

# AÇIK KAYNAK KODLU VERİ MADENCİLİĞİ PROGRAMLARI: WEKA'DA ÖRNEK UYGULAMA

Murat DENER  
Gazi Üniversitesi  
Elektronik-Bilgisayar Eğitimi Bölümü  
[muratedener@gazi.edu.tr](mailto:muratedener@gazi.edu.tr)

Murat DÖRTERLER  
Gazi Üniversitesi  
Elektronik-Bilgisayar Eğitimi Bölümü  
[dorterler@gazi.edu.tr](mailto:dorterler@gazi.edu.tr)

Abdullah ORMAN  
Gazi Üniversitesi  
Elektronik-Bilgisayar Eğitimi Bölümü  
[abdullah@gazi.edu.tr](mailto:abdullah@gazi.edu.tr)

## ÖZET

*Veri Madenciliği, veriden bilgi elde etme amaçlı kullanılan teknikler bütünüdür. İstatistiksel analiz tekniklerinin ve yapay zekâ algoritmalarının bir arada kullanılarak veri içerisindeki gizli bilgilerin açığa çıkarılması ve verinin nitelikli bilgiye dönüştürülmesi sürecidir. Veri Madenciliği uygulamalarını gerçekleştirmek için ticari ve açık kaynak olmak üzere birçok program mevcuttur. Bu makalede açık kaynak kodlu Veri Madenciliği programlarından olan RapidMiner(YALE), WEKA ve R anlatılmış olup, bu programların karşılaştırılmalarına yer verilmiştir. Ayrıca WEKA'da örnek bir uygulama sunulmuştur. Gerçekleştirilen uygulamanın lisansüstü eğitimi veren tüm Enstitülere yararlı olacağı değerlendirilmektedir.*

## ABSTRACT

*Data Mining is a technique designed to extract information from data sets. It is a process used to reveal hidden information in data and transform data into codified information by using a combination of both statistical analysis techniques and artificial intelligence algorithms. A lot of softwares exist for implement to Data Mining Applications with the inclusion of commercial and open source. In this article, RapidMiner(YALE), WEKA and R are explained which are open source softwares. Comparison of this softwares are mentioned. Also, sample application is showed in WEKA. It is claimed that this application is useful for Institutes.*

**Anahtar Kelimeler:** Veri Madenciliği, Açık Kaynak, WEKA, Örnek Uygulama

## 1. GİRİŞ

Günümüzde birçok kaynaktan veri alıp bu verileri veritabanlarında saklayan kurumların amaçlarından biri de ham verileri bilgiye dönüştürmektir. Bu işlem yani veriyi bilgiye dönüştürme işlemi Veri Madenciliği olarak adlandırılmaktadır. Son yıllarda ölçüm cihazlarının artmasına paralel olarak veri

sayısı ve türleri artmaktadır. Veri toplama araçları ve veri tabanı teknolojilerindeki gelişmeler, bilgi

depolarında çok miktarda bilginin depolanmasını ve çözümlenmesini gerektirmektedir. Bilgisayar teknolojilerindeki gelişmeler doğrultusunda Veri Madenciliği yöntemleri ve programlarının amacı büyük miktarlardaki verileri etkin ve verimli hale getirmektedir. Bilgi ve tecrübeyi birleştirmek için Veri Madenciliği konusunda geliştirilmiş yazılımların kullanılması gerekmektedir. Hızla artan veri kayıtları (GB/saat), Otomatik istasyonlar, Uydu ve uzaktan algılama sistemleri, Teleskopla uzay taramaları, Gen teknolojisindeki gelişmeler, Bilimsel hesaplamalar, benzetimler, modeller, Veri Madenciliğini zorunlu kılmıştır.

Teknolojinin gelişimiyle bilgisayar ortamında ve veritabanlarında tutulan veri miktarının artması, yeni veri toplama yolları, otomatik veri toplama aletleri, veritabanı sistemleri, bilgisayar kullanımının artması, büyük veri kaynakları (İş dünyası: Web, e-ticaret, alışveriş, hisse senetleri,...), bilim dünyası (Uzaktan algılama ve izleme, bioinformatik, simülasyonlar..) toplum (haberler, digital kameralar, YouTube, Facebook...) neden Veri Madenciliği sorusuna cevap vermektedir [1].

Veri Madenciliği uygulamalarını gerçekleştirmek için programlara ihtiyaç duyulur. Bu kapsamda, SPSS Clementine, Excel, SPSS, SAS, Angoss, KXEN, SQL Server, MATLAB ticari ve RapidMiner(YALE), WEKA, R, C4.5, Orange, KNIME açık kaynak olmak üzere birçok program geliştirilmiştir.

Bu çalışmada Veri Madenciliği Açık Kaynak Kodlu programlarına değinilmiş, programlar karşılaştırılmış ve örnek bir uygulama gerçekleştirilmiştir. İkinci bölümde Veri Madenciliği, üçüncü bölümde Açık Kaynak Kodlu Veri Madenciliği Programlarından RapidMiner(YALE), WEKA ve R anlatılmıştır. Dördüncü Bölümde bu programlar karşılaştırılmıştır. Beşinci bölümde ise WEKA ile gerçekleştirilen örnek bir uygulama sunulmuştur.

## 2. VERİ MADENCİLİĞİ

Veri Madenciliği; veri ambarlarındaki tutulan, çok çeşitli ve çok miktarda veriye dayanarak daha önce keşfedilmemiş bilgileri ortaya çıkarmak, bunları karar verme ve eylem planını gerçekleştirmek için kullanma sürecidir. Büyük miktarda veri içinden, gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların aranmasıdır. Veri Madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir. Veriler arasındaki ilişkiyi, kuralları ve özellikleri belirlemekten bilgisayar sorumludur. Amaç, daha önceden fark edilmemiş veri desenlerini tespit edebilmektir.

Etkin bir Veri Madenciliği Uygulaması için farklı tipteki verileri ele alma, Veri Madenciliği algoritmasının etkinliği ve ölçeklenebilirliği, sonuçların yararlılık, kesinlik ve anlamlılık kısıtlarını sağlaması, keşfedilen kuralların çeşitli biçimlerde gösterimi, farklı ortamlarda yer alan veri üzerinde işlem yapabilme, gizlilik ve veri güvenliği özelliklerinin sağlanması gereklidir. Alternatif olarak Veri Madenciliği aslında bilgi keşfi sürecinin bir parçası şeklinde kabul görmektedir. Bilgi keşfi sürecinin aşamaları aşağıda verilmiştir.

- 1-Veri Temizleme (gürültülü ve tutarsız verileri çıkarmak)
- 2-Veri Bütünleştirme (birçok veri kaynağını birleştirebilmek)
- 3-Veri Seçme (Yapılacak olan analiz ile ilgili olan verileri belirlemek)
- 4-Veri Dönüşümü (Verinin Veri Madenciliği tekniğinden kullanılabilir hale dönüşümünü gerçekleştirmek)
- 5-Veri Madenciliği (Veri örüntülerini yakalayabilmek için akıllı metotları uygulamak)
- 6-Örüntü Değerlendirme (Bazı ölçümlere göre elde edilmiş bilgiyi temsil eden ilginç örüntüleri tanımlamak)
- 7-Bilgi Sunumu (Madenciliği yapılmış olan elde edilmiş bilginin kullanıcıya sunumunu gerçekleştirmek), [2,3]

Veri Madenciliği çalışmaları yapmak için hem ticari hem de açık kaynak programlar geliştirilmiştir. Programlar içerisinde birçok algoritma bulunmaktadır. Bu algoritmaları kullanarak elimizde bulunan verilerden, anlamlı bilgiler çıkarılabilmektedir.

## 3. AÇIK KAYNAK KODLU VERİ MADENCİLİĞİ PROGRAMLARI

Veri Madenciliği uygulamaları yapmak için bilgisayar programı kullanmak gereklidir. Bu

kapsamda birçok yazılım geliştirilmiştir. Bu bölümde Açık Kaynak Kodlu Veri Madenciliği Programlarından olan RapidMiner(YALE), WEKA ve R programlarına değinilmiştir.

### 3.1. RAPİDMİNER (YALE)

Amerika'da bulunan YALE üniversitesi bilim adamları tarafından Java dili kullanılarak geliştirilmiştir. YALE'de çok sayıda veri işlenerek, bunlar üzerinden anlamlı bilgiler çıkarılabilir. Aml, arff, att, bib, clm, cms, cri, csv, dat, ioc, log, mat, mod, obf, bar, per, res, sim, thr, wgt, wls, xrff uzantılı dosyaları desteklemektedir. Diğer programlar gibi birkaç tane format desteklememesi YALE'nin artılarından [4].

Makine Öğrenme Algoritmaları olarak Destek Vektör Makinelerini içeren büyük sayıdaki öğrenme modelleri için sınıflandırma ve regresyon, Karar Ağaçları, Bayesian, Mantıksal Kümeler, İlişkilendirme Kuralları ve Kümeleme için birçok algoritma (k-means, k-medoids, dbscan), WEKA'da olan her şey, veri ön işleme için ayırma, normalizasyon, filtreleme gibi özellikler, genetik algoritma, yapay sinir ağları, 3D ile verileri analiz etme gibi birçok özelliği bulunmaktadır. 400'den fazla algoritmaya sahiptir. Oracle, Microsoft SQL Server, PostgreSQL, veya mySQL veritabanlarından veriler YALE'ye aktarılabilir. Eğer veritabanı yönetim sistemi desteklenmiyorsa, jdbc driveri classpath değişkenine eklenerek sorun giderilebilir.

YALE'de veri kümesi XML olarak ifade edilir. Aşağıda örnek veri kümesi verilmiştir.

```
<attributeset default source
="golf.dat">
<attribute
name ="Outlook"
sourcecol ="1"
valuetype ="nominal"
blocktype ="single value"
classes ="rain overcast sunny"/>
<attribute
name ="Temperature"
sourcecol ="2"
valuetype ="integer"
blocktype ="single value"/>
<attribute
name ="Humidity"
sourcecol ="3"
valuetype ="integer"
blocktype ="single value"/>
<attribute
name ="Wind"
sourcecol ="4"
valuetype ="nominal"
blocktype ="single value"
classes ="true false "/>
</label
```

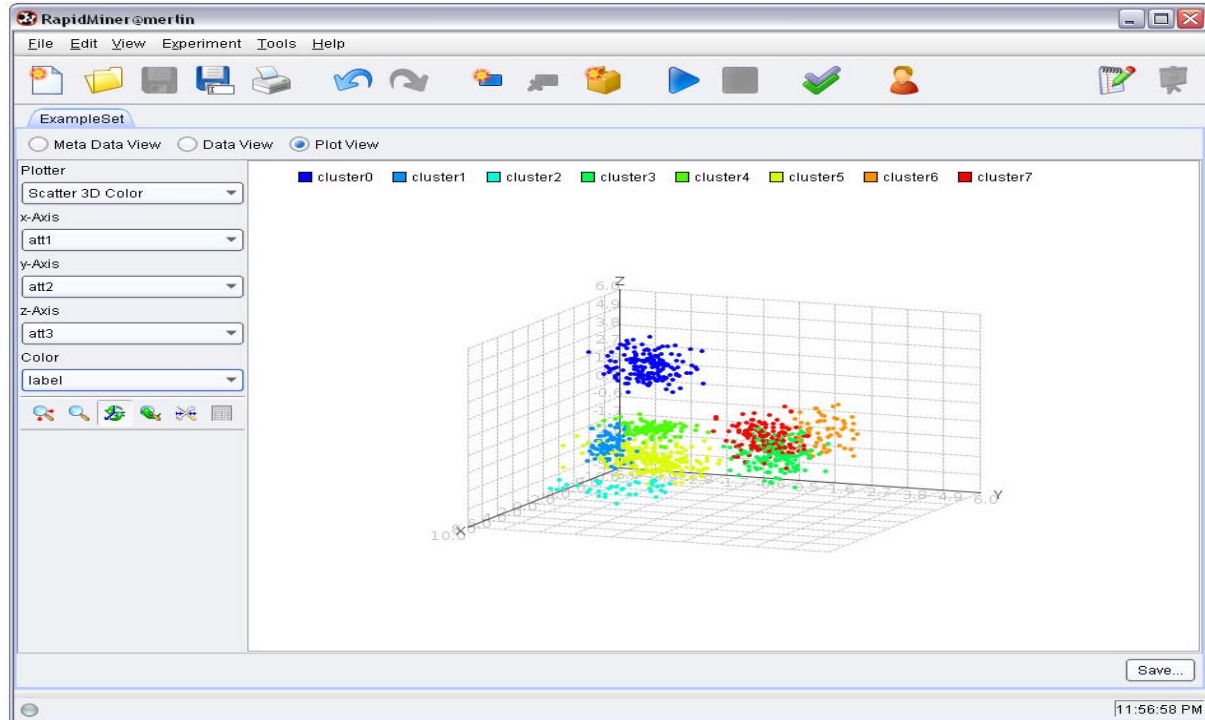
```
name ="Play"  
sourcecol ="5"  
valuetype ="nominal"  
blocktype ="single value"  
classes ="yes no"/>  
</attributeset>
```

İçerisinde yüzlerce özellik barındırdığı gibi kullanıcıya yakınlığı açısından da diğer

programlardan oldukça üstündür. YALE ilk çalıştırıldığında, New diyerek yeni bir uygulama oluşturulabilir, Open diyerek te varolan uygulamalar açılabilir. Program bünyesinde her bir algoritma için örnek bulunmaktadır. Şekil 1'de YALE'de açılış ekranı, Şekil 2'de ise YALE'de kümeleme örneği verilmiştir.



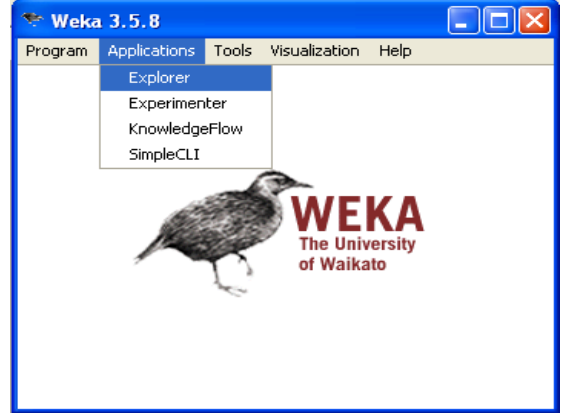
Şekil 1. YALE'de Açılış Ekranı



Şekil 2. YALE'de Kümeleme

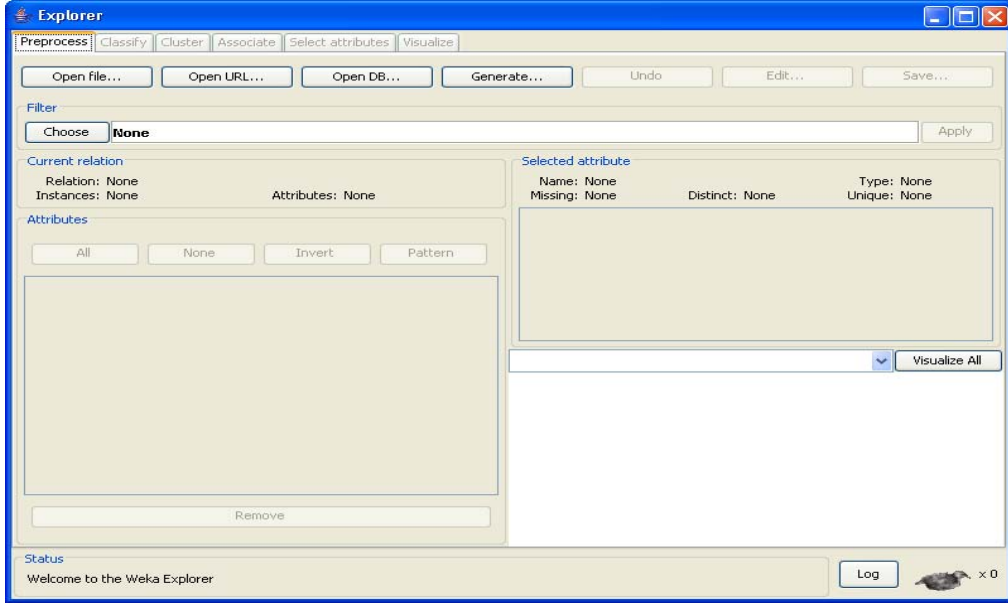
### 3.2. WEKA

WEKA bir proje olarak başlayıp bugün dünya üzerinde birçok insan tarafından kullanılmaya başlanan bir Veri Madenciliği uygulaması geliştirme programıdır. WEKA java platformu üzerinde geliştirilmiş açık kodlu bir programdır. WEKA çalıştırıldıktan sonra Şekil 3’de görüldüğü gibi, Application menüsünde çalışılabilecek modlar listelenmektedir. Bunlar komut modunda çalışmayı sağlayan Simple CLI, projeyi adım adım görsel ortamda gerçekleştirmeyi sağlayan Explorer ve projeyi sürükleyip bırak yöntemiyle gerçekleştirmeyi sağlayan KnowledgeFlow seçenekleridir.



Şekil 3. WEKA’da Applications Menü

Explorer seçeneği seçildikten sonra üzerinde çalışılacak verilerin seçilmesi, bu veriler üzerinde temizleme ve dönüştürme işlemlerinin gerçekleştirilebilmesini sağlayan Şekil 4’deki ekran ile karşılaşılır.



Şekil 4. WEKA’da Veri Seçimi

Arff, Csv, C4.5 formatında bulunan dosyalar WEKA’da import edilebilir. Herhangi bir text soyadaki verileri WEKA ile işlemek olanaksızdır. Ayrıca Jdbc kullanılarak veritabanına bağlanıp burada da işlemler yapılabilir. WEKA’nın içerisinde Veri İşleme, Veri Sınıflandırma, Veri Kümeleme, Veri İlişkilendirme özellikleri mevcuttur.

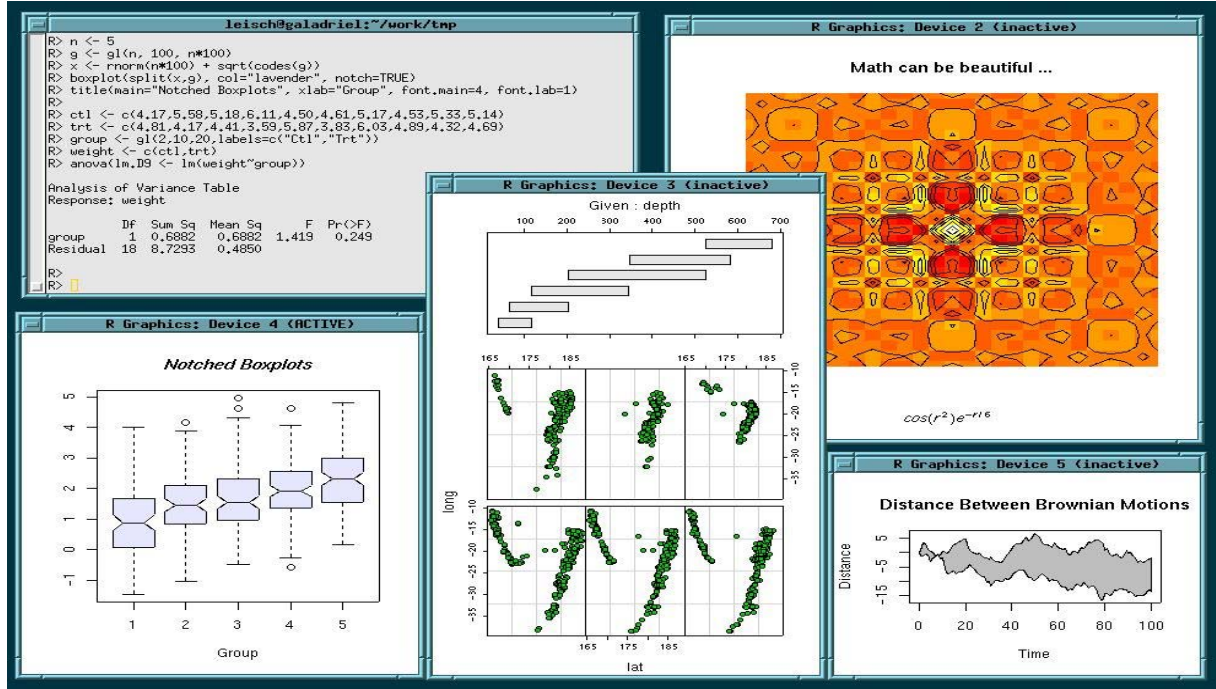
Bu adımdan sonra yapılacak olan projenin amacına göre açılan sayfadaki uygun tabdaki (Sınıflandırma, Kümeleme, İlişkilendirme) uygun algoritma veya algoritmalar seçilerek veriler üzerine uygulanmakta ve en doğru sonucu veren algoritma seçilebilmektedir.

### 3.3. R

Grafikler, istatistiksel hesaplamalar, veri analizleri için geliştirilmiş bir programdır. S diline benzer bir GNU projesidir. Yeni Zelanda’da bulunan Auckland Üniversitesi İstatistik Bölümünde bilim adamlarından olan Robert Gentleman ve Ross Ihaka tarafından geliştirilmiştir. R & R olarak ta bilinir. R, farklı uygulamalar ile S diline üstünlük sağlamaktadır. Lineer ve lineer olmayan modelleme, klasik istatistiksel testler, zaman serileri analizi, sınıflandırma, kümeleme gibi özellikleri bünyesinde bulundurmaktadır. R, Windows, MacOS X ve Linux sistemleri üzerinde çalışabilmektedir [5].

R yaygın olarak pencereci sistemlerde kullanılır. R'nin X Window sistemi üzerinde kullanılması tavsiye edilmektedir. Açık sistemlerin kullanıcıya sunduğu en büyük özelliklerinden biri olan X Window, Linux'un doğduğu andan itibaren destek görmeye başlamıştır. İnternet üzerinde bedava dağıtılmasıyla Linux dağıtımı altında bir standart olarak kendine yer edinmiştir. X Window, istemci-sunucu modeline göre çalışır. Ana makina üzerinde çalışan X sunucusu, grafik donanımı üzerindeki

tüm giriş-çıkış yetkilere sahiptir. Bir X istemcisi, sunucuya bağlanarak istediği işlemleri sunucuya yaptırır. İstemcinin görevi emir vermek, sunucunun ise verilen emri görünür hale getirmektir [6]. Windows veya MacOS üzerinde R'yi çalıştırmak için uzman yardımına ihtiyaç vardır. Kullanıcılar, R'yi çoğunlukla Unix makineler üzerinde çalıştırırlar. Şekil 5'de Unix makinelerde R programının ekran görüntüsü verilmiştir.



Şekil 5. Unix makinelerde R

R'yi Unix makinelerde çalıştırabilmek için aşağıdaki adımlar izlenir.

—Problemi çözümü için gereken veri dosyaları barındırmak için dizin oluşturulur.

```
$ mkdir work
$ cd work
```

—R programının çalıştırılması için aşağıdaki komut yazılır.

```
$ R
```

—R programından çıkmak için aşağıdaki komut yazılır.

```
> q()
```

—Fonksiyonların özelliklerini öğrenmek için aşağıdaki komutlar yazılabilir.

```
> help(solve)
> ?solve
```

Verilerin işleniş şekli de aşağıda verilmektedir.

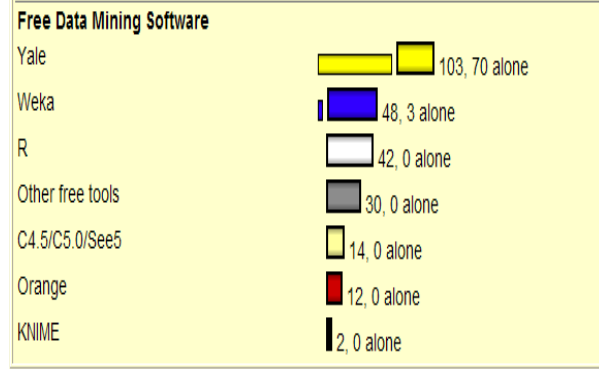
```
> incomes <- c(60, 49, 40, 61, 64, 60,
59, 54, 62, 69, 70, 42, 56,
61, 61, 61, 58, 51, 48, 65, 49, 49,
41, 48, 52, 46, 59, 46, 58, 43)
```

#### 4. PROGRAMLARIN KARŞILAŞTIRILMASI

YALE, WEKA ve R dâhil olmak üzere açık kaynak kodlu Veri Madenciliği programları arasında liderdir. Hem kullanım kolaylığı hem de içerisinde yüzlerce özelliği barındırması YALE'yi WEKA'dan üstün kılmaktadır. YALE'de 3D görsellerin fazlalığı kullanıcıya oldukça yardımcı olmaktadır. WEKA'nın kullanımı da kolaydır fakat desteklediği algoritmaların sayısı YALE'ye göre daha azdır. YALE 22'ye yakın dosya formatını desteklerken, WEKA'nın desteklediği dosya formatı sayısı 4 ile sınırlıdır. Ancak çoğu Veri Madenciliği uygulamasını geliştirmede WEKA yeterli olmaktadır. Bundan dolayı çoğu kullanıcı WEKA'yı tercih etmektedir. R ise hem kullanım kolaylığı hem de desteklediği algoritmalar ile YALE ve WEKA'nın altında bulunmaktadır. R, Unix makinelerde yaygın olarak kullanılmaktadır. R'yi Windows sistemi üzerinde kullanabilmek uzman yardımı istemektedir. Bundan dolayı R, YALE ve WEKA'ya göre fazla tercih edilmemektedir.

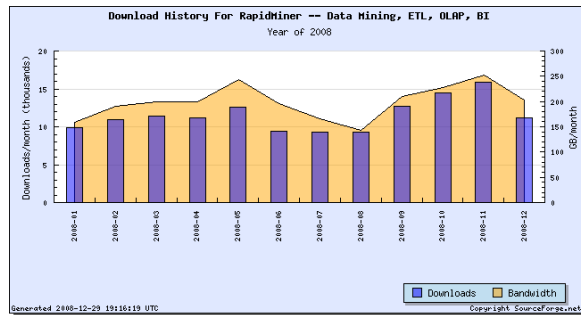


2007 yılında yapılan anket sonucunda Şekil 6’da verilen bilgiler elde edilmiştir. Bu anketin yapıldığı site Veri Madenciliği uzmanlarının ziyaret ettiği bir sitedir. Birinci çubuk, seçeneklerden sadece birinin seçildiği oyları temsil ederken ikinci çubuk birkaç seçeneğin seçildiği oyları temsil etmektedir.

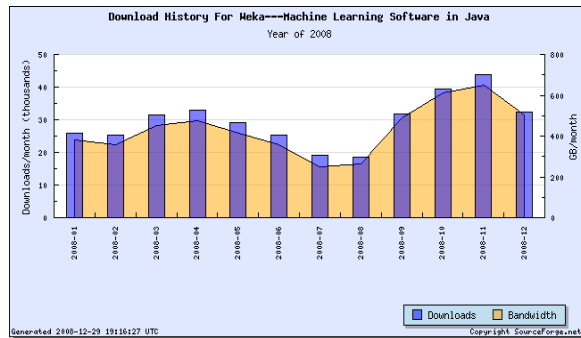


Şekil 6. Açık Kaynak Kodlu Veri Madenciliği Programları [7]

2008 yılında RapidMiner (YALE) (Şekil 7) ve WEKA (Şekil 8) için download grafikleri aşağıda verilmiştir. Grafiklerden de anlaşılacağı gibi WEKA, RapidMiner (YALE) ‘ye göre daha fazla download edilmiştir. Yukarıda ki istatistik ve aşağıdaki istatistikler arasındaki farklılığın sebebi, WEKA’ nın daha gözde olmasına rağmen uzmanlar arasında RapidMiner(YALE)’nin daha çok rağbet görmesidir.



Şekil 7. RapidMiner(YALE) için Download Sayıları [8]



Şekil 8. WEKA için Download Sayıları [9]

## 5. ÖRNEK UYGULAMA

### 5.1. Problem Tanımı

Uygulama, Gazi Üniversitesi Fen Bilimleri Enstitüsü verilerinden yararlanılarak gerçekleştirilmiştir. Bu çalışmada lisansüstü (Yüksek Lisans, Doktora) öğrencilerine ait 11809 adet veri kullanılmıştır. Öğrencilerin mezun olduğu lisans bölümüyle, devam ettiği lisansüstü bölümün aynı veya farklı olması durumunun öğrencinin başarısına etkisi araştırılmıştır. Lisans programını farklı bir üniversitede bitirmiş olan öğrenciler ile lisans programını Gazi Üniversitesinde bitirmiş olan öğrencilerin derslerdeki başarı oranları incelenmiştir. Ankara dışında ikamet eden öğrencilerle, Ankara içinde ikamet eden öğrenciler akademik başarı ölçüt alınarak kıyaslanmış olup bununla birlikte; lisansüstü derslerde doktora ve yüksek lisans öğrencilerinin, kız ve erkek öğrencilerin başarı notları kıyaslanmıştır. Yapılan çalışma sonucunda, lisansüstü programlara devam eden öğrencilerin farklı kriterlere göre başarısızlıkları ve bu başarısızlıkların nedenini bulup, çözümlenmek hedeflenmiştir. Bulunan sonuçlar, üniversite bünyesinde gerçekleştirilen Performans Programı, Akademik Değerlendirme ve Kalite Geliştirme dokümanlarını hazırlarken yardımcı olacaktır. Uygulama WEKA 3.5.8 programı yardımıyla gerçekleştirilmiştir.

### 5.2 Yapılan Çalışmada Veri Madenciliği Süreci

#### 5.2.1 Veri Temizleme

Veri temizleme, veri madenciliği sürecinin en kritik ve zaman alıcı adımıdır. Veri kümesinde bulunan notu niteliği genellikle yüksek oranlarda kayıp değer içermektedir. Bilimsel Hazırlık, Yüksek Lisans Tezi, Seminer, Doktora Tezi, Yeterlik Aşaması, Uzmanlık Alan Dersi’ne ait notlar veritabanında bulunmamaktadır. Bu tür kategorik verilerdeki kayıp değer problemi çözmek için, kayıp değeri bir değerle kodlama yöntemi kullanılır. Bu yöntemde göre, kayıp değerli alana “u” gibi bir değer atanır. Eğer bir niteliğe ait verilerin büyük bir çoğunluğu eksikse bu nitelik veri tabanından çıkarılmalıdır. Bir niteliği veri tabanından çıkarma kararı alırken sadece kayıp değerlerin toplam içindeki büyüklüğünü değil, aynı zamanda bu niteliğin kayıp değer içermesi nedeniyle dikkate almak gerekir. Nümerik nitelikler için kayıp değer sorununu çözmek biraz daha zordur. Çünkü çözmeye çalışmak bu niteliğe ait istatistikleri ve veri dağılımını değiştirebilir. Nümerik veri alanlarındaki kayıp değer sorununu çözmek için çeşitli yöntemler söz konusudur:

• Kayıp değerli alanlara, o niteliğe ait diğer değerlerin ortalaması atanır. Bu yöntem, basit olmakla birlikte verinin dağılımı üzerinde azımsanmayacak bir etkiye sahiptir. Bu nedenle, sadece verinin dağılımını minimal şekilde etkileyen durumlarda kullanılmalıdır.

• Mevcut değerler kullanılarak verinin dağılımı elde edilir ve kayıp değerli alanlara bu dağılıma uygun olarak değer atanır. Bu yöntem, verinin dağılımını çok fazla değiştirmez. Fakat kurulan modelde değer atanan değişken çok önemli ise veri madenciliği sonuçlarını etkileyecektir.

Kayıp değer problemini çözmek için “Structured Query Language” (SQL) komutları kullanılmıştır. Notu niteliğine ait verilerin (yukarıda belirlenen derslere ait) eksik olduğundan dolayı bu verilerin veritabanından çıkarılmasına karar verilmiştir. Nümerik kayıp değerler için yazılan SQL sorgusuyla yukarıda belirlenen dersler silinmiştir.

```
Delete from tablo_adi where dersad like '%Tez%' and dersad like '%Seminer%' dersad like '%Bilimsel Hazırlık%' dersad like '%Yeterlik Aşaması%' dersad like '%Uzmanlık Alan Dersi%'
```

Silinen satırlardaki derslerin kredisi 0 olduğundan dolayı ortalamayı etkilememektedir.

Mevcut kredisi olan derslerde ise boş olan notu alanına, o dönemdeki sınıf ortalaması alınarak veri düzeltme işlemi yapılmıştır.

```
Update tablo_adi set notu="Avg(notu)" where donem="Aynı_Donem" and dersadi="Aynı_dersadi"
```

Veri temizleme sonucunda, veritabanında 11809 adet veriden, 6341 adet veri kalmıştır. Niteliksiz verileri, veritabanından çıkararak, bulunacak sonuçların doğruluğu artırılmıştır.

## 5.2.2 Veri Dönüştürme

Veri temizlemeden sonraki adım veri dönüştürmedir. Bu veri dönüştürme işleminde uzman görüşü olarak nitelendirebileceğimiz Enstitü Sekreterinden bilgiler alınmış ve buna göre aşağıdaki dönüşümler elde edilmiştir.

1- SQL komutları ile notu alanı sayılarla derecelendirilmiştir.

```
Update tablo_adi set notu="5" where notu="AA" or notu="BA"
Update tablo_adi set notu="4" where notu="BA" or notu="CB"
Update tablo_adi set notu="3" where notu="CC" or notu="DC"
Update tablo_adi set notu="2" where notu="DD" or notu="FD"
Update tablo_adi set notu="1" where notu="FF" or notu="G" or notu="D"
```

2- Öğrencilerin lisans programından mezun olduğu üniversite, aşağıdaki sql komutları ile değiştirilmiştir.

```
Update tablo_adi set Mezun_Oldugu_Universite="aynı" where Mezun_Oldugu_Universite = "Gazi Üniversitesi"
Update tablo_adi set Mezun_Oldugu_Universite="farklı" where Mezun_Oldugu_Universite <> "Gazi Üniversitesi"
```

3- Öğrencilerin lisans programından mezun olduğu bölüm ve devam ettiği bölüm karşılaştırılarak, aşağıdaki sql komutları ile değiştirilmiştir.

```
Update tablo_adi set Mezun_Oldugu_Bolum="aynı" where Mezun_Oldugu_Bolum = Devam_Ettiği_Bolum
Update tablo_adi set Mezun_Oldugu_Bolum="farklı" where Mezun_Oldugu_Bolum <> Devam_Ettiği_Bolum
```

4- Öğrencilerin ikamet ettiği yeri tutan İkamet\_Ettiği\_İl alanı aşağıda gösterildiği gibi güncellenmiştir.

```
Update tablo_adi set İkamet_Ettiği_İl="1" where Adres_İl = "Ankara"
Update tablo_adi set İkamet_Ettiği_İl="0" where Adres_İl <> "Ankara"
```

Tablo 1’de veri dönüşümünden önce ve sonraki nitelik isimleri gösterilmektedir.

Tablo 1. Nitelik İsimleri

Nitelikler	
Veri Dönüşümünden Önce	Veri Dönüşümünden Sonra
Notu (AA, BA, BB, CB, CC, DC, DD, FD, G, D)	Notu (5, 4, 3, 2, 1)
Mezun Olduğu Üniversite (Abant İzzet Baysal, Afyon Kocatepe, Akdeniz, Anadolu, Ankara, Atatürk, Atılım, Balıkesir, Başkent, Celal Bayar, Cumhuriyet, Çanakkale 18 Mart, Çankaya, Çukurova, Dicle, Doğu Akdeniz, Dumlupınar, Ege, Erciyes, Fırat, Gazi, Gaziantep, Gaziosmanpaşa, Hacettepe, İnönü, İstanbul Teknik, İstanbul Ticareti İstanbul, İzmir Yüksek Teknoloji, Karadeniz Teknik, Kocaeli, Marmara, Mersin, Mustafa Kemal, Niğde, 19 Mayıs, Odtü, Osman Gazi, Sakarya, Selçuk, Süleyman Demirel, Trakya, Uludağ, Yıldız Teknik, Zonguldak Karaelmas)	Mezun Olduğu Üniversite (aynı, farklı)
Mezun Olduğu Bölüm (Biyoloji, Fizik, İstatistik, Kimya, Matematik, Elek. Elektronik Müh., Endüstri Müh., İnşaat Müh., Makine Müh., Mimarlık, Şehir ve Bölge Planlama, Kimya Müh., Makine Eğitimi, Metal Eğitimi, Yapı Eğitimi, Mobilya ve Dekorasyon Eğitimi, Elektrik Eğitimi, Endüstriyel Teknoloji Eğitimi, Trafik Planlaması ve Uygulaması, Kazaların Teknik ve Çevresel Araştırması, İleri Teknolojiler, Çevre Bilimleri, Orman Müh., Bilgisayar Müh.)	Mezun Olduğu Bölüm (aynı, farklı)
İkamet Ettiği İl (Adana, Adıyaman, Afyon, Amasya, Ankara, Antalya, Aydın, Balıkesir, Bolu, Bursa, Çankırı, Çorum, Denizli, Diyarbakır, Düzce, Elazığ, Eskişehir, Gaziantep, İçel, İstanbul, İzmir, Karabük, Kastamonu, Kayseri, Kırıkkale, Kırklareli, Kırşehir, Kocaeli, Konya, Kütahya, Malatya, Manisa, Muş, Nevşehir, Osmaniye, Sakarya, Samsun, Sivas, Tokat, Trabzon, Yozgat)	İkamet Ettiği İl (1, 0)

### 5.2.3 Modelleme

Veri temizleme ve dönüşümünden sonraki adım modelleme adımdır. Farklı modeller veri kümesi

üzerinde denenerek doğruluğu en yüksek olan model seçilir.



Tablo 2. Veritabanı İstatistikleri

Niteliği	Özelliği / Grubu	Kişi Sayısı
Cinsiyeti	E : Erkek	4016
	K : Kız	2325
Programı	DR : Doktora	2532
	YL : Yüksek Lisans	3809
Notu	5 : Çok iyi	3461
	4 : İyi	1783
	3 : Orta	520
	2 : Zayıf	163
	1 : Çok kötü	414
Mezun Olduğu Üniversite	Aynı : Aynı üniversiteden	3905
	Farklı : Farklı Üniversitede	2436
İkamet Ettiği İl	1 : Aynı ilde ikamet etmekte	5306
	0 : Farklı ilde ikamet etmekte	1035
Mezun Olduğu Bölüm	Aynı : Aynı bölümden	5253
	Farklı : Farklı bölümden	1088

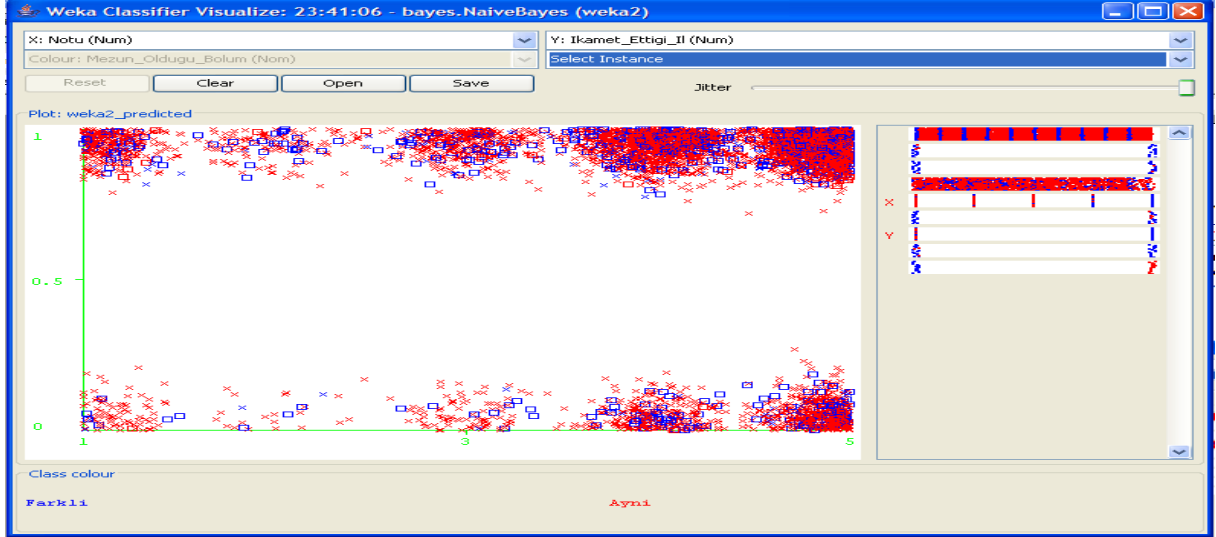
Belirlene hedeflere ulaşmak için kullanılan algoritmalar ve bu algoritmaların doğruluklarının karşılaştırılması Tablo 3'de gösterilmiştir.

Tablo 3. Sınıflandırma Algoritmaları ve Doğrulukları

ALGORİTMA	NAIVE BAYES	KSTAR	RBFNETWORK	J.48	JRIP	RIDOR
Doğru Olarak Sınıflandırılan Örnek Sayısı	5434	5352	5364	5253	5311	5271
Yanlış Olarak Sınıflandırılan Örnek Sayısı	907	989	977	1088	1030	1070
Kappa İstatistiği	0.367	0.1481	0.3537	0	0.0885	0.0294
Ortalama Mutlak Hata	0.2195	0.2467	0.2188	0.2843	0.2694	0.1687
Ortalama Hata Karakök	0.3337	0.3457	0.3476	0.377	0.3687	0.4108
Görelî Mutlak Hata	%77.1815	%86.7651	%76.943	%99.9734	%94.7277	%59.3417
Görelî Hata Karakök	%88.5109	%91.6863	%92.2014	%100	%97.7933	%108.9563

Tablo 3'deki değerler WEKA paket programı yardımıyla elde edilmiştir. WEKA paket programında veri kümesi için sırasıyla Naive Bayes, Kstar, RBFNetwork, J.48, JRIP, Ridor algoritmaları seçilerek program çalıştırılmış ve elde edilen sonuçlarla Tablo 3 hazırlanmıştır. Ayrıca HyperPipes, VFI gibi birçok algoritma denenmiştir.

Doğru olarak sınıflandırılan örnek sayısı 5000'den az olduğu için değerlendirmeye alınmamıştır. Tablo 3'den de görüldüğü gibi doğruluğu en yüksek olan sınıflandırma algoritması Naive Bayes olduğu için uygulamanın bu bölümünde Naive Bayes algoritması esas alınacaktır.



Şekil 9. Naive Bayes Sınıflandırıcı

Veri kümesine Naive Bayes algoritması uygulandığında Mezun\_Olduğu\_Bölüm alanı için Şekil 9 elde edilmektedir. Kırmızı işaretler mezun olduğu bölüm aynı olan öğrencileri, mavi ise mezun olduğu bölümü farklı olan öğrencileri göstermektedir. X ekseninde, 1,2,3,4,5 olmak üzere başarı notları, Y ekseninde ise 1,0 olmak üzere ikamet edilen il gösterilmektedir. Naive Bayes algoritmasının sonuçlarını aşağıdaki şekilde değerlendirmek mümkündür.

- Mezun olduğu bölümde lisansüstü eğitime devam eden öğrenciler ele alındığında, ikamet yeri ile lisansüstü eğitimi gördüğü yer aynı olan öğrencilerin başarısı, ikamet yerinin lisansüstü eğitimi gördüğü yerden farklı olan öğrencilerin başarısından daha fazladır. Görüldüğü gibi yukarıda bulunan kırmızı işaretler, aşağıda bulunanlara göre oldukça fazladır. 3,4,5 notlarına göre bu başarı beklendiği gibi olabilir. Ama 1,2 notlarında görülen olay beklenilmeyen bir durumdur. Şöyle ki; aynı yerde ikamet eden öğrencilerin yoğunluğu, farklı yerde ikamet eden öğrencilerin yoğunluğundan 1 notuna göre daha fazladır. Bilindiği gibi 1 notu, FF, Girmede, Devamsız notlarına karşılık gelmektedir. Bu durumun, aynı yerde ikamet eden öğrencilerin iş hayatlarındaki yoğunluklarından meydana geldiği düşünülmektedir. Bu sorunu gidermek için, bu öğrencilerin iş yerlerinden, üniversite tarafından 2 günlük izin alınarak bölümde çalışması desteklenebilir.

X ve Y eksenlerine veri kümesindeki diğer özellikler teker teker yerleştirildiğinde aşağıdaki sonuçlar elde edilmektedir.

- Mezun olduğu bölümde lisansüstü eğitime devam eden öğrenciler ele alındığında, doktora öğrencileri ve yüksek lisans öğrencilerinin, 3 notu hariç diğer başarı notları, ortalama olarak aynıdır. 3 notunda ise yüksek lisans öğrencilerin çoğunluğu fark edilmektedir. Bilindiği gibi 3 notu, CC ve DC notuna karşılık gelmektedir. Bunun sebebi olarak, Yüksek Lisans öğrencilerinin geçme notunun CC, doktora öğrencilerinin geçme notunun ise CB olmasıdır. Buradaki not kriteri başarıyı tetiklemiştir. Yüksek Lisans öğrencilerinin de geçme notu CB'ye yükseltilecek başarının yükselmesi sağlanabilir.
- Mezun olduğu bölüm dışında lisansüstü eğitime devam eden öğrenciler ele alındığında, doktora öğrencileri, yüksek lisans öğrencilerine göre başarı olarak büyük bir üstünlük sağlamaktadır. Bu üstünlüğün sebebi doktora öğrencilerinin genel olarak yüksek lisans öğrencilerinden daha fazla bilgi sahibi olması olarak açıklanabilir. Bu yüzden farklı bölümde lisansüstü eğitime devam etse bile başarısı etkilenmemektedir. Yüksek lisans öğrencilerinin bu zayıflığı, farklı bölümlerde lisansüstü eğitimi yapmak isteyen öğrencilerin zorunlu olarak alması gereken Bilimsel Hazırlık dersinin süresini uzatarak giderilebilir.
- Mezun olduğu üniversitede ve mezun olduğu üniversite dışında lisansüstü eğitime devam eden Doktora ve Yüksek lisans öğrencilerinin başarı notları ortalama olarak aynıdır. Bu durum şunu göstermektedir ki, farklı üniversiteden gelen öğrenciler yeni eğitim yerine hızlıca ayak uydurabilmektedir. Bu kısımda, mezun olduğu üniversitede lisansüstüne devam eden öğrencilerin daha başarılı olması beklense de, farklı üniversiteden

gelen öğrencilerin kaliteli olmasından dolayı bir eşitlik söz konusudur.

## 6. SONUÇLAR

Artan veri miktarından dolayı bilgiye ulaşamama sorunu neticesinde ortaya çıkan alan Veri Madenciliği olarak nitelendirilmektedir. Veri Madenciliği uygulamaları yapmak için bilgisayar programlarına ihtiyaç vardır. Bu programlar içerisinde veri kümeleme, karar ağaçları, bayes sınıflandırıcılar, apriori yöntemi gibi birçok algoritma mevcuttur. Algoritmalar sayesinde işlenen verilerden, bilgi çıkarımı yapılabilmektedir. Bu çalışmada Açık Kaynak Kodlu Veri Madenciliği programlarından RapidMiner(YALE), WEKA, R anlatılmış ve farkları üzerinde durulmuştur. WEKA'nın en çok kullanılan Veri Madenciliği programı olduğu görülmüştür. WEKA'da örnek bir uygulama sunulmuştur. Gerçekleştirilen uygulamadan elde edilen sonuçların lisansüstü eğitimi veren tüm Enstitülere yararlı olacağı düşünülmektedir.

## 7. KAYNAKLAR

1. Kudyba, S., "Managing Data Mining", CyberTech Publishing, 2004, 146-163.
2. Han, J. ve Kamber M., "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 2001.
3. Delen, D., Walker, G., Kadam, A., 'Predicting breast cancer survivability: a comparison of three data mining methods', Artificial Intelligence in Medicine, vol 34, June 2005, pp113-127
- 4  
<http://surfnet.dl.sourceforge.net/sourceforge/YALE/rapidminer-4.2-tutorial.pdf>
- 5- <http://cran.r-project.org/doc/manuals/R-intro.pdf>
- 6- Hania Gajewska, Mark S. Manasse and Joel McCormack, Why X Is Not Our Ideal Window System , Software — Practice & Experience vol 20, issue S2 (October 1990)
- 7- [http://www.kdnuggets.com/polls/2007/data\\_mining\\_software\\_tools.htm](http://www.kdnuggets.com/polls/2007/data_mining_software_tools.htm)
- 8-[http://sourceforge.net/project/stats/detail.php?group\\_id=5091&ugn=yale&type=prdownload&mode=year&package\\_id=0&release\\_id=0&file\\_id=0](http://sourceforge.net/project/stats/detail.php?group_id=5091&ugn=yale&type=prdownload&mode=year&package_id=0&release_id=0&file_id=0)
- 9-[http://sourceforge.net/project/stats/detail.php?group\\_id=5091&ugn=weka&type=prdownload&mode=year&package\\_id=0&release\\_id=0&file\\_id=0](http://sourceforge.net/project/stats/detail.php?group_id=5091&ugn=weka&type=prdownload&mode=year&package_id=0&release_id=0&file_id=0)