

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



UZAKTAN EĞİTİM SİSTEMLERİNDEKİ DERS
VERİLERİNE BİRLİKTELİK ANALİZİ

18011002 – Denizay Biberoğlu
20011626 – Yusuf Yemliha Çelik

BİLGİSAYAR PROJESİ

Danışman
Öğr. Gör. Dr. Ahmet Elbir

Aralık, 2022

TEŞEKKÜR

Öncelikle bize proje süreci boyunca her türlü desteği veren Dr. Ahmet ELBİR ve bu konuda çalışmalarıyla bilim dünyasına katkıda bulunmuş araştırmacılara, pandas kütüphanesini oluşturmuş ve büyümesine katkıda bulunan topluluk üyelerine, zaman zaman algoritma kısmında sorularımıza cevap veren Doç.Dr. Ayşe Betül OKTAY ve üniversite eğitim hayatımız boyunca bize doğrudan veya dolaylı katkılarda bulunmuş Bilgisayar Mühendisliği akademisyen üyeleri ve çalışanlarına, bize bu imkanı sağladığı için Yıldız Teknik Üniversitesi'ne teşekkür ederiz.

Denizay Biberoğlu
Yusuf Yemliha Çelik

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	viii
ÖZET	ix
ABSTRACT	x
1 Giriş	1
1.1 Birliktelik Analizi Nedir?	1
1.2 Birliktelik Analizinin Önemi	1
1.3 Birliktelik Analizi Nerede Kullanılır?	2
1.4 Birliktelik Analizi Sırasında Karşılaşılabilecek Zorluklar Nelerdir?	2
1.5 Birliktelik Analizi İçin Kullanılabilecek Yöntemler Nelerdir?	2
2 Ön İnceleme	3
2.1 Apriori Algoritması Nedir?	3
2.1.1 Örnek Üzerinden Anlatım	3
3 Fizibilite	8
3.1 Teknik Fizibilite	8
3.1.1 Donanım Fizibilitesi	8
3.1.2 Yazılım Fizibilitesi	8
3.2 İş Gücü ve Zaman Fizibilitesi	9
3.3 Yasal Fizibilite	9
3.4 Ekonomik Fizibilite	9
4 Sistem Analizi	10
4.1 Hedefler	10
4.2 Gereksinimler	10
4.2.1 Veri Ön İşleme	10

5	Sistem Tasarımı	14
5.1	Yazılım Tasarımı	14
5.1.1	Kullanılan Yazılım Dilleri	14
5.1.2	Kullanılan Kütüphaneler	14
5.1.3	Kullanılan Uygulamalar	14
5.1.4	Kullanılan Veri Seti	14
5.1.5	Girdiler ve Çıktılar	14
5.1.6	Ön İşleme	15
6	Uygulama	16
6.1	İstatistiksel Verilerin Çıkartılması	16
6.2	Birliktelik Analizi ve Birliktelik Kurallarının Çıkartılması	17
6.3	Uygulama	20
7	Deneyisel Sonuçlar	21
7.1	Mesleki Seçmeli Dersler	21
7.2	Sosyal Seçmeli Dersler	23
8	Sonuç	24
	Referanslar	25
	Özgeçmiş	26

KISALTMA LİSTESİ

AIS	Agrawal, Imielinski, Swami
SETM	Set Oriented Mining
UZEM	Uzaktan Eğitim Uygulama ve Araştırma Merkezi

ŞEKİL LİSTESİ

Şekil 3.1	Gantt Diyagramı	9
Şekil 4.1	Saf Veri	10
Şekil 4.2	Kısmen Temizlenmiş Veri	11
Şekil 4.3	Mesleki Seçmeli Dersler	12
Şekil 4.4	Sosyal Seçmeli Dersler	12
Şekil 4.5	Mesleki seçmeli dersler, "department_code" sütunu öğrencinin bulunduğu bölümü, "class_department_code" ise dersin alındığı bölümün kodunu gösteriyor.	12
Şekil 4.6	Bilgisayar Mühendisliği Bölümünün mesleki seçmeli dersleri için Oneshot tablosu.	13
Şekil 6.1	İstenilen bir bölüm için ders kayıtlarını sıralı bir şekilde oluşturan metod.	16
Şekil 6.2	Bilgisayar Mühendisliği bölümü için ders kayıt istatistiği.	16
Şekil 6.3	İstenilen bir bölüm için, farklı bölümlerden alınmış dersleri içeren kayıtları gösteren metod. Büyükten küçüğe olacak şekilde, hangi bölümlerden ders alındığını çıktı olarak veriyor.	17
Şekil 6.4	Bilgisayar Mühendisliği bölümü için farklı bölümlerden alınmış derslerin istatistiği.	17
Şekil 6.5	İstenilen bir bölüm veya ders için kural çıkartılması. İlk olarak sık öge kümeleri oluştur, daha sonra sık öge kümeleri kullanılarak kurallar oluşturulur. Kurallar önce Lift, daha sonra Support ve Confidence'a göre sıralanarak en iyi 10 kural çıktı olarak verilir.	18
Şekil 6.6	Bilgisayar Mühendisliği Bölümü için en yüksek Support ve Confidence değerlerine sahip 10 kural. Veri Madenciliği dersi 2 dönem de açıldığı için ve öğrenciler tarafından çok tercih edildiği için kurallarda yüksek sıralarda bulunduğu görülebilir.	18
Şekil 6.7	Bilgisayar Mühendisliği Bölümü, Doğal Dil İşleme dersi için birliktelik kuralları. Burda Doğal Dil İşleme dersi ile birlikte alınmış en çok 10 dersi görebilmekteyiz.	19
Şekil 6.8	Şekil 6.7'deki birliktelik kurallarının yuvarlak diyagram olarak gösterimi.	19

Şekil 6.9	Bilgisayar Mühendisliği Bölümü, Uzman Sistemlere Giriş dersi için uygulama üzerinden istatistik ve birliktelik kuralları gösterimi.	20
Şekil 7.1	Bilgisayar Mühendisliği bölümü Uzman Sistemlere Giriş dersi için birliktelik kuralları.	21
Şekil 7.2	Bilgisayar Mühendisliği bölümü İstatistiksel Veri Analizi dersi için birliktelik kuralları.	22
Şekil 7.3	Elektronik ve Haberleşme bölümü Güç Elektroniği dersi için birliktelik kuralları.	22
Şekil 7.4	Elektronik ve Haberleşme bölümü mesleki seçmeliler için birliktelik kuralları.	23
Şekil 7.5	Elektronik ve Haberleşme bölümü İşletme Bilimine Giriş dersi için birliktelik kuralları.	23

TABLO LİSTESİ

Tablo 1.1	Birliktelık Analizi Algoritmalarının Karşılaştırılması	2
Tablo 2.1	Market Fişlerinin İçeriği	3
Tablo 2.2	Ürünlerin Fişlerde olup olmama durumunu gösteren tablo	4
Tablo 2.3	Frekans ve Support Değerleri	4
Tablo 2.4	Minimum support değeri %40'dan fazla olan ürünlerin tablosu .	5
Tablo 2.5	Ürünlerin ikili kombinasyonu ve değerleri	5
Tablo 2.6	Ürünlerin üçlü kombinasyonundan sonra minimum support değerinin altındaki satırlar silindikten sonra elde edilen tablo. .	5
Tablo 2.7	Birliktelık kuralları	6
Tablo 2.8	Hesaplamalar	7

UZAKTAN EĞİTİM SİSTEMLERİNDEKİ DERS VERİLERİNE BİRLİKTELİK ANALİZİ

Denizay Biberoğlu

Yusuf Yemliha Çelik

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Öğr. Gör. Dr. Ahmet Elbir

Birliktelik analizi, verilerin birbirleriyle olan ilişkilerinin analiz edildiği veri madenciliği yöntemine denir. Birliktelik analizi elimizdeki bir veri seti içindeki ilişkileri analiz ederek belirli kurallar çıkarmak için kullanılır. Bu tespitlerin veya analizlerin ortaya çıkması kadar bunları etkili kullanıp fayda sağlamakta bir o kadar önemlidir. Bu analizleri etkili kullanabilmek için kullandığımız yöntemler ve bu yöntemlerin kullandığı kriterler önemlidir. Örneğin birliktelik analizi yapmak için kullanılan algoritmalarından birisi olan Apriori algoritması verilen veri setinden iyi kurallar üretmek için Support (destek), Confidence (güven) ve Lift (ilginçlik) ölçütlerine önem verir.

Proje kapsamında UZEM verilerinden öğrencilerin seçmiş oldukları dersler üzerinde birliktelik analizi yapılmış ve kayıtlar üzerinden belirli istatistiksel veriler elde edilmiştir. Proje süresince, başlangıçta veri incelenmiş, gereksiz veriler temizlenmiş ve gerekli olduğu durumlarda yeni veriler oluşturulmuştur. Daha sonra istenilen senaryolar için veri düzenlemiş ve belirli yazılım kütüphaneleri aracılığıyla birliktelik kuralları ve istatistiksel veriler oluşturulmuştur.

Proje sonucu olarak öğrencilere ders seçimi sürecince yardımcı olabilecek, her bölüm ve ders için istatistiksel verileri gösteren bir program ortaya çıkarılmıştır.

Anahtar Kelimeler: birliktelik analizi, apriori, veri madenciliği

ABSTRACT

ASSOCIATION ANALYSIS OF COURSE DATA IN ONLINE EDUCATION SYSTEMS

Denizay Biberoglu

Yusuf Yemliha Çelik

Department of Computer Engineering

Computer Project

Advisor: Assist. Prof. Dr. Ahmet Elbir

Association analysis is a data mining method that analyzes the relationships between data. Association analysis is used to extract certain rules by analyzing the relationships within a data set. As important as the emergence of these determinations or analyzes, it is just as important to use them effectively and benefit from them. The methods we use to use these analyzes effectively and the criteria used by these methods are important. For example, the Apriori algorithm, which is one of the algorithms used for association analysis, emphasizes Support, Confidence and Lift (interestingness) criteria to produce good rules from the given data set.

Within the scope of the project, association analysis was performed on the courses selected by students from UZEM data and certain statistical data were obtained from the records. During the project, the data was initially analyzed, unnecessary data was cleaned and new data was created when necessary. Then, the data was organized for the desired scenarios and association rules and statistical data were created through specific software libraries.

As a result of the project, a program that can help students in the course selection process and shows statistical data for each department and course has been created.

Keywords: association analysis, apriori, data mining

Veri madenciliği, verilerin farklı açılardan analiz edilmesi ve yararlı bilgiler halinde özetlenmesi sürecidir. Veri madenciliği, verileri analiz etmek için kullanılan analitik bir araçtır. Kullanıcıların verileri analiz etmesine, kategorize etmesine ve veriler arasındaki ilişkileri özetlemesine olanak tanır. Teknik olarak veri madenciliği, büyük ilişkisel veri tabanlarında korelasyonlar veya örüntüler bulma sürecidir. Anomali tespiti, kümeleme, birliktelik analizi, regresyon, özetleme, sınıflandırma vb. gibi bazı yaygın görevleri içerir [1]. Proje kapsamında veri madenciliği yöntemlerinden biri olan birliktelik analizi kullanılacaktır.

1.1 Birliktelik Analizi Nedir?

Birliktelik analizi, veritabanlarındaki öge kümeleri arasındaki ilginç korelasyonları, sık örüntüleri, ilişkileri veya rastlantısal yapıları çıkarmayı amaçlar. Birliktelik kuralları telekomünikasyon ağları, piyasa ve risk yönetimi, envanter kontrolü vb. gibi çeşitli alanlarda yaygın olarak kullanılmaktadır. Çeşitli birliktelik madenciliği teknikleri ve algoritmaları daha sonra kısaca tanıtılacak ve karşılaştırılacaktır.

Birliktelik analizi, belirli bir veri tabanından önceden tanımlanmış minimum destek (support) ve güveni (confidence) sağlayan birliktelik kurallarını bulmaktır. Problem genellikle iki alt probleme ayrıştırılır. Birincisi, veritabanında önceden tanımlanmış bir eşiğin üstünde bulunan öge kümelerini bulmaktır; bu öge kümelerine sık veya büyük öge kümeleri denir. İkinci problem ise bu büyük öge kümelerinden minimum güven (confidence) kısıtlamaları ile birliktelik kuralları oluşturmaktır [2].

1.2 Birliktelik Analizinin Önemi

Birliktelik analizi günlük hayatta birçok alanda kullanılabilir. Büyük veri setleri içindeki korelasyonları bulmak, bunlarla ilgili kurallar çıkarmak özellikle büyük market zincirleri, deprem analizi veya telekomünikasyon ağları için oldukça faydalı

olabilir.

Anlaşılması kolay olması, kullanışlılığı ve mümkün olan bütün örüntüleri ortaya çıkarması tekniği güçlü ve önemli hale getirmiştir. Ancak mümkün olan bütün olasılıkları ortaya çıkarması, daha sonra bu kuralların analiz edilmesi gerektiği için bir zayıflık haline de gelebilir.

1.3 Birliktelik Analizi Nerede Kullanılır?

Büyük market zincirleri tarafından kullanılan birliktelik analizi, müşterilerin hangi ürünleri beraber aldığını öğrenmek için kullanılır. Büyük market zincirleri daha sonra bu kuralları kullanarak birlikte alınan ürünler üzerinde kampanya getirerek satışlarını arttırmayı amaçlar.

Birliktelik analizinin daha birçok farklı kullanım alanı olabilir. Örneğin bir çalışmada Düzce ilçesinde yer alan 2112 binaya ait deprem riskinin analizi yapılmıştır [3].

1.4 Birliktelik Analizi Sırasında Karşılaşılabilecek Zorluklar Nelerdir?

Birliktelik analizinin iyi sonuç vermesi için analiz edilecek veri setinin çok küçük olmaması gerekmektedir. Sonuçta hangi öğelerin birlikte geçtiğini bulmaya çalıştığımız için, veri setimiz ne kadar büyük olursa algoritmamız o kadar iyi sonuç alır.

1.5 Birliktelik Analizi İçin Kullanılabilecek Yöntemler Nelerdir?

Birliktelik analizi için AIS Algoritması, SETM Algoritması, Apriori ve Aprioritid Algoritması gibi algoritmalar kullanılır. Proje kapsamında genel olarak Apriori algoritması kullanmaya karar verilmiştir.

Tablo 1.1 Birliktelik Analizi Algoritmalarının Karşılaştırılması

Özellikler	AIS	SETM	Apriori
Veri Desteği	Düşük	Düşük	Orta
İlk safha hızı	Yavaş	Yavaş	Hızlı
İleri safha hızı	Yavaş	Yavaş	Yavaş
Hassasiyet	Çok düşük	Düşük	Düşük

2 Ön İnceleme

Proje boyunca Apriori algoritması kullanılacağından Apriori algoritmasından biraz bahsedelim.

2.1 Apriori Algoritması Nedir?

Apriori Algoritmasının ismi, bilgileri bir önceki adımdan elde ettiği için “prior (önceki)” anlamında Apriori’dir. Bu algoritma temelinde iteratif (tekrarlayan) bir niteliğe sahiptir ve hareket bilgileri içeren veri tabanlarında sık geçen öge kümelerinin keşfedilmesinde kullanılır [4].

Birliktelik kuralı madenciliği, tüm sık geçen öğelerin bulunması ve sık geçen bu öğelerden güçlü birliktelik kurallarının üretilmesi olmak üzere iki aşamalıdır. Birliktelik kuralının ilk aşaması için kullanılan Apriori Algoritması, sık geçen öğeler madenciliğinde kullanılan en popüler ve klasik algoritmadır. Bu algoritmada özellikler ve veri, “boolean” ilişki kuralları ile değerlendirilir [4].

2.1.1 Örnek Üzerinden Anlatım

Farklı kategorilerde ürün satışının yapıldığı bir marketin satışlarına Apriori Algoritması uygulayıp birliktelik kuralları çıkartalım.

Tablo 2.1 Market Fişlerinin İçeriği

Fiş No	Detay
1	Ekmek, Süt, Çerez, Salça
2	Bira, Bebek Bezi, Ekmek, Yumurta
3	Bira, Kola, Bebek Bezi, Süt
4	Bira, Ekmek, Bebek Bezi, Süt, Çerez
5	Kola, Ekmek, Bebek Bezi, Süt
6	Bira, Ekmek, Bebek Bezi, Süt, Salça
7	Kola, Ekmek, Bebek Bezi, Süt

İlk olarak bu verilerin düzenlenip boolean biçimli bir tablo oluşturmamız gerekiyor. Tabloda, tüm satılan ürünlerin sütun olarak bulunmalı ve her ürünün bir fişte olup olmadığını göstermeliyiz.

Tablo 2.2 Ürünlerin Fişlerde olup olmama durumunu gösteren tablo

Fiş No	Ekmek	Süt	Çerez	Salça	Bira	Bebek Bezi	Yumurta	Kol
1	1	1	1	1	0	0	0	0
2	1	0	0	0	1	1	1	0
3	0	1	0	0	1	1	0	1
4	1	1	1	0	1	1	0	0
5	1	1	0	0	0	1	0	1
6	1	1	0	1	1	1	0	0
7	1	1	0	0	0	1	0	1

İlerlemeden önce frekans ve support kavramlarını açıklamamız gerekiyor.

Bir ürünün toplam satış miktarına frekans denilmektedir. Support ise frekans değerinin toplam girdi sayısına bölünmesi ile elde edilir. Yukarıdaki örnekte Ekmeğin frekansı 6, support'u ise $6/7$ 'dir.

İlerleyen kısımda minimum support olarak %40 değeri kullanılacaktır.

2.1.1.1 Birinci Adım

İlk olarak elimizdeki verilerde her ürünün support değerinin hesaplanmasıdır. Aşağıdaki tabloda sonuçlar bulunmaktadır:

Tablo 2.3 Frekans ve Support Değerleri

Ürün	Frekans	Support
Ekmek	6	0,857143
Süt	6	0,857143
Çerez	2	0,285714
Salça	2	0,285714
Bira	4	0,571429
Bebek Bezi	6	0,857143
Yumurta	1	0,142857
Kola	3	0,428571

Tablo 2.3'da her ürünün frekans ve support değerleri bulunmaktadır. Sonraki adımda ise minimum support değeri altındaki değerleri eleyeceğiz.

2.1.1.2 İkinci Adım

Minimum support değeri olan %40'ın altında support değeri olan ürünlerimizi tablodan çıkartalım.

Tablo 2.4 Minimum support değeri %40'dan fazla olan ürünlerin tablosu

Ürün	Frekans	Support
Ekmek	6	0,857143
Süt	6	0,857143
Bira	4	0,571429
Bebek Bezi	6	0,857143
Kola	3	0,428571

Elimizde kalan ürünlerin birlikte olma durumunu incelemek için yeni bir tablo oluşturacağız. Yeni tablo yukarıdaki ürünlerin ikili kombinasyonu olacağı için 10 adet satır içerecek.

Tablo 2.5 Ürünlerin ikili kombinasyonu ve değerleri

Ürün	Frekans	Support
Ekmek, Süt	5	0,71
Ekmek, Bira	3	0,42
Ekmek, Bebek Bezi	5	0,71
Ekmek, Kola	2	0,28
Süt, Bira	3	0,42
Süt, Bebek Bezi	5	0,71
Süt, Kola	3	0,42
Bira, Bebek Bezi	4	0,57
Bira, Kola	1	0,14
Bebek Bezi, Kola	3	0,42

İlk tabloda uyguladığımız işlemin aynısını uygulayarak support değeri %40'ın altındaki değerleri tablomuzdan kaldırıyoruz ve bu sefer üçlü, dörtlü... kombinasyonlar yaparak minimum support değerinin altındaki verileri sildikten sonra hiçbir satır kalmayana kadar devam ediyoruz.

Tablo 2.6 Ürünlerin üçlü kombinasyonundan sonra minimum support değerinin altındaki satırlar silindikten sonra elde edilen tablo.

Ürün	Frekans	Support
Ekmek, Süt, Bebek Bezi	4	0,57
Ekmek, Bira, Bebek Bezi	3	0,42
Süt, Bira, Bebek Bezi	3	0,42
Süt, Bebek Bezi, Kola	3	0,42

Malesef dörtlü kombinasyon yaptıktan sonra minimum support değerini karşılayan bir veri elde edemiyoruz. Bu yüzden üçlü kombinasyonda durmamız gerekiyor.

2.1.1.3 Üçüncü Adım

En son elimize geçen birliktelikler tablosundan kural çıkartma aşamasına geldik. "Ekmek, Süt, Bebek Bezi" birlikteliğinin kurallarını çıkartalım.

"Ekmek, Süt, Bebek Bezi" birlikteliğinden bu kümenin alt küme sayısı olan 8 adet kural çıkartabiliriz.

Tablo 2.7 Birliktelik kuralları

A		B
Ekmek, Süt, Bebek Bezi	=>	null
null	=>	Ekmek, Süt, Bebek Bezi
Ekmek	=>	Süt, Bebek Bezi
Süt, Bebek Bezi	=>	Ekmek
Ekmek, Bebek Bezi	=>	Süt
Süt	=>	Ekmek, Bebek Bezi
Ekmek, Süt	=>	Bebek Bezi
Bebek Bezi	=>	Ekmek, Süt

Kurallarımızı çıkardığımıza göre bunları ölçmek için bazı metrikleri hesaplamamız gerekiyor. Bu metrikler aşağıda verilmiştir:

- Confidence (Güven)
- Lift (İlginçlik)
- Conviction
- Leverage
- Coverage

Bu değerler arasında bizim için en önemli olan iki değer Confidence (Güven) ve Lift (İlginçlik).

$$Confidence(A,B) = \frac{support(A,B)}{support(A)}$$

Confidence değeri 0 ile 1 arasındadır. Değer 1'e ne kadar yakınsa kuralımızın o kadar güvenli ve doğru olduğunu gösterir

$$Lift(A,B) = \frac{support(A,B)}{support(A) * support(B)}$$

$$Conviction(A,B) = \frac{1 - support(B)}{1 - Confidence(A,B)}$$

$$Leverage(A,B) = support(A,B) - (support(A) * support(B))$$

$$Coverage(A,B) = support(A)$$

Tablo 2.8 Hesaplamalar

A => B	sup(A,B)	sup(A)	sup(B)	Conf	Lift	Conv	Lev	Cove
Ekmek => Süt, Bebek B.	0,57	0,86	0,71	0,67	0,93	0,86	-0,04	0,86
Süt, Bebek B. => Ekmek	0,57	0,71	0,86	0,80	0,93	0,71	-0,04	0,71
Ekmek, Bebek B. => Süt	0,57	0,71	0,86	0,80	0,93	0,71	-0,04	0,71
Süt => Ekmek, Bebek B.	0,57	0,86	0,71	0,67	0,93	0,86	-0,04	0,86
Ekmek, Süt => Bebek B.	0,57	0,71	0,86	0,80	0,93	0,71	-0,04	0,71
Bebek B. => Ekmek, Süt	0,57	0,86	0,71	0,67	0,93	0,86	-0,04	0,86

Yukarıdaki tablodan yapabileceğimiz çıkarımlar:

- "Süt ve Bebek Bezi" alanların "Ekmek" alma olasılığı %80'dir
- "Ekmek" alan müşterinin "Süt ve Bebek Bezi" ürünlerini de alma olasılığı %67'dir

Bu bölümde, projenin fizibilitesi hakkında net bilgiler edinmek için projenin çeşitli yönleri analiz edilmiştir.

3.1 Teknik Fizibilite

3.1.1 Donanım Fizibilitesi

Yapılan en ağır işlemler pandas kütüphanesi üzerinden veri temizleme işlemleri olması ve bu işlemler önceden yapıldığı için zaman kısıtı bulunmamaktadır. Bu nedenle yüksek donanımlara sahip bir sistem olması gerekmemektedir. İşlemlerin yapıldığı bilgisayarın donanım bilgisi ise aşağıdadır:

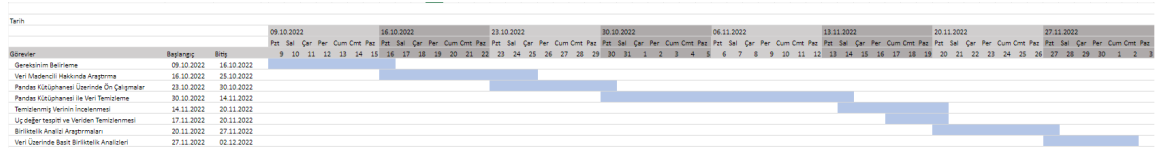
- İşlemci: AMD Ryzen 5 5600x
- Bellek: 16GB DDR4
- Sabit Sürücü: 512GB SSD
- Grafik Kartı: Nvidia Geforce RTX 2060

3.1.2 Yazılım Fizibilitesi

Kullanılan yazılımlar işletim sistemine bağlı olmadığı için belirli bir işletim sistemi belirlenmemiştir. Geliştirme süreci boyunca Python Notebook kullanılmıştır. Python dilinin kullanılma nedeni veri madenciliği konusunda dünya çapında en çok kullanılan kütüphanelerden biri olan pandas kütüphanesini içermesidir. Python'un Notebook şeklinde kullanılma sebebi ise veri madenciliği uygulamalarında bu şekilde kullanmanın çok daha verimli olmasıdır. Editör olarak VSCode kullanılmıştır.

3.2 İş Gücü ve Zaman Fizibilitesi

Projenin tasarım ve uygulama süreci Şekil 3.1'deki Gantt diyagramında belirtilmiştir.



Şekil 3.1 Gantt Diyagramı

3.3 Yasal Fizibilite

Projemiz mevcut kanun ve yönetmeliklere tamamen uygundur. Fikri ve sınai hiçbir hakkı ihmal etmemektedir. Sistemimizde kullanılacak olan yazılım ve programlardan hiçbiri özel lisans ve izin gerektirmemektedir. Ayrıca, Kişisel veriler kullanıcıların/kişilerin rızası olmadan kullanılmayacaktır.

3.4 Ekonomik Fizibilite

Geliştirme sürecinde kullanılan tüm programlama dilleri ve kütüphaneleri ücretsiz olduğundan ve proje sonucu kullanımı gerektirmediğinden proje kapsamında herhangi bir gider bulunmamaktadır.

4

Sistem Analizi

4.1 Hedefler

Uzaktan eğitim süreci boyunca ders kayıt verileri kullanılarak öğrencilerin seçmiş oldukları dersler arasında ilişkiler kurmak ve bu ilişkileri kullanarak öğrencilere ders önerisi yapılmasıdır. Aynı zamanda her bölüm için ders kayıtları ile ilgili istatistiklerin çıkartılıp bir GUI üzerinden kullanıcıya gösterilmesidir.

4.2 Gereksinimler

4.2.1 Veri Ön İşleme

Elimizdeki veri üzerinde çalışma yapmadan önce veride bazı ön işlemler yapılmıştır.

4.2.1.1 Verinin İncelenmesi

	period	stdno	Code	classname
0	GÜZ DÖNEMİ	10022902	ATA1031 - 1	Atatürk İlkeleri Ve İnkılap Tarihi 1
1	GÜZ DÖNEMİ	11071138	ATA1031 - 1	Atatürk İlkeleri Ve İnkılap Tarihi 1
2	GÜZ DÖNEMİ	12071011	ATA1031 - 1	Atatürk İlkeleri Ve İnkılap Tarihi 1
3	GÜZ DÖNEMİ	12071904	ATA1031 - 1	Atatürk İlkeleri Ve İnkılap Tarihi 1
4	GÜZ DÖNEMİ	13073017	ATA1031 - 1	Atatürk İlkeleri Ve İnkılap Tarihi 1
...
434677	BAHAR DÖNEMİ	20746011	YTT5116 - 1	Sosyal Pedogoji
434678	BAHAR DÖNEMİ	20746017	YTT5116 - 1	Sosyal Pedogoji
434679	BAHAR DÖNEMİ	207B6004	YTT5116 - 1	Sosyal Pedogoji
434680	BAHAR DÖNEMİ	207B6005	YTT5116 - 1	Sosyal Pedogoji
434681	BAHAR DÖNEMİ	207B6008	YTT5116 - 1	Sosyal Pedogoji

Şekil 4.1 Saf Veri

Elimizde bulunan veride toplam 434681 ders kaydının olduğu görülmüştür. Bu veri üzerinde çalışma yapabilmek için elimizdeki veriden yeni veriler üretme, gereksiz verilerin temizlenmesi işlemlerinin yapılmasına karar verilmiştir.

4.2.1.2 Verinin Temizlenmesi

Elimizdeki veride her bir kaydın hangi bölüme ait olduğu bilgisi bulunmadığı için bu bilgiyi elimizdeki veriden çıkarma kararı verilmiştir. Bunun için öğrenci numarasındaki bölüm numarası kısmının kullanılması kararlaştırılmıştır. Öğrenci numarasının 3, 4 ve 5. numaraları bölüm numarasını temsil ettiği için öğrenci numarasından bu sayılar çıkartılarak yeni bir sütun oluşturulacaktır.

Bu amaçla öğrenci numarasında harf bulunan veya öğrenci numarasındaki bölüm numarası kısmı, üniversitemizdeki 48 bölümden herhangi birini temsil etmeyen satırlar veriden silinmiştir.

	std_no	department_no	class_code	class_name	period
0	03071034	071	ATA1031	Atatürk İlkeleri Ve İnkılap Tarihi 1	GÜZ DÖNEMİ
1	09042008	042	ATA1031	Atatürk İlkeleri Ve İnkılap Tarihi 1	GÜZ DÖNEMİ
2	09065086	065	ATA1031	Atatürk İlkeleri Ve İnkılap Tarihi 1	GÜZ DÖNEMİ
3	08012020	012	ATA1031	Atatürk İlkeleri Ve İnkılap Tarihi 1	GÜZ DÖNEMİ
4	06071048	071	ATA1031	Atatürk İlkeleri Ve İnkılap Tarihi 1	GÜZ DÖNEMİ
...
256182	21097003	097	TRO4402	Dil Ve Kültür	BAHAR DÖNEMİ
256183	22084037	084	TRO4402	Dil Ve Kültür	BAHAR DÖNEMİ
256184	16031065	031	TRO4512	Çağdaş Türk Edebiyatı	BAHAR DÖNEMİ
256185	21097003	097	TRO4512	Çağdaş Türk Edebiyatı	BAHAR DÖNEMİ
256186	22084037	084	TRO4512	Çağdaş Türk Edebiyatı	BAHAR DÖNEMİ

Şekil 4.2 Kısmen Temizlenmiş Veri

Daha sonra üniversitemizdeki 48 bölüm tarafından verilen tüm mesleki ve sosyal seçmeli derslerin listesi çıkarılmış, verimizdeki "class_code" sütunu bu liste içinde olmayan satırlar silinmiştir. Bu sayede elimizde mesleki ve sosyal seçmeli ders kayıtları kalmıştır.

	std_no	department_no	department_code	class_department_code	class_code	class_name	period	cnt
0	08011013	011	BLM	BLM	BLM3120	Bilgiye Erişim Ve Arama Motorları	GÜZ DÖNEMİ	1369
1	09011033	011	BLM	BLM	BLM3120	Bilgiye Erişim Ve Arama Motorları	GÜZ DÖNEMİ	1369
2	08011057	011	BLM	BLM	BLM3580	Sistem Programlama	GÜZ DÖNEMİ	1369
3	08011060	011	BLM	BLM	BLM3590	İstatistiksel Veri Analizi	GÜZ DÖNEMİ	1369
4	09011033	011	BLM	BLM	BLM3720	Bilgisayar Grafiğine Giriş	GÜZ DÖNEMİ	1369
...
31469	18027603	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31470	18027604	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31471	18027613	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31472	19027601	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31473	19027605	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590

Şekil 4.3 Mesleki Seçmeli Dersler

	std_no	department_no	department_code	class_department_code	class_code	class_name	period	cnt
0	08033065	033	ISL	IKT	IKT1121	Sosyoloji	GÜZ DÖNEMİ	458
1	08012037	012	ELM	ISL	ISL1611	İşletme Bilimine Giriş	GÜZ DÖNEMİ	759
2	06014037	014	EHM	ISL	ISL3621	Üretim Yönetimi	GÜZ DÖNEMİ	576
3	01071116	071	MİM	ITB	ITB3010	Sosyoloji	GÜZ DÖNEMİ	784
4	09011033	011	BLM	ITB	ITB3020	Felsefeye Giriş	GÜZ DÖNEMİ	489
...
9559	19087604	087	SYP	SYP	SYP2192	Kültür Yönetimi Ve Etkenleri 2	BAHAR DÖNEMİ	107
9560	19087607	087	SYP	SYP	SYP2192	Kültür Yönetimi Ve Etkenleri 2	BAHAR DÖNEMİ	107
9561	19087608	087	SYP	SYP	SYP2192	Kültür Yönetimi Ve Etkenleri 2	BAHAR DÖNEMİ	107
9562	19087609	087	SYP	SYP	SYP2192	Kültür Yönetimi Ve Etkenleri 2	BAHAR DÖNEMİ	107
9563	20087601	087	SYP	SYP	SYP2192	Kültür Yönetimi Ve Etkenleri 2	BAHAR DÖNEMİ	107

Şekil 4.4 Sosyal Seçmeli Dersler

4.2.1.3 Yeni Veriler Üretme

Veri temizleme ve analiz işlemlerinde kullanılmak üzere her kaydın bulunduğu bölüm kodu ve numarasına ihtiyacımız bulunmaktadır. Bunun için temizlenmiş veri üzerinde öğrenci numarasının 3, 4 ve 5. sayıları kullanılarak her kaydın bulunduğu bölüm kodu çıkartılmıştır. Ayrıca öğrenciler başka bölümlerden de ders seçebildiği için, alınan dersin de bölüm konu ders kodundan çıkartılmıştır.

	std_no	department_no	department_code	class_department_code	class_code	class_name	period	cnt
0	08011013	011	BLM	BLM	BLM3120	Bilgiye Erişim Ve Arama Motorları	GÜZ DÖNEMİ	1369
1	09011033	011	BLM	BLM	BLM3120	Bilgiye Erişim Ve Arama Motorları	GÜZ DÖNEMİ	1369
2	08011057	011	BLM	BLM	BLM3580	Sistem Programlama	GÜZ DÖNEMİ	1369
3	08011060	011	BLM	BLM	BLM3590	İstatistiksel Veri Analizi	GÜZ DÖNEMİ	1369
4	09011033	011	BLM	BLM	BLM3720	Bilgisayar Grafiğine Giriş	GÜZ DÖNEMİ	1369
...
31469	18027603	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31470	18027604	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31471	18027613	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31472	19027601	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590
31473	19027605	027	TDE	TDE	TDE4372	Klasik Türk Edebiyatında Mazmunlar	BAHAR DÖNEMİ	590

Şekil 4.5 Mesleki seçmeli dersler, "department_code" sütunu öğrencinin bulunduğu bölümü, "class_department_code" ise dersin alındığı bölümün kodunu gösteriyor.

4.2.1.4 Verinin Dönüştürülmesi

Veriyi Apriori algoritmasına beslemek için veriyi "Oneshot" şekline dönüştürmeliyiz. Öğrenci numaralarının satır, ders kodlarının da sütun olduğu, her bir satır için alınmış olan dersler için 1 değeri, alınmamış olan dersler için 0 değeri içeren bir tablo oluşturmamız gerekiyor. Bunu TransactionEncoder methodu ile gerçekleştirip Oneshot Tablomuzu oluşturabiliyoruz.

	Aynı Olay Simülasyonu	Bilgisayar Grafisine Giriş	Bilgiye Erişim Ve Arama Motorları	Biyoenformatiğe Giriş	Büyük Veri İşleme Ve Analizi	Doğal Dil İşlemeye Giriş	Görüntü İşleme	Mobil Programlamaya Giriş	Robot Teknolojisine Giriş	Sayısal İşaret İşleme	Sistem Programlama	Uzman Sistemlere Giriş	Madencilik Giriş	Veri Gerçekleştirilmesi	Veritabanı Sistemlerinin Gerçekleştirilmesi	Yapay Sinir Ağlarına Giriş	Yapay Zeka	Yazılım Kalite ve Test Süreci	Yönetim Bilgi Sistemleri	İleri Ağ Programlama	İstatistiksel Veri Analizi
0	False	False	False	False	True	False	False	False	False	True	False	True	True	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	True	False	False
2	False	False	True	False	False	False	False	False	True	True	False	False	False	False	False	False	False	False	True	False	False
3	False	False	False	False	False	False	False	False	False	True	True	True	False	False	False	False	False	True	True	False	False
4	True	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	True
...
332	False	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False	False	False
333	False	False	False	False	False	False	False	False	True	True	True	True	False	False	False	False	False	False	False	True	False
334	True	True	False	True	False	False	False	False	True	False	False	False	True	True	True	False	False	False	False	False	False
335	True	False	False	False	False	False	False	False	True	True	False	False	False	False	False	False	False	True	False	False	True
336	False	True	False	True	False	False	False	False	True	False	False	True	False	False	False	False	True	True	False	False	False

Şekil 4.6 Bilgisayar Mühendisliği Bölümünün mesleki seçmeli dersleri için Oneshot tablosu.

5.1 Yazılım Tasarımı

5.1.1 Kullanılan Yazılım Dilleri

- Python3

5.1.2 Kullanılan Kütüphaneler

- Pandas
- Numpy
- matplotlib

5.1.3 Kullanılan Uygulamalar

- Jupyter-Notebook
- VSCode

5.1.4 Kullanılan Veri Seti

- Okul tarafından temin edilen Uzaktan Eğitim Ders Kayıt Verileri

5.1.5 Girdiler ve Çıktılar

5.1.5.1 Girdiler

- Okul tarafından temin edilmiş olan UZEM Ders kayıt verileri

5.1.5.2 Çıktılar

- Seçilen bölüm ve dersler için istatistikler ve birliktelik kurallarını gösteren, GUI içeren uygulama.

5.1.6 Ön İşleme

Verilerin Ön işlenmesi sırasında aşağıda belirtilen yöntemler kullanılmıştır:

- Gereksiz verilerin temizlenmesi
- Verinin dönüştürülmesi

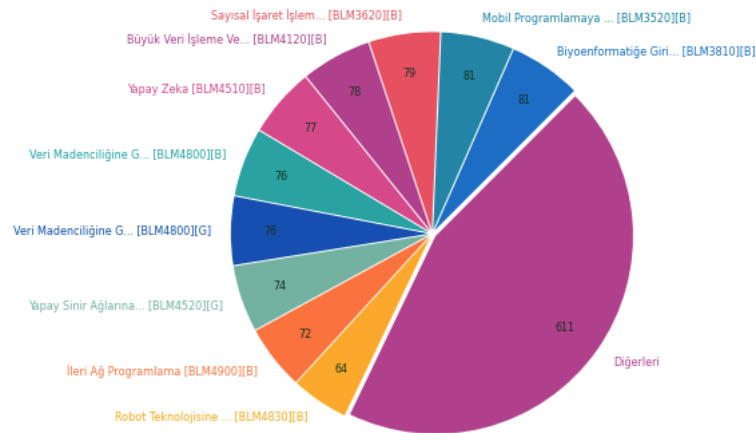
Bu bölümde uygulamanın içeriği ve çıktıları bulunmaktadır.

6.1 İstatistiksel Verilerin Çıkartılması

İstatistiksel veri olarak her bölümdeki kayıtların analiz edilmesi ve her bölüm öğrencilerinin başka bölümlerden aldığı derslerin analiz edilmesi bulunmaktadır. Aşağıdaki kod parçaları ile bu istatistiksel veriler çıkartılmaktadır.

```
def class_enrollment_cnt(dep_code, df_in):
    """verilen bölüm kodunun ders kayıtlarının büyükten küçüğe doğru sıralı bir şekilde
    her ders için kaç kayıt olduğunu döndürür"""
    df = df_in[df_in.department_code == dep_code]
    df = df[['department_no', 'department_code', 'class_department_code', 'class_code', 'class_name', 'period']]
    df['cnt'] = df.groupby(['class_code', 'period'])['class_code'].transform('count')
    df = df.drop_duplicates(subset=df.columns.difference(['class_name']))
    df = df.sort_values('cnt', ascending=False).reset_index().drop('index', axis=1)
    return df
```

Şekil 6.1 İstenilen bir bölüm için ders kayıtlarını sıralı bir şekilde oluşturan metod.



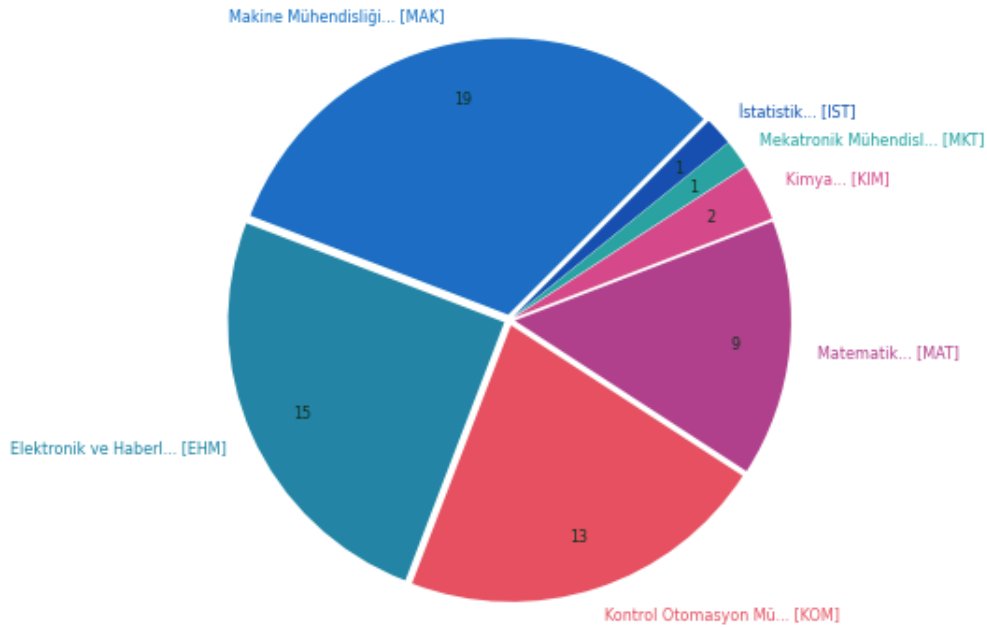
Şekil 6.2 Bilgisayar Mühendisliği bölümü için ders kayıt istatistiği.

```
def different_dep_class_enrollment_cnt_2(df_in):
    dep_codes = df_data['department_code'].unique()
    df2 = pd.DataFrame()
    for dep in dep_codes:
        df = df_in[df_in.department_code == dep]
        df = df[df.department_code != df.class_department_code]
        df['to_drop'] = df.apply(lambda row:
                                row.class_code in social_elective_courses[social_elective_courses.department_code == row.department_code].reset_index().at[0, 'social_elective_courses'] or
                                row.class_code in vocational_elective_courses[vocational_elective_courses.department_code == row.department_code].reset_index().at[0, 'vocational_elective_courses'] or
                                row.class_code in mandatory_courses[mandatory_courses.department_code == row.department_code].reset_index().at[0, 'mandatory_courses'], axis=1)
        df = df[df.to_drop == False].reset_index().drop(['index', 'to_drop'], axis=1)

        df = df[['department_no', 'department_code', 'class_department_code', 'class_code', 'class_name']]
        df['cnt'] = df.groupby('class_department_code')['class_department_code'].transform('count')
        df = df[['department_code', 'class_department_code', 'cnt']]
        df = df.drop_duplicates()
        df = df[df['class_department_code'] != 'ITB']
        df = df.sort_values('cnt', ascending=False).reset_index().drop('index', axis=1)
        df2 = pd.concat([df2, df])

    df2 = df2.reset_index().drop('index', axis=1)
    return df2
```

Şekil 6.3 İstenilen bir bölüm için, farklı bölümlerden alınmış dersleri içeren kayıtları gösteren metod. Büyükten küçüğe olacak şekilde, hangi bölümlerden ders alındığını çıktı olarak veriyor.



Şekil 6.4 Bilgisayar Mühendisliği bölümü için farklı bölümlerden alınmış derslerin istatistiği.

6.2 Birliktelik Analizi ve Birliktelik Kurallarının Çıkartılması

Birliktelik kurallarının oluşturulması hem bölüm hem de ders bazından gerçekleştirilebilmektedir. Bölüm bazında bölüm içindeki en yüksek Support ve Confidence'a sahip 10 kural, ders bazında ise o dersin kuralın sol kısmında (antecedents) bulunduğu kurallar tablo ve yuvarlak diyagram olarak çıkartılmaktadır.

```

def create_frequent_itemsets(dep_str, df_in):
    """verilen bölümün kayıtlarına göre frequent itemsets oluşturur"""
    dep_onehot_df = oneshot_df(dep_str, df_in)
    dep_freq_itemsets = apriori(dep_onehot_df, min_support = 0.02, use_colnames=True)
    return dep_freq_itemsets
✓ 0.2s

def create_rules(dep_str, df_in):
    """verilen bölüm ve verilen ders için öneri yapar"""
    dep_freq_itemsets = create_frequent_itemsets(dep_str, df_in)
    rules = association_rules(dep_freq_itemsets, metric="lift", min_threshold=1)
    rules = rules.sort_values('confidence', ascending=False)
    rules = rules.reset_index().drop('index', axis=1)

    _, cnt = dep_records_by_code(dep_str, df_in)
    rules['cnt'] = rules.apply(lambda row: round(cnt * row["support"]), axis=1)
    rules["antecedents"] = rules["antecedents"].apply(lambda x: ', '.join(map(str, x)))
    rules["consequents"] = rules["consequents"].apply(lambda x: ', '.join(map(str, x)))
    rules = rules[rules["antecedents"].str.contains(',') == False]
    rules = rules[rules["consequents"].str.contains(',') == False]
    rules['antecedent support'] = rules['antecedent support'].apply(lambda x: round(x, 2))
    rules['consequent support'] = rules['consequent support'].apply(lambda x: round(x, 2))
    rules['support'] = rules['support'].apply(lambda x: round(x, 2))
    rules['confidence'] = rules['confidence'].apply(lambda x: round(x, 2))
    rules['lift'] = rules['lift'].apply(lambda x: round(x, 2))
    return rules
✓ 0.3s

def rules10(dep_str, df, antecedents=""):
    """bölüm için veya ders için top10 kuralı döndürür\n
    antecedents girilmezse bölüm, girilirse girilen antecedents için oluşturulu\n
    antecedents list şeklinde olmalıdır. Örn: ['Veri Madenciliğine Giriş']"""
    df = create_rules(dep_str, df)
    if len(antecedents) > 0:
        df['to_drop'] = df.apply(lambda row: row.antecedents.find(antecedents), axis=1)
        df = df[df['to_drop'] == False].drop('to_drop', axis=1)
    df = df.sort_values('support', ascending=False)
    df = df.head(10)
    df = df.sort_values('confidence', ascending=False)
    df = df.reset_index().drop('index', axis=1)

    df[['Antecedents', 'Consequents', 'Antec. Sup.', 'Conseq. Sup.', 'Support', 'Confidence', 'Count']] \
    = df[['antecedents', 'consequents', 'antecedent support', 'consequent support', 'support', 'confidence', 'cnt']]
    return df[['Antecedents', 'Consequents', 'Antec. Sup.', 'Conseq. Sup.', 'Support', 'Confidence', 'Count']]

```

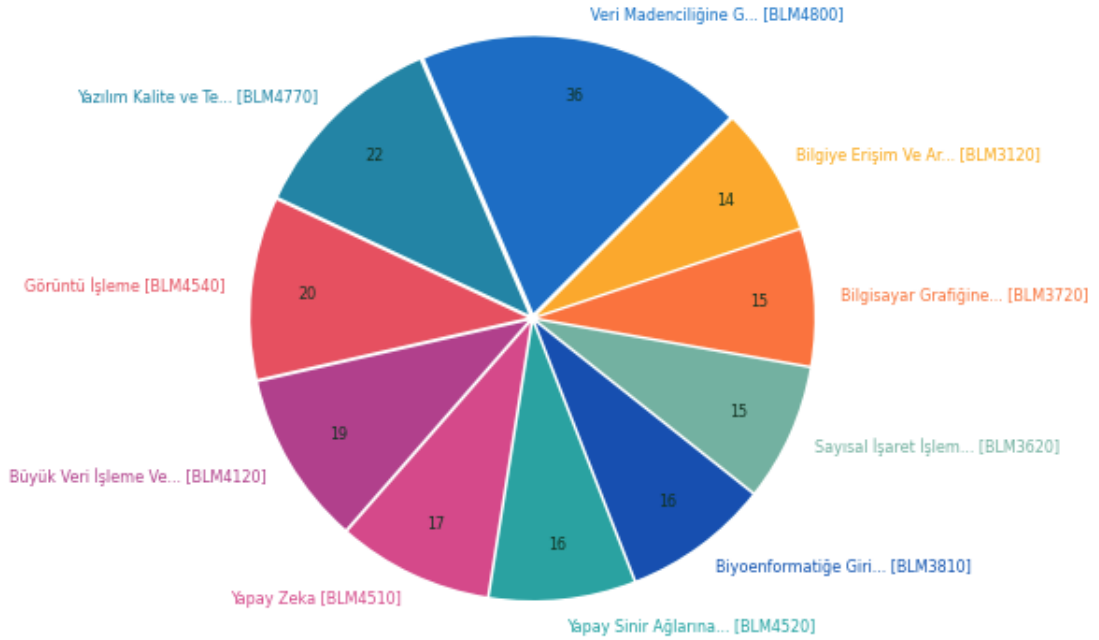
Şekil 6.5 İstenilen bir bölüm veya ders için kural çıkartılması. İlk olarak sık öğe kümeleri oluştur, daha sonra sık öğe kümeleri kullanılarak kurallar oluşturulur. Kurallar önce Lift, daha sonra Support ve Confidence'a göre sıralanarak en iyi 10 kural çıktı olarak verilir.

	Antecedents	Consequents	Antec. Sup.	Conseq. Sup.	Support	Confidence	Count
0	Büyük Veri İşleme Ve Analizi	Veri Madenciliğine Giriş	0.23	0.45	0.15	0.63	49
1	Doğal Dil İşlemeye Giriş	Veri Madenciliğine Giriş	0.19	0.45	0.11	0.57	36
2	Biyoenformatiğe Giriş	Veri Madenciliğine Giriş	0.24	0.45	0.12	0.49	40
3	Mobil Programlamaya Giriş	Veri Madenciliğine Giriş	0.24	0.45	0.12	0.48	39
4	Yapay Zeka	Büyük Veri İşleme Ve Analizi	0.22	0.23	0.11	0.48	36
5	Veri Madenciliğine Giriş	Büyük Veri İşleme Ve Analizi	0.45	0.23	0.15	0.32	49
6	Veri Madenciliğine Giriş	Biyoenformatiğe Giriş	0.45	0.24	0.12	0.26	40
7	Veri Madenciliğine Giriş	Mobil Programlamaya Giriş	0.45	0.24	0.12	0.26	39
8	Veri Madenciliğine Giriş	Yapay Zeka	0.45	0.22	0.11	0.25	38
9	Veri Madenciliğine Giriş	Sayısal İşaret İşleme	0.45	0.23	0.11	0.24	36

Şekil 6.6 Bilgisayar Mühendisliği Bölümü için en yüksek Support ve Confidence değerlerine sahip 10 kural. Veri Madenciliği dersi 2 dönem de açıldığı için ve öğrenciler tarafından çok tercih edildiği için kurallarda yüksek sıralarda bulunduğu görülebilir.

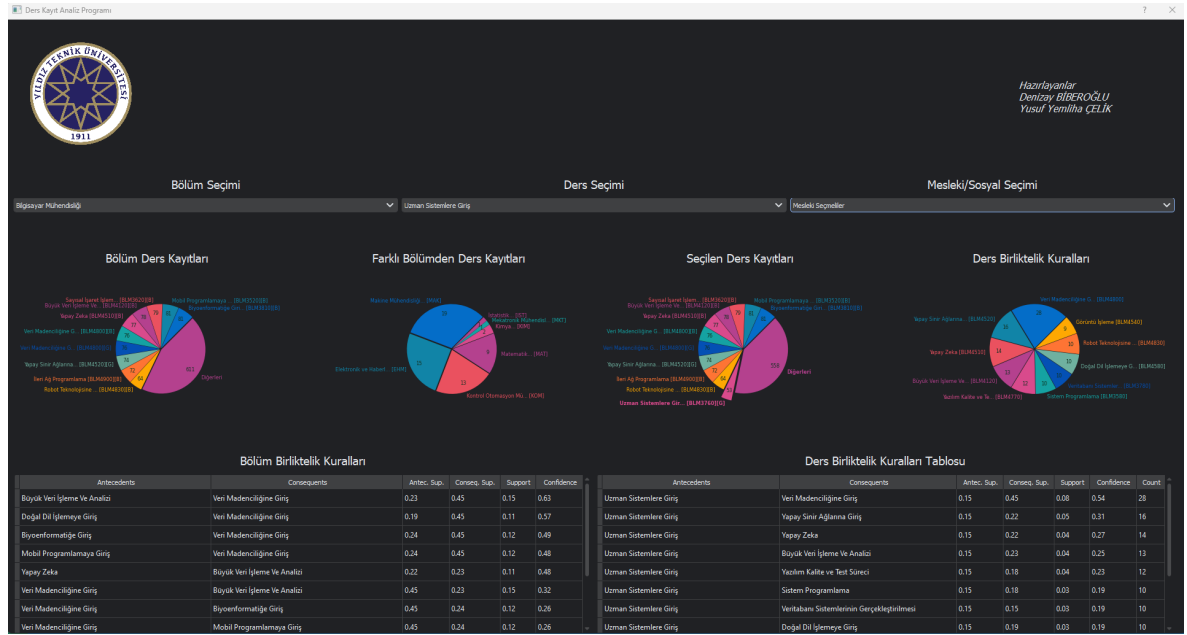
	Antecedents	Consequents	Antec. Sup.	Conseq. Sup.	Support	Confidence	Count
0	Doğal Dil İşlemeye Giriş	Veri Madenciliğine Giriş	0.19	0.45	0.11	0.57	36
1	Doğal Dil İşlemeye Giriş	Yazılım Kalite ve Test Süreci	0.19	0.18	0.07	0.35	22
2	Doğal Dil İşlemeye Giriş	Görüntü İşleme	0.19	0.16	0.06	0.32	20
3	Doğal Dil İşlemeye Giriş	Büyük Veri İşleme Ve Analizi	0.19	0.23	0.06	0.30	19
4	Doğal Dil İşlemeye Giriş	Yapay Zeka	0.19	0.22	0.05	0.27	17
5	Doğal Dil İşlemeye Giriş	Yapay Sinir Ağlarına Giriş	0.19	0.22	0.05	0.25	16
6	Doğal Dil İşlemeye Giriş	Biyoenformatiğe Giriş	0.19	0.24	0.05	0.25	16
7	Doğal Dil İşlemeye Giriş	Sayısal İşaret İşleme	0.19	0.23	0.04	0.24	15
8	Doğal Dil İşlemeye Giriş	Bilgisayar Grafiğine Giriş	0.19	0.18	0.04	0.24	15
9	Doğal Dil İşlemeye Giriş	Bilgiye Erişim Ve Arama Motorları	0.19	0.18	0.04	0.22	14

Şekil 6.7 Bilgisayar Mühendisliği Bölümü, Doğal Dil İşleme dersi için birliktelik kuralları. Burda Doğal Dil İşleme dersi ile birlikte alınmış en çok 10 dersi görebilmekteyiz.



Şekil 6.8 Şekil 6.7'deki birliktelik kurallarının yuvarlak diyagram olarak gösterimi.

6.3 Uygulama



Şekil 6.9 Bilgisayar Mühendisliği Bölümü, Uzman Sistemlere Giriş dersi için uygulama üzerinden istatistik ve birliktelik kuralları gösterimi.

Projemiz sonunda ortaya çıkan uygulama Şekil 6.9'da görülmektedir. Kullanıcı, "Bölüm Seçimi" kısmından okulumuzdaki herhangi bir bölümü seçebilir. Bölüm seçtikten sonra o bölümün en çok seçilen dersleri (seçime bağlı olarak mesleki seçmeli veya sosyal seçmeli) ve o bölümün öğrencilerinin en çok diğer hangi bölümlerden ders seçimi yaptığını gösteren iki adet yuvarlak diyagram, alt kısımda da bölüme ait en yüksek Confidence değerine sahip 10 birliktelik kuralı görülmektedir.

Bölüm seçtikten sonra bölüm seçme kısmının bir sağındaki "Ders Seçimi" kısmı aktif hale gelmektedir. Bu kısımdan kullanıcı seçtiği bölümdeki herhangi bir derse ait birliktelik kurallarını görüntüleyebilmektedir.

En sağda ise "Mesleki/Sosyal Seçimi" kısmı bulunmaktadır. Burdan kullanıcı mesleki seçmeli dersler için mi yoksa sosyal seçmeli dersler için mi istatistik ve birliktelik kurallarının görüntülenmesini istediğini seçebilmektedir.

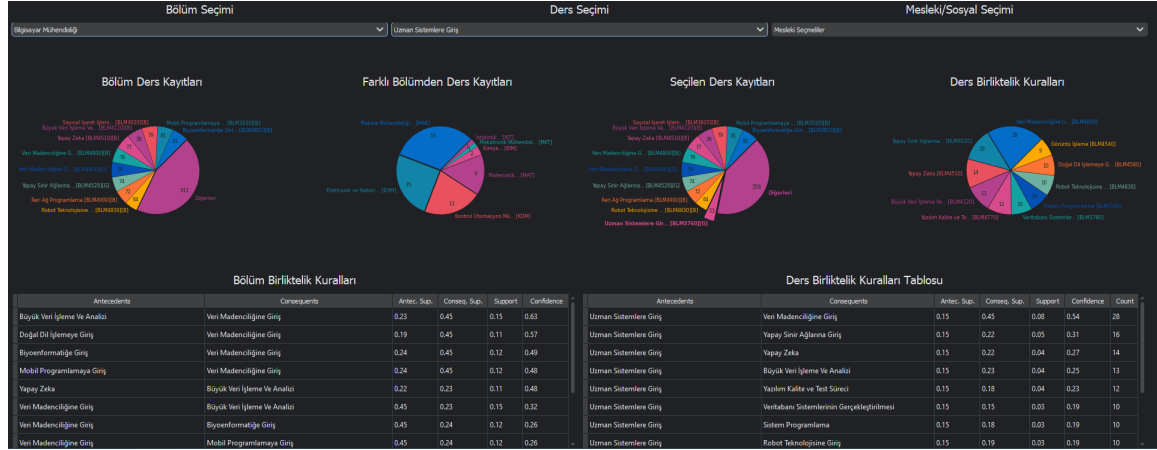
7

Deneyisel Sonuçlar

Bu bölümde GUI üzerinden çeşitli derslerin birliktelik kuralları çıkartılacak ve yorumlanacaktır.

Tüm kurallar önce Support daha sonra Confidence'a göre sıralandıktan sonra ilk 10 kural uygulamada gösterilmekte ve görselleştirilmektedir. Support ve Confidence değerleri raporda daha önce de bahsettiğimiz gibi, birliktelik kurallarının en önemli 2 metriği olduğu için bu değerlere göre sıralanmıştır.

7.1 Mesleki Seçmeli Dersler

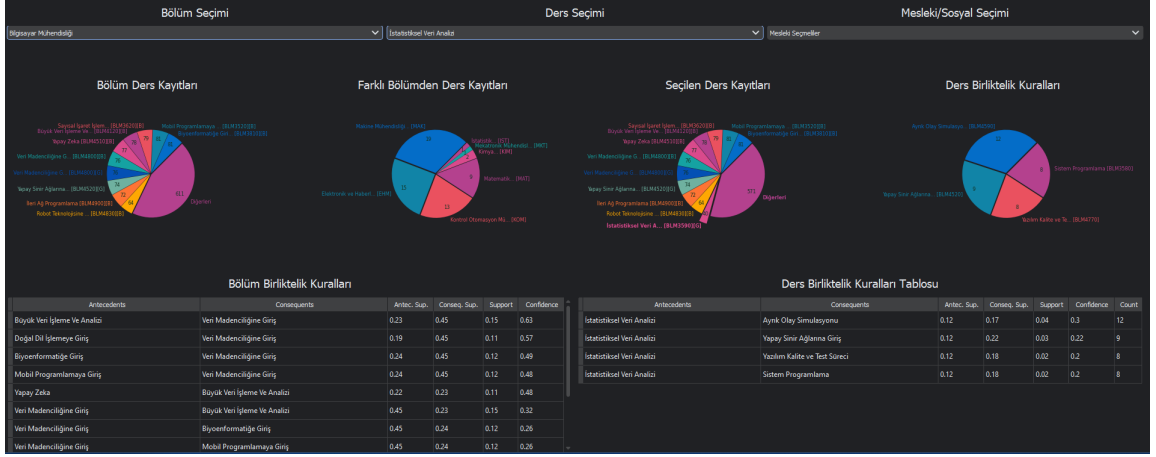


Şekil 7.1 Bilgisayar Mühendisliği bölümü Uzman Sistemlere Giriş dersi için birliktelik kuralları.

Resimde de görülebileceği üzere Uzman Sistemlere Giriş ile birlikte alınan en çok dersin Veri Madenciliğine Giriş olduğunu görüyoruz. Veri Madenciliğine Giriş dersi 2 dönem de açılığı için ve öğrenciler tarafından çok tercih edilen bir ders olduğu için çoğu dersin birliktelik kurallarında en üst sırada yer alıyor.

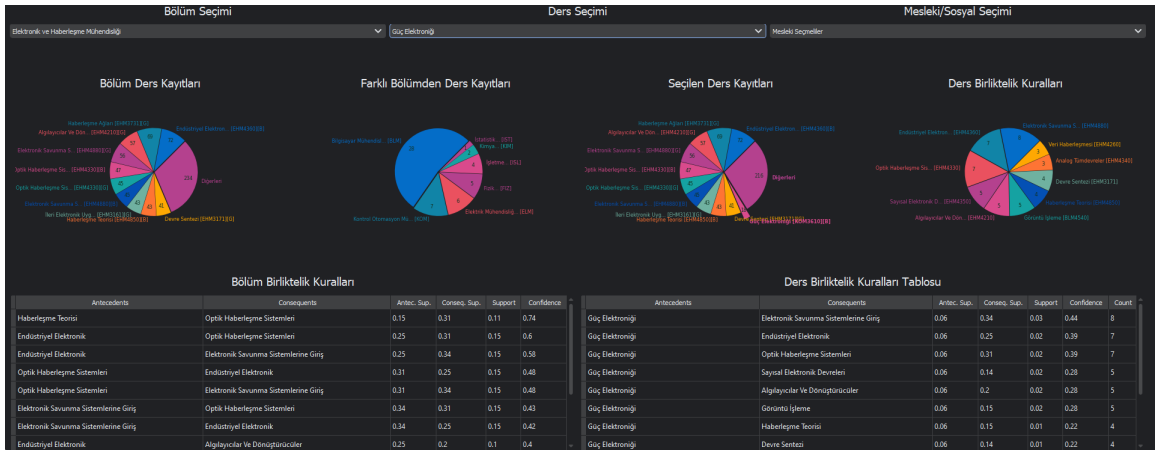
Veri Madenciliğine Giriş'den sonra ise Uzman Sistemlere Giriş dersiyle çok yakından

alakalı 2 dersin, Yapay Sinir Ağlarına Giriş ve Yapay Zeka derslerinin tercih edildiğini görüyoruz. Burdan bu üç dersin genellikle birlikte tercih edildiğini çıkartabiliriz.



Şekil 7.2 Bilgisayar Mühendisliği bölümü İstatistiksel Veri Analizi dersi için birliktelik kuralları.

Bilgisayar Mühendisliği bölümünde bir başka ders olan İstatistiksel Veri Analizi dersiyle en çok tercih edilen dersin ise yine bu ders ile yakından alakası olan Ayrık Olay Simülasyonu olduğunu görüyoruz.



Şekil 7.3 Elektronik ve Haberleşme bölümü Güç Elektroniği dersi için birliktelik kuralları.

Elektronik ve Haberleşme Mühendisliği bölümüne bakacak olursak birlikte alınan derslerde ilk sıralarda yine bu dersle alakalı olan Mekatronik, Endüstriyel Elektronik gibi derslerin bulunduğunu görüyoruz.

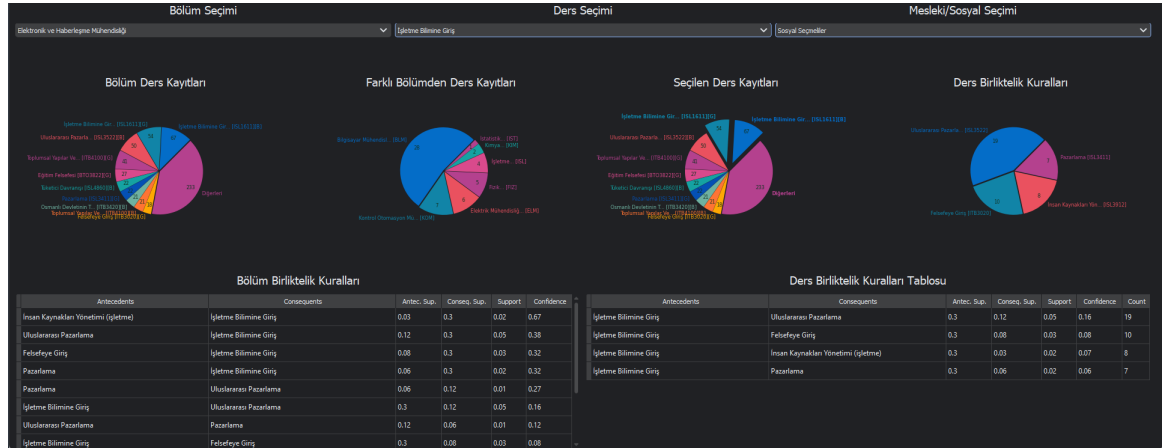
Bölüm Birliktelik Kuralları					
Antecedents	Consequents	Antec. Sup.	Conseq. Sup.	Support	Confidence
Haberleşme Teorisi	Optik Haberleşme Sistemleri	0.15	0.31	0.11	0.74
Endüstriyel Elektronik	Optik Haberleşme Sistemleri	0.25	0.31	0.15	0.6
Endüstriyel Elektronik	Elektronik Savunma Sistemlerine Giriş	0.25	0.34	0.15	0.58
Algılayıcılar Ve Dönüştürücüler	Elektronik Savunma Sistemlerine Giriş	0.2	0.34	0.1	0.53
Algılayıcılar Ve Dönüştürücüler	Endüstriyel Elektronik	0.2	0.25	0.1	0.51
Optik Haberleşme Sistemleri	Elektronik Savunma Sistemlerine Giriş	0.31	0.34	0.15	0.48
Optik Haberleşme Sistemleri	Endüstriyel Elektronik	0.31	0.25	0.15	0.48
Elektronik Savunma Sistemlerine Giriş	Optik Haberleşme Sistemleri	0.34	0.31	0.15	0.43

Şekil 7.4 Elektronik ve Haberleşme bölümü mesleki seçmeliler için birliktelik kuralları.

Elektronik ve Haberleşme Mühendisliğinde bölüm bazında bakacak olursak ilk sırada Haberleşme Teorisi ve Optik Haberleşme Sistemlerini birlikte görüyoruz. Yine birbiriyle oldukça alaklı 2 dersin, çok yüksek bir Confidence değeri ile (0.74) bir kural oluşturduğu görülüyor.

7.2 Sosyal Seçmeli Dersler

Sosyal Seçmeli dersler için elimizdeki veriler Mesleki Seçmelilere göre çok daha az olduğu için öncelikle burdaki kural sayısının Mesleki Seçmeli derslere göre çok daha az olduğunu görüyoruz. O yüzden bu bölümde sadece tek bir örnek vereceğiz.



Şekil 7.5 Elektronik ve Haberleşme bölümü İşletme Bilimine Giriş dersi için birliktelik kuralları.

Şekil 7.5’de de görülebileceği üzere, İşletme Bilimine Giriş dersi ve Uluslararası Pazarlama dersinin birlikte alındığını söyleyebiliriz. Sosyal seçmeli dersler için elimizde az miktarda veri bulunuyor, dolayısıyla çok fazla kural oluşturamıyoruz ve oluşturduğumuz kurallarda Support ve Confidence değerleri düşük çıkıyor.

8 Sonuç

Bu kısımda uygulama ve bu uygulamadan elde edilen sonuçlardan bahsedilmiştir.

Birliktelik analizi günümüz dünyasında da bir çok şirket tarafından kullanılan ham veriden çeşitli algoritmalar yardımıyla ilişkiler kurmamızı sağlayan veri madenciliği türüdür. Buna gerçek dünyadan örnek vermek gerekirse; e-ticaret siteleri, müzik dinleme uygulamaları ve dizi izleme platformları örnek verilebilir. Bu platformlar kullanıcılarından elde ettikleri verileri kazanç sağlama amacıyla analiz edip büyümeyi amaçlayabilirler.

Projemizin amacı UZEM verilerinden bölümlere göre öğrencilerin en çok tercih ettiği dersler, hangi dersleri birlikte alındığı ve en çok hangi bölümden ders alındığı gibi istatistiksel verilerin çıkartılmasıdır.

Verilen veriler kullanılmaya uygun hale getirildikten sonra Apriori algoritmasıyla birliktelik analizi yapılmış ve bulunan değerler Support ve Confidence değerine göre sıralanmış olup çıkan sonuçlar grafikler ve tablolar ile sunulmuştur.

Proje sonunda gerçekleştirdiğimiz uygulamamız Öğrenciler ders seçerken hangi derslerin daha çok birlikte alındığını görebilmelerini ve bu istatistiklere göre fikir edinip seçim yapmalarına katkıda bulunabilir. Ayrıca okulumuzun bu verileri değerlendirip öğrencilerin tercihlerine göre ilgili derslere yatırım yapmasına katkı sağlayabilir.

- [1] T. A. Kumbhare and S. V. Chobe, “An overview of association rule mining algorithms,” *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014.
- [2] S. Kotsiantis and D. Kanellopoulos, “Association rules mining: A recent overview,” *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 71–82, 2006.
- [3] L. SABAH and H. BAYRAKTAR, “Veri madenciliği birliktelik kuralları ile binaların risk durumlarının analizi: Kaynaşlı, düzce örneği,” *Gazi Mühendislik Bilimleri Dergisi*, vol. 6, no. 1, pp. 70–78, 2020.
- [4] A. Çotur. “Apriori algoritması.” (2019), [Online]. Available: <https://cotur.medium.com/apriori-algoritmas%C4%B1-a90a38856373> (visited on 12/02/2022).

BİRİNCİ ÜYE

İsim-Soyisim: Denizay Biberoglu
Doğum Tarihi ve Yeri: 31.05.2000, İstanbul
E-mail: denizay.biberoglu@std.yildiz.edu.tr
Telefon: 0553 012 41 35

İKİNCİ ÜYE

İsim-Soyisim: Yusuf Yemliha Çelik
Doğum Tarihi ve Yeri: 15.08.1997, Adana
E-mail: yemliha.celik@std.yildiz.edu.tr
Telefon: 0532 698 33 61

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python
Gerekli RAM: 8GB
Gerekli Disk: 10GB