YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

WEB KULLANIM MADENCİLİĞİNDE BİRLİKTELİK KURALLARININ UYGULANMASI

Bilgisayar Müh. Hikmet Mukadder OKTAY

FBE Bilgisayar Mühendisliği Anabilim Dalı Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. Songül ALBAYRAK

İSTANBUL, 2009

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

ÖNSÖZ

Dünya üzerindeki en hızlı, en büyük ve sınırları tahmin edilemeyen bilgi ve veri kaynağı, şüphesiz ki Internet'tir. Internet dünyasında, her gün binlerce yeni web sayfasının kullanıma açıldığı, binlerce yeni makalenin paylaşıldığı, sayısı bilinemeyen soruların sorulup cevaplarının alındığı düşünülürse, bu dünyadaki veri miktarının ne kadar büyük olduğu ve olacağı göz ardı edilemez bir gerçek olarak kabul edilmelidir.

Internet dünyasındaki verinin büyüklüğü, bu verilerin hepsinin faydalı, kullanılabilir bilgi olduğu anlamına gelmemektedir. Verilerin içinden anlamlı bilgileri çıkarmak, bu bilgileri çeşitli istatistiklerde kullanmak, verileri işledikten sonra ortaya çıkan anlamlı sonuçlardan yola çıkarak çeşitli güncellemeler yapabilmek için yığın halinde duran bu anlamsız verileri anlamlı, kullanılabilir hale getirmek gerekmektedir. Verilerden anlamlı, kullanılabilir, faydalı veri üretmek için kullanılan disiplin veri madenciliğidir. Veri madenciliği algoritmalarının internet ortamındaki kaynaklar üzerinde uygulandığı disiplin de web madenciliğidir.

Bu tez çalışmasında ele alınan temel konu, web madenciliğinin üzerinde en çok çalışma yapılan, ilgi çeken ve kullanılan çeşidi olan web kullanım madenciliğidir. Öncelikle veri madenciliğinin ne olduğu, veri madenciliğinde kullanılan teknikler ele alınmış olup, daha sonra web madenciliği ve türleri, internet ortamında kullanılan log dosyalarının yapısı ve tez çalışmasının asıl konusu olan web kullanım madenciliğinin aşamaları ele alınmıştır. Son olarak da web kullanım madenciliğinin uygulanacağı veri seti üzerinde bir veri madenciliği algoritması olan Apriori algoritması ile birkaç bilim adamının web sayfalarında bulunulan süre ve erişim sıklığı bilgisini alarak PageRank algoritmasına değişik bir bakış açısı getirdiği TFPR algoritması uygulanarak, sonuçlara ve kurallara ulaşılmıştır.

Yüksek lisans eğitimim boyunca verdikleri eğitimle bana yol gösteren Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümündeki değerli hocalarıma,

Hiçbir zaman desteğini eksik etmeyen ve bana hep güvenerek tez çalışmamı bitirmemde büyük yardımları olan tez danışmanım Sayın Hocam Yrd. Doç. Dr. Songül ALBAYRAK'a,

Telif haklarından dolayı ismini bu çalışmada yazamadığım; ancak bu çalışmada kullandığım veri setini bana sağlayan kuruma,

Tez çalışmam süresince benden bilgisini ve desteğini esirgemeyen Bilgisayar Öğretmeni arkadaşım Erhan AKPINAR' a,

Ve çocukluğumdan bu zamana kadarki gelişimimde, eğitim hayatımda ve özel hayatımda, kısacası her anımda yanımda olan ve benden hiçbir desteğini esirgemeyen, benim için en önemli ve değerli varlıklar olan canım anneme, babama ve kardeşime teşekkür ediyorum.

ÖZET

WEB KULLANIM MADENCİLİĞİNDE
BİRLİKTELİK KURALLARININ UYGULANMASI

Hikmet Mukadder OKTAY

Bilgisayar Mühendisliği, Yüksek Lisans Tezi

Bilgisayar ve teknolojinin hayatımızın her anında bizimle birlikte olması, ihtiyaç duyulan her türlü bilginin bilgisayar ortamında aranmasına ve saklanmasına sebep olmaktadır. Aranan bilginin bulunabileceği en derin kaynak ise Internet' tir. İçeriği katlandıkça artan bu bilgi haznesindeki verilerin kullanılabilir kısmı son kullanıcı için anlamlıyken, bu kullanılabilir verilerin arkasındaki saklanmış, anlamsız veriler ise işlenmeye hazır olarak beklemektedir. Veri madenciliği de bu ihtiyacı karşılamak üzere ortaya atılan bir disiplindir. Veri madenciliği birliktelik kuralları, sınıflandırma, kümeleme gibi yöntemlerle anlamsız verilerden anlamlı, faydalı bilgileri çıkararak; bu faydalı bilgilerin daha sonra alınacak kararlarda, yapılacak iyileştirme çalışmalarında kullanılmasına rehberlik etmektedir.

Internet' in her yerden erişilebilmesi ve istenilen her türlü bilgiyi insanlara sağlayabilmesi, gözlerin web sitelerine çevrilmesine neden olmuştur. Web sitesi yöneticileri rekabet ortamında, site içeriği ve teknik altyapı açısından da web sitelerinin güçlü olması için çalışmalar yapmaktadır. Bu noktada, web sitelerindeki anlamsız verilerden faydalı bilgiyi üretmek için web madenciliği kavramı ortaya çıkmıştır. Web madenciliği, web sitesi verilerine veri madenciliği yöntemlerinin uygulanması işlemidir. Web madenciliğinin üç çeşidi olan "web kullanım madenciliği", "web içerik madenciliği" ve "web yapı madenciliği" bölümlerinden en çok üzerinde çalışma yapılan ve bu tez çalışmasının da konusu olan web kullanım madenciliğidir. Web kullanım madenciliği ile siteyi ziyaret eden kullanıcıların profillerinin değişimleri takip edilebilir. Ayrıca sitede beğenilen ya da beğenilmeyen köşeler ortaya çıkarılarak site tasarımı buna göre güncellenerek daha çok kullanıcının web sitesini ziyaret etmesi sağlanabilir.

Bu tez çalışmasındaki amaç, çok ziyaret edilen, güncel bir portal sitesinden alınan kullanıcı log kayıtları üzerinde web kullanım madenciliği yöntemini uygulayarak, saklı bilgileri ortaya çıkarmaktır. Bunu gerçekleştirmek için, veri madenciliği yöntemlerinden olan birliktelik analizinin en sık kullanılan algoritması Apriori algoritması ile bazı bilim adamları tarafından ortaya atılan TFPR algoritması seçilmiştir. Apriori algoritması ile sonuçlara ve kurallara ulaşılmış, TFPR algoritması ile de en sık ziyaret edilen ve edilme olasılığı en yüksek olan sayfalar bulunmuştur.

Anahtar Kelimeler: Web kullanım madenciliği, Web madenciliği, Veri madenciliği, Apriori algoritması, Log dosya formatları,TFPR algoritması.

ABSTRACT

IMPLEMENTATION OF ASSOCIATION RULES IN WEB USAGE MINING

Hikmet Mukadder OKTAY

Computer Engineering, Master Thesis

Information that is needed is found and recorded in a computer environment because of computers and technology are always with us in our lives. The deepest source where the information being searched can be found at, is Internet. While useful part of data in this information source that has a content increasing many times more is meaningful for the end user; hidden and meaningless data behind this useful data is ready to be processed. Data mining is a discipline that meets this need. Data mining guides to use this useful data for making decisions and improving workings by finding out meaningful, useful data from meaningless data using the methods like association rules, classification and clustering.

Accessing Internet from everywhere and Internet supplying every kind of wanted information to people, caused people' eyes turning to web sites. In a competition environment, web site administrators are working to make their web sites to be more powerful with the site content and the technical background. At this point, to produce meaningful data from meaningless data in web sites, the term, web mining arised. Web mining is a process of applying data mining techniques to web sites data. Web usage mining is the one that the most workings are being done about it and it is the subject of this thesis among the three types of web mining; web usage mining, web content mining and web structure mining. With web usage mining, user profile changes of who visits the web site can be tracked. Also, by finding the liked and disliked parts of the web site, web site design can be improved to make more users visiting the web site.

The purpose of this thesis is to find out the hidden information by using web usage mining at the user log records of a live, frequently visited portal site. To achieve this, the most frequently used algorithm of one of the techniques in data mining, that is the association rules, Apriori algorithm and TFPR algorithm that was proposed by a few scientists were chosen. By applying Apriori algorithm, rules and results were discovered and by TFPR algorithm, the most frequent visited and the most probable being visited pages were found.

Keywords: Web usage mining, Web mining, Data mining, Apriori algorithm, Log file formats, TFPR algorithm.

1.GİRİŞ

Bilgisayarın yaşamımızın her alanına girmesiyle birlikte, bilgisayar ve teknolojideki gelişmelere paralel olarak, bilgisayar ve teknoloji piyasasındaki rakip firmaların artması ve kullanılan donanımsal ürünlerin de ucuzlamasıyla beraber oluşan tüm veriler uzun süre depolanabilmekte, dolayısıyla da çok büyük kapasiteli veritabanların oluşmasına neden olabilmektedir; ancak bu veritabanlarındaki veriler, işlenmeye hazır halde olan veriler değildir. Bu nedenle büyük veritabanlarında istenilen anlamlı, kullanılabilir ve ilginç bilgiye erişmek için yeni bir disiplin ihtiyacı doğmuştur. Bu disiplin, veri madenciliği ismini almıştır.

Bilgisayarların yaşamımıza daha çok girmesiyle beraber, artık her yaptığımız işlem dijital ortamlarda kayıt altına alınmaya başlandı. En basitinden marketlerde yaptığımız alışverişlerde aldığımız herbir ürün, hangi ürünle beraber hangi ürünleri aldığımız, aldıktan sonra iade ettiğimiz ürünler veritabanlarında tutulmaktadır. Bankalarda, hastanelerde ve daha birçok yerde yaptığımız işlemler kayıt altına alınmakta ve gerektiği zaman da işlenerek kullanılabilmektedir. Kayıt altına alınan veriler sadece metinsel olarak değil, görsel olarak da olabilmektedir. Alışveriş merkezlerine ya da mağazalara gelen müşterilerin mağaza içindeki tüm görüntüleri kayıt altına alınmaktadır. Emniyet müdürlüklerinde ehliyet ya da pasaport gibi işlemler yaptırılmaya gidildiğinde, gerektiğinde kullanılmak ve herhangi bir suç olayında suçluyu araştırmak üzere kullanılacak veritabanına parmak izleri kaydettirilmektedir. Bütün bu veriler veritabanlarında dururken, diğer taraftan da bu verilerden ihtiyaç duyulan, anlamlı bilgiyi çıkartmak için işlemlerin yapılması gerektiği aşikardır.

Veri madenciliği, büyük miktardaki veriden anlamlı bilginin çıkarılması ile ilgili yeni ve güncel bir teknik olup, pazarlama, bankacılık, sigortacılık, tıp, asayiş sektörü başta olmak üzere birçok sektörde etkin şekilde kullanılmaktadır. Veri madenciliği uygulamalarından biri olan ve bu tez çalışmasının asıl konusu olan web madenciliği ise web verileri üzerinde veri madenciliği fonksiyonlarının uygulanmasıdır.

Web madenciliği terimi ilk kez 1996 yılında ortaya çıkmasına rağmen, günümüze kadar bu konu üzerinde yapılmış pek çok çalışma bulunmaktadır. Web madenciliği konusu üzerinde veri madenciliği algoritmaları uygulanabileceği gibi, bilim adamları ve akademisyen kişiler kendi oluşturduğu algoritmaları da işlenecek veriye uygulayarak başarılı sonuçlara varabilmektedir.

2007 yılında Melbourne üniversitesindeki birkaç araştırmacı, bu tez çalışmasında da uygulanan algoritmalardan biri olan TFPR algoritmasını ortaya atmışlardır. Bu algoritma Google arama motorunun bir arama yapıldığında bulduğu web sayfalarını sıralamak için kullandığı PageRank algoritmasına benzer bir algoritmadır. Bu algoritmadaki amaç, web sayfalarının kullanıcıya getirilmesine kadar geçen süreyi minimuma indirmek ve bunun için de doğru web sayfa tahminlerinde bulunabilmektir. Bu algoritmada 2 temel ölçüt kullanılmaktadır. Bunlardan birisi sayfanın kullanıcılar tarafından ziyaret edilme sıklığı, diğeri ise sayfada geçirilen toplam süredir. Bu 2 ölçüt kullanılarak, hangi web sayfalarına daha çok istekte bulunabileceği belirlenebilmekte ve düşük değere sahip olan sayfaların sunucu belleğinde tutulmaması sağlanarak, ziyaret edilme olasılığı yüksek olan sayfalara bellekte yer açılması sağlanabilmektedir (Guo, Y., Ramamohanarao, K., Park , L., 2007).

2007 yılında Hefei Üniversitesindeki birkaç akademisyen, bir e-ticaret sitesini ziyaret eden kullanıcıların sitede bıraktığı web log kayıtları üzerinde "*Ant Colony Model*" algoritmasını uygulayarak, kullanıcıların bir sonraki ziyaret edeceği sayfayı büyük bir başarı oranı ile tahmin etmeyi başarmışlardır. Ant Colony Model algoritmasında, siteyi ziyaret eden kullanıcılar yapay karıncalar olarak düşünülmüştür. Karıncalar yiyecek kaynakları ve yuvaları arasında gidip gelirken takip ettikleri yol üzerinde kendilerine has bir salgı bırakmakta ve bu salgı izleri takip edildiğinde yiyecek kaynağı ile yuvaları arasındaki en kısa mesafe bulabilmektedir. İzleyecekleri yolu, bu salgı izlerinde oluşan salgı yoğunluklarına göre olasılık teorisine göre seçmektedirler. Karıncalar gibi, web kullanıcılarının sayfalar arasındaki geçişleri de olasılık teorisine göre olmakta ve kullanıcının gezinti davranışları dinamik olarak ortaya çıkarılabilmektedir (Ling, H., Liu, Y., Yang , S., 2007).

2008 yılında yapılan bir çalışmada da haber sitelerinde kullanıcıya gösterilecek haber başlıklarının seçimi için yeni bir algoritma geliştirilmiştir. Bu algoritma veri madenciliği tekniklerinden kümeleme ile web kullanım madenciliğini kullanarak, kullanıcıların ilgisini çekecek haber başlıklarının görüntülenmesini sağlamaktadır. Algoritmada kümeleme algoritmalarından K-Means algoritması var olan haber başlıklarına uygulanarak, haberleri kategorilere ayırmakta kullanılmaktadır. Böylece farklı kategorilerde yer alan haber başlıkları kullanıcının karşısına getirilmemektedir. Kümeleme işlemi bittikten sonra, elde edilen farklı kategorilerdeki haberlerin web log kayıtlarında bulunan kayıtlarına web kullanım madenciliği tekniği uygulanmıştır. Bu algoritmada da sayfayı ziyaret etme sıklığı ve sayfada geçirilen toplam süre ölçütleri hesaplanarak, bu değerlerin yüksek olduğu sayfaların haber başlıkları

içerisinde üst sıralarda yer alması sağlanmıştır (Ping, N., Jianxin, L., Xiaomin , Z., Keyan, R., 2008).

Bu tez çalışmasında web madenciliğinin bir türü olan web kullanım madenciliği ile ilgili olarak işlemler uygulanmıştır. Veriler üzerinde 2 farklı algoritma uygulanmıştır. Bunlardan birisi veri madenciliği tekniklerinden biri olan birliktelik kuralları yöntemlerinden Apriori algoritmasıdır. Diğeri ise Avusturalya'da bir üniversitedeki bilim adamlarının ortaya attığı TFPR algoritmasıdır. İşlenen veri kaynağı güncel bir portal sitesinin log dosyalarıdır.

2. WEB MADENCİLİĞİ (WEB MINING)

Web Madenciliği (Web Mining), web tarafında elde edilen dataya veri madenciliği (data mining) tekniklerinin uygulanarak, bu anlamsız, gizli ve çokça bilgiden; gerekli, yararlı ve işlenebilir bilgiyi ortaya çıkarmaktır.

Birçok araştırmacıya göre, web madenciliği terimi ilk kez 1996 yılında Etzioni tarafından ortaya atılmıştır. Etzioni'ye göre, web madenciliği veri madenciliği tekniklerini kullanarak World Wide Web'te (WWW) bulunan dosya ve servislerden otomatik olarak örüntü (pattern) bulmak ve öngörülmeyen bilgiye ulaşmak için yapılan tüm işlemleri kapsar (Etzioni, O., 1996).

World Wide Web, çeşitli formatlarda bulunan birçok tipteki bilgiyi içeren çok büyük bir kaynaktır. Araştırmacılar bu dağınık web veri deposundaki insan davranışlarını araştırmaya ve sanal çevrelerdeki insan davranışlarını anlamak için çeşitli modeller üretmeye başladılar. Veri madenciliği teknikleri, Internet ortamına uygulandığında, gizli, tahmin edilebilir bilgiyi açığa çıkarmak ve büyük veritabanlarından anlamlı örüntüler, profiller ve yeni akımları keşfetmek mümkün olmaktadır. Web madenciliği bilgiyi keşfetmek için adım adım işlemesi gereken bir süreçtir ve günümüzde Web'teki tüketici ve iş aktivitelerini anlamak için olmazsa olmaz bir strateji haline gelmiştir.

Günümüzde Web üzerindeki data o kadar artmıştır ve artmaktadır ki,bu bilgilerin işlenmesi gerektiği gerçeği göz ardı edilemez hale gelmiştir. Aşağıda bu artan bilginin büyüklüğüyle ilgili çeşitli istatistiki bilgiler bulunmaktadır:

- Amerika'daki New York Times gazetesinin hafta sonu versiyonu içindeki bilgiler, 17. yüzyılda yaşayan bir kişinin tüm hayatı boyunca beyninde tuttuğu ve öğrendiği bilgi miktarından bile fazladır.
- Bir yıl içinde üretilen tüm bilim yazılarını bir kişinin okuyabilmesi için, bu kişinin 460 yılının her saniyesini bu yazılara okumaya adaması gerekmektedir.
- Her gün internete 15 milyon yeni belge ekleniyor ve internet ortamında araştırma yapmak isteyen çoğu kişinin ilk başvurduğu site olan Google, web içindeki bilgilerin yalnızca 500'de 1'ini bizlere sunabiliyor. Geri kalan kısma "Karanlık Web" deniyor. (ALTI ÜSTÜ TASARIM)

Web madenciliğin günümüzde uygulanmış örneği olarak Şekil 2.1' de görüldüğü gibi *amazon.com* internet sitesi verilebilir.

Şekil 2.1 Amazon.com sitesindeki web madenciliği uygulaması örneği

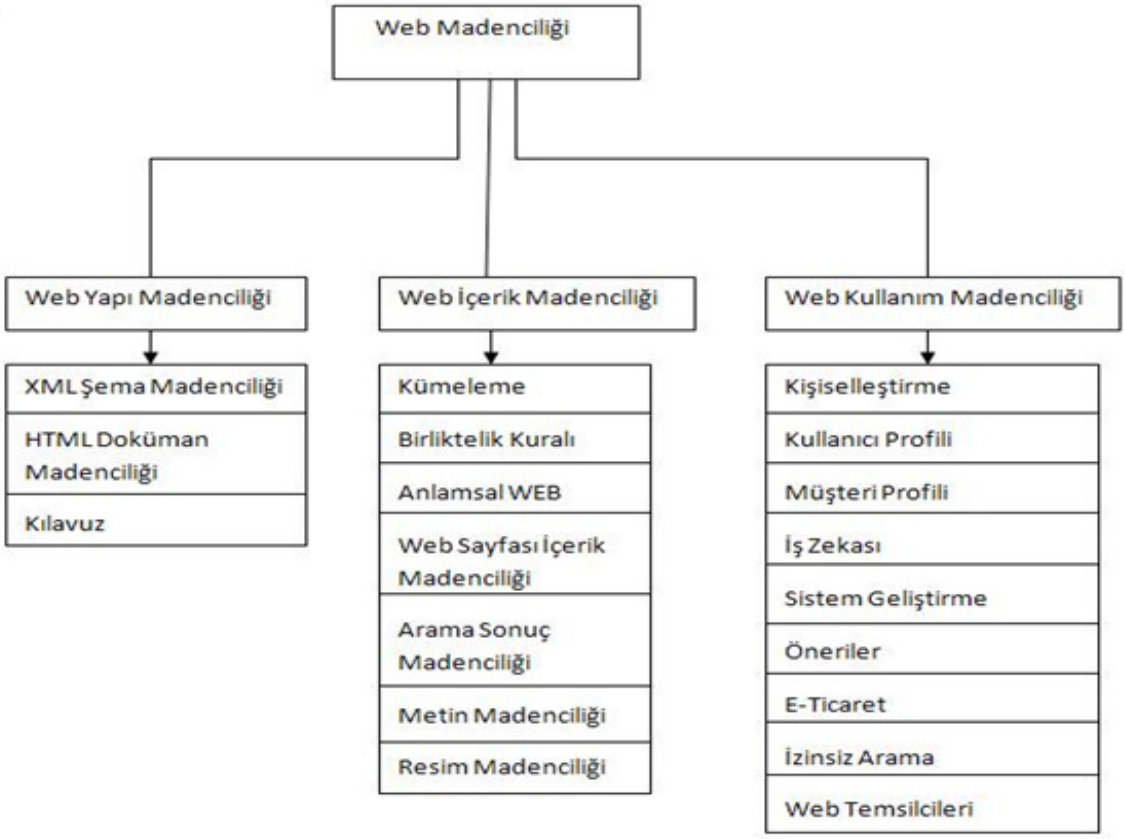
Yapılan birçok makale çalışmalarında da; web madenciliği konusunda kapsamlı ve derin araştırmalar yapılmış, önemli literatür bilgileri verilmiş, web madenciliği ve web kullanım madenciliği uygulamaları ele alınmıştır. Takci ve Soğukpınar yaptıkları makalelerde kütüphane kullanıcılarının veritabanlarını kullanarak, kullanıcıların web üzerindeki davranışları ile ilgili analiz yapmışlardır. Web kullanım madenciliği için geliştirilen yazılımlarda, internet kullanıcı davranışlarının tutulduğu tüm kayıt dosyaları (access log, agent log, error log, referrer log) için istatistiksel analizler yapılabilmektedir. Sunucularda tutulan her kayıt dosyasındaki metinsel verilerin formatı birbirinden farklıdır. Metin tabanlı verilerden sağlıklı bilgi çıkarımı yapabilmek için kayıt dosyalarındaki gürültülü ve gereksiz verilerden ayıklanması gerekmektedir; çünkü kayıt dosyalarındaki gereksiz ve gürültülü veriler hata oranını arttırmaktadır.

2.2 Web Madenciliği Türleri

Web madenciliği; Web Kullanım Madenciliği, Web İçerik Madenciliği ve Web Yapı Madenciliği olmak üzere üç kategoriye ayrılmaktadır. Ayrıca bu kategorilerin kendi aralarında temel farkları göz önüne alınarak kıyaslanması Çizelge 2.1 ve Şekil 2.2' de gösterilmiştir (R., Türkoğlu, İ. ve Poyraz, M. , 2007). Aşağıdaki bölümlerde bu üç web madenciliği türünün içeriği ve uygulama alanları ile ilgili ayrıntılı bilgi verilecektir.

Çizelge 2.1 Web madenciliğindeki verilerin incelenmesi

	Web İçerik Madenciliği	**Web Yapı Madenciliği**	**Web Kullanım Madenciliği**
Veri	Metin belgeleri ve HTML	HTML linkleri	Sunucu ve tarayıcı kayıtları (logs), çerezler, kullanıcı profilleri, kullanıcı sorguları, meta data
Verinin Şekli	Yapısız ve karışık	Link yapısı	Kullanıcı etkileşimi ve davranışı
Gösterilimi	İlişkisel ve sınıflandırmalı	Grafiksel	İlişkisel tablolar ve grafiksel



Şekil 2.2 Web madenciliğinin sınıflandırılması

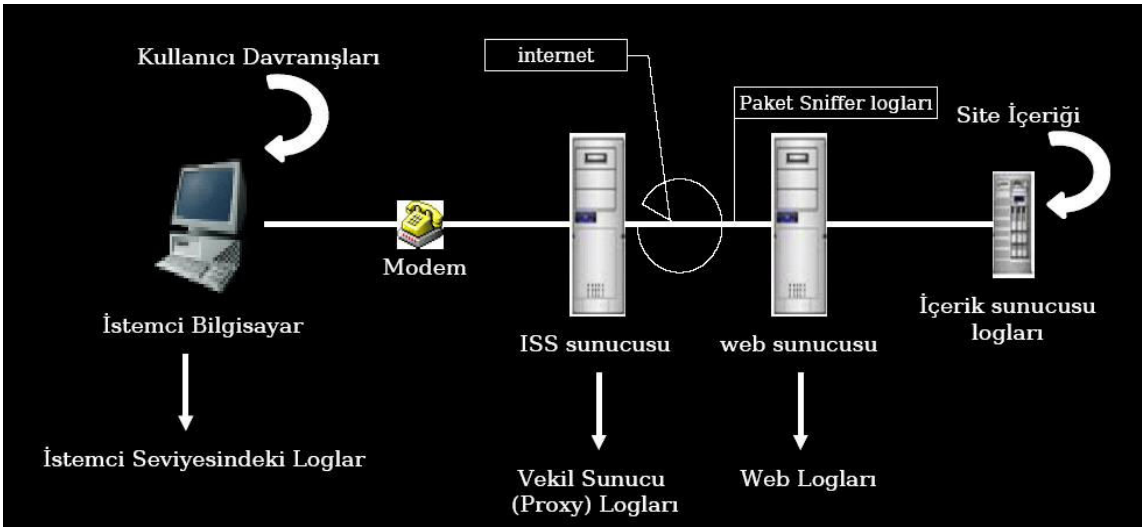
2.2.1 Web Kullanım Madenciliği (Web Usage Mining)

Bu tez çalışmasının kapsamının asıl konusunu oluşturan Web Kullanım Madenciliği, uyarlanabilir Web site tasarımı, Web sunucu tasarımı, kullanıcı sınıflandırması ve Web site kişiselleştirmesi gibi alanlarda kullanılmak üzere, büyük Web veri depolarındaki kullanım logları üzerinde veri madenciliği tekniklerinin kullanılması işlemidir. Web kullanım madenciliğinin işleyeceği veriler erişim kayıtları, kullanıcı tarafında bulunan çerezler, kullanıcı profilleri, metadata gibi veriler olup, bu veriler istemcilerde, sunucularda ve Proxy sunucularında depolanmaktadır (Cooley, R., Srivastava, J., Deshpande, M. ve Tan,P., 2000).

Web kullanım madenciliği işlemi genel olarak 2 yaklaşımda incelenebilir (Borges, J. ve Levene, M., 1999). Birinci yaklaşımda, uyarlanabilir veri madenciliği tekniği uygulanmadan

önce Web sunucu kullanım verilerini ilişkisel tablolarla eşleştirir. Diğer yaklaşım ise, log verilerini özel ön işlem tekniklerinden yararlanarak log verilerini direk olarak kullanır.

Genel olarak web kullanım madenciliğinin işlediği veriler şekil 2.3' teki gibi gösterilebilir:



Şekil 2.3 Web kullanım madenciliğinin işlediği veriler

Web Kullanım Madenciliği kullanılarak, bir web sitesi erişim verilerinden şu sorulara cevap alınabilir:

- Siteyi bugün kaç kişi ziyaret etti?
- Siteye kimler link vermiş?
- En çok hangi sayfadan sonra site terkedilmiş ?
- En çok ziyaret edilen sayfa hangisi?
- Siteyi internette bulabilmek için hangi anahtar kelimeler kullanılmış?
- Kullanım sürelerinin günlere ve saatlere göre dağılımı nasıldır?
- Sayfalara göre istemlerin dağılımı nasıldır?
- Ulaşılmayan sayfalar hangileridir?
- Ulaşılamayan linkler hangileridir?
- İstemlerin statülerine göre dağılımı nasıldır?
- Siteye saldırı var mı yok mu?
- Kullanıcıların profilleri nasıldır?
- Kullanıcıların zaman içindeki değişimleri nasıldır?
- Sitede beğenilen sayfalar, beğenilmeyen sayfalar hangileridir?
- Sitenin içeriği nasıl olmalıdır?

2.2.1.1 Sunucu Taraflı Veriler

Sunucu taraflı veriler, web kullanım madenciliği için en önemli kaynaklardan birisidir; çünkü site ziyaretçilerinin davranışlarını net bir şekilde kayıt ederler. Sunucu loglarında tutulan veriler, siteyi ziyaret eden her kullanıcının her işlemi hakkında bilgi verir. Web sunucuları çerezler(cookie) ve talep edilen veri sorguları gibi çeşitli bilgileri de tutar.

2.2.1.2 İstemci Taraflı Veriler

İstemci taraflı veriler, veri yığını yeteneklerini çoğaltmak için, uzaktan bir ajan kullanarak (Javascript ve Java applet leri) ya da kullanılan tarayıcının kaynak kodunu değiştirerek hazırlanabilir. İstemci taraflı veriler ön belleğe alma ve oturum tanımlama problemlerini iyileştirme özelliği sayesinde sunucu taraflı verilere göre avantajları vardır. Bununla birlikte, site kullanım istatistiklerini toplamaya imkan vermesi açısından sunucu logları kadar faydalı bilgi içermez.

2.2.1.3 Proxy Sunucu Taraflı Veriler

Bir Web proxy sunucusu, Web sunucusu ve istemci tarayıcıları arasında veriyi saklamak için ara bir seviye olarak işlev görür. Proxy sunucularının sakladığı verilerin performansı gelecekteki sayfa isteklerini tahmin etme kabiliyetlerine bağlıdır. Proxy izleri çeşitli istemciden çeşitli sunuculara yapılan gerçek HTTP isteklerini ortaya çıkarabilir. Bu tür veriler, ortak bir proxy sunucusunu paylaşan anonim kullanıcıların tarayıcı davranışlarını karakterize etmek için işlenebilecek bir veri kaynağı olarak kullanılabilir.

2.2.2 Web İçerik Madenciliği (Web Content Mining)

Ses, görüntü, video, bağlantılı ve bağlantısız metinler içeren ve çoğu belli bir düzene sahip olmayan çoklu web sayfalarından otomatik bilgi çıkarımı, web içerik madenciliğinin ilgi alanına girmektedir. Bu verilerden anlamlı bilgi çıkarmak için akıllı programlar kullanılmaktadır. Bu programların amacı, web sayfalarında dolaşarak bilgiler toplamaktır. Arama motorları, web içerik madenciliği tekniklerinden faydalanmaktadır.

Web içerik madenciliği, metin madenciliği ve veri madenciliği ile ilgili olmasına rağmen aralarında bir takım farklılıklar vardır. Web içerik madenciliği, veri madenciliği ile ilgilidir; çünkü web dokümanları içerisindeki verileri çıkarmak için veri madenciliği tekniklerini kullanır. Veri madenciliğinde, tam olarak yapısal veriler kullanılırken; web verileri kısmi yapılı ve yapısız verilerdir.

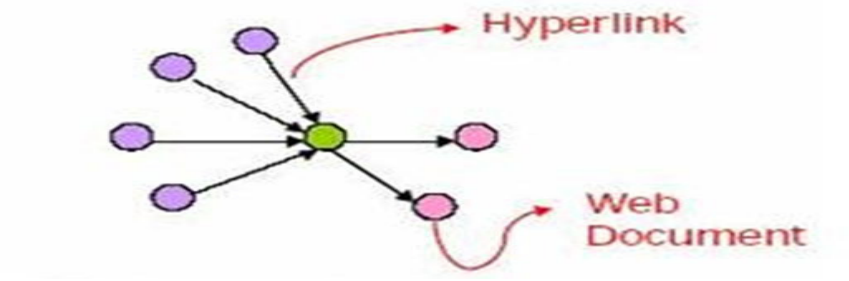
Aynı şekilde, web içerik madenciliği metin madenciliğiyle ilgilidir; çünkü web üzerindeki bilgilerin çoğu metin tabanlıdır. Web içerik madenciliği ile metin madenciliği arasındaki fark ise metin madenciliğinin tamamen yapısal olmayan veriler üzerinde odaklanmış olmasıdır.

Web içerik madenciliğinde kullanılan iki yaklaşım vardır:

- Information Retrieval Approach (IR): Kullanıcı profili baz alınarak kullanıcılara gösterilen bilgileri filtrelemek ve bilgiye erişimi geliştirmek için kullanılan yöntemdir.
- Database Approach: Web' teki veriyi modellemek ve veriyi bütünleştirerek daha karmaşık bir yapıya sokmak için kullanılan yöntemdir. Bu yöntem sayesinde anahtar kelime tabanlı arama yerine daha gelişmiş sorgular çalıştırmak mümkün olur.

2.2.3 Web Yapı Madenciliği (Web Structure Mining)

Web sayfaları arasındaki bağlantıları ya da bir web sayfasındaki bağlantılar arasındaki ilişkileri inceleyerek, sonucunda bilgi üretir. Web yapı madenciliği, linklerin topolojisinden faydalanarak farklı siteler arasındaki benzerlik ve ilişki gibi bilgileri üretir, sayfaların link tasarımlarını ortaya çıkarmamıza yardımcı olur. İlgili araştırmalar "hyperlink" düzeyinde yapılıyorsa "Hyperlink Analysis" adını alır. Şekil 2.4' te web grafik yapısı görülmektedir. Web dokümanları arasındaki oklar iki sayfa arasındaki ilişkiyi temsil etmektedir.



Şekil 2.4 Web sayfaları arasındaki link bağlantısı

Web dokümanları arasındaki linkler bir araya getirildiğinde "Web Graph Structure" elde edilir. Bu yapı sayesinde iki nokta arasındaki en kısa yola ulaşılabilir. Bu bilgi web sayfaları arasındaki ilişkiyi belirlemek açısından son derece önemlidir. İki sayfa arasında doğrudan bir link yoksa, o link arasındaki bağlantıya ve komşuluk ilişkisine kolay bir şekilde erişilebilir. Web yapı madenciliği sayesinde, araştırılan konu ile ilgili bir sayfayı sisteme vererek onunla ilgili tüm sayfalara erişebilir, web sayfaları arasındaki benzerlik ilişkilerini çıkarabiliriz.

2.3 Web Madenciliğinde Kullanılan Araçlar

Web madenciliğine yardımcı olmak için birçok ücretli ya da ücretsiz program bulunmaktadır. Bu programlara örnek olarak şunlar verilebilir:

❖ SpeedTracer

IBM' in geliştirdiği ücretli bir web madenciliği aracıdır. SpeedTracer, kullanıcının gezindiği sayfalar hakkında şablonlar çıkaran, raporlayan ve bu sayede site yöneticilerine web sayfasının yapısını düzenleme imkanı veren web kullanım madenciliği ve analizi aracıdır. Uygulama, yenilikçi çıkarım algoritmaları kullanarak kullanıcının izlediği yolları yeniden yaratır ve kullanıcı oturumlarını tanımlar. Gelişmiş madencilik araçları kullanıcının web sitesindeki hareketlerini açığa çıkarır. Program çıktısı olarak web sitesi yöneticilerine kullanıcı davranışlarını daha iyi anlamalarına yardımcı olacak, sayfalardaki gezinti bilgilerinden oluşan bir koleksiyon sunulmaktadır.

SpeedTracer üç çeşit istatistik oluşturmaktadır: Kullanıcı temelli (user-based), gezilen yol temelli (path-based), grup temelli (group-based).

- Kullanıcı temelli istatistikler, kullanıcının giriş zamanlarını ve sistemde kaldığı süreyi tam olarak göstermektedir.
- Gezilen yol temelli istatistikler, sıklıkla gezilen yolları tanımlar.
- Grup temelli istatistikler, sıklıkla ziyaret edilen web sayfası grupları hakkında bilgi verir.

❖ Clementine

SPSS firması tarafından ücretli olarak geliştirilmiştir. SPSS CLEMENTINE, veri madenciliği uygulamaları için geliştirilmiş bütünsel bir görsel modelleme gerecidir. Veriye kolayca erişme, veriyi modellemeye hazırlama, modelleri hızlı bir şekilde oluşturma, birden fazla modeli ardışık olarak uygulama ve farklı model sonuçlarını kolayca karşılaştırmalarını sağlar. Clementine, iş kullanıcılarına elde ettikleri sonuçları diğer iş prosesleri ile kolayca entegre edebilme olanağı verir, bu sayede kuruluşlar dinamik bir karar destek sistemi oluşturabilirler.

❖ **Net Analysis**

Net Genesis firmasının ücretli, ödüllü çevrimiçi davranış analizi çözümü olan Net Analysis, çevrimiçi rekabetçi ortamlarda e-iş yatırımları için gerekli yüksek düzeyde esneklik ve genişletilebilirlik sağlamaktadır. Yüksek düzeydeki esnekliği ve fonksiyonelliği ile Net Analysis, herbir şirketin belirli e-müşteri ihtiyaçlarını karşılarken, şirketin mevcut mimarisine de kolayca uyum gösterir.

❖ **Weblog_parse**

ACME Labs yazılımı olan weblog_parse, log dosyalarını işleyen ücretsiz bir araçtır. Weblog_parse, web log dosyasından belirli alanları çıkarır. Web sunucusunun log dosyasını okur ve sadece kullanıcı tarafından istenen alanları listeler.

❖ **Weblog**

Darryl C. Burgdorf tarafından geliştirilen Weblog, giriş loglarını ayrıntılı şekilde analiz eden ücretsiz bir araçtır. Ay, hafta, gün ve saat bazında site üzerindeki aktivitelerin (toplam sayfa ziyareti, ne kadar veri transferi olduğu, popüler sayfalar, vs..) izlenmesine imkan tanır.

❖ **Analog**

Cambridge Üniversitesi İstatistiksel Laboratuvarı tarafından geliştirilen Analog, web sunucusu log dosyalarını analiz eden ücretsiz bir programdır. Program sayesinde bir web sitesindeki kullanıcı aktiviteleri hakkında bilgi sahibi olunabilir.

3. LOG DOSYALARI

3.1 Log Dosyası Nedir?

Bir internet sitesinin üzerinde hizmet verdiği web sunucusu, bu internet sitesindeki hareketleri kayıt etmek için tipik olarak ASCII metin formatında log dosyaları üretir. Tarayıcıdan bir internet sayfasına erişilmeye çalışıldığında, web sunucusu istenilen sayfayı kendi veritabanından ya da dosya sisteminden bulup getirerek, istek gönderilen sayfayı görmeyi sağlar. Web sunucusu görmek istenilen bu sayfaki bilgiler ile resim, video gibi yapıları getirdikten sonra, bu görevi bir log dosyasına kaydeder. Daha sonra bu log dosyaları kullanılarak, sitedeki kullanıcı trafiği, en çok tıklanan sayfalar gibi çeşitli alanlarda analiz yapılır. Herbir log dosyası formatı genelde aynı tip bilgiler barındırsa da, belli başlı log dosya formatı bulunmaktadır. Log dosyasının hangi formatta olduğu, dosyanın baş tarafında gösterilerek belirlenir. Proxy sunuculardа ise, log dosyaları düz metin formatında bulunabilir.

3.2 Log Dosyası Türleri

Web kullanım madenciliği uygulamalarının asıl kaynağı web sunucularında oluşturulan web log dosyalarıdır. Dört çeşit log dosyası türü bulunmaktadır.

- **Erişim Kayıt Dosyaları**

Bir internet sayfasını görüntülemek için web sunucusuna istek gönderen bir kullanıcı web tarayıcısının bu istek kaydı, erişim kayıt dosyasına bir kayıt olarak kaydedilir. Erişim kayıt dosyasının formatı, bulunduğu işletim sistemine bağlı olarak farklılık gösterebilir.

- **Hata Kayıt Dosyaları**

Web sunucunun üzerinde hata veren, gerçekleştirilemeyen işlemler için kaydedilen kayıt dosyalarıdır.

- **İstek Kayıt Dosyaları**

İnternet kullanıcısının sayfa isteklerinin tutulduğu kayıt dosyalarıdır.

- **Etmen Kayıt Dosyaları**

İnternet kullanıcısının kullandığı istemci bilgisayarın, işletim sistemi, web tarayıcısının adı, sürümü bilgilerinin tutulduğu kayıt dosyalarıdır.

İstek ve etmen kayıt dosyasının web sunucusu üzerinde tutulup tutulmayacağı, kullanılan log dosya formatına bağlıdır.

3.3 Log Dosyası Formatları

Interneti kapsayan çoğu durumda, internet sitesinin yayın yapmasına olanak sağlayan bir web sunucusu bulunur. Bu web sunucusu tipik olarak ASCII karakterler içeren metin tabanlı dosyalar üretirler. Bu dosyalar sayesinde de, internet sitesini ziyaret eden kullanıcıların aktiviteleri kayıt altına alınıp, daha sonra çeşitli analizlerde kullanılarak sitenin istatistiki bilgileri çıkarılıp, daha verimli duruma nasıl getirilebileceği saptanabilir. Bu kayıt edilen dosyalara "*log dosyaları*" denmektedir.

Log dosyalarında en çok kullanılan formatlar NCSA ve W3SVC olmakla birlikte çeşitli formatlar bulunmaktadır. En çok kullanılan formatlar şunlardır: NCSA, W3C Extended, Sun ONE Web Server, IBM Tivoli Access Manager WebSEAL, WebSphere Application Server Logs (IBM).

3.3.1 NCSA Log Dosya Formatı

NCSA log dosya formatları, HTTP sunucularında genel format olarak kabul edilmiştir. 3 türü bulunmaktadır.

3.3.1.1 NCSA Common Log Dosya Formatı (Erişim Dosyası)

NCSA Common log dosya formatı,temel HTTP erişim bilgilerini içerir. İstekte bulunan istemcinin bilgileri ile birkaç bilgi daha içerir; ancak çerez (cookie),kullanıcı ajanı (user agent) gibi bilgiler içermez. Bilgiler tek bir dosyada tutulur. NCSA Common log dosyasındaki alanlar sırasıyla şunlardır:

host rfc931 username date:time request statuscode bytes

Örnek :

125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043

- **Host**: HTTP isteğini yapan istemcinin IP numarası ya da host/subdomain adresi bilgisidir.
- **Rfc931**: HTTP isteğini yapan istemciyi ayırt etmek için kullanılan belirleyicidir. Eğer hiçbir değeri yoksa, ”-” karakteri koyulur.
- **Username**: Doğrulamak amacı ile kullanılan kullanıcı adı ya da kullanıcı ID’si bilgisidir. Eğer hiçbir değeri yoksa, ”-” karakteri koyulur.
- **Date:Time Timezone**: HTTP isteğinin yapıldığı gün ve saati gösteren bilgidir. Bu bilgideki alanlar ve açıklamaları ise şöyledir:

dd/MMM/YYYY: hh:mm:ss +-hhmm
dd: Gün
MMM: Ay
YYYY: Yıl
hh: Saat
mm: Dakika
ss: Saniye
hhmm: Zaman bölgesi

- **Request**: HTTP isteği bilgisini tutar. Bu bilgi 3 alandan oluşur: İstek yapılan sayfa(index.html), istek yapılan sayfaya gönderilen HTTP metodu (GET, POST, ...) ve HTTP protokol versiyonu (1.0, 2.0, ...)
- **Statuscode**: HTTP isteğinin başarılı bir şekilde yapılıp yapılmadığını belirten sayısal bir koddur. statuscode alanının alabileceği ve en çok karşılaşılan değerler şunlardır:

200 - OK : Aranan belge bulundu ve gönderildi.

301 - Moved Permamently : Aranan belge temelli başka bir adrese aktarıldı.

302 - Temporarily Redirected : Aranan belge geçici olarak başka bir adrese aktarıldı.

400 - Cannot Be Found : Erişmek istediğiniz dosya bulunamıyor.

404 - Site Not Found : Aradığınız site (URL) bulunamıyor.

Statuscode alanının alabileceği tüm değerler Ek 1’de mevcuttur.

- **Bytes**: HTTP isteği sonucu gönderilen veri boyutunu tutan sayısal bir alandır.

3.3.1.2 NCSA Combined Log Dosya Formatı

NCSA Combined log dosya formatı, NCSA Common log dosya formatının bir uzantısıdır. Common log dosyası formatındaki alanlara ek olarak 3 tane seçimli alanı mevcuttur. NCSA Combined log dosyasındaki alanlar sırasıyla şunlardır:

host rfc931 username date:time request statuscode bytes referrer user_agent cookie

Örnek:

125.125.125.125 - dsmith [10/Oct/1999:21:15:05 +0500] "GET /index.html HTTP/1.0" 200 1043 "http://www.ibm.com/" "Mozilla/4.05 [en] (WinNT; I)" "USERID=CustomerA;IMPID=01234"

- **Referrer**: HTTP isteği yapan kullanıcıyı internet sitesine bağlayan bağlantı adresi bilgisidir.
- **User_agent**: HTTP isteği yapan kullanıcının kullandığı web tarayıcı ve platform bilgileridir.
- **Çerezler (Cookies):** Çerezler (cookie) , web sunucusu tarafından istemci bilgisayara gönderilebilen bilgi parçacıklarıdır. İstemci bilgisayarı bu bilgileri saklayabilir ve sonradan yaptığı ek isteklerde bu bilgileri HTTP sunucusuna geri gönderebilir. HTTP sunucusu bir HTTP isteğine karşılık birden fazla çerez oluşturabilir. Eğer sunucu çerezleri oluşturma seçeneğiyle kurulduysa, çerezler dosya olarak kaydedilir.

3.3.1.3 NCSA Seperate (Three-Log Format)

NCSA Seperate log dosya formatında, toplanan bilgi tek bir dosya yerine, 3 parçaya bölünerek saklanır. Bu 3 log dosyası erişim dosyası, gönderici logu ve agent logu olarak işlev görür. NCSA Seperate log dosya formatı temel bilgileri NCSA Common log dosya formatında bir dosyada saklar. Gönderici ve ajan loglarını da sonraki dosyalarda saklar. Bu formatta çerez bilgisi kayıt edilmez.

- Common veya Erişim Logu (Access Log):NCSA Common log dosyası formatında tutulur.
- Gönderici Logu (Referral Log) :Common log'taki herbir kayda karşılık gelen bir kayıt tutulur. Bu dosyadaki alanlar sırasıyla şunlardır:

date:time referrer

Örnek:

[10/Oct/1999:21:15:05 +0500] http://www.ibm.com/index.html

Date:Time Timezone:HTTP isteğinin yapıldığı gün ve saat bilgisidir.
Referrer: Kullanıcının ulaşmak istediği HTTP kaynağının URL adresi bilgisidir.

- Agent Log: Gönderici logu gibi, Commonlog'taki herbir kayda karşılık burada bir kayıt tutulur. Bu dosyadaki alanlar şunlardır:

Date:Time Timezone: HTTP isteğinin yapıldığı gün ve saat bilgisidir.

Ajan(Agent): HTTP isteklerini yapan HTTP istemcisi bilgisidir. HTTP istemcisi için Web tarayıcısına HTTP isteği yaparken kendisini isimle tanıtmak alışılmış bir olaydır. Zorunlu olmamakla birlikte, çoğu HTTP istemcisi kendisini isim yoluyla tanıtır. Web sunucusu da bu isim bilgisini Agent log dosyasına yazar.

3.3.2 W3C Extended Log Dosya Formatı

Bu log dosya formatı Microsoft Internet Information Server(IIS) tarafından kullanılmaktadır. Bu log dosyası ASCII karakterlerini içeren satır kümelerinden oluşan kayıtlardan meydana gelir. Herbir kayıt,tek bir HTTP işlemiyle ilişkili sıralanmış alanlardan oluşur. Alanlar boşluk ile birbirinden ayrılır. Eğer bir alanda kayıt yoksa, "-" karakteri ile belirtilir. Yönergeler ise "#" karakteri ile başlar ve loglama işleminin kendisi hakkında bilgi tutar.

Bu tez çalışmasının analiz edeceği log dosyaları da W3C Extended Log formatında olup, yönergelerde bulunan alanlar sırasıyla şunlardır (W3C) :

#Software:

#Version:

#Date:

#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status

Örnek:

#Software: Microsoft Internet Information Services 6.0

#Version: 1.0

#Date: 2008-06-01 00:01:01

#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status

2008-06-01 00:01:00 W3SVC1497983639 62.244.196.150 GET /en/Sirket_Listele.asp iPage=0&MemberID=&Language=&Tip=0&Sayfa=6&category=13 80 - 74.6.22.153 Mozilla/5.0+(compatible;+Yahoo!+Slurp;+http://help.yahoo.com/help/us/ysearch/slurp) 200 0 0

- **Version**: Extended log dosya formatının kullandığı versiyon numarasıdır.
- **Fields**: Herbir kayıtta tutulan alanların listesidir.
- **Software**: Log u üreten yazılım bilgisidir.
- **Start-Date**: Log işlemine başlandığı gün ve saat bilgisidir.
- **End-Date**: Log işleminin bittiği gün ve saat bilgisidir.
- **Date**: Kaydın dosyaya eklendiği gün ve saat bilgisidir.
- **Remark**: Yorum bilgisidir.

Yönerge alan ve versiyon bilgisi gereklidir ve kayıt bilgilerinden önce yer almalıdır. Yönergede “#Fields” ile başlayan satırda yer alan S,C ve R harflerinin anlamları ise Çizelge 3.1’ de görülebilir.

Çizelge 3.1 Yönergede yer alan S, R, C harflerinin anlamı

Harf	Anlamı
C	Client (İstemci)
S	Server (Sunucu)
R	Remote (Uzaktan Erişim)
CS	Client to Server (İstemciden Sunucuya)
SC	Server to Client (Sunucudan İstemciye)
SR	Server to Remote Server (Sunucudan Uzaktaki Sunucuya)
RS	Remote to Server (Uzaktaki Sunucudan Sunucuya)
X	Uygulamaya Özgü Belirleyici

W3C Extended log dosya formatında kaydedilen kayıtlardaki alanlar ise şunlardır:

❖ **Ön ek gerekmeyenler**:

- **Date** : İşlemin tamamlandığı tarih bilgisidir.
- **Time** : İşlemin tamamlandığı zaman bilgisidir.
- **Time-taken** : İşlemin tamamlanması için geçen zamanın saniye birimindeki değeridir.
- **Bytes** : Transfer edilen bilginin byte birimindeki değeridir.
- **Cached** : Eğer ön bellekte saklama işlemi var ise 1, yok ise 0 değerini tutar.

❖ **Ön ek gerekenler**:

- **Ip** : İstemcinin IP adresi ve port numarası bilgisidir.
- **Dns** : DNS ismi bilgisidir.
- **Status** : Durum kodu bilgisidir.
- **Comment** : Durum koduyla dönen yorum bilgisidir.
- **Method** : Gönderilen isteğin gönderildiği metot türü bilgisidir.
- **Uri** : URI bilgisidir.
- **Uri-Stem** : URI'nin sadece gövde bölümünün bilgisidir.
- **Uri-Query** : URI'nin sorgu bölümünün bilgisidir.

3.3.3 Sun ONE Web Server (iPlanet) Log Dosya Formatı

One Web Server formatındaki bir log dosyası, ASCII karakterler içeren satırlar dizisinden oluşmaktadır. Herbir satır bir bildirim ya da bir girişten oluşmaktadır. Girişler tek bir HTTP işlemiyle ilişkili alanlar kümesinden oluşur. Alanlar boşluklarla ayrılır ve eğer bir alan kayıt edilmeyecekse, bu alan için '-' işareti koyulur. Log dosyasının başında bulunan bildirimler, girişlerde bulunan bilgi ve format hakkında bilgi verir.

Format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] "%reqpb.clf.-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%

Örnek:

10.101.128.66 - - [03/Apr/2002:00:42:11 -0500] "GET /multimedia/flash/dis.swf HTTP/1.1" 304

3.3.4 IBM Tivoli Access Manager WebSEAL Log Dosya Formatı

IBM Tivoli Access Manager WebSEAL, Web tabanlı kaynaklar için bir güvenlik yönetimi kaynağıdır. Bu formattaki dosyaları alabilmek için, WebSEAL NCSA Combined Log Formatında log dosyaları üretecek şekilde ayarlanmalıdır.

3.3.5 WebSphere Application Server (WAS) Log Dosya Formatı

WebSphere Application Server, uygulama seviyesinde loglama sağlayan Analytic Logging Service (ALS) isimli bir API sunmaktadır. Bu API Web uygulamalarından çağırılarak, loglama sağlanabilmektedir. ALS API'sinin desteklediği üç tane loglama metodu bulunmaktadır: HTTP, veritabanı ve dosya. Loglanan veri için format hangi metodun kullanıldığına göre değişir; ancak hepsinde ortak olan beş alan bulunmaktadır:

appid reqid httpdata cookie appdata

Örnek:

<custCareID> <01010123-1234> <-> <->

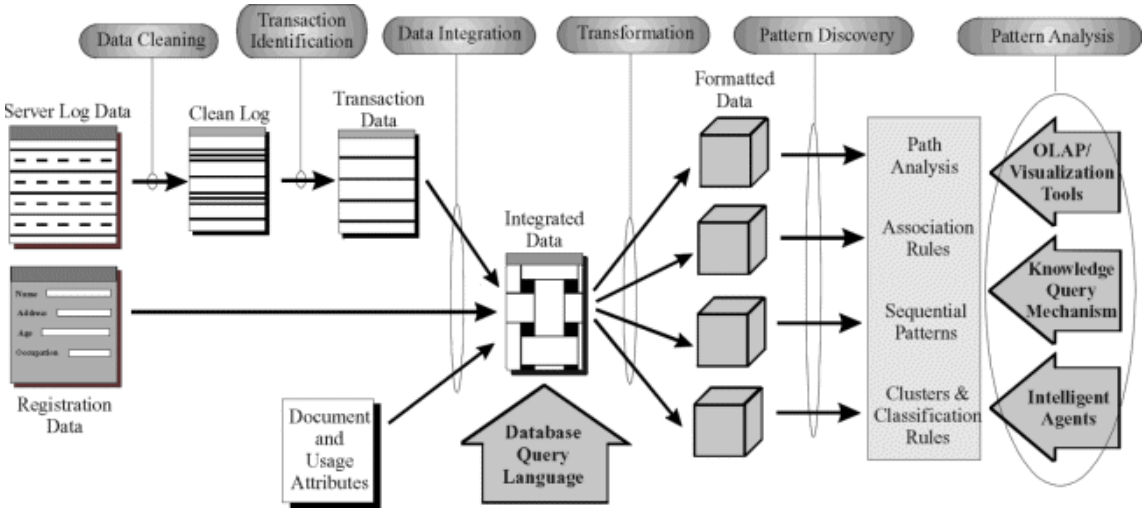
<target=TP600E;crosssells=TP770,TP380;ads=pIII>

- **Appid** : İlgili kaydın uygulama ID'sidir. Appid birçok uygulamanın aynı ortamda loglama yapmasına imkan vermektedir.
- **Reqid** : HTTP isteğinin ilinti ID'sidir.
- **Httpdata** : NCSA Combined Log foarmtındaki alanları tutar.
- **Cookie** : Çerez (cookie) bilgisidir.
- **Appdata** : Uygulama verisidir. Uygulama verisi herhangir bir uygulama seviyesindeki veriyi içerebilir. Bu alandaki değerler, ';' ile ayrılan ve '=' işaretinin iki tarafında bulunan verilerden oluşur.

Tüm log dosyası formatları ve örnekleri Ek 2'de bulunmaktadır.

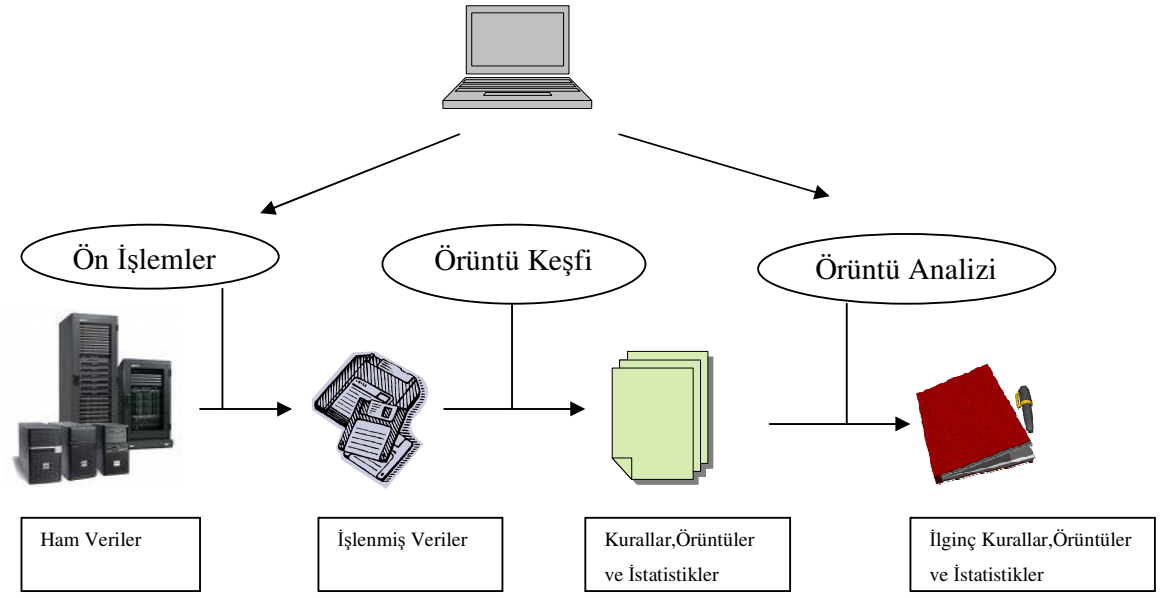
4. WEB KULLANIM MADENCİLİĞİ İŞLEM BASAMAKLARI

Web kullanım madenciliği tekniği, işleyeceği ham veriyi öncelikle belirli işlem basamaklarından geçirmelidir. İşlenmeye hazır hale gelen veri, kullanılacak veri madenciliği algoritması uygulanarak, analiz edilebilecek bilgilere dönüşür. Daha sonra bu bilgilerden de anlamlı sonuçlar çıkarılarak, web kullanım madenciliğinin amacına ulaşılmış olunur.



Şekil 4.1 Web kullanım madenciliği genel mimarisi

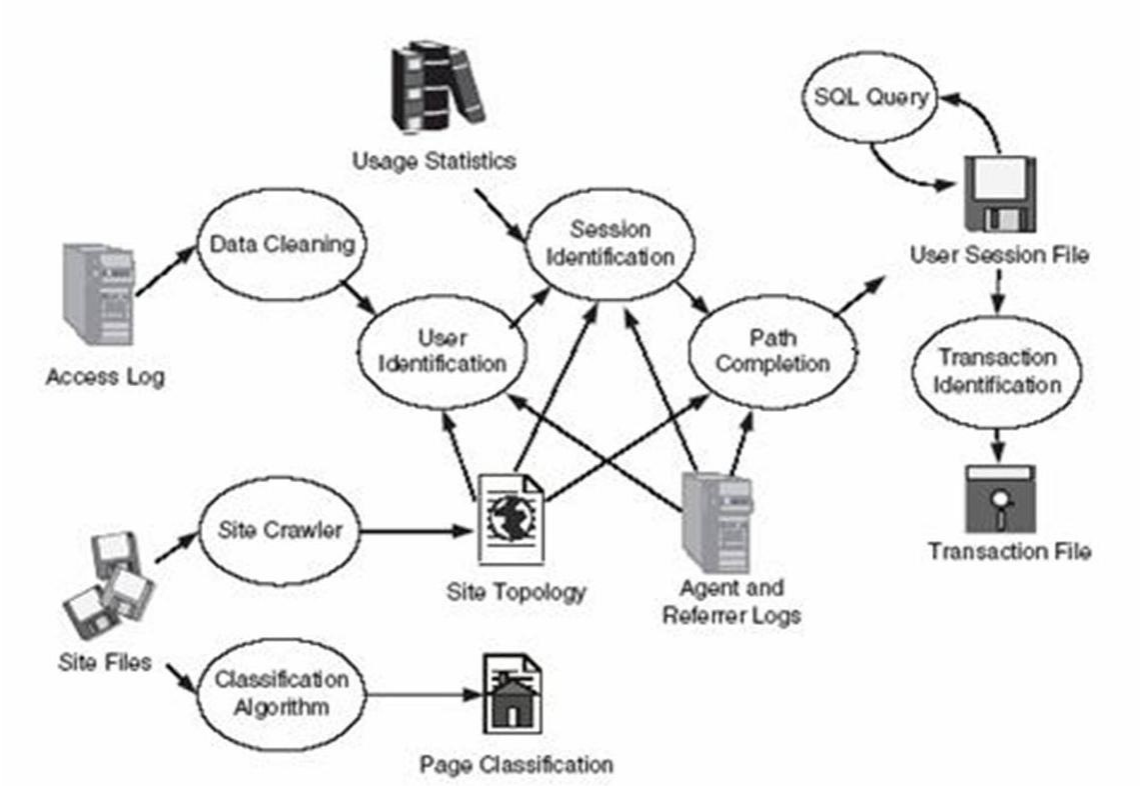
Web kullanım madenciliğinin işlem basamakları Şekil 4.2 'de de görüldüğü gibi, Ön işlem, Örüntü Keşfi ve Örüntü Analizi olarak söylenebilir (Cooley, R., Srivastava, J., Deshpande, M. ve Tan,P., 2000).



Şekil 4.2 Web kullanım madenciliği işlem basamakları

4.1 Ön İşlemler

Ön işlemler basamağının verileri sunucu logları, web sitesi dosyaları ve isteğe bağlı olmak suretiyle bir önceki analizden elde edilen kullanıcı istatistikleridir. Ön işlemler sonucunda elde edilen veriler ise, kullanıcı oturum dosyası, işlem dosyası, site topolojisi ve sayfa sınıflandırmalarıdır. Güvenilir, sağlam bir kullanıcı oturum dosyası yaratmak için en büyük engellerden birisi, tarayıcı ve proxy sunucularının gezinilen web sitesi verilerini ön belleğine almasıdır. Ön belleği işlemi, tarayıcıda tekrar o sayfaya tıklandığında, sayfayı yeniden yüklemek yerine, ön bellekten getirmesine sebep olmakta, bu da web kullanım madenciliğinde güvenilir veriye erişimi zorlaştırmaktadır. Bunu engellemek için kullanılan bazı metodlar olmakla birlikte, bu metodlar yine de tam olarak düzgün veriyi bize sağlayamamaktadır. Bunun gibi diğer düzgün olmayan ham veriyi işlemeye hazır hale getirmek için çeşitli ön işlemlerden geçirmek gerekmektedir.



Şekil 4.3 Web kullanım madenciliğindeki ön işlem basamakları

4.1.1 Veri Temizleme

Web logları üzerindeki ilgisiz verileri temizlemek için kullanılan tekniklerin başarısı, sadece kullanım madenciliği için değil, tüm log analiz tipleri için çok önemlidir. Eğer web loglarındaki bilgiler web sitesindeki kullanıcı erişimlerini tam, yanlışsız bir şekilde gösteriyorsa, oluşturulan kullanıcı istatistikleri ve diğer örüntü keşifleri faydalı bilgi sağlar. Veri temizleme basamağı 3 aşamada yapılacak işlemlerle tamamlanabilir. Bu aşamalar şöyledir:

HTTP protokolü web sunucusundan istekte bulunulan herbir dosya için ayrı bir bağlantıya ihtiyaç duyar. Bu yüzden de, kullanıcının bir web sayfasındaki kısmi bir bölümü görüntülemek istemesi bile, log dosyasına birçok satırın eklenmesine sebep olur. İstekte bulunan herbir sayfa için, o web sayfasındaki şekiller, scriptler HTML dosyasına ek olarak yüklenir. Web kullanım madenciliğinde kullanıcının sayfadaki gezinme verilerini elde edebilmek için, ilgisiz olan verilerden kurtulmak gerekir. Web loglarında kayıt altına alınmış veride temizleme yapılarak, bu ilgisiz verilerden kurtulmak mümkündür. Web log dosyasında, istekte bulunan sayfalardan uzantısı resim dosyaları(.jpeg,.gif,.jpg,.png,.ico,.asf,...), script,web

sitesi ve işletim sistemi dosyaları (.js,.vbs,.exe,.dll,.src,count.cgi,.asmx,.axd,.ashx,.xml,...), HTML stil dosyaları(.css,...), metin dosyaları(.txt,...), flash dosyaları(.swf,...) gibi gerekli olmayan yapılara ait olan uzantıları temizlemek gerekmektedir.

İstekte bulunulan herbir web sayfasına erişim sağlanamayabilinir. Ek 1'deki listede görüldüğü gibi, web sayfasına erişim sorununun çeşitli sebepleri olabilir. Bir sayfada teknik bir sorun olsa ya da kullanıcı herhangi bir sebepten dolayı web sayfasına erişemiyor olsa bile, istekte bulunduğu her bir sayfa için log dosyasına kayıt atılmaktadır. Log dosyasından StatusCode alanına bakılarak erişilemeyen sayfalar temizlenmeli, sadece StatusCode değeri 200 (Aranan belge bulundu ve gönderildi.) olan kayıtlar kalmalıdır.

Bir diğer aşama da, web loglarından web sitesine ulaşan faydasız, küçük programlardan, web sitesinde dolaşarak bilgi toplamayı amaçlayan yapılardan (agent, spider, crawler) kurtulmaktır. Bu faydasız yapıların sayısı gün geçtikçe artmakta, web siteleri bu yapılardan kurtulamayabilmektedir ve bunların siteye erişimleri de log dosyalarına kayıt olarak eklenmektedir. Bazı internet kaynaklarında güncel olarak bu yapıların listesi bulunabilmektedir. Web logları bu kaynaklardaki listeler doğrultusunda, zararlı yapılardan temizlenebilmektedir.

Hem dosya uzantılarına göre, hem StatusCode değerine bakılarak, hem de zararlı yapıların web loglarında bıraktığı kayıtları temizleme işlemi yapıldıktan sonra, veri temizleme aşaması tamamlanmış olup, sıradaki aşama olan kullanıcı tanımlama aşamasına geçilebilir.

4.1.2 Kullanıcı Tanımlama

Kullanıcı tanımlama, bir web sayfasına kimin eriştiği ve hangi sayfalara eriştiği bilgisini tanımlama aşamasıdır. Eğer kullanıcıların siteye giriş bilgileri (kullanıcı adı ve şifre) varsa, kullanıcıları tanımlamak kolaydır; ancak çoğu kullanıcı bu bilgileri girmekten kaçınmaktadır. Ayrıca aynı bilgisayarı kullanabilecek birçok kullanıcının bulunması, bir kullanıcının farklı tarayıcı kullanabilmesi, güvenlik duvarının kullanılabilmesi, proxy sunucular üzerinden web sitesine erişildiğinde birçok kullanıcının aynı IP adresiyle internete çıkması gibi durumlardan dolayı, kullanıcı tanımlama işlemi zor ve karmaşık bir hale gelmektedir. Cookie' ler kullanılarak kullanıcının sitedeki hareketleri izlenebilir; ancak birçok kullanıcı bilgilerinin elde edilmemesi amacıyla cookie leri silebilmektedir. Bu yüzden kullanıcı tanımlama işlemini yapabilmek için, başka bir yol kullanılmaktadır.

Web log dosyalarında aynı IP değerine sahip kayıtlar olsa bile, eğer bu kayıtlarda bulunan tarayıcı ya da işletim sistemi bilgisi farklı ise, bu farklı kayıtlar farklı kullanıcı olarak kabul edilmektedir. Bu kabulle birlikte, kullanıcı tanımlama işleminde şöyle bir teknik kullanılmaktadır: Log dosyalarındaki aynı tarayıcı ya da işletim sistemi bilgilerine sahip kayıtlar için kullanıcılar belirlenirken, eğer kullanıcının istekte bulunduğu bir sayfaya, o kullanıcının önceki eriştiği sayfaların hiçbirisinden gelinmediyse, bu erişim de farklı bir kullanıcı tarafından yapılmış anlamına gelmektedir. Bir örnek verilerek şöyle açıklanabilir:

Çizelge 4.1 Erişim, Referrer ve Agent log örneği

IP Adresi	Erişilen Sayfa	Hangi Sayfadan Gelindiği(Referrer)	Tarayıcı
10.1.1.2	A.html	-	Mozilla
10.1.1.2	B.html	A.html	Mozilla
10.1.1.2	L.html	-	Mozilla
10.1.1.2	F.html	B.html	Mozilla
10.1.1.2	R.html	L.html	Mozilla
10.1.1.2	C.html	-	Opera
10.1.1.2	D.html	C.html	Opera

Çizelge 4.1' deki verilerin işlenecek veri olduğunu kabul edelim. Bu veriler erişim zamanına göre sıralıdır ve StatusCode değeri 200 olan, erişimde sorun olmayan sayfaların verileridir. Herbir tarayıcı için kendi içinde kullanıcı tanımlama işlemi yapılmalıdır. Tarayıcı değeri Mozilla olan kayıtlar için kullanıcı tanımlama işlemini uygularsak şöyle bir sonuçla karşılaşırız: İlk olarak A.html sayfasına erişim sağlanmıştır. Daha sonra A.html sayfasından B.html sayfasına tıklanmıştır. Bu iki sayfaya istekte bulunan aynı kişidir. 3.satırdaki kayıt L.html sayfası içindir ve bu sayfaya hiçbir sayfadan tıklanarak gelinmediği için, bu sayfaya

tıklayan kullanıcının yeni bir kullanıcı olduğu kabul edilmektedir. 4.satırda B.html sayfasından F.html sayfasına geçiş yapılmıştır. B.html sayfası ilk kullanıcının eriştiği sayfalarda bulunduğu için, B.html sayfasına erişen kullanıcı ilk kullanıcıdır. 5.satırda L.html sayfasından R.html sayfasına erişilmiştir ve L.html sayfası ikinci kullanıcının eriştiği sayfalarda bulunmaktadır. Bu yüzden R.html sayfasına erişen kullanıcı ikinci kullanıcıdır. 6.satırda tarayıcı tipi değiştiği için, yeni bir kullanıcı istekte bulunmuş demektir. Bu kullanıcı da öncelikle C.html sayfasına, daha sonra da C.html sayfasından D.html sayfasına istekte bulunmuştur. Buna göre bu log örneğinde toplam 3 kullanıcı olmakla birlikte, bu kullanıcıların eriştiği sayfa bilgileri şöyledir:

1.Kullanıcı: A.html – B.html – F.html

2.Kullanıcı: L.html – R.html

3.Kullanıcı: C.html – D.html

Her ne kadar, kullanıcı tanımlama işlemi için bu tekniğe göre algoritma oluşturulsa da, aynı bilgisayarda iki farklı tarayıcı kullanarak web sitelerine erişen aynı kullanıcının bulunabileceği, ya da aynı IP adresini, aynı tarayıcıyı ve aynı web sayfalarına erişen 2 farklı kullanıcının olabileceği gerçeği de göz ardı edilmemelidir; ancak yine de bu tekniğe göre oluşturulan algoritmalar, doğruya yakın sonuçlar vermektedir.

4.1.3 Oturum Tanımlama

Oturum, bir kullanıcının web sayfasında bulunduğu süre içerisinde yaptığı gezinmeler bütünüdür. Oturum tanımlama aşamasının amacı, herbir kullanıcının sayfa erişimlerini oturumlara ayırmaktır. Bunu sağlamanın en basit yöntemi, timeout süresine bakarak oturumlara ayırmaktır. Eğer sayfalar arasındaki süre belli bir limiti aşarsa, kullanıcının yeni bir oturum açtığı kabul edilmektedir. Birçok ticari ürün varsayılan süreyi 30 dakika olarak kabul etmektedir. Bu tez çalışması kapsamında, oturum tanımlama için kullanılan timeout süresi 30 dakika olarak kabul edilmiştir.

4.1.4 Yol Tamamlama

Tam olarak kullanıcı oturumları belirlemekteki başka bir problem, erişim loglarında kayıt edilmeyen önemli erişim bilgilerinin bulunmasıdır. Bu problemin çözümü, yol tamamlama aşamasının ilgi alanına girmektedir. Kullanıcı tanımlama için kullanılan teknikler, yol tamamlama için de kullanılabilir. Eğer istekte bulunulan sayfaya kullanıcının istekte bulunduğu en son sayfadan gelinmediyse, kullanıcının hangi sayfadan geldiği bilgisine ulaşmak için referrer log incelenmelidir. Eğer istekte bulunulan sayfa kullanıcının önceden eriştiği sayfalar arasında bulunuyorsa, kullanıcının tüm tarayıcılarda bulunan "Back" tuşuna bastığı kabul edilmektedir. Eğer referrer log bilgisi anlaşılır, açık değilse site topolojisi de aynı amaç için kullanılabilir. Eğer istekte bulunulan sayfaya kullanıcının önceden eriştiği birden fazla sayfadan erişilebiliniyorsa, en son erişilen sayfadan bu sayfaya erişildiği kabul edilmektedir.

4.2 Örüntü Keşfi

İşlenecek ham veriye ön işlemler basamağındaki işlemler uygulandıktan sonra, veri işlemeye hazır hale gelir. Ön işlem basamağından sonraki basamak örüntü keşfi basamağıdır. Örüntü keşfi, anlamsız verilerden önemli ve gerekli bilgiyi ortaya çıkarmak için, veriye web madenciliği tekniklerinin uygulanma işlemidir. Yaygın olarak kullanılan web madenciliği teknikleri şöyledir (Cooley, R., Srivastava, J., Deshpande, M. ve Tan,P., 2000):

4.2.1 İstatistik

İstatistikler teknikler, kullanıcının web sitesine ziyaretleri hakkındaki bilgileri ortaya çıkarmak içinn kullanılan en çok bilinen metodlardan birisidir. Oturum dosyasını analiz ederek, sayfa görüntüleme bilgileri, harcanan süre gibi farklı değişkenler üzerinde farklı açıklamalı istatistikler analiz teknikleri uygulanabilir. Birçok web trafiği analiz programları en çok erişilen sayfalar, sayfanın ortalama görüntülenme süresi gibi bilgileri içeren raporları periyodik olarak üretirler. Bu sayede web sayfasındaki güvenlik sorunları, sistem performansı ve benzeri konularda bilgiler elde edilebilir.

- Hangi kullanıcılar tarafından hangi sayfalar kullanılıyor?
- Hangi web tarayıcıları ile sayfalara erişiliyor?
- Resim ve diğer bağlı dosyalar olmadan kaç ziyaretçi var?

4.2.2 Birliktelik Kuralları (Association Rules)

Veritabanı içinde yer alan kayıtların birbiriyle olan ilişkilerini inceleyerek, hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan veri madenciliği yöntemleri birliktelik kuralları olarak tanımlanır. Birliktelik kuralları özellikle pazarlama alanında uygulama alanı bulmuştur. Pazar sepeti analizleri adı verilen uygulamalar, ilişkilendirme kuralları tekniğini kullanarak sonuçlara varır. Sonuçlardan faydalanarak da, müşterilerin alışveriş alışkanlıkları belirlenmeye çalışılır. Pazar sepeti analizleri yardımıyla bir müşteri herhangi bir ürünü aldığında, sepetine başka hangi ürünleri de koyduğu belirli bir olasılığa göre ortaya konur. Birlikte satın alınan ürünler belirlendiğinde, mağazalarda raflar ona göre düzenlenerek müşterilerin bu ürünlere daha kolayca erişmeleri sağlanabilir.

Web kullanım madenciliği kapsamında ilişkilendirme kuralları, belirlenen eşik değerini geçen destek değerine sahip, birlikte erişilen sayfalar için uygulanır. Bu sayfalar web sitesi üzerinde birbirine bağlanmamış olabilir, farklı bölümlerde yer alıyor olabilirler. Örneğin, bir kullanıcı bir web sitesinde hem bilgisayar ile ilgili bölümlerde gezinmişken, daha sonra spor ile ilgili bölümlerdeki sayfalarda da gezinmiş olabilir. Bu tür sonuçlara varılmasıyla da, web sitesinin tasarımcısı web sitesinin tasarımı bu tür sonuçlara göre güncelleyerek, kullanıcılar için daha kullanışlı bir web sitesi sağlamış olur.

Birliktelik kurallarının uygulanıp kuralların çıkartılabilmesi için çeşitli algoritmalar geliştirilmiştir. AIS algoritması 1993' te ortaya atılmış, daha sonra yine 1993 yılında SETM algoritması geliştirilmiştir. Daha sonra 1994 yılında bugün de çok kullanılan Apriori ve AprioriTid algoritmaları geliştirilmiştir.

Bu tez çalışması kapsamında işlenecek veri üzerinde birliktelik kuralları yöntemi uygulanmıştır. Bu yöntemi kullanan Apriori algoritması, web logları üzerine uygulanmış ve gerekli ilişki ve kurallar çıkarılmıştır.

Birlikteki kuralları algoritmalarından önemli olanlar kısaca aşağıdaki gibi açıklanabilir:

- **AIS Algoritması**

AIS algoritması geniş nesne kümelerini üretmek için geliştirilmiştir. Veritabanındaki nesne isimlerinin A' dan Z' ye sıralanması kısıtını taşır. AIS algoritması veritabanını birçok kez tarar ve her tarama esnasında tüm işlemleri okur.

- **SETM Algoritması**

SETM algoritmasında L_k geniş nesne kümesinin herbir elemanı iki parametreden oluşur: bunların bir tanesi nesne ismi, diğeri bu nesneyi ayırt etmeye yarayan bir özellik numarasıdır. Algoritma içinde bu numara TID (Transaction Identification) olarak kullanılır. Bu algoritmada TID bilgisinin de tutulması algoritmanın yer kamaşıklığını arttırmaktadır.

- **AprioriTid Algoritması**

Algoritmalar desteği hesaplamak için tüm veritabanını tarar; ancak her aşamada veritabanının tamamının taranmasına gerek olmayabilir. Agrawal, Apriori algoritmasıyla birlikte AprioriTid algoritmasını da sunmuştur. Bu algoritmada da Apriori algoritmasındaki gibi aday nesne kümelerini belirlemek için apriori-gen fonksiyonunu kullanır; ancak ilk geçişten sonra destek seviyesini bulmak için veritabanı taranmaz. Bunun için SETM algoritmasında olduğu gibi herbir eleman için TID bilgisi de bulunmaktadır.

- **OCD Algoritması**

Geniş nesne kümelerini belirlemek için veritabanından alınmış küçük örneklerin çok iyi sonuçlar verebileceği fikrine dayaya OCD (Offline Candidate Determination) algoritması 1994'te ortaya atılmıştır.

- **CARMA Algoritması**

Kullanıcıya her taramadan sonra oluşan kuralları gösterip, minimum destek ve güven seviyelerini değiştirme olanağı veren CARMA algoritması 1999' da Hidber tarafından ortaya atılmıştır.

4.2.2.1 Destek ve Güven Ölçütleri

Birliktelik kuralları, işlemlerden oluşan ve herbir işlemin de elemanlarının birlikteliğinden oluştuğu düşünülen bir veritabanında, bütün birliktelikleri tarayarak, sık
tekrarlanan birliktelikleri veritabanından ortaya çıkarmaktır (Agrawal, R. ve Shafer, J.C., 1996).

Birliktelik kuralının matematiksel modeli Agrawal, Imielinski ve Swami tarafından 1993 yılında sunulmuştur (Agrawal vd., 1993). Bu modelde, $I = \{i_1, i_2, .., i_m\}$ kümesine"ürünler" adı verilmektedir. *D,* veri bütünlüğündeki tüm hareketleri, T ise ürünlerin her bir hareketini simgeler. *TID* ise, her harekete ait olan tek belirteçtir.

Birliktelik kuralı şu şekilde tanımlanabilir;

$A_1, A_2, ..., A_m \rightarrow B_1, B_2, ..., B_n$ (4.1)

Eşitlik 4.1' de yer alan, A_i ve B_j, yapılan iş veya nesnelerdir. Bu kural, genellikle "$A_1,A_2,...,A_m$" iş veya nesneleri meydana geldiğinde, sık olarak "$B_1,B_2, ...,B_n$" iş veya nesnelerinin aynı olay veya hareket içinde yer aldığını belirtir.

Birliktelik kuralında incelenecek değerler arasındaki ilişkileri ortaya koymak için ***"destek (support)"*** ve ***"güven (confidence)"*** gibi iki ölçütten yararlanılır. Bu ölçütlerin hesaplanmasında "destek sayısı" adı verilen bir değer kullanılır. "*Destek ölçütü*" bir ilişkinin tüm olaylar içinde hangi oranda tekrarlandığını belirler. "*Güven ölçütü*", bir değerin diğer değerle aynı olayda geçme olasılığını ortaya koyar.

A değerini içeren olayların, B değerini de içerme durumu, yani birliktelik kuralı A→B biçiminde gösterilir. Bu durumda kural ölçütü Eşitlik 4.2' deki gibi ifade edilebilir.

$$\text{destek}(A,B) \rightarrow \frac{\text{sayı}(A,B)}{N} \quad (4.2)$$

Eşitlik 4.2' de sayı(A,B) destek sayısı, A ve B değerlerini birlikte içeren olayların sayısını göstermektedir. N ise, tüm olayların sayısını göstermektedir. A ve B değerlerinin birlikte bulunma olasılığını ifade eden güven ölçütü Eşitlik 4.3' te görüldüğü gibi, şu şekilde hesaplanır:

$$\text{güven}(A,B) \rightarrow \frac{\text{sayı}(A,B)}{\text{sayı}(A)} \quad (4.3)$$

Birliktelik kuralları belirlenirken, destek ve güven ölçütlerinin dışında, bu değerleri karşılaştırmak için "eşik değer"e gereksinim vardır. Hesaplanan destek ve güven ölçütlerinin destek ve güven değerlerinden büyük olması beklenir. Hesaplanan destek ve güven ölçütü ne kadar büyükse, birliktelik kurallarının da o derece güçlü olduğuna karar verilir. Güven değerinin % 100 olması durumunda, kural bütün veri analizlerinde doğrudur ve bu kurallara "kesin" denir.

Birliktelik kuralına ilişkin olarak geliştirilen bazı algoritmalar şunlardır; AIS (Agrawal ve diğerleri., 1993), SETM (Houtsma ve Swami, 1995), Apriori (Agrawal ve Srikant, 1994), Partition (Savasere ve diğerleri., 1995), RARM - Rapid Association Rule Mining (Das ve diğerleri., 2001), CHARM (Zaki ve Hsiao, 2002). Bu algoritmalar içerisinde, ilk olanı AIS, en bilineni ise Apriori algoritmasıdır (Agrawal R. ve Srikant R., 1995) .

4.2.2.2 Apriori Algoritması

Literatürde birliktelik kuralı çıkaran değişik algoritmalar bulunmaktadır. Apriori Algoritması, birliktelik kuralı çıkarım algoritmaları içerisinde en fazla bilinen algoritmadır (Agrawal, R., Imielinski, T. ve Swami, A., 1993). Apriori Algoritmasının ismi, bilgileri bir önceki adımdan aldığı için "prior" anlamında Apriori'dir (Agrawal, R. ve Srikant, R., 1994). Bu algoritma temelinde iteratif (tekrarlayan) bir niteliğe (Han, J. ve Kamber, M., 2006) ve hareket bilgileri içeren veritabanlarında sık geçen öğe kümelerinin keşfedilmesinde kullanılır. Apriori algoritmasının sözde kodu (4.4)' te verilmiştir (Agrawal, R., Imielinski, T. ve Swami, A., 1993).

Apriori Algoritmasına özüne göre, eğer k-öğe kümesi (k adet elemana sahip öğe kümesi) minimum destek ölçütünü sağlıyorsa, bu kümenin alt kümeleri de minimum destek ölçütünü sağlar. Bu algoritmada sık geçen öğe kümelerini bulmak için birçok kez veritabanını taramak gerekir. İlk taramada bir elemanlı minimum destek değerini sağlayan sık geçen öğe kümeleri bulunur. İzleyen taramalarda bir önceki taramada bulunan sık geçen öğe kümeleri aday kümeler adı verilen yeni potansiyel sık geçen öğe kümelerini üretmek için kullanılır. Aday kümelerin destek değerleri tarama sırasında hesaplanır ve aday kümelerinden minimum destek değerini sağlayan kümeler o geçişte üretilen sık geçen öğe kümeleri olur. Sık geçen öğe kümeleri bir sonraki geçiş için aday küme olurlar. Bu süreç yeni bir sık geçen öğe kümesi bulunmayana kadar devam eder (Alataş, B., 2003).

Apriori algoritmasında önce aday öğe kümeleri oluşturulur. Bu kümeler potansiyel olarak sık geçen öğe kümeleridir. C ile gösterilir ve C[1], C[2], C[3], ….C[k] olarak k-öğe kümesini oluştururlar. Her c[k] öğe kümesi c[k-1] öğe kümesini içerir ve C[1] < C[2] < C[3]< ….< C[k] şeklinde sıralıdır. Sık geçen k-öğe kümeleri ise L ile gösterilir ve minimum destek kıstaslarını sağlarlar.

Klasik Apriori Algoritması Özet Kodu

1) L_1 *= {sık geçen 1-öğe kümesi};*

2) for (k=2; $L_{k-1} \neq \emptyset$*; k++) do begin*

3) C_k*=apriori-gen (*L_{k-1}*); // Yeni adaylar*

4) forall transactions-hareketler t $\in D$ *do begin*

5) C_t *= subset (*C_k*, t); // Adaylar t içindedir*

6) forall candidates – adaylar c $\in Ct$ *do*

7) c.count++;

8) end

9) $L_k = \{c \in C_k \mid c.count \geq minsup\}$

10) end

11) Answer = $\cup_k L_k$*;* (4.4)

Apriori algoritmasının klasik özet kodunda yer alan Apriori-Gen fonksiyonu, (k-1) adet öğeye sahip L_{k-1} sık geçen öğe kümesini kullanarak k adet öğeye sahip aday kümeleri oluşturur. Bu fonksiyon ile, ilk önce, L_{k-1} sık geçen öğe kümesine kendisi ile birleştirme (join) işlemi uygulanır. Birleştirme işleminde L_{k-1} sık geçen öğe kümesinin her satırında yer alan son öğe haricinde diğer öğelerin çapraz olarak benzerliği aranır ve son öğe haricinde diğer öğelerle yakalanan benzerliklerden yeni aday öğe kümeleri oluşturulur. Oluşan kümeler budama (prune) adımı ile budanarak fonksiyondan dönülür. Apriori-Gen fonksiyonunun sözde kodu (4.5)' teki gibidir (Agrawal, R., Imielinski, T. ve Swami, A., 1993).

Apriori-Gen Fonksiyonu

1)insert into C_k

select $p.items_1, p.items_2,.....,p.items_{k-1}, q.item_{k-1}$

from $L_{k-1}\, p, L_{k-1}\, q$

where $p.item_1=q.item_1, ..., p.item_{k-2}=q.item_{k-2},$

$p.item_{k-1} < q.item_{k-1};$

2)forall itemsets $c \in C_k$ *do*

forall (k-1)-subsets s of c do

if $(s \notin L_{k-1})$ *then*

delete c from C_k (4.5)

Budama işleminde; c aday kümesinin (k-1) öğeye sahip alt kümelerinden L_{k-1} sık geçen öğe kümesinde yer almayan tüm alt kümeler silinir. Budama işleminde, C_k aday öğe kümesindeki öğelerin alt kümelerinin L_{k-1} sık geçen öğe kümesindeki varlığı kontrol edilir, bir öğenin alt kümelerinden biri, L_{k-1} sık geçen öğe kümesinde yer almıyorsa ilgili öğe değerlendirme dışı kalır ve C_k aday öğe kümesinden silinir (Agrawal, R. ve Srikant, R., 1994).

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

Kod içinde kurulan döngü yapısı ile ilk aşamada L_1 sık geçen öğe kümesinin öğelerinin ikili kombinasyonuna benzer bir şekilde ($L_1 \infty L_1$) yeni bir küme oluşur. Bu işleme birleştirme (join) adı verilir. Bu işlem ile oluşan kümelere de aday öğe kümeler adı verilir ve C harfi ile simgelenir. Oluşan bu aday öğe kümesinin her elemanı iki adet öğeden oluştuğu için C_2 ifadesi ile isimlendirilir. Bu aday küme apriori-gen fonksiyonu ile budama işlemine tabi tutulur ve C_2 kümesinin elemanlarına ait alt kümelerinin L_1 öğe kümesinde olup olmadığına bakılır. Alt kümelerden L_1 içinde yer almayan küme elemanları C_2 aday kümesinden silinir. Apriori algoritması uygulanan veri topluluğu tekrar taranarak budama işleminden geçen C_2 aday kümesi elemanlarının kaç adet hareket kaydı içinden geçtiği (destek ölçütü) bulunur.

Bulunan destek sayaç bilgileri doğrultusunda C_2 aday kümesi elemanlarının destek ölçütü minimum destek değerine eşit veya büyük destek değerine sahip olan elemanları L_2 sık geçen öğe kümesini oluşturur. Döngü bir sonraki aşamada L_2 kümesi öğelerinin üçlü kombinasyonu ile yeni bir aday öğe kümesi oluşturur ve bu küme C_3 ifadesi ile simgelenir. İlk aşamada olduğu gibi bu kümede budama işleminden geçer ve budama işleminden sonra minimum destek seviyesinin üstünde kalan elemanları ile L_3 sık geçen öğe kümesi oluşturulur. Döngü her dönüşünde öğe sayısını artırarak devam eder. Bu süreç yeni bir sık geçen öğe kümesi bulunamayana kadar sürer.

4.2.3 Sınıflandırma

Sınıflandırma, bir veriyi daha önceden tanımlanmış sınıflara dağıtma tekniğidir. Resim, örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, kalite kontrol çalışmaları ve pazarlama konuları sınıflandırma tekniklerinin bolca kullanıldığı alanlardır. Web sitesinde web master veya pazarlamacı sınıflandırma tekniğini kullanarak müşterilerinin hangi sınıf veya kategoride bir profile sahip olduğunu belirleyebilir. Sınıflandırma işleminde, verilen bir sınıf veya kategorinin özelliklerini en iyi biçimde açıklamak için seçim ve açığa çıkarma uygulamalarına ihtiyaç duyulur. Sınıflama iki adım içeren bir işlemdir. Birinci adımda tahmin için kullanılacak bir model oluşturulmaktadır. İkinci adımda, oluşturulan bu model, sınıfı belli olmayan veriler üzerinde uygulanarak sınıflar tahmin edilmektedir. Sınıflandırma; karar ağaçları, Bayes sınıflayıcıları, en yakın komşu ve destek vektör makineleri gibi denetlenen tümevarımsal öğrenim algoritmaları kullanarak yapılabilir (Cooley, R., Srivastava, J., Deshpande, M. ve Tan,P., 2000). Örneğin, "*Genç kadınlar küçük araba satın alır, yaşlı, zengin erkekler büyük, lüks araba satın alır.*" gibi bir sınıflandırma yapılırsa, genç kadınların okuduğu bir dergiye reklam verilirken küçük araba modelinin reklamı verilebilir. Sınıflandırma, matematiksel olarak şöyle tanımlanabilir:

$D=\{t_1,t_2,...,t_n\}$, bir veritabanı olsun ve herbir t_i bir kaydı temsil etsin.
$C=\{c_1,c_2,...,c_m\}$, ise m adet sınıftan oluşan sınıflar kümesini temsil etsin.

$f: D \rightarrow C$ ve herbir t_i bir sınıfa dahil olmalıdır.
Herbir C_j ayrı bir sınıftır ve herbir sınıf kendisine ait kayıtları içerir.

Yani;

$$C_j=\{t_i \mid f(t_i) = C_j,\ 1\leq i\leq n \text{ ve } t_i \in D\} \qquad (4.6)$$

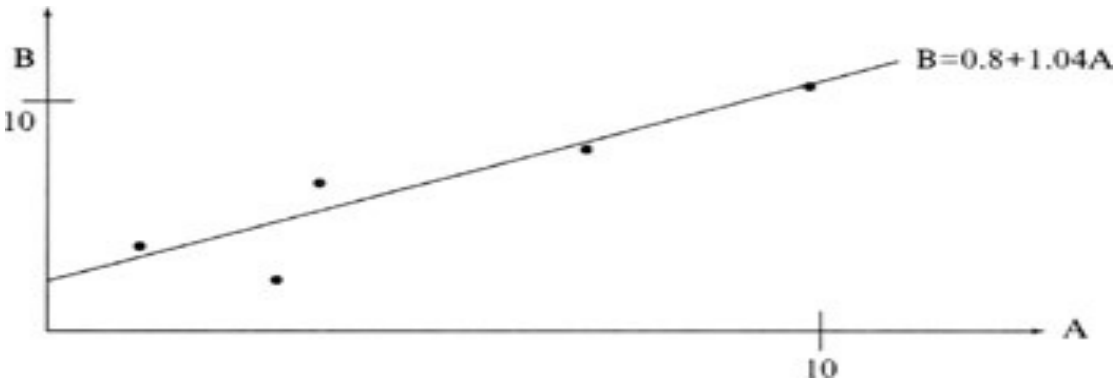
4.2.4 Eğri Uydurma

Sürekli değişkenlerin öngörüsü, regresyon (eğri uydurma) olarak adlandırılan bir istatistiksel yöntemle tespit edilebilir. Regresyon analizinin amacı değişik girdi değişkenlerini çıktı değişkeni ile ilişkilendirecek en iyi modelin çıkarılmasıdır. Regresyon analizi bir Y değişkeninin diğer bir veya daha çok X1, X2,…,,Xn değişkenleri ile ilişkisinin belirlenmesi sürecidir. Y, yanıt çıktısı veya bağımlı değişken olarak adlandırılır. Xi değişkenleri, girdi veya bağımsız değişkenler olarak adlandırılır. Bir veri kümesinde bulunan ilişki, regresyon denklemi (modeli) ile karakterize edilir. En çok yaygın regrasyon modeli denklemi;

$$Y = \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \ldots + \beta_n \cdot X_n \tag{4.7}$$

Örneğin,

A	B
1	3
8	9
11	11
4	5
3	2



Verilen örnek veri setindeki A ve B değişkenleri arasındaki ilişki Eşitlik 4.8 ile ifade edilebilir.

$$B= \alpha+\beta.A \quad (4.8)$$

α ve β katsayılarının değerleri matematiksel olarak bulunur:

$\alpha= 0.8$

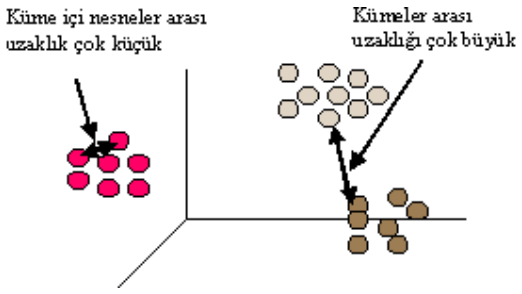
$\beta= 1.04$

Optimal regrasyon denklemi ise Eşitlik 4.9' daki gibi bulunur.

$$B=0.8+1.04\beta \quad (4.9)$$

4.2.5 Kümeleme

Soyut ve somut benzer objelerin bir grupta toplanması kümeleme olarak adlandırılır. Benzer ya da birbiriyle ilişkili olan obje grupları aynı küme oluştururken, farklı obje ya da ilişki olmayan objeler başka bir küme oluşturur (Han, J. ve Kamber M., 2006).



Şekil 4.4 Küme yapısı

Şekil 4.4' te görüldüğü gibi küme içindeki nesneler arasındaki benzerlik (Intra-Cluster similarity) fazla olduğu için aynı kümede yer almaktadır, oysaki kümeler arası benzerlik (Inter-Cluster similarity) bir küme içindeki nesneler ile diğer küme içindeki nesneler arasındaki benzerliğin az olduğunu göstermektedir (Han, J. ve Kamber, M., 2006).

Veri matrisinde n veri sayısı, p nitelik sayısı olmak üzere,

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \tag{4.10}$$

Fark Matrisi:

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & (3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix} \tag{4.11}$$

İki veri arasındaki uzaklık (d(i,j)) ile ifade edilir. Veriler arasındaki benzerlik ve fark ölçümü, eğer p = 2 ve d değişkeni Öklid Uzaklığı (Euclidean Distance) ise Eşitlik 4.12' deki gibidir.

$$d(i,j)= \sqrt{(x_{i1}- x_{j1})^2 + (x_{i2} - x_{j2})^2} \tag{4.12}$$

Özellikleri:

1.$d(i,j) \geq 0$

2.$d(i,i) = 0$

3.$d(i,j) = d(j,i)$

4.$d(i,j) \leq d(i,h)+d(h,j)$

Kümeleme örüntü tanıma, görüntü işleme ekonomi bilimi (özellikle market araştırma), dünya çapında ağ (world wide web) üzerinde doküman sınıflandırılması, benzer ortak arkadaş grupları keşfetme, veri madenciliği, istatistik, biyoloji ve makine öğrenmesi (machine learning) gibi pek çok alanda kullanılır (Han, J. ve Kamber, M., 2006).

Kümeleme modelinde, sınıflandırma modelinde olan veri sınıfları yoktur. Verilerin herhangi bir sınıfı bulunmamaktadır. Sınıflandırma modelinde, verilerin sınıfları bilinmekte ve yeni bir veri geldiğinde bu verinin hangi sınıftan olabileceği tahmin edilmektedir. Oysa kümeleme modelinde, sınıfları bulunmayan veriler gruplar halinde kümelere ayrılırlar. Bazı uygulamalarda kümeleme modeli, sınıflama modelinin bir önişlemi gibi görev alabilmektedir (Özekes, S., 2003).

Marketlerde farklı müşteri gruplarının keşfedilmesi ve bu grupların alışveriş örüntülerinin ortaya konması, biyolojide bitki ve hayvan sınıflandırmaları ve işlevlerine göre benzer genlerin sınıflandırılması, şehir planlanmasında evlerin tiplerine, değerlerine ve coğrafik konumlarına göre gruplara ayrılması gibi uygulamalar tipik kümeleme uygulamalarıdır. Kümeleme aynı zamanda web üzerinde bilgi keşfi için dokümanların sınıflanması amacıyla da kullanılabilir (Özekes, S., 2003).

Web madenciliğinde genel olarak iki kümeleme yaklaşımı vardır: Kullanıcı grupları (User Clusters) ve Sayfa Grupları (Page Clusters) .

Kullanıcı grupları yaklaşımında amaç, benzer sayfa görüntülemesi yapan kullanıcıları tespit edip bir grup içerisine almaktır. Bu yöntem özellikle web kişileştirme işleminde oldukça yararlıdır. Örneğin, bir portal içerisinde oyun ve spor sayfalarına girenleri bir grup içerisinde toplayıp, kişilerin bir sonraki bağlantısında oyun ve spor konulu reklamların ekrana gelmesi gibi.

Sayfa grupları yaklaşımında ise, benzer içerikli sayfaların bir araya gruplandırılması, özellikle arama motorları için çok yararlı olmaktadır. Böylelikle bir kullanıcının aramış olduğu bilgilere daha hızlı şekilde ulaşabilmesi sağlanır. Buna örnek olarak doküman kümeleme verilebilir.

Doküman kümelemede, doküman içinde geçen terimlere göre aynı konudaki dokümanlar gruplanır. Her doküman içinde sık geçen terimler bulunur ve bu terimlerden ve ağırlıklardan yararlanarak bir benzerlik ölçütü geliştirilir. Bulunan ölçülere göre kümeleme yapılarak yeni bir dokümanın hangi dokümanlarla benzer olduğu tespit edilir.

Genel olarak başlıca kümeleme yöntemleri dört farklı şekilde sınıflandırılabilir:

- Bölümleme Metotları (Partitioning methods): Veriyi bölerek, her grubu belirlenmiş bir kritere göre değerlendirir.Bu grupta yer alan örnek algoritmalar, K-Means Algoritması, K-Medoids Algoritması dır.
- Hiyerarşik Metotları (Hierarchical methods): Veri kümelerini ya da nesneleri önceden belirlenmiş bir kritere göre hiyerarşik olarak ayırır.
- Yoğunluk Tabanlı Metotlar (Denisty-base methods): Nesnelerin yoğunluğuna göre kümeleme oluşturur.
- Model Tabanlı Metotlar (Model-base methods): Her kümenin bir modele uyduğu var-sayılır ve bu modellere uyan verileri gruplandırır.

4.2.6 Sıralı Örüntüler

Sıralı örüntüler; oturumlar arasında örüntü bulmaya çalışır. Sıralı örüntü bulma işleminde, belirli zaman aralıklarında oturumlar incelenir ve karşılaştırmalar yapılır. Sıralı örüntülerin bulunması gelecekteki eğilimi tahmin edecek web pazarlamacıları için oldukça anlamlıdır. Böylece ilanlar belirli kullanıcı gruplarına yönlendirilebilecektir. Sıralı örüntüler için, eğilim analizi, değişen nokta bulma veya benzerlik analizleri gibi bazı geçici analiz tipleri kullanır.

5. GELİŞTİRİLEN UYGULAMA

Bu tez çalışmasının konusu olan web kullanım madenciliği için kullanılan veri setinin kaynağı, çok sık giriş yapılan ve içeriği sürekli artan, bir Türk portal sitesinin, içeriği video olan bölümüdür. Veri setinin hangi siteye ait olduğu bilgisinin gizli tutulacağı, telif haklarından dolayı portal sitesinin sahiplerinin isteği üzerine imzalanan bir anlaşmayla sağlanmış olduğundan, sitenin adı gizli tutulacaktır. Sunucularda kullanıcıların hareketlerinin kayıt olduğu log dosyalarından veriler ve örnekler sunulurken, bu portal sitesinin adının geçtiği yerler, "***webmining***" olarak değiştirilecek ve değiştirilen yerler kırmızı renkle yazılacaktır. Geliştirilen uygulama programlama dili olarak Visual Studio .NET 2005 ortamında C# dili kullanılarak yazılmış olup, veritabanı programı olarak Microsoft SQL Server 2005 kullanılmıştır.

Bu bölümde tez çalışmasında yapılam tüm işlemler adım adım gösterilecek olup, yazılan kod ve veritabanı sorgularından örnekler verilerek, yapılan işlemler detaylı olarak anlatılacaktır. Kullanıcıların tıkladıkları video linklerine birliktelik kuralı uygulanarak, sonuçta çıkan kurallar belirtilecektir.

5.1 Kullanılan Veri Seti

Bu tez çalışmasında kullanılan log dosyaları, portal sitesinin sunucularından temin edilmiş olup, loglar sunucudan metin dosya olarak alınmıştır. İşlem yapılan portal sitesinin popüler ve çok tıklanan bir site olmasından dolayı, bir log dosyasının boyutu ortalama olarak 150 MB' ı bulmaktadır. Bir metin dosyası için 150 MB büyük bir değer olarak kabul edilebilir. Bu yüzden siteye yapılan sadece tek bir günlük kullanıcı erişimlerinin tutulduğu log dosyaları veritabanına alınmıştır. Bir günlük erişimler bile veritabanına atıldıktan sonra, veritabanında yaklaşık 90 GB yer kaplamaktadır. Log dosyalarının formatı W3C Extended log formatıdır. Log dosyasının header bilgisi Şekil 5.1' de görüldüğü gibi şöyledir:

```
#Software: Microsoft Internet Information Services 6.0

#Version: 1.0

#Date: 2009-02-05 23:00:00

#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent)
cs(Cookie) cs(Referer) cs-host sc-status sc-substatus sc-win32-status sc-bytes cs-bytes
```

Şekil 5.1 Kullanılan log dosyasındaki header bilgisi

Log dosyasında kayıtlı olan, bir kullanıcının bir hareket satırı bilgisi ise Şekil 5.2' de görülebilir.

2009-02-05 22:59:59 192.168.34.73 GET /ep/SevenlerForContent.aspx contentname=video&contentmcid=1200013090&p=1 80 - 78.172.188.105 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+AntivirXP08;+GTB5;+.NET+CLR+2.0.50727) webminingUv=TRxq12337880809849954792699 2xqIstanbulxq080309;+__utma=21975975.677974878.1233788378.1233788378.1233874489.2;+__utmb=21975975;+__utmz=21975975.1233788378.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(none);+cityList2=75918%7C%DDstanbul%3B75784%7CAnkara%3B75919%7C%DDzmir%3B;+vsd={911C73F9-A878-4F8F-AE6C-CA0F899D46B8};+numToCompare=1233874789686;+__utmc=21975975;+cookieControl=cookieControl;+mytag=g1xj2xc69xco0xgrup3xi8xi11;+loginstate=6499C4B8387F4D92878F7D54DAB29690MIHFBgkrBgEEAYI3WAOggbcwgbQGCisGAQQBgjdYAwGggaUwgaICAwIAAQICZgECAUAECHJmPjwCQ2SoBBB1RLr01JMrDR2TKM0IoSMXBHgJE6vFrQIhreW4bXkplusBBsvYuVd3VNjwTDL2DNMMWQ7lyM10MXZTrL9nANNZMYVplusInZplusBMP0wmnQplusLqJrqe5DTRlIp9eEJM0LtYy8plus7VorH4lplusGjcyIxgj0Nuhx2Y47NyfL4IDE2G0iBJ8Xi4wzjgMapD1FLp1HxKwequals;+rememberanceexx=2;+rememberthisuserexx=Y29icmV0dGk3Nzc3;+uvid=a327bcee5d0a8bde41c0325e12bd3dc1cobretti7777;+wmdata=-3:::-3;+eksavupload=1;+mylcount=-1.1233874837765.0.1233874837765.-1.1233874837765.0.1233874838937;+on_my=1%3Acobretti7777%3A0%3A%3A0;+imprv=cobretti7777%7C1;+ASP.NET_SessionId=k2nxiwrjonwlhffzb3bha055;+__utma=34328245.841498267.1233874854.1233874854.1233874854.1;+__utmb=34328245;+__utmc=34328245;+__utmz=34328245.1233874854.1.1.utmccn=(referral)|utmcsr=video.webmininglutmcct=/|utmcmd=referral;+wvsdwebmining=1233874855812;+__webminginguv=1233876665359 http://video.webmining.com/cemalulas/cici_kedi/13090/ webmining.com 200 0 0 6953 1699

Şekil 5.2 Kullanılan log dosyasındaki bir kayıt satırı

Şekil 5.2' de görüldüğü üzere, log dosyasındaki herbir satır, log dosyası formatına uygun olarak sunucularda tutulmaktadır. Kullanıcıların sitedeki bir videoya tıklamaları için üye olmaları zorunlu olmadığı için, log dosyasında da cs-username alanına karşılık gelen değer "-" olarak bulunmaktadır. cs(User-Agent) alanına karşılık gelen değer, versiyonlarıyla beraber kullanılabilecek tüm tarayıcı isimleridir. cs-uri-stem alanı, istekte bulunulan sayfanın bağlantı adresini tutarken, cs(Referer) alanı ise, bu istekte bulunulan sayfaya hangi bağlantı adresinden tıklanarak geldiği bilgisini tutmaktadır. c-ip alanı sitede harekette bulunan kullanıcının IP

bilgisini, s-ip bilgisi de sabit değerde olup, sunucunun IP bilgisini ifade etmektedir. sc-status alanına karşılık gelen değer de, kullanıcının istekte bulunduğu bağlantı adresine erişip erişemediği, eğer erişemediyse aldığı hatanın kodunu tutmaktadır. Bu alanın alabileceği tüm değerler Ek 1'de mevcuttur.

İşlenecek log dosyaları veritabanındaki T_ORIGINAL_DATA tablosuna alınmış olup, tablo üzerinde geri dönülemeyecek bir işlem yapıldığında log dosyalarındaki kayıtların tekrar atılmasını önlemek amacıyla, bu tablodaki kayıtlar üzerinde hiçbir işlem yapılmamıştır ve log dosyalarındaki veriler orijinal haliyle bu tabloda tutulmaktadır. Bu tablonun bir kopyası olan T_LOG_DATA tablosunda asıl web kullanım madenciliği ile ilgili işlemler yapılmıştır. T_LOG_DATA ve T_ORIGINAL_DATA tablolarının yapısı şöyledir:

Table - dbo.T_LOG_DATA

Column Name	Data Type	Allow Nulls
RECORD_ID	bigint	☐
DATE	datetime	☑
TIME	nvarchar(250)	☑
S_IP	nvarchar(250)	☑
CS_METHOD	nvarchar(250)	☑
CS_URI_STEM	nvarchar(250)	☑
CS_URI_QUERY	nvarchar(250)	☑
S_PORT	nvarchar(250)	☑
REFERRER	nvarchar(250)	☑
CS_USER_NAME	nvarchar(250)	☑
C_IP	nvarchar(250)	☑
CS_USER_AGENT	nvarchar(250)	☑
CS_COOKIE	nvarchar(250)	☑
CS_HOST	nvarchar(250)	☑
SC_STATUS	nvarchar(250)	☑
SC_SUBSTATUS	nvarchar(250)	☑
SC_WIN32_STATUS	nvarchar(250)	☑
SC_BYTES	nvarchar(250)	☑
CS_BYTES	nvarchar(250)	☑
CREATED_DATETIME	datetime	☑
DELETED	int	☑
		☐

Şekil 5.3 T_LOG_DATA tablosunun yapısı

Şekil 5.3' te görüldüğü gibi, T_LOG_DATA tablosundaki kolonlar, log dosyası formatındaki alanlara göre oluşturulmuştur. Bu kolonlardan farklı olarak, RECORD_ID kolonu bu tablonun birincil anahtar alanı olup, birincil anahtar olduğundan dolayı da bu tabloda otomatik olarak index olarak da yaratıldığından, çeşitli sorgularla tablodan veri çekildiğinde, veriye daha çabuk ulaşmayı sağlamaktadır. DELETED kolonu ise, çeşitli işlemler esnasında göz ardı edilecek verileri belirlemek için kullanılmaktadır. DELETED kolonu varsayılan değer olarak 0 olmakla birlikte; değeri 0 ise, o satırdaki veri göz ardı edilmeyecek, 1 ise satırdaki veri göz ardı edilecek demektir. İlgili sorgu cümlelerinde DELETED=0 filtresi eklenerek işlemler yapılmıştır.

T_ORIGINAL_DATA tablosundaki kayıt sayısı 10.454.274 satırdır. Yani, bu değer işlenecek veri olan bir günlük kullanıcı erişimlerinin sunucuda bıraktığı kayıt sayısıdır. Bir günlük log sayısının bu kadar fazla olması, bu siteye erişimin çok yüksek olduğunu göstermektedir.

5.2 Veri Seti Üzerinde Yapılan Ön İşlem Aşaması

Kullanılan veri setini işlemeye başlamadan önce, web kullanım madenciliğinin ilk aşaması olan ön işlemdeki adımlar log dosyalarına uygulanmıştır. Bu aşama atlanarak herhangi bir web kullanım madenciliği işlemi yapılması doğru değildir; çünkü bu aşamayla beraber işlenecek veri seti üzerindeki tüm gereksiz, kirli, gürültülü veriler de temizlenmiş olmaktadır. O yüzden ön işlem aşaması, web kullanım madenciliği için en önemli aşamalardan birisidir. Log dosyaları üzerinde yapılan ön işlemler sırayla şöyledir:

1. Öncelikle log kayıtlarında istekte bulunulan sayfaların uzantılarına bakılarak, kayıtlarda silme işlemi gerçekleştirilmiştir. Tarayıcıdan bir sayfaya istekte bulunulduğunda, sayfanın içeriğini göstermekle birlikte, o sayfadaki resimler, scriptler ve arka tarafta yüklenen diğer yapılar da log dosyasına bir kayıt olarak atılır; ancak bu kayıtların web kullanım madenciliği için bir önemi yoktur, o yüzden bu kayıtların silinmesi gerekmektedir. İstekte bulunulan sayfa uzantısı .gif, .jpeg, .jpg, .ico, .png, .js, .css, .swf, .asmx, .xml, .axd, .ashx olan kayıtlar T_LOG_DATA tablosundan silinmiştir. Toplam kayıt sayısı olan 10.454.274 kaydın içinde 4.839.012 kayıt silinmiştir. Bu da log kayıt sayısının %46' sı kadarıdır. Bu işlem T_LOG_DATA tablosunun CS_URI_STEM kolonundaki değere göre bakılarak yapılmıştır. Silme işlemi için yazılan ve çalıştırılan SQL sorgu cümlesi şöyledir:

DELETE FROM T_LOG_DATA WHERE (CS_URI_STEM LIKE '%.gif') OR (CS_URI_STEM LIKE '%.jpeg') OR (CS_URI_STEM LIKE '%.jpg') OR (CS_URI_STEM LIKE '%.ico') OR (CS_URI_STEM LIKE '%.png') OR (CS_URI_STEM LIKE '%.js') OR (CS_URI_STEM LIKE '%.css') OR (CS_URI_STEM LIKE '%.swf') OR (CS_URI_STEM LIKE '%.asmx') OR (CS_URI_STEM LIKE '%.xml') OR (CS_URI_STEM LIKE '%.axd') OR (CS_URI_STEM LIKE '%.ashx')

Silinen kayıtlara ilişkin istatistiki bilgi Çizelge 5.1 'de gösterilmektedir:

Çizelge 5.1 Log dosyasından silinen uzantılar ve silinme sayıları

Dosya Uzantısı	Silinen Kayıt Sayısı
.gif	2.239.197
.jpeg	1.412.098
.js	1.212
.css	1.472
.swf	1.036
.asmx	413.300
.xml	67.814
.axd	13.964
.ashx	688.919
.jpg	0
.ico	0
.png	0

2. Log kayıtlarından istekte bulunulan sayfaların uzantısına bakılarak gerekli temizleme işlemi yapıldıktan sonra, 2.işlem olarak, SC_STATUS alanının değerine bakılarak temizleme işlemi yapılmıştır. Bir sayfa görüntülenmeye çalışıldığında her zaman başarılı bir şekilde sayfaya erişilemeyebilinebilinir. Sunucudaki bir sorundan ya da bağlantı yapan kullanıcının internet hızının yavaşlığından zaman aşımına uğraması gibi sebeplerden dolayı, sayfa görüntülenmesinde sorun yaşanabilir. Bu gibi durumlarda log dosyasına bir kayıt atılır ve bu kayıtlar, log kayıtlarında SC_STATUS alanı 200 değerinden farklıysa anlaşılabilmektedir. 2.işlem olarak da log kayıtlarında SC_STATUS alanı 200 'den farklı olanlar silinmiştir ki bu kayıtların web kullanım madenciliği için bir önemi yoktur. Toplam kayıt sayısı olan 10.454.274 kaydın içinde 571.416 kayıt silinmiştir. Yani bir günlük kullanıcı erişimi içinde, 571.416 kez istekte bulunulan sayfaya çeşitli sebeplerden dolayı erişilememiştir. Bu da yaklaşık %5' e denk gelmektedir. Çizelge 5.2' de log kayıtlarının SC_STATUS değerine göre sayıları belirtilmiştir. Silme işlemi için yazılan ve çalıştırılan SQL sorgu cümlesi şöyledir:

DELETE FROM T_LOG_DATA WHERE SC_STATUS!=200

Çizelge 5.2 Log kayıtlarındaki SC_STATUS değerine göre kayıt sayıları

SC_STATUS Değeri	Kayıt Sayısı
200	9882858
500	401390
302	151394
404	18275
400	182
403	81
304	77
501	14
206	3

3. İkinci adımdaki SC_STATUS alanına göre silme işlemi tamamlandıktan sonra, 3. işlem olarak CS_USER_AGENT alanının değerine göre silme işlemi yapılmıştır. Sitede istekte bulunulan sayfayı görüntülemek için kullanılan tarayıcı adı CS_USER_AGENT alanına kaydedildiği gibi, crawler (spider) dediğimiz yapılar da görüntülemeye çalıştığımız sayfada bulundukça log dosyasına bir kayıt olarak atılır; ancak crawler ‘ ların web kullanım madenciliği için bir önemi yoktur. O yüzden bu kayıtların log kayıtlarından silinmesi gerekmektedir.

Crawler denilen yapılar, arama motorlarının bir websiteyi gezen ve bilgi toplayan programlarıdır. Arama motoru örümceği de denilmektedir. Arama motorları siteleri otomatik olarak ziyaret eder ve hemen hemen tüm sayfalarınızı hafızalarına alırlar. Bunun için de bir çeşit robot kullanırlar. Bu robotlar sitelerden bir sayfayı ziyaret ettikten sonra o sayfada bulunan tüm linkleri de ziyaret ederler. Mesela anasayfayı ziyaret ederlerse ve buradan bütün sayfalara link verilmişse, oraları da dolaşacak demektir. Ayrıca site dışından bir siteye link verildiyse orayı da ziyaret edecektir ve eğer bu site listelerinde yoksa onu da ekleyecektir. Crawler’ ların ve kötü amaçlı yazılımların listesi, *http://www.user-agents.org/* internet adresinden güncel olarak indirilmiştir ve temizleme işlemi de bu listeye göre yapılmıştır. Bu site; tarayıcılar, crawler’lar, proxy sunucular, yükleme araçlarının isimlerini içermektedir ve içerik günlük olarak güncellenmektedir. Bu adımda silme işlemi gerçekleştirilirken güncel listedeki tipi crawler ve kötü amaçlı yazılım olanlar bulunup, T_LOG_DATA tablosundaki CS_USER_AGENT alanı bu filtrelenmiş veri içerisinde olan kayıtlar tablodan silinmiştir. Silinen toplam kayıt sayısı 3.774.452 satırdır. Bu değer, toplam kayıt sayısının %37’ si kadardır.

Bu kayıtları temizlerken karşılaşılan bir sorun şudur: Güncel crawler ve kötü amaçlı yazılım listesindeki bazı kayıtlar T_LOG_DATA tablosundaki CS_USER_AGENT kolonuyla uyuşmaktayken, silme işlemi için yazılan SQL sorgu cümlesinden bu eşleşen kayıtlar dönmemiştir. Bunun sebebi güncel listedeki noktalama işaretleri ile log kayıtlarındaki CS_USER_AGENT alanındaki değerlerin noktalama işaretlerin farklılık göstermesi oldu.

Örneğin :

CS_USER_AGENT alanındaki değer:
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+1.1.4322)

Güncel listedeki değer:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)

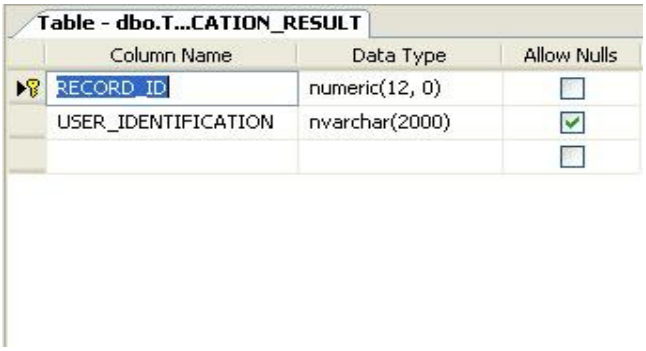
Yukarıdaki örnekte görüldüğü gibi, log kayıtlarındaki CS_USER_AGENT alanında "+" karakteri varken, güncel listede "+" karakteri yerine boşluk karakteri bulunmaktadır. Bu gibi durumlara karşı, SQL sorgu cümlesi de bu ihtiyaca göre yazılmıştır. Yazılan bir örnek SQL cümlesi şöyledir:

DELETE FROM T_LOG_DATA WHERE REPLACE (REPLACE(CS_USER_AGENT,'+',' ') ,' ','') IN (SELECT REPLACE(REPLACE(SPIDER_NAME,'+',' '),' ','') FROM T_SPIDER_LIST)

Bu 3 adımdaki ön işlemler tamamlandıktan sonra T_LOG_DATA tablosunda 985.458 kayıt kalmıştır. Hiçbir işlem yapılmadan önce 10.454.274 kayıt varken, ön işlem aşamasından sonra 985.458 kayıt kalmıştır. Yani toplam log kayıtlarında yaklaşık **%90** oranında bir veri temizleme işlemi gerçekleşmiştir.

5.3 Kullanıcı Tanımlama Aşaması

5.2'deki veri seti üzerinde ön işlemler yapıldıktan sonra geriye kalan 985.458 kayıt üzerinde kullanıcı tanımlama işlemi gerçekleştirilmiştir. Bu işlemi gerçekleştirmek için P_USER_IDENTIFICATION isimli bir SQL prosedürü yazılmıştır. Herbir kullanıcı tanımı T_USER_IDENTIFICATION_RESULT tablosunda bulunmaktadır. Bu tablonun kolonları Şekil 5.4' te görülmektedir.

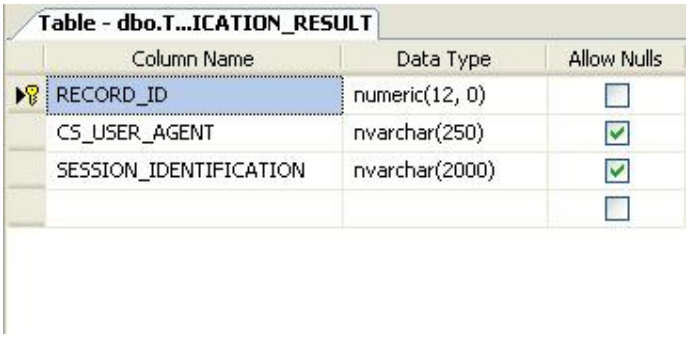


Şekil 5.4 T_USER_IDENTIFICATION_RESULT tablosunun yapısı

P_USER_IDENTIFICATION isimli prosedür şöyle çalışmaktadır: T_LOG_DATA tablosundaki kayıtlar DATE ve TIME kolonuna göre artan şekilde sıralanarak çekilmiştir. Bu kayıtlar üzerinde teker teker ilerleyerek, aynı tarayıcıya sahip kayıtlar gruplanmıştır. Herbir tarayıcı grubu için REFERRER kolonuna bakılmıştır. Eğer bu kolonun değeri boşsa, bu yeni bir kullanıcı demektir ve bu yeni kullanıcının ilk eriştiği sayfa adresi de CS_URI_STEM kolonunun değeridir. Yeni bir kullanıcı tanımı bulunana kadar CS_URI_STEM kolonunun değerinden itibaren erişilen tüm sayfaların bağlantı bilgileri bulunmuştur ve sayfaları belirleyebilmek için, sayfalar arasına "*" işareti eklenerek T_USER_IDENTIFICATION tablosuna bir kayıt olarak yazılmıştır. Eğer REFERRER kolonunun değeri boş değilse ve bu değer kullanıcının eriştiği sayfaların isimleri içinde varsa, CS_URI_STEM kolonunun değeri de kullanıcının eriştiği sayfaların sonuna "*" karakteri ile eklenmiştir ve yeni bir kullanıcı tanımıyla karşılaşılınca oluşturulan sayfa bağlantıları bilgisi T_USER_IDENTIFICATION_RESULT tablosuna bir kayıt olarak atılmıştır. Prosedürün çalıştırılması sonucu T_USER_IDENTIFICATION_RESULT tablosundaki kayıt sayısı 985.367 olarak çıkmıştır. Prosedürün kodları Ek 3' te mevcuttur.

5.4 Oturum Tanımlama Aşaması

Oturum tanımlama aşaması, kullanıcı tanımlama aşamasında saptanan kullanıcıların sitedeki hareketlerini oturumlara bölmek için yapılmaktadır. Oturumları tanımlamak için P_SESSION_IDENTIFICATION isimli bir SQL prosedürü yazılmıştır. Bu prosedürün kodları Ek 4' te mevcuttur. Prosedürün kayıtları yazdığı tablo, T_SESSION_IDENTIFICATION_RESULT tablosudur. Bu tablonun kolonları Şekil 5.5' te görülebilir.



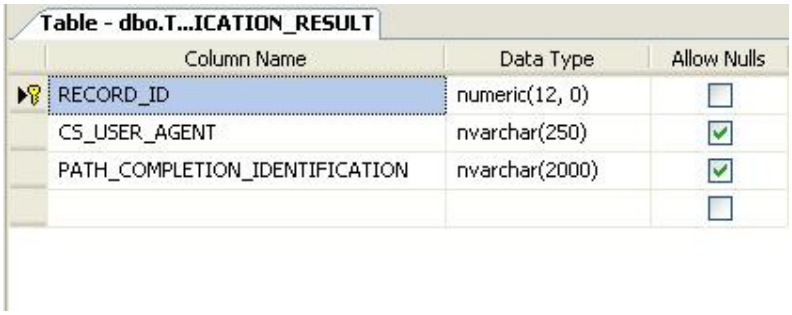
Şekil 5.5 T_SESSION_IDENTIFICATION_RESULT tablosunun yapısı

Oturum tanımlama aşamasının mantığı kullanıcı tanımlamanın mantığıyla aynı olup, tek fark eğer bir kullanıcının ard arda gelen 2 sayfa isteği arasındaki süre 30 dakikayı geçmiş ise, bu yeni bir oturum anlamına gelmektedir. Prosedürde de kullanıcı tanımlama esnasındaki işlemler yapılmış olup, iki sayfa isteği arasındaki 30 dakika limiti kontrol edilerek, eğer limit aşılmışsa T_SESSION_IDENTIFICATION_RESULT tablosuna bir kayıt olarak atılmıştır. Limit aşılmamışsa, kullanıcı tanımlamadaki mantığa göre istekte bulunulan sayfalar "*" işareti ile birbirine eklenerek, yeni bir oturum bulununcaya kadar hafızada tutulmuştur. Yeni bir oturum bilgisi ile karşılaşınca bu hafızadaki bilgi T_SESSION_IDENTIFICATION_RESULT tablosuna yazılmıştır. Prosedür çalıştırıldıktan sonra, T_SESSION_IDENTIFICATION_RESULT tablosundaki kayıt sayısı 980.804 olarak görülmüştür.

5.5 Yol Tamamlama Aşaması

Oturum tamamlama aşaması tamamlandıktan sonra, yol tamamlama aşamasına geçilmiştir. Bu aşama oturum tamamlama aşamasındaki eksik verilerin giderilmesi amaçlanmaktadır. Bu eksik veri şundan dolayı kaynaklanabilmektedir: Kullanıcı sayfalar arasında gezinirken klavyenin BackSpace tuşuna basıp bi önceki sayfaya geri dönerse, bu geri dönme işlemi log dosyasına kayıt olarak atılmamaktadır; ancak bu geri dönme işlemi de web kullanım madenciliği için gerekli bir bilgi olmakla birlikte, oturumların tam olarak tamamlanabilmesi için gereklidir. Bu yüzden bu geri dönme işlemlerinin de kullanıcının oturum bilgisine eklenmesi gerekmektedir.

Yol tamamlama aşaması için P_PATH_COMPLETION prosedürü yazılmıştır. Bu prosedür yol tamamlama bilgilerini bulduktan sonra, bu bilgileri T_PATH_COMPLETION tablosuna yazmaktadır. Bu tablonunu yapısı Şekil 5.6' da görülebilir.



Şekil 5.6 T_PATH_COMPLETION tablosunun yapısı

5.6 Kullanıcı Tanımlama Bilgilerine Apriori Algoritmasının Uygulanması

Birliktelik kurallarının bulunmasında kullanılan en yaygın algoritma, bu tez çalışmasında da kullanılan algoritmalardan birisi olan Apriori algoritmasıdır. Apriori algoritması, 5.3. bölümde anlatılan kullanıcı tanımlama işlemi sonucu oluşan, siteyi ziyaret eden kullanıcıların eriştikleri sayfa bağlantı bilgileri üzerinde uygulanmıştır.

Algoritma için yazılan SQL prosedürü P_APRIORI_ALGORITHM ismindedir. Prosedür şöyle çalışmaktadır: Öncelikle hangi sayfadan hangi sayfaya gidildiğini, IP ve tarayıcı bazında tutan T_FROM_TO_TIME_TABLES tablosundan; FROM_PAGE ve TO_PAGE kolonlarındaki sayfa bağlantı adresleri DISTINCT kullanılarak çekilmiştir. 760.947 satır kayıt olan T_FROM_TO_TIME_TABLES tablosundan bu işlem sonrasında elde edilen ve Apriori algoritmasının uygulandığı kayıt sayısı 47.480 olmuştur. Elde edilen herbir sayfa bağlantı adresinin, T_FROM_TO_TIME_TABLES tablosunda kaç kayıtta olduğu tespit edilmiştir. Daha sonra herbir sayfa için tekrarlanan bu işlem sonucunda elde edilen sayfanın toplam kaç kayıtta olduğu T_APRIORI_HEADER tablosundaki COUNT_ITEM kolonuna yazılmıştır. T_APRIORI_HEADER tablosundaki ITEM_LEVEL kolonuna da ilerlenen herbir adım için 1'den başlayıp teker teker arttırılarak değer yazdırılmıştır. T_APRIORI_HEADER tablosunun yapısı Şekil 5.7' deki gibidir.

Table - dbo.T_APRIORI_HEADER

Column Name	Data Type	Allow Nulls
RECORD_ID	bigint	☐
COUNT_ITEM	numeric(18, 4)	☑
SUPPORT	numeric(18, 4)	☑
LEVEL_NUMBER	tinyint	☑
DELETED	tinyint	☑
		☐

Şekil 2.7 T_APRIORI_HEADER tablosunun yapısı

Daha sonra T_APRIORI_HEADER tablosundaki SUPPORT alanı için hesaplama yapılmıştır. Bu alanın değerini hesaplamak için T_APRIORI_HEADER tablosundaki COUNT_ITEM kolonundaki değer, T_FROM_TO_TIME_TABLES tablosundaki toplam satır sayısına bölünmüştür ve SUPPORT alanına yazılmıştır. Böylece herbir sayfa bağlantı adresinin, tek başına kaç kere tıklandığı ve toplam adaylar arasındaki destek değeri de hesaplanmıştır. T_APRIORI_HEADER tablosunun detay tablosu olan T_APRIORI_DETAIL tablosu da T_APRIORI_HEADER tablosundaki herbir kayıt için kullanıcı tarafından tıklanan sayfa isimlerini tutmaktadır. T_APRIORI_DETAIL tablosunun kullanılmasının sebebi, Apriori algoritmasının ileriki adımlarında, kullanıcının birden fazla tıkladığı sayfalar için Apriori algoritmasının uygulanacağı ve bir tabloda tekrarlanan kayıtları tutmak yerine normalizasyon işleminin gerekli olduğu düşünülerek, tekrar eden COUNT_ITEM, SUPPORT, LEVEL_NUMBER alanlarının header tablosunda; tıklanan sayfa adreslerinin de detay tablosunda tutulmasının hem kontrol hem de hız açısından daha uygun olduğunun düşünülmesidir. T_APRIORI_DETAIL tablosunun yapısı Şekil 5.8' deki gibidir.

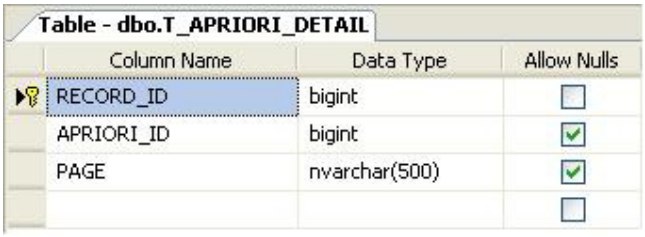


Table - dbo.T_APRIORI_DETAIL

Column Name	Data Type	Allow Nulls
RECORD_ID	bigint	☐
APRIORI_ID	bigint	☑
PAGE	nvarchar(500)	☑
		☐

Şekil 3.8 T_APRIORI_DETAIL tablosunun yapısı

1.adımdan sonra, T_APRIORI_HEADER tablosundaki toplam kayıt sayısı 47.480 iken, %10 olarak kabul edilen destek değerini aşan kayıt sayısı 697 olarak görülmüştür. Bu da, bu sitedeki linklerin video içerikli olduğu için kullanıcıların genelde bir video izledikten sonra siteyi terk ettiklerini göstermektedir.

1 elemanlı sayfa bağlantı adresleri için hesaplamalar bitip destek değerini aşan sayfalar bulunduktan sonra, 2 elemanlı sayfa bağlantı adresleri için prosedür çalışmaya devam etmiştir. 2 elemanlı sayfa bağlantı adres bilgilerini, beraber bulunma sayılarını ve destek değerlerini tutmak için T_APRIORI_HEADER tablosuna kayıtlar aynı şekilde atılmış, sadece LEVEL_NUMBER değeri 2 olarak set edilmiştir.

2. adımdaki işlemler için öncelikle T_APRIORI_HEADER tablosundaki destek eşik değerini aşan sık geçen öğe kümeleri olan, 697 kaydın ikili kombinasyonları bulunup, 697*697=485.809 kayıtlık aday öğe kümesi üzerinde Apriori algoritması uygulanmıştır. T_APRIORI_HEADER tablosundaki minimum destek değerini aşan kayıtlar için, T_APRIORI_HEADER ve T_APRIORI_DETAIL tabloları joinlenerek elde edilen herbir satırda T_APRIORI_DETAIL tablosunda bulunan sayfa bilgilerinden 2.adım için gerekli olan 2 adet sayfa bilgisi elde edilmiştir. Elde edilen kayıtlar üzerinde teker teker ilerlenerek, T_FROM_TO_TIME_TABLES tablosundaki kayıtlar C_IP kolonuna göre gruplandıktan sonra, iki adet sayfa adresinin sırasıyla, FROM_PAGE ya da TO_PAGE kolonunda bulunan sayfa ismiyle eşleşip eşleşmediğine bakılmıştır. Eğer her iki sayfa adresi de eşleşmişse prosedürde tanımlı olan @PAGE_RECORD_COUNT değişkeni 1 arttırılarak bu işlem tüm C_IP değerleri için yapılmış ve iki sayfanın birlikte tıklandığı toplam kayıt sayısı bulunmuştur. Herbir ikili sayfa kombinasyonu için Apriori algoritması çalıştıktan sonra, T_APRIORI_HEADER tablosuna ikili sayfa kombinasyonunun birlikte tıklanma sayısı, birlikte tıklanma sayısının toplam kayıt sayısına bölünmesiyle elde edilen destek değeri ve LEVEL_NUMBER=2 olarak yazılmış; T_APRIORI_DETAIL tablosuna da T_APRIORI_HEADER tablosuna yazılan herbir kayıt için onun detayını oluşturan iki adet sayfa adresi yazılmıştır. 2.adımdan sonra elde edilen ve minimum destek değeri aşan sık geçen öğe kümesi sayısının 483 olduğu görülmüştür.

Bu işlemler 3. ve 4. adımlar için de devam etmiştir. 3.adımdan sonra elde edilen sık geçen öğe kümesi sayısı 271; 4.adımdan sonra elde edilen sık geçen öğe kümesi sayısı ise 192 olmuştur. 4.adımda elde edilen 192 kayıt içinde de kabul edilen eşik güven değeri olan %75 değerini geçen kayıt sayısının 112 olduğu görülmüştür. Yani, Apriori algoritması sonucunda 112 adet birliktelik kuralına erişilmiştir.

İşlem yapılan bir günlük veriler içinde, Apriori algoritması dördüncü adımdan sonra durmuştur. Dördüncü adımdan sonra da çalışan Apriori algoritması sonucunda elde edilen aday öğe kümelerinin hepsinin destek değerleri %10 destek eşik değerinin altında kaldığı için algoritma 5.adımda sonlanmıştır. Kabul edilen minimum destek değeri için bir kullanıcının bu sitede maksimum 4 adet videoyu izlediği sonucuna ulaştırmıştır.

Tez çalışmasında kullanılan portal sitesinde toplam 18 adet kategori bulunmakta ve sitedeki videolar bu kategorilere göre ayrıştırılmıştır. Sitede bulunan kategori listesi şöyledir:

- Amatör
- Animasyon
- Motorlu Araçlar
- Müzik
- Sanat
- Otomobil
- Çocuk
- Reklamlar
- Eğlence
- Spor
- Bilim-Teknik
- Seyahat-Gezi
- Hayvanlar
- Komedi
- İnsanlar
- Sinema-TV
- Haber
- Ünlüler

Apriori algoritması sonucunda çıkan ikili, üçlü ve dörtlü kurallar için 10' ar tane kural Çizelge 5.3, Çizelge 5.4 ve Çizelge 5.5' te görülmektedir. Çizelgelerde her sayfa bağlantı adresi için, adresin kategorisine göre verilmiş kısa isimler bulunmaktadır. Bu kısa isimlerin hangi adrese denk geldiği bilgisi ise Ek 8' de görülebilir. Çizelge 5.5'te de bulunan kurallara bakıldığında, bir kullanıcının izlediği videolar genelde aynı kategoride yer almakla beraber; eğlence ve komedi videolarının da birlikte bir kullanıcının izlediği videolar arasında bulunabileceği gözlemlenmiştir.

Ortaya çıkan sonuçlardan bir tanesi de, video izleyen kullanıcıların izledikleri video içeriklerinin en çok eğlence kategorisinde yer alan 18 yaş üzeri izlenebilen videoları içermesi olmuştur. Bu kategoriyi sırasıyla komedi, sinema-televizyon ve motorlu araçlar kategorisindeki videolar takip etmektedir.

Çizelge 5.3 İkili birliktelik kuralları (10 tane)

Kullanıcı Tanım Bilgisi	Destek Değeri
ANASAYFA → MA1	40,3
ANASAYFA → ST1	39,2
ANASAYFA → K1	35,3
ANASAYFA → E1	30,7
ANASAYFA → H1	28,6
ANASAYFA → K2	27,1
ANASAYFA → E2	25,9
ANASAYFA → M1	25,6
ANASAYFA → ST2	23,8
ANASAYFA → E3	19,2

Çizelge 5.4 Üçlü birliktelik kuralları (10 tane)

Kullanıcı Tanım Bilgisi	Destek Değeri
(ANASAYFA , MA1) → ST1	39,8
(ANASAYFA , MA1) → M1	38,5
(ANASAYFA , H1) → M1	37,1
(ANASAYFA , SEARCH10) → K1	31,4

(ANASAYFA , SG1) → E1	28,3
(ANASAYFA , ST1) → K2	27,1
(ANASAYFA , E2) → E4	25,6
(ANASAYFA , E5) → M1	24,1
(ANASAYFA , ST2) → E3	22,8
(ANASAYFA , E3) → K1	17,9

Çizelge 5.5 Dörtlü birliktelik kuralları (10 tane)

Kullanıcı Tanım Bilgisi	**Destek Değeri**	**Güven Değeri**
(ANASAYFA , SEARCH1 , H1) → M1	33,1	92,3
(ANASAYFA , SEARCH2 , ST3) → ST1	28,7	89,4
(ANASAYFA , SEARCH3, E6) → K3	24,3	85,1
(ANASAYFA , ST1 , ST4) → ST5	21,8	83,3
(ANASAYFA , SEARCH4, E5) → E3	20,2	76
(ANASAYFA , SEARCH5 , MA1) → MA2	19,4	79,2
(ANASAYFA , SEARCH7, E7) → E8	17,7	79,4
(ANASAYFA , SEARCH7, MA3) → MA4	16,1	80,2
(ANASAYFA , SEARCH8, K4) → K5	14,3	78,4
(ANASAYFA , SEARCH9, K6) → K7	12,8	75,2

Apriori algoritmasının farklı destek ve güven değerleri için çalışma süresi ve sonuçta çıkan birliktelik kuralları sayısı Çizelge 5.6' da verilmiştir. Çizelge 5.6' da görüldüğü gibi, aynı güven değeri için destek değeri küçüldükçe algoritmanın çalışma süresi uzamakta ve ortaya çıkan birliktelik kuralı sayısı da artmaktadır. Aynı güven değeri için destek değeri büyüdükçe algoritmanın çalışma süresi kısalmakta ve birliktelik kuralı sayısı da azalmaktadır. Diğer taraftan, aynı destek değeri için güven değeri arttıkça çalışma süresi uzamamakla birlikte ortaya çıkan birliktelik kuralı sayısı azalmaktadır. Aynı destek değeri için güven değeri azaldıkça da yine çalışma süresi değişmemekle beraber ortaya çıkan birlikteli kuralı sayısı artmaktadır.

Çizelge 5.6 Apriori Algoritmasının Farklı Destek ve Güven Değerleri İçin Ürettiği Sonuçlar

DESTEK DEĞERİ	GÜVEN DEĞERİ	ÇALIŞMA SÜRESİ	BİRLİKTELİK KURALI SAYISI
%15	%75	9 sa. 10 dak.	53
%10	%75	10 sa. 47 dak.	112
%7	%75	12 sa. 19 dak.	263
%10	%90	10 sa. 47 dak.	62
%10	%60	10 sa. 47 dak.	185

Apriori Algoritması İçin Yapılan Optimizasyon Çalışması

Apriori algoritması birliktelik kurallarının çıkarılmasında kullanılan en yaygın algoritma olmasıyla birlikte, büyük veri kümeleri uygulandığında yavaş çalışan bir algoritma olduğu bilinmektedir. Apriori algoritması, herbir adımda belirlenecek olan yaygın nesne kümeleri için veritabanını tarar ve yaygın nesne kümelerini belirler. Veritabanı boyutu yüksek olduğunda ise bu işlem çok uzun sürebilmektedir. Bu tez çalışmasında Apriori algoritmasının daha hızlı çalışması için optimizasyon çalışması yapılmıştır. Bu optimizasyon çalışmasında yapılan işlemler şöyledir:

- Apriori algoritmasının asıl veri kaynağı olan ve sitede bulunan kullanıcıların tıkladığı sayfa bağlantı adreslerini tutan T_USER_IDENTIFICATION_RESULT tablosu, algoritmada direk kullanılmak yerine, bu tablo P_TIME_FREQUENCY_SESSION_IDENTIFICATION prosedürü kullanılarak T_FROM_TO_TIME_TABLES tablosuna dönüştürülmüştür. Prosedür, Apriori algoritması için geliştirilen prosedür çalıştırılmadan önce bir kez çalıştırılmıştır ve T_FROM_TO_TIME_TABLES tablosunu doldurmuştur. Prosedür T_USER_IDENTIFICATION_RESULT tablosundan herbir kullanıcının tıkladığı bağlantı adreslerini DISTINCT sorgu çekerek T_FROM_TO_TIME_TABLES tablosuna yazmış, böylece tekrarlayan kayıtlar üzerinde gereksiz işlem yapmak için harcanan zamanın büyük bir oranda azalmasını sağlamıştır.
- T_USER_IDENTIFICATION_RESULT tablosundaki kayıt sayısı 985.367 ve veritabanında bu tablonun kapladığı disk boyutu 308.273 MB iken; T_FROM_TO_TIME_TABLES tablosundaki kayıt sayısı nın760.947 ve disk boyutunun da 274.063 MB olduğu görülmüştür.
- T_FROM_TO_TIME_TABLES tablosuna kayıtlar, bu tabloda bulunan ve kullanıcıyı tanımlamaya yarayan C_IP kolonuna göre ayrıştırılarak yazılmıştır. Apriori algoritması için geliştirilen prosedür de bu tablodan verileri okurken C_IP kolonuna göre gruplayarak okumaktadır. Apriori algoritması prosedürü çalışırken, aday öğelerin C_IP' ye gruplandırılarak çekilen kayıtlar içinde var olup olmadığına bakarken eğer aranan aday öğe kullanıcının tıkladığı sayfalarda bulunmuyorsa, işlem o C_IP için durdurulmuş ve diğer C_IP kümesine geçmiştir. Böylece, gereksiz yere prosürün çalışması engellenmiştir.
- Apriori algoritmasının herbir adımında elde edilen değerleri tutmak için kullanılan T_APRIORI_HEADER ve T_APRIORI_DETAIL tabloları arasında gerekli ilişkiler veritabanı düzeyinde tanımlanmıştır ve bu tablolarda yaratılan index ler ile de bu tablolardan veri çekildiği zamanki sürenin daha kısa olması sağlanmıştır. Ayrıca herbir adımda elde edilen sonuçları belirlemek için T_APRIORI_HEADER tablosuna LEVEL_NUMBER kolonu eklenerek, Apriori prosedürünün n. adımı için T_APRIORI_HEADER ve T_APRIORI_DETAIL tabloları joinlenerek ve T_APRIORI_HEADER tablosundan sadece LEVEL_NUMBER=n-1 olan kayıtlar çekilerek kayıtlara ulaşılmıştır.

5.7 Kullanıcı Tanımlama Bilgilerine TFPR Algoritmasının Uygulanması

TFPR algoritması (Access **T**ime-Length and **F**requency-Based **P**age**R**ank), Google sitesinin kullandığı PageRank algoritmasının temeline, web sayfalarında bulunulma süresi ve web sayfalarına erişim sıklığı bilgilerini ekleyerek, ziyaret edilme olasılığı en yüksek sayfaları bulmak amaçlı geliştirilen bir algoritmadır. Bu tez çalışmasında da, kullanıcı tanımlama bilgileri üzerinde TFPR algoritması uygulanarak, log kayıtları kullanılan web sitesinde en çok ziyaret edilen ve ileride ziyaret edilme olasılığı en yüksek sayfalar bulunmuştur.

PageRank algoritması, Google sitesinin site sıralamasında kullandığı algoritmanın sonucu web sitelerine verdiği değeri gösteren, basitleştirilmiş 0'dan 10'a kadar olan bir değerdir. Bu değer genel olarak özgün bir içeriğe, sayfaya verilmiş bağlantılara ve bağlantı veren sayfaların kalitesine bağlı olarak değişir. Eğer bir A sitesi B sitesinin linkini yayınlamışsa bunun nedeni B sayfasının A sayfası ziyaretçileri tarafından dolaşılabilecek olarak düşünülmüş olmasıdır. Bu yapıya göre A sayfası B sayfasının PageRank değerini yükseltmiş olacaktır.

Google sitesinin kurucularına göre, dünyadaki tüm sitelerin PageRank değeri toplamı 1 sayısına eşittir. PageRrank değerinin Google kurucuları tarafından açıklanan resmi formülü (5.1) 'deki gibidir.

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn)) \qquad (5.1)$$

Bu formülde, "PR(A)" değişkeni A sitesine ait PageRank değeridir. Bu değer tüm siteler için ilk başta 1 kabul edilmektedir. "d" değeri "damped down" faktörü denilen özel bir katsayıdır ve Google tarafından değeri 0.85 olarak kabul edilmektedir. "PR(Tn)" değişkeni, A sitesine link veren herhangi bir sitenin PageRank değeridir. "C(Tn)" ise, A sitesine link veren sitenin başka sitelere verdiği link sayı adetidir (Guo, Y., Ramamohanarao, K., Park , L., 2007).

Bu tez çalışmasında uygulanan TFPR algoritması ise, Google sitesinin kullandığı PageRank algoritmasını baz alarak, web sayfasında kullanıcıların bulunma süreleri ve web sayfasına erişim sıklığı parametrelerini kullanarak PageRank değerine benzer TFPR değerini hesaplamaktadır. Bu değerin yüksek olması, işlem yapılan web sayfasının ziyaret edilme olasılığı yüksek olması anlamına gelmektedir.

TFPR algoritması ön işlem basamaklarından geçmiş ve kullanıcı oturum bilgileri elde edilmiş web log kayıtları üzerine uygulanmaktadır. Bir web sitesi G isminde bir graf olarak düşünülürse, düğümler web sitesindeki sayfalar ve grafın kenarları da web sayfaları arasındaki bağlar olarak kabul edilebilir. Herbir düğüm ve kenarın bir ağırlığı vardır. Bir "u" düğümündeki t_u ağırlığı u sayfasını görüntüleyen tüm kullanıcıların bu sayfada bulunduğu toplam süredir. $t_{(v,u)}$ ağırlığı ise, v ve u sayfalarının arka arkaya görüntülendiği durumlarda u sayfasında harcanan toplam süredir. Eğer daha fazla kullanıcı v ve u sayfalarını sırasıyla takip ederse ve u sayfasında daha uzun süre bulunurlarsa, $t_{(v,u)}$ değeri daha da büyük olacaktır. Böylece $t_{(v,u)}$ değeri hem erişim zaman uzunluğu hem de u sayfasındaki erişim sıklığı bilgisine göre değişmektedir (Guo, Y., Ramamohanarao, K., Park , L., 2007).

B_u, u sayfasına link veren tüm sayfalar ise t_u değeri (5.2)'deki gibi hesaplanır.

$$t_u = \sum_{w \in B_u} t_{(w,u)} \tag{5.2}$$

Google sitesindeki PageRank değeri hesaplamasında kullanılan sabit değer olan damped down değeri TFPR algoritmasında "a" tanımı ile verilmiştir. F_v kümesi de, v sayfasının link verdiği tüm sayfalar kümesi olarak tanımlanmıştır. PageRank değerinin tanımına da uygun olarak, erişim süresi uzunluğu ve frekans tabanlı PageRank algoritmasının yani TFPR algoritmasının genel denklemi (5.3)' teki gibidir (Guo, Y., Ramamohanarao, K., Park , L., 2007).

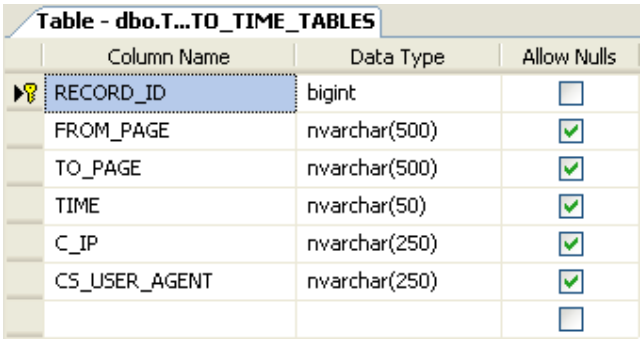
$$\mathrm{TFPR}_{i+1}(u) = \alpha \times \frac{t_u}{\sum_{v \in G} t_v} + (1-\alpha) \times \sum_{v \in B_u} \frac{TFPR_i(v) \times t_{(v,u)}}{\sum_{w \in F_v} t_{(v,w)}} \tag{5.3}$$

(5.3) denkleminde görüldüğü gibi TFPR algoritması rekürsif bir algoritmadır. Bir sayfanın TFPR değeri hesaplanırken, o sayfaya link veren her sayfa için, link veren sayfanın link verdiği diğer sayfalardaki bulunulma süreleri de hesaplanarak bir değere ulaşılmaktadır.

TFPR algoritması ile gerekli bilgiler edinildikten sonra tez çalışmasında kodlama kısmına geçilmiştir. TFPR algoritması, ön işlem basamaklarından geçen kullanıcı oturum bilgileri üzerine uygulanarak hesaplamalar yapılmıştır.

İlk olarak ön işlem basamaklarından geçmiş log kayıtlarının tutulduğu T_LOG_DATA tablosundaki kayıtlar üzerinde P_TIME_FREQUENCY_SESSION_IDENTIFICATION prosedürü uygulanmıştır. Bu prosedür, T_LOG_DATA tablosu üzerindeki kayıtları T_FROM_TO_TIME_TABLES tablosuna atarken, T_FROM_TO_TIME_TABLES tablosundaki TIME kolonuna da T_LOG_DATA tablosundaki tıklanan sayfa ile ondan bir sonra tıklanan sayfa arasında geçen süreyi hesaplayarak yazmaktadır. Prosedürün kodları Ek 7'de mevcuttur.

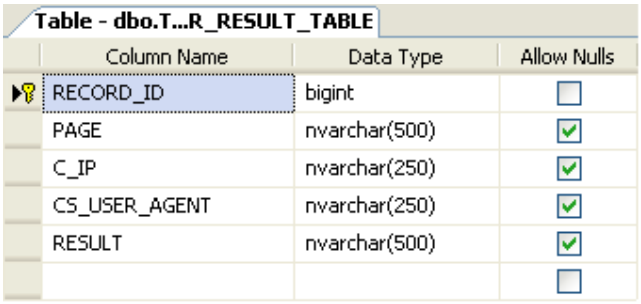
T_FROM_TO_TIME_TABLES tablosunun yapısı Şekil 5.9' daki gibidir.



Şekil 5.9 T_FROM_TO_TIME_TABLES tablosunun yapısı

P_TFPR_ALGORITHM algoritması web sitesindeki herbir sayfa için çağırılarak, o sayfanın TFPR değerini hesaplamaktadır. Prosedür öncelikle kendisine parametre olarak gelen web sayfası için, bu sayfaya link veren sitedeki diğer sayfalar için, bu sayfalara link veren diğer sayfaları bularak bu sayfalar için de P_TFPR_ALGORITHM algoritmasını çağırarak rekürsif işlem yapmaktadır. Bu işlem, TFPR değeri hesaplanacak sayfaya link veren tüm sayfalar için rekürsif hesaplamalar yapıldıktan sonra sonlanmaktadır. Elde edilen TFPR değeri, TFPR değeri hesaplanan sayfanın adı ile birlikte T_TFPR_RESULT_TABLE tablosuna yazılarak, tüm sayfalar için sonuçlar bir tabloda tutulmuştur. P_TFPR_ALGORITHM prosedürünün kodlaması Ek 6' da mevcuttur.

T_TFPR_RESULT_TABLE tablosunun yapısı Şekil 5.10' daki gibidir.



Şekil 5.10 T_TFPR_RESULT_TABLE tablosunun yapısı

P_TFPR_ALGORITHM algoritmasının uygulanmasıyla elde edilen sonuçların, TFPR değeri en yüksekten düşüğe doğru sıralanmış olarak ilk 60 kaydı Çizelge 5.7' de görülmektedir. TFPR değeri yüksek çıkan sayfalar, web sitesinde kullanıcılar tarafından görüntülenme olasılığı en yüksek olan sayfalar olarak kabul edilir.

Çizelge 5.7 P_TFPR_ALGORITHM Algoritmasının Sonuçları

WEB SAYFASI ADI	**TFPR DEĞERİ**
http://webmining.com/	0.00467
http://webmining.com/ep/FirstGuncellemeMode.aspx?ViewIndex=8&username=adnan_akalin	0.00399
http://video.webmining.com/tumaycakir/kadinlarda_dagitti/264590/12c4d9695cad0a8dc7c4449bfe2b9e21	0.00156
http://video.webmining.com/sedar_08/guzellerde_kayar/270258/8abfadbd3c4c41fd19b206a29aba42f0	0.00104
http://video.webmining.com/	0.00077
http://webmining.com/ep/eksenhome.aspx	0.00056
http://webmining.com/ep/Default.aspx	0.00050
http://webmining.com/ep/UserFriendPage.aspx	0.00046
http://video.webmining.com/architect.tunc/Steve_Martin_Show/271117/	0.00038
http://video.webmining.com/korkusuz067108/ibo_show_hadise_dum_tek_tek/272348/	0.00037
http://video.webmining.com/r.emiryigit/Davos/272731/233f13dd0078270832e252337a8626da	0.00033
http://webmining.com/ceyda_naz08	0.00029
http://video.webmining.com/tarik1982/kizlar_ve_erkekler_arasindaki_fark/268170/	0.00028
http://video.webmining.com/ozer_1151/Felegin_Carki/272656/	0.00025
http://video.webmining.com/teknisyenarif/planda_hesaba_katilmayan_s	0.00024

ey_neydi_NISSAN/271463/	
http://webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=sexy	0.00023
http://webmining.com/ep/ExenHomePage.aspx?mt=2	0.00023
http://webmining.com/ep/ExenHomePage.aspx?mt=3	0.00019
http://webmining.com/06nur	0.00019
http://webmining.com/ep/ExenHomePage.aspx?mt=1	0.00018
http://video.webmining.com/zeyna_34_08/ZEYNA/264030/	0.00016
http://webmining.com/tuba1986_26	0.00014
http://www.mynet.com/2008/11/5/59901	0.00013
http://video.webmining.com/wigproductions/Arap_Kizlari/41686/	0.00013
http://webmining.com/basak_toksoy	0.00012
http://video.webmining.com/yuksek_topuklu_kiz/NURGUL_YESILCAY_50_YKR/264442/	0.00012
http://webmining.com/ezgi_gull	0.00012
http://video.webmining.com/adaca1907/ozan_arif/217638/	0.00011
http://video.webmining.com/spedex/muhtesem_otesi_bir_klip_ve_sarki/1914/	0.00011
http://webmining.com/53shekill53	0.00010
http://video.webmining.com/hayalet_erge/Ozan_Arif_4_NISAN/215140/	0.00009
http://webmining.com/ep/GrupTakilmaMode.aspx?McId=3000009917	0.00009
http://webmining.com/adventure74	0.00009
http://video.webmining.com/teknisyenarif/essekten_yol_yardimi/272631/	0.00009
http://webmining.com/ep/EngelliProfile.aspx?u=0_toprak	0.00009
http://video.webmining.com/wanted25251980/Mehmet_Ali_den_sakincali_hareketler/271715/	0.00009
http://video.webmining.com/usecret/Ruzgarin_Vahsi_Etkisi/1722/	0.00009
http://video.webmining.com/cafeses/Recep_ivedik2/250964/	0.00009
http://webmining.com/beni_sevmiyom	0.00008
http://video.webmining.com/burakanlayis/Ozan_Arif_Kormusun_Siiri/241675/	0.00008
http://video.webmining.com/can46100/Helin_d_n_arem/251647/4a2c463962f4b05711dd827d6cc1e7f6	0.00008
http://video.webmining.com/ruyaalemi34/aydemir_akbassss/20692/	0.00007
http://video.webmining.com/pointless_darkness/paris_hilton/393/	0.00007
http://webmining.com/kalbim_sin35	0.00006
http://webmining.com/ep/GroupHomePage.aspx?mt=1&st=0	0.00006
http://webmining.com/kara_elmasim_35	0.00006
http://video.webmining.com/ahmetim1979/ilyas_salman_ve_senersen_iklisinn_mutesem_kapisma/272686/	0.00006
http://webmining.com/equal_free	0.00006
http://webmining.com/1559mustafamert	0.00006
http://video.webmining.com/tolga_seyhan2008/TURK_HALK_MUZIGI/224481/	0.00005
http://video.webmining.com/acheronx/sarhos_kizlar/9757/	0.00005
http://video.webmining.com/slayer890/agla_yarali_kalbim/1419/	0.00005
http://video.webmining.com/tolga_aslan23/sansli_adam/269158/7045132	0.00005

5d44725760294800f92e7d94e	
http://video.webmining.com/naqoyqatsi/Fedde_Le_Grand_Put_Your_Hands_Up_darMEth/206/	0.00005
http://video.webmining.com/mortalci_21/Kamuran_Akkor_Bir_Atese_Attin_Beni/40999/	0.00005
http://webmining.com/ep/UserFriendPage.aspx?username=1559mustafamert	0.00005
http://video.webmining.com/acheronx/Is_kazasi/175/	0.00004

5.8 Diğer İstatistiki Sonuçlar

Apriori ve TFPR algoritmalarının uygulanmasıyla elde edilen sonuçlar ve kurallar dışında çeşitli istatistiki bilgilere de ulaşılmıştır. Bunun için Visual Studio.NET 2005 ortamında istenen istatistik bilgiler için ekranlar tasarlanmıştır. Bu ekranlar çalıştırıldığında oluşan sonuçlar aşadaki şekillerde gösterilmektedir.

Şekil 5.11' daki ekranda, en çok istekte bulunulan sayfalar görüntülenme sayısına göre azalan şeklinde görülmektedir.

SAYFA BAĞLANTI ADRESİ	GÖRÜNTÜLENME SAYISI
http://video.webmining.com/tumaycakir/kadinlarda_dagitti/264590/12c4d9695cad0a8dc7c4449bfe2b9e21	11917
http://video.webmining.com/usecret/Ruzgarin_Vahsi_Etkisi/1722/	4534
http://video.webmining.com/sedar_08/guzellerde_kayar/270258/8abfadbd3c4c41fd19b206a29aba42f0	4255
http://video.webmining.com/pointless_darkness/paris_hilton/393/	4166
http://video.webmining.com/53shekill53/ferrari_lamgorghini/270244/fe7a8f956161a4ce6fa0dbf6f72f8d25	4114
http://video.webmining.com/naqoyqatsi/Aaliyah_Try_Again_Warrier/334/	3478
http://video.webmining.com/volkan_nasli/aldatma/266945/83138714f70f29a6cdbe94824aab27d5	3233
http://video.webmining.com/gultekin.serkan/talihsiz_kizlar/267130/3c71e26f207360db56fcf8ac561506be	3064
http://video.webmining.com/alfadizaynbilisim/Beyaz_Sahan_Atismasi/270560/f075615d42d43d4db4fd928c91c3e983	3057
http://video.webmining.com/ekipcalismasi/Benim_icin_Davos_bitmistir/272654/8e964e97858e65203e118901354bca5f	2980

1 2 3 4 5 6 7 8 9 10

Şekil 5.11 En çok istekte bulunulan sayfalar (görüntüleme sayısı sıralı)

Şekil 5.11' deki istatistiki sonuçlar dışında ayrıca web kullanım madenciliği esnasında yapılan işlemlerin sonuçları için de ekranlar tasarlanmıştır. Bu ekranların görüntüleri ise aşağıdaki şekillerdeki gibidir.

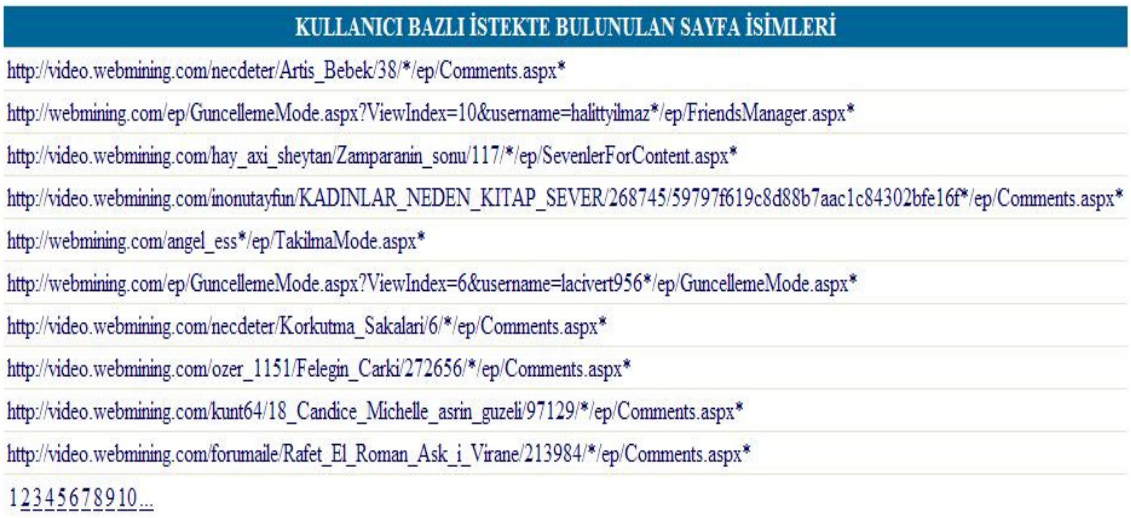
Şekil 5.12' de portal sitesindeki loglardan, veritabanına aktarılıp ön işlem aşamalarından geçenler gösterilmiştir.

GÜN.AY.YIL	ZAMAN	İSTEKTE BULUNULAN SAYFA	HANGİ SAYFADAN GELİNDİĞİ BİLGİSİ	
01.02.2009 00:00:00	05:59:21	/ep/Comments.aspx	http://video.webmining.com/aysosan/dansci_penguen/8177/	Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+tr;+rv:1.
01.02.2009 00:00:00	05:59:20	/ep/Default.aspx	-	Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1
01.02.2009 00:00:00	05:59:20	/ep/UserFavoritesPage.aspx	-	Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.go
01.02.2009 00:00:00	05:59:20	/~Ajax/menuHandler.ashx	-	-
01.02.2009 00:00:00	05:59:20	/~Ajax/menuHandler.ashx	-	Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+tr-TR;+r
01.02.2009 00:00:00	05:59:20	/ep/TakilmaMode.aspx	-	Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.go
01.02.2009 00:00:00	05:59:20	/ep/SevenlerForContent.aspx	http://video.webmining.com/acheronx/sarhos_kizlar/9757/	Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+5.1
01.02.2009 00:00:00	05:59:20	/ep/Comments.aspx	http://video.webmining.com/orhanmuhsuroglu/YENI_JURRASIC_PARK/424/	Mozilla/4.0+ (compatible;+MSIE+7.0;+Windows+NT+6.0;+Trident/4.(
01.02.2009 00:00:00	05:59:20	/__utm.gif	http://webmining.com/ep/HaberlesmeMode.aspx?ViewIndex=9	Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0
01.02.2009 00:00:00	05:59:20	/ep/SearchResults.aspx	-	Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.go

... 21 22 23 24 25 26 27 28 29 30 ...

Şekil 5.12 Kullanıcı log dosyalarındaki kayıtlar

Şekil 5.13' te kullanıcı tanımlama aşaması tamamlandıktan sonra ortaya çıkan kullanıcıların istekte bulundukları sayfaların adresleri görülmektedir.



KULLANICI BAZLI İSTEKTE BULUNULAN SAYFA İSİMLERİ
http://video.webmining.com/necdeter/Artis_Bebek/38/*/ep/Comments.aspx*
http://webmining.com/ep/GuncellemeMode.aspx?ViewIndex=10&username=halittyilmaz*/ep/FriendsManager.aspx*
http://video.webmining.com/hay_axi_sheytan/Zamparanin_sonu/117/*/ep/SevenlerForContent.aspx*
http://video.webmining.com/inonutayfun/KADINLAR_NEDEN_KITAP_SEVER/268745/59797f619c8d88b7aac1c84302bfe16f*/ep/Comments.aspx*
http://webmining.com/angel_ess*/ep/TakilmaMode.aspx*
http://webmining.com/ep/GuncellemeMode.aspx?ViewIndex=6&username=lacivert956*/ep/GuncellemeMode.aspx*
http://video.webmining.com/necdeter/Korkutma_Sakalari/6/*/ep/Comments.aspx*
http://video.webmining.com/ozer_1151/Felegin_Carki/272656/*/ep/Comments.aspx*
http://video.webmining.com/kunt64/18_Candice_Michelle_asrin_guzeli/97129/*/ep/Comments.aspx*
http://video.webmining.com/forumaile/Rafet_El_Roman_Ask_i_Virane/213984/*/ep/Comments.aspx*

1 2 3 4 5 6 7 8 9 10 ...

Şekil 5.13 Kullanıcı tanımlama işlemi sonrasında, kullanıcının istekte bulunduğu sayfalar

Şekil 5.14' te oturum tanımlama aşamasından sonra, ortaya çıkan oturumlara göre istekte bulunulan sayfaların bilgileri görülmektedir.

OTURUM BAZLI İSTEKTE BULUNULAN SAYFA İSİMLERİ
http://video.webmining.com/bozbey008/kolbastiiiiii/229576/*/ep/SevenlerForContent.aspx*
http://video.webmining.com/hay_axi_sheytan/Canli_yayinda_sunucuya_okkali_tokat/56649/*/ep/SevenlerForContent.aspx*
http://video.webmining.com/hakankara53/Katie_Price_Sexy_Video/90639/*/ep/Comments.aspx*
http://video.webmining.com/naqoyqatsi/Aaliyah_Try_Again_Warrier/334/*/ep/SevenlerForContent.aspx*
http://video.webmining.com/bozbey008/kolbastiiiiii/229576/*/ep/Comments.aspx*
http://foto.webmining.com/kat/yeniler/3*/ep/HaberlesmeMode.aspx*
http://video.webmining.com/53shekill53/ferrari_lamgorghini/270244/fe7a8f956161a4ce6fa0dbf6f72f8d25*/ep/SevenlerForContent.aspx*
http://video.webmining.com/pointless_darkness/paris_hilton/393/*/ep/SevenlerForContent.aspx*
http://video.webmining.com/hay_axi_sheytan/Canli_yayinda_sunucuya_okkali_tokat/56649/*/ep/Comments.aspx*

1 2 3 4 5 6 7 8 9 10 ...

Şekil 5.14 Oturum tanımlama işlemi sonrasında oturumdaki istekte bulunulan sayfalar

Şekil 5.15' te http://www.user-agents.org/ sitesinden güncel olarak alınan tarayıcı/crawler listesindeki tanımlar
görülmektedir.

TARAYICI/CRAWLER İSMİ	TARAYICI/CRAWLER TİPİ
Mozilla/4.0 (compatible; ibisBrowser)	Tarayıcı
Mozilla/4.0 (compatible; Lotus-Notes/5.0; Windows-NT)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; AOL 4.0; Windows 98)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Mac_PowerPC)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; MSIECrawler; Windows 95)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Windows 95)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; MSN Companion 2.0; 800x600; Compaq)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Windows CE; PPS; 240x320)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Windows NT Windows CE)	Tarayıcı
Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)	Tarayıcı

... 31 32 33 34 35 36 37 38 39 40 ...

Şekil 5.15 Güncel tarayıcı/crawler listesi

Şekil 5.16' daki ekranda görüldüğü gibi, dışarıdan log dosyası okunup veritabanına eklenebilmektedir.

Dosya Yükle

GÜN.AY.YIL	ZAMAN	İSTEKTE BULUNULAN SAYFA	HANGİ SAYFADAN GELİNDİĞİ BİLGİSİ	TARAYICI/CRAWLER ADI	HTTP STATUS KODU	KULLANICI IP NUMARASI	FORM METODU	SUNUCUNUN ALDIĞI BYTE SAYISI
01.02.2009 00:00:00	05:59:59	/ep/UserFriendPage.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	329
01.02.2009 00:00:00	05:59:58	/ep/SevilenForVideo.aspx	-	-	200	77.79.72.2	GET	119
01.02.2009 00:00:00	05:59:58	/ep/SearchResults.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	312
01.02.2009 00:00:00	05:59:58	/ep/UserFriendPage.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	324
01.02.2009 00:00:00	05:59:58	/ep/UserFriendPage.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	322
01.02.2009 00:00:00	05:59:58	/ep/SearchResults.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	309
01.02.2009 00:00:00	05:59:58	/ep/SearchResults.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	364
01.02.2009 00:00:00	05:59:58	/ep/SevilenForVideo.aspx	-	-	200	77.79.72.2	GET	119
01.02.2009 00:00:00	05:59:57	/ep/UserBlogPage.aspx	-	Mozilla/5.0+ (compatible;+Googlebot/2.1;++http://www.google.com/bot.html)	200	66.249.71.142	GET	321

Şekil 5.16 Log kayıt dosyası okuyup veritabanına ekleme

6. SONUÇLAR

Internet kullanıcılarını anlamanın en iyi yolu onları tanımaktır. Tanıyabilmek için de, kullanıcıların ziyaret ettiği web sitesinden gezinirken yaptığı her hareketin, tıkladıkları her bağlantının kayıt altında tutulduğu kullanıcı log dosyalarının analiz edilmesidir. Bu dosyaların analiz işlemi için bu çalışmanın temel konusu olan web kullanım madenciliği kullanılırken, web kullanım madenciliğinin uygulanması esnasında da veri madenciliği yöntemlerinden faydalanılmaktadır.

Bu tez çalışması kapsamında internet kullanıcılarının davranışlarını çözümleyerek, onlara daha iyi hizmet verebilmek ve web sitesi sahiplerine ve tasarımcılarına da kullanıcı log dosyalarındaki veriye web kullanım madenciliği uygulandıktan sonra çıkan kurallar ve sonuçlara göre web sitesinde değişiklik yaparak daha çok kullanıcı ve daha kaliteli site içeriği sağlayabileceklerini göstererek, ışık tutmak amaçlanmıştır. Bu amaç için, veri madenciliği yöntemlerinden en uygun yöntem olan birliktelik kuralları yöntemi kullanılmış, bu yöntemin en önemli algoritması olan Apriori algoritması da kullanıcı log verisi üzerine uygulanarak çıkan sonuçlar paylaşılmıştır.

Apriori algoritması birliktelik kurallarının uygulanması için kullanılan en yaygın algoritma olmakla birlikte, geniş nesne kümesinden aday nesne kümelerini oluşturma esnasında destek eşik değerini aşan öğeleri bulmak için veritabanı her defasında taranmaktadır. Bu da birliktelik kuralı bulunacak veri kümesindeki öğe sayısı çok büyükse, veritabanı işlemleri uzun süreceğinden zaman karmaşıklığını arttırmaktadır. Bu çalışmada da karşılaşılan en büyük sorun, zamanla ilgili olmuştur. Kayıt sayısı çok fazla olduğu için, ilgili tablolara indeks eklenmesine rağmen apriori algoritmasının çalışması esnasında çok uzun süren sorgular olduğu tespit edilmiştir.

Birliktelik kurallarının uygulanmasında en önemli işlemlerden birisi de, destek değerinin de çok iyi bir biçimde belirlenmesidir. Büyük bir destek değeri çıkabilecek birliktelik kurallarının oluşmamasına sebep olabilirken, küçük bir destek değeri de gereksiz yere kural oluşturmayacak öğeler üzerinde tekrar tekrar işlem yapmayı gerektirebilmektedir. Bu da algoritmanın sonuç verme zamanını geciktirebilir.

Apriori algoritmasının daha hızlı çalışması ve bununla birlikte, veritabanı tarama sayısının azaltılması için veri setinin parçalara ayrılarak tarama yapılması, algoritmanın çalışma süresini bir nebze olsun kısaltabilir. Yaygın olmayan öğe kümeleri belirlendikten sonra bu

öğelerin veritabanında bulunmasının da bir gereği yoktur. Bu kayıtlar silinerek hem veri setindeki öğe sayısı azalacak, hem de böylece algoritmanın çalışma süresi kısalacaktır. Ayrıca uzmanlar veritabanı tarama sayısını azaltmak için Hash yönteminin kullanılmasını önermektedir.

Bu tez çalışmasında kullanılan diğer bir algoritma olan TFPR algoritması da Apriori algoritması gibi kullanıcıların web sitesindeki davranışlarını anlamada yardımcı olur. Bu algoritmayı daha verimli bir şekilde kullanabilmek için web log datası, kullanıcının hangi sayfadan gelip hangi sayfaya gittiği bilgisi ve bu iki sayfa arasında geçirdiği sürenin tutulduğu bir ara tabloya kullanıcı IP bazında ayrıştırarak kaydedilmiştir. Bu ara tablodan algoritmayı uygulamak daha hızlı olmakla birlikte daha kesin sonuçlara varılmasını sağlamıştır. Web log datası optimize edilmeden kullanılarak TFPR algoritmasının çalışma süresi yaklaşık 53 saat sürmüşken, oluşturulan ara tablo kullanılarak TFPR alogritmasının çalışma süresi 7 saat 38 dakika sürmüştür. Bu da zaman açısından yaklaşık 8 kat verim sağlamıştır.

KAYNAKLAR

Agrawal, R. ve Shafer, J.C., (1996), "Parallel Mining of Association Rules: Design, Implementation and Experience", IBM Research Report RJ 10004.

Agrawal, R., Imielinski, T. ve Swami, A., (1993), "Mining Association Rules Between Sets of Items in Large Databases", In ACM SIGMOD Conf. Management of Data.

Agrawal, R. ve Srikant, R., (1994), "Fast Algorithms for Mining Association Rules", In Proceedings of the 20th International Conference on Very Large Databases (VLDB '94), Santiago, Chile, 487-489.

Agrawal R. ve Srikant R., (1995), "Mining Sequential Patterns", 11th International Conference on Data Engineering, Taipei, Taiwan, 3-14.

Alataş, B., (2003), "Nicel Birliktelik Kurallarının Keşfinde Bulanık Mantık ve Genetik. Algoritma Yaklaşımı", Fırat Ünv. Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, Elazığ.

Borges, J. ve Levene, M., (1999), "Data Mining of User Navigation Patterns", Proceedings of the WEBKDD'99 Workshop on Web Usage Analysis abd User Profiling, 15.08.1999, San Diego, CA, USA, 31-36 .

Cooley, R., Srivastava, J., Deshpande, M. ve Tan,P., (2000), "Web Usage Mining:,Discovery and Applications of Usage Patterns from Web Data", PhDthesis,Dept. Of Computer Science, University of Minnesota.

Daş, R., Türkoğlu, İ. ve Poyraz, M. , (2007), "Web Kayıt Dosyalarından İlginç Örüntülerin Keşfedilmesi", Fırat Üniv.Fen ve Müh. Bil. Dergisi 19 (4), 493-503.

Etzioni, O., (1996), "The World Wide Web: Quagmire or Gold Mine", Communications of the ACM, 39(11): 65-68.

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

Guo, Y., Ramamohanarao, K., Park , L., (2007), "Personalized Pagerank for Web Page Prediction Based on Access Time-Length and Frequency", Dept. Of Computer Science and Software Engineering, Melbourne University, 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007.

Han, J. ve Kamber, M., (2006), Data Mining Concepts and Techniques, Morgan Kauffmann Publishers Inc., 1-35.

Han, J. ve Kamber M., (2006), Data Mining Concepts and Techniques, Second Edition.

Jörn, D., (2003), Introduction to Fraud Detection.

Ling, H., Liu, Y., Yang , S., (2007), "An Ant Colony Model for Dynamic Mining of Users Interest Navigation Patterns", School of Management, Hefei University of Technology, 2007 IEEE International Conference on Control and Automation, May 30- June 1 2007, Guangzhou, China.

Özekes, S., (2003), "Veri Madenciligi Modelleri ve Uygulama Alanları", İstanbul Ticaret Üniversitesi Dergisi, Yıl 2, S. 3, s. 65–82.

Ping, N., Jianxin, L., Xiaomin , Z., Keyan, R., (2008), "News Content Recommendation Model Based on Feedback of Web Usage", State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 2009 World Congress on Computer Science and Information Engineering, Beijing.

Sertaç, Ö., (2005), "Veri Madenciliği Kavramı ve Gelişim Süreci", Yeditepe Üniversitesi.

INTERNET KAYNAKLARI

IBM, http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html

USER AGENTS, http://www.user-agents.org/

W3C , http://www.w3.org/TR/WD-logfile.html

YAZ GELISTIR, http://www.yazgelistir.com/Makaleler/1000002145.ygpx

ALTI ÜSTÜ TASARIM
http://www.altiustutasarim.com/arsiv/2006/11/samanliktaki_kullanici.php

EKLER

Ek 1 SC_STATUS Alanının Alabileceği Değerler

200 OK : Aranan belge bulundu ve gönderildi.
204 No Content : Belge bulundu;fakat gönderilecek bir içerik yok . (Browser'da mevcut görüntü değişmez)
301 Moved Permamently : Aranan belge temelli başka bir adrese aktarıldı. (Yeni adres Location Header-Yer Başlığı bölümüne yazılır)
302 Moved Temporarily : Aranan belge geçici olarak başka bir adrese aktarıldı. (Yeni adres Location Header-Yer Başlığı bölümüne yazılır)
400 Cannot be Found : Erişmek istediğiniz dosya bulunamıyor.
401 Unauthorized : Erişmek istediğiniz belge korunmuştur. (Browser, kullanıcı adı ve parola soran bir pencere açabilir)
403 Forbidden : Erişmek istediğiniz belgeye erişim yasaklanmıştır.
403.10 Access Forbidden : Erişmek için sunduğunuz yetki kaydı yeterli değil.
403.11 Access Forbidden : Erişmek için sunduğunuz parola değişmiş.
404 Site not Found : Aradığınız site (URL) bulunamıyor.
405 Resource not Allowed : Dosya, yanlış adres verdiğiniz için bulunamıyor .
406 Not Acceptable : Browser'ınız istediğiniz kaynağı görüntüleme yeteneğinde olmadığı için istediğiniz dosya gönderilmiyor .
410 Does not Exist : Aradığınız dosya temelli bulunamıyor.
412 Precondition Failed : Dosya istemcinin ileri sürdüğü ön şartlara uymadığı için gönderilmiyor.
414 URI Too Long : Dosya için yazılan yol çok uzun.
500 Internal Server Error : Dahili sunucu hatası oluştu.
501 Not Implemented : İstediğiniz yazılım/medya türü bu sunucuda uygulanamıyor.
502 Bad Gateway : Sunucu geçit olarak görev yaparken, kendisinden sonraki bir sunucudan hatalı yanıt aldı.

Ek 2 Log Dosyası Formatları ve Örnekleri

❖ Microsoft IIS 3.0 and 2.0 formatı:

157.55.69.103, -, 12/6/96, 7:08:22, W3SVC, WEBSERVER, 206.129.192.10, 10, 286, 14167, 200, 0, GET, /info/default.asp, Mozilla/2.0 (compatible; MSIE 3.0; Windows 95), http://www.yourcompany.com/default.htm, 35bebd61b31211cfbdcd00c04fd611cf, -,

❖ Microsoft IIS4.0 (W3SVC formatı):

#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 1999-01-24 00:00:06
#Fields: date time c-ip cs-username s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status sc-bytes cs-bytes time-taken s-port cs-version cs(User-Agent) cs(Cookie) cs(Referer)
1999-01-24 00:00:05 208.208.7.34 - W3SVC1 WEBSERVER 206.129.192.10 GET /hitlist/newreports/mwhlcol.gif - 200 0 1119 366 32507 80 HTTP/1.0 Mozilla/4.08+[en]+(Win95;+U) - http://www.marketwave.com/hitlist/newreports/complete_navbar.htm
1999-01-24 00:00:05 208.208.7.34 - W3SVC1 WEBSERVER 206.129.192.10 GET /hitlist/newreports/MWHLGraph24183.GIF - 200 0 9729 373 32967 80 HTTP/1.0 Mozilla/4.08+[en]+(Win95;+U) - http://www.marketwave.com/hitlist/newreports/complete_report.htm

❖ Netscape formatı:

format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] "%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length% "%Req->headers.referer%" "%Req->headers.user-agent%" "%Req->headers.cookie%"
207.225.167.48 - - [08/Dec/1997:02:00:39 -0700] "GET /BannersMsg.class HTTP/1.0" 200 1326 "-" "Mozilla/4.03 [en] (Win95; U)"
207.225.167.134 - - [08/Dec/1997:02:02:20 -0700] "GET / HTTP/1.0" 200 9989 "-" "Mozilla/2.02E (Win95; U)" "35bebd61b31211cfbdcd00c04fd611cf"
207.225.167.134 - - [08/Dec/1997:02:02:22 -0700] "GET /gfx/navbar/bg.gif HTTP/1.0" 304 0 "http://www.azfamily.com/" "Mozilla/2.02E (Win95; U)"

"35bebd61b31211cfbdcd00c04fd611cf"
207.225.167.134 - - [08/Dec/1997:02:02:22 -0700] "GET /gfx/index/hdr/monday.gif HTTP/1.0" 200 3469 "http://www.azfamily.com/" "Mozilla/2.02E (Win95; U)" "35bebd61b31211cfbdcd00c04fd611cf"

❖ **Apache formatı:**

168.191.4.27 - - [28/Dec/1997:01:04:56 -0600] "GET / HTTP/1.0" 200 12077 "http://www.yahoo.com/Business_and_Economy/Companies/Packaging/" "Mozilla/4.02 [en]C-DIAL (Win95; U)"
168.191.4.27 - - [28/Dec/1997:01:06:01 -0600] "GET /whatsnew14.html HTTP/1.0" 200 4791 "http://www.castlerockcontainer.com/" "Mozilla/4.02 [en]C-DIAL (Win95; U)"
168.191.4.27 - - [28/Dec/1997:01:06:23 -0600] "GET /castlerock2.html HTTP/1.0" 200 9836 "http://www.castlerockcontainer.com/whatsnew14.html" "Mozilla/4.02 [en]C-DIAL (Win95; U)"

❖ **O'Reilly WebSite formatı:**

04/03/98 05:23:43 scan.palacenet.net www2.palacenet.net HEAD / Ipswitch_WhatsUp/3.0 200 0 15
04/03/98 05:23:44 pm2-11.palacenet.net www.microt.com GET /PRODUCTS/Images/USR/usr_modem_ani.gif http://www.palacenet.net/ Mozilla/3.01C-KIT (Win95; U) 304 0 0
04/03/98 05:24:24 pm2-18.palacenet.net www.microt.com GET /PRODUCTS/Images/USR/usr_modem_ani.gif http://www.palacenet.net/ Mozilla/3.01C-KIT (Win16; U) 304 0 0
04/03/98 05:24:41 pm2-18.palacenet.net www.microt.com GET /PRODUCTS/Images/USR/usr_modem_ani.gif http://www.palacenet.net/ Mozilla/3.01C-KIT (Win16; U) 206 5356 828
04/03/98 05:24:45 scan.palacenet.net www2.palacenet.net HEAD / Ipswitch_WhatsUp/3.0 200 0 0
04/03/98 05:25:03 pm1-17.palacenet.net www.microt.com GET /PRODUCTS/Images/USR/usr_modem_ani.gif http://www.palacenet.net/ Mozilla/3.01C-KIT (Win16; U) 304 0 172

❖ **WebSTAR formatı:**

!!WebSTAR STARTUP 05/17/99:12:08
#Version: 1.0
#Software: WebSTAR/3.0.2
#Start-Date: 05/17/99:12:08
#Fields: BYTES C-DNS C-IP CS(COOKIE) CS(HOST) CS(REFERER) CS(USER-AGENT) CS-HOST CS-IP CS-METHOD CS-STATUS CS-URI CS-URI-QUERY CS-URI-STEM DATE SC-STATUS TIME TIME_TAKEN
0 - 128.171.118.183:80 "" "www.pdadash.com" "" "Mozilla/4.5 [en] (WinNT; I)" - 128.171.118.183:80 CONDITIONAL_GET 304 /pdadash/advertise/images/goamerica/an-minstrel2.gif - /pdadash/advertise/images/goamerica/an-minstrel2.gif 1999-05-22 304 07:00:15 0
0 - 128.171.118.183:80 "" "www.pdadash.com" "" "Mozilla/4.5 [en] (WinNT; I)" - 128.171.118.183:80 CONDITIONAL_GET 304 /pdadash/advertise/images/goamerica/an-minstrel2.gif - /pdadash/advertise/images/goamerica/an-minstrel2.gif 1999-05-22 304 07:00:17 0
0 - 169.229.52.69:80 "" "www.pdadash.com" "" "Mozilla/4.5 [en] (Win95; I)" - 169.229.52.69:80 CONDITIONAL_GET 304 /pdadash/advertise/images/paragraph/po_and_cg_1.gif - /pdadash/advertise/images/paragraph/po_and_cg_1.gif 1999-05-22 304 07:00:18 3 0 -

- **Proxy Sunucusu Log Dosyası Format Örnekleri**

❖ **MS Proxy 1.0 formatı:**

-, -, MSProxy/1.0, N, 7/10/97, 0:16:08, W3Proxy, TERMINATOR, -, cnn.com, -, 80, 672, 9184, 172, http, tcp, GET, http://www.cnn.com/images/9706/nav/nav_bar_main.gif, -, VFInet, 200
-, -, MSProxy/1.0, N, 7/10/97, 0:18:07, W3Proxy, TERMINATOR, -, webserver.marketwave.com, -, 80, -, 705, 146, http, tcp, GET, http://www.marketwave.com/products.htm, -, VCache, 304
-, -, MSProxy/1.0, N, 7/10/97, 0:20:07, W3Proxy, TERMINATOR, -, www.microsoft.com, -, 80, 235, 14939, 202, http, tcp, GET, http://www.microsoft.com/library/images/gifs/rotateads/IISSide.gif, -, VCache, 304

❖ **MS Proxy 2.0 "regular" formatı:**

206.129.192.39, anonymous, -, N, 8/5/98, 18:12:58, 1, -, -, www.hawaii.rr.com, -, 80, 906, 10626, 11016, http, tcp, -, http://www.hawaii.rr.com/cgi-bin/RoadRunner/News/news.cgi, -, Inet, 200, 0
206.129.192.39, anonymous, -, N, 8/5/98, 18:12:58, 1, -, -, www.hawaii.rr.com, -, 80, 516, 3265, 3597, http, -, -, http://www.hawaii.rr.com/Around_Town/images/rrh_left.gif, -, Inet, 200, 0
206.129.192.39, anonymous, -, N, 8/5/98, 18:12:58, 1, -, -, www.hawaii.rr.com, -, 80, 1000, 17791, 18127, http, tcp, -, http://www.hawaii.rr.com/Around_Town/images/hi-bkgrd_mat.gif, -, Inet, 200, 0

❖ **MS Proxy 2.0 "verbose" formatı:**

206.129.192.32, anonymous, Mozilla/4.05 [en] (WinNT; I ;Nav), N, 8/5/98, 18:09:16, W3Proxy, LIVE1, -, www.digitaria.com, 192.215.146.37, 80, 141, 1217, 1564, http, tcp, GET, http://www.digitaria.com/pages/workf.html, text/html, Inet, 200, 0
206.129.192.32, anonymous, Mozilla/4.05 [en] (WinNT; I ;Nav), N, 8/5/98, 18:09:17, W3Proxy, LIVE1, -, www.digitaria.com, 192.215.146.37, 80, 375, 2997, 3344, http, tcp, GET, http://www.digitaria.com/pages/workm.html, text/html, Inet, 200, 0
206.129.192.32, anonymous, Mozilla/4.05 [en] (WinNT; I ;Nav), N, 8/5/98, 18:09:17, W3Proxy, LIVE1, -, www.digitaria.com, 192.215.146.37, 80, 468, 5471, 5818, http, tcp, GET, http://www.digitaria.com/pages/workt.html, text/html, Inet, 200, 0

❖ **Netscape Proxy formatı:**

format=%Ses->client.ip% 146.127.62.22 %Req->vars.pauth-user% [%SYSDATE%] "%Req->reqpb.proxy-request%" %Req->srvhdrs.clf-status% %Req->vars.p2c-cl% %Req->vars.remote-status% %Req->vars.r2p-cl% %Req->headers.content-length% %Req->vars.p2r-cl% %Req->vars.c2p-hl% %Req->vars.p2c-hl% %Req->vars.p2r-hl% %Req->vars.r2p-hl% %Req->vars.xfer-time% %Req->vars.actual-route% %Req->vars.cli-status% %Req->vars.svr-status% %Req->vars.cch-status%
146.127.123.16 146.127.62.22 - [10/Dec/1997:00:30:09 -0500] "GET http://www.nba.com/bulls/ HTTP/1.0" 200 8816 200 8816 - - 321 164 359 164 1 SOCKS(146.127.11.3:1080) FIN FIN NON-CACHEABLE
146.127.253.84 146.127.62.22 - [10/Dec/1997:00:30:12 -0500] "GET

http://www.pathfinder.com/NY1/bug.html HTTP/1.0" 200 377 200 377 - - 392 203 418 203 1 SOCKS(146.127.11.3:1080) FIN FIN REFRESHED

146.127.253.84 146.127.62.22 - [10/Dec/1997:00:30:12 -0500] "GET http://www.pathfinder.com/NY1/images/steel.gif HTTP/1.0" 304 - 304 - - - 443 142 468 142 0 SOCKS(146.127.11.3:1080) FIN FIN UP-TO-DATE

Ek 3 P_USER_IDENTIFICATION Prosedürünün Kodlaması

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[P_USER_IDENTIFICATION]
AS
BEGIN
DECLARE @VALUE NVARCHAR(100)
DECLARE @USERIDEN NVARCHAR(4000)
DECLARE @ALUSER NVARCHAR(4000)
DECLARE @CS_USER_AGENT NVARCHAR(255)
DECLARE @CS_URI_STEM NVARCHAR(255)
DECLARE @REFERRER NVARCHAR(255)
DECLARE @BROWSER NVARCHAR(255)
DECLARE @INDIS INT

SET @USERIDEN=''
SET @ALUSER=''
SET @BROWSER=''
SET @INDIS=0

DECLARE TEMP_TABLE_CURSOR CURSOR FOR
SELECT CS_USER_AGENT,CS_URI_STEM,REFERRER FROM T_LOG_DATA
ORDER BY DATE,TIME
OPEN TEMP_TABLE_CURSOR
FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@CS_USER_AGENT,@CS_URI_STEM,@REFERRER
WHILE @@FETCH_STATUS =0
BEGIN
SET @INDIS=0

IF(UPPER(@BROWSER)=UPPER(@CS_USER_AGENT))
BEGIN
DECLARE DATA_CURSOR CURSOR FOR
SELECT VALUE FROM Split(CONVERT(VARCHAR,@USERIDEN),'*')
OPEN DATA_CURSOR
FETCH NEXT FROM DATA_CURSOR INTO @VALUE


WHILE @@FETCH_STATUS =0
BEGIN
IF (UPPER(@VALUE)=UPPER(@REFERRER))
BEGIN
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
SET @INDIS=@INDIS+1
END
FETCH NEXT FROM DATA_CURSOR INTO @VALUE
```

```
END
CLOSE DATA_CURSOR
DEALLOCATE DATA_CURSOR

IF(@INDIS=0)
BEGIN
SET @ALUSER=@USERIDEN
INSERT INTO T_USER_IDENTIFICATION_RESULT
SELECT @ALUSER
SET @USERIDEN=''
IF(@REFERRER!='-')
BEGIN
SET @USERIDEN=@USERIDEN+@REFERRER+'*'+@CS_URI_STEM+'*'
END
ELSE
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
END
END

ELSE
BEGIN
IF(@USERIDEN!='')
BEGIN
SET @ALUSER=@USERIDEN
INSERT INTO T_USER_IDENTIFICATION_RESULT
SELECT @ALUSER
END
SET @USERIDEN=''
SET @BROWSER=@CS_USER_AGENT
IF(@REFERRER!='-')
SET @USERIDEN=@USERIDEN+@REFERRER+'*'+@CS_URI_STEM+'*'
ELSE
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
END

FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@CS_USER_AGENT,@CS_URI_STEM,@REFERRER
END
CLOSE TEMP_TABLE_CURSOR
DEALLOCATE TEMP_TABLE_CURSOR
END
```

Ek 4 P_SESSION_IDENTIFICATION Prosedürünün Kodlaması

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[P_SESSION_IDENTIFICATION]
AS
BEGIN
DECLARE @VALUE NVARCHAR(100)
DECLARE @USERIDEN NVARCHAR(4000)
DECLARE @ALUSER NVARCHAR(4000)
DECLARE @CS_USER_AGENT NVARCHAR(255)
DECLARE @CS_URI_STEM NVARCHAR(255)
DECLARE @REFERRER NVARCHAR(255)
DECLARE @DATE DATETIME
DECLARE @TIME NVARCHAR(255)
DECLARE @BROWSER NVARCHAR(255)
DECLARE @INDIS INT
DECLARE @START_DATETIME NVARCHAR(8)
SET @USERIDEN=''
SET @ALUSER=''
SET @BROWSER=''
SET @INDIS=0
SET @START_DATETIME=CONVERT(DATETIME,'00:00:00')
DECLARE TEMP_TABLE_CURSOR CURSOR FOR
SELECT CS_USER_AGENT,CS_URI_STEM,REFERRER,TIME FROM T_LOG_DATA
GROUP BY CS_USER_AGENT,DATE,TIME,CS_URI_STEM,REFERRER
ORDER BY CS_USER_AGENT,DATE,TIME
OPEN TEMP_TABLE_CURSOR
FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@CS_USER_AGENT,@CS_URI_STEM,@REFERRER,@TIME
WHILE @@FETCH_STATUS =0
BEGIN
SET @INDIS=0
IF(UPPER(@BROWSER)=UPPER(@CS_USER_AGENT))
BEGIN
DECLARE DATA_CURSOR CURSOR FOR
SELECT VALUE FROM Split(CONVERT(VARCHAR,@USERIDEN),'*')
OPEN DATA_CURSOR
FETCH NEXT FROM DATA_CURSOR INTO @VALUE
WHILE @@FETCH_STATUS =0
BEGIN
IF (UPPER(@VALUE)=UPPER(@REFERRER))
BEGIN
IF(ABS(DATEDIFF(MINUTE,@TIME,@START_DATETIME))>30)
BEGIN
SET @ALUSER=@USERIDEN
INSERT INTO T_SESSION_IDENTIFICATION_RESULT
SELECT @CS_USER_AGENT,@ALUSER
```

```
IF(@REFERRER!='-')
SET @USERIDEN=@USERIDEN+@REFERRER+'*'+@CS_URI_STEM+'*'
ELSE
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
END
ELSE
BEGIN
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
SET @INDIS=@INDIS+1
END
END
FETCH NEXT FROM DATA_CURSOR INTO @VALUE
END
CLOSE DATA_CURSOR
DEALLOCATE DATA_CURSOR
IF(@INDIS=0)
BEGIN
SET @ALUSER=@USERIDEN
INSERT INTO T_SESSION_IDENTIFICATION_RESULT
SELECT @CS_USER_AGENT,@ALUSER
SET @USERIDEN=''
IF(@REFERRER!='-')
BEGIN
SET @USERIDEN=@USERIDEN+@REFERRER+'*'+@CS_URI_STEM+'*'
END
ELSE
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
END
SET @START_DATETIME=CONVERT(DATETIME,@TIME)
END
ELSE
BEGIN
IF(@USERIDEN!='')
BEGIN
SET @ALUSER=@USERIDEN
INSERT INTO T_SESSION_IDENTIFICATION_RESULT
SELECT @CS_USER_AGENT,@ALUSER
END
SET @START_DATETIME=@TIME
SET @USERIDEN=''
SET @BROWSER=@CS_USER_AGENT
IF(@REFERRER!='-')
SET @USERIDEN=@USERIDEN+@REFERRER+'*'+@CS_URI_STEM+'*'
ELSE
SET @USERIDEN=@USERIDEN+@CS_URI_STEM+'*'
END
FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@CS_USER_AGENT,@CS_URI_STEM,@REFERRER,@TIME
END
CLOSE TEMP_TABLE_CURSOR
DEALLOCATE TEMP_TABLE_CURSOR
```

```
END
```

Ek 5 P_APRIORI_ALGORITHM Prosedürünün Kodlaması

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go


CREATE PROCEDURE [dbo].[P_APRIORI_ALGORITHM]
AS
BEGIN
DECLARE @PAGE NVARCHAR(500)
DECLARE @RECORD_COUNT NUMERIC(18,0)
DECLARE @PAGE_RECORD_COUNT NUMERIC(18,0)
DECLARE @TEMP_COUNT_1 NUMERIC(18,0)
DECLARE @C_IP NVARCHAR(250)

SELECT @RECORD_COUNT=COUNT(1) FROM dbo.T_FROM_TO_TIME_TABLES
SET @TEMP_COUNT_1=0

DECLARE DATA_CURSOR CURSOR FOR
SELECT DISTINCT PAGE FROM (
SELECT DISTINCT FROM_PAGE PAGE  FROM dbo.T_FROM_TO_TIME_TABLES
UNION ALL
SELECT DISTINCT TO_PAGE  FROM dbo.T_FROM_TO_TIME_TABLES) DATAS

OPEN DATA_CURSOR
FETCH NEXT FROM DATA_CURSOR INTO @PAGE
WHILE @@FETCH_STATUS=0
BEGIN

SET @PAGE_RECORD_COUNT=0

DECLARE DATA_CURSOR_DETAIL CURSOR FOR
SELECT DISTINCT C_IP FROM T_FROM_TO_TIME_TABLES

OPEN DATA_CURSOR_DETAIL
FETCH NEXT FROM DATA_CURSOR_DETAIL INTO @C_IP
WHILE @@FETCH_STATUS=0
BEGIN

SELECT @TEMP_COUNT_1=COUNT(1) FROM T_FROM_TO_TIME_TABLES
WHERE C_IP=@C_IP AND (FROM_PAGE=@PAGE OR TO_PAGE=@PAGE)

IF(@TEMP_COUNT_1>0)
begin
```

```
SET @PAGE_RECORD_COUNT=@PAGE_RECORD_COUNT+1
end

FETCH NEXT FROM DATA_CURSOR_DETAIL INTO @C_IP
END
CLOSE DATA_CURSOR_DETAIL
DEALLOCATE DATA_CURSOR_DETAIL


INSERT INTO dbo.T_APRIORI_HEADER
SELECT
@PAGE_RECORD_COUNT,(@PAGE_RECORD_COUNT/@RECORD_COUNT),1,0

INSERT INTO dbo.T_APRIORI_DETAIL
SELECT @@IDENTITY,@PAGE


FETCH NEXT FROM DATA_CURSOR INTO @PAGE
END
CLOSE DATA_CURSOR
DEALLOCATE DATA_CURSOR

END
```

Ek 6 P_TFPR_ALGORITHM Prosedürünün Kodlaması

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[P_TFPR_ALGORITHM]
(
@U_NODE NVARCHAR(500),
@RESULT DECIMAL(38,20) OUTPUT
)
AS
BEGIN
DECLARE @V_NODE NVARCHAR(500)
DECLARE @C_IP NVARCHAR(250)
DECLARE @CS_USER_AGENT NVARCHAR(250)
DECLARE @BIASING_FACTOR DECIMAL(3,2)
DECLARE @FIRST_PART DECIMAL(38,20)
DECLARE @SECOND_PART DECIMAL(38,20)
DECLARE @RECURSIVE_RESULT DECIMAL(38,20)
DECLARE @SELECT_COUNT INT
DECLARE @TOTAL_TIME DECIMAL(38,20)
DECLARE @ROWCNT int
DECLARE @MAXROWS int
SET @ROWCNT = 1
SET @BIASING_FACTOR= 0.15
SET @SECOND_PART= 0
SET @RECURSIVE_RESULT=0

CREATE TABLE #WHILETABLE
  (ROWNUM          int IDENTITY (1,1) Primary key,
   FROM_PAGE   Nvarchar(500)
   )
INSERT INTO #WHILETABLE
SELECT DISTINCT FROM_PAGE FROM T_FROM_TO_TIME_TABLES WHERE
TO_PAGE=@U_NODE

SET @FIRST_PART=@BIASING_FACTOR*(SELECT
dbo.F_CALCULATE_TIME(@U_NODE,''))
/(SELECT SUM(CONVERT(DECIMAL,TIME)) FROM  dbo.T_FROM_TO_TIME_TABLES)

SELECT  @MAXROWS =COUNT(*) FROM #WHILETABLE

WHILE EXISTS  (SELECT ROWNUM,FROM_PAGE
               FROM #WHILETABLE
              WHERE ROWNUM <= @MAXROWS
               AND ROWNUM = @ROWCNT
)
 BEGIN
```

```
SELECT @V_NODE=FROM_PAGE
FROM #WHILETABLE
WHERE ROWNUM <= @MAXROWS
AND ROWNUM = @ROWCNT
SELECT @ROWCNT = @ROWCNT + 1
SET @SELECT_COUNT=(SELECT COUNT(1) FROM dbo.T_TEMP WHERE
FROM_PAGE=@U_NODE AND TO_PAGE=@V_NODE)
IF(@SELECT_COUNT=0)
BEGIN
SET @SELECT_COUNT=(SELECT COUNT(1) FROM dbo.T_FROM_TO_TIME_TABLES
WHERE TO_PAGE=@V_NODE)

IF(@SELECT_COUNT>0)
BEGIN

INSERT INTO T_TEMP
SELECT @U_NODE,@V_NODE
BEGIN TRY
EXEC dbo.P_TFPR_ALGORITHM @V_NODE,@RESULT OUTPUT
END TRY
BEGIN CATCH
SET @RESULT = 0
END CATCH
END

SET @TOTAL_TIME=(SELECT SUM(CONVERT(DECIMAL,TIME)) FROM
dbo.T_FROM_TO_TIME_TABLES WHERE FROM_PAGE=@V_NODE)
SET @SECOND_PART=@SECOND_PART+((SELECT
dbo.F_CALCULATE_TIME(@U_NODE,@V_NODE))/
@TOTAL_TIME*ISNULL(@RESULT,0))
END
END
DROP TABLE #WHILETABLE
SET @SECOND_PART=@SECOND_PART*(1-@BIASING_FACTOR)
SET @RESULT=@FIRST_PART+@SECOND_PART

END
```

Ek 7 P_TIME_FREQUENCY_SESSION_IDENTIFICATION Prosedürünün Kodlaması

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO
ALTER  PROCEDURE [dbo].[P_TIME_FREQUENCY_SESSION_IDENTIFICATION]
AS
BEGIN
DECLARE @C_IP NVARCHAR(100)
DECLARE @TIME DATETIME
DECLARE @REFERRER NVARCHAR(100)
DECLARE @CS_USER_AGENT NVARCHAR(250)
DECLARE TEMP_TABLE_CURSOR CURSOR FOR
SELECT C_IP,TIME,REFERRER,CS_USER_AGENT FROM dbo.T_LOG_DATA_TEMP
OPEN TEMP_TABLE_CURSOR
FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@C_IP,@TIME,@REFERRER,@CS_USER_AGENT
WHILE @@FETCH_STATUS =0
BEGIN

INSERT INTO
dbo.T_FROM_TO_TIME_TABLES(TIME,TO_PAGE,C_IP,FROM_PAGE,CS_USER_AGEN
T)
SELECT TOP 1 ABS(DATEDIFF(SECOND,@TIME,TIME)) ,REFERRER
TO_PAGE,C_IP,@REFERRER,@CS_USER_AGENT FROM_PAGE FROM
dbo.T_LOG_DATA
WHERE C_IP=@C_IP AND TIME>@TIME AND REFERRER!=@REFERRER
ORDER BY TIME

FETCH NEXT FROM TEMP_TABLE_CURSOR INTO
@C_IP,@TIME,@REFERRER,@CS_USER_AGENT
END
CLOSE TEMP_TABLE_CURSOR
DEALLOCATE TEMP_TABLE_CURSOR
END
```

Ek 8 Sayfa Bağlantı Adresleri

ANASAYFA :http://video.webmining.com/

MA1:http://video.webmining.com/53shekill53/ferrari_lamgorghini/270244/fe7a8f956161a4ce6fa0dbf6f72f8d

MA2:http://video.webmining.com/vitamin45/motorsiklet_kazasi/274556/b0f62b71d3547d446f60ecf5d783de8a

MA3:http://video.webmining.com/53shekill53/CEKILIN_TREN_GELIYOR/274070/

MA4:http://video.webmining.com/partiefendisi/Hizli_Tren/269117/561527f4c10b3639b91f216754b620d3

ST1:http://video.webmining.com/alfadizaynbilisim/Beyaz_Sahan_Atismasi/270560/f075615d42d43d4db4fd928

ST2:http://video.webmining.com/architect.tunc/Steve_Martin_Show/271117/dffeca5578c98123f6fca03d51bc2

ST3:http://video.webmining.com/abdi_kaplan/beyaz_showda_beyazi_saf_disi_birakan_sahan_gokbaka/107223

ST4:http://video.webmining.com/cafeses/Recep_ivedik2/250964/

ST5:http://video.webmining.com/recbymehmetergin/Recep_ivedik_2_Sahan_Gokbakar_Sinema_Fragmani_12_S/251386/

K1:http://video.webmining.com/gultekin.serkan/talihsiz_kizlar/267130/3c71e26f207360db56fcf8ac561506

K2:http://video.webmining.com/teknisyenarif/sakar_kamyoncu_amuda_kalkan_kamyon/264532/84ab6ad648521

K3:http://video.webmining.com/alevtopu50/Kadinlar_teknolojiye_uzaktir/209241/

K4:http://video.webmining.com/sserkan_16/bu_adamlar_saka_konusunda_cok_yaraticilar_vallahi/271959/

K5:http://video.webmining.com/hayalet_sedat00/cirkin_kadin_yoktur_fotoshopsuz_kadin_vardi/56745/

K6:http://video.webmining.com/necdeter/Sener_Sen_efsanesi/152/

K7:http://video.webmining.com/ahmetim1979/ilyas_salman_ve_senersen_iklisinn_mutesem_kapisma/272686/

E1:http://video.webmining.com/fikri77/motorsuz_ucaklara_ne_denir_50_sarisin/263773/e8d17f3a946e0b45

E2:http://video.webmining.com/iibrahim_1989i/aman_tanrim_bu_ne_gusellik_beyler_karar_sizin_spr/10618/

E3:http://video.webmining.com/tumaycakir/kadinlarda_dagitti/264590/12c4d9695cad0a8dc7c4449bfe2b9e21

E4:http://video.webmining.com/odem_gfb/sexy_kizlar_sutyen_cikartma_teknigi/267315/89dedaeb67f08e195af50d405d7cc91a

E5:http://video.webmining.com/pointless_darkness/paris_hilton/393/

E6:http://video.webmining.com/alevtopu50/Italyan_guzel/138047/

E7:http://video.webmining.com/sevdaseli_sevdaseli/TANYELI_GOBEK_SOW/264675/

E8:http://video.webmining.com/wigproductions/Arap_Kizlari/41686/

H1:http://video.webmining.com/ekipcalismasi/Benim_icin_Davos_bitmistir/272654/8e964e97858e65203e118

M1:http://video.webmining.com/b_hasan/Hadise_D_m_Tek_Tek_Eurovision_2009_Turkey/256181/d9ee068b7d7251da28bed8e6d7627b1c

SG1:http://video. webmining.com/usecret/Ruzgarin_Vahsi_Etkisi/1722/

SEARCH1:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=davos

SEARCH2:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=beyaz+show

SEARCH3:http://webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=fashion

SEARCH4:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=clip

SEARCH5:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=araba&p=1

SEARCH6:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=dans&p=10

SEARCH7:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=tren&p=7

SEARCH8:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=webmining+komik

SEARCH9:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=film

SEARCH10:http://video.webmining.com/ep/SearchResults.aspx?SearchType=TagSearch&cname=video&tname=sexy

ÖZGEÇMİŞ

Doğum Tarihi	12.05.1984	
Doğum Yeri	İstanbul	
Lise	1995 - 2002	Beyoğlu Anadolu Lisesi
Lisans	2002 - 2006	Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
Yüksek Lisans	2006 -	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı

Çalıştığı Kurum(lar)

2007 - Devam ediyor	İdea Teknoloji Çözümleri Yazılım Uzmanı