

**T.C.  
HALIÇ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**RASTLANTISAL YAPAY SINIR AĞLARININ VERİ  
MADENCİLİĞİNDE KULLANILMASI**

**YÜKSEK LİSANS TEZİ**

**Hazırlayan  
Sabri Serkan GÜLLÜOĞLU**

**Tez Danışmanı  
Prof.Dr. Ali OKATAN**

**Temmuz 2007  
İSTANBUL**

HALIC UNIVERSITY  
THE INSTITUTE OF SCIENCE  
COMPUTER ENGINEERING

USING PROBABILISTIC NEURAL NETWORKS ON DATA MINING

MASTER THESIS

Prepared By  
Sabri Serkan GÜLLÜOĞLU

Supervisor  
Prof. Dr. Ali OKATAN

July 2007  
İSTANBUL

T.C.  
HALIÇ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Programı Yüksek Lisans öğrencisi Sabri Serkan Güllüoğlu tarafından hazırlanan “**Raslantısal Yapay Sinir Ağlarının Veri Madenciliğinde Kullanılması**” adlı bu çalışma jürimizce Yüksek Lisans Tezi olarak Kabul Edilmiştir.

Tez Savunma Tarihi : 10.07.2007

( Jüri Üyesinin Ünvanı , Adı , Soyadı ve Kurumu ) :

İmzası :

Jüri Üyesi: Prof.Dr.Ali OKATAN  
(Danışman)



Jüri Üyesi : Prof.Dr.Bekir KARAOĞLU  
(Elektronik ve Hab.ABD Öğr.Üyesi)



Jüri Üyesi :Yrd.Doç.Dr. Murat BEKEN



## İÇİNDEKİLER

ÖNSÖZ.....	III
ŞEKİLLER LİSTESİ.....	IV
ÖZET.....	VI
ABSTRACT.....	VIII
<b>1.GİRİŞ.....</b>	<b>1</b>
1.1.Çalışmanın Amacı.....	1
<b>2.YAPAY SİNİR AĞLARI.....</b>	<b>2</b>
2.1. Yapay Sinir Ağları'na (YSA) Giriş.....	2
2.1.1. YSA'nın Avantajları.....	4
2.1.2. YSA'nın Dezavantajları.....	4
2.2. YSA'nın Tanımı ve Modeli.....	5
2.2.1. Biyolojik Sinirin Yapısı.....	5
2.3. YSA'nın İşlem Elemanı.....	6
<b>3. YSA'DA EĞİTME VE ÖĞRENME.....</b>	<b>7</b>
3.1.Yapay Sinir Ağları Tanımlanması.....	7
3.2. Ağırlık Uzayı.....	8
<b>4. SINIFLANDIRMA.....</b>	<b>8</b>
4.1. Eğitim ve Test Etme Sınıflandırıcı Süreçleri.....	9
4.1.1. Bayes karar teorisi.....	10
4.1.2. Eğitim.....	11
<b>5. OLASILIKSAL SİNİR AĞLARI.....</b>	<b>11</b>
5.1. Olasılıksal Sinir Ağı.....	11
5.1.1. Matematiksel temel.....	11
5.1.2. Eğitim.....	13
5.1.3. Test etme.....	13
5.1.4. Uyarılama.....	13
<b>6. VERİ MADENCİLİĞİ.....</b>	<b>14</b>
6.1. Veri Madenciliği.....	14
6.1.1. Veri Madenciliği Nedir?.....	15
6.1.2. Veri madenciliğinin sınırları.....	15
6.2. Veri Madenciliğinde Önemli Sorunlar.....	17
6.2.1. Veritabanlarının büyük boyutları.....	17
6.3. İlişki Analizi.....	17
6.3.1. Temel Kavramlar.....	19
6.4. Sınıflandırma ve Tahmin Yürütme.....	20

6.5. Olasılıksal Sinir Ağları Çalışması.....	21
6.5.1. Proje Kapsamında Yapılanlar .....	22
6.6. Proje Algoritma Tanımlanması .....	33
6.6.1. Olasılıksal Sinir Ağlarında Eğitim Aşamaları.....	36
6.6.2. Son Aşama .....	39
6.6.2.1. Test Aşaması .....	39
6.3. Programın Ayrıntılı İçeriği.....	54
6.3.1. Ürünü Temsil Sınıfın Oluşturulması .....	54
6.3.1.1. ProductInfo.cs Sınıfı.....	54
6.3.2. ProductInfo sınıfından nesne oluşturulması .....	55
6.3.3. Kodun Eğitim Kısım.....	56
6.3.4. Verilerin Test Aşaması .....	60
<b>7. VERİ MADENCİLİĞİ ÇALIŞMASI.....</b>	<b>67</b>
7.1. Proje 1.Kısımın Hazırlık Aşaması.....	67
7.1.1. Ragesh Agrawal'ın geliştirdiği veritabanı gösterim tabloları.....	68
7.2. Hazırlanan Proje Hakkında.....	69
7.3. İstatistik.....	75
7.4. İkinci Bölüm.....	75
7.4.1. Karar Ağacı Mekanizması .....	75
7.5. Üründen Müşteri Profili.....	82
7.5.1. Birinci Hesaplama .....	82
7.5.2. İkinci Hesaplama.....	84
7.5.3. Üçüncü Hesaplama.....	86
7.6. Müşteriden Ürün Bulunması .....	91
<b>8. SONUÇ.....</b>	<b>95</b>
<b>9. KAYNAKLAR.....</b>	<b>96</b>
<b>10. ÖZGEÇMİŞ.....</b>	<b>97</b>

## ÖNSÖZ

Sayın Prof. Dr. Ali OKATAN 'a, tez çalışmasının gerçekleştirilmesinde gerekli yönlendirici desteği sağladığı, tez metnini inceleyerek biçim ve içerik bakımından son şeklini almasında katkıda bulunduğu için teşekkür ederim.

Sayın Öğretim Görevlisi Oğuz KARAN 'a tez metnini inceleyerek biçim ve içerik bakımından son şeklini almasında katkıda bulunduğu ve yardımlarını esirgemediği için teşekkür ederim.

Sayın Araştırma Görevlisi Atınç Yılmaz' a katkılarından dolayı teşekkür ederim.

Ayrıca Sevgili Eşim Nilgün GÜLLÜOĞLU 'na, yoğun çalışma zamanlarımda desteğini esirgemediği için teşekkür ederim.

Sabri Serkan Güllüoğlu  
Temmuz, 2007

## Şekiller Listesi

Şekil 1. Basit Bir Neron Yapısı.....	5
Şekil 2. Yapay Nöron Modeli.....	6
Şekil 3. Tek katmanlı YSA.....	6
Şekil 4. Üç Katmanlı YSA.....	7
Şekil 5. YSA'da Eğitim.....	8
Şekil 6. Eğitim Süreci.....	10
Şekil 7. Olasılıksal Sinir Ağı Yapısı.....	13
Şekil 8. Yapay sinir ağ örüntü sınıflandırıcıları.....	14
Şekil 9. Veri Tablosu.....	16
Şekil 10. Yaş, meslek, semt ve Ürün Matrisi.....	22
Şekil 11. Yaş Aralıklarının Sayısal Eşleştirilmesi.....	23
Şekil 12. Meslek Aralıklarının Sayısal Eşleştirilmesi.....	24
Şekil 13. Semt Aralıklarının Sayısal Eşleştirilmesi.....	24
Şekil 14. MathLab'de Not Defterinden Matrislerin Alınması.....	25
Şekil 15. MathLab'de Not Defterinden Matrislerin Alınmış Gösterimi.....	26
Şekil 16. MathLab'de Not Defterinden Yaş Matrisinin Alınması.....	26
Şekil 17. MathLab'de Not Defterinden Yaş Matrisinin 2*2 lik Gösterimi.....	27
Şekil 18. MathLab'de Not Defterinden Yaş Matrisinin Oluşturulması.....	28
Şekil 19. MathLab'de Not Defterinden Meslek Matrisinin Alınması.....	28
Şekil 20. MathLab'de Not Defterinden Meslek Matrisinin 2*2 lik Gösterimi.....	29
Şekil 21. MathLab'de Not Defterinden Meslek Matrisinin 2*2 lik Gösterimi.....	30
Şekil 22. MathLab'de Not Defterinden Semt Matrisinin Alınması.....	30
Şekil 23. MathLab'de Not Defterinden Semt Matrisinin Oluşturulması.....	31
Şekil 24. MathLab'de Not Defterinden Semt Matrisinin 2*2 lik Gösterimi.....	31
Şekil 25. MathLab'de Not Defterinden Semt Matrisinin 2*2 lik Gösterimi.....	32
Şekil 26. MathLab'de Semt, Yaş ve Meslek Özellikleri Ele Alınarak Çizilen 3 Boyutlu Grafik.....	32
Şekil 27. Olasılıksal Sinir Ağı Yapısı.....	34
Şekil 28. PNN Windows Uygulaması Çalıştırılmış Hali.....	52
Şekil 29. PNN Uygulamasında Ele Alınan Özelliklerin Windows Uygulaması gösterimi...60	
Şekil 30. PNN Uygulama Eğitim Aşaması.....	63
Şekil 31. Eğitim İşleminin Tamamlandığı Göstergesi.....	64
Şekil 32. PNN Windows Uygulaması Test Aşamasında.....	65
Şekil 33. PNN Programı Sonuç.....	66
Şekil 34. Veritabanı Gösterim Tabloları.....	68
Şekil 35. Veritabanı İlişkiler Gösterimi.....	69
Şekil 36. Excel de Ürün Gösterimi.....	70
Şekil 37. Excel de Sipariş Tablosu 1.....	70
Şekil 38. Excel de Sipariş Tablosu 2.....	71
Şekil 39. İlişkilendirilmiş Veri Madenciliği Arayüzü.....	72
Şekil 40. Tekli Alınan Ürünler Listesi.....	73
Şekil 41. İkili Alınan Ürünler Listesi.....	73
Şekil 42. Üçlü Alınan Ürünler Listesi.....	74
Şekil 43. Müşteri Yüzdeler Dilimleri.....	74
Şekil 44. Karar Ağacı Oluşturulan Windows Uygulaması.....	76
Şekil 45. Üründen Müşteri Profili.....	79
Şekil 46. Müşteri Tablosu.....	80
Şekil 47. Ürün Tablosu.....	81

Şekil 48. Birinci Hesap ta Oluşacak Olan Karar Ağacı.....	83
Şekil 49. İkinci Hesapta Oluşacak olana Karar Ağacı.....	85
Şekil 50. Ürün Profiline Cinsiyet, Yaş, Medeni Durum Değerlendirmesi.....	87
Şekil 51. Üçüncü Hesapta Oluşacak Olan Karar Ağacı.....	88
Şekil 52. Müşterilerin Sipariş Ettiği Ürün Tablosu.....	90
Şekil 53. Müşteriden Ürün Bulunma Sonucu Oluşan Tablo.....	91
Şekil 54. Sadece Cinsiyet Durumuna Göre Listeleme.....	92
Şekil 55. Cinsiyet ve Medeni Durum İrdelenmesi.....	94



T.C.  
HALIÇ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
YÜKSEK LİSANS TEZİ

RASTLANTISAL YAPAY SİNİR AĞLARININ VERİ MADENCİLİĞİNDE KULLANILMASI

Hazırlayan  
Sabri Serkan Güllüoğlu

Tez Danışmanı  
Prof. Dr. Ali Okatan

Temmuz, 2007

**ÖZET**

Bilgisayar Teknolojisi ve veri toplama araçlarındaki gelişmeler ile birlikte, veri analizi yöntemlerinde hızlı bir gelişim görülmektedir. Veri toplama araçlarındaki gelişim ile birlikte işletmelerde toplanan veriler her geçen daha büyük boyutlara ulaşmaktadır. İşletmelerin, faaliyetlerini yürütebilmeleri ve çeşitli kararlar alabilmeleri için veritabanlarında veya veri ambarlarında toplanan bu verilere ihtiyaçları vardır. Örneğin bir işletmenin pazardaki mevcut konumunun belirlenmesi ile elde edilen bilgilerin ışığında, gerek ürün gerekse Pazar ile ilgili alınacak yeni kararlar stratejik değerdedir. Bu nedenle, işletmelerin pazarlama kararlarını destekleyen bilgi sistemlerine uzun süredir ilgi yoğunudur. Veri madenciliği bu işlemi yapan en önemli araçlardan biridir. Büyük hacimli verilerin içinden çeşitli teknikler ile gizli kalmış örüntü, kural ve ilişkilerin otomatik olarak kolayca ortaya çıkarılması süreci olarak tanımlanabilir.

Bu çalışmada ilk olarak olasılıksal sinir ağı yardımıyla kişilerin markette ürün tercihlerinin sınıflandırılması üzerinde durulmuştur. İkinci kısımda ise ilişkilendirilmiş veri madenciliği ile markette birlikte alınan ürünlerin benzerliği saptanacaktır.

Tez kapsamında hedeflenen, market 'te bulunan müşteri ve ürünlerin veri madenciliği teknikleriyle özdevimli bir şekilde gruplanması ve bunların en etkili bir şekilde eşleştirilmesinin sağlanması ve herhangi bir markette alışveriş yapan müşterilerin semt, yaş ve meslek özelliklerine göre tercih ettikleri ürünleri belirlemek ve bu verileri bir ortamda saklamak, daha sonra ise bu bilgiye

dayanarak sınıflandırma yöntemlerinden birini kullanarak yeni müşterilerin bu özelliklerine göre tercih ettikleri ürünleri tespit etmektir.

**Anahtar Sözcükler:** Örüntü, Sınıflandırıcı, Veri madenciliği, müşteri ve ürün gruplandırılması ve eşleştirmesi.

HALIC UNIVERSITY  
THE INSTITUTE OF SCIENCE  
COMPUTER ENGINEERING

USING PROBABILISTIC NEURAL NETWORKS ON DATA MINING

Prepared By  
Sabri Serkan GÜLLÜOĞLU

Supervisor  
Prof. Dr. Ali OKATAN

July, 2007  
İstanbul

**ABSTRACT**

With the evolution of computer technology and data collection tools, fast development are observed on data analysis method.

With the improvement on data collection tools, the data that collected are rising day by day. For business actions, the business need data for giving some decisions that are collected in databases or data storehouses.

In this project firstly I am related with the customer product selections on markets and for making this there were used Probabilistic Neural Network algorithms. In the second section we have determined that in market which products are bought together and for this we can use the associative data mining tools and algorithms.

Our real aim is grouping the product and customer with using data mining techniques and providing the appropriate matching. Customers that are shopping on some markets, we have to determine the products that are select by customers, we have to use some properties; state, age and job.

For this properties we can do this. And then we have hidden these data, and with based on these data, we could find new customer selections with their properties for example state, job and age. We have done this with a classification method. These method is probabilistic neural network.

**Key words:** Pattern, Classifier, Data Mining, grouping and matching customer and product.

## 1.GİRİŞ

### 1.1. Çalışmanın Amacı

Tez çalışmasının amacı rastlantısal sinir ağıları algoritmalarını kullanarak veri madenciliğinin temeline örnek oluşturacak projeyi hazırlamaktır.

Projedeki asıl amacımız herhangi bir markette alışveriş yapan müşterilerin semt, yaş ve meslek özelliklerine göre tercih ettikleri ürünleri belirlemek ve bu verileri bir ortamda saklamak, daha sonra ise bu bilgiye dayanarak sınıflandırma yöntemlerinden birini kullanarak yeni müşterilerin bu özelliklerine göre tercih ettikleri ürünleri tespit etmektir.

Neticede elimizde olan bu veriler ile, semt yaş ve meslek özelliklerine göre hangi ürünün hangi semtte hangi yaşa ve mesleğe hitap ettiğini bileceğiz. Bu sayede ileride açılacak bir markette ürün seçiminde önceden kararlar verilebilecektir.

Parametrik olmayan istatistiksel sınıflandırıcıyla verinin yapısı hakkında varsayım yapılamaz. Onun yerine, sonrasal olasılıklar direkt veriden tahmin edilebilir. Böylece yaygın yaklaşım, olasılıksal sinir ağıları algoritmasıdır.

Bu yöntem, örüntü tanımanın en klasik metotlarından birisi olup, tanımlanması istenen örüntünün vektörünü, veritabanındaki en yakın komşusunun sınıfına dâhil ederek tanımlar. Yöntem, örnek vektörün istatistiksel dağılımından bağımsız olup, yalnızca en yakın komşunun sınıfına göre bir sınıflandırma yaparak tanıma işlemini gerçekleştirir. Bu yöntemde, örnek örüntünün vektörü alınarak, veri tabanındaki her bir vektöre olan uzaklığı ölçülür. En çok kullanılan mesafe ölçüsü öklit uzaklığı olsa da, başka herhangi bir ölçü de kullanılabilir. Tanımlanacak olan örnek örüntünün vektörü, veri tabanındaki kendisine en çok benzeyen, örüntünün sınıfından sayılır. Müşteri ve ürün fazlalığı sebebiyle müşteriler ve ürünler iyi bir şekilde gruplanamakta ve eşleştirilememektedir.

Sınıflandırma aşamasının amacı, örüntüleri özellik uzaylarına göre kendilerine en yakın sınıflara minimum hata ile eşlemektir.

Sınıflandırıcının başarımında iyi belirlenmiş özellikler kilit rolü oynarlar. Örüntü sınıflandırıcılarını geleneksel ve akıllı olarak gruplara ayırmak mümkündür. Geleneksel sınıflandırma algoritmaları istatistiksel bir yapı olan Bayes karar teorisi üzerine kuruludur. Bunların dezavantajları, özellik uzayını sınıflandırma uzayını dönüştürürken, bir gürültünün çıkması ve her bir sınıf için hata kriterinin belli olmamasıdır. Geleneksel sınıflandırıcılara; çok değişkenli Gauss modelleri, en yakın komşu, maksimum olasılık, ikili ağaç sınıflandırıcıları ve Fisher' in doğrusal sınıflandırıcıları örnek olarak verilebilir.

Buna karşın akıllı sınıflandırma yapıları genellikle YSA tabanlı olup, günümüzde en yaygın kullanılan ve başarımını ispatlamış çok güçlü sınıflandırıcı türleridirler. Özellikle de genelleme yetenekleri, örüntü tanıma uygulamalarının çok büyük boyutlu verileri açısından önemlidir.

Olasılıksal sinir ağı (PNN, Probabilistic Neural Network) bir Bayes sınıflandırıcının parametrik olmayan bir gerçekleştirini göz önünde bulundurabilir. Bilinmeyen şart yoğunlukları Parzen pencereleri kullanılarak tahmin edilir ve Bayes karar kuralı en yüksek olasılığın çıkış kategorisini elde etmek için uygulanır. PNN eğitim verisinin temel dağıtımına göre hiç bir varsayımın yapılmadığı serbest model (modelden bağımsız) bir tahmin edici olarak düşünülebilir.

## **2.YAPAY SİNİR AĞLARI**

### **2.1. Yapay Sinir Ağlarına (YSA) Giriş**

İnsanlığın doğayı araştırma ve taklit etme çabalarının en son ürünlerinden bir tanesi, yapay sinir ağları (YSA) teknolojisidir. YSA, basit biyolojik sinir sisteminin çalışma şekli simüle edilerek tasarlanan programlama yaklaşımıdır. Simüle edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağ oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar,

normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik (synaptic) bağlantıların ayarlanması ile olur. Yani, insanlar doğumlarından itibaren bir yaşayarak öğrenme süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitime yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıkıtları verilerinin işlenmesiyle, yani eğitime algoritmasının bu verileri kullanarak bağlantı ağırlıklarını (weights of the synapses) bir yakınsama sağlanana kadar, tekrar tekrar ayarlanmasıyla olur.

YSA'lar, ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematiksel sistemlerdir. Bir işlem birimi, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır. Genelde, işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır.

Sinirsel (neural) hesaplamaların merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. YSA'lar, geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. YSA'lar ise herbiri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimlerinden oluşmaktadır. En basit şekilde, bir işlem birimi, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, işlem birimlerinin çalışma şekli yanıtıcı şekilde basittir. Sinirsel hesaplamaların gücü, toplam işlem yükünü paylaşan işlem birimlerinin birbirleri arasındaki yoğun bağlantı yapısından gelmektedir.

Çoğu YSA'da, benzer karakteristiğe sahip nöronlar tabakalar halinde yapılandırılırlar ve transfer fonksiyonları eş zamanlı olarak çalıştırılırlar. Hemen hemen tüm ağlar, veri alan nöronlara ve çıktı üreten nöronlara sahiptirler.

YSA'nın ana ögesi olan matematiksel fonksiyon, ağın mimarisi tarafından şekillendirilir. Daha açık bir şekilde ifade etmek gerekirse, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA'ların davranışları, yani girdi veriyi çıktı veriye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir.

#### **2.1.1. YSA'nın Avantajları:**

- Kullanıcı veri örneklerine dayanan tekil çözümler üretirler
- Algoritmayı bilmeye gereksinim göstermezler
- Çok az yazılım ya da hiç yazılım gereksinimi yoktur, daha çok donanım işlem gücüne gereksinim duyarlar
- Değişik birçok probleme çözüm bulabilirler
- Paralel yapısından dolayı işlem yoğunluklu çok sayıda probleme hızlı çözüm getirirler
- Karmaşık karar alanlarını kendi içlerinde üretirler
- Gürültü yok etmede iyi başarımlar sağlarlar
- Hata dayanma yeteneğini kendiliğinden oluştururlar.

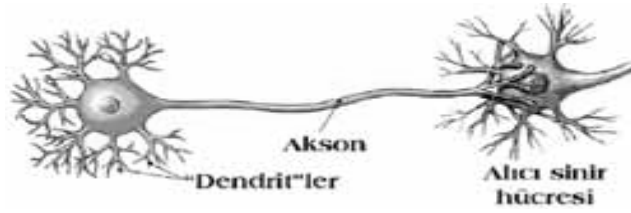
#### **2.1.2. YSA'nın Dezavantajları:**

- Bütün işlemsel problemlere uygulanamazlar
- Eğitim ve test veri örneklerine gereksinim duyarlar.

## 2.2. YSA' nın Tanımı ve Modeli

### 2.2.1. Biyolojik Sinirin (Nöronun) Yapısı

Sinir sistemini oluşturan ve birbirlerine Axon ve Dentritlerle bağlı, aralarında elektriksel akış olan, birbiriyle etkileşim içinde olan, canlılarda bütün refleksel, hareketsetel, algısal, düşünsel olayları sağlayan canlı hücrelere sinir hücreleri denir.



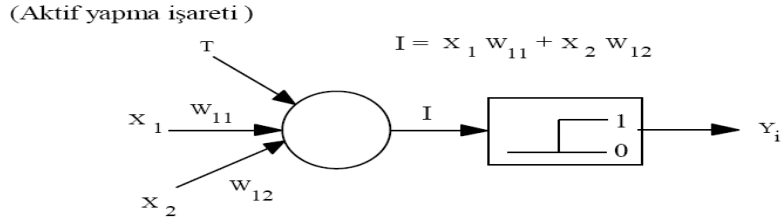
**Şekil 1. Basit Bir Neron Yapısı**

Sinir sistemi yaklaşık olarak 1 trilyon sinir hücresi içerir. Sinir sisteminin temel yapıtaşı olan nöronlar, çeşitli uyarılara kasılarak yanıt verirler. Nöron hücre gövdesi dentrit denilen kıvrımlı uzantılar ve somanın dalları sayesinde nöronu dallarına bağlayan tek sinir fiberli aksondan oluşur.

Dendritler hücreye gelen girişleri toplarlar. Dendrit tarafından alınan işaretler hücrede birleştirilerek bir çıkış darbesi üretilip üretilmeyeceğine karar verilir. Eğer bir iş yapılacaktır üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilir. Fakat korteks, işin yapılabilmesi için hangi nöron harekete geçirilecekse, sadece ona komut gönderir. Sinir hücrelerinin bilgi göndermesi üç çeşittir.

Her bir gönderme tepkisine sinaps adı verilir. Bir sinir hücresine giren sinapslar, o hücrenin özelliklerine göre işlenir ve tek bir çıktı (sinaps) axondan gönderilir.





**Şekil 2. Yapay Nöron Modeli**

### 2.3. YSA' nın İşlem Elemanı

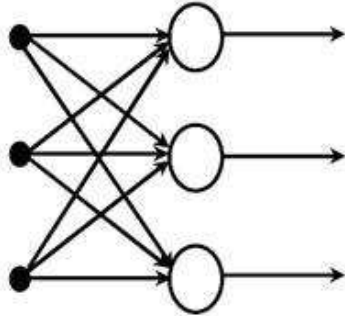
Yapay Sinir Ağlarının Yapısı ve Temel Elemanları Sinir hücreleri bir araya gelerek YSA oluştururlar fakat bunlar rastgele olmaz. Hücreler 3 katman halinde ve her katman içinde paralel olarak bir araya gelerek YSA'yı oluştururlar. Bu katmanlar;

Girdi Katmanı; Dış dünyadan bilgileri alarak ara katmanlara transfer etmekle sorumlu olup bazı ağlar da herhangi bir bilgi işleme olmaz.

Ara Katmanlar; Çıktı katmanına bilgileri işleyerek gönderir. Bilgilerin işlenmesi burada olur ve YSA'da birden fazla ara katman olabilir.

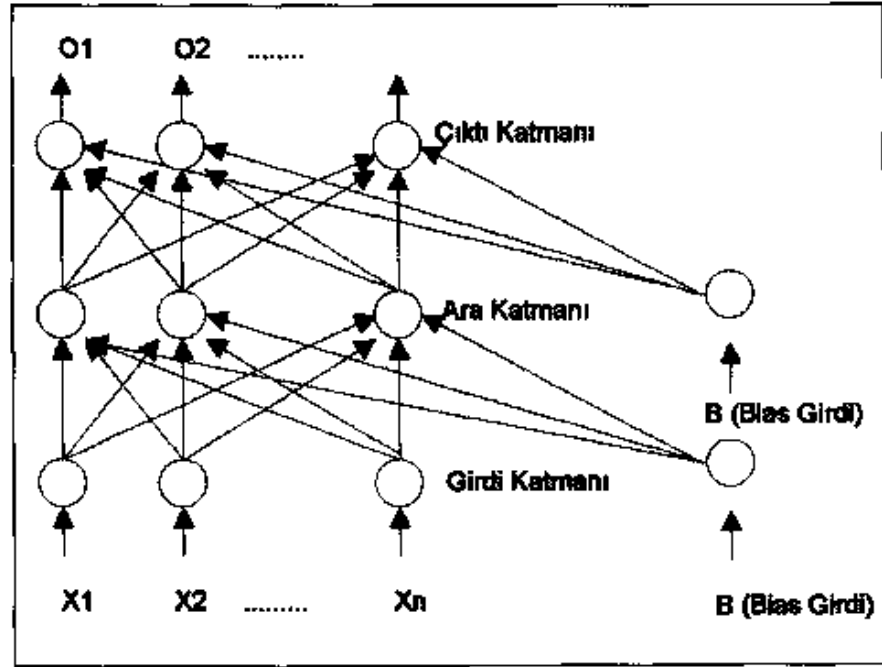
Çıktı Katmanı; Ara katmandan gelen bilgiler proses elemanları ile işlenir. Üretilen çıktı dış dünyaya gönderilir.

Her katmanda birbirine paralel elemanlar söz konusudur. Proses elemanlarını gösteren çizgiler ise ağıın bağlantılarını göstermektedir. Böylece bir yapay sinir ağı oluşur ve bunların ağırlık değerleri öğrenme sırasında belirlenir.



**Şekil 3. Tek katmanlı YSA**

Tek katmanlı ağların çeşitli uygulamalarda sınırlı yetenekler gösterilmiştir. Fakat bu ağlar iki veya daha fazla nöral katmanını kaskat bağlayarak Şekil 3 te Tek katmanlı YSA oluşturulan çok katmanlı ağlara yol açmışlardır. Çok katmanlı ağların çalışması tek katmanlı ağların çalışmasına benzer. Her katmanın çıkışı önceki katmanın çıkışlarından ağırlıklı toplam olarak üretilir. Çok katmanlı YSA Şekil 4'de gösterilmiştir.



Şekil 4. Üç Katmanlı YSA

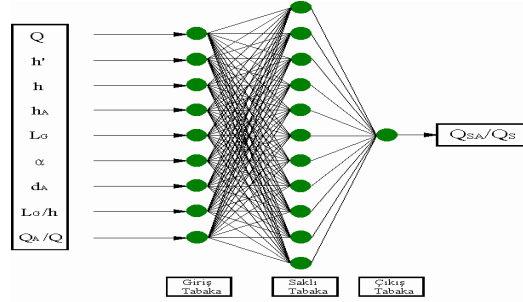
Çok katmanlı ağda her katmanda nöron sayısı farklı olabilir. Araştırmacılar çok katmanlı ağlarının sistematik eğitilmesine imkan sağlayan algoritmalar geliştirmişlerdir.

### 3. YSA'DA EĞİTME VE ÖĞRENME

#### 3.1. Yapay Sinir Ağları Tanımlanması

YSA belirli bir probleme göre programlanmadığı halde o problemi çözmeyi öğrenebilir. Eğitime ve Öğrenme hemen-hemen bütün sinir (nöral) ağların temelidir. Öğrenme, ağdaki nöronların değiştirilmesi ile

değil, nöronlar arasındaki bağlantı ağırlıklarının değiştirilmesi ile sağlanır.



Şekil 5. YSA'da Eğitim

### 3.2. Ağırlık Uzayı

YSA öğrenme işlemi, işlem elemanlarının bağlantı ağırlıkları değiştirilerek sağlanır. Böylece tanımlanan bu ağırlıklar değiştirilerek, öğrenmede iyi bir model kullanılması ve ağırlıkların bu modele göre değiştirilmesi esastır. Basit bir matematiksel model olarak her bir işlem elemanının “n” adet gerçek ağırlığı olduğu düşünülür ve N adet işlem elemanı göz önüne alınırsa;

$$w = (w_{11}, w_{12}, \dots, w_{1n}, w_{21}, w_{22}, \dots, w_{2n}, w_{n1}, w_{n2}, \dots, w_{nn})^T$$

$$w = (w_1^T, w_2^T, w_3^T, \dots, w_N^T)$$

$$w_1, w_2, \dots, w_n :$$

İşlem elemanlarının ağırlık vektörleridir. [ ]<sup>T</sup>, matris transpoze işlemini göstermektedir.

## 4. SINIFLANDIRMA

Sınıflandırma evresinin amacı, örüntüleri özellik uzaylarına göre kendilerine en yakın sınıflara minimum hata ile eşlemektir. Sınıflandırıcının başarımında iyi belirlenmiş özellikler kilit rolü oynarlar. Örüntü sınıflandırıcılarını geleneksel ve akıllı olarak gruplara ayırmak mümkündür. Geleneksel sınıflandırma algoritmaları istatistiksel bir yapı olan Bayes karar teorisi üzerine kuruludur. Bunların dezavantajları, özellik

uzayını sınıflandırma uzayını dönüştürürken, bir gürültünün çıkması ve her bir sınıf için hata kriterinin belli olmamasıdır.

Geleneksel sınıflandırıcılara; çok değişkenli Gauss modelleri, en yakın komşu, maksimum olabilirlik, ikili ağaç sınıflandırıcıları ve Fisher'in doğrusal sınıflandırıcıları örnek olarak verilebilir. Buna karşın akıllı sınıflandırma yapıları genellikle YSA tabanlı olup, günümüzde en yaygın kullanılan ve başarımını ispatlamış çok güçlü sınıflandırıcı türleridirler. Özellikle de genelleme yetenekleri, örüntü tanıma uygulamalarının çok büyük boyutlu verileri açısından önemlidir.

Bu bölümde örüntü tanımada sıkça kullanılan farklı sınıflandırıcı türleri sunulacaktır. Mimarileri, matematiksel altyapıları, eğitime metotları ve test etme prosedürleri değerlendirilecektir. Bunlar:

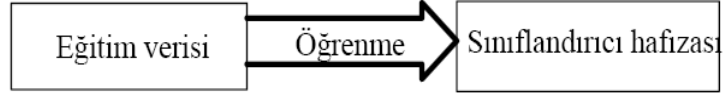
- Parametrik Sınıflandırıcılar:
  - Bayes
  - Maksimum olabilirlik
- Parametrik Sınıflandırıcılar
  - En yakın komşu (k-NN)
  - Parzen Pencereleeri
- Akıllı Sınıflandırıcılar
  - Yapay Sinir Ağı Sınıflandırıcıları
  - Bulanık Sınıflandırıcılar
  - Nöral – Bulanık (ANFIS) Sınıflandırıcılar

#### **4.1. Eğitim ve Test Etme Sınıflandırıcı Süreçleri**

Tüm sınıflandırma algoritmaları bulunan geleneksel Bayes yapısıdır. En yaygın kullanılan öğrenme süreci Şekil 6'da gösterilmiştir. Eğitim verisi sınıflandırıcının hafızası içinde şifrelenir. Önceki bölümlerde de belirtildiği gibi, sınıflandırıcı hafızasını uyarlama parametresi  $\phi$  olacaktır. Aynı şekilde, yaygınca kullanılan test etme sistemi ise Şekil 6'da verilmiştir. Hata oran yaklaşımcısı, doğru sınıflandırmanın basit bir sayacıdır. Sınıflandırıcı yöntemlerinde, sonrasal olasılıklar veya eşdeğer

nicelikler tahmin etmede tek farktır. En yaygın karar süreci, sonrasal olasılıklar üzerine temellenen yalnızca Bayes karar kuralıdır.

Sınıflandırıcılar için şu dikkate alınmalıdır ki; sonrasal olasılıklar doğrudan bir tahmin etme değildir, karar, eşit önceseller için şart yoğunlukları (maksimum olabilirlik) temeli üzerinde varsayılır.



Şekil 6. Eğitme Süreci

#### 4.1.1. Bayes karar teorisi

Bayes teorisi sınıflandırma işlemine bir olasılık problemi gibi yaklaşmaktadır.

Temel Bayes sınıflandırıcısının da kullanılan büyüklükler şu şekilde tanımlanır :

$$\Omega = \{W_1, W_2, \dots, W_s\}$$

$$A = \{\alpha_1, \alpha_2 \dots \alpha_a\}$$

$$\underline{x} \in \mathfrak{R}^d$$

Olasılıklar Bayes kuralıyla birlikte ilişkilendirildiğin de :

$$p(w_j | \underline{x}) = \frac{p(\underline{x} | w_j) p(w_j)}{p(\underline{x})}$$

$$p(\underline{x}) = \sum_{j=1}^s p(\underline{x} | w_j) p(w_j)$$

Bayes karar kuralına göre,

$$p(w_i | \underline{x}) > p(w_j | \underline{x}) \text{ ise Karar } w_i \quad \forall j \neq i.$$

#### 4.1.2. Eğitim

Eğitim her bir kategori için çok olasılıklı gauss yoğunluğundan birinin tahmin edilmesi ile oluşur. Yukarıdaki denklemlerdeki maksimum olabilirlik formülasyonunun sonuçlarını kullanarak her bir kategori için kovaryant matris ve istenilen vektörün hesaplanmak için tahmin etme problemi tanzim edilir. Tanımlanmış veri gereklidir. Bayes yapısında şart yoğunlukları olarak  $p(x|w)$ , tahmin edilen yoğunluklar kullanılır. Her bir kategori için  $N$  eğitim örneklerinin sayısı üzerinde matematiksel olarak yüklenen bir gereksinim vardır.

Yani,

$$N \gg d$$

Burada  $d$  problemin boyutudur.

### 5. OLASILIKSAL SINIR AĞLARI

#### 5.1. Olasılıksal Sinir Ağı

Olasılıksal sinir ağı (PNN, Probabilistic Neural Network) bir Bayes sınıflandırıcısının parametrik olmayan bir gerçekleştirilmesini göz önünde bulundurabilir. Bilinmeyen şart yoğunlukları Parzen pencereleri kullanılarak tahmin edilir ve Bayes karar kuralı en yüksek olasılığın çıkış kategorisini elde etmek için uygulanır. PNN eğitim verisinin temel dağıtımına göre hiç bir varsayımın yapılmadığı serbest model (modelden bağımsız) bir tahmin edici olarak düşünülebilir.

##### 5.1.1. Matematiksel temel

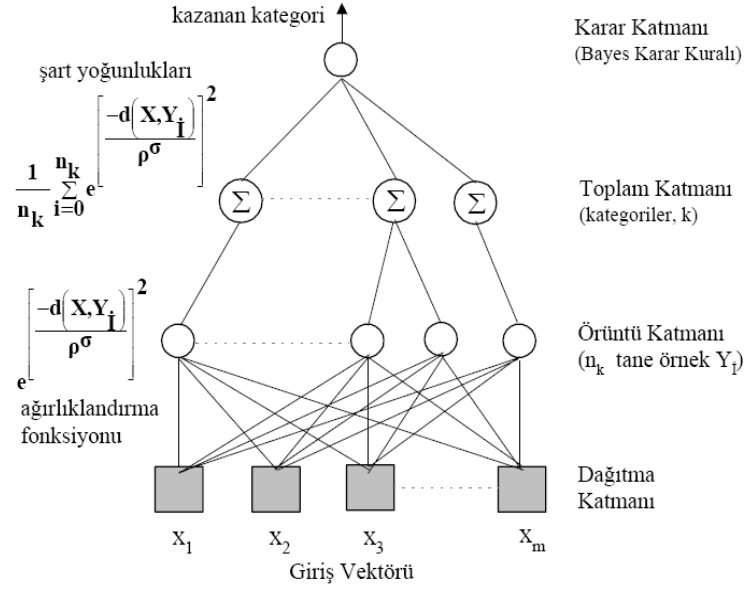
PNN 'nin temelini teşkil eden fikir Parzen pencerelerini kullanarak bilinmeyen şart yoğunluklarını tahmin etmektir. Parzen 'nin tekniği verilen bir kategorinin her bir eğitim vektörü çevresinde  $n$ -boyutlu bir Gauss fonksiyonu merkezlemektir. Verilen kategorinin doğru olasılık yoğunluk fonksiyonlarının (oyf) bir tahmini olarak bu bireysel gauss servislerinin üstüne konan, ölçekleme (bölüntü) ile ayrılır.  $n$  artırılmalar ile,

tahmin edilen oyf yaklaşımları doğru oyf 'yi verir. Matematiksel olarak, tahmin edilen oyf şu şekildedir;

$$f(x) = \frac{1}{N} \sum_{i=0}^{N-1} e^{-\frac{d(x,y_i)^2}{\rho^2}}$$

Burada  $\rho$  düz sabittir,  $d(x, y_i)$  saklanmış  $y_i$  örneği ve  $x$  test örneği arasındaki uzaklıktır,  $\rho$  çok değişkenli örnek(model) vektöründeki bileşenlerin sayıdır ve  $n$  ise verilen kategorideki eğitilen kategorilerin sayısıdır. Bu formülasyonun avantajı şudur ki, birkaç örnek oyf temelini teşkil eden, esnek bir tahmin yapmak için gereklidir.

PNN bu tahmin etme sürecinin paralel bir uygulamasıdır. Dağıtma (yayıma) katmanı bir bağlantı noktası gibi temel hizmet verir ve örüntü katmanının nöronları arasında giriş vektörünün elemanlarını yayar. Bu ikinci katman kategori tarafından organize edilir, her bir kategoriden her biri için eğitim örneği bir nöron ile gerçekleştirilir. Her bir örüntü katmanı birimi test vektörü ve saklanmış eğitim vektörü arasındaki öklit uzaklığını hesaplar. Gauss doğrusalsızlığı bu uzaklık ölçümü ile uygulanır. Toplama katmanındaki her bir birim, verilen bir sınıfın örüntü katman çıkışlarını içerir ve ölçekleme (bölütleme) bu kategoride eğitim örneklerinin sayısının tersi ile toplanır. Toplama katmanının çıkışları tahmin edilen şart yoğunluklarıdır. Son katman maksimum olasılığın kategorisini belirleyen karar katmanıdır.



**Şekil 7. Olasılıksal Sinir Ağı Yapısı**

### 5.1.2. Eğitim

Eğitme, örüntü vektörlerinin örüntü katman birimlerine doğrudan pay edildiği andır. Kodlamanın meydana gelmediği basit durumdur.

### 5.1.3. Test etme

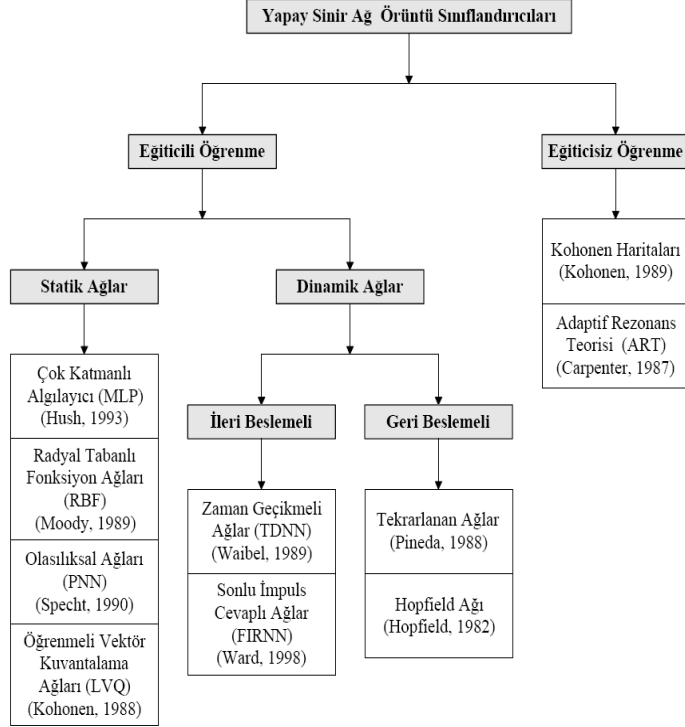
Bir test vektörü PNN' ye verildiğinde, kararda güvenilir bir tahmin kadar iyi olan kazanan kategori çıkışı oluşturur. Bu güvenilir tahmin Bayes karar kuralı kullanılarak, tahmin edilen şart yoğunluklarından hesaplanan sonrasal olasılık ile verilir. Tekrar belirtelim ki denk önceseller, karar maksimum olabilirlik kriteri temelindedir ki, o da şart yoğunluklarıdır.

### 5.1.4. Uyarılama

KNN sınıflandırıcısı ile benzer kolay bir şekilde PNN'de uyarlanır. Aynı önemli noktalar göz önüne alınır ve bu nedenle uyarılama algoritması da benzerdir. Bununla birlikte, madem PNN test örüntüsü ve en yakın saklanan prototip arasındaki uzaklık çıkış değilse, bazı ilave hesaplamalar bu minimum uzaklığı bulmak için gereklidir. Diğer bir deyişle, algoritma



aynıdır.



Şekil 8. Yapay sinir ağ örüntü sınıflandırıcıları

## 6. VERİ MADENCİLİĞİ

### 6.1. Veri Madenciliği

Veri madenciliği üzerine yaptığımız çalışmada perakende satış yapan büyük sayılarda müşteriye sahip marketlerin veritabanlarını ele aldık. Bu veritabanlarını ilişkilendirilmiş veri madenciliği ve karar ağaçları modellerini kullanarak analiz ettik.

İlk kısımda ilişkilendirilmiş veri madenciliği , ikinci kısımda ise karar ağaçları mekanizması uygulandı.Karar ağaçları mekanizmasında yazılan algoritmanın eğitildiği ve daha sonra tahmin mekanizmasını işlediği görülecektir.

### 6.1.1. Veri Madenciliği Nedir?

Veri madenciliği sadece elektronik ticaret değil, her türlü veri uygulamalarında kullanılabilir tekniklere verilen genel addır. Dünya üzerindeki veriler her sene 2ye katlanmaktadır. Yüksek kapasiteli bilgisayarlar sayesinde veri saklamak daha kolay ve daha ucuz olmuştur. Birbirlerine rakip olan firmaların çalışmaları, veri madenciliğinin önemini ortaya çıkarmıştır. Araştırmacıların geniş ve dağınık veri kümeleri üzerinde yapmış oldukları çalışmalar son derece önemlidir.

Geniş veri kümelerinden düzensizlikleri, ilişkileri, desenleri ortaya çıkarmakta kullanılır. Bizim örneğimizde de olduğu gibi alışveriş yapan müşterilerin alışkanlıkları gibi karar verme mekanizmaları için önemli bulgular elde edilebilir. Geniş hacimli ve çok boyutlu veri madenciliği için yeni algoritmalar ve sistemlerin geliştirilmesi Eldeki verilerden, üstü kapalı, çok net olmayan, önceden bilinmeyen potansiyel olarak kullanışlı verinin çıkarılmasıdır. Geleneksel istatistikten farkları burada ortaya çıkar. Burada bize gereken; veriyi alabileceğimiz bir veritabanıdır. Biz uygulamamızda Access veritabanı kullandık.

### 6.1.2. Veri madenciliğinin sınırları

- 1- Sınırlı bilgi
- 2- Eksik değerler ve gürültü(hatalar) : Gürültünün de bir sebebi vardır.
- 1- Gelişen bellek ve işlem hızı sayesinde bir kaç yıl önce madencilik yapılamayan veriler üzerinde çalışmayı mümkün hale getirmiştir. Yani insanların geliştirdiği vasat performansa sahip algoritmalar yüksek kapasiteli bilgisayarlar sayesinde fark edilmez olmuştur.

Ticari Amaç: Kar Kar'ın Yolu: Müşterileri ve müşteri faaliyetlerini daha iyi anlamaktır.

Örnek olarak büyük bir süpermarketin en basit fatura kayıtları incelendiğinde traş bıçağı alan müşterilerin %56 sının kalem pilde aldığı

ortaya çıkmıştır. Buna dayanarak firma traş bıçağı ve kalem pil reyonlarını bir araya getirmek suretiyle kalem pil satışlarını %14 arttırmıştır. Buna benzer örnekler karşısında stratejilerimizi geliştirebiliriz. Bizim yaptığımız çalışma veri madenciliği altında keşif (discovery) olmuştur.

Küçük uygulama projemde veritabanı sistemleri ile veri madenciliği disiplinlerini birleştirmeye çalıştım. Perakende veri madenciliği bizlere şunları sunmaktadır. Müşteri tarafından bakacak olursak;

- 1) Müşteri alış davranışı belirlenmesi
- 2) Müşteri hatırlama ve memnuniyetinin sağlanması

Projede bu iki hususa dikkat edilmiştir.

- En önemli sorun ise, ölçek sorunudur. Veri madenciliği alanındaki birçok algoritma, orta büyüklükte veritabanlarıyla iyi çalışır, ancak çok büyük veritabanlarıyla zorluk çeker. Çünkü büyük veritabanlarında öznitelik ve nesne sayısı çok fazladır.

(Örneğin 200 milyon telefon görüşmesinden oluşan bir veritabanını yönetebilecek bir algoritma yoktur.

- Veri madenciliği algoritmalarının çoğu da büyük veritabanlarından veri gruplarının seçilmesinden sonra, bu azaltılmış veri gruplarını yönetebilir.



**Şekil 9. Veri Tablosu**

## 6.2. Veri Madenciliğinde Önemli Sorunlar

### 6.2.1. Veritabanlarının büyük boyutları

- Sırf veritabanlarının boyutu yüzünden, veri madenciliği yöntemlerinden herhangi birinin ham veriyle başarılı olma olasılığı yoktur. Veri madenciliği yöntemleri, bu şekilde elde edilen sonuçların tüm veritabanını temsil edebileceğini umarak, veritabanından bir örneğin çıkarılmasını gerektirebilir.

Bilgi bulma süreci aşamasında, kullanılacak veri grubu, veri madenciliği yazılımı tarafından çözümlene için hazırlanır. Bunun için;

- Veri madenciliği yönteminde girdi olarak, veritabanındaki hangi bilgilerin kullanılacağına karar verilmesi gerekir
- Sonra, seçilen veriler temizlenir.
- Veri madenciliğinde birçok farklı yöntem ve teknik kullanılır. Bu teknikler, veriyi kümeleme, sınıflandırma, kavramların tanımlanması, eksik değerleri tahmin vb amaçlarla kullanılır.

## 6.3. İlişki Analizi

İlişki analizi, belirli bir veri kümesinde yüksek sıklıkta birlikte görülen öznitelik değerlerine ait ilişki kuralların keşfidir. Market-Basket analizi ve transaction veri analizinde sıkça kullanılır.

İlişki ya da birliktelik analizi, bir veri kümesinde kendiliğinden, sıklıkla gerçekleşen, birlikte ya da aynı süre içinde alınma, yapılma, oluşma gibi etkileri keşfetme temeline dayanır. Bu yöntem bankacılık işlemlerinin analizinde ya da sepet analizi tekniğinde yaygın olarak kullanılır. Sepet analizi, bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın alma eğiliminde olduğunun

belirlenmesiyle müşteriye daha fazla ürün satılması yollarından biridir. Sepet Analizi ile örneğin müşteriler bira satın aldığında %75 ihtimalle cips de alırlar şeklinde bir ilişki ortaya çıkarılabilir. Bunun sonucunda bira ile cips yan yana raflara yerleştirilebilir veya bira alanlar cips aldığında cips fiyatında indirim yapılacak şekilde kampanyalar oluşturularak satışlar artırılabilir.

Daha resmi olarak, İlişkisel kurallar,  $X \Rightarrow Y$  şeklinde gösterilir.

**Örnek:** Veriler AllElectronics verilerinde, veri madenciliği sistemi şu şekildeki bir ilişkisel kuralı bulabilir:

$\text{Age}(X, "20 \dots 29") \Delta \text{income}(X, "20K \dots 29K") \Rightarrow \text{buys}(X, "CD \text{ player}')$   
[support= 2%, confidence=60%]

Burada X, müşteriye temsil eden bir değişkendir. Kural, all electronics müşterilerinden çalışanların % 2 si, 20...29 yaşları arasında 20K..29K gelire CD player almışlardır. Bu yaşta ve bu gelirdeki insanların CD player alma olasılıkları %60 tır.

Buradaki ilişkisel kural birden fazla attribute arasındadır. Çokboyutlu veri tabanları düşünülerek terminolojide değişiklik yapıldığında, her attribute'un bir boyut olarak düşünüldüğünde, yukarıdaki kural bir çok boyutlu ilişkisel kural olarak kabul edilebilir.

AllElectronics verilerini satan bir market düşünelim. Hangi ürünlerin sıklıkla birlikte satın alındıkları bulunmak istenmektedir. Kuralın örneği:

$\text{Contains}(T, "computer") \Rightarrow \text{contains}(T, "software')$   
[support = 1%, confidence = %50]

Bu kural bize şunları anlatmaktadır. Transaction T, eğer bilgisayar içeriyorsa %50 ihtimal ile yazılım da içerir. Ve %1 oranında bütün tamsactionlar iki ürünü de birlikte içerirler. Bu kural daha basit bir şekilde de ifade edilebilir:  $\text{Computer}=\text{software}[1\%,50\%]$

### 6.3.1. Temel Kavramlar

- Frequent patterns : Database içinde sık sık bir arada görünen verilerdir.
- Itemset  $X=\{x_1, \dots, x_k\}$ : Database içindeki verilerden oluşan settir  $\{I\}$

Itemset  
= $\{A,B,C,D,E,F\}$

Transaction-id	Items bought
15	A, B, C
30	A, C
45	A, D
60	B, E, F

- $X \rightarrow Y$  Kural yapısı:

■  $\text{buys}(x, \text{"SQLServer"}) \wedge \text{buys}(x, \text{"Book"}) \rightarrow \text{buys}(x, \text{"DBMiner"})$

- Support ve confidence değerleri:

Support : İşlemlere başlamadan kuralların geçerliliğini belirlemek üzere kullanıcının belirlediği bir değerdir.  $X \rightarrow Y$  X 'in ve Y 'nin bir arada tüm database içinde bulunma olasılığıdır.

$$(X \text{ sayısı} + Y \text{ sayısı}) / \text{Toplam İşlem sayısı}$$

Confidence : Şarta bağlı olasılığı olarakta geçer.  $X \rightarrow Y$  X ve Y nin bir arada X değerini içeren transactionların içinde bulunma olasılığıdır.

$(X \text{ sayısı} + Y \text{ sayısı}) / X$ 'i içeren işlem sayısı

Yukarıdaki tablo üzerinden incelersek.

$A \rightarrow C$  Support değeri %50

Confidence değeri %66.7

#### 6.4. Sınıflandırma ve Tahmin Yürütme

Sınıflandırma, sınıfı belirlenmemiş verilerin hangi sınıfa dahil olacaklarını tahmin etmeye elde mevcut olan verileri farklı sınıflara koyabilmek için bazı fonksiyonlar içerir. Türetilmiş olan yöntemler, bir eğitim verisinin analizine dayanmaktadır.

Model nasıl sunulur. Çok değişik şekillerde olabilir. (IF-THEN) kuralları, karar ağaçları, matematiksel formüller ya da yapay sinir ağları gibi. Karar ağacı, ağaç yapısında olan bir akış şeması şeklindedir. Her node, bir attribute üzerindeki test işlemini temsil eder. Her dallanma test işleminin sonucunu temsil eder, ve sonuç olarak ağaç sınıflar ile son bulur. Karar ağaçları kolayca sınıflandırma kurallarına dönüştürülebilirler.

Yapay sinir ağlarında (*artificial neural networks*) amaç fonksiyon birbirine bağlı basit işlemci ünitelerinden oluşan bir ağ üzerine dağıtılmıştır. Yapay sinir ağlarında kullanılan öğrenme algoritmaları veriden üniteler arasındaki bağlantı ağırlıklarını hesaplar.

Yapay sinir ağları, sonuçları bilinen belirli veri kümeleri üzerinde algoritmaları çalıştırarak eğitilirler. Bunun sonucunda yapay sinir ağının içindeki ağırlık belirlenir. Bu ağırlıklar yeni gelen verilerin işlenmesinde kullanılır. Karmaşık veri tabanları üzerinde daha iyi sonuçlar üretebilirler. Ama varılan sonucun olabileceklerin arasında en iyisi olduğunun garantisi yoktur.

Ekonomi, tıp, ticari kurumlar için önemli müşterilerin belirlenmesi, kredi kartı sahteciliklerinin tespiti gibi alanlarda uygulanmaktadır. Tahmin edilen değerler sayısal ya da kategorik değerler olabilirler. Ama sınıflandırma işlemi Gerçekleştirilirken üzerinde işlem yapılacak olan eğitim verisi değerleri sayısal değerler olmalıdırlar.

### **6.5. Olasılıksal Sinir Ağları Çalışması**

Yaptığımız bu çalışmanın amacı müşterilerin market alışverişlerinde aldıkları ürünlerde benzerliklerini bulup, bu kişileri yas meslek ve oturdukları semtlere göre gruplandırmaktır. Daha sonra yeni veriler girdiğimizde, yani örneğin Bahçeşehir' de ikamet eden yaşı 18 olan ve öğrenci olan bir kişi örneğin Cola ürününün hangi markasını tercih edecektir, buna karar veren bir program ortaya çıkarmaktır.

Bunun karar vermek amacıyla öncelikle elimizde hazır bulunan bir test verisine ihtiyacımız olacaktır. Bu test verisi 4 sütun ve bizim belirleyeceğimiz müşteri sayısından meydana gelecektir. Biz burara 4 sütun ve örnek olarak 1000 müşteri aldık. Müşteri sayımız yani örnek bilgimiz ne kadar çok olursa o kadar kesine yakın sonuçlar elde ederiz.

Öyleyse bu verileri nerede tutacağız, bu verileri not defterimizde saklayabiliriz. Daha sonra bu verileri program yardımıyla buradan okuyup üzerinde uygulama gerçekleştirebiliriz.

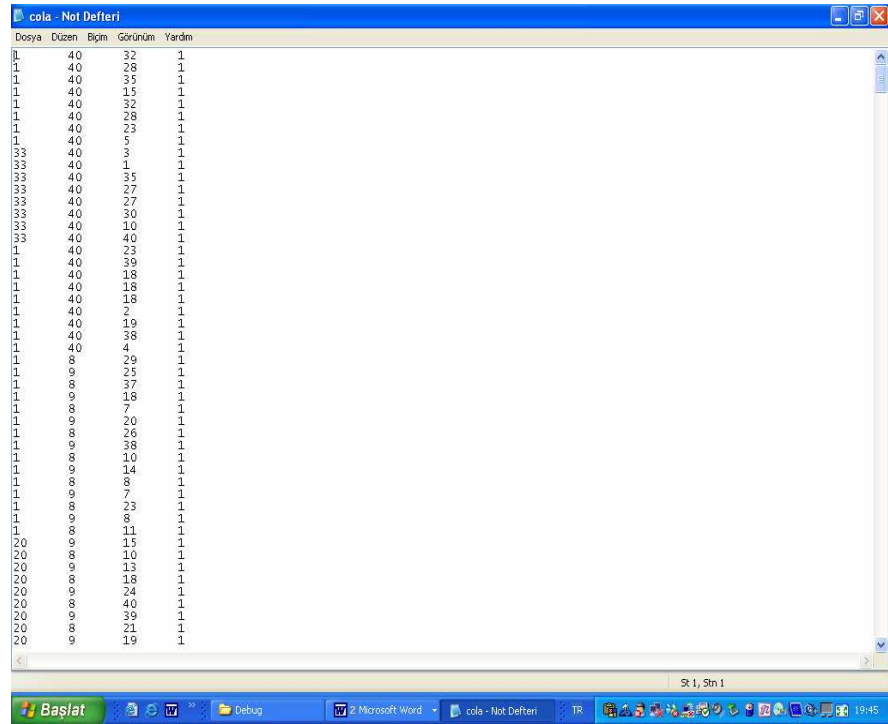
Verilerimizi okuyabilmek için ileride göreceğimiz dizi tanımlamamızı yapacağız. Verilerde ilk sütunu Yaş özelliği alacaktır. İkinci sütunu ise meslek , üçüncü sütunumuz semt ve son sütunumuz ise seçilen ürün grubu olarak alacağız. Seçilen ürünleri sabitleyip 4 'e ayırdık. Yani diyelim ki Cola ürününün dört markası var. 1 numaralı marka pepsi, iki numaralı ürünümüz coca cola, üç numaralı ürün grubumuzca turca ve son grubumuzu ise uludağ kola olarak nitelendirebiliriz.

Buradaki amaç aslında açıktır. Yani örneğin yapılan testlerde coca cola ürününü alanların genç ve pahalı bir semtte oturduğunu gördük.



### 6.5.1. Proje Kapsamında Yapılanlar

Öncelikle cola.txt dosyamız hazırlanmıştır.Burada müşteri sayısı 1000 alınacak şekilde veriler girilmiştir.Buradaki dizimiz 4 kolon ve 1000 satırdan oluşacaktır. 1.sütun yas, ikinci sütun meslek,üçüncü sütun semt ve 4.sütun ise müşterilerine tercihinine göre değişen Kola ürünlerinin markalarını ifade etmektedir.1 numaralı ürün Coca cola , iki numara ile ifade edilen ürün Pepsi Cola , üç numara ile ifade edilen ürün cola turca ve 4 numara ile ifade edilen ürün ise Uludağ Kolayı ifade ediyor.



Dosya	Düzen	Bölüm	Görünüm	Yardım
1	40	32	1	1
1	40	28	1	1
1	40	35	1	1
1	40	15	1	1
1	40	32	1	1
1	40	28	1	1
1	40	23	1	1
1	40	5	1	1
33	40	3	1	1
33	40	1	1	1
33	40	35	1	1
33	40	27	1	1
33	40	27	1	1
33	40	30	1	1
33	40	10	1	1
33	40	40	1	1
33	40	23	1	1
1	40	39	1	1
1	40	18	1	1
1	40	18	1	1
1	40	18	1	1
1	40	2	1	1
1	40	19	1	1
1	40	38	1	1
1	40	4	1	1
1	8	29	1	1
1	9	25	1	1
1	8	37	1	1
1	9	18	1	1
1	8	7	1	1
1	9	20	1	1
1	8	26	1	1
1	9	38	1	1
1	8	10	1	1
1	9	14	1	1
1	8	8	1	1
1	9	7	1	1
1	8	23	1	1
1	9	8	1	1
1	8	11	1	1
20	9	15	1	1
20	8	10	1	1
20	9	13	1	1
20	8	18	1	1
20	9	24	1	1
20	8	40	1	1
20	9	39	1	1
20	8	21	1	1
20	9	19	1	1

Şekil 10. Yaş, meslek, semt ve Ürün Matrisi

Yaş, meslek ve semt ler 1-40 arası rakam değerleri verilmiştir. Örneğin yaş grubunu ele alacak olursak;

Aşağıda ki tablodan görüldüğü üzere 14-15 yaş grubu 1 numarasını temsil ediyor ve tablonun sonuna gelindiğinde 92-93 yaş grubu 40 numara ile ifade edilmektedir. Verileri bu şekilde girmemizin nedeni karmaşıklığa engel olmaktır.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															
3															
4		yaş													
5		14	15	1		48	49	18							
6		16	17	2		50	51	19							
7		18	19	3		52	53	20							
8		20	21	4		54	55	21							
9		22	23	5		56	57	22							
10		24	25	6		58	59	23							
11		26	27	7		60	61	24							
12		28	29	8		62	63	25							
13		30	31	9		64	65	26							
14		32	33	10		66	67	27							
15		34	35	11		68	69	28							
16		36	37	12		70	71	29							
17		38	39	13		72	73	30							
18		40	41	14		74	75	31							
19		42	43	15		76	77	32							
20		44	45	16		78	79	33							
21		46	47	17		80	81	34							
22						82	83	35							
23						84	85	36							
24						86	87	37							
25						88	89	38							
26						90	91	39							
27						92	93	40							
28															
29															
30															
31															

**Şekil 11. Yaş Aralıklarının Sayısal Eşleştirilmesi**

Meslek grubuna bakacak olursak yine burada 1- 40 rakamlar arası değerler almışlardır.

Aşağıdaki tabloda görüldüğü üzere burada 1 numara Akademik Kariyer yapan kişileri 40 numara ise öğrenci olanlar listelemektedir.

meslek	Insan Kaynakları	20
Akademik	İnşaat	21
Bankacılık/Finans	İthalat/İhracat	22
Bilgisayar/Bilişim	Kalite / AR-GE	23
Çocuk Gelişimi ve Eğitimi	Kimya	24
Danışmanlık	Kozmetik	25
Denetim	Mağazacılık	26
Depo/Antrepo	Medya	27
Eğitim	Mimarlık	28
Eğlence	Muhasebe	29
Elektrik/Elektronik	Mühendislik	30
Enerji	Otomotiv	31
Gayrimenkul	Otelcilik	32
Gıda	Restaurant İşletmecisi	33
Giyim Sanayi	Sağlık/Tıp	34
Güvenlik	Sigortacılık	35
Halkla İlişkiler	Spor	36
Hukuk	Telekomünikasyon	37
İdari İşler/Sekreteryası	Ulaştırma	38
İlaç Endüstrisi	emekli öğrenci	39
		40

## Şekil 12. Meslek Aralıklarının Sayısal Eşleştirilmesi

Semt lere bakılacak olursa yine 1- 40 arası rakam değerleri ile kodlama yapılmıştır.

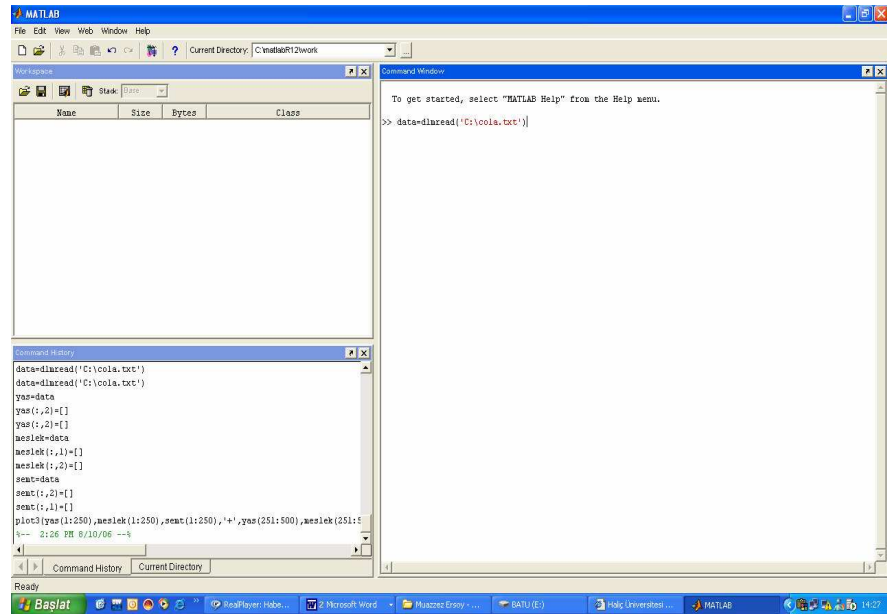
semt	esenerler	17
Avcılar	eyüp	18
Bahçeşehir	fatih	19
Esenkent	gaziosmanpaşa	20
Küçükçekmece	güngören	21
Sefaköy	kağıthane	22
Buyukçekmece	zeytinburnu	23
Bakırköy	Kadıköy	24
Etiler	erenköy	25
Esenyurt	maltepe	26
yeşilköy	k.bakkalköy	27
florya	tuzla	28
bayrampaşa	kartal	29
beşiktaş	ümraniye	30
büyükdüğü	yakacık	31
baykoz	sultanbeyli	32
eminönü	acıbadem	33
	üşkudar	34
	reşadiye	35
	kurtköy	36
	samandıra	37
	yukarı dudullu	38
	anadoluhisari	39
	nva	40

## Şekil 13. Semt Aralıklarının Sayısal Eşleştirilmesi

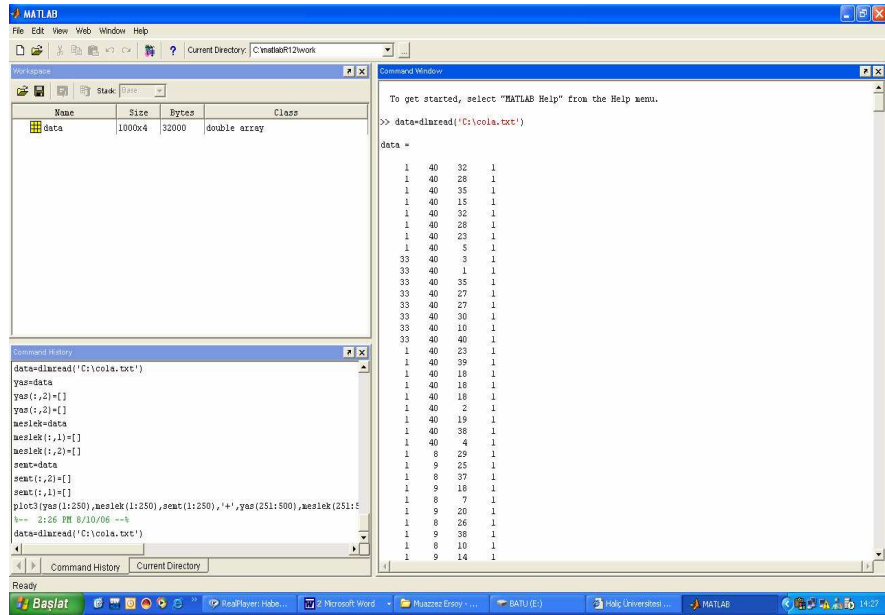
Yukarıdaki tablodan görüldüğü üzere 1 numara Avcılar, 2 numara Bahçeşehir ve 40 numara Riva yı temsil etmektedir.

Proje kapsamında MatLab 6.5 ile müşterilerin tercih ettikleri ürünleri görebilmek için tablolar oluşturulmuştur. Zaten 1- 40 arası değerleri kodlamamızın sebebi de tablo oluşturduğunda düzgün bir görüntü elde edilmek istenmesindedir. Öncelikle Kola ürünü için elde edilen veriler cola.txt dosyasına kayıt edilir. Daha sonra

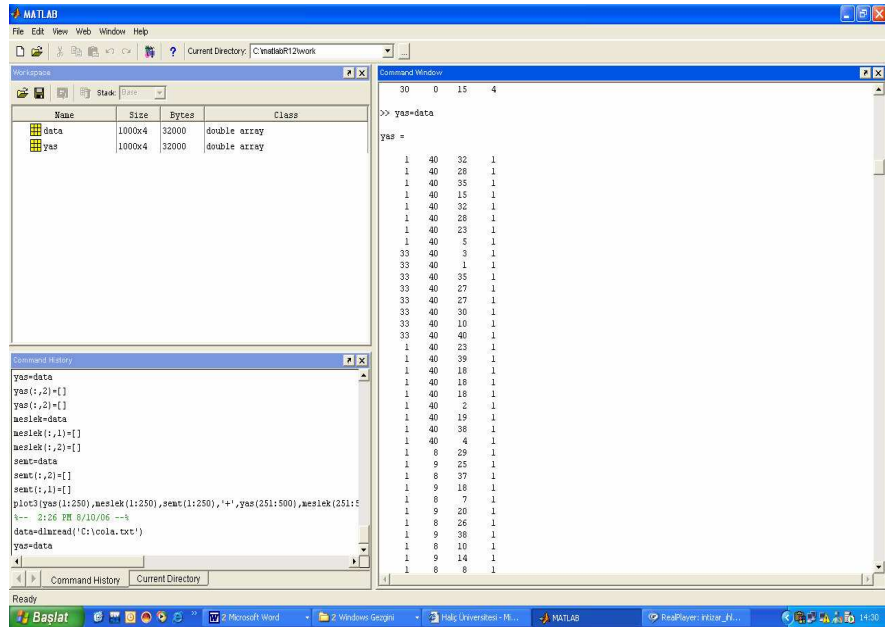
- `data=dlmread('C:\cola.txt')` komutu ile dosyamızdaki veriler matris şeklinde sol tarafta görüldüğü gibi  $1000 \times 4$  lük olarak oluşturulur. Sağ tarafta ise dosyamızdaki veriler aynen yansıtılır.



Şekil 14. MathLab'de Not Defterinden Matrislerin Alınması



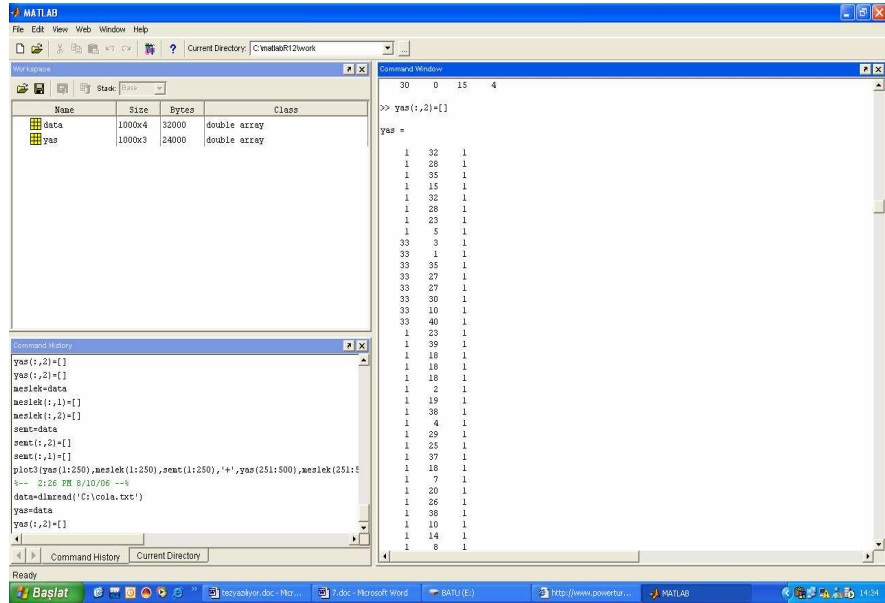
**Şekil 15. MathLab’de Not Defterinden Matrislerin Alınmış Gösterimi**  
Mathlab de bu arayüzün sol alt kısmında ise yazdığımız komutların history si tutulmaktadır.



**Şekil 16. MathLab’de Not Defterinden Yaş Matrisinin Alınması**

Yukarıdaki arayüzde görüldüğü üzere yas matris bilgisini tutabilmek için öncelikle komutu yazıyoruz.

Yas=data dediğimizde biraz önce oluşturduğumuz datalar Yas matrisine aynı şekilde aktarılır. Yani Yas matrisi 1000\*4 lük veri tutmaktadır. Ancak burada bizim ihtiyacımız olan sadece yas bilgisi ve seçilen ürün bilgisidir.Bunu yaptıktan sonra matrisimiz 1000\*2 lik olacaktır.

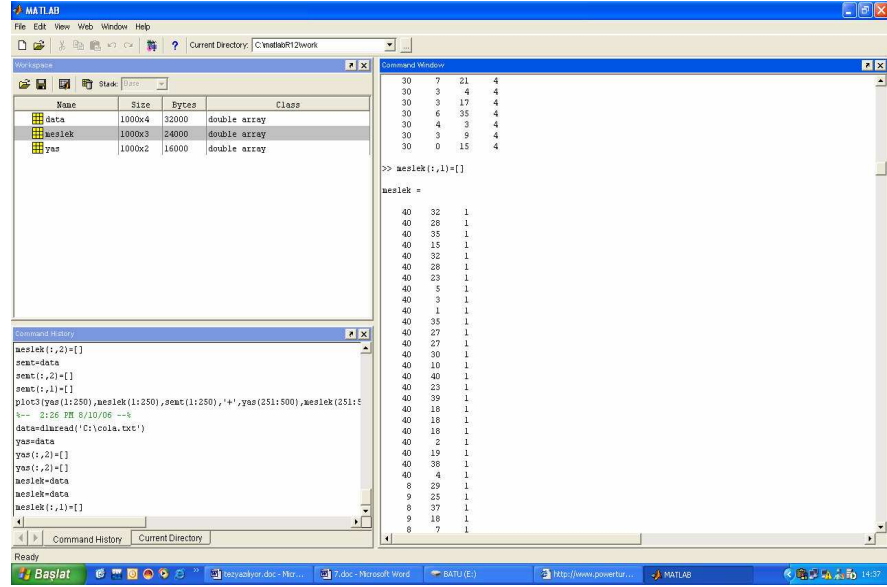


**Şekil 17. MathLab’de Not Defterinden Yaş Matrisinin 2\*2 lik Gösterimi**

Yukarıda komut satırında görüldüğü üzere amacımız 2.ve 3.sütunları matrisimizden çıkarmak olacaktır. Bu nedenle yas(:,2)=[] komutunu yazıyoruz ve 2.sütunu silmiş oluyoruz.



Yukarıdaki komut satırında meslek=veri diyerek veri da yer alan 1000\*4 lük bilgiyi meslek matrisine aktarmış oluyoruz. Burada bizim işimize yarayacak olan sütunlar sadece meslek ve ürün grubu sütunları olmalıdır.

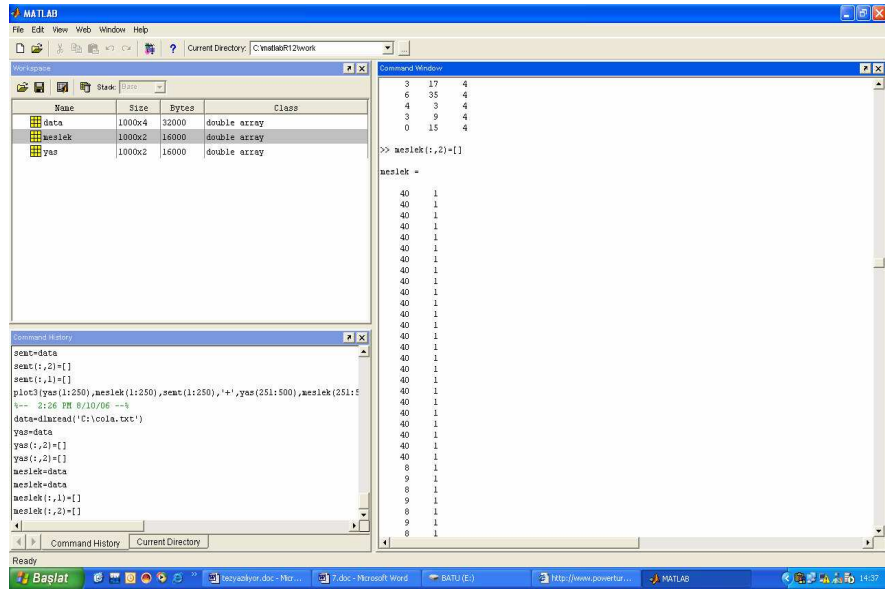


Şekil 20. MathLab'de Not Defterinden Meslek Matrisinin 2\*2 lik Gösterimi

Bu nedenle

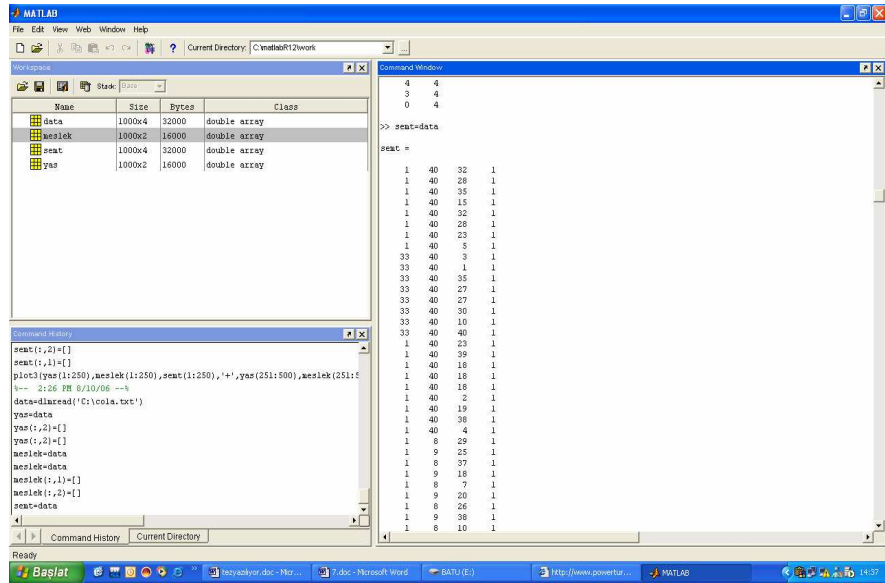
Meslek(:,1)=[] diyerek 1.sütunu silmiş olacağız. Burada bizim ihtiyacımız olan 2.sütun ve ürün grubu olan 4.sütundur.





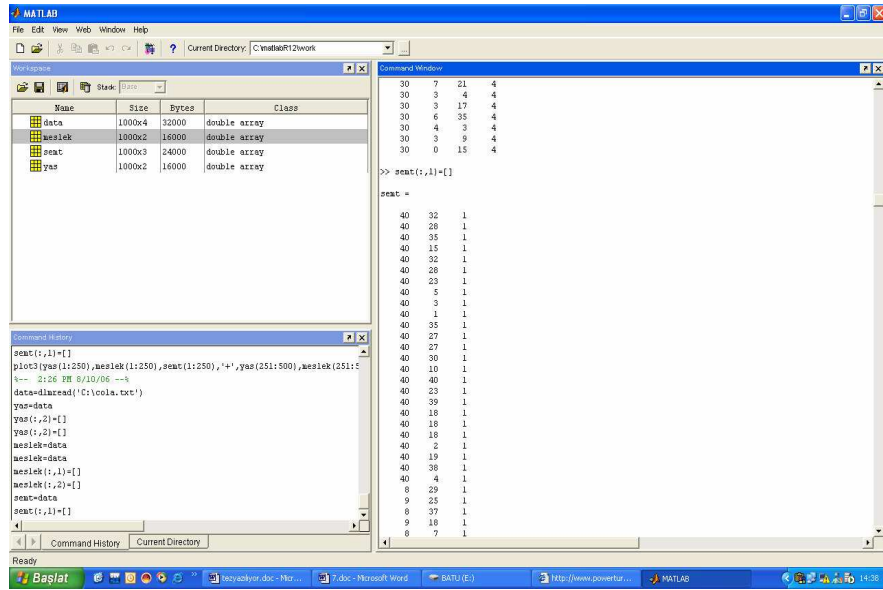
**Şekil 21. MathLab’de Not Defterinden Meslek Matrisinin 2\*2 lik Gösterimi**

Yukarıda Meslek(:,2)=[] diyerek 2.sütunu da siliyoruz.



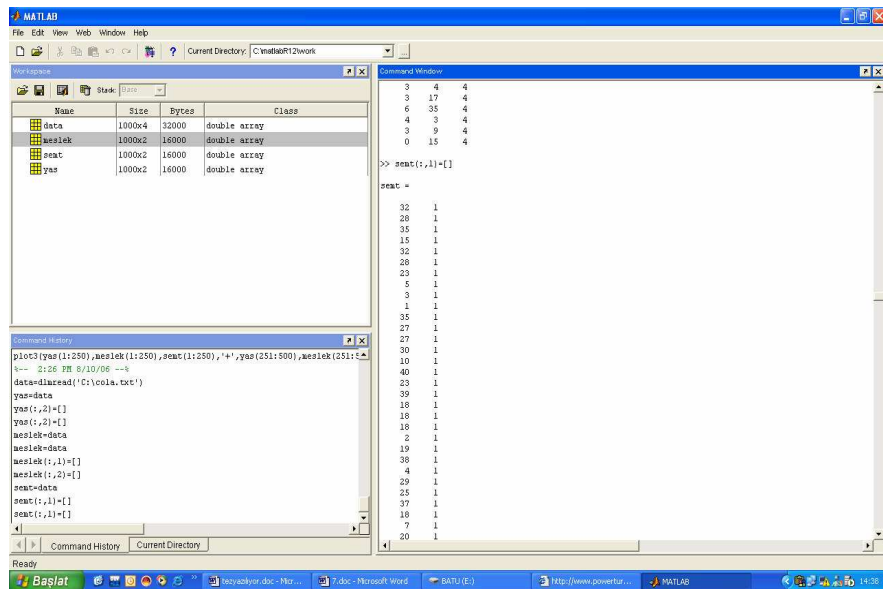
**Şekil 22. MathLab’de Not Defterinden Semt Matrisinin Alınması**

Yukarıda komut satırında ihtiyacımız olan semt özellikleri matrisini sağlamak amacıyla semt=data diyoruz ve 1000\*4 lük matrisimiz oluşmuş oluyor.



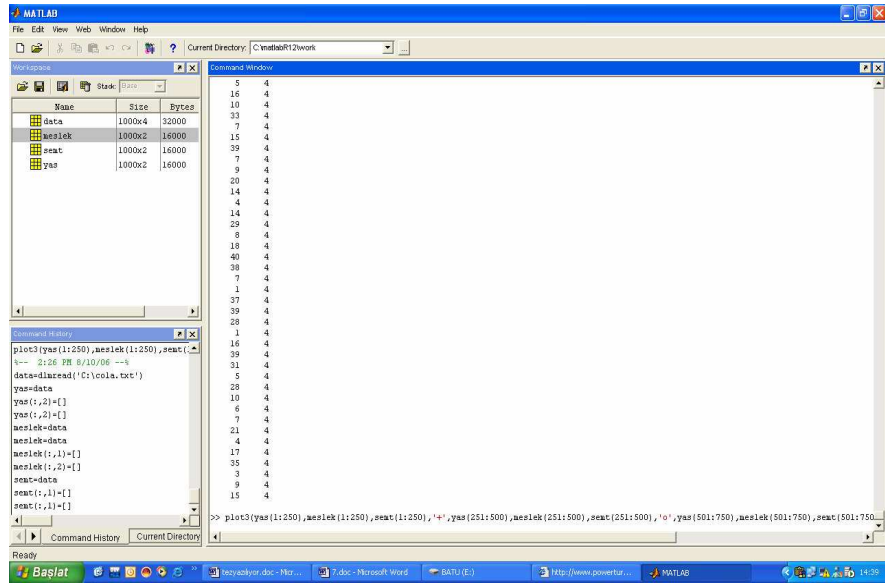
Şekil 23. MathLab’de Not Defterinden Semt Matrisinin Oluşturulması

Burada amacımız 3.sütun ve 4.sütun olan ürün grubunu bırakıp geriye kalan sütunları matristen çıkarmaktır.  $semt(:,1)=[ ]$  diyerek 1.sütünü silmiş oluyoruz.



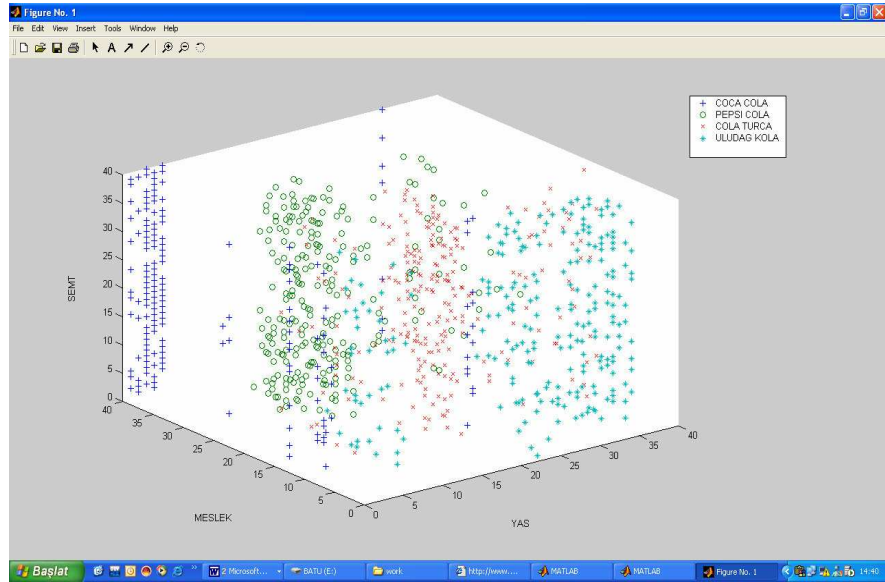
Şekil 24. MathLab’de Not Defterinden Semt Matrisinin 2\*2 lik Gösterimi

Daha sonra  $semt(:,1)=[ ]$  diyerek 1.sütünü tekrar siliyoruz ve elimizde 1000\*2 lik bir matris oluşmuş oluyor.



**Şekil 25. MathLab’de Not Defterinden Semt Matrisinin 2\*2 lik Gösterimi**

Bu oluşan 4 matris ara yüzün sol üstünde yer almaktadır. 1000\*4 data y1, 1000\*2 meslek’i,1000\*2 semt i ve son olarak 1000\*2 ide yas ı temsil etmektedir.

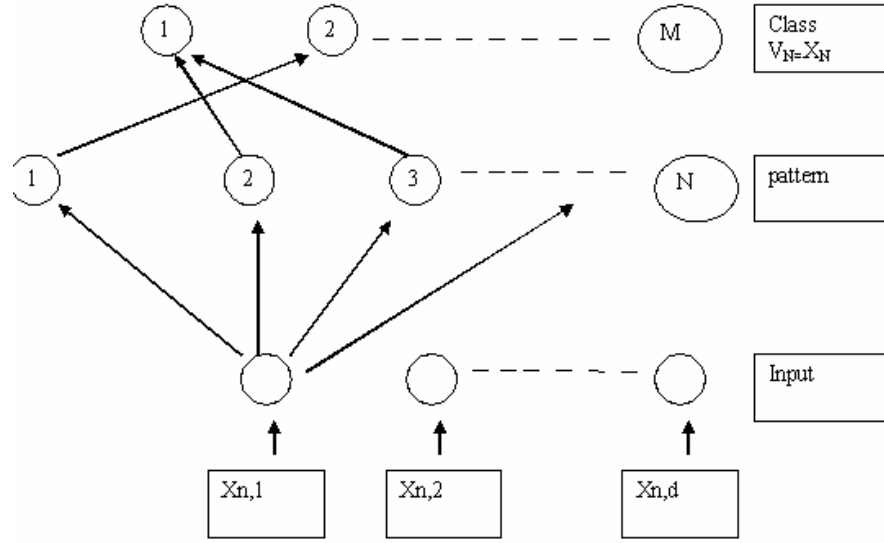


**Şekil 26. MathLab’de Semt,Yaş ve Meslek Özellikleri Ele Alınarak Çizilen 3 Boyutlu Grafik**

Matrislerin oluşturulmasından sonra 3 boyutlu grafik çizimini oluşturacağız. Bu grafik semt, meslek ve yaş doğrularından meydana gelecektir. Yani y eksenimiz Semt, x eksenimiz Meslek ve Z eksenimiz Yaş özelliklerini temsil edecektir. Grafiğimizi çizmek için öncelikle örnek sayımızın yani müşteri sayımızın 1000 olduğunu hatırlayalım. Öyleyse 1-250 müşteri arasına aynı ürün grubunu yani 1 numaralı ürünü, 251-500 arasına 2 numaralı ürünü, 501-750 arasına 3 numaralı ürünü ve 751-1000 arasına ise 4 numaralı ürünü veririz. Bu bölünmenin amacı müşteri ve alınan ürünlerin düzenli bir dağılım göstermesi ve bu dağılımdan daha sonra Probabilistic Neural Network(Olasılığa Dayalı Yapay Sinir Ağları) yardımı ile verileri eğitmek ve sonrasında doğru sonuçlar elde etmektir. Burada Coca Cola artı + işareti ile , Pepsi Cola o işareti ile, Cola Turca x işareti ile ve Uludağ Kola \* işareti ile temsil edilmektedir. Oluşan grafiğin 3 boyuttan fazla olması insan beynini oldukça yoracağından 3 boyut uygun görülmüştür. Görüldüğü üzere örneğin Coca Cola ürünün satın alanlar her semte yayılmıştır, Coca cola ürünün alan meslek grubu ise 35-40 arasındaki grupta toplanmıştır. Yaş a bakılacak olursa 0-5 yaş grubu bu ürünü tercih etmiştir. Artı işareti ile temsil edilen Coca Cola 20-25 ile 5-15 arası meslek ve 0-25 arası yaş grubundan da tercih edilmiştir. Ancak bizim için önemli olan tercih grubunun yoğun olduğu kısımlardır.

#### **6.6. Proje Algoritma Tanımlanması**

Olasılıksal Sinir Ağı (Probabilistic Neural Network - PNN) Bayes-Parzen kestiriciler olarak da bilinir.



**Şekil 27. Olasılıksal Sinir Ağı Yapısı**

Buradaki en alt kısım girişleri ifade etmektedir. Orta katman ise müşteri sayımızı, son olarak çıkış katmanı yani en üst katman ise ürün grupları ifade etmektedir.

Programın girişinde verileri girerken öncelikle normalize etmemiz gerekecektir. Çünkü normalize ettiğimizde 1 veya 1'e çok yakın sonuçlar elde edebiliyoruz. Bu sayede doğru bulma payımız daha da artıyor.

Giriş değerlerimizi dizi şeklinde alıyoruz. Bizim uygulamamızda 3 değer olduğundan, yani Yaş meslek ve semt olacak şekilde 1000 müşterimiz yani örneğimiz olsun.

$$\bar{x}_n = \begin{bmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,d} \end{bmatrix}$$

Giriş değerlerine aynı zamanda parametrelerimiz olarak ta adlandırabiliriz.

Mesela giriş değerlerimiz 12,9,11 olsun, yani yaş 36-37 aralığında, mesleği Eğlence Sektöründe, semt i ise Florya olsun. Daha önce anlattığım

üzere normalize edebilmek için Vektörümüzün elemanlarını 12,9,11 alıyoruz.

$$12^2+9^2+11^2=144+81+121=346 \text{ sonucunu buluyoruz.}$$

Daha sonra normalize edebilmek için her birine teker teker bu sonuca bölüyoruz.

$$12/(12^2+9^2+11^2)^{1/2}=0,645$$

$$9/(12^2+9^2+11^2)^{1/2}=0,483$$

$$11/(12^2+9^2+11^2)^{1/2}=0,591$$

Bu çıkan değerlerin karesini alıp toplarsak 1'e yakın bir değer elde edeceğiz. Böylece normalize etmiş oluyoruz.

$$0,645^2+0,483^2+0,591^2=$$

$$0,4160+0,2332+0,3492=0,9984$$

Görüldüğü üzere 0,9984 rakamı 1'E oldukça yakındır.Öyleyse normalize edilmiş değerleri kullanmak bize daha doğru sonuçlar verecektir.

N boyutlu uzayda , eğer iki vektör V1 ve V2 arasındaki uzaklık birbirine çok yakın ise

$V1 \cdot V2^T$  nin sonucu 1'e çok yakın çıkar.

$$V1 \cdot V2^T = V1^T \cdot V2$$

Örnek verecek olursak 3 boyutlu uzayda [3 4 5] vektörü ile [2.7 3.7 5.3] vektörlerini normalize edip çarparsak çıkan değer..... dir.

Yani özetleyecek olursak Olasılıksal Sinir Ağları yönteminde;

- 1.Önce giriş değerleri normalize ediliyor
- 2.Eğitme işlemi yapılıyor
- 3.Test ediliyor

Öncelikle giriş değerlerimizi vektör olarak alıyoruz

$$\bar{x}_n = \begin{bmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,d} \end{bmatrix}$$

Daha sonra Eğitim aşamasına geçiyoruz.

#### 6.6.1. Olasılıksal Sinir Ağlarında Eğitim Aşamaları

1.

$$j = 0, a_{ji} = 0, j = 1, 2, \dots, N, i = 1, 2, \dots, M$$

Burada 1 den n' ye kadar yani bizim uygulamamızda 1'den 1000'e kadar müşterilerimiz yani sample 'ımız dolaşılıyor. Çıkışımız ise yani çıkış sınıfımız ise 1'den 4 e kadar dolaşılıyor.

2.

$$j = j + 1$$

Burada 1.sample'dan sonra ikinci sample'a geçiş sağlamak amacıyla her seferinde müşteri sırası 1 artırılıyor.

3.

$$x_{jk} = x_{jk} / \left( \sum_{i=1}^d j^2 i \right)^{1/2}$$

$$\bar{x}_j = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jd} \end{bmatrix}$$

Burada daha önce açıklamış olduğum gibi, g, r, ş vektörleri normalize ediliyor. Yani bu aşama normalize aşaması olarak geçiyor.

4.

$$V_{jk} = X_{jk}$$

$$\bar{V}_j = \bar{X}_j$$

Burada giriş vektörlerinden ara katmana , yani müşterilerimize doğru vektör eşitlemesi yapılıyor. Yani burada ağırlıklar vektör kullanılarak update ediliyor.

5.

$$\text{Eğer } X_j \in W_i, a_{ji} = 1$$

Bu aşamada ise Xj belirlediğim kümede ise onun değerini 1 yapıyor geri kalanını 0 yapıyor. 1 yapıyor derken ürün numarası ile karıştırılmamalıdır. Buradaki 1 logic 1 anlamındadır. Yani bunu onayla gerisini 0 yap yani onaylama.

Burada ise eğer sonucu müşteriye gelinmemişse 2.basamaktan tekrar başlat yoksa programı sonlandır anlamına gelmektedir.

6. Eğer  $(j \neq N)$  basamak 2'ye git.

Eğitme işleminden sonra veriler test aşaması için kullanılacaktır. Yeni verilen test verisi öncelikle normalize edilecek ve giriş vektörü yerlerini alacaktır.

Her bir örüntü birimi normalize edilmiş vektörler ile çarpılacak ve bir net k sonucu elde etmiş olacağız.



İç çarpım=scalar çarpım

$$net_k = V_k^T x_t \dots \rightarrow \text{benzerlik ölçüsü}$$

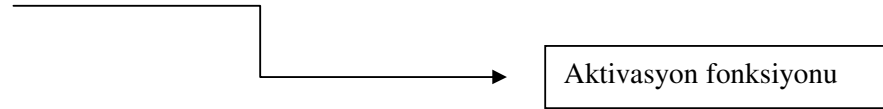
scalar çarpım=bileşenlerin çarpımlarının toplamı

Daha sonra net k formülde yerine konulur.

$$e^{(net_k - 1) / \sigma^2}$$

Yukarıda sigma değerini sabit olarak 1 alıyoruz ve çıkan formülü bu şekilde hesaplıyoruz.

$$\begin{aligned} \phi \left( \frac{x - \bar{V}_k}{h_n} \right) &\propto e^{-\frac{(x - \bar{V}_k)^T (x - \bar{V}_k)}{2 \sigma^2}} \\ &= \frac{(\bar{x}^T x) + (\bar{V}_k^T \bar{V}_k) - 2 \bar{x}^T \bar{V}_k}{2 \sigma^2} \\ &= \frac{(\bar{x}^T \bar{V}_k - 1)}{2 \sigma^2} \\ &= \frac{(\bar{x}^T \bar{V}_k - 1)}{\sigma^2} \\ &e^{(net_k - 1) / \sigma^2} \end{aligned}$$



## 6.6.2. Son Aşama

### 6.6.2.1. Test Aşaması

1.

İlk olarak 1. maddemizde test verimizi ele alıyoruz. Yani  $k$  müşterilerimizi ve  $g_i$  ise sınıflandırmak için kullanılan parametreleri ifade ediyor.

$$k = 0, g_i = 0$$

2.

Burada  $k$ 'yı her seferinde 1 arttırıyoruz. Böylece bir sonraki müşterimize geçmiş oluyor.

$$k = k + 1$$

3.

Burada  $X_t$  ile yani normalize edilmiş vektörler ile çarpım sağlanıyor.ve sonucunda net  $k$  hesaplanıyor.

$$net_k = \overline{V}_k \overline{x}^t$$

4.

Eğer  $a$  ki yani müşterimiz 1'se daha sonra  $g$ 'ye net  $k$  değerini ve  $g_i$  kadar ekle diyoruz.

Eğer;

$$a_{ki=1}$$

$$g_i = g_i + e^{\left(\frac{net_k - 1}{\sigma^2}\right)}$$

5.

Bu aşamada ise müşteri sayımız 1000 'e eşit olmadığı sürece ilerle diyoruz.

Eğer;

$(k \neq N)$

6.

$class = \arg \max g_i(x)$

$$P(x) = \sum_{n=1}^N \phi(\cdot)$$

$$g_i = \sum_{k=1}^N a_{ki} e^{\left(\frac{net_k - 1}{\sigma^2}\right)}$$

Proje'de tasarlanan program Visual.NET 2005 ortamında C# programlama dili ile yazılmıştır. Oluşturulan ara yüzde Ürün Seçenekleri ve Kişi özellikleri yer almaktadır. Ürün seçeneklerinde Şampuan, Gazete, Traş Jileti, Çay, Sigara ve Kola ürünleri örnek alınmıştır. Seçilmek istenen ürünün yanındaki kutucuk tıklanır ve seçim yapılmış olur. Daha sonra hemen ürünlerin yanında bulunan Eđit butonuna basılır. Bu buton ile program klasörümüzün içinde bulunan Kola text dosyasındaki deęerler, Probabilistic Neural Network Algoritmasına tabi tutularak eđitilmiş olur.

Öncelikle programa gözetilcek olursa;

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Threading;

namespace PNNSample
{
    public partial class PNNSampleForm : Form
    {
```

```

public PNNSampleForm()
{
    InitializeComponent();

    m_products = new ProductInfo[6]
        {
            new
ProductInfo("Şampuan", "shampoo.trn", "Blendax",
"Elidor", "Pantane", "Elseve"),
            new
ProductInfo("Traş Jileti", "gilette.trn", "Zaza",
"Gilette Blue", "Permatik", "Derby"),
            new
ProductInfo("Sigara", "cigar.trn", "Tekel 2000",
"Marlbora", "Winston", "Bafra"),
            new
ProductInfo("Gazete", "newspaper.trn",
"Milliyet", "Hürriyet", "Cumhuriyet", "Sabah"),
            new
ProductInfo("Çay", "tea.trn", "Rize", "Filiz",
"Lipton", "Deren"),
            new
ProductInfo("Kola", "cola.trn", "Coca Cola",
"Pepsi Cola", "Cola Turka", "Uludağ Kola"),
        };

    int index = 0;
    string str = "";

    foreach (RadioButton rb in
m_groupBoxProductOptions.Controls)
        str += index++ + "=>" + rb.Text +
"\n";
}

```

```

        MessageBox.Show(str);
    }
    private void m_buttonTrain_Click(object
sender, EventArgs e)
    {
        Thread thread = new Thread(new
ThreadStart(trainThreadProc));

        thread.IsBackground = true;
        thread.Start();
    }
    private void trainThreadProc()
    {
        try
        {
            string productName = "";
            for (int i = 0; i <
m_groupBoxProductOptions.Controls.Count; ++i)
            {
                if
(((RadioButton)m_groupBoxProductOptions.Controls[
i]).Checked == true)
                {
                    productName =
((RadioButton)m_groupBoxProductOptions.Controls[i
]).Text;

                    m_productIndex = i;
                    break;
                }
            }
        }
    }

```

```

        string fileName =
getProductFileName(productName);

        train(fileName);
        m_buttonTest.Enabled = true;
        m_groupBoxTest.Enabled = true;
        MessageBox.Show("Eğitime İşlemi
tamamlandı");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private string getProductFileName(string
productName)
{
    foreach (ProductInfo pi in
m_products)
        if (productName == pi.Name)
            return pi.FileName;

    return "";
}
private void m_buttonTest_Click(object
sender, EventArgs e)
{
    Thread thread = new Thread(new
ThreadStart(testThreadProc));
    thread.IsBackground = true;
    thread.Start();
}

```

```

private void testThreadProc()
{
    try
    {
        double x1 =
m_comboBoxAge.SelectedIndex;
        double x2 =
m_comboBoxJob.SelectedIndex;
        double x3 =
m_comboBoxState.SelectedIndex;

        test(x1, x2, x3, 1);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void test(double x1, double x2,
double x3, double sigma)
{
    int i, j;
    double temp = Math.Sqrt(x1 * x1 + x2
* x2 + x3 * x3);

    x1 = x1 / temp;
    x2 = x2 / temp;
    x3 = x3 / temp;

    double[] g = new double[5] { 0, 0, 0,
0, 0 };

    double[] net = new double[1001];

```

```

        for (i = 1; i <= 1000; i++)
        {
            net[i] = v[i, 1] * x1 + v[i, 2] *
x2 + v[i, 3] * x3;
            for (j = 1; j <= 4; j++)
                if (a[i, j] == 1) g[j] = g[j]
+ Math.Exp((net[i] - 1) / (sigma * sigma));
        }
        m_labelFirst.Text = g[1].ToString();
        m_labelSecond.Text = g[2].ToString();
        m_labelThird.Text = g[3].ToString();
        m_labelFourth.Text = g[4].ToString();
        m_labelResult.Text =
m_products[m_productIndex].Marks[getMax(g[1],
g[2], g[3], g[4])].ToString();
    }
    private int getMax(params double []
values)
    {
        double max = values[0];
        int index = 0;

        for (int i = 1; i < values.Length;
++i) {
            if (values[i] > max)
            {
                max = values[i];
                index = i;
            }
        }

        return index;
    }
}

```



```

private void train(string fileName)
{
    string a1 = "";
    string b = "";
    string c = "";
    string d = "";

    double[,] x = new double[1001, 5];

    FileStream fs = new
FileStream(fileName, FileMode.Open,
FileAccess.Read);
    sr = new StreamReader(fs);

    string line;
    int j = 1;
    int i;
    while (j <= 1000)
    {
        line = sr.ReadLine();
        a1 = "";
        b = "";
        c = "";
        d = "";

        i = 0;
        while (line[i] != '\t')
        {
            a1 += line[i];
            i++;
        }
        i++;
        while (line[i] != '\t')

```

```

    {
        b += line[i];
        i++;
    }
    i++;
    while (line[i] != '\t')
    {
        c += line[i];
        i++;
    }
    i++;
    while (i < line.Length)
    {
        d += line[i];
        i++;
    }
    x[j, 1] = double.Parse(a1);
    x[j, 2] = double.Parse(b);
    x[j, 3] = double.Parse(c);
    x[j, 4] = double.Parse(d);
    j++;

    double temp;

    for (i = 1; i <= 1000; i++)
    {
        temp = Math.Sqrt(x[i, 1] *
x[i, 1] + x[i, 2] * x[i, 2] + x[i, 3] * x[i, 3]);
        x[i, 1] = x[i, 1] / temp;
        x[i, 2] = x[i, 2] / temp;
        x[i, 3] = x[i, 3] / temp;
        v[i, 1] = x[i, 1];
        v[i, 2] = x[i, 2];
    }

```

```

        v[i, 3] = x[i, 3];

    }

    for (i = 1; i <= 1000; i++)
        for (int k = 1; k <= 4; k++)
            a[i, k] = 0;

    for (i = 1; i <= 1000; i++)
    {
        string uu = x[i,
4].ToString();

        int u = Int32.Parse(uu);
        a[i, u] = (double)1;
    }
}

sr.Close();

}

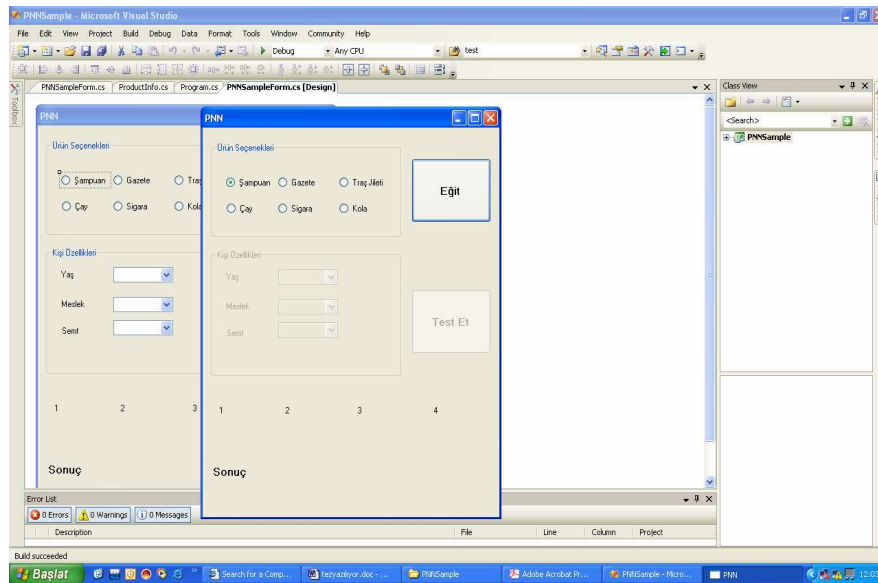
private ProductInfo[] m_products;
private int m_productIndex;
private double[,] a = new double[1001,
5];

private double[,] v = new double[1001,
4];

private StreamReader sr;
}
}

```

Programda eğitim aşağıdaki kısımda yapılmıştır. `butonTrain_Click` yani `eğit` butonuna basıldığında eğitime kodu verilir.



```
private void train(string fileName)
{
    string a1 = "";
    string b = "";
    string c = "";
    string d = "";

    double[,] x = new double[1001, 5];

    FileStream fs = new
FileStream(fileName, FileMode.Open,
FileAccess.Read);
    sr = new StreamReader(fs);

    string line;
    int j = 1;
    int i;
    while (j <= 1000)
    {
```

```
line = sr.ReadLine();
a1 = "";
b = "";
c = "";
d = "";

i = 0;
while (line[i] != '\t')
{
    a1 += line[i];
    i++;
}
i++;
while (line[i] != '\t')
{
    b += line[i];
    i++;
}
i++;
while (line[i] != '\t')
{
    c += line[i];
    i++;
}
i++;
while (i < line.Length)
{
    d += line[i];
    i++;
}
x[j, 1] = double.Parse(a1);
x[j, 2] = double.Parse(b);
x[j, 3] = double.Parse(c);
```

```

x[j, 4] = double.Parse(d);
j++;

double temp;

for (i = 1; i <= 1000; i++)
{
    temp = Math.Sqrt(x[i, 1] *
x[i, 1] + x[i, 2] * x[i, 2] + x[i, 3] * x[i, 3]);
    x[i, 1] = x[i, 1] / temp;
    x[i, 2] = x[i, 2] / temp;
    x[i, 3] = x[i, 3] / temp;
    v[i, 1] = x[i, 1];
    v[i, 2] = x[i, 2];
    v[i, 3] = x[i, 3];

}

for (i = 1; i <= 1000; i++)
    for (int k = 1; k <= 4; k++)
        a[i, k] = 0;

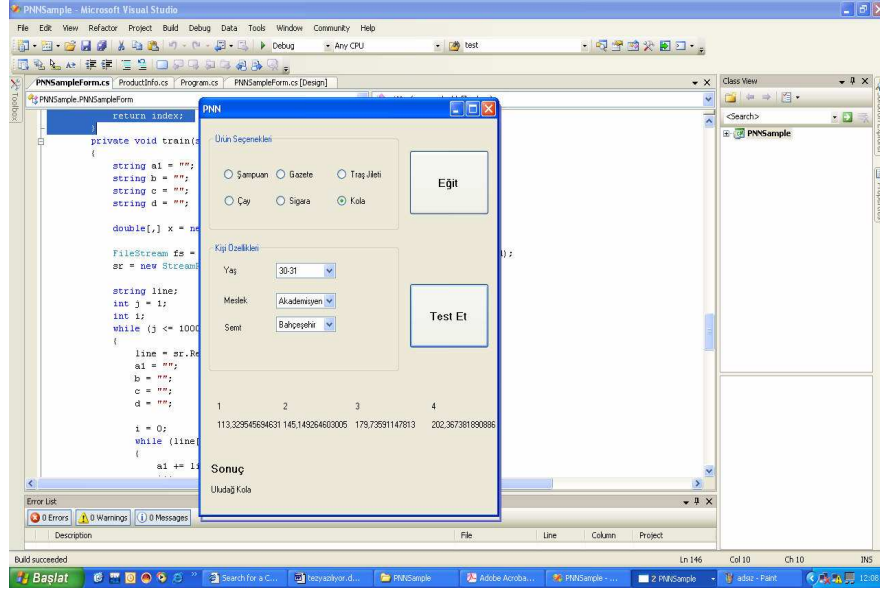
for (i = 1; i <= 1000; i++)
{
    string uu = x[i,
4].ToString();

    int u = Int32.Parse(uu);
    a[i, u] = (double)1;
}

}
sr.Close();}

```

Eğitimden sonraki aşama ise test etme kısmıdır. Aşağıdaki kod parçasında verileri test eden algoritma verilmiştir. Burada arayüzde test et butonuna asılmalıdır.



Şekil 28. PNN Windows Uygulaması Çalıştırılmış Hali

```
private void test(double x1, double x2, double x3, double sigma)
{
    int i, j;
    double temp = Math.Sqrt(x1 * x1 + x2 * x2 + x3 * x3);

    x1 = x1 / temp;
    x2 = x2 / temp;
    x3 = x3 / temp;

    double[] g = new double[5] { 0, 0, 0, 0, 0 };

    double[] net = new double[1001];

    for (i = 1; i <= 1000; i++)
```

```

        {
            net[i] = v[i, 1] * x1 + v[i, 2] *
x2 + v[i, 3] * x3;
            for (j = 1; j <= 4; j++)
                if (a[i, j] == 1) g[j] = g[j]
+ Math.Exp((net[i] - 1) / (sigma * sigma));
        }
        m_labelFirst.Text = g[1].ToString();
        m_labelSecond.Text = g[2].ToString();
        m_labelThird.Text = g[3].ToString();
        m_labelFourth.Text = g[4].ToString();
        m_labelResult.Text =
m_products[m_productIndex].Marks[getMax(g[1],
g[2], g[3], g[4])].ToString();
    }
    private int getMax(params double []
values)
    {
        double max = values[0];
        int index = 0;

        for (int i = 1; i < values.Length;
++i) {
            if (values[i] > max)
            {
                max = values[i];
                index = i;
            }
        }

        return index;
    }
}

```



## 6.3. Programın Ayrıntılı İçeriği

### 6.3.1. Ürünü Temsil Sınıfın Oluşturulması

#### 6.3.1.1. ProductInfo.cs Sınıfı

İlk olarak ProductInfo Sınıfı tanımlanmıştır. Bunun nedeni daha sonra program içinde ürünler ve markaları belirtilecektir.

```
namespace PNNSample
{
    struct ProductInfo
    {
        public ProductInfo(string name, string
fileName, params string [] marks)
```

//Yukarıdaki satırda ilk parametre ürün adı, ikinci parametre verilerin çekileceği dosya ismi, 3. parametre ise 1 ürünün alt markalarını temsil edecektir.

```
    {
        Name = name;
        FileName = fileName;
        Marks = new string[4]{marks[0],
marks[1], marks[2], marks[3]};
    }
```

Yukarıdaki kısımda ise markalar için 4 lük dizi oluşturulmuştur. Çünkü ürün marka adedimiz 4 'tür.

```
        public string Name;
        public string FileName;
```

```
        public string [] Marks;
    }
}
```

### 6.3.2. ProductInfo sınıfından nesne oluşturulması

Arayüzün bu kısmında yani Yaş, Meslek ve Semt kutucuklarından alınacak değerler için kod kısmı aşağıda belirtilmiştir.

Burada toplam 6 ürünümüz olduğu için 6 lık dizi oluşturulmuştur. Bu dizi içine ürünlerimiz yerleştirilmiştir.

```
m_products = new ProductInfo[6]
{
    new
ProductInfo("Şampuan", "shampoo.trn", "Blendax",
"Elidor", "Pantane", "Elseve"),
    new
ProductInfo("Traş Jileti", "gilette.trn", "Zaza",
"Gilette Blue", "Permatik", "Derby"),
    new
ProductInfo("Sigara", "cigar.trn", "Tekel 2000",
"Marlbora", "Winston", "Bafra"),
    new
ProductInfo("Gazete", "newspaper.trn",
"Milliyet", "Hürriyet", "Cumhuriyet", "Sabah"),
    new
ProductInfo("Çay", "tea.trn", "Rize", "Filiz",
"Lipton", "Deren"),
    new
ProductInfo("Kola", "cola.trn", "Coca Cola",
"Pepsi Cola", "Cola Turka", "Uludağ Kola"),
};

int index = 0;
```

```

        string str = "";

        foreach (RadioButton rb in
m_groupBoxProductOptions.Controls)
            str += index++ + "=>" + rb.Text +
"\n";

        MessageBox.Show(str)
    }

```

### 6.3.3. Kodun Eğitme Kısmı

Bu kısımda ise dosyadan okumak amacıyla fileName türünden train fonksiyonu yaratılıyor.

```

private void train(string fileName)
{
    string a1 = "";
    string b = "";
    string c = "";
    string d = "";

    double[,] x = new double[1001, 5];

```

Yukarıdaki satırda ise 4 lük sütün ve 1000 'lik satır açılıyor.

Aşağıdaki kısımda ise dosyadan okuma gerçekleşiyor.

```

        FileStream fs = new
FileStream(fileName, FileMode.Open,
FileAccess.Read);
        sr = new StreamReader(fs);

```

Aşağıda ise verileri okurken 1000'e kadar döngünün dönmesi sağlanıyor.

```
string line;
int j = 1;
int i;
while (j <= 1000)
{
    line = sr.ReadLine();
    a1 = "";
    b = "";
    c = "";
    d = "";
```

Yukarıda ise satır satır okunacağı için a1 yani 1.sütun un 1.satırını,sonra 2.sütunun 1.satırını,sonra 3.sütunun 1.satırını ve daha sonra ise 4.sütunun 1.satırını olarak geçiyor.

Daha sonra döngü döndükçe okuma işlemi 2.satırdan 1000.satıra kadar devam ediyor.

```
i = 0;
while (line[i] != '\t')
{
    a1 += line[i];
    i++;
}
i++;
while (line[i] != '\t')
{
    b += line[i];
    i++;
```

```

    }
    i++;
    while (line[i] != '\t')
    {
        c += line[i];
        i++;
    }
    i++;
    while (i < line.Length)
    {
        d += line[i];
        i++;
    }
    x[j, 1] = double.Parse(a1);
    x[j, 2] = double.Parse(b);
    x[j, 3] = double.Parse(c);
    x[j, 4] = double.Parse(d);
    j++;

```

Yukarıdaki while döngülerinde ki amaç ise okuma işlemi yapılırken veriler arasında boşluklar olacağından o kısımları geçmesini sağlamaktır.

Daha sonra alınan temp değişkeni yaratılıyor. Ve döngü 1000'e kadar dönerken normalize edebilmek amacıyla 1.satırdaki verilerin kareleri toplamının karekökleri temp değişkeninde saklanıyor.

Daha sonra ise bu temp değişkenleri her bir değer için bölüm haline geliyor.

```
double temp;
```

```
for (i = 1; i <= 1000; i++)
```

```
{
```

Burada `math.sqrt` fonksiyonun karakök alma işlemi için kullanılıyor.

```
temp = Math.Sqrt(x[i, 1] *  
x[i, 1] + x[i, 2] * x[i, 2] + x[i, 3] * x[i, 3]);  
x[i, 1] = x[i, 1] / temp;  
x[i, 2] = x[i, 2] / temp;  
x[i, 3] = x[i, 3] / temp;  
v[i, 1] = x[i, 1];  
v[i, 2] = x[i, 2];  
v[i, 3] = x[i, 3];
```

```
}
```

```
for (i = 1; i <= 1000; i++)  
    for (int k = 1; k <= 4; k++)  
        a[i, k] = 0;
```

```
for (i = 1; i <= 1000; i++)  
{  
    string uu = x[i,  
4].ToString();  
    int u = Int32.Parse(uu);  
    a[i, u] = (double)1;
```

```
}
```

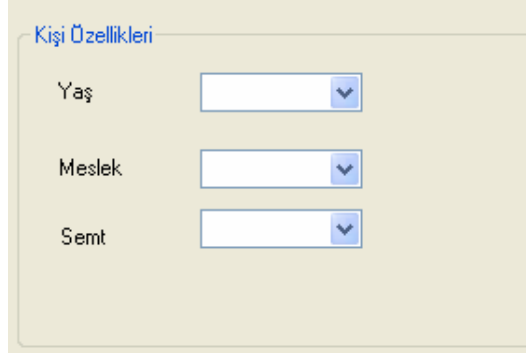
```
}
```

```
sr.Close();//dosya kapatılıyor...
```

```
}
```

### 6.3.4. Verilerin Test Aşaması

Bu kısımda kullanıcının gireceği Yas meslek ve semt özelliği kutucukları için test sınıfı yazılıyor.



**Şekil 29. PNN Uygulamasında Ele Alınan Özelliklerin Windows Uygulaması olarak gösterimi**

```
private void test(double x1, double x2, double
x3, double sigma)
{
    int i, j;
```

Bu kısımda ise alınan değerler train aşamasındaki gibi normalize edilmek için kareleri alınıp toplanıyor ve kökleri alınıp temp değişkenine atanıyor.

```
double temp = Math.Sqrt(x1 * x1 + x2
* x2 + x3 * x3);
```

bu değerler temp değişkenine bölünüyor ki normalize edilmiş olsun.

```
x1 = x1 / temp;
```

```

x2 = x2 / temp;
x3 = x3 / temp;

double[] g = new double[5] { 0, 0, 0,
0, 0 };

double[] net = new double[1001];

```

Bu kısımda ise 1000'e kadar dönen döngü açılıyor. Daha sonra her bir sample yani satır için net k değeri hesaplanmak isteniyor.

```

for (i = 1; i <= 1000; i++)
{
    net[i] = v[i, 1] * x1 + v[i, 2] *
x2 + v[i, 3] * x3;

```

Net k değeri hesaplandıktan sonra ise sıra gi değerini bulmaya geliyor. Zaten daha önce net k bulunmuştu, denklemde yerine konulup gi hesaplanıyor.

```

for (j = 1; j <= 4; j++)

```

Aşağıdaki satırda ise çıkış sınıfı 1 e eşitse gi ye net k değerini topla komutunu vermiş oluyor.

```

    if (a[i, j] == 1) g[j] = g[j]
+ Math.Exp((net[i] - 1) / (sigma * sigma));
}
m_labelFirst.Text = g[1].ToString();

```



```

        m_labelSecond.Text = g[2].ToString();
        m_labelThird.Text = g[3].ToString();
        m_labelFourth.Text = g[4].ToString();
        m_labelResult.Text =
m_products[m_productIndex].Marks[getMax(g[1],
g[2], g[3], g[4])].ToString();
    }

```

Aşağıdaki kısımda ise çıkacak olan 4 değerden hangisinin büyük olduğunu bulabilmek için getMax fonksiyonu yazılıyor.

```

private int getMax(params double []
values)
{
    double max = values[0];
    int index = 0;

    for (int i = 1; i < values.Length;
++i) {
        if (values[i] > max)
        {
            max = values[i];
            index = i;
        }
    }return index;
}

```

PNN

Ürün Seçenekleri

Şampuan  Gazete  Traş Jileti

Çay  Sigara  Kola

Eğit

Kişi Özellikleri

Yaş

Meslek

Semt

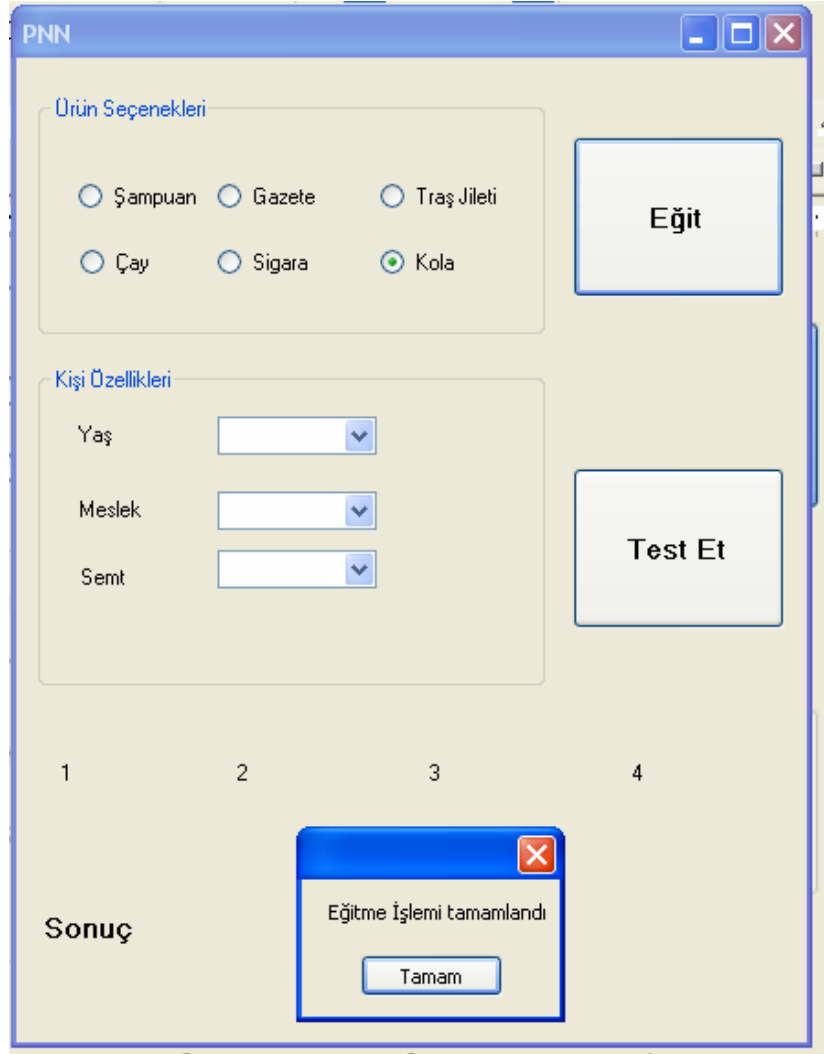
Test Et

1 2 3 4

Sonuç

**Şekil 30. PNN Uygulama Eğitim Aşaması**

Eğitme butonuna basıldıktan kısa bir süre sonunda Eğitim İşlemi Tamamlandı uyarısı veren bir pencere açığa çıkar. Bu pencere ekranda belirdiğinde aşağıda görüldüğü üzere Kullanıcı özellikleri kutucukları aktif konuma geçmiş olacaktır.



**Şekil 31. Eğitime İşleminin Tamamlandığı Göstergesi**

Daha sonra Kullanıcı özelliklerinden rasgale veriler belirlenir. Burada Yas kutucuğundan 16-17 yaş aralığı, Meslek kutucuğundan öğrenci mesleği ve oturduğu semt olarak Bahçeşehir seçilmiştir.

PNN

Ürün Seçenekleri

Şampuan  Gazete  Traş Jileti

Çay  Sigara  Kola

Eğit

Kişi Özellikleri

Yaş 16-17

Meslek öğrenci

Semt Bahçeşehir

Test Et

1 2 3 4

Sonuç

Şekil 32. PNN Windows Uygulaması Test Aşamasında

Seçim yapıldıktan sonra Test et butonuna basılır. Daha sonra aşağıda görüldüğü gibi görüldüğü gibi 4 değer açığa çıkacaktır. Bu değerlerden maksimumu alınarak ürün sonucu verilmiştir. Yani 206.9975 sonucuyla Coca-Cola ürünü tercih edilmiştir. Daha önce belirttiğim üzere 1-2-3-4 gruplar sırasıyla Coca-Cola, Pepsi Cola, Cola Turca ve Uludağ Kola ürünlerini ifade etmektedir.

PNN

Ürün Seçenekleri

Şampuan  Gazete  Traş Jileti

Çay  Sigara  Kola

Eğit

Kişi Özellikleri

Yaş 16-17

Meslek öğrenci

Semt Bahçeşehir

Test Et

1	2	3	4
206,997500417373	191,886062131226	147,923437226852	111,743364475507

**Sonuç**

Coca Cola

Şekil 33. PNN Programı Sonuç

Yukarıdaki tablodan anlaşılacağı üzere 16-17 yaş grubunda, öğrenci ve semti Bahçeşehir olan müşteriler büyük olasılıkla Coca Cola ürününü tercih edeceklerdir.

## 7. VERİ MADENCİLİĞİ ÇALIŞMASI

### 7.1. Proje 1.Kısımın Hazırlık Aşaması

Bu projeyi hazırlamadan önce, Ragesh Agrawal'ın sunduğu makaleden esinlendiğimi belirtmek isterim. Ragesh Agrawal makalesinde bir perakende markette yapılan satışları göz önüne almaktadır. Amacı insanların alışveriş alışkanlıklarını saptamaktır. Yani bir nevi alışveriş yapan insanların tercih ettiği 2li ve 3 lü ürünlerdir. Müşteri farkında olmaksızın bebek bezi alırken yanında bira da almaktaydı.

Bunu farkederek Ragesh Agrawal insanların alışkanlıklarını belirleyerek market sahibinin reyonlarını buna göre dizmesini sağlamıştır. Ragesh Agrawal bu işi yapabilmek için bir nevi algoritma üretmiştir kafasında.

Aşağıda görmüş olduğunuz tabloda ilk tabloda müşterilerin yapmış olduğu alışverişler görülmektedir. 100. müşteri 1 3 ve 4. ürünleri almıştır. 200. müşteri ise 2 3 ve 5. ürünleri tercih etmiştir. 300. Müşteri ise 1 2. 3. ve 5. ürünleri almıştır, son olarak 400 .müşteri 2 ve 5. ürünleri satın almıştır. Bu oluşturulan veritabanı sayesinde müşterilerin almış oldukları ürünler ve müşteri sıraları kayıt edilmiş olmaktadır. Hemen sağdaki 2 .tabloda ise ürünler kendi içinde kümelendirilmiştir. En sağdaki 3. tabloda ise ürünlerin alınan sayısı belirtilmektedir. Burada 4. ürün sadece 1 müşteri tarafından tercih edildiği için tabloda yer verilmemiştir. Görüldüğü gibi 1 nolu üründen 2 adet , 2 nolu üründen 3 adet, 3 nolu üründen 3 adet , 5 nolu üründen ise 3 adet alınmıştır. Hemen alttaki 2. sıradaki ilk tabloda ikili ürün kümeleri oluşturulmuştur. Yani burada 2 li ürünlerden birlikte satılma miktarı tespit edilecektir. Hemen ortadaki tabloda ise müşterilerin aldıkları 2li ürünler görülmektedir. Kısaca anlatmak istenirse 100 nolu müşteri alışverişinde 1 ve 3 ürünlerini bir arada almıştır. 200 nolu müşteri ise (2,3), (2,5) ve (3,5) ürünlerini bir arada almıştır. Burada yanlış anlaşılabilir bir konudan bahsetmek isterim. 200 nolu müşteri diyerek aslında 200. alışveriş anlamındadır. 300. müşteri ise (1,2) (1,3) (1,5) (2,3) (2,5) ve (3,5) ürünlerini bir arada tercih etmiştir.

Ortadaki tablolardan en sağdaki ise belirtmek istenen 1,3 ürünlerinin bir ara da 2 defa tercih edildiğidir. Burada (2,3) ürünleri 2 defa, (2,5) ürünleri 3 defa, (3,5) ürünleri ise 2 defa tercih edilmiştir.

En alt tablolarda ise 3 lü grup halinde alınan ürünlerin sayısından söz edilmektedir. Orta tablodan söz etmek gerekirse 2,3,5 ürünleri 200 müşteri tarafından birlikte alınmıştır. aynı zamanda 300 alışveriş sırasında ise 2,3 ve 5 ürünleri de bir ara da alınmıştır. En sağdaki tabloda ise 2,3, ve 5 ürünlerinin 2 defa tercih edildiği belirtilmiştir.

### 7.1.1. Ragesh Agrawal'ın Makalesinde; Geliştirdiği Veritabanı Gösterim Tabloları

Ragesh Agrawal'ın perakende market satış karını artırmak için geliştirdiği tablolar.

Database					
TID	Items	TID	Set-of-Itemsets	Itemset	Support
100	1 3 4	100	{ {1}, {3}, {4} }	{1}	2
200	2 3 5	200	{ {2}, {3}, {5} }	{2}	3
300	1 2 3 5	300	{ {1}, {2}, {3}, {5} }	{3}	3
400	2 5	400	{ {2}, {5} }	{5}	3

Itemset	TID	Set-of-Itemsets	Itemset	Support
{1 2}	100	{ {1 3} }	{1 3}	2
{1 3}	200	{ {2 3}, {2 5}, {3 5} }	{2 3}	2
{1 5}	300	{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }	{2 5}	3
{2 3}	400	{ {2 5} }	{3 5}	2
{2 5}				
{3 5}				

Itemset	TID	Set-of-Itemsets	Itemset	Support
{2 3 5}	200	{ {2 3 5} }	{2 3 5}	2
	300	{ {2 3 5} }		

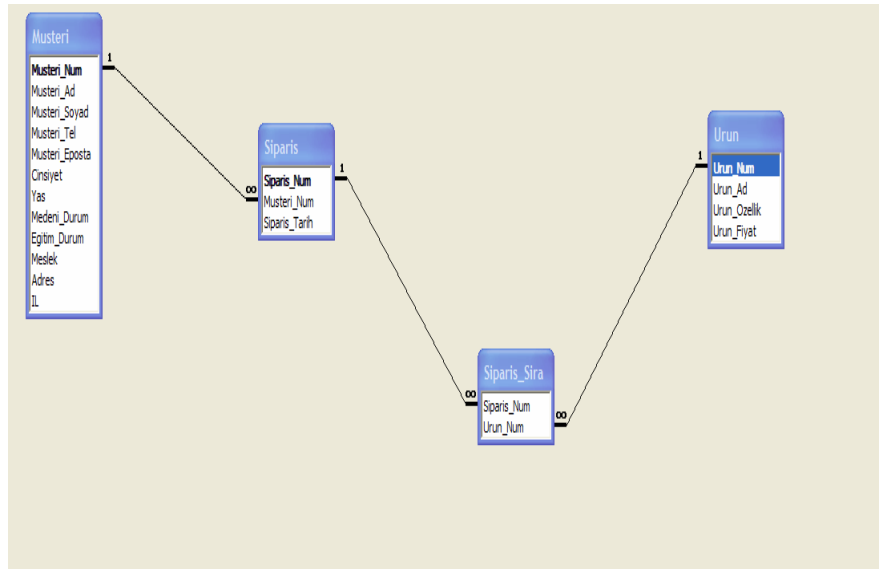
Şekil 34. Veritabanı Gösterim Tabloları

## 7.2. Hazırlanan Proje Hakkında

Burada yüksek miktarda sipariş gerçekleştiren şirketlerin veritabanlarını inceleyerek birleşme kurallarını ortaya çıkarmayı amaçlayan algoritmalar geliştirdik. Büyük perakende satış yapan organizasyonlarının en büyük sıkıntısı yüzleştikleri karar destek problemdir. Bu organizasyonlar çok kısa bir zamanda çok büyük miktarlarda satış verisini veritabanlarına eklerler. Bu veritabanları ürünlerin pazarlanmasında kullanılacak önemli verileri içinde barındırır. Bu birleşme kurallarını bulmak özellikle zıt pazarlamada (cross marketing) çok işe yarar. Ürün katalogları tasarlanmasında ve reyonların tasarlanmasında çok işe yarar.

Proje C# ortamında geliştirilmiştir. Veritabanı olarak Access kullanılmıştır. Öncelikle oluşturmuş olduğum veritabanını göstermek istiyorum.

Yukarıdaki tabloda görüldüğü gibi 4 ayrı tablo yaratılmıştır. Bunlar Müşteri, Sipariş, Sipariş\_Sıra ve Ürün tablolarıdır. Yukarıda oluşturduğum tablolar arasında ilişkiler oluşturdum. Daha sonra sql komutları ile database den verileri çektim ve c#ta kullandım. Aşağıda ilişkiler tablosunu görmekteyiz;



Şekil 35. Veritabanı İlişkiler Gösterimi



Yukarıda görüldüğü gibi bu tabloda müşteri kayıtları yapılmıştır. Toplam müşteri adedi 270'tir. Burada Müşteri\_Num, yani müşteri numarası, müşteri adı , müşteri soyadı , müşteri telefonu ve müşteri e-posta adresi bilgileri müşteri adedine göre girilmiştir.

Unun_Num	Unun_Ad	Unun_Ozellik	Unun_Fiyat
1	Migros Sele Zeytin	450g Baharatlı	2,49
2	Bahçıvan Kaşar	Ditimli Light 22g	3,99
3	Migros Light Kaşık Nektar	1 lt	1,19
4	Filiz Tagliatelle-Fettucine-Linguine	250g	1,45
5	Maret Mangal Sosu	300g	3,19
6	Talasco Garlic Sos	60ml	5,49
7	Tat Mantar	340g	2,99
8	Pinar Ekstra Light Süt	1 lt	1,59
9	Ruf Rus Pastası	700g	4,25
10	Bosphorus Aknık Tuzluk Biberlik		11,9
11	Tat Kuntuluş Domates	200g	1,99
12	Maret Boncuk Sucuk	Acılı 250g	5,99
13	DANONE DOĞAL SÜT	1 KG. 4LU	6,75
14	NESCAFÉ GOLD	200 GR	11,29
15	SESU SIRLI ILIK AGDA	360 GR	7,65
16	YUMOS	4 KG LAVANTİ	6,59
17	DOVE 2 LI BODY FIRMING JEL	200 ML	16,35
18	SÜTAŞ KOYUN PEYNİRİ	650 GR	8,99
19	SELPAK HAVLU	6LI	5,29
20	YÖRSAN TAVA YOĞURT	2 KG	5,25
21	DOVE BODY SILK	300 ML	6,99
22	BLUELINE KLİMA	9.000 BTU	359
23	FILİZ AKIN BAGIS BİLEZİĞİ		1
24	CAPPY %100 PORTAKAL MEYVES	200 ml	0,6
25	CAPPY %100 PORTAKAL SUYU	1 L	1,99
26	CAPPY KARIŞIK MEYVE NEKTARI	200 ML	0,55
27	CAPPY KAYISI	1 L	1,48
28	CAPPY KAYIS METAL KUTU		0,85
29	CAPPY TROPİK	330 ML	0,85
30	CAPRI SUN PORTAKAL	200 ML	0,49
31	DİMES ANANAS NEKTARI	1 L	2,1
32	KAVAKLİDERE ÜZÜM SUYU	6 LI KIRMIZI	7,29
33	LIPTON ICE TEA ŞEFTALI LIGHT	330 ML	0,85
34	LIPTON ICE TEA LİMON	330 ML	0,85
35	MİGROS PORTAKAL	2,5 LT	1,38

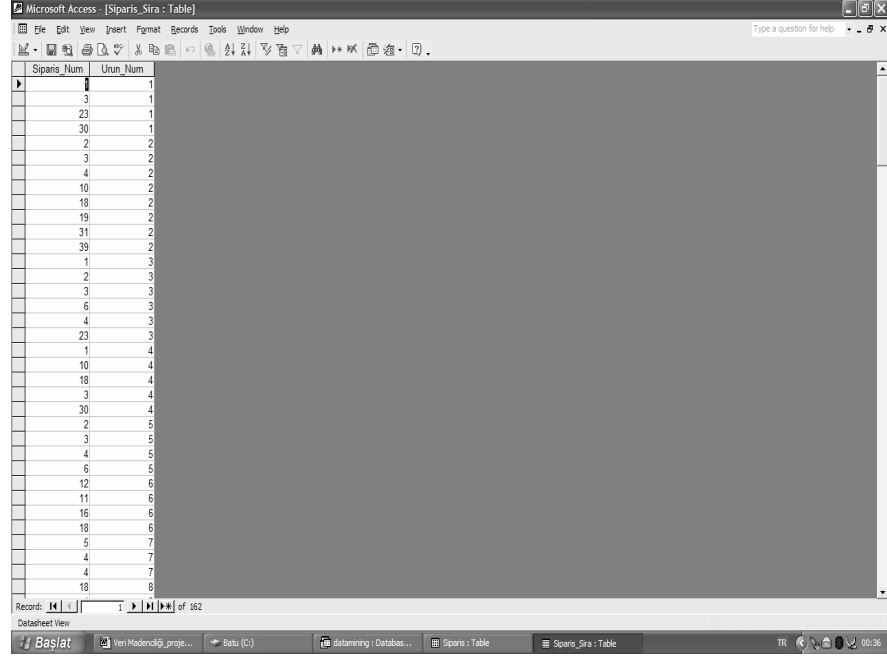
Şekil 36. Excel de Ürün Gösterimi

Yukarıdaki tabloda ise Migros ürünlerinin bir kısmı 414 adet ürün girilmiştir. Bura da ürün numarası, ürün adı, ürün özelliği ve ürün fiyatı bilgileri girilmiştir.

Siparis_Num	Musteri_Num	Siparis_Tarih
1	5	01.06.2003
2	45	03.06.2003
3	123	04.07.2003
4	98	02.09.2003
5	102	02.09.2003
6	33	04.09.2003
7	21	05.09.2003
8	33	06.09.2003
9	26	06.09.2003
10	78	07.09.2003
11	59	08.09.2003
12	67	09.09.2003
13	54	11.09.2003
14	33	13.09.2003
15	24	14.09.2003
16	56	14.09.2003
17	99	15.09.2003
18	125	15.09.2003
19	223	16.09.2003
20	228	17.09.2003
21	266	17.09.2003
22	232	18.09.2003
23	211	18.09.2003
24	169	19.09.2003
25	138	19.09.2003
26	122	20.09.2003
27	135	21.09.2003
28	123	22.09.2003
29	155	22.09.2003
30	167	22.09.2003
31	234	23.09.2003

Şekil 37. Excel de Sipariş Tablosu 1

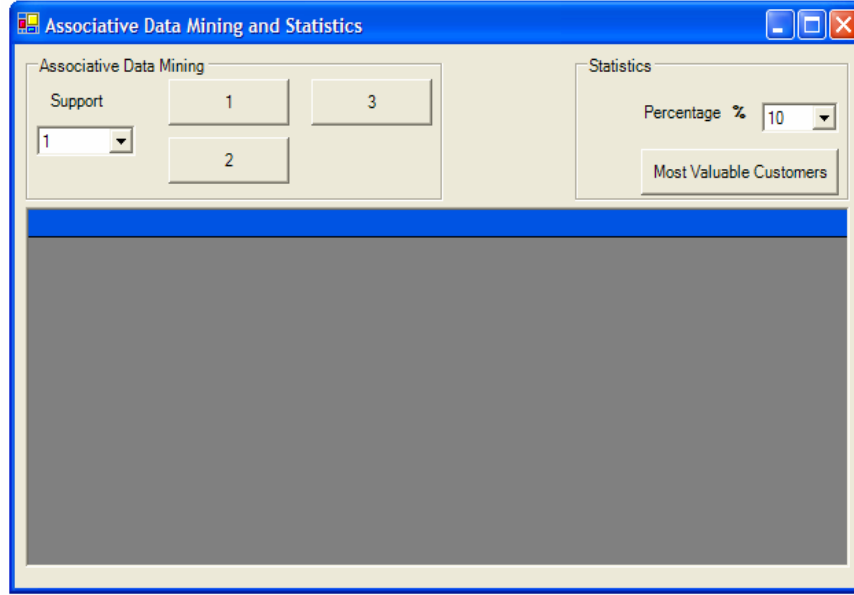
Yukarıdaki tabloda ise Sipariş numarası, müşteri numarası ve sipariş tarihi görülmektedir. Buradaki sipariş numarası otomatik olarak artmaktadır. Müşteri numarası ise aklımızdan rasgele 270'le kadar girdiğimiz değerlerden oluşmaktadır.



Siparis_Num	Urun_Num
1	1
3	1
23	1
30	1
2	2
3	2
4	2
10	2
18	2
19	2
31	2
39	2
1	3
2	3
3	3
6	3
4	3
23	3
1	4
10	4
18	4
3	4
30	4
2	5
3	5
4	5
5	5
12	6
11	6
16	6
18	6
5	7
4	7
4	7
18	8

Şekil 38. Excel de Sipariş Tablosu 2

Yukarıdaki tabloyu oluşturabilmek için öncelikle bir önceki tablodan girdiğimiz Sipariş numarasına göre ürün numaralarını girmiş bulunmaktayız.



### Şekil 39. İlişkilendirilmiş Veri Madenciliği Arayüzü

Yukardaki tabloda ise programı çalıştırdığında karşımıza gelen ekran görülmektedir. Soldaki kısımda support bir ürünün kaç defa yani kaç adet alındığını belirtmektedir. Biz bu ekrandan support 1 seçebiliriz yani örneğin 2 seçtiğimizde 2 ve daha fazla sayıda satılan ürünler karşımıza çıkacaktır. Yani alınan ürün miktarı ve daha fazlası ekrana basılacaktır. Kesinlikle daha azı basılamaz. Support combo box ın yanında 3 adet buton bulunmaktadır. Bu butonlardan 1.si tekli ürünlerin yani müşteri gelip tek bir ürün satın alıyor ve sadece ekrana bunu basıyor. 2.butona basarsak göreceğiz ki 2 li alınan ürünlerin alınma sayısı ekrana basılacaktır.3.butona basmamız sonucunda ise 3lü ürünlerin yani 3 ü bir arada alınan ürünlerin alınma sayısını ekrana basacaktır.

Örnek vermek gerekirse 1 nolu butona basalım ve neler olduğunu görelim;

Ürün	Özellik	Fiyat	Satım Adedi
Migros Sele Ze	450g Baharatlı	2.49 YTL	4
Bahçivan Kaşar	Dilimli Light 225	3.99 YTL	8
Migros Light Ka	1 lt	1.19 YTL	6
Filiz Taglatelle-	250g	1.45 YTL	5
Maret Mangal S	300g	3.19 YTL	4
Tabasco Garlic	50ml	5.49 YTL	4
Tat Mantar	340g	2.99 YTL	3
Pinar Ekstra Lip	1 lt	1.59 YTL	5
Ruf Rus Pastas	700g	4.25 YTL	1
Tat Kurutulmuş	200g	1.99 YTL	4
Maret Borçuk S	Acılı 250g	5.99 YTL	2
DANONE DOĞ	1 KG. 4LU	6.75 YTL	2

**Şekil 40. Tekli Alınan Ürünler Listesi**

Burada görüldüğü gibi tekli alınan ürünler ve satım adetleri listelenmiştir. Örneğin görüldüğü üzere Migros Sele Zeytin 4 adet satılmıştır.

İkinci butona basılacak olunursa; burada 2ye basıyorsunuz.

Ürün1	Özellik1	Fiyat1	Ürün2	Özellik2	Fiyat2	Eşleşme
Migros Sele Ze	450g Baharatlı	2.49 YTL	Bahçivan Kaşar	Dilimli Light 225	3.99 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	Migros Light Ka	1 lt	1.19 YTL	3
Migros Sele Ze	450g Baharatlı	2.49 YTL	Filiz Taglatelle-	250g	1.45 YTL	3
Migros Sele Ze	450g Baharatlı	2.49 YTL	Maret Mangal S	300g	3.19 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	DOVE BODY S	300 ML	6.99 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	MARET HESA	250 GR ACILI	4.99 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	GAJA HURRI		1.10 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	ETİ FİNDIKLI	80 GR	1.28 YTL	1
Migros Sele Ze	450g Baharatlı	2.49 YTL	PINAR MADR	8 LT.	2.25 YTL	1
Bahçivan Kaşar	Dilimli Light 225	3.99 YTL	Migros Light Ka	1 lt	1.19 YTL	3
Bahçivan Kaşar	Dilimli Light 225	3.99 YTL	Filiz Taglatelle-	250g	1.45 YTL	3

**Şekil 41. İkili Alınan Ürünler Listesi**

Burada görüldüğü üzere 2 ürün aynı satırda birlikte listelenmiş ve bu şekilde yani ikili satılma adedi görüntülenmiştir. Örnek verecek olursak Migros Sele Zeytin ve Pınar Madran Su 1 defa birlikte satılmıştır.

3. Butona basılacak olursa gelin bakalım neler göreceğiz;

in1	Özellik1	Fiyat1	Ürün2	Özellik2	Fiyat2	Ürün3
ros Sele Ze	450g Baharatlı	2,49 YTL	Bahçivan Kaşar	Dilimli Light 225	3,99 YTL	Migros Lig
ros Sele Ze	450g Baharatlı	2,49 YTL	Bahçivan Kaşar	Dilimli Light 225	3,99 YTL	Filiz Taglı
ros Sele Ze	450g Baharatlı	2,49 YTL	Bahçivan Kaşar	Dilimli Light 225	3,99 YTL	Maret Man
ros Sele Ze	450g Baharatlı	2,49 YTL	Migros Light Ka	1 lt	1,19 YTL	Filiz Taglı
ros Sele Ze	450g Baharatlı	2,49 YTL	Migros Light Ka	1 lt	1,19 YTL	Maret Man
ros Sele Ze	450g Baharatlı	2,49 YTL	Migros Light Ka	1 lt	1,19 YTL	MARET M
ros Sele Ze	450g Baharatlı	2,49 YTL	Migros Light Ka	1 lt	1,19 YTL	ETİ FIND
ros Sele Ze	450g Baharatlı	2,49 YTL	Migros Light Ka	1 lt	1,19 YTL	PINAR M.
ros Sele Ze	450g Baharatlı	2,49 YTL	Filiz Taglatelle-	250g	1,45 YTL	Maret Man
ros Sele Ze	450g Baharatlı	2,49 YTL	Filiz Taglatelle-	250g	1,45 YTL	DOVE BOI
ros Sele Ze	450g Baharatlı	2,49 YTL	Filiz Taglatelle-	250g	1,45 YTL	GALA HC

Şekil 42. Üçlü Alman Ürünler Listesi

Bu tabloda görüldüğü üzere 3 lü ürünler bir arada görünmektedir. Yani 3 ürün özellik ve fiyat ve isimleri ile birlikte ve satım adetleri belirtilmiştir. Örneğin Migros Sele Zeytin, Migros Light Kaşar ve Filiz Taglitide birarada 2 kez satılmıştır. Yani bu ürünler 2 kez tercih sebebi olmuştur aynı anda, aynı alışveriş zamanında.

Sayfanın sağ tarafında görünen olayda ise ;

Musteri_Ad	Musteri_Soyad	Musteri_Tel	Musteri_Eposta	Harcama
Hamdi	AKSOY	0216665577	(null)	438,39
Cengiz	BAŞER	(null)	(null)	373,00
Aytekin	HIZAL	(null)	(null)	58,60
Bayram	EFE	(null)	(null)	58,61
Esat	KALÇA	(null)	(null)	48,70
Birkan	KURT	(null)	(null)	30,75
Ali	KAZDAL	(null)	(null)	27,25
İsmail	MERTTORK	(null)	(null)	23,72

Şekil 43. Müşteri Yüzdeler Dilimleri

Burada percentage seçilmelidir öncelikle, yani en fazla alışveriş yapan insanların yüzdesi seçilecektir başlangıçta. Örneğin database de alışveriş yapan 30 insan var diyelim böylece 30 insanın %20 si alınarak 6 kişi ekrana basılacaktır. Bu kişiler in telefon,e-posta ve alışveriş tutarları fazladan aza doğru sıralanmıştır.

Açıklamak istediğim bir başka mevzu ise burada faydalandığım iki bölüm; Ekranın sol tarafı tamamiyle associative data mining sağ tarafı ise istatistik ile ilgilidir.

### **7.3. İstatistik**

SQL ile kolaylıkla bir çok istatistiksel bilgi ortaya çıkar. avg, sum, min, max, group by order by ifadeleri ile bir çok sorgu yapılır. Ancak veri madenciliğinde: Sql yetmez.Sql ile çekilen bilgi üzerinde çalışmak gerekir. Yani algoritmalar geliştirmek gerekir. Şimdide bu buton işlemlerini gerçekleştirebilmek için geliştirdiğim algoritmaların kodunu görmektesiniz.

### **7.4. İkinci Bölüm**

#### **7.4.1. Karar Ağacı Mekanizması**

Karar ağaçları (decision trees), sınıflandırma, kümeleme ve tahmin modellerinde kullanılan bir tahmin tekniğidir. Sorunla ilgili araştırma alanını alt gruplara ayırmak için kullanılır.

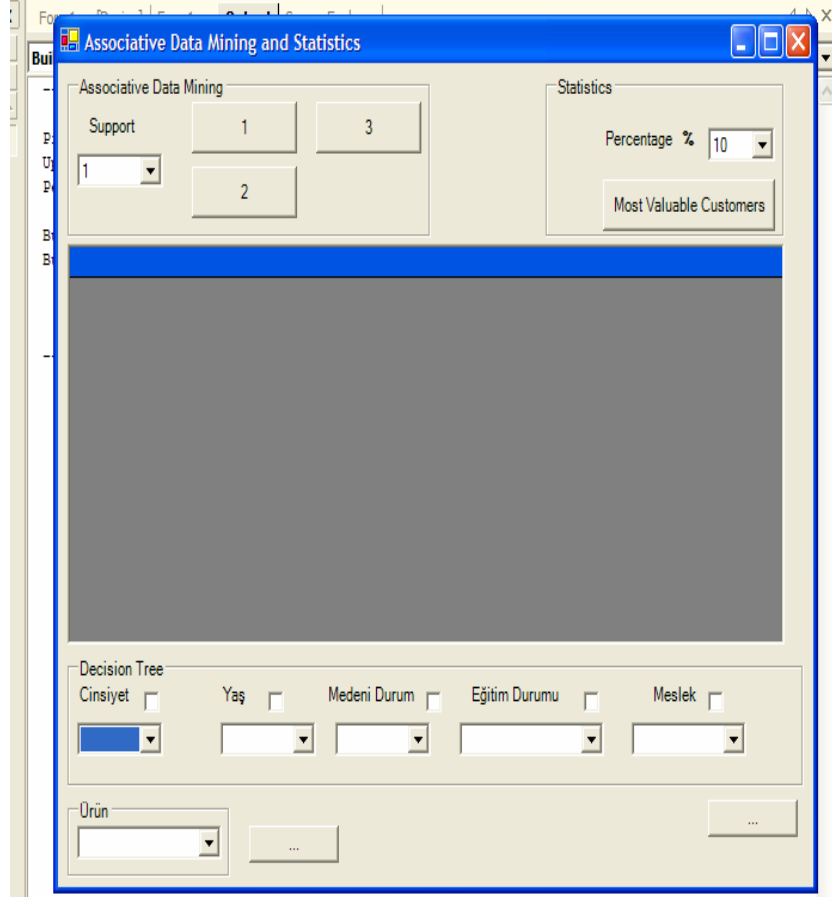
Karar ağaçlarında kök ve her düğüm bir soruyla etiketlenir. Düğümlerden ayrılan dallar ise ilgili sorunun olası yanıtlarını belirtir. Her dal düğümü de söz konusu sorunun çözümüne yönelik bir tahmini temsil eder.

Karar ağaçları, üç bölümden oluşan bir modeldir.

1. Tanımdaki gibi bir karar ağacı
2. Ağacı oluşturacak bir algoritma
3. Ağacı veriye uygulayacak ve söz konusu sorunu çözecek bir algoritma.

Karar ağaçları;

- Eğitici örnekteki veriyi sınyan bir algoritma aracılığıyla gerçekleştirilir
- Karar ağacı tekniklerinin çoğu, birbirlerinden ağacın nasıl oluşturulduğıyla ayrılır.



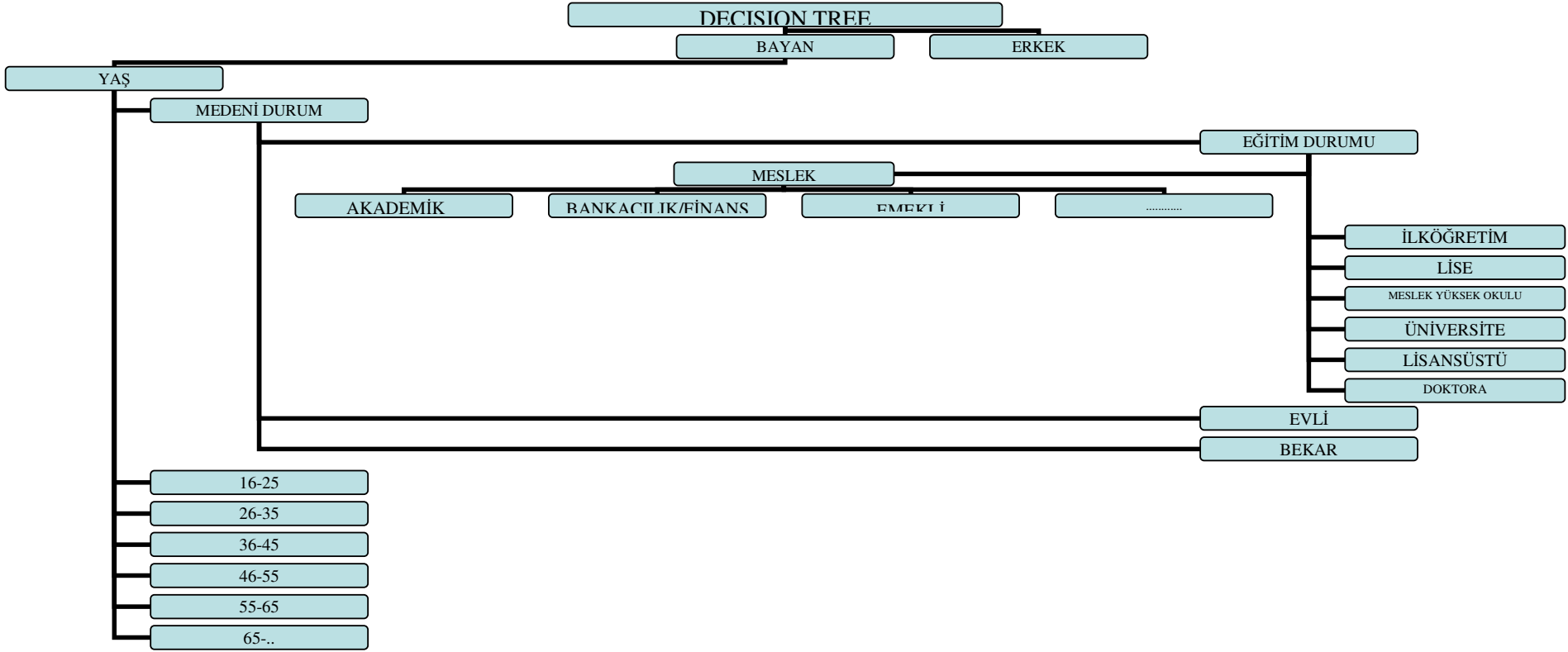
**Şekil 44. Karar Ağacı Oluşturulan Windows Uygulaması**

Bu bölümde tablonun alt bölümüyle ilgileneceğiz. Burada 2 amacımız var. Birincisi ise sadece ürün kutudan seçilecek, ürün seçildikten sonra cinsiyet, yaş, medeni durum meslek ve eğitim durumundan herhangi biri veya hepsine form doldurulacak ve bu ürünü alan müşterilerin cinsiyet, yaş, medeni durum, eğitim durumu ve meslek leri ile ilgili bilgiler sql veritabanından c# kodu le çekilip görüntülenecektir. İkinci kısımda cinsiyet, yaş, medeni durum ve

meslek bilgilerinin hep birlikte, drtl, l, ikili veya tek olarak iaretlendikten butona basılmasının ardından oluşacak olan ifadeleri greceđiz.

Decision Tree altında alıőan ikinci mekanizmayı anlatmaya baőlayalım. Burada ilk olarak yukarıda bahsettiđim gibi cinsiyet, yaő, medeni durum, eđitim durumu ve meslek bilgilerini barındıran karar ađacını ızdım. Aőađıdaki tabloda ncelikle biz yani kullanıcılar bayan mı erkek mi olduđunu belirtiyoruz. Tabi istenirse cinsiyet girilmeden diđer zellikler de girilip program alıőır. Ancak bunun sonucunda őyle bir durum meydana gelir. Karőımıza gelecek olan ekranda rnlerin bayan mı erkeklerin mi aldıđı bilinemez. Ancak istenirse aőađıdaki tablo 5! Faktriyel deđiőik biimde tasarlanabilir. Yani ncelikle yaőta girilebilir veya medeni durum veya meslek veya eđitim durumu. Ancak en mantıklısı baőlangıta mőterinin cinsiyetinin belirtilmesi bizim faydamızdır. Yani bu grupta 120 farklı şekilde yapılabilirdi.





İlk olarak ürün seçimi kısmını anlatacağımdan yukarıda bahsetmişim. Aşağıdaki tabloda örnek müşteri tablosunu listeledim. Burada müşterilerin numaraları, isimleri, cinsiyet bilgileri, yaş, medeni durum, eğitim durumu ve meslekleri listelendi. Şimdi hazırlamış olduğumuz ara yüzden ürün seçimini “Bahçıvan Kaşar olarak yapıyorum ve bu ürünü alan müşterilerin cinsiyet, medeni durum ve eğitim durumlarına göre listelenmesini istiyoruz.”

The screenshot shows the 'Associative Data Mining and Statistics' software interface. The window title is 'Associative Data Mining and Statistics'. The interface is divided into several sections:

- Associative Data Mining:** Contains 'Support' buttons (1, 2, 3) and a dropdown menu set to '1'.
- Statistics:** Contains a 'Percentage %' dropdown menu set to '10' and a 'Most Valuable Customers' button.
- Decision Tree:** Contains checkboxes for 'Cinsiyet', 'Yaş', 'Medeni Durum', 'Eğitim Durumu', and 'Meslek'. 'Cinsiyet', 'Medeni Durum', and 'Eğitim Durumu' are checked. Below each checkbox is a dropdown menu.
- Ürün:** Contains a dropdown menu showing 'Bahçıvan Kaşar' and a search button.

A large grey area in the center of the window is reserved for the results table.

#### Şekil 45. Üründen Müşteri Profili

Yukarıda görüldüğü gibi cinsiyet, medeni durum ve eğitim durumu kutucuklarına doldurulmuştur. Daha sonrada ürün listeden seçilmiştir

Şekil 46. Müşteri Tablosu

Musteri_Num	Musteri_Ad	Musteri_Soyad	Cinsiyet	Yas	Medeni_Durum	Egitim_Durum	Meslek
6	Havva	KÖSE	Bayan	1971	Bekar	Danışmanlık	Üniversite
7	Hüseyin	AK	Erkek	1964	Evli	Eğitim	Üniversite
8	Meliha	KUK	Bayan	1950	Evli	Eğlence	Lise
9	Muhammet	PALA	Erkek	1988	Bekar	Öğrenci	Lise
10	Murat	AKYILDIZ	Erkek	1978	Evli	Elektrik/Elektronik	Üniversite
11	Neriman	ÖZGÜLEN	Bayan	1962	Evli	Halkla İlişkiler	Üniversite
12	Recep	SARGUN	Erkek	1943	Evli	Emekli	Üniversite
13	Ali Rıza	HANEDAR	Erkek	1976	Bekar	Giyim Sanayi	Lise
14	Ferat	BARMAK	Erkek	1983	Bekar	Hukuk	Üniversite
15	İsmail	METİN	Erkek	1959	Evli	İlaç Endüstrisi	Üniversite
16	Yusuf	HASANÇEBİ	Erkek	1962	Evli	İnşaat	Lise
17	Betül	GÖKÇE	Bayan	1983	Bekar	Sekretarya	Lise
18	Muhammet	KARALI	Erkek	1961	Evli	İthalat/İhracat	Meslek Yüksek Okulu
19	Naci	ÇİVELEK	Erkek	1951	Evli	Kozmetik	Lise
20	Yusuf Ziya	KATMER	Erkek	1958	Evli	Mağazacılık	İlköğretim
21	Erdem	KATIRCI	Erkek	1970	Bekar	Medya	Doktora
22	İsmail	UZUNHÜSEYİNOĞLU	Erkek	1980	Bekar	Öğrenci	Üniversite
23	Mehmet	BAYRAM	Erkek	1984	Bekar	Öğrenci	Üniversite
24	Muhammet	AKYILDIZ	Erkek	1986	Bekar	Spor	Lise
25	Necmettin	DEMİRBAŞ	Erkek	1983	Bekar	Mimarlık	Lisansüstü

Şekil 47. Ürün Tablosu

Urun_Ad	Urun_Ozellik	Urun_Fiyat
Migros Sele Zeytin	450g Baharatlı	2,49
Bahçivan Kaşar	Dilimli Light 225g	3,99
Migros Light Karışık Nektar	1 lt	1,19
Filiz Tagliatelle- Fettucine-Linguine	250g	1,45
Maret Mangal Sosis	300g	3,19
Tabasco Garlic Sos	60ml	5,49
Tat Mantar	340g	2,99
Pınar Ekstra Light Süt	1 lt	1,59
Ruf Rus Pastası	700g	4,25
Bosphorus Akriik Tuzluk Biberlik		11,9
Tat Kurutulmuş Domatez	200g	1,99
Maret Boncuk Sucuk	Acılı 250g	5,99
DANONE DOĞAL SÜT	1 KG. 4'LÜ	6,75
NESCAFE GOLD	200 GR	11,29
SESU SIRLI ILIK AGDA	360 GR	7,65
YUMOS	4 KG LAVANTA	6,59
DOVE 2 LI BODY FIRMING JEL	200 ML	16,35
SÜTAŞ KOYUN PEYNİRİ	650 GR	8,99
SELPK HAVLU	6'LI	5,29
YÖRSAN TAVA YOĞURT	2 KG	5,25

## 7.5. Üründen Müşteri Profili

Aşağıdaki tabloda “Bahçivan Kaşar ürünü , satın almış olduğu ürünler arasında olan müşteriler listelenmiştir.Şimdi bizim yapacağımız bu tablodan “Bahçivan Kaşar” ürününü hangi cinsiyet,medeni durum ve eğitim durumundaki insanların tercih ettiklerini, yüzde olarak ifade etmektir.

Şimdi üç duruma göre hesaplama yapacağım. Birincisi sadece cinsiyet kutucuğu seçilirse,ikinci kısımda ise hem cinsiyet hem medeni duruma göre üçüncüsünde ise cinsiyet, medeni durum ve eğitim durumlarına göre yüzdeler çıkarılacaktır.

### 7.5.1.Birinci Hesaplama

1.Toplam 9 müşterimiz var. Bunlardan 3/9’u bayan,6/9 ‘u ise erkektir.

Bayan 3/9

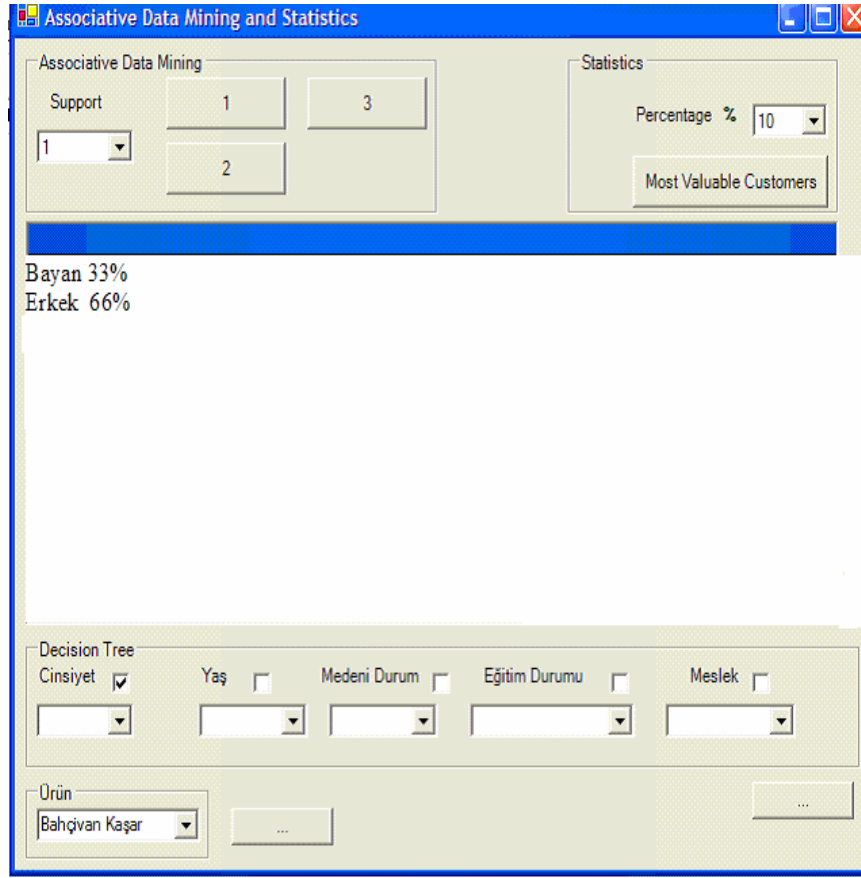
Erkek 6/9

### Algoritmanın Yazılımı

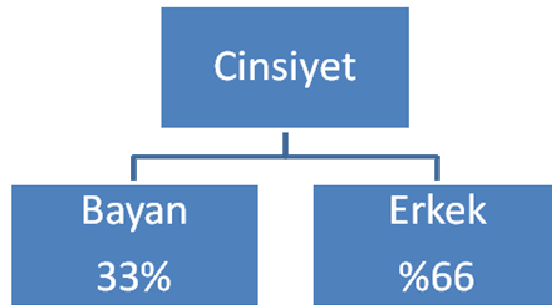
IF “Bahçivan Kaşar” THEN Erkek=”33 percent”

IF “Bahçivan Kaşar” THEN Bayan=”66 percent”

Aşağıdaki arayüzde görüldüğü üzere sadece cinsiyet kutucuğu işaretlenmiştir.



Şekil 48. Birinci Hesap ta Oluşacak Olan Karar Ağacı



## 7.5.2. İkinci Hesaplama

1. Erkek Evli oranı 3/9

Erkek Bekar oranı 3/9

2. Bayan Bekar oranı 2/9

Bayan Evli oranı 1/9

Bunu daha sonra hazırladığımız arayüzde %yüzde şeklinde göreceksiniz.

## Algoritmanın Yazılımı

IF “Bahçivan Kaşar” THEN Erkek&Evli=”33 percent”

IF “Bahçivan Kaşar” THEN Erkek&Bekar=”33 percent”

IF “Bahçivan Kaşar” THEN Bayan&Bekar=”22 percent”

IF “Bahçivan Kaşar” THEN Bayan&Evli=”11 percent”

Aşağıdaki arayüzde görüldüğü üzere cinsiyet ve medeni durum kutuları doldurulmuştur.

Associative Data Mining and Statistics

Associative Data Mining

Support 1 3

1

2

Statistics

Percentage % 10

Most Valuable Customers

1. Erkek Evli %33  
Erkek Bekar %33  
2. Bayan Bekar %22  
Bayan Evli %11

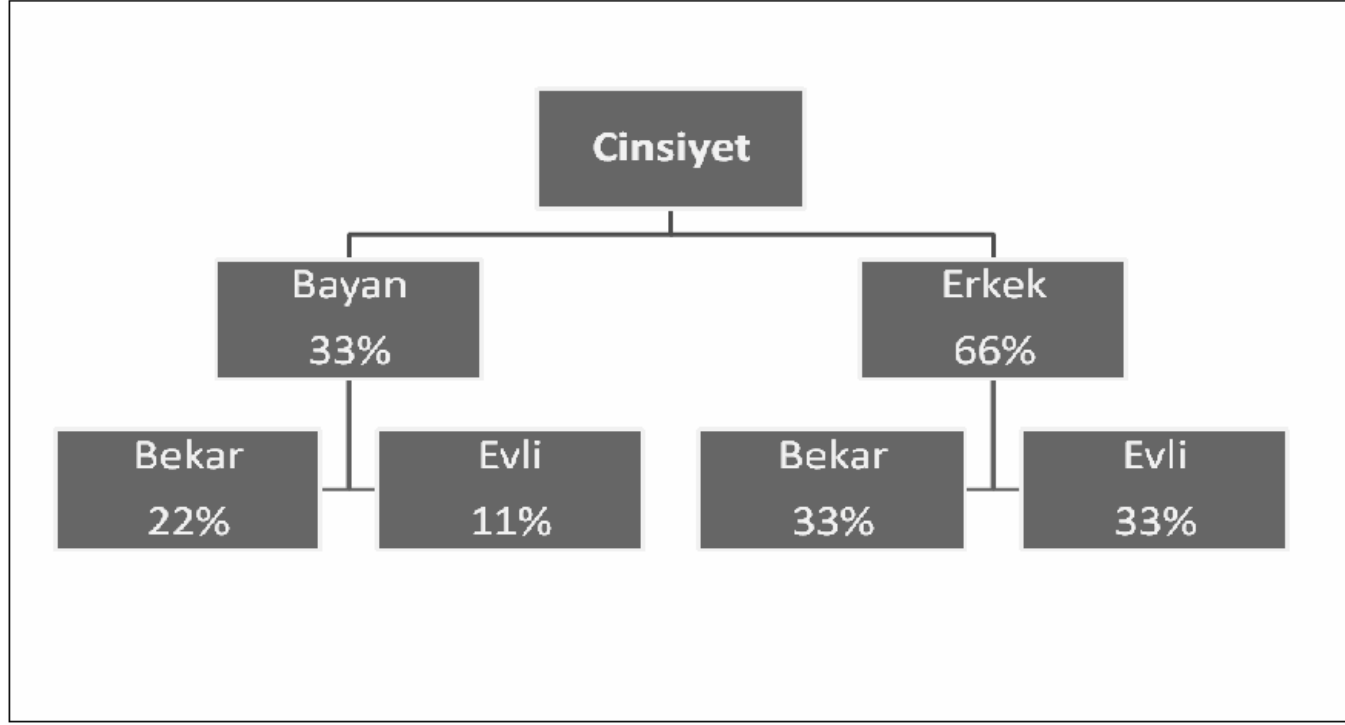
Decision Tree

Cinsiyet  Yaş  Medeni Durum  Eğitim Durumu  Meslek

Ürün

Bahçivan Kaşar

Şekil 49. İkinci Hesapta Oluşacak olana Karar Ağacı





### 7.5.3. Üçüncü Hesaplama

Evli Erkek ve Üniversite mezunu 2/9

Evli Erkek ve Meslek Yüksek Okulu mezunu 1/9

Bekar Erkek ve Üniversite mezunu 1/9

Bekar Erkek ve Lise mezunu 1/9

Bekar Erkek ve Lisansüstü mezunu 1/9

Evli Bayan ve Üniversite mezunu 1/9

Bekar Bayan ve Üniversite mezunu 2/9

Aşağıdaki ara yüzde görüldüğü üzere sadece cinsiyet, medeni durum ve eğitim durumu kutucukları işaretlenmiştir.

### Algoritmanın Yazılımı

IF “Bahçivan Kaşar” THEN Erkek&Evli&Üniversite mezunu=”22 percent”

IF “Bahçivan Kaşar” THEN Erkek&Evli&Meslek Yüksek Okulu mezunu=”11 percent”

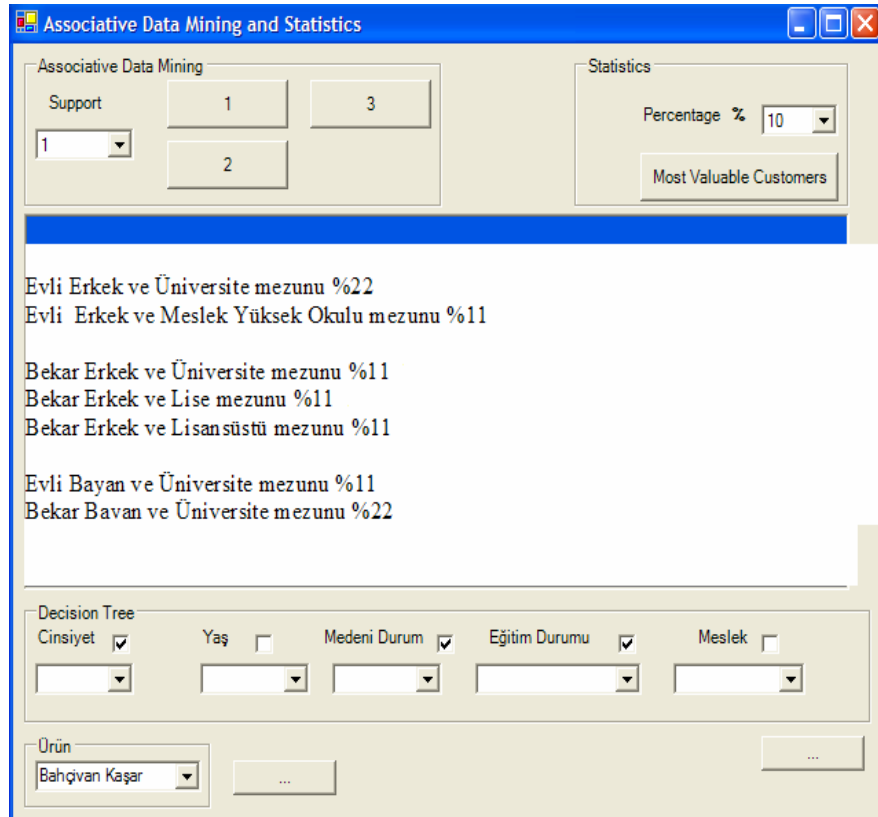
IF “Bahçivan Kaşar” THEN Erkek&Bekar&Üniversite mezunu=”11 percent”

IF “Bahçivan Kaşar” THEN Erkek&Bekar&Lise mezunu=”11 percent”

IF “Bahçivan Kaşar” THEN Erkek&Bekar&Lisansüstü mezunu=”11 percent”

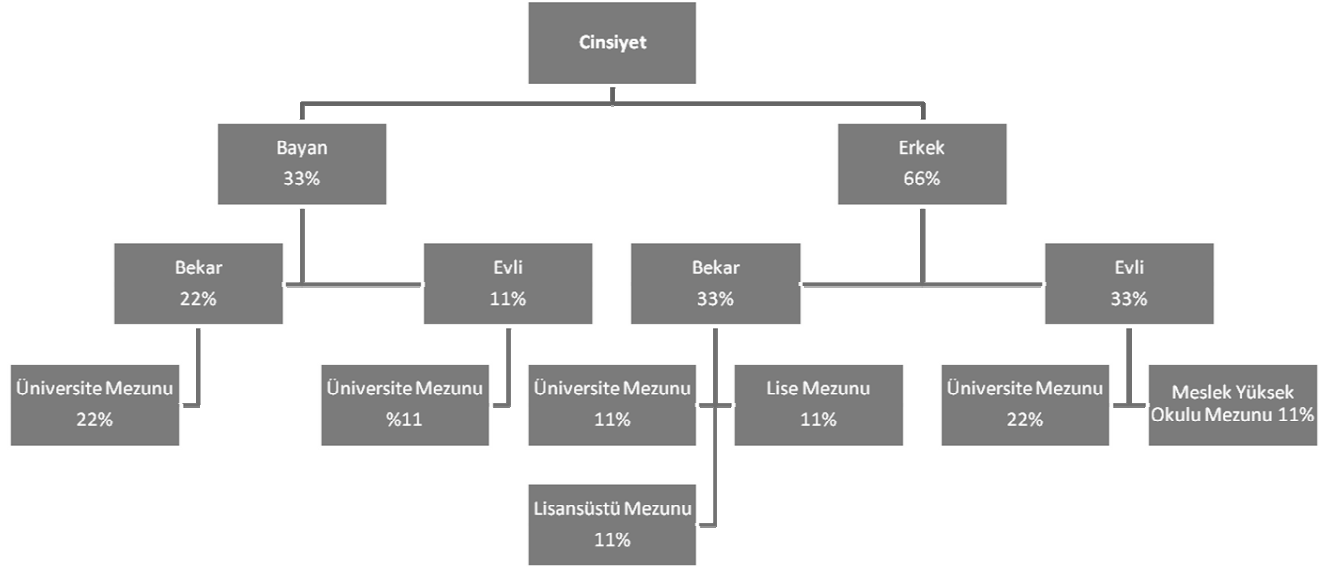
IF “Bahçivan Kaşar” THEN Bayan&Evli&Üniversite mezunu=”11 percent”

IF “Bahçivan Kaşar” THEN Bayan&Bekar&Üniversite mezunu=”22 percent”



**Şekil 50. Ürün Profiline Cinsiyet, Yaş, Medeni Durum Değerlendirmesi**

Şekil 51. Üçüncü Hesapta Oluşacak Olan Karar Ağacı



Sip_No	Müsteri_Nosu	Ürün1	Ürün2	Ürün3
101	6	Bahçivan Kaşar	Tabasco Garlic Sos	Tat Kurutulmuş Domatez
102	7	Tat Mantar	Ruf Rus Pastası	Tat Kurutulmuş Domatez
103	8	Tat Kurutulmuş Domatez	Pınar Ekstra Light Süt	Bahçivan Kaşar
104	9	Maret Boncuk Sucuk	YÖRSAN TAVA YOĞURT	Bosphorus Akriklik Tuzluk Biberlik
105	10	NESCAFE GOLD	Bahçivan Kaşar	Migros Sele Zeytin
106	11	DOVE 2 LI BODY FIRMINING JEL	Pınar Ekstra Light Süt	Bahçivan Kaşar
107	12	SÜTAŞ KOYUN PEYNİRİ	Tabasco Garlic Sos	Bahçivan Kaşar
108	13	Pınar Ekstra Light Süt	Tat Mantar	Pınar Ekstra Light Süt
109	14	DANONE DOĞAL SÜT	YUMOS	DOVE 2 LI BODY FIRMINING JEL
110	15	YÖRSAN TAVA YOĞURT	SELPK HAVLU	SÜTAŞ KOYUN PEYNİRİ
111	16	YUMOS	DANONE DOĞAL SÜT	DOVE 2 LI BODY FIRMINING JEL
112	17	Tat Kurutulmuş Domatez	Tabasco Garlic Sos	DANONE DOĞAL SÜT
113	18	Bahçivan Kaşar	Pınar Ekstra Light Süt	DANONE DOĞAL SÜT

114	19	Pınar Ekstra Light Süt	YÖRSAN TAVA YOĞURT	DOVE 2 LI BODY FIRMLNG JEL
115	20	Tat Kurutulmuş Domatez	SELPAK HAVLU	DANONE DOĞAL SÜT
116	21	Tat Mantar	Pınar Ekstra Light Süt	SÜTAŞ KOYUN PEYNİRİ
117	22	Pınar Ekstra Light Süt	SELPAK HAVLU	YUMOS
118	23	Maret Boncuk Sucuk	Bahçivan Kaşar	SELPAK HAVLU
119	24	YÖRSAN TAVA YOĞURT	Maret Boncuk Sucuk	Bahçivan Kaşar
120	25	YÖRSAN TAVA YOĞURT	Bahçivan Kaşar	Tat Mantar
121	13	Tat Kurutulmuş Domatez	Pınar Ekstra Light Süt	Pınar Ekstra Light Süt
122	8	Pınar Ekstra Light Süt	YÖRSAN TAVA YOĞURT	YUMOS
123	6	Tat Mantar	DANONE DOĞAL SÜT	Bahçivan Kaşar
124	9	Ruf Rus Pastası	Tat Kurutulmuş Domatez	YÖRSAN TAVA YOĞURT
125	14	YÖRSAN TAVA YOĞURT	SÜTAŞ KOYUN PEYNİRİ	Tat Mantar
126	19	SELPAK HAVLU	Maret Boncuk Sucuk	YÖRSAN TAVA YOĞURT

**Şekil 52. Müşterilerin Sipariş Ettiği Ürün Tablosu**

## 7.6. Müşteriden Ürün Bulunması

Burada ise müşterinin profiline göre aldığı ürünler sıralanacaktır.Örneğin yukarıdaki tablodan hangi ürünün kaç defa satın aldığı belirlenebilir.Daha sonra bunun yüzdesi verilebilir.

Bahçivan Kaşar	Tabasco Garlic Sos	Tat Kurutulmuş Domatez
Tat Kurutulmuş Domatez	Pınar Ekstra Light Süt	Bahçivan Kaşar
DOVE 2 LI BODY FIRMINING JEL	Pınar Ekstra Light Süt	Bahçivan Kaşar
Tat Kurutulmuş Domatez	Tabasco Garlic Sos	Bahçivan Kaşar

### Şekil 53. Müşteriden Ürün Bulunma Sonucu Oluşan Tablo

İlk olarak bayan müşterilerimizin hangi ürünleri yüzde olarak tercih ettiğini bulalım.

Bahçivan Kaşar toplam sayısı	:4
Tabasco Garlic sos	:2
Tat Kurutulmuş Domatez	:3
Pınar Ekstra Light Süt	:2
Dove 2 li Body Firming Jel	:1

Yani bayanların alımına göre yüzde tercihleri;

Toplam ürün adedi:12;

Bayan için alınma oranları

Bahçivan Kaşar	4/12
Tabasco Garlic sos	2/12
Tat Kurutulmuş Domatez	3/12

Pınar Ekstra Light Süt 2/12

Dove 2 li Body Firming Jel 1/12

### Algoritmanın Yazılımı

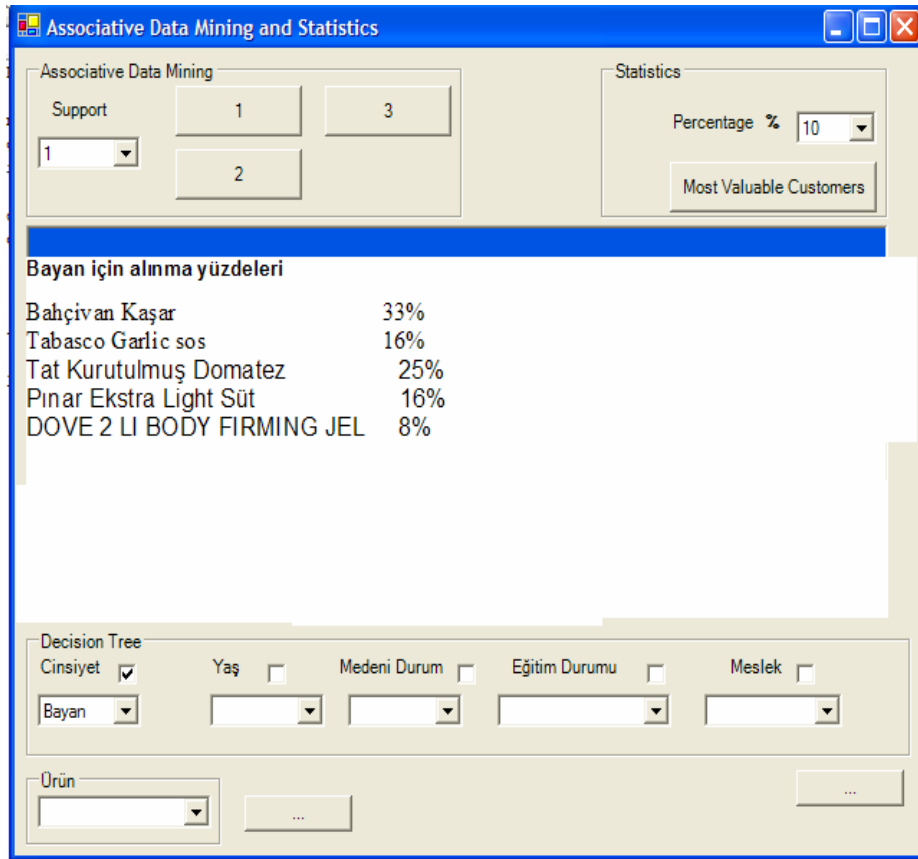
IF Bayan THEN buys\_ Bahçivan Kaşar="33 percent"

IF Bayan THEN buys\_ Tabasco Garlic sos="16 percent"

IF Bayan THEN buys\_ Tat Kurutulmuş Domatez="25 percent"

IF Bayan THEN buys\_ Pınar Ekstra Light Süt="16 percent"

IF Bayan THEN buys\_ Dove 2 li Body Firming Jel="8 percent"



### Şekil 54. Sadece Cinsiyet Durumuna Göre Listeleme

Yukarıda sadece bayan müşteriler için ürün seçimi yapılmıştır. Şimdi hem bayan hem de bekâr olan müşterilerimizin yapmış oldukları tercihlere bakalım.

Hem bekâr hem de bayan olan müşterilerimizin aldıkları ürünler;

Tat Kurutulmuş Domatez	Pınar Ekstra Light Süt	Bahçivan Kaşar
Dove 2 li BODY Firming Jel	Pınar Ekstra Light Süt	Bahçivan Kaşar

Görüldüğü üzere Toplam olarak 6 üründen dağılım;

Tat Kurutulmuş Domatez toplam satım :1  
Pınar Ekstra Light Süt :2  
Bahçivan Kaşar :2  
Dove 2 LI BODY FIRMING JEL :1

Alınma oranları;

Bayan Bekar için alınma oranları

Tat Kurutulmuş Domatez :%16  
Pınar Ekstra Light Süt :%33  
Bahçivan Kaşar :%33  
DOVE 2 li Body Firming Jel :%16

#### **Algoritmanın Yazılımı**

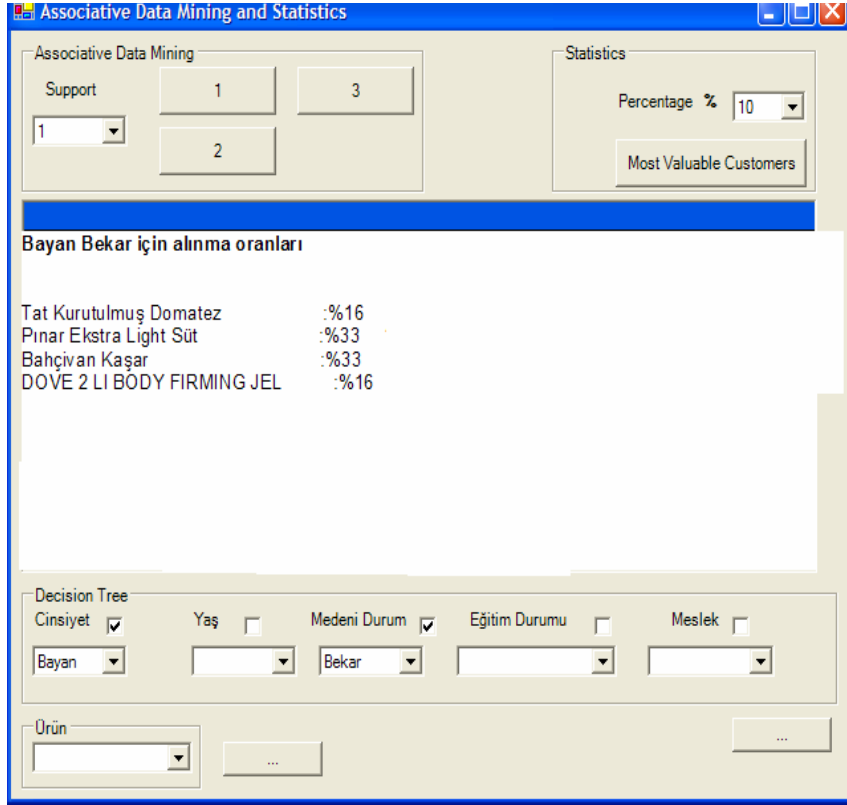
IF Bayan and Bekar THEN buys\_Tat Kurutulmus Domatez="16 percent"

IF Bayan and Bekar THEN buys\_Pınar Ekstra Light Süt="33 percent"

IF Bayan and Bekar THEN buys\_Bahçivan Kaşar="33 percent"

IF Bayan and Bekar THEN buys\_DOVE 2 LI BODY FIRMING JEL="16 percent"





### Şekil 55. Cinsiyet ve Medeni Durum İrdelenmesi

Yukarıda ki tabloda görüldüğü üzere cinsiyet ve medeni durum işaretlenmiştir.

## 8. SONUÇ

Veri madenciliğinin ve olasılıksal sinir ağlarının uygulanabilmesi için yığın halinde verilerin elimizde bulunması vazgeçilmez bir koşuldur. Veri madenciliği farklı formatlarda çok sayıda kütükte yığın halindeki veriler arasında gizli bir şekilde bulunan mesajları çekip çıkarmamıza yarayan bir araçtır. Veri madenciliği çeşitli açılardan geleneksel istatistiksel yöntemlerle önemli farklılıklar gösterir. Özellikle zaman içinde verinin azlığının değil, çokluğunun bir sorun olması ve bilgisayarların veri saklama ve işleme hızlarındaki inanılmaz artışların sonucunda veri madenciliğinin güncelliği her geçen gün artmış ve artmaktadır. Veri madenciliğinde kullanılmak üzere verilerin ön işlemeden geçirilmesi aynı anlama gelmek üzere verilerin veri madenciliğinde için hazır duruma getirilmesi veri madenciliğinin en önemli aşamalarındandır.

Özetlemek gerekirse; Veri madenciliği olayının sınırı yoktur. Aslında ilk bölüm ikinci bölümün hazırlanmasına temel hazırlamıştır. Decision Tree mekanizmasına temel oluşturduğunu göreceğiz projenin. Yani markete gelen müşterilere aynı Migros 'da benzeri bir indirim kartı verilebilir. İndirim kartında asıl amaç müşterilerin kartı alabilmek için başlangıçta doldurdıkları form içindir. Bu sayede MİGROS yöneticileri müşterilerinin kişisel bilgileri hakkında bilgi sahibi olabilmektedir.

Örneğin bu formda müşterilerden Bayan/Erkek, yaş, doğum yeri, eğitim durumu ve mesleği bilgileri istenebilir. Bu sayede müşterilerin yaklaşık profili çıkarılır ve hangi müşterini n neyi ne zaman daha çok tercih ettiği belirlenebilmektedir. Bu sayede örneğin 9 ila 12 saatleri arasında 20 ila 25 yaş arası bayanlar en fazla alışveriş yapmış ve aldıkları ürünler genelde kozmetik ürünleriymiş. Bu sayede MİGROS yönetimi bu saat dilimleri arasında kozmetik de tanıtım için bir bayan koyabilir ve satışları bir nevi arttırmayı sağlamış olur.

Bu çalışmadan görüldüğü üzere yığın veriler üzerinde yapılacak veri madenciliği ve sınıflandırma yöntemleri ve çalışmaları ile, müşteri odaklı pazarlarda faaliyet gösteren işletmelerin pazarlama stratejilerini oluşturmada ve rekabetçi üstünlük sağlamada önemli bir karar destek aracı haline gelmiştir.

## 9. KAYNAKLAR

KARLIK,Bekir; **Yapay Sinir Ağları Dersi Ders Notları** , Bilgisayar Mühendisliği Yüksek Lisans Programı Ders Kaynak Notları , T.C.Haliç Üniversitesi, İstanbul, 2004

S.P. Banks ; **Signal Processing, Image Processing and Pattern Recognition**, Prentice Hall, 1990.

Y.H. Pao ; **Adaptive Pattern Recognition and Neural Networks**, Addison Wesley,1989.

R.O. Duda, P.E. Hart ; **Pattern Classification and Scene Analysis**, Wiley, 1973.

AGRAWAL,Ragesh ; **“Fast Algorithms for Mining Association Rules”**, Ramakrishnan Srikant IBM Almaden Research Center

VAHAPLAR,Alper ;**Veri Madenciliği ve Elektronik Ticaret** , Dr.Mustafa Murat İnceoğlu Data Mining: An Overview

W.SEİFERT, Jeffrey ; **Analyst in Information Science and Techonology Policy Resources,Science,and Industry Division**

BERRY,J. A. ,LINOFF G.; **Data Mining Techniques: For Marketing, Sales, and Customer Support**, John Wiley & Sons Inc.,1997

SPECHT, D. F. , **“Probabilistic Neural Networks”**, *Neural Networks*, Vol. 3, pp. 109-118, 1990.

## 10. ÖZGEÇMİŞ

8 Kasım 1981 'de İstanbul'da doğdum. 1989–1997 yılları arasında ilk ve orta öğrenimimi Kdz. Ereğli'de tamamladım. 1997–2000 Yılları arasında Ted Kdz. Ereğli Koleji Vakfı Özel Lisesinin Fen Bölümünde eğitim aldıktan sonra 2000–2004 yılları arasında Bahçeşehir Üniversitesi Bilgisayar Mühendisliği Bölümünde eğitim gördüm. 2004–2007 tarihleri arasında Haliç Üniversitesinde Bilgisayar Mühendisliği Yüksek Lisans programında yaptım. Şu an halen Haliç Üniversitesi'nde Okutman Kadrosunda görev almaktayım.

S.SERKAN GÜLLÜOĞLU