyon yönteminin öğrenmeye sağladığı katkıyı ve kuralları kural setinden bulma zamanına katkısını düşünecek olur isek zaman açısından en karlı yöntem eliminasyon metoduyla yapılan yöntemdir sonucuna varılabilir.

Sorgular ve kural tablosunun sorguyu sağlayan sağ tarafı için belli kriterler belirleyip bu kriterlere göre kural kalitesi test edilebilir. Böylece metodun en verimli olduğu ya da verimsiz olduğu noktaları saptamak mümkündür.

Örneğin

Sorgunun anahtar(key)=x

Sorguyu sağlayan kural tablosunun sağ tarafının dizin(index)=y

Bu durumda sol anahtar(key)=x olma özelliğine ve sağ dizin(index)=y olma özelliğine sahip olmalıdır. Bu durumu sağlayan sorgular çalıştırılır ve zaman ölçümü yapılır. Böylece sorgunun hangi durumları için kural tablosunun hangi değerlerini kullanırsak verimi maksimum yaparız sorusunu cevaplamış oluruz.

6.3 Uygulama Sonuçları(2)

Orijinal tablo:episode5

Tablo sütun sayısı:11

Satır sayısı:41046

Tablo anahtar değeri:ID

Tablo dizin(index) değeri:EPISODES

Kural tablosu1:eliruleepisode5

Sütun elemesinden sonra oluşturulan kurallardan meydana gelen tablo

Tablo sütun sayısı:13

Toplam kural sayısı(satır sayısı):2681

Kural tablosu2:esizruleepisode5

Sütun elemesi yapılmadan oluşturulan kurallardan meydana gelen tablo

Tablo sütun sayısı:13

Toplam kural sayısı(satır sayısı):8507

Anahtar(key),dizin(index) ve özelliiksiz(none) sütunlardan oluşan kural tablosu için belirlenen kriterler aşağıdaki gibidir

Çizelge 6.2 Kural tablosu için belirlenen kriterler

Sol Taraf	Sağ Taraf
anahtar(key)=	dizin(index)=
anahtar(key)=	özelliiksiz(none)=
anahtar(key)<	dizin(index)=
anahtar(key)<	özelliiksiz(none)=
anahtar(key)<	dizin(index)>=
anahtar(key)>	özelliiksiz(none)>=
dizin(index)=	anahtar(key)=
dizin(index)=	özelliiksiz(none)=
dizin(index)<	anahtar(key)=
dizin(index)<	özelliiksiz(none)=
dizin(index)=	özelliiksiz(none)>=
dizin(index)<	özelliiksiz(none)>=
özelliiksiz(none)=	anahtar(key)=
özelliiksiz(none)=	dizin(index)=
özelliiksiz(none)>	anahtar(key)=
özelliiksiz(none)>	dizin(index)=
özelliiksiz(none)=	anahtar(key)>=
özelliiksiz(none)=	dizin(index)>=
özelliiksiz(none)>	dizin(index)>=
özelliiksiz(none)>	anahtar(key)>=

Bu kriterleri sağlayan sorgular orijinal tablo,eliruleepisode5 ve esizruleepisode5 üzerinde çalıştırılır. Her sorgu zaman ölçümündeki hata payını engellemek amacıyla onar kere çalıştırılır.Ölçülen toplam zamanların ortalaması alınır.

Örneğin sol özelliiksiz(none)= ve sağ dizin(index)= durumunu sağlayan 118 sorgu vardır.Bu 118 sorgu kendi içinde onar kere çalıştırılır ve elimizde ortalaması alınmış 118 tane zaman olur. 118 sorgu orijinal tablo,eliruleepisode5 ve esizruleepisode5 üzerinde çalıştırılır.Sorguların çalışmasının ardından kıyaslama yapmak için 118*3 tane zaman olur.

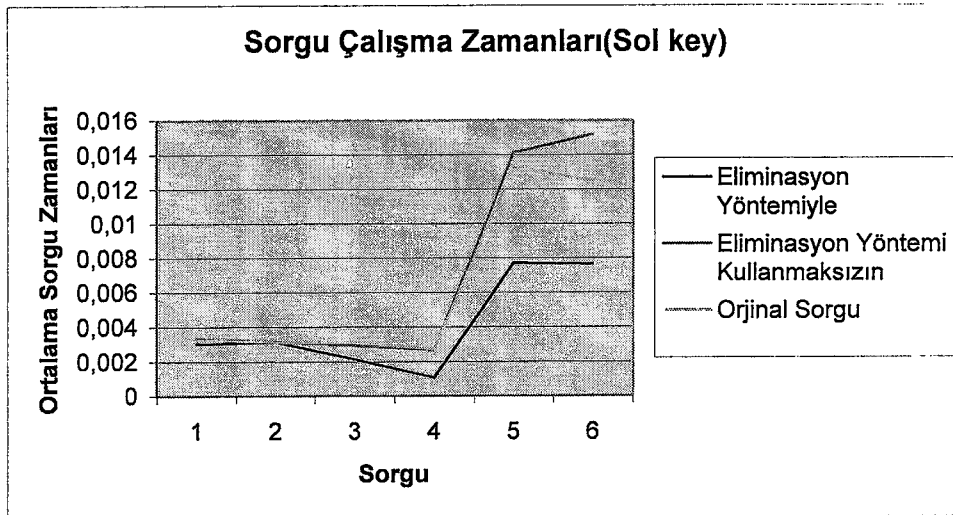
Her kriteri sağlayan sorgu sayıları eşit olmadığından durumu daha açıkça ifade etmek için bulunan zamanların ortalamasını alır isek sonuçlar aşağıdaki gibi olmaktadır.

Çizelge 6.3 Çizelge 6.2'deki kriterler için alınan zaman sonuçları

		Eliminasyon Yöntemiyle	Eliminasyon Yöntemi Kullanmaksızın	Orjinal Sorgu
anahtar(key)=	dizin(index)=	0,003018725	0,003321	0,003283
anahtar(key)=	özelliiksiz(none)=	0,003148437	0,003163297	0,003134574
anahtar(key)<	dizin(index)=	0,002148438	0,002929688	0,003125
anahtar(key)<	özelliiksiz(none)=	0,001074219	0,002692522	0,002734375
anahtar(key)<	dizin(index)>=	0,007724	0,014132	0,013299
anahtar(key)>	özelliiksiz(none)>=	0,007658203	0,015191592	0,01250186
dizin(index)=	anahtar(key)=	0,003055245	0,00320173	0,00327846
dizin(index)=	özelliiksiz(none)=	0,025513174	0,025901884	0,024291514
dizin(index)<	anahtar(key)=	0,001953125	0,002050781	0,004882813
dizin(index)<	özelliiksiz(none)=	0,002636719	0,004003906	0,004492187
dizin(index)=	özelliiksiz(none)>=	0,025172335	0,02689951	0,024820006
dizin(index)<	özelliiksiz(none)>=	0,25661765	0,277971814	0,256778492
özelliiksiz(none)=	anahtar(key)=	0,003094	0,003664873	0,017021669
özelliiksiz(none)=	dizin(index)=	0,003712	0,003975	0,018035
özelliiksiz(none)>	anahtar(key)=	0,004003906	0,004101562	0,016992187
özelliiksiz(none)>	dizin(index)=	0,00390625	0,004980469	0,018066406
özelliiksiz(none)=	anahtar(key)>=	0,016210938	0,018750001	0,019140625
özelliiksiz(none)=	dizin(index)>=	0,018847656	0,032128908	0,018945312
özelliiksiz(none)>	dizin(index)>=	0,081416015	0,145732419	0,141210941
özelliiksiz(none)>	anahtar(key)>=	0,140507814	0,142148438	0,143789064

Tabloyu grafiklerle ifade edecek olur isek

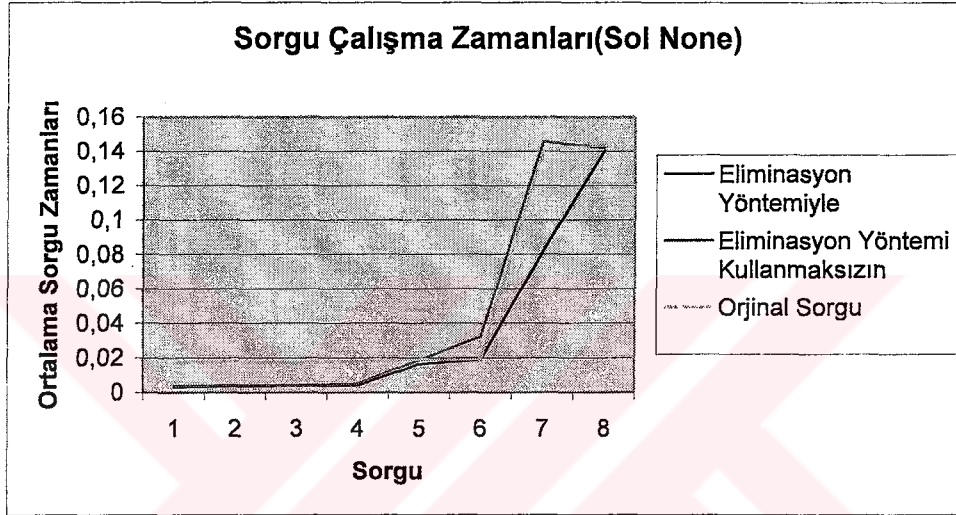
Sol Tarafın anahtar(key)= olma özelliğini sağladığı durumlardaki grafik



Şekil 6.4 Kural tablosunun sol tarafının anahtar(key) olma durumunda çalışma zamanları

Zaman Kazancındaki verimin en yüksek olduğu noktalar 4,5 ve 6'ncı noktalardır.Tablodan bu noktalara karşılık gelen kriterlere bakar isek bu noktaların sol taraftaki anahtar(key)> durumuna denk geldiğini görürüz.Anahtar(key)= durumunda ise çizgiler çakışma göstermektedir. Bunun sebebi sorgular anahtar değerler üzerinde ve eşitlik durumunda hızlı çalışmaktadırlar. Kural setini kullanmak bu durumlarda çok da anlamlı değildir. Çizgilerdeki çakışma normaldir.

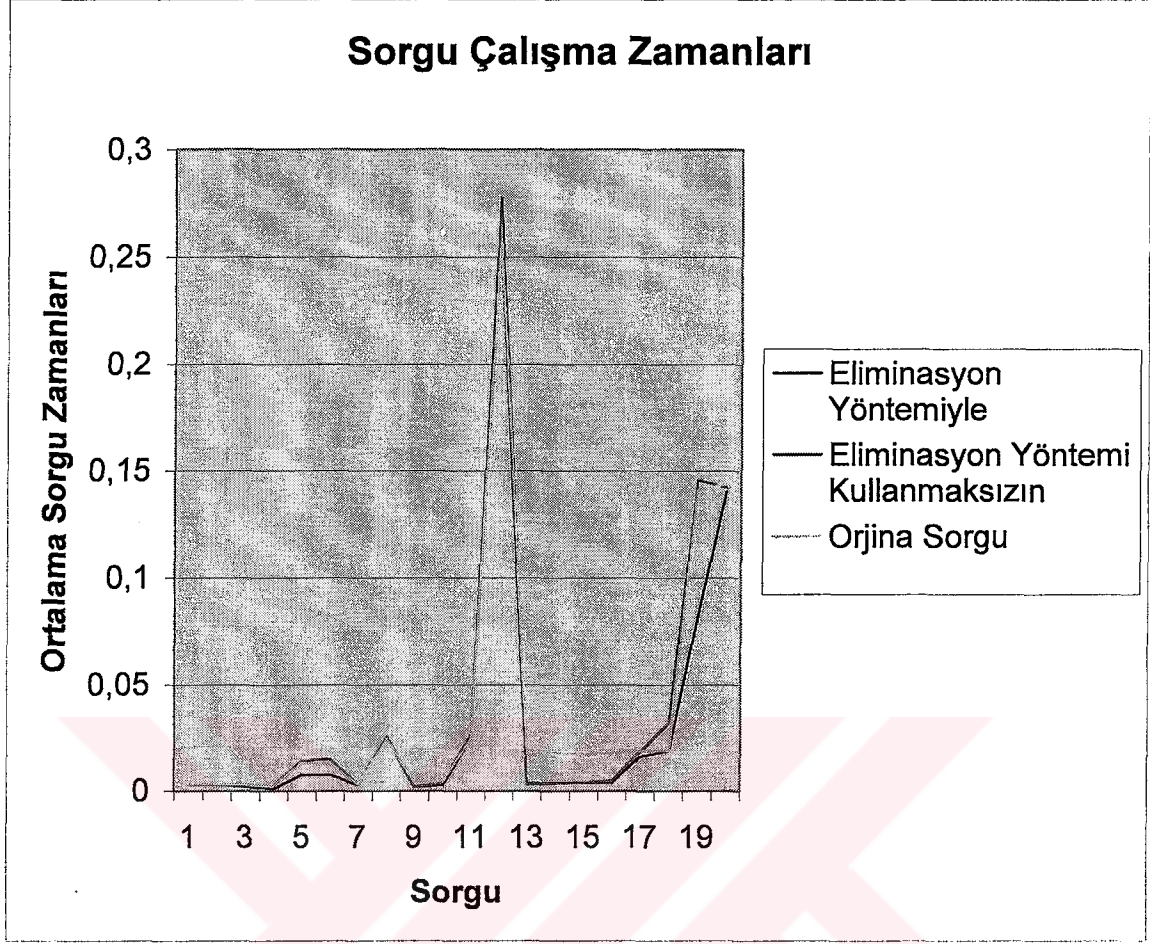
Sol tarafın özelliiksiz(none) olma özelliğini sağladığı durumlardaki grafik



Şekil 6.5 Kural tablosunun sol tarafının anahtar(key) olma durumunda çalışma zamanları

Zaman kazancındaki verimin en yüksek olduğu noktalar 1,2,3,4,7 ve 8'inci noktalardır.Genelde sol taraf özelliiksiz(none) olduğu durumlarda her noktada zaman kazancı mevcuttur.Çünkü Özelliiksiz(none) değerler üzerinde sorgunun çalışması yavaştır.1 ve 2'inci noktalar sol özelliiksiz(none)= ve sağ anahtar(key)=, dizin(index)= kriterini sağlayan sorgulardır. Her iki tarafta da eşitlik olmasına rağmen sağ tarafın anahtar özelliğine sahip olması sorguyu hızlandırır. Kural seti yardımıyla oluşturulan sorgu orijinal sorguya oranla daha hızlı çalışır.Verimin düşük olduğu noktalar sol özelliiksiz(none)= ve sağ anahtar(key)>=,dizin(index)>= kriterini sağlayan sorgulardır.Sorgu eşitlik durumunda daha hızlı çalışır.Sağ tarafın büyük eşittir olma özelliği sorguyu yavaşlatan bir unsurdur.Verimin düşük olması normaldir.

Tüm tabloyu tek bir grafikte gösterecek olur isek



Şekil 6.6 Çizelge 6.3'ün grafikte ifadesi

Grafikte zaman kazancının maksimum olduğu noktalar 13,14,15,16'ncı noktalardır. Bu noktaların tablo değerlerine bakar isek sol tarafı özelliksiz(none) olma özelliğine ve sağ tarafı anahtar(key)=,dizin(index)= olma özelliğine sahip noktalardır. Bu durumda uygulanan metodun en verimli özelliksiz(none) değerler üzerinde sorgu çalıştırırken ve kural tablosunda anahtar(key)=,dizin(index)= olma özelliğini sağladığı noktalarda çalıştığını söyleyebiliriz.

Metotların birbirine yüzde olarak kazançları da şu şekilde gösterilebilir.

Çizelge 6.4 Metotların birbiriyle yüzde olarak kıyaslanması

Eliminasyonlu-orijinal sorgu	Eliminasyonsuz-orijinal sorgu	Eliminasyonlu-eliminasyonsuz
0,329233316	0,130496464	0,198336

7 DİĞER UYGULAMALAR

Bu bölümde istatistik metotlar, korelasyon ve regresyon analizi kullanılarak veritabanı üzerinde yapılan analizler özetlenmektedir. Regresyon ve korelasyon analizi ile veritabanındaki null değerler için tahminler yapılmıştır. Ekonomik veriler esas alınarak bu veriler incelenmiş ve otomatik sonuç alan bir sistem geliştirilmiştir.

7.1 Çoklu Regresyon Analizini Kullanarak Kayıp Data Tahmin Yöntemi

Regresyon'un en az iki değişken arasındaki ilişkinin denklem ile ifadesi olduğunu biliyoruz. Eğer değişkenler arasındaki ilişki denklem ile ifade edilebilirse, böylece bilinen değişkenler yardımıyla bilinmeyen değişken değerlerini tahmin etmek mümkündür. Veritabanları içinde büyük miktarda datayı barındıran sistemlerdir, ve zaman içinde bu dataların zarar görmesi, kaybolması yada erişimi mümkün olmayan duruma gelmesi mümkündür. Zarar görmüş, kaybolmuş datayı tekrar geri kazanmanın mümkün olmadığı durumlarda datayı istatistik metotlarla tahmin etmek kullanıcıya, datayı analiz eden kişiye yardımcı bir faktör olabilir. Yapılması gereken en küçük kareler metodunu veritabanı üzerinde dinamik olarak çalıştırmak, ve regresyon katsayılarını tahmin etmektir. Tespit edilen regresyon katsayıları ulaşmak istenen data için fikir verebilir.

7.1.1 Çoklu Regresyon Analizi ile Kayıp Data Tahmin Yöntemini Kullanan Dinamik Sistem

Tüm uygulamalar için kullanılan:

Program Geliştirme Ortamı: Visual Basic 6.0

Veritabanı: SQL Server 8.0

Veri Erişim Yöntemi: ADO

Bu uygulama için kullanılan:

Veri Tablosu: insan2

Veri Sayısı: 1023

Sütun Sayısı: 15

Tablo Anahtar Değeri: yok

Tablo dizin(index) değeri: yok

ÇALIŞMAK İSTEDİĞİN DATASETİ SEC

insan1
insan2
ruletest3
rulemek

ÇALIŞMAK İSTEDİĞİN KOLOMU SEC

sirano
kilo
boy
yas

SEÇTİĞİNİZ DATASET VE BAĞIMLI DEĞİŞKEN İÇİN HESAPLANMIŞ REGRESYON KATSAYILARI

217,044873076316
0,960955471425851
0,101869830319397
1,05665359458776

KAYIP DATA BUL

Sütun	kilo	Gerçek Değer	
Satır	1	Tahmini Değer	56,13075401695
Kayıp Datayı Tahmin et		Hata Payı	
		Süre	2,624023

Şekil 7.1 FMD programı kayıp data tahmin ekranı

Programın ilk aşaması kullanıcının üzerinde çalışmak istediği veri tablosunu ve bu veri tablosunun içinden hangi sütun üzerinde tahmin yapmak istiyorsa belirlenmesidir. Bu belirlenmenin ardından kullanıcı devreden çıkar ve program seçilen tablo içinden verileri okuyarak seçilen sütun değeri için çoklu regresyon katsayılarını oluşturur.

Artık belirlenen katsayılar yardımıyla tablonun belirlenen sütunu için hangi satır değeri için tahmin yapmak istiyor ise programa hesaplatmamız mümkün olmaktadır.

Yukarıda ki uygulamada 1000 datalık insan2 tablosu üzerinde, kilo değeri için regresyon katsayıları (denklemleri) hesaplanmıştır. Kilo değerinin birinci satır değeri için programın yaptığı tahmin 56,130 olup aynı satırın tablo orijinal değeri 56'dır. Kayıp data tahmin süresi 2,6 sn'dir. (İnsan2 tablosunun 30 kayıtlık kısmı Ek 4'de mevcuttur)

7.2 Çoklu Regresyon Analizi ile 1981-2002 Yılları Arasındaki Sigorta Primlerindeki Değişimlerin İncelenmesi

Çoklu regresyon analizini uygularken kullanılan değerler birbiriyle ne kadar ilişkili olursa, ilişkili ola durumuna korelasyon diyoruz bulacağımız tahmini değerlerin gerçeğe yakın olma olasılığı o kadar artar. Çünkü bağımsız değişken olan x'ler y'nin açıklayıcısı olmalı ki, x değerlerini kullanarak y değerlerine ilişkin doğru tahmin yapabilelim.

Aşağıdaki tabloda 1981-2002 yılları arasındaki sigorta primlerindeki değişimi görüyoruz. Tüm değerler www.hazine.gov.tr adresinden alınmıştır.

Çizelge 7.1 1981-2002 yılları arasındaki sigorta primleri(dolar) değişimi

Yıl	Üretilen Toplam Prim (Milyon\$)	Prim Artışı%	Kişi Başına Düşen Prim (Milyon\$)	Kişi Başına Düşen Prim Artışı%	Nüfus(Milyon)
1981	203,0000	0,0000	4,4615	0,0000	45,5000
1982	206,0000	1,4778	4,4111	-1,1297	46,7000
1983	203,0000	-1,4563	4,2380	-3,9250	47,9000
1984	231,0000	13,7931	4,7047	11,0120	49,1000
1985	247,0000	6,9264	4,9105	4,3755	50,3000
1986	281,0000	13,7652	5,4669	11,3305	51,4000
1987	363,0000	29,1815	6,9011	26,2344	52,6000
1988	401,0000	10,4683	7,4674	8,2055	53,7000
1989	497,0000	23,9401	9,0528	21,2311	54,9000
1990	710,0000	42,8571	12,6560	39,8014	56,1000
1991	964,4829	35,8427	16,8322	32,9978	57,3000
1992	1187,2565	23,0977	20,2604	20,3669	58,6000
1993	1562,8567	31,6360	26,0476	28,5645	60,0000
1994	1066,0467	-31,7886	17,3623	-33,3439	61,4000
1995	1377,4683	29,2128	21,8646	25,9312	63,0000
1996	1534,8893	11,4283	24,3000	11,2797	64,6000
1997	1811,0654	17,9932	29,5563	20,1141	62,6000
1998	2119,1901	17,0134	32,7106	15,1332	64,7860
1999	2314,4244	8,5556	35,9466	6,9809	64,3850
2000	2846,5256	22,9906	43,5842	21,2468	65,3110
2001	2032,6227	-28,5928	30,6908	-29,5826	66,2290
2002	2426,3825	19,6124	36,1392	17,9894	67,1400

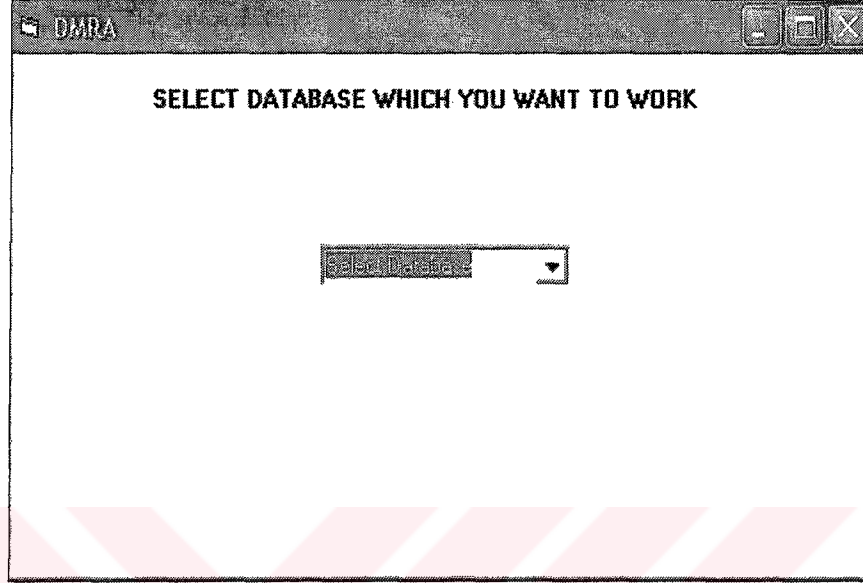
Veri analizi için kullanılan

Program Geliştirme Ortamı: Visual Basic 6.0

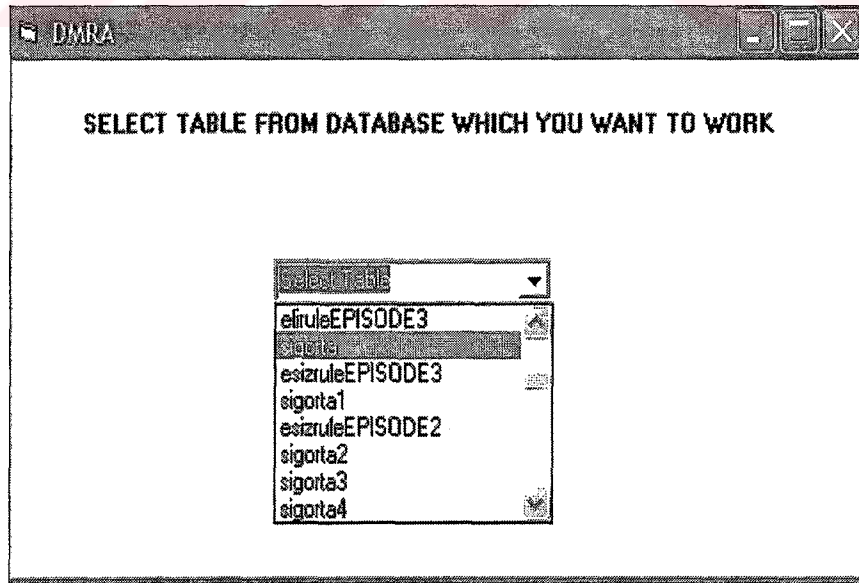
Veritabanı: SQL Server 8.0

Veri Erişim Yöntemi:ADO

Program Ara yüzleri şekil 7.2, 7.3 ve 7.4’de gösterilmiştir.



Şekil 7.2 DMRA programı,veritabanı seçim ekranı



Şekil 7.3 DMRA programı tablo seçim ekranı

DMRA

Insert Column:

Execute

Yıl	dolanurehm	primatidolar	kisbesatindolar	kisbesatit	ruhus
1992	1187,26854150545	23,0977276882421	20,260305376357	98,098953939843	58,6
1993	1962,8568086936	31,6358712527157	28,047611347686	109,6254924491	60
1994	1066,0467296133	-31,7885903177347	17,362245965388	80,2271325931572	61,4
1995	1377,48834612938	23,2127547391584	21,8645769226897	94,2797189370856	63
1996	1534,68931380835	11,4282620451981	24,3	97,6164321272718	64,6
1997	1811,06538918576	17,3932238040124	29,5852662570966	126,514359794868	62,6
1998	2118,19007351286	17,0134492761966	32,7106176268201	91,0753103233133	64,78
1999	2314,42441276806	8,55561037151016	35,946595435803	72,0872025630567	64,38
2000	2846,52561726454	22,5806494526508	43,584168322038	81,0130425780734	65,31
2001	2032,52268734738	-28,5928476406321	30,6908265601445	37,8958067905473	66,22
2002	2426,38254617683	19,6124234440381	36,1391502260475	45,1660096838315	67,14

Chose Dependent Variable:

Calculate Regression Coefficients:

1948,1397389523
6,42015786718275E-03
-9,19267871267946E-04
-0,31683516874562
-5,14176705200953E-03

Time= 0,1298828 Sn

R2= 0,991352332167324
R= 0,995667078981385

Time= 0,1591797 Sn

F Oran= 365,667841629499
MSR= 175,55680427847
MSE= 0,478581180364689

Time= 0,1601563 Sn

Back

Şekil 7.4 DMRA programı, regresyon analizi yapan ekran

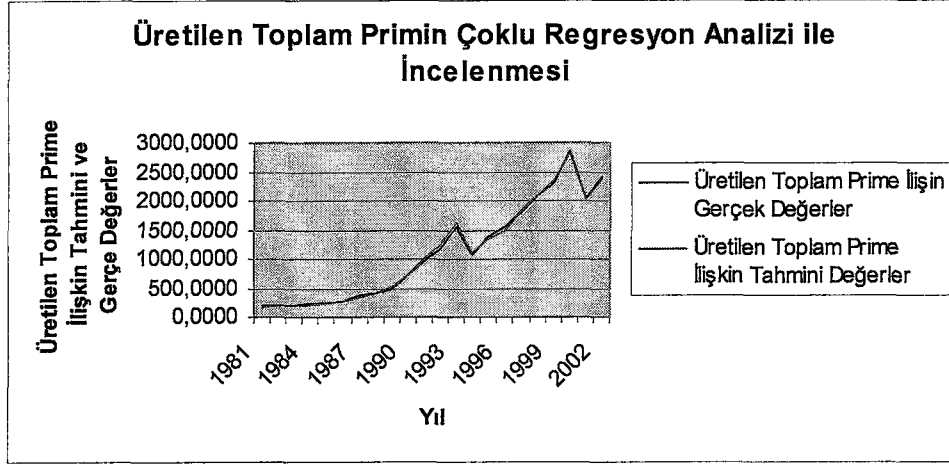
Kullanılan programın ilk bölümü kullanıcı isteğine göre veritabanı ve veri seti seçimi. Veri seti seçiminin ardından karşımıza çıkan ekran yukarıda da görüldüğü üzere üç ana bölümden oluşuyor. Birinci bölüm seçim bölümü, seçim işleminde hangi sütunlar üzerinde analiz yapmak istiyor ise belirlenir ve ikinci bölümde analiz edilecek veriler program çalıştırılarak ekranda görüntülenir. Üçüncü bölüm analiz bölümüdür. Analiz bölümü yine kendi içinde 3 ana parçadan oluşmuştur. Bunlardan ilki regresyon katsayılarının hesaplandığı bölümdür ki, analiz işlemi bu katsayıların oluşturduğu denklem yardımıyla yapılır. İkinci bölüm belirginlik katsayısının hesaplandığı bölümdür. Belirginlik katsayısı bize bağımlı değişkenin diğer tüm değişkenlerle ilişkisini gösteren katsayıdır. Belirginlik katsayısı değişkenin diğer değişkenlerle bağımlılığıyla ilgili fikir verir. Üçüncü bölüm regresyonun anlamlılığının sınıandığı bölümdür. Burada hesaplanan F oran, MSR ve MSE analizin anlamlılığının sınıanmasında kullanılan değerlerdir.

Programın Çalıştırılmasının Ardından Elde Edilen Tahmini Değerler:

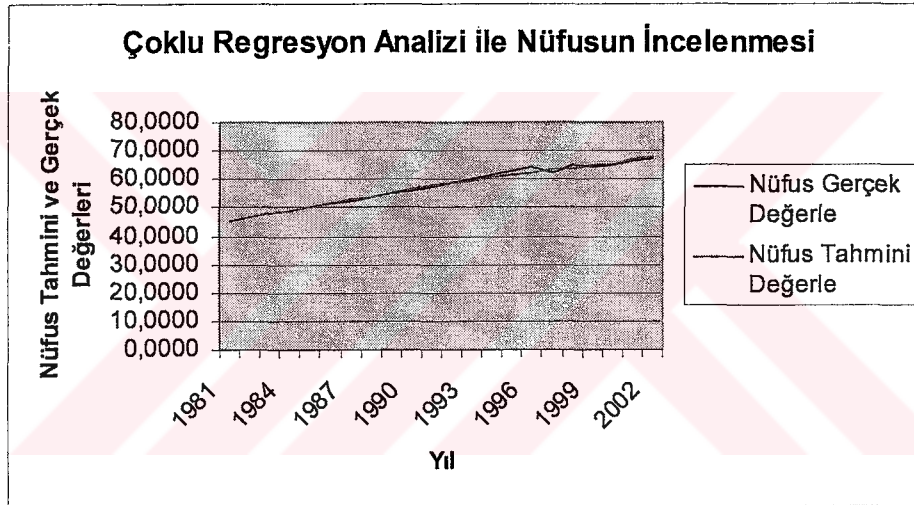
Çizelge 7.2 Çizelge 7.1 için hesaplanan tahmini değerler

Tahmini Değerler					
Yıl	Üretilen Toplam Prim (Milyon\$)	Prim Artışı%	Kişi Başına Düşen Prim (Milyon\$)	Kişi Başına Düşen Prim Artışı%	Nüfus(Milyon)
1981	177,1583762	2,037146438	4,77380691	-1,9194138	45,16530218
1982	200,2992191	0,956215538	4,42111921	-0,5816183	46,75822433
1983	194,1542205	-1,81705516	4,32891054	-3,5306235	47,92220185
1984	214,3282395	13,55708314	4,95439519	11,2188374	49,18151942
1985	236,1169194	6,75349147	5,09427768	4,54682646	50,38197786
1986	269,0217468	13,8692573	5,70230502	11,2033927	51,59107699
1987	353,6872025	29,0778195	7,11519741	26,2526359	52,9169447
1988	409,2084317	10,47205148	7,42285165	8,19964181	54,07299969
1989	505,1376627	23,77479428	9,02335692	21,3390496	55,33512136
1990	725,2015532	42,55859268	12,5025419	39,9841517	56,58953009
1991	1004,924466	35,1976376	16,2392583	33,5755712	57,74404799
1992	1241,202752	22,05134505	19,408123	21,4020815	58,78583148
1993	1610,088051	30,274018	25,2159816	29,8911472	59,676873
1994	1099,453249	-32,38754036	16,8695415	-32,571299	60,70664845
1995	1337,990051	28,7811762	22,4685858	26,2759157	61,52354299
1996	1480,713227	13,68033532	25,1140492	9,10865736	62,18923318
1997	1859,476419	21,04429149	28,8728481	17,1961099	63,47288533
1998	2086,96658	16,98528626	33,1352888	15,1405816	63,938394
1999	2337,876001	7,852082481	35,5332074	7,69633805	65,12646578
2000	2828,19971	22,43094639	43,7073851	21,7597118	65,42213638
2001	2048,591398	-29,12421565	30,5299406	-28,949198	67,33726011
2002	2365,415832	19,92988844	37,1318854	17,5750195	67,71278262

Bulunan tahmini değerleri grafikler yardımıyla gözlenen değerlerle kıyaslayacak olursak



Şekil 7.5 Üretilen toplam primin gözlenen değerleriyle tahmini değerleri



Şekil 7.6 Nüfus değerlerinin gözlenen değerleri ve tahmini değerleri

Grafiklerden de görüldüğü üzere bulunan tahmini değerler gerçek değerlere oldukça yakındır, bunun sebebi veri seti içinde birbiriyle yüksek korelasyonu olan değerler vardır. Bu yüksek korelasyon gerçek ve tahmini değer eğrilerinin neredeyse birbirleriyle çakışmasına yol açmıştır.

7.3 1981-2002 Yılları Arasındaki Sigorta Primlerindeki Değişimleri Kullanarak Yapılan Trend Analizi

Trend denklemi $Y=a+bx$ olup burada ki a ve değerlerinin tahmini regresyon analizinde olduğu gibi en küçük kareler yöntemiyle hesaplanır. Aslında trend denklemine tek değişkenli

regresyon denklemi demek mümkündür. Regresyon analizinde bilinen değişkenler yardımıyla bilinmeyeni hesaplayabiliyordur. Burada bilinen değişkenin yıl olduğunu kabul edersek gözlenen değerlere göre oluşturulmuş denklemde yıl değerini x yerine koyduğumuzda o yıl için beklenen tahmini değer hesaplanabilir. Dolayısıyla trend analizi için regresyon analizinde kullanılan sistemi ve ara yüzleri kullanmak mümkündür.

Aşağıdaki tabloda 1981-2001 yılı arasında gözlenen değerleri görülmektedir.

Aşağıdaki tabloda görülen değerler yıllara göre üretilen toplam sigorta primleridir.(milyon dolar cinsinden)

2002 yılı gerçek değerinin (2426,3825 \$) bilmediğimizi farz edip trend denlemini oluşturalım ve bu değere ne kadar yaklaşabiliyoruz görelim.

Çizelge 7.3 1981-2001 yılları arasında üretilen toplam primin gözlenen değerleri

Yıl	Üretilen Toplam Prim (Milyon\$)
1981	203,0000
1982	206,0000
1983	203,0000
1984	231,0000
1985	247,0000
1986	281,0000
1987	363,0000
1988	401,0000
1989	497,0000
1990	710,0000
1991	964,4829
1992	1187,2565
1993	1562,8567
1994	1066,0467
1995	1377,4683
1996	1534,8893
1997	1811,0654
1998	2119,1901
1999	2314,4244
2000	2846,5256
2001	2032,6227

Bağımlı değişken y'nin yıl olduğunu ve bağımsız değişken x'in toplam üretilen prim olduğunu varsayar isek en küçük kareler metodu yardımıyla oluşan tren denklemi

$$Y_t = -248428,7678 + 125,3059x$$

$$Y_{2002} = 2433,5467 \$$$

$$\text{Üretilen toplam prim için gerçek değer} = 2426,3825 \$$$

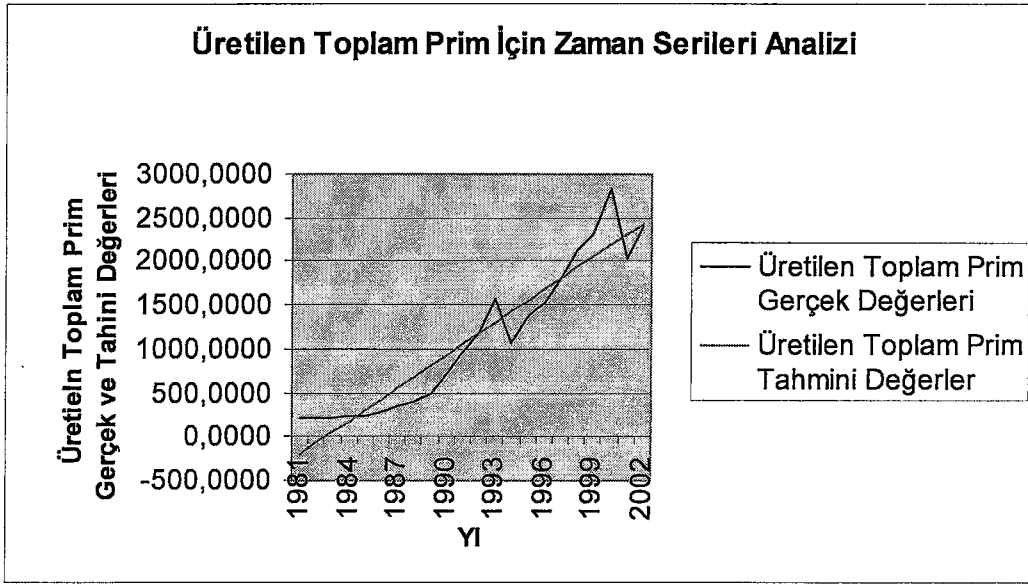
$$\text{Üretilen toplam prim için tahmini değer} = 2433,5467 \$$$

$$\text{Hata Payı: } 7,1641 \$$$

Çizelge 7.4 Çizelge 6.3'ün gözlenen değerleri için tahmini değerler

	Tahmini Değerler
Yıl	Üretilen Toplam Prim (Milyon\$)
1981	-197,8762
1982	-72,5703
1983	52,7355
1984	178,0414
1985	303,3472
1986	428,6531
1987	553,9589
1988	679,2648
1989	804,5706
1990	929,8765
1991	1055,1823
1992	1180,4882
1993	1305,7940
1994	1431,0999
1995	1556,4057
1996	1681,7116
1997	1807,0174
1998	1932,3233
1999	2057,6291
2000	2182,9350
2001	2308,2408
2002	2433,5467

Trend grafiği ise



Şekil 7.7 Üretilen toplam prim gerçek ve tahmini değerleri

Grafikten de görüldüğü üzere 1981-2002 yılları arasında gerçekleşen dolar bazında yıllık prim üretimi yükselen bir trende sahiptir. Bulunan tahmini değerler gerçek değerlere bir hayli yakın olup bazı noktalarda çakışmalar mevcuttur. Eğrilerin birbirine yakınlığı tahmini değerlerin gerçek değerlerle olan ilişkisini gösterir. Buna dayanarak bundan on yıl sonra üretilecek primler için bile tahmin yapmamız söz konusu olabilir.

8 SONUÇLAR

Bu çalışmada ulaşılan hedefleri aşağıdaki gibi sıralayabiliriz

- i. Visual Basic programlama ortamından SQL Server veritabanına bağlantı sağlanması.
- ii. Visual Basic programla ortamından gönderilen bir sorgunun cevap olarak ifade edilmesi.
- iii. Program her çalıştırıldığında SQL Server veri tabloların son halinin alınıp sistemin dinamikliğin korumasının sağlanması.
- iv. SQL Server veritabanından bilgilerin son güncel halinin alınıp istatistik metotlar yardımıyla analizi.
- v. İstatistik metotların parametrelerinin veritabanından otomatik okunmasıyla sistemin dinamikliğinin sağlanması.
- vi. Uygulanan metotların boyut(satır,sütun sayısı) ayırt etmeksizin her türlü veri tablosu üzerinde uygulanabiliyor olmasının sağlanması.
- vii. Analiz edilen bilgilerin veritabanı için anlam ifade edecek şekilde derlenmesi.
- viii. İstatistik metotlar yardımıyla elde edilen bilgilerin yardımıyla veri tabloların eliminasyonunun yapılması.
- ix. Eliminasyonu yapılan veritabanında öğrenme tekniğine göre veritabanı üzerinde otomatik olarak sorgular çalıştırılarak kuralların öğrenilmesi.
- x. Veritabanında eliminasyon metodu kullanmaksızın öğrenme tekniğine göre veritabanı üzerinde otomatik olarak sorgular çalıştırılarak kuralların öğrenilmesi.
- xi. Eliminasyon metodu kullanarak ve kullanmaksızın oluşturulan sorguların öğrenme zamanlarının ve sorgu kalitelerinin zaman hesaplaması yaparak kıyaslanması, analiz edilmesi.
- xii. Bulunan zaman değerlerinin grafikler kullanılarak ifade edilmesi.
- xiii. Veritabanlarındaki veriye ulaşılarak veriden anlamlı bilgi çıkarılması.
- xiv. İstatistik metotlar yardımıyla kaybolmuş,zarar görmüş verinin tahmin edilmesi.
- xv. Tahmin değerlerin anlamlılığının test edilmesi.

KAYNAKLAR

Armutlu İ.H. (2000), İşletmelerde Uygulamalı İstatistik

Codd E.F. (1970), A. relational model of data for large shared data banks, Communication of the ACM, 13:377-387.

Çil B. (2002), İstatistik

Çubukçu F. (1998), Visual Basic Başvuru Kılavuzu

Elmasri R. ve Navathe S.B. (1994), Fundamentals of Database Systems, The Benjamin/Cummings Publishing Company, 1994

Gökçe B.(2003), Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi

Hamburg M. (1974) , Statistical Analysis For Decision Making

Hsu C. ve Knoblock C.A. (1993), Learning database abstractions for query reformulation Knowledge Discovery in Database Workshop, 276-290.

Korth H.F. ve Silberschartz A.(1991), Database System Concepts. McGraw-Hill Book Company.

Lowden B.G.T. ve Lim K.Y. (1995), A data-driven semantic optimizer. In Proceeding of ISCISX Konferans Notları, Türkiye.

Piatetsky G. ve Shapiro (1991), Discovery, analysis and presentation of strong rules. Knowledge Discovery in Database, 229-248.

Sayli A. ve Lowden B.G.T. (1996), The use of statistics in semantic query optimization. Thirteenth European Meeting on Cybernetics and Systems Research, Vienna, 991-996

Sayli A. ve Lowden B.G.T. (1997), A fast transformation method to semantic query optimization, International Database Engineering and Application Symposium, Canada.

Shapiro J.R. (2001), SQL Server 200 The Complete Reference

Siegel M.D. (1988), Automatic rule derivation for semantic query optimization, 2. International Conference on Expert Database Systems, 669-698.

EKLER

- Ek 1 EPISODE5Tablo veri bilgileri (İlk 100 Kayıtlık)
- Ek 2 EPISODE5 Tablosu için eliminasyon metoduyla öğrenilen kurallar (10 sorgu için)
- Ek 3 EPISODE5 Tablosu için eliminasyonsuz öğrenilen kurallar (10 sorgu için)
- Ek 4 İnsan2 Tablo veri bilgileri (İlk 30 kayıt)



Ek i EPISODE5 Tablo veri bilgileri (100 Kayıtlık)

Çizelge Ek 1.1 EPISODE5 Tablo veri bilgileri (100 Kayıtlık)

ID	EPISODES	ADMISSIONS	MALE	EMERGENCY	WAITING_LIST	AGE_0_14	AGE_15_59	AGE_60_74	AGE_75	DAY_CASE
1	2	1	0	0	2	0	2	0	0	2
2	6	5	1	6	1	1	3	2	0	0
3	116	103	68	112	1	41	73	2	0	0
4	44	39	25	42	0	3	41	0	0	0
5	5	4	4	4	0	1	4	0	0	0
6	4	3	3	3	0	1	1	2	0	0
7	63	48	36	63	0	7	54	2	0	0
8	1403	1187	635	1361	6	414	570	202	217	2
9	87	64	42	78	4	11	32	23	20	1
10	23	18	14	18	3	10	10	2	1	0
11	22	20	16	20	2	8	11	1	2	0
12	275	225	125	266	4	49	131	39	56	1
13	12	9	6	12	0	6	4	1	1	0
14	19	18	11	19	0	13	4	2	0	0
15	1	1	0	1	0	0	0	1	0	0
16	18	16	9	17	1	6	12	0	0	0
17	6	6	2	6	0	4	2	0	0	0
18	74	70	30	72	0	29	34	6	5	0
19	1	1	0	1	0	1	0	0	0	0
20	3	3	1	3	0	2	1	0	0	0
21	13	10	2	12	0	1	5	2	5	0
22	128	114	50	116	2	62	42	14	10	0
23	2276	1984	1122	1975	262	334	1138	422	381	248
24	18	13	8	15	3	0	9	0	9	1
25	1956	1321	662	1793	35	13	230	491	1222	6
26	2686	2553	1261	388	2152	92	1479	771	344	2236
27	150	137	63	85	64	17	72	26	35	63
28	6	5	0	6	0	0	4	0	2	0
29	2	1	0	2	0	0	2	0	0	0
30	3	3	1	3	0	0	0	0	3	0

31	2	2	1	2	0	1	0	22	0	1	0	1	0
32	58	45	19	56	0	19	0	22	0	24	12	24	0
33	108	99	39	102	1	39	9	63	15	21	15	21	1
34	43	29	23	39	3	23	4	31	4	4	4	4	3
35	2	2	0	0	2	0	0	1	1	0	1	0	1
36	6	3	1	5	1	1	2	3	0	1	0	1	1
37	2	2	1	1	1	1	0	2	0	0	0	0	0
38	42	31	34	39	1	34	1	37	4	0	4	0	2
39	1	1	1	1	0	1	0	1	0	0	0	0	0
40	1	1	1	0	1	1	0	0	1	0	1	0	0
41	1	1	0	1	0	0	1	0	0	0	0	0	0
42	7	5	6	7	0	6	1	3	2	1	2	1	0
43	4	4	4	1	3	4	1	3	0	0	0	0	2
44	153	138	70	98	40	70	37	69	24	21	24	21	36
45	163	147	73	155	1	73	118	36	3	6	3	6	1
46	8	7	3	8	0	3	1	6	0	1	0	1	0
47	3	3	2	3	0	2	1	1	0	1	0	1	0
48	1	1	0	0	1	0	0	0	0	1	0	1	0
49	2712	2649	1463	2640	15	1463	2677	9	9	17	9	17	5
50	51	46	24	44	1	24	20	15	5	11	5	11	0
51	70	69	38	67	3	38	70	0	0	0	0	0	0
52	2138	2104	1119	2123	4	1119	2052	53	13	20	13	20	2
53	13790	13447	7160	13623	64	7160	12143	1027	287	332	287	332	47
54	243	239	116	189	51	116	173	42	12	16	12	16	48
55	10450	9706	5105	9774	401	5105	6279	2285	792	1090	792	1090	345
56	366	260	229	291	53	229	3	198	113	52	113	52	24
57	131	93	89	105	20	89	1	83	29	18	29	18	10
58	183	142	100	89	69	100	5	120	39	19	39	19	28
59	61	49	41	44	14	41	1	42	8	10	8	10	10
60	36	33	19	10	23	19	2	29	3	2	3	2	4
61	19	13	12	9	10	12	1	9	7	2	7	2	4
62	56	39	34	31	18	34	0	38	10	8	10	8	4
63	57	51	32	40	11	32	2	36	11	8	11	8	8
64	11	8	8	7	4	8	0	6	3	2	3	2	0
65	59	46	33	37	14	33	10	31	11	7	11	7	6

66	39	32	18	29	9	1	25	6	7	3
67	16	10	10	14	1	1	8	2	5	1
68	1197	889	752	945	178	52	669	293	182	77
69	34	30	15	18	14	0	32	2	0	6
70	8	7	3	5	2	0	3	2	3	0
71	154	110	95	118	24	3	87	40	24	8
72	30	19	23	27	1	3	13	12	2	0
73	14	13	9	9	4	2	10	1	1	0
74	896	666	557	675	144	53	535	203	105	63
75	205	139	126	120	22	32	115	53	5	9
76	3	2	1	2	0	0	2	0	1	0
77	63	48	30	39	8	4	46	11	2	2
78	352	267	167	207	80	24	242	66	20	13
79	65	60	38	19	36	0	29	29	7	9
80	375	339	145	131	224	55	277	33	9	90
81	189	120	85	143	33	4	165	13	7	11
82	55	44	34	23	28	11	36	6	2	16
83	4	4	2	2	2	0	2	1	1	1
84	2	2	1	0	2	0	2	0	0	1
85	6	5	4	4	1	0	2	3	1	1
86	100	69	48	65	24	4	50	24	22	4
87	6	4	4	5	1	0	4	2	0	1
88	14	9	5	8	5	2	12	0	0	0
89	4	3	3	4	0	0	1	2	1	0
90	5	2	4	5	0	0	5	0	0	0
91	136	97	73	112	13	6	96	20	13	4
92	1	1	1	0	1	0	1	0	0	1
93	1	1	0	1	0	0	0	0	1	0
94	1	1	1	0	0	1	0	0	0	0
95	5	5	2	1	4	0	1	1	3	2
96	5	1	5	5	0	0	5	0	0	0
97	2	2	1	0	2	0	2	0	0	1
98	5	4	2	3	1	0	5	0	0	0
99	2	2	2	0	2	0	2	0	0	0
100	4	3	2	4	0	0	0	1	3	0

Ek 2 EPISODE5 Tablosu için eliminasyon metoduyla öğrenilen kurallar (İlk 10 sorgu için)

Çizelge Ek 2. 1 EPISODE5 Tablosu için eliminasyon metoduyla öğrenilen kurallar (İlk 10 sorgu için)

Antatt	antop	antvalue	antindex	antkey	antın	consatt	consop	consvalue	consindex	conskey	consin
EPISODES	=	1	1	0	1995	AGE_0_14	<=	1	0	0	16199
EPISODES	=	1	1	0	1995	AGE_0_14	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_15_59	<=	1	0	0	6408
EPISODES	=	1	1	0	1995	AGE_15_59	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_60_74	<=	1	0	0	15340
EPISODES	=	1	1	0	1995	AGE_60_74	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_75	<=	1	0	0	16950
EPISODES	=	1	1	0	1995	AGE_75	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	ID	<=	41259	0	1	41046
EPISODES	=	1	1	0	1995	ID	>=	15	0	1	41032
EPISODES	=	2	1	0	1412	AGE_0_14	<=	2	0	0	18466
EPISODES	=	2	1	0	1412	AGE_0_14	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_15_59	<=	2	0	0	8220
EPISODES	=	2	1	0	1412	AGE_15_59	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_60_74	<=	2	0	0	17207
EPISODES	=	2	1	0	1412	AGE_60_74	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_75	<=	2	0	0	19018
EPISODES	=	2	1	0	1412	AGE_75	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	ID	<=	41249	0	1	41036
EPISODES	=	2	1	0	1412	ID	>=	1	0	1	41046
EPISODES	=	3	1	0	1077	AGE_0_14	<=	3	0	0	20115
EPISODES	=	3	1	0	1077	AGE_0_14	>=	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_15_59	<=	3	0	0	9582
EPISODES	=	3	1	0	1077	AGE_15_59	>=	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_60_74	<=	3	0	0	18609
EPISODES	=	3	1	0	1077	AGE_60_74	>=	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_75	<=	3	0	0	20480
EPISODES	=	3	1	0	1077	AGE_75	>=	0	0	0	41046
EPISODES	=	3	1	0	1077	ID	<=	41248	0	1	41035
EPISODES	=	3	1	0	1077	ID	>=	20	0	1	41027

EPISODES	=	4	1	0	928	AGE_0_14	<=	4	0	0	0	21310
EPISODES	=	4	1	0	928	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_15_59	<=	4	0	0	0	10673
EPISODES	=	4	1	0	928	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_60_74	<=	4	0	0	0	19682
EPISODES	=	4	1	0	928	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_75	<=	4	0	0	0	21601
EPISODES	=	4	1	0	928	AGE_75	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	ID	<=	41246	0	1	0	41033
EPISODES	=	4	1	0	928	ID	>=	6	0	1	0	41041
EPISODES	=	5	1	0	715	AGE_0_14	<=	5	0	0	0	22308
EPISODES	=	5	1	0	715	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_15_59	<=	5	0	0	0	11620
EPISODES	=	5	1	0	715	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_60_74	<=	5	0	0	0	20530
EPISODES	=	5	1	0	715	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_75	<=	5	0	0	0	22488
EPISODES	=	5	1	0	715	AGE_75	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	ID	<=	41198	0	1	0	40985
EPISODES	=	5	1	0	715	ID	>=	5	0	1	0	41042
EPISODES	=	6	1	0	641	AGE_0_14	<=	6	0	0	0	23140
EPISODES	=	6	1	0	641	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_15_59	<=	6	0	0	0	12430
EPISODES	=	6	1	0	641	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_60_74	<=	6	0	0	0	21256
EPISODES	=	6	1	0	641	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_75	<=	5	0	0	0	22488
EPISODES	=	6	1	0	641	AGE_75	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	ID	<=	41223	0	1	0	41010
EPISODES	=	6	1	0	641	ID	>=	2	0	1	0	41045
EPISODES	=	7	1	0	572	AGE_0_14	<=	7	0	0	0	23882
EPISODES	=	7	1	0	572	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_15_59	<=	7	0	0	0	13138
EPISODES	=	7	1	0	572	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_60_74	<=	7	0	0	0	21876

EPISODES	=	7	1	0	572	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_75	<=	7	0	0	0	23864
EPISODES	=	7	1	0	572	AGE_75	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	ID	<=	41258	0	1	1	41045
EPISODES	=	7	1	0	572	ID	>=	42	0	1	1	41005
EPISODES	=	8	1	0	578	AGE_0_14	<=	8	0	0	0	24510
EPISODES	=	8	1	0	578	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_15_59	<=	8	0	0	0	13724
EPISODES	=	8	1	0	578	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_60_74	<=	7	0	0	0	21876
EPISODES	=	8	1	0	578	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_75	<=	8	0	0	0	24410
EPISODES	=	8	1	0	578	AGE_75	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	ID	<=	41193	0	1	1	40980
EPISODES	=	8	1	0	578	ID	>=	46	0	1	1	41001
EPISODES	=	9	1	0	500	AGE_0_14	<=	9	0	0	0	25046
EPISODES	=	9	1	0	500	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_15_59	<=	9	0	0	0	14336
EPISODES	=	9	1	0	500	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_60_74	<=	8	0	0	0	22445
EPISODES	=	9	1	0	500	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_75	<=	8	0	0	0	24410
EPISODES	=	9	1	0	500	AGE_75	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	ID	<=	41256	0	1	1	41043
EPISODES	=	9	1	0	500	ID	>=	129	0	1	1	40918
EPISODES	=	10	1	0	491	AGE_0_14	<=	10	0	0	0	25563
EPISODES	=	10	1	0	491	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_15_59	<=	10	0	0	0	14829
EPISODES	=	10	1	0	491	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_60_74	<=	9	0	0	0	22931
EPISODES	=	10	1	0	491	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_75	<=	10	0	0	0	25365
EPISODES	=	10	1	0	491	AGE_75	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	ID	<=	41100	0	1	1	40887
EPISODES	=	10	1	0	491	ID	>=	122	0	1	1	40925

Ek 3 EPISODE5 Tablosu için eliminasyonsuz öğrenilen kurallar (İlk 10 sorgu için)

Çizelge Ek 3. 1 EPISODE5 Tablosu için eliminasyonsuz öğrenilen kurallar (İlk 10 sorgu için)

antatt	antop	antvalue	antindex	antkey	antrn	consatt	consop	consvalue	consindex	conskey	consrn
EPISODES	=	1	1	0	1995	ADMISSIONS	<=	1	0	0	2319
EPISODES	=	1	1	0	1995	ADMISSIONS	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_0_14	<=	1	0	0	16199
EPISODES	=	1	1	0	1995	AGE_0_14	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_15_59	<=	1	0	0	6408
EPISODES	=	1	1	0	1995	AGE_15_59	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_60_74	<=	1	0	0	15340
EPISODES	=	1	1	0	1995	AGE_60_74	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	AGE_75	<=	1	0	0	16950
EPISODES	=	1	1	0	1995	AGE_75	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	DAY_CASE	<=	1	0	0	14774
EPISODES	=	1	1	0	1995	DAY_CASE	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	EMERGENCY	<=	1	0	0	7857
EPISODES	=	1	1	0	1995	EMERGENCY	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	ID	<=	41259	0	1	41046
EPISODES	=	1	1	0	1995	ID	>=	15	0	1	41032
EPISODES	=	1	1	0	1995	MALE	<=	1	0	0	7514
EPISODES	=	1	1	0	1995	MALE	>=	0	0	0	41046
EPISODES	=	1	1	0	1995	WAITING_LIST	<=	1	0	0	10403
EPISODES	=	1	1	0	1995	WAITING_LIST	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	ADMISSIONS	<=	2	0	0	3812
EPISODES	=	2	1	0	1412	ADMISSIONS	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_0_14	<=	2	0	0	18466
EPISODES	=	2	1	0	1412	AGE_0_14	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_15_59	<=	2	0	0	8220
EPISODES	=	2	1	0	1412	AGE_15_59	>=	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_60_74	<=	2	0	0	17207

EPISODES	=	2	1	0	1412	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	2	1	0	1412	AGE_75	<=	2	0	0	0	19018
EPISODES	=	2	1	0	1412	AGE_75	>=	0	0	0	0	41046
EPISODES	=	2	1	0	1412	DAY_CASE	<=	2	0	0	0	16799
EPISODES	=	2	1	0	1412	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	2	1	0	1412	EMERGENCY	<=	2	0	0	0	10169
EPISODES	=	2	1	0	1412	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	2	1	0	1412	ID	<=	41249	0	0	1	41036
EPISODES	=	2	1	0	1412	ID	>=	1	0	0	1	41046
EPISODES	=	2	1	0	1412	MALE	<=	2	0	0	0	9166
EPISODES	=	2	1	0	1412	MALE	>=	0	0	0	0	41046
EPISODES	=	2	1	0	1412	WAITING_LIST	<=	2	0	0	0	12344
EPISODES	=	2	1	0	1412	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	ADMISSIONS	<=	3	0	0	0	4932
EPISODES	=	3	1	0	1077	ADMISSIONS	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_0_14	<=	3	0	0	0	20115
EPISODES	=	3	1	0	1077	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_15_59	<=	3	0	0	0	9582
EPISODES	=	3	1	0	1077	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_60_74	<=	3	0	0	0	18609
EPISODES	=	3	1	0	1077	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	AGE_75	<=	3	0	0	0	20480
EPISODES	=	3	1	0	1077	AGE_75	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	DAY_CASE	<=	3	0	0	0	18229
EPISODES	=	3	1	0	1077	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	EMERGENCY	<=	3	0	0	0	11886
EPISODES	=	3	1	0	1077	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	ID	<=	41248	0	0	1	41035
EPISODES	=	3	1	0	1077	ID	>=	20	0	0	1	41027
EPISODES	=	3	1	0	1077	MALE	<=	3	0	0	0	10421
EPISODES	=	3	1	0	1077	MALE	>=	0	0	0	0	41046
EPISODES	=	3	1	0	1077	WAITING_LIST	<=	3	0	0	0	13734
EPISODES	=	3	1	0	1077	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	ADMISSIONS	<=	4	0	0	0	5845
EPISODES	=	4	1	0	928	ADMISSIONS	>=	0	0	0	0	41046

EPISODES	=	4	1	0	928	AGE_0_14	<=	4	0	0	0	21310
EPISODES	=	4	1	0	928	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_15_59	<=	4	0	0	0	10673
EPISODES	=	4	1	0	928	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_60_74	<=	4	0	0	0	19682
EPISODES	=	4	1	0	928	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	AGE_75	<=	4	0	0	0	21601
EPISODES	=	4	1	0	928	AGE_75	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	DAY_CASE	<=	4	0	0	0	19284
EPISODES	=	4	1	0	928	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	EMERGENCY	<=	4	0	0	0	13316
EPISODES	=	4	1	0	928	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	ID	<=	41246	0	0	1	41033
EPISODES	=	4	1	0	928	ID	>=	6	0	0	1	41041
EPISODES	=	4	1	0	928	MALE	<=	4	0	0	0	11460
EPISODES	=	4	1	0	928	MALE	>=	0	0	0	0	41046
EPISODES	=	4	1	0	928	WAITING_LIST	<=	4	0	0	0	14853
EPISODES	=	4	1	0	928	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	ADMISSIONS	<=	5	0	0	0	6579
EPISODES	=	5	1	0	715	ADMISSIONS	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_0_14	<=	5	0	0	0	22308
EPISODES	=	5	1	0	715	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_15_59	<=	5	0	0	0	11620
EPISODES	=	5	1	0	715	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_60_74	<=	5	0	0	0	20530
EPISODES	=	5	1	0	715	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	AGE_75	<=	5	0	0	0	22488
EPISODES	=	5	1	0	715	AGE_75	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	DAY_CASE	<=	5	0	0	0	20144
EPISODES	=	5	1	0	715	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	EMERGENCY	<=	5	0	0	0	14433
EPISODES	=	5	1	0	715	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	ID	<=	41198	0	0	1	40985
EPISODES	=	5	1	0	715	ID	>=	5	0	0	1	41042
EPISODES	=	5	1	0	715	MALE	<=	5	0	0	0	12294

EPISODES	=	5	1	0	715	MALE	>=	0	0	0	0	41046
EPISODES	=	5	1	0	715	WAITING_LIST	<=	5	0	0	0	15786
EPISODES	=	5	1	0	715	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	ADMISSIONS	<=	6	0	0	0	7276
EPISODES	=	6	1	0	641	ADMISSIONS	>=	1	0	0	0	40892
EPISODES	=	6	1	0	641	AGE_0_14	<=	6	0	0	0	23140
EPISODES	=	6	1	0	641	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_15_59	<=	6	0	0	0	12430
EPISODES	=	6	1	0	641	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_60_74	<=	6	0	0	0	21256
EPISODES	=	6	1	0	641	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	AGE_75	<=	5	0	0	0	22488
EPISODES	=	6	1	0	641	AGE_75	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	DAY_CASE	<=	6	0	0	0	20902
EPISODES	=	6	1	0	641	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	EMERGENCY	<=	6	0	0	0	15388
EPISODES	=	6	1	0	641	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	ID	<=	41223	0	0	1	41010
EPISODES	=	6	1	0	641	ID	>=	2	0	0	1	41045
EPISODES	=	6	1	0	641	MALE	<=	6	0	0	0	13050
EPISODES	=	6	1	0	641	MALE	>=	0	0	0	0	41046
EPISODES	=	6	1	0	641	WAITING_LIST	<=	6	0	0	0	16563
EPISODES	=	6	1	0	641	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	ADMISSIONS	<=	7	0	0	0	7894
EPISODES	=	7	1	0	572	ADMISSIONS	>=	2	0	0	0	38727
EPISODES	=	7	1	0	572	AGE_0_14	<=	7	0	0	0	23882
EPISODES	=	7	1	0	572	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_15_59	<=	7	0	0	0	13138
EPISODES	=	7	1	0	572	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_60_74	<=	7	0	0	0	21876
EPISODES	=	7	1	0	572	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	AGE_75	<=	7	0	0	0	23864
EPISODES	=	7	1	0	572	AGE_75	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	DAY_CASE	<=	7	0	0	0	21527
EPISODES	=	7	1	0	572	DAY_CASE	>=	0	0	0	0	41046

EPISODES	=	7	1	0	572	EMERGENCY	<=	7	0	0	0	16198
EPISODES	=	7	1	0	572	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	ID	<=	41258	0	0	1	41045
EPISODES	=	7	1	0	572	ID	>=	42	0	0	1	41005
EPISODES	=	7	1	0	572	MALE	<=	7	0	0	0	13769
EPISODES	=	7	1	0	572	MALE	>=	0	0	0	0	41046
EPISODES	=	7	1	0	572	WAITING_LIST	<=	7	0	0	0	17254
EPISODES	=	7	1	0	572	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	ADMISSIONS	<=	8	0	0	0	8496
EPISODES	=	8	1	0	578	ADMISSIONS	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_0_14	<=	8	0	0	0	24510
EPISODES	=	8	1	0	578	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_15_59	<=	8	0	0	0	13724
EPISODES	=	8	1	0	578	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_60_74	<=	7	0	0	0	21876
EPISODES	=	8	1	0	578	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	AGE_75	<=	8	0	0	0	24410
EPISODES	=	8	1	0	578	AGE_75	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	DAY_CASE	<=	8	0	0	0	22117
EPISODES	=	8	1	0	578	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	EMERGENCY	<=	8	0	0	0	16894
EPISODES	=	8	1	0	578	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	ID	<=	41193	0	0	1	40980
EPISODES	=	8	1	0	578	ID	>=	46	0	0	1	41001
EPISODES	=	8	1	0	578	MALE	<=	8	0	0	0	14353
EPISODES	=	8	1	0	578	MALE	>=	0	0	0	0	41046
EPISODES	=	8	1	0	578	WAITING_LIST	<=	8	0	0	0	17898
EPISODES	=	8	1	0	578	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	ADMISSIONS	<=	9	0	0	0	9031
EPISODES	=	9	1	0	500	ADMISSIONS	>=	2	0	0	0	38727
EPISODES	=	9	1	0	500	AGE_0_14	<=	9	0	0	0	25046
EPISODES	=	9	1	0	500	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_15_59	<=	9	0	0	0	14336
EPISODES	=	9	1	0	500	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_60_74	<=	8	0	0	0	22445

EPISODES	=	9	1	0	500	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	AGE_75	<=	8	0	0	0	24410
EPISODES	=	9	1	0	500	AGE_75	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	DAY_CASE	<=	9	0	0	0	22612
EPISODES	=	9	1	0	500	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	EMERGENCY	<=	9	0	0	0	17548
EPISODES	=	9	1	0	500	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	ID	<=	41256	0	0	1	41043
EPISODES	=	9	1	0	500	ID	>=	129	0	0	1	40918
EPISODES	=	9	1	0	500	MALE	<=	9	0	0	0	14912
EPISODES	=	9	1	0	500	MALE	>=	0	0	0	0	41046
EPISODES	=	9	1	0	500	WAITING_LIST	<=	9	0	0	0	18405
EPISODES	=	9	1	0	500	WAITING_LIST	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	ADMISSIONS	<=	10	0	0	0	9499
EPISODES	=	10	1	0	491	ADMISSIONS	>=	3	0	0	0	37234
EPISODES	=	10	1	0	491	AGE_0_14	<=	10	0	0	0	25563
EPISODES	=	10	1	0	491	AGE_0_14	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_15_59	<=	10	0	0	0	14829
EPISODES	=	10	1	0	491	AGE_15_59	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_60_74	<=	9	0	0	0	22931
EPISODES	=	10	1	0	491	AGE_60_74	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	AGE_75	<=	10	0	0	0	25365
EPISODES	=	10	1	0	491	AGE_75	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	DAY_CASE	<=	10	0	0	0	23067
EPISODES	=	10	1	0	491	DAY_CASE	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	EMERGENCY	<=	10	0	0	0	18130
EPISODES	=	10	1	0	491	EMERGENCY	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	ID	<=	41100	0	0	1	40887
EPISODES	=	10	1	0	491	ID	>=	122	0	0	1	40925
EPISODES	=	10	1	0	491	MALE	<=	10	0	0	0	15393
EPISODES	=	10	1	0	491	MALE	>=	0	0	0	0	41046
EPISODES	=	10	1	0	491	WAITING_LIST	<=	10	0	0	0	18919
EPISODES	=	10	1	0	491	WAITING_LIST	>=	0	0	0	0	41046

Ek 4 İnsan2 Tablosu veri bilgileri (İlk 30 kayıt)

Çizelge Ek 4. 1 İnsan2 Tablosu veri bilgileri (İlk 30 kayıt)

sıfano	kilo	boy	yas	cocuks	m.hal	cinsiyet	maas	kidem	araba	arabafiyat	arabamodel	arabamarka	kangrup	annekangrup	babakangrup
0	56	165	26	0	0	1	727	2	1	7619	1997	5	1	1	1
1	66	170	37	2	1	1	1766	6	1	33689	1994	1	1	1	2
2	75	180	31	2	1	0	1740	5	1	33689	1994	1	2	2	1
3	57	160	29	2	1	1	1391	4	1	24687	2002	2	2	2	2
4	82	190	26	0	0	0	454	2	0	0	0	0	3	1	2
5	64	168	27	2	1	1	1365	3	1	24687	2002	2	4	4	1
6	68	175	28	0	1	0	895	4	1	10850	1995	3	2	2	2
7	78	185	24	0	0	0	428	1	0	0	0	0	2	2	1
8	63	169	23	1	1	0	428	1	0	0	0	0	2	2	2
9	53	162	26	0	0	1	727	2	1	7619	1997	5	2	2	2
10	70	175	30	2	1	1	1444	6	1	24687	2002	2	2	1	2
11	69	174	33	1	1	0	869	3	1	10850	1995	3	1	1	2
12	65	167	36	4	1	1	10182	12	1	264150	2003	4	1	1	1
13	75	182	26	0	0	0	454	2	0	0	0	0	2	2	1
14	71	174	39	4	1	0	10261	15	1	264150	2003	4	1	1	2
15	53	158	34	1	1	1	10130	10	1	264150	2003	4	3	3	1
16	75	182	28	0	0	0	506	4	0	0	0	0	1	1	1
17	60	163	40	3	1	1	4565	11	1	108125	2003	1	1	1	1
18	61	165	31	2	1	0	1470	7	1	24687	2002	2	4	4	4
19	82	190	21	0	0	0	428	1	0	0	0	0	1	1	1
20	69	172	28	3	1	1	1391	4	1	24687	2002	2	4	4	4
21	57	163	25	0	0	0	843	2	1	10850	1995	3	2	2	2
22	66	172	27	1	1	0	480	3	0	0	0	0	3	1	2
23	70	180	25	0	1	1	727	2	1	7619	1997	5	1	1	1
24	52	160	30	0	1	1	869	3	0	10850	1995	3	2	1	2
25	61	165	35	2	1	1	1819	8	1	33689	1994	1	4	2	4
26	47	155	20	0	0	1	428	1	0	0	0	0	2	2	2
27	70	177	21	0	0	0	428	1	0	0	0	0	4	4	3

28	81	187	34	0	1	0	1740	5	1	33689	1994	1	1	1	1	1
29	66	170	45	2	1	1	4644	14	1	108125	2003	1	1	1	1	2



ÖZGEÇMİŞ

Doğum Tarihi	14.11.1978	
Doğum Yeri	İstanbul	
Lise	1989-1996	Özel Işık Lisesi
Lisans	1997-2001	Yıldız Teknik Üniversitesi Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programı
Yüksek Lisans	2001-2004	Yıldız Teknik Üniversitesi Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programı

