



---

**T.C.  
NAMIK KEMAL ÜNİVERSİTESİ  
BİLİMSEL ARAŞTIRMA PROJELERİ  
KOORDİNASYON BİRİMİ (NKÜBAP)**

---

**BİLİMSEL ARAŞTIRMA PROJELERİ  
SONUÇ RAPORU**

---

NKUBAP.00.17.AR.13.15 nolu Proje  
AKILLI DERS YÖNETİM SİSTEMİ İLE  
PROGRAMLAMA ÖDEVLERİ İÇİN  
İNTİHAL TESPİTİ UYGULAMASI

Yürütücüsü:  
Yrd.Doç.Dr. Erdinç UZUN  
Araştırmacılar:  
Araş. Gör. Cihat ERDOĞAN  
Araş. Gör. Ahmet SAYGILI  
2015

NKUBAP.00.17.AR.13.15 no'lu "AKILLI DERS YÖNETİM SİSTEMİ İLE PROGRAMLAMA ÖDEVLERİ İÇİN İNTİHAL TESPİTİ UYGULAMASI" adlı proje Namık Kemal Üniversitesi Bilimsel Araştırma Proje Birimi tarafından desteklenmiştir.

**T.C.  
Namık Kemal Üniversitesi  
Bilimsel Araştırma Projeleri Birimi**

**AKILLI DERS YÖNETİM SİSTEMİ İLE PROGRAMLAMA ÖDEVLERİ  
İÇİN İNTİHAL TESPİTİ UYGULAMASI**

**(Proje No: NKUBAP.00.17.AR.13.15)**

**Proje Ekibi:**

**Yürütücüsü:**

Yrd.Doç.Dr. Erdinç UZUN

**Araştırmacılar:**

Araş. Gör. Cihat ERDOĞAN

Araş. Gör. Ahmet SAYGILI

TEKİRDAĞ-2015  
Her hakkı saklıdır.

**T.C.**  
**NAMIK KEMAL ÜNİVERSİTESİ**  
**BİLİMSEL ARAŞTIRMA PROJELERİ**  
**SONUÇ RAPORU**

**ÖNSÖZ**

Ödev eğitiminin önemli bir parçası olmasının yanı sıra verilen ödevin doğru bir şekilde değerlendirilmesi de eğitime önemli bir katkı sağlayacaktır. Bu projenin amacı, hiyerarşik kümeleme yöntemlerini kullanarak ödev değerlendirme sürecine katkıda bulunacak bir yazılım geliştirme olarak planlanmıştır. Bu proje sonunda, tüm üniversitesin kullanabileceği web tabanlı, sadece ödev değerlendirmeyi kolaylaştıran değil, ayrıca dünya ve yerel olarak ders notu paylaşılabilen, ödev ve uygulama takibini kolaylaştıran, bir çok farklı ortamda (cep telefonu, tablet, bilgisayar vs.) çalışabilen web tabanlı bir yazılım geliştirilmiştir. Geliştirilen uygulama "<http://bilgmuuh.nku.edu.tr/adys>" üzerinde aktif durumdadır ve kullanılmaktadır.

Namık Kemal Üniversitesi Bilimsel Araştırma Projeleri (NKUBAP.00.17.AR.13.15), Bilgi İşlem Daire Başkanlığı'na, uygulama için önerilerde bulunan hocalarımıza ve projeyi değerlendiren hakemlere desteklerinden dolayı teşekkür eder, geliştirilen yazılımın tüm üniversitemizde eğitime olumlu bir etki yapmasını dileriz.

## İÇİNDEKİLER

Özet .....	vii
Abstract .....	viii
1. Giriş .....	1
2. Gereç ve Yöntem .....	3
2.1. <i>İntihal Tespit Modülü</i> .....	3
2.1.1. Doküman Benzerliği (Document Similarity) .....	3
2.1.2 Hiyerarşik Kümeleme (Hierarchical Clustering) .....	4
2.1.3. Dendrogram ve Küme Sayısı (Dendrogram and Number of Clusters) .....	7
2.2. Uygulama Tarafı .....	9
3. Bulgular ve Tartışma/Sonuç .....	9
4. Kaynaklar .....	11
Ekler .....	13

## ŞEKİLLER DİZİNİ

<b>Şekil 1.</b> Tek Bağlantı, Tam Bağlantı ve Ortalama Grup Algoritmaları .....	5
<b>Şekil 2.</b> Hiyerarşik kümeleme algoritması başlangıç durumu - 1.....	6
<b>Şekil 3.</b> Hiyerarşik kümeleme algoritması başlangıç durumu - 2.....	6
<b>Şekil 4.</b> Hiyerarşik kümeleme algoritması 2. Döngü .....	7
<b>Şekil 5.</b> Örnek veriler için oluşturulan dendrogram .....	8
<b>Şekil 6.</b> Cophenetic korelasyon katsayısına için kullanılan matrisler .....	10

**TABLO DİZİNİ**

<b>Tablo 1.</b> Algoritma Adımları ve Sonuçlar .....	7
--	---

**ALGORİTMA DİZİNİ**

<b>Algoritma 1.</b> Ödev Değerlendirme için Hiyerarşik Kümeleme Algoritması .....	5
---	---

<b>Algoritma 2.</b> Dendrogram çizimi Algoritması .....	8
---	---

## Özet

Ödevler, eğitimin en önemli öğelerinden biridir. Günümüzde, ödevler internet üzerinden verilebilmekte ve aynı şekilde alınabilmektedir. Verilen bir ödevin değerlendirilmesinde gönderilen ödevin genelde sadece doğru olup olmadığı kontrol edilebilmektedir. Kimlerin benzer ödev gönderdiği, ödevlerin hangi oranda benzerliğe sahip olduğu ve ödevlerin hangi kısımlarının benzerliğinin fazla olduğu gibi ayrıntılar ne yazık ki değerlendirmeyi zorlaştırmaktadır. Bu projenin amacı, hiyerarşik kümeleme yöntemlerini kullanarak ödev değerlendirme sürecine katkıda bulunacak bir makine öğrenmesi sistemini geliştirmektir.

Bu projede, "Akıllı Ders Yönetim Sistemi - (ADYS)" adında bir web tabanlı uygulama geliştirdik. Bu uygulamada öğretim üyesi ders notlarını, uygulama notlarını ve ödevleri internet üzerinden düzenleyebilmektedir. Öğrencilerde öğretim üyesinin verdiği notları görebilmelerin yanı sıra kendine ait ödevi bu sistem üzerinden dersin öğretim üyesine gönderebilmektedir. Öğrenci bir sıkıştırılmış dosya (zip veya rar), bir metin belgesi (txt, cs, aspx, java, php vb.) veya zenginleştirilmiş metin belgesi (doc, xls, ppt veya pdf gibi) olarak ödevi gönderebilmektedir. Öğretim üyesi ödevleri aldıktan sonra ödevleri inceleyerek uygulama üzerinden öğrenciye notunu vermektedir. Bu güne kadar gönderilen ödevler ayrıntılı incelendiğinde birçok öğrencinin benzer ödevler gönderdiği görülmüştür. Ancak öğrenciye verilen not genelde ödevin doğru olup olmadığına ait bir nottur. Orijinal ödev yapan öğrenciler benzer ödev gönderen öğrencilerle aynı notu almaktadır. Bu sebepten dolayı, bu projede öğretim üyesi için orijinal ödev kontrolünü kolayca sağlayabileceği bir sistem tasarlanmıştır. Böylece, bu uygulama üzerinden verilen ödevlerin öğrencileri orijinal ödev hazırlamaya teşvik edeceği düşüncesindeyiz.

**Anahtar Kelimeler:** Web tabanlı ders yönetim sistemleri, İntihal Tespiti, Web Programlama



## Abstract

Assignments are one of the most important subjects of education. Recently, assignments can be given through internet and answers can be easily collected via the same way. In assessment of a given assignment, the uploaded assignment can only be checked whether it is correct or not. Unfortunately, details such as students who submitted a similar assignment, the similarity of assignments and the similar parts of an assignment makes the assessment difficult. The aim of this project is to develop a machine learning system that will contribute the assessment process of assignments by using hierarchical clustering methods.

In this project, we have developed a web-based application, namely “Intelligent Lesson Management System”. In this application, a lecturer arranges the assignments, lecture contents and notes, and grades through a web application. Likely a student receives these notes and sends his/her own assignments in compressed formats (zip or rar) and text formats (txt, cs, aspx, java, php, and etc.) as well as enriched text formats (doc, xls, ppt or pdf) through this application. After the assignments are taken by a teacher, the grades are given from the web page. When the assignments are investigated in detail, it is observed that most of the parts of answers are very similar. However the grades are given according to the right answers but not how much they are original. Some students may copy the answers from each other, and this causes these students to get the same grade. Similarly some students who give original answers get the same grades with the students who give correct answers. In order to prevent this situation, this project aims an application which is guided the lecturer to assess the originality of the answers. Therefore, the assignments given by this application will force the students to prepare original answers.

**Keywords:** *Web-based Course Management System, Plagism Detection, Web Programming*

# 1. Giriş

Bu projede, öğrencilerin ödevleri arasındaki yakınlık derecelerini bulabilen ve çıkan bu derecelendirme üzerinden hiyerarşik kümeleme metotlarına göre ödevleri kümeleyebilen bir sistem geliştirilmiştir. Proje sayesinde:

- Ödevler daha kısa sürede ve daha doğru biçimde değerlendirilebilecek
- Öğrencilerin birbirinden yaptıkları alıntılar tespit edilebilecek
- Orijinal ödevler daha rahat tespit edilebilecek
- Puanlama sistemi daha adil olabilecek
- Öğrenciler orijinal ödev hazırlamaya teşvik edilebilecektir.

Ödevlerin ayrıntılı şekilde değerlendirilmesi ve benzer ödevlerin tespit edilmesi çok zaman alıcı bir işittir. Hem asistanların hem de dersi veren öğretim üyelerinin çok fazla zamanını almaktadır. Ayrıca çoğu üniversitedeki asistan sayısının azlığı göz önüne alındığında ödevlerin değerlendirilmesi zorlaşmaktadır. Bu projede, alıntı tespitini otomatik olarak gerçekleştirebilen bir web uygulaması geliştirilmiştir.

Alıntılarının tespit edilmesi literatürde çok popüler bir konudur ve literatür özetinde belirteceğimiz gibi birçok alanda farklı uygulamaları vardır. Fakat ödev alıntı sistemleri üzerine uygulama sayısının az olduğu ve hatta web tabanlı bir uygulamanın olmadığı görülmüştür. Bu çalışmada, sınıf içerisindeki öğrencilerin ödev hazırlarken birbirinden yaptıkları alıntılarının tespitini kolaylaştıran bir uygulama geliştirilmiştir. Geliştirilmiş uygulama sayesinde:

- Gönderilen dokümanlar otomatik olarak incelenip yakınlık derecelendirmesi belirlenecektir.
- Bu yakınlık derecelendirmesi bir dendrogram üzerinden hiyerarşik kümeleme tekniklerine göre benzer ödevler kümelenebilecektir.
- Bu sistem sayesinde eğitimin en önemli parçalarından biri olan ödevler daha iyi bir şekilde değerlendirme yapılmasına olanak sağlanacaktır.

İntihal tespiti, sadece ödev alanında değil akademik alanda tez veya makale değerlendirilmesi (Arda, 2003) ve diğer bir taraftan internet ortamındaki alıntı / benzer haberlerin veya web sayfalarının bulunması gibi birçok farklı alanda kullanılmaktadır. Ödev intihali, kendi içinde arkadaştan yapılan ödev intihali ve başka kaynaklardan yapılan ödev intihali olmak üzere ikiye ayrılabilir. Tez, makale, haber siteleri ve web sayfaları gibi kaynaklar üzerinde intihalleri / alıntılarını bulmak için verilerin depolanması ve hızlı olarak işlenebilmesi için büyük sunuculara ihtiyaç vardır. Büyük kaynaklara sahip olan ve farklı değerlendirmelerin yapılabildiği olanak sağlayan internet üzerinde birçok yazılım göze çarpmaktadır. Örneğin bu yazılımlardan ithenticate.com 17 milyar web sayfası ve 86 milyon araştırma makalesi gibi büyük bir kaynağa sahiptir. Bununla birlikte, www.turnitin.com, www.jplag.de ve www.plagiarism.com gibi intihal tespit eden sistemler vardır. Ancak, bu yazılımlar ticari, algoritmaları gizli ve çok büyük kaynaklara ihtiyaç duymaktadırlar. Bu noktada, günümüzde yaygınlaşan açık kaynak kodlu yazılımlar düşünülebilir. Bunlardan biri WCopyfind (plagiarism.bloomfieldmedia.com) yazılımıdır. Bu yazılım sadece iki dokümanı inceleyerek benzer kısımların altı çizik olarak gösteren bir html raporu üretmektedir. Bu yazılım sadece bir

bilgisayarda çalışan masaüstü uygulamasıdır. Geliştirilmesi düşünülen proje ise internet üzerinden kullanılabilir.

Programlama ödevlerinde internet üzerinden bilgi kullanımı serbesttir. Kod yazan kişi internet üzerinden problemlerine ait çözümleri bularak daha iyi bir yazılım üretebilir. Ama aynı çıktığı üretmesi rağmen kodlar birbirinden farklı olur. Bu açıdan literatürde birçok çalışma göze çarpmaktadır. Bu konudaki ilk uygulama 1981 yılında Fortran kaynak kodları üzerine geliştirilen bir araçtır (John, 1981). Sam (1981) benzer bir yapıyı Pascal üzerine taşımıştır. Whale (1990) Pascal ve Prolog kodlarını yorumlayan Plague adını verdiği sistemi geliştirmiştir (Whale I - II, 1990). Chen ve ark. (2004) Java ve C++ kaynak kodlarını destekleyen SID adında bir sistem geliştirmiştir. Bu tür çalışmalarda satır sayısı, değişken sayısı, alt program sayısı, çağrılan alt program sayısı ve benzeri parametreler kullanılır. Engels ve ark. (2007) ilk defa yapay sinir ağlarını kullanmışlardır. Bu çalışmada öncelikle ellerindeki dokümanlardan çeşitli özellikler çıkarmışlar ve bu dokümanları eğitim verisi olarak kullanarak intihal tespitine bir çözüm aramışlardır. Programlama kodu konusunda en çok bilinen uygulama ise 1994 yılında Stanford Üniversitesi'nde geliştirilmeye başlanan ve halen güncellemeleri çıkan MOSS (for a Measure of Software Similarity) uygulamasıdır (Schleimer, 2003). Bu uygulama, projede önerdiğimiz gibi yazılım kodları arasındaki benzerlikleri için bir değer üreten ve intihalleri tespit etmekte kullanılan bir programdır. Ancak, sadece üyelik sistemi üzerinden kullanılabilen ve web servisi bazında çalışan bir uygulamadır. Bu web servisi ticari olarak satılmaktadır. Bu uygulama benzer olarak özellikle metin benzerliklerinin araştırılmasına yönelik birçok algoritma literatürde tanıtılmıştır (Wise, 1992; Heintze, 1996; Lyon ve ark., 2001; Yerra ve Ng, 2005; Samuel ve Zelda, 2006; Lingxiao ve ark., 2007). Ancak geliştirilen algoritmalar genelde iki metin arasındaki benzer kısımlarının veya benzerlik oranının bulunması üzerinedir. Bu projede, önerilen uygulama dendrogram denilen bir ağaç grafiği şeklinde tüm öğrencilerin ödevleri arasındaki yakınlığı görsel olarak görmemize olanak sağlamaktadır. Bu sayede ikişerli gruplar halinde ödevler karşılaştırılması yerine tüm ödevler bir grafik üzerinden incelenebilmektedir. Bu grafiğin elde edilebilmesi için en uygun yapı hiyerarşik kümeleme modelidir. Hiyerarşik kümeleme modeli yöntem bölümünde anlatılacaktır.

Hiyerarşik kümeleme doküman sınıflandırması, tıp, ekonomi ve bunlar gibi birçok farklı alan problemlerinin çözümünde kullanılmaktadır. Örneğin, Fung ve ark. (2004) hiyerarşik doküman kümelemede büyük miktarda veride ve anlamlı kümeleme etiketi oluşturmada karşılaşılan zorlukları aşmayı anlatmışlardır. Çiftçi (2011) beyin bölgelerini işlevsel olarak kümelendirmek için kullanmıştır. Karabulut ve ark. (2008) ise çalışmalarında Türkiye'deki illerin sosyo-ekonomik benzerliklerini bu kümeleme metodu üzerinden incelemişlerdir. Bu proje, Fung ve ark. (2004) benzer şekilde doküman kümeleme üzerine olacak fakat onların belirttiği gibi kelimeler bir kısmı değil öğrencinin ödev içinde kullandığı tüm kelimeler dikkate alınacaktır. Genelde bu tür çalışmalarda kümeleme veya dendrogram çizimi aşamalarında Matlab veya Python içindeki hazır kütüphaneler tercih edilir. Bu uygulamada, Yeşilbudak ve ark. (2011) nesneye yönelik yaklaşımı göz önüne alınarak bu çalışmaya benzer şekilde tüm kodlar .Net Framework platformunda C# programlama dili ile yazılmıştır.

Eğitim üzerine yapılan araştırmalarda öğrencilerin büyük çoğunluğu ödevlerde ve sınavlarda kopya çekme davranışlarının olduğunu görülmüştür (Lanier, 2006). Öğrencilerin ödevleri aşırma, kopya etmek ve hatta satın alma gibi aşırma yöntemlerine başvurdukları tespit edilmiştir (Park, 2003). Türkiye'de, ödev intihal tespiti üzerine uygulama bazında bir çalışma göze çarpmaktadır. Yapılan çalışmalarda mevcut intihal tespitleri incelenmiş ve öğrencilerin niye intihal yaptıkları araştırılmıştır (Uzun ve ark., 2007).

## 2. Gereç ve Yöntem

Proje, hem teorik kısımdan hem de geliştirilen uygulama kısmından oluşmaktadır. Teorik kısımda, intihal tespiti için kullanılacak en uygun makine öğrenmesi metodunun araştırılması yapılmıştır. Uygulama kısmında, güncel yazılım geliştirme ortamı ASP.Net MVC, en popüler veritabanı yönetim sistemlerinden MySQL, sunucu tarafında C# .Net ve istemci tarafında JQuery ve Javascript programlama dilleri yanı sıra görsel tasarım içinse Bootstrap kütüphanesi kullanılmıştır. Bu iki kısım ayrı ayrı anlatılacaktır.

### 2.1. İntihal Tespit Modülü

Geliştirilen intihal tespit modülü birbirini tamamlayan 4 aşamadan oluşmaktadır.

- Gelen ödev dokümanları içinden metnin çıkarılması
- Metinler üzerinden dokümanların benzerliklerinin bulunması
- Bu benzerlik değerleri üzerine Hiyerarşik kümelemenin uygulanması
- Dendrogram üretimi ve dendrogram üzerinden küme sayısının belirlenmesi

Genelde bilgisayar ortamında hazırlanan ödevler bir çok dosya tipinde (örneğin .c, .cs, .java, .py, .php, .docx, .doc, .xls, .xlsx, .ppt, .pptx, .pdf, .zip ve .rar gibi) gönderilmektedir. Gönderim işlemi bittikten sonra geliştirilen uygulama metin elde etme aşamasına başlar. Öncelikle, gelen ödev sıkıştırılmış bir dosya ise öğrenci numarası içeren bir klasör oluşturulur. Bu aşamada uygulama iç içe geçmiş birden fazla sıkıştırılmış dosya var ise rekürsif bir şekilde sıkışmış dosya kalmayınca kadar tüm dosyaları bu klasör içine açar. Tüm dosyalar açıldıktan sonra farklı tipteki dosyalara metin olarak erişmemizi sağlayan uygun filtreleme metodu kullanılır. Bu çalışmada, filtreleme işlemi için IFilter Arayüzü<sup>1</sup> kullanılmıştır.

#### 2.1.1. Doküman Benzerliği (Document Similarity)

Metinler elde edildikten sonra ikinci aşamada tüm öğrenci dokümanları birbiri ile karşılaştırılıp bir sayısal değer üretilmelidir. Bu sayısal değer üretimi için Bilgi Erişimi (Information Retrieval) alanında kullanılan Cosine Benzerliği (Cosine Similarity), Jaccard Benzerlik Katsayısı (Jaccard Similarity Coefficient) ve Dice'in Katsayısı (Dice's Coefficient) olmak üzere üç farklı benzerlik metodu denemiştir. Bu üç benzerlik metodu, metin madenciliğinde dokümanların karşılaştırılmasında ve veri madenciliğinde ise kümeler arasındaki uyumu ölçmede sıklıkla kullanılmaktadır (Tan ve ark., 2005). Cosine benzerlik metodunda, iki vektör arasındaki kosinüs açısının ölçümünden elde edilen değer benzerlik için kullanılır.

$$\cos\_sim(X, Y) = 1 - \frac{X \bullet Y}{|X||Y|} = 1 - \frac{x_i x_j + y_i y_j}{\sqrt{x_i^2 + x_j^2} \sqrt{y_i^2 + y_j^2}}$$

Burada, X ve Y farklı iki doküman olmak üzere  $x_i$  ve  $y_i$  benzer kelime sayısını ve  $x_j$  ve  $y_j$  dokümanlardaki tüm kelime sayısını vermektedir. Jaccard Benzerlik Katsayısı, benzer eleman

---

<sup>1</sup>IFilter Arayüzünü bilgisayarda farklı tipte dosyaların içindeki metinlere ulaşmanıza olanak sağlayan bir kütüphanedir. (<http://msdn.microsoft.com/en-us/library/ms691105>)

sayısının toplam kelime sayısına oranıdır.

$$jac\_sim(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} = 1 - \frac{x_i}{x_j + y_j - x_i}$$

Burada,  $x_i$  ve  $y_j$  benzer kelime sayıları eşit olduğu için sadece  $x_i$  alınmıştır. Başka bir deyişle  $x_i$  iki dokümandaki benzer kelime sayıdır. Dice katsayısı (Dice, 1945), "Lee Raymond Dice" tarafından farklı bir alan için önerilmiş fakat doküman benzerliğinde kullanılan bir metottur.

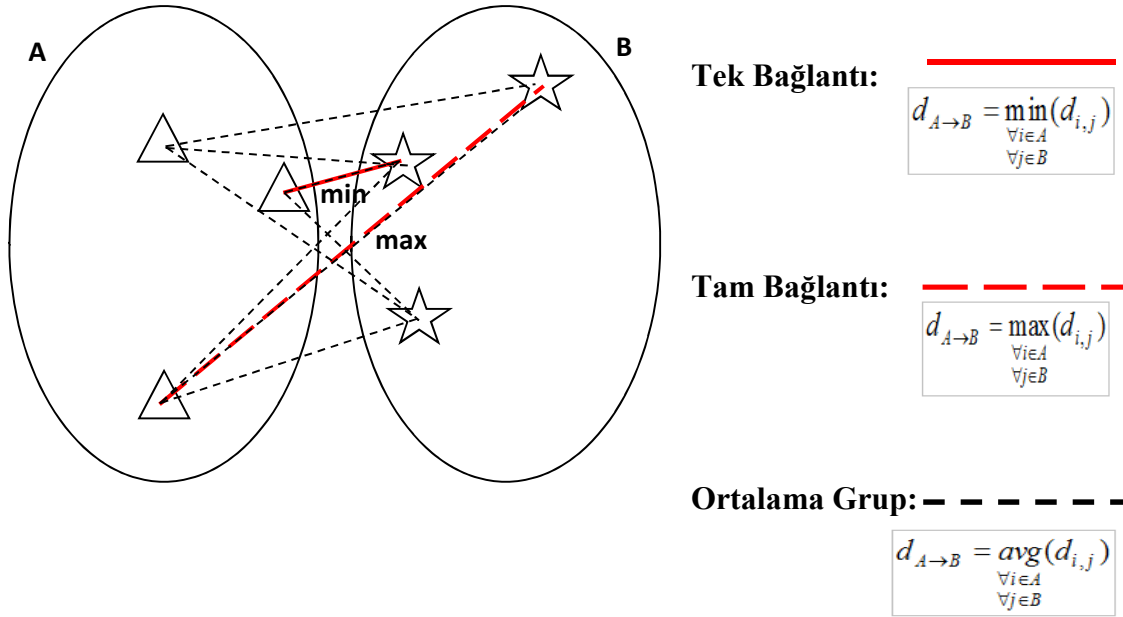
$$dice\_sim(X, Y) = 1 - \frac{2|X \cap Y|}{|X| + |Y|} = 1 - \frac{2x_i}{x_j + y_j}$$

Burada, benzer kelime sayısı toplam kelime sayısından çıkarma işlemi yerine 2 ile çarpılmaktadır. Tüm metotlarda  $[0, 1]$  arasında bir değer üretilmektedir. Değer 0'a yakın çıktığında benzerlik fazla iken değer 1'e yaklaştığında benzerliğin az olduğu anlamına gelmektedir. Testler bölümünde tüm metotların incelemesi yapılacaktır.

### 2.1.2 Hiyerarşik Kümeleme (Hierarchical Clustering)

Hiyerarşik kümeleme, diğer kümeleme yöntemlerine göre daha bilgilendirici bir çıktı üretir. Hiyerarşik kümelemede, küme sayısı önceden belirlenmesi gerekmemektedir. Hiyerarşik kümeleme, birleştirici ve ayrıştırıcı hiyerarşik kümeleme yöntemleri olmak üzere iki türe sahiptir. Birleştirici hiyerarşik kümelemede başlangıçta her bir değer bağımsız bir küme olarak değerlendirilir ve bu değerler çeşitli algoritmalarla birleştirilip her aşamada bir üst küme oluşturulur. Ayrıştırıcı kümelemede ise başlangıçta tüm değerler bir küme olarak değerlendirilip yine çeşitli ayrıştırma algoritmalarıyla alt kümeler elde edilir. Genelde literatürde, bu çalışmada olduğu gibi birleştirici kümeleme yöntemi tercih edilmektedir (Manning ve ark., 2008).

Hiyerarşik kümelemede, sayısal değerler üzerinden farklı algoritmalarla göre bir küme üretilir. Bu noktada üç farklı Tek Bağlantı, Tam Bağlantı ve Ortalama Grup algoritmalarından biri kullanılabilir. Tek bağlantı kümeleme, her adımda en küçük mesafeye sahip en yakın üyeler kullanarak iki kümenin birleştirilmesidir. Bu algoritmaya, en yakın komşu algoritması da denir. Tam bağlantı kümeleme, tek bağlantının tam tersine her adımda en büyük mesafeye sahip en uzak üyeler kullanılarak iki kümenin birleştirilmesidir. Bu algoritmaya bu sebepten dolayı en uzak komşu algoritması da denir. Ortalama grup algoritması ise iki karşılıklı küme arasındaki tüm mesafelerin ortalamasıdır. Bu algoritma ile tek bağlantı ve tam bağlantı algoritmalarında elde edilen değerlerin arasında bir değer elde edilir (Manning ve ark., 2008).



**Şekil 1.** Tek Bağlantı, Tam Bağlantı ve Ortalama Grup Algoritmaları

Şekil 1’de, A ve B adında iki küme verilmiştir. Öncelikle, iki kümenin her elemanının  $d_{i,j}$  mesafeleri hesaplanır. Sonra istenilen algoritmaya göre min, max veya avg değerleri hesaplanır. Şimdi, ödev değerlendirme için geliştirilen hiyerarşik kümeleme algoritmasını inceleyelim. Aşağıda bu algoritmanın kaba kodu gösterilmektedir.

1. Küme Sayısı: Öğrenci Sayısı;
2. Benzerlik Matrisi: Öğrenci Ödev Benzerlik oranlarını içeren matris
3. while Küme Sayısı = 1 do
4. Benzerlik Matrisi içindeki minimum değeri bul
5. Minimum değere ait satır ve sütunu matriste birleştir
6. Matriste işlem yapılan satır ve sütun değerini güncelle (min, max veya avg)
7. Küme sayısı bir azalt
8. end

**Algoritma 1.** Ödev Değerlendirme için Hiyerarşik Kümeleme Algoritması

Algoritma 1, küme sayısı tek oluncaya kadar devam eder. Başlangıç aşamasında (öğrenci sayısı X öğrenci sayısı) kadar bir matris yaratılıp her döngü sonunda matris boyutu bir azalır. Döngü işlemi başlarken öncelikle matris içindeki minimum değer bulunur. Bu Minimum değer tek bağlantı (min) algoritması ile karıştırılmamalıdır. Minimum değere ait satır ve sütun bilgilerinden istenilen üç algoritmadan birisi kullanılarak yeni matris oluşturulur. Şimdi, hiyerarşik kümeleme algoritması daha iyi anlaşılması için sayısal bir örnek üzerinden anlatılacaktır. Bu örnekte 6 öğrencinin ödevleri arasında doküman benzerliğinden elde edilen değerler kullanılacaktır.

**Başlangıç Durumu: Ödev Benzerlik Oranları**

	1	2	3	4	5	6
1	0.000	0.458	0.177	0.419	0.4	0.161
2	0.458	0.000	0.486	0.263	0.263	0.462
3	0.177	0.486	0.000	0.000	0.300	0.004
4	0.419	0.263	0.300	0.000	0.000	0.276
5	0.419	0.263	0.300	0.000	0.000	0.276
6	0.161	0.462	0.004	0.276	0.276	0.000

**Birinci Döngü Minimum Seçimi**

	1	2	3	4	5	6
1						
2	0.458					
3	0.177	0.486				
4	0.419	0.263	0.300			
5	0.419	0.263	0.300	0.000		
6	0.161	0.462	0.004	0.276	0.276	

**Şekil 2.** Hiyerarşik kümeleme algoritması başlangıç durumu - 1

Şekil 2'de, matris incelendiğinde matristeki değerlerin yarısının tekrar olduğu görülmektedir. Bu yüzden  $i < j$  türünden bir şart ile matrisin sadece yarısı incelenebilir. Kalan elemanlar arasında en küçük değer 4 ve 5. öğrenci ödevlerinin benzerlik oranını gösteren 0 değeridir. Başka bir deyişle bu iki öğrencinin ödevleri tamamen aynıdır. 4 ve 5'ten yeni bir küme üretilebilir. Şimdi algoritmadaki 5. ve 6. adımları uygulayalım. (Şekil 3)

**5. Adım**

	1	2	3	4, 5	6
1					
2	0.458				
3	0.177	0.486			
4,5	???	???	???	????	
6	0.161	0.462	0.004	???	

???: Algoritma tarafından belirlenecek değerler

**6. Adım**

	1	2	3	4, 5	6
1					
2	0.458				
3	0.177	0.486			
4, 5	0.419	0.263	0.300		
6	0.161	0.462	0.004	0.276	

**Şekil 3.** Hiyerarşik kümeleme algoritması başlangıç durumu - 2

(4, 5) kümesi yapıldıktan sonra yeni değerler hesaplanmalıdır. Bu hesap sırasında Şekil 1'deki matris bilgileri kullanılır. Bu aşamada min, max veya avg algoritmalarından biri kullanılır. Bu örnekte min değerini kullanılacaktır. Min yani Tek Bağlantı hesabı:

$$d_{(4,5) \rightarrow 1} = \min(0.419, 0.419), \quad d_{(4,5) \rightarrow 2} = \min(0.263, 0.263), \quad d_{(4,5) \rightarrow 3} = \min(0.300, 0.300), \\ d_{(4,5) \rightarrow 6} = \min(0.276, 0.276)$$

Tek bağlantıya göre bu kümelerden en küçük değerler seçilir. İki ödev birbiriyle aynı olduğu için tüm benzerlikler aynı çıkmıştır. Şimdi ikinci döngüye geçelim. Yeni oluşan matriste en küçük değer 0.004'tür. Başka bir deyişle 3 ve 6'ncı öğrencilerden yeni bir küme oluşturulacaktır. (Şekil 4)

## 2. Döngü Başlangıç

	1	2	3	4, 5	6
1					
2	0.458				
3	0.177	0.486			
4, 5	0.419	0.263	0.300		
6	0.161	0.462	0.004	0.276	

## 2. Döngü Sonu

	1	2	3,6	4, 5
1				
2	0.458			
3, 6	0.161	0.462		
4, 5	0.419	0.263	0.276	

### Şekil 4. Hiyerarşik kümeleme algoritması 2. Döngü

Her döngünün sonunda küme sayısı 1 azalır ve küme sayısı 1 oluncaya kadar aynı işlemler tekrar eder. Her adım sonunda bir birleşim işlemi olur. Tablo 1'de adım adım oluşan küme, seçilen minimum değer ve küme ilişkisi verilmiştir.

**Tablo 1.** Algoritma Adımları ve Sonuçlar

Adım	Kümeler	Minimum
0.	1, 2, 3, 4, 5, 6	-
1.	4, 5	0.000
2.	3, 6	0.004
3.	1, (3, 6)	0.161
4.	2, (4, 5)	0.263
5.	(1, (3, 6)), (2, (4, 5))	0.276

Algoritma içindeki minimum değer dendrogram çiziminde ve test aşamasındaki cophenetic korelasyon matrisinin oluşturulmasında kullanılacaktır. Bu korelasyon matrisi sayesinde kullanılan doküman benzerliği metotlarının ve hiyerarşik kümeleme algoritmalarının etkinliği test edilecektir.

### 2.1.3. Dendrogram ve Küme Sayısı (Dendrogram and Number of Clusters)

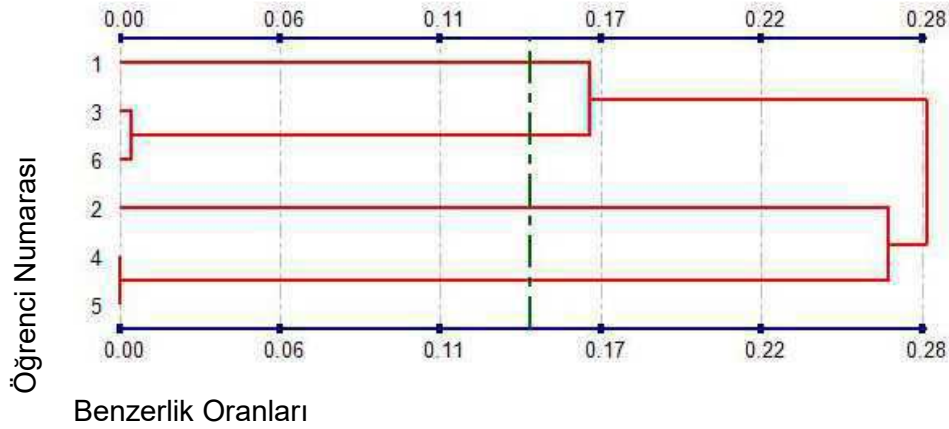
Hiyerarşik kümeleme çıktılarını en iyi şekilde yorumlamak için dendrogram kullanılır. Dendrogram çizimi için C# kütüphanesindeki System.Drawing kütüphanesi kullanılmıştır. Dendrogram çiziminin algoritması aşağıdaki gibidir.



1. Öğrenci sayısına uygun beyaz bir çizim alanı oluştur.
2. Maksimum değer: Kümeleme sonuçlarındaki maksimum sayı.
3. Maksimum değere göre üst ve alt cetvelleri x koordinatında oluştur.
4. y koordinatına öğrenci numaralarını kümeleme sonuçlarına göre yerleştir.
5. Dersin öğretim üyesinin belirlediği kesme değerini çiz
6. while tüm kümeleme sonuçları bitinceye kadar do
7. Kümeleme sonucundaki minimum değere göre uygun koordinatları hesapla
8. Bulunan koordinata kadar iki kümeleme sonucundan x ekseninde birer doğru çiz
9. İki doğrunun sonunu birleştir
10. end

### Algoritma 2. Dendrogram çizimi Algoritması

Dendrogram çizimi bittikten sonra ".jpg" uzantılı bir dosya olarak bilgisayara kaydedilir. Algoritma 2'de 5. satırda dersin öğretim üyesi kesme değeri belirlemektedir. Bu kesme değeri oluşacak küme sayısını belirler. Şimdi, oluşan dendrogramı inceleyelim. (Şekil 5)



--- : Öğretim üyesi tarafından belirlenen kesme değeri

### Şekil 5. Örnek veriler için oluşturulan dendrogram

Tablo 1'deki en büyük değer 0.28 olarak alınmıştır. Öncelikle (4, 5)'in 0 değeri çizilmiş ardından (3, 6)'nın 0.004 değeri çizilmiştir. Bu şekilde çizim algoritması tüm kümeleme sonuçlarının çizimini yapar. Dendrogram sayesinde öğretim üyesi çok rahat şekilde yapılan ödevleri yorumlayabilir. Örneğin 4 ve 5 nolu öğrencilerin aynı ödevi gönderdiği 3 ile 6 nolu öğrencilerin ise çok az bir farklılık içerdiğini görmektedir. 1 nolu öğrencinin 3 ve 6 nolu öğrencilere benzer bir ödev yaptığı görülmektedir. Bu benzerlik öğretim üyesi tarafından intihal oluşturacağı düşünülürse 0.161 büyük bir kesme değeri seçilebilir. Eğer farklı bir ödev yaptığı düşüncesinde ise şekildeki gibi 0.15 kesme değerini seçebilir. Bu kesme değeri kullanıldığında 1, (3, 6), 2, (4, 5) olmak üzere 4 adet küme oluşmaktadır.

## 2.2. Uygulama Tarafı

Akıllı Ders Yönetim Sistemi (ADYS), 2010 yılından bu yana ASP.Net Web Forms teknolojisine üzerine C# programlama diliyle MySQL veritabanını kullanan bir uygulama olarak geliştirilmeye başlandı. Değişen dünya koşulları (tablet, akıllı cep telefonları) teknolojik alt yapısını güncellemesini gerektirmiştir. Güncellenmiş ADYS, web tarafında bir çok güncel teknolojiyi desteklemektedir.

- Asp.Net MVC: Microsoft'un MVC teknolojisi üzerine kurulu güncel web sitesi geliştirme ortamıdır.
- JQuery: bir Javascript uygulama kütüphanesidir. İstemci tarafında yönetimi kolaylaştıran bir dildir.
- Bootstrap: açık kaynak kodu bir CSS uygulama çatısıdır. Tabletler, akıllı telefonlar ve bilgisayar için farklı ve cihaz büyüklüğüne uygun web siteleri yapmanızı sağlar.
- Diğer Kütüphaneler:
  - jTable, AJAX tabanlı tablolar yapılmasını ve yönetimi kolaylaştıran JQuery eklentisi.
  - jquery-validation-engine, form kontrollerini kolaylaştıran görsel JQuery eklentisi.
  - jquery-file-upload, dosya yükleme olayını kolaylaştıran JQuery eklentisi.
  - Timepicker, tarih ve saat seçimini sağlayan JQuery-UI eklentisi.
  - Countdown, sayaç kontrolünü kolaylaştıran JQuery eklentisi.

Uygulamanın özelliklerine, ekran görüntülerine ve kullanım özelliklerine "<http://bilgmuh.nku.edu.tr/adys/Home/About>" üzerinden ulaşabilirsiniz. Veritabanı tasarımı, "<http://bilgmuh.nku.edu.tr/adys/OpenCourse/Course/BMB202--Veritabanı-Yönetimi/92>" üçüncü hafta ders notlarında bulunmaktadır. Ayrıca, kullanım dökümanı, veritabanı tasarımı ve örnek dendrogram çıktıları proje dosyamızın ek bölümündedir.

## 3. Bulgular ve Tartışma/Sonuç

Uygulamadaki doküman benzerlik metotlarından ve hiyerarşik kümeleme algoritmalarından hangi çiftin en iyi sonuç verdiği bulmak için 54 adet ödevden oluşan bir veri seti hazırlanmıştır. Bu veri seti, son 2 yılı kapsayan 6 öğretim üyesine ait 18 farklı dersten oluşmaktadır. Genelde, Nesneye Yönelik Programlama, İşletim Sistemleri, Web Programlama, C Programlama Dili ve Sayısal Analiz gibi kod tabanlı dersler içermektedir. Bu kodlar, C#, Java, HTML, PHP ve Javascript gibi dilleriyle yazılmıştır.

Kümelemenin ne kadar iyi olduğu test etmek için hiyerarşik kümelemeden çıkan sonuçlar ile başlangıç değerlerinin cophenetic korelasyon katsayısına bakılır. Bu testler için geçmiş senelerin ödevlerinden elde ettiğimiz 6 öğretim üyesinin 18 farklı dersine ait 54 farklı ödevi kullandık. Ödevler Nesneye Yönelik Programlama, İşletim Sistemleri, Web Programlama, C Programlama Dili ve Sayısal Analiz gibi kod tabanlı dersler içermektedir. Bu kodlar, C#, Java, HTML, PHP ve Javascript gibi dilleriyle yazılmıştır. Ödevleri değerlendirme kullanılan cophenetic korelasyon katsayısının formülasyonu:

$$c = \frac{ks * \sum_{i < j} y_{i,j}^2 - \sum_{i < j} y_{ij} * \sum_{i < j} z_{ij}}{\sqrt{(ks * \sum_{i < j} y_{i,j}^2 - (\sum_{i < j} y_{ij})^2) * (ks * \sum_{i < j} z_{i,j}^2 - (\sum_{i < j} z_{ij})^2)}}$$

Burada,  $y \in Y$  başlangıç matrisi ve  $z \in Z$  hiyerarşik kümelemeden çıkan sonuçlardan elde edilen matristir. ks matris içinde  $i < j$  bölümünde bulunan eleman sayısıdır. Bu değerler üzerinden cophenetic korelasyon katsayısına (c) hesaplanır. Hiyerarşik kümeleme işlemindeki örnek veriler üzerinden cophenetic korelasyon katsayısı bulalım. (Şekil 6)

Y matrisi

	1	2	3	4	5	6
1						
2	0.458					
3	0.177	0.486				
4	0.419	0.263	0.300			
5	0.419	0.263	0.300	0.000		
6	0.161	0.462	0.004	0.276	0.276	

Z Matrisi

	1	2	3	4	5	6
1						
2	0.276					
3	0.161	0.276				
4	0.276	0.263	0.276			
5	0.276	0.263	0.276	0.000		
6	0.161	0.276	0.004	0.276	0.276	

**Şekil 6.** Cophenetic korelasyon katsayısına için kullanılan matrisler

Şekil 6'teki Y matrisi başlangıç değerlerini içerir. Z matrisi oluşturulurken ise Tablo 1'deki değerler göz önüne alınır. Örneğin 1, (3, 6) değeri 0.161 Z matrisindeki (1,3) ve (1,6)'ya yazılır. Bu örnek için c değeri %87.02'dir. Hiyerarşik kümelemede metotların ve algoritmaların uygunluğu c değerinin yüksek çıkması ile ölçülebilir.

Sonuç olarak:

Bu projede, hiyerarşik kümeleme üzerinden öğrencilerinden birbirinden yaptıkları intihalleri kolayca tespit etmemizi sağlayan bölümümüzde geliştirilen bir uygulama anlatılmıştır. Bu uygulamanın sonuçları denenen algoritmalar ve uygulamanın getirileri olmak üzere iki farklı açıdan incelenebilir. Uygulama geliştirilirken denenen doküman benzerliği metotları açısından Jaccard Benzerlik Katsayısı ve denenen hiyerarşik kümeleme algoritmaları açısından Ortalama Grup algoritması çifti en iyi sonuçları vermiştir. Uygulamanın getirileri iki farklı başlık altında toplanabilir.

- Öğretim üyeleri ödevleri daha kısa sürede ve ödev orijinalliğini daha rahat değerlendirme imkânına kavuşmuşlardır. Bu sayede ödev notu verilirken sadece ödevin doğruluğuna değil ödevin ayrıca orijinalliğine göz önüne alınma imkânı oluşmuştur.
- Öğrenciler açısından ödevlerdeki intihal oranı azaltılmıştır. Bu sayede öğrencilerin ödevleri daha dikkatli hazırladıkları gözlenmiştir.

Açılan üniversite sayısının artması buna paralel olarak bölüm sayısının artması öğretimin en önemli parçalardan biri olan ödevin verilmesini ve değerlendirilmesi zorlaştırmaktadır. 4 yıldır öğrenci alan bölümümüzde öğretim üyesi ve asistan sıkıntısı olması sebebiyle ödevlerin genelde doğruluğuna bakılmaktaydı. Ödev açısından sadece doğruluğa bakıldığı fark eden öğrenciler intihal yoluna başvurmaya başlamışlardı. Bölümümüzde geliştirilen bu uygulama sayesinde intihal oranının azalmaya başladığı ve öğrencileri daha orijinal ödev hazırlamaya teşvik edildiği görülmüştür.

Bu uygulama sayesinde öğretim üyesi bir web ara yüzünden ders notlarını, uygulama notlarını ve ödevleri öğrenci ile paylaşabilmektedir. Öğrenci aynı sistem üzerinden ödevin son gönderme tarihine kadar ödevini gönderebilmekte ve ders notlarını indirebilmektedir. Web oturum yönetimi sayesinde her öğrenci aldığı dersi görebilmekte ve aldığı dersin ödevini indirebilmektedir. Öğretim üyesi de oturum yönetimi sayesinde sadece kendi dersinde değişiklik yapma hakkına sahiptir. Gönderilen ödevler, ödev orijinalliği değerlendirme uygulaması ile değerlendirilmektedirler. Ödev orijinalliği değerlendirme işlemi ne yazık ki ödev sayısı ve gönderilen dosya boyutuna bağlı olarak çok uzun süren bir işlemdir. Bundan sonraki diğer bir amacımız iki uygulamanın birbirine etkin bir şekilde haberleşmesini sağlamaktır. Ödev tarihi bittiği anda ödev otomatik olarak değerlendirilecek ve öğretim üyesi web sayfası üzerinden kesme değerini değiştirerek küme sayısını düzenleme imkânı sağlanacaktır. Gönderilen ödevler şu an öğretim üyesi bilgisayarına indirilip açılabilir. Diğer bir amacımız sunucu kapasitesini arttırdıktan sonra sunucu tarafında ödevi web sayfası haline dönüştürüp web sayfası içinde ödevleri gösterme ve iki veya daha fazla ödevi karşılaştırmasının yapılabilmesi üzerine olacaktır.

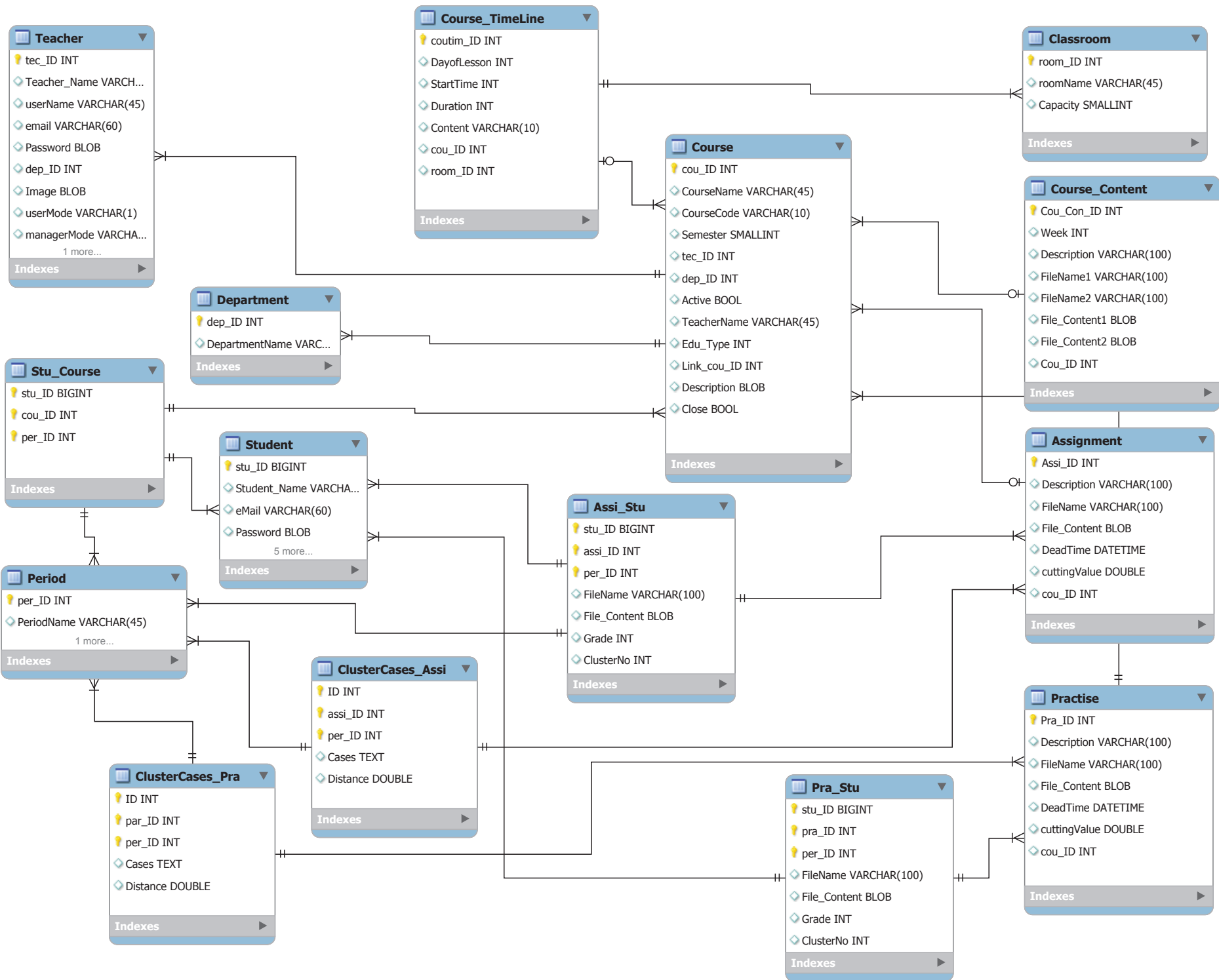
## 4. Kaynaklar

- Arda B., "Bilimsel bilgi üretiminde yayın etiği", Sağlık Bilimlerinde Süreli Yayıncılık Ulusal Sempozyumu, (2003).
- Chen C., Francia B., Li M., Mckinnon B. and Seker A., Shared information and program plagiarism detection. Transactions on Information Theory, IEEE. 50(7): 1545-1551, (2004).
- Çiftçi K., Minimum spanning tree reflects the alterations of the default mode network during Alzheimer's Disease, Annals of Biomedical Engineering, Cilt 39, Syf. 1493-1504, doi: 10.1007/s10439-011-0258-9, (2011).
- Dice, L. R., "Measures of the Amount of Ecologic Association Between Species". Ecology, 26 (3): 297-302, 1945.
- Engels S., Lakshmanan V. and Craig M., Plagiarism detection using feature-based neural networks, ACM. 39: 34-38, (2007).
- Fung B. K., Wang K. and Ester M., Hierarchical Document Clustering, Encyclopedia of Data Warehousing and Mining, Idea Group Reference, Cilt 1, A-H, 555-559, (2004).
- Heintze N., Scalable document fingerprinting, In Proceedings of the Second USENIX Workshop on Electronic Commerce, syf. 191-200, (1996).
- John L., Lancaster A., Sposato P., A plagiarism detection system, ACM. 13: 21-25, (1981).
- Karabulut M., Gürbüz M. ve Sandal E. K., Hiyerarşik Küme Tekniği Kullanılarak Türkiye'deki İllerin Sosyo-ekonomik Benzerliklerinin Analizi, SDÜ Sosyal Bilimler Enstitüsü Dergisi, Cilt 3, No 5, 65-78, (2008).
- Lanier M.M., Academic Integrity and distance Learning. Journal of Criminal Justice Education, Vol. 17 No.2, (2006).
- Lingxiao J., Su Z. and Chiu E., Context-based detection of clone-related bugs, Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, Dubrovnik, Croatia, ACM, (2007).
- Lyon C., Malcolm J. and Dickerson B., Detecting short passages of similar text in large document collections, In Proceedings of Conference on Empirical Methods in Natural Language Processing, 2001.
- Manning, C. D., Raghavan, P. ve Schütze, H., Introduction to Information Retrieval, Cambridge University Press, <http://informationretrieval.org/>, 2008.

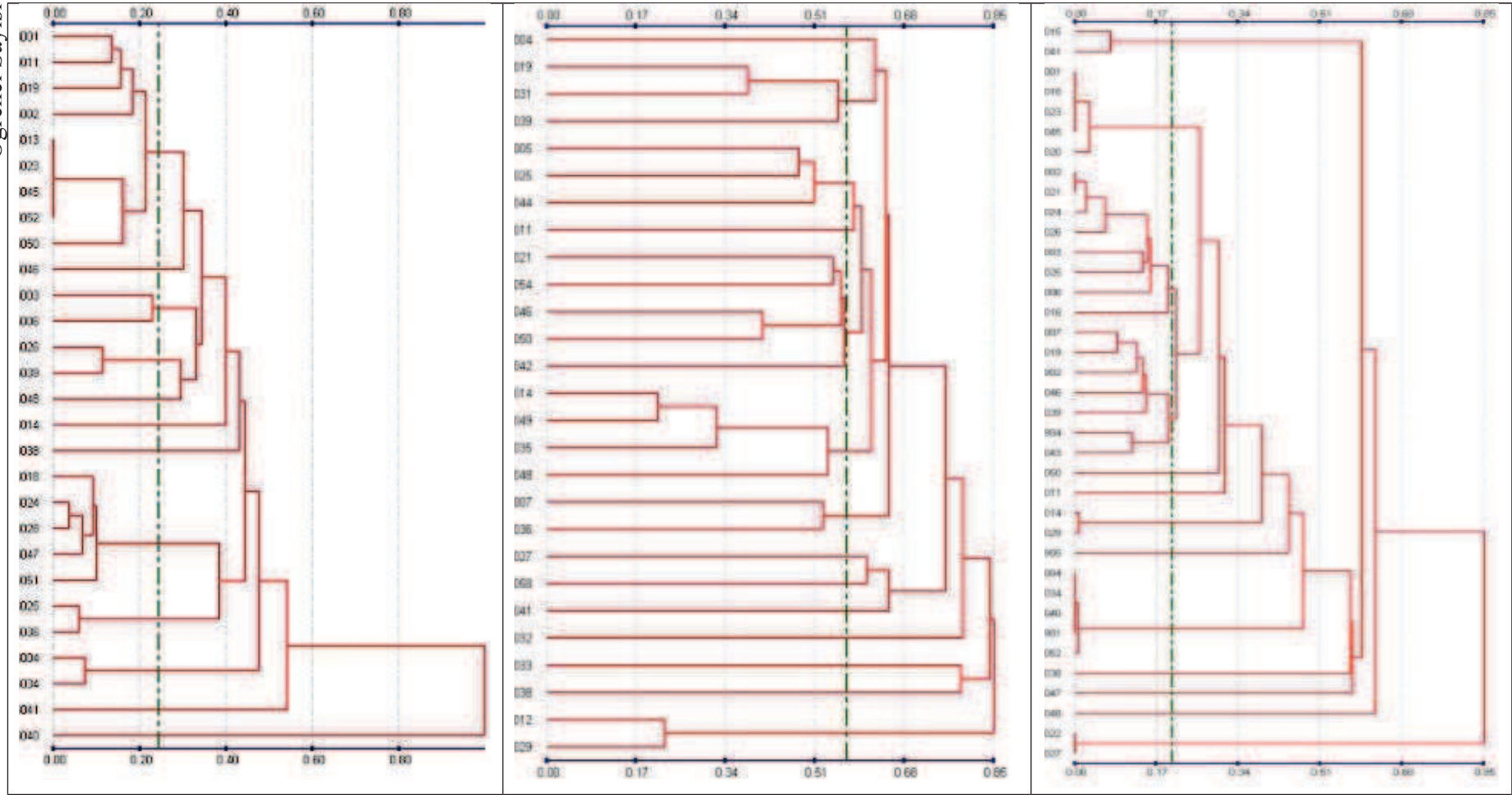
- Park C., In Other (People's) Words: Plagiarism by university students literature and lessons, *Assesment & Evaluation in Higher Education*, Vol. 28 No. 5, (2003).
- Sam G., A tool that detects plagiarism in Pascal programs, *ACM*. 13: 15-20, (1981).
- Samuel M. and Zeldia F., Similarity and originality in code: plagiarism and normal variation in student assignments" *Proceedings of the 8th Australian conference on Computing education*, Cilt 52. Hobart, Australia, Australian Computer Society, Inc., (2006).
- Schleimer S., Wilkerson D. S. and Aiken A., Winnowing: local algorithms for document fingerprinting, In *Proceedings of the 2003 ACM SIGMOD - International Conference on Management of Data*, syf. 76-85, (2003).
- Tan, P. N., Steinbach, M. ve Kumar, V., *Introduction to Data Mining*, Addison-Wesley, ISBN 0-321-32136-7, Bölüm 8; syf. 500, 2005.
- Uzun E., Karakuş T., Kurşun E. ve Karaaslan H., Öğrenci gözüyle aşırma(intihal): neden ve çözüm önerileri, *Akademik Bilişim '07, Bildiri Kitabı*: syf. 183-188, (2007).
- Whale G. Identification of Program Similarity in Large Populations, *Computer Journal*, 33 (2), pp 140-146, (1990).
- Whale G. Software Metrics and Plagiarism Detection, *J. Systems and Software*, 13 (2), pp 131-138, (1990).
- Wise M., Detection of similarities in student programs: YAP'ing may be preferable to plague'ing, *ACM*, 24: 268-271, (1992).
- Yerra R. and Ng Y., A Sentence-Based Copy Detection Approach for Web Documents, *Fuzzy Systems and Knowledge Discovery*: 557-570, (2005).
- Yeşilbudak M., Kahraman H.T. ve Karacan H., Veri madenciliğinde nesne yönelimli birleştirici hiyerarşik kümeleme modeli, *Gazi Üniversitesi Mühendislik Mimarlık Dergisi*, Cilt 26, No 1, 27-39, (2011).

## **Ekler**

1. Veritabanı Tasarımı (1 sayfa)
2. Geçmiş yıllardaki durumları gösteren dendrogram çıktıları (1 sayfa)
3. Uygulama tarafı (7 sayfa)



Öğrenci Sayısı



(a) Nesneye Yönelik Programlama (1)

(b) Nesneye Yönelik Programlama (2)

(c) İşletim Sistemleri I

Ödevlerde intihal durumları gösteren dendrogramlar (Dendrograms showed the plagiarism cases of assignments)



# Akıllı Ders Yönetim Sistemi

## ADYS

Amaçlar

Destek

Teknoloji

Öğrenci

Akademisyen

İletişim

ADYS, derslerde karşılaştığımız ders notu, uygulama ve ödev notu paylaşımlarını / yönetimlerini kolaylaştırmak için 2010 yılından bu yana geliştirilmesi devam eden uygulamamız 2013 yılından beri NKU-BAP tarafından desteklenmektedir.

## Amaçlar

Akademisyelerin, ders notları öğrencilerle ve isterse dünya ile paylaşmasını sağlamak

Akademisyelerin, uygulama ve ödevleri teslimini, takibini ve kontrolünü kolaylaştırmak

Akademisyenlerin, uygulama ve ödevleri intihalleri bulmasını hızlandırmak ve daha adil bir not sistemine imkan vermek

Öğrencilere, daha iyi ve orjinal ödev hazırlamasını teşvik etmek

## Destek

ADYS, 18.12.2013 tarihinden bu yana NKUBAP.00.17.AR.13.15 protokol nolu 'Akıllı Ders Yönetim Sistemi ile Programlama Ödevleri için İntihal Tespiti Uygulaması' başlıklı projemiz kapsamında NKU-BAP tarafından desteklenmektedir.

## Teknoloji

ADYS, 2010 yılından bu yana ASP.Net Web Forms teknolojisine üzerine C# programlama diliyle MySQL veritabanını kullanan bir uygulama olarak geliştirilmeye başlandı. Değişen dünya koşulları (tablet, akıllı cep telefonları) teknolojik alt yapısını güncellemesini gerektirmiştir. ADYS, web tarafında bir çok güncel teknolojiyi desteklemektedir.

JQuery, bir Javascript uygulama kütüphanesidir. İstemci tarafında

yönetimi kolaylaştıran bir dildir.

Bootstrap: açık kaynak kodu bir CSS uygulama çatısıdır. Tabletler, akıllı telefonlar ve bilgisayar için farklı ve cihaz büyüklüğüne uygun web siteleri yapmanızı sağlar.

ASP.Net MVC: Microsoft'un MVC teknolojisi üzerine kurulu güncel web sitesi geliştirme ortamıdır.

Diğer Kütüphaneler:

JTable, AJAX tabanlı tablolar yapılmasını ve yönetimi kolaylaştıran JQuery eklentisi.

jQuery-Validation-Engine, form kontrollerini kolaylaştıran görsel JQuery eklentisi.

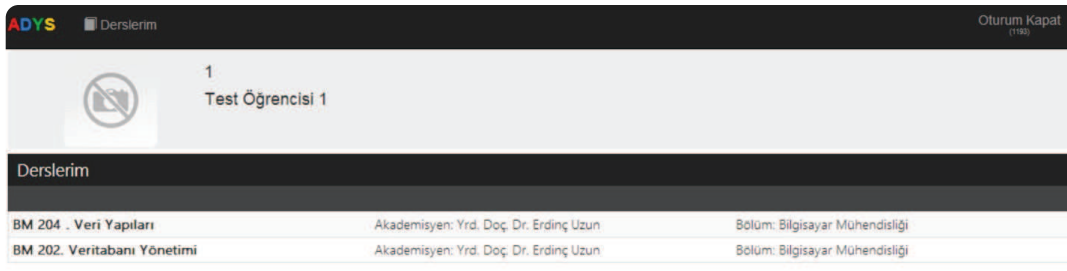
jQuery-File-Upload, dosya yükleme olayını kolaylaştıran JQuery eklentisi.

Timepicker, tarih ve saat seçimini sağlayan JQuery-UI eklentisi.

Countdown, sayaç kontrolünü kolaylaştıran JQuery eklentisi.

## Öğrenci

Öğrenci, kendisine ait öğrenci numarası ve şifre ile sisteme bağlandığında aldığı dersler karşısına çıkar. Dersler arasında seçim yaparak o derse ait ders notları, uygulamalar ve ödevler bölümüne ulaşır.



The screenshot shows the ADYS student interface. At the top, there is a navigation bar with the ADYS logo and a 'Derslerim' button. On the right, there is a 'Oturum Kapat' button. Below the navigation bar, there is a header section with a '1' and 'Test Öğrencisi 1'. The main content area is titled 'Derslerim' and contains a table of courses.

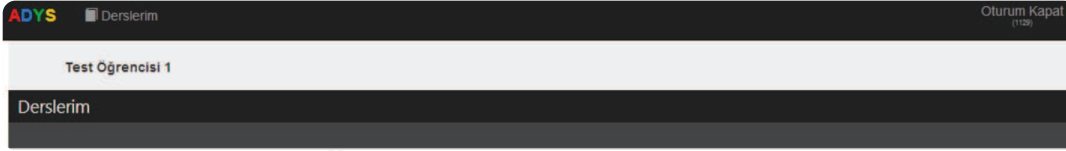
Derslerim		
BM 204 . Veri Yapıları	Akademisyen: Yrd. Doç. Dr. Erdiñ Uzun	Bölüm: Bilgisayar Mühendisliđi
BM 202. Veritabanı Yönetimi	Akademisyen: Yrd. Doç. Dr. Erdiñ Uzun	Bölüm: Bilgisayar Mühendisliđi

© 2014 - NKÜ Bilgisayar Mühendisliđi Bölümü

## Ders Notları

Öğrenci, bir ders seçimi yaptıđında "Ders Notları" karşısına çıkar. Ders notlarını kendi bilgisayarına indirebilir.

\*Ayrıca, öğrenci ismi üzerine tıklayarak üst bölümü kapatabilir ve tekrar tıklayarak açabilir.



## Uygulama ve Ödev

Öğrenci, uygulama dersinde uygulamasını veya verilen ödevleri takip için "Uygulamalar" ve "Ödevler" sekmelerini kullanır. Bu bölümlerden uygulama veya ödev içeriğini indirebilir. Kalan süreye göre ödevini gönderebilir. Ödevini gönderdikten sonra "Dosya Öğrenci" bölümünden dosyanın doğru bir şekilde ulaşıp ulaşmadığını kontrol edebilir.

\*Dosya boyutu 1 MB ile sınırlıdır. Sıkıştırma programları dosyanızı küçültebilir veya büyük boyutlu resimlerinizi resim düzenleme programları ile boyutlarını küçültebilirsiniz.

\*\*"Beni Hatırla" yı tıklamadıysanız 1200 saniye sonra sayfanın kapatılacağını unutmayın. "Beni hatırla" tıkladığınızda oturumuz hep açık kalır, güvenlik açısından kendi bilgisayarınız hariç bu sekmeyi kullanmanız önerilmez. Veya işiniz bittiğinde "Oturum Kapat" ile oturumu sonlandırabilirsiniz. Aksi taktirde gönderdiğiniz ödevler silinebilir veya çalınabilir.


No	Açıklama	Dosya	Durum	Dosya Öğrenci
1	VT Tasarımı ve Klasik VTYS işlemleri	vt_odev_1.pdf	Dosya İşlemleri 1 gün 13:26:04	Dosya Yok!
2	C# - ADO.Net	vt_odev_2.pdf	Süre Dolmuş!	Dosya Yok!
3	Full-text Search	vt_odev_3.pdf	Süre Dolmuş!	Dosya Yok!

© 2014 - NİÜ Bilgisayar Mühendisliği Bölümü

## Akademisyen

Akademisyen, kendisine ait email kullanıcı ismi ve şifre ile sisteme bağlandığında aldığı dersler karşısına çıkar. Dersler arasında seçim yaparak o derse ait ders notları, uygulamalar ve ödevler bölümüne ulaşır.

ADYS
Oturum Kapat (1196)



Yrd. Doç. Dr. Erdiç Uzun

Güz Dönemi Bahar Dönemi

**Örgün Öğretim**

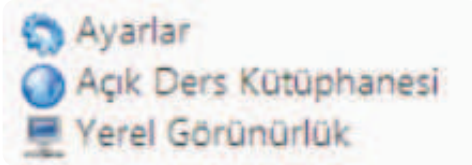
BM201. Nesneye Yönelik Programlama	Bölüm: Bilgisayar Mühendisliği	Ayarlar
BM305. Web Tasarımı ve Programlama	Bölüm: Bilgisayar Mühendisliği	Ayarlar
BMB409. Proje 1 (Şube 2)	Bölüm: Bilgisayar Mühendisliği	Ayarlar

**İkinci Öğretim**

BM201. Nesneye Yönelik Programlama	Bölüm: Bilgisayar Mühendisliği	Ayarlar
BM305. Web Tasarımı ve Programlama	Bölüm: Bilgisayar Mühendisliği	Ayarlar
BM409. Proje 1 (Şube 2)	Bölüm: Bilgisayar Mühendisliği	Ayarlar

© 2014 - NKÜ Bilgisayar Mühendisliği Bölümü

## Dünya ve Yerel Paylaşım



Akademisyen, "Ayarlar" bölümü dünya ile veya sadece öğrencileriyle derslerini paylaşabilir. Tek tıklama ile dersini sadece öğrencilerle veya tüm dünya ile paylaşım açar. Dersinizi tüm dünya ile paylaşım açtığınızda "Açık Ders Kütüphanesi" bölümünde dersiniz görünür duruma gelir. Ayrıca tüm ders notları, uygulamalar ve ödevler için aynı paylaşım imkanları vardır.

## Ders Notları

Akademisyen, bir ders seçimi yaptığında "Ders Notları" karşısına çıkar. Ders notlarını yükleyebilir veya bilgisayarına indirebilir.

\*Ayrıca, akademisyen ismi üzerine tıklayarak üst bölümü kapatabilir ve tekrar tıklayarak açabilir.

Ders Notları
Uygulamalar
Ödevler
Asistan Seçimi
Dönem Öğrencileri

Açık Ders Kütüphanesi - Ders Notları

Hafta	Açıklama
-1	Slaytlar (Beyaz Fon) Vize 6 hafta
-1	Vize Sonuçları
0	Geçmiş yılların soruları
1	Temel Kavramlar
2	Veri Yapılarına Giriş
3	Veri Modelleri

Dosya
Ödev İşlemleri
Paylaşım

vy_forPrint.zip	Dosya İşlemleri	Paylaşım
vy_vize.zip	Dosya İşlemleri	Paylaşım
vy_odevlar.rar	Dosya İşlemleri	Paylaşım
Ders01.pdf	Dosya İşlemleri	Paylaşım
Ders02.zip	Dosya İşlemleri	Paylaşım
Ders03.zip	Dosya İşlemleri	Paylaşım

Yeni kayıt ekle
X

Hafta

Açıklama

İptal Kaydet

Akademisyen, sağ üsteki "+Yeni Kayıt" bölümünden kayıt ekleyebilir veya kayıt düzenleme ile kayıt üzerinde değişiklik yapabilir. Ders not bilgileri girildikten sonra derse ait dosyanızı sisteme yükleyebilirsiniz. Akademisyen dosyasını yüklemek için "Dosya İşlemleri" bölümüne tıklar.

\*Dosya boyutu 10 MB ile sınırlıdır. Sıkıştırma programları dosyanızı küçültebilir veya büyük boyutlu resimlerinizi resim düzenleme programları ile boyutlarını

küçültebilirsiniz.

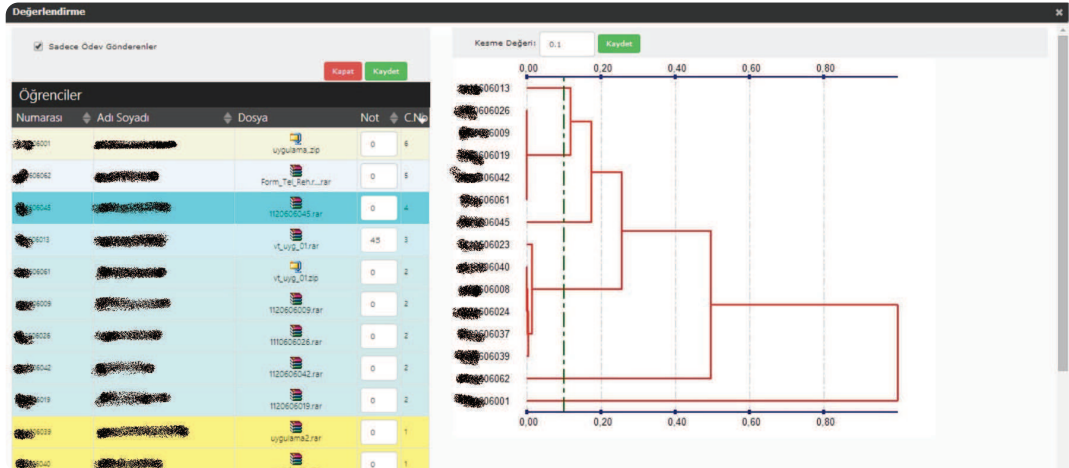
\*\*Ders notlarınızı, uygulamalarınızı ve ödevlerinizi tek tıklama ile dünya ve öğrenci ile paylaşabilirsiniz.

## Uygulama ve Ödev

Akademisyen, uygulama ve ödevler için süre verebilir. Süre verildikten sonra uygulama/ödevde geri sayım işlemi başlar. Geri sayım bittiğinde otomatik değerlendirme işlemi başlar. Değerlendirme işleminde arka planda bir program öğrencilerin ne kadar benzer uygulama veya ödev yaptıklarını değerlendirir. Bu değerlendirme işlemi ödev sayısına, dosya boyutlarına göre farklılık gösterebilir. Ortalama 5 dakika içinde tamamlanır. Ardından sonuçlar ikonuna tıklayarak gönderilen ödevleri ve benzerlik durumlarını görebilirsiniz.

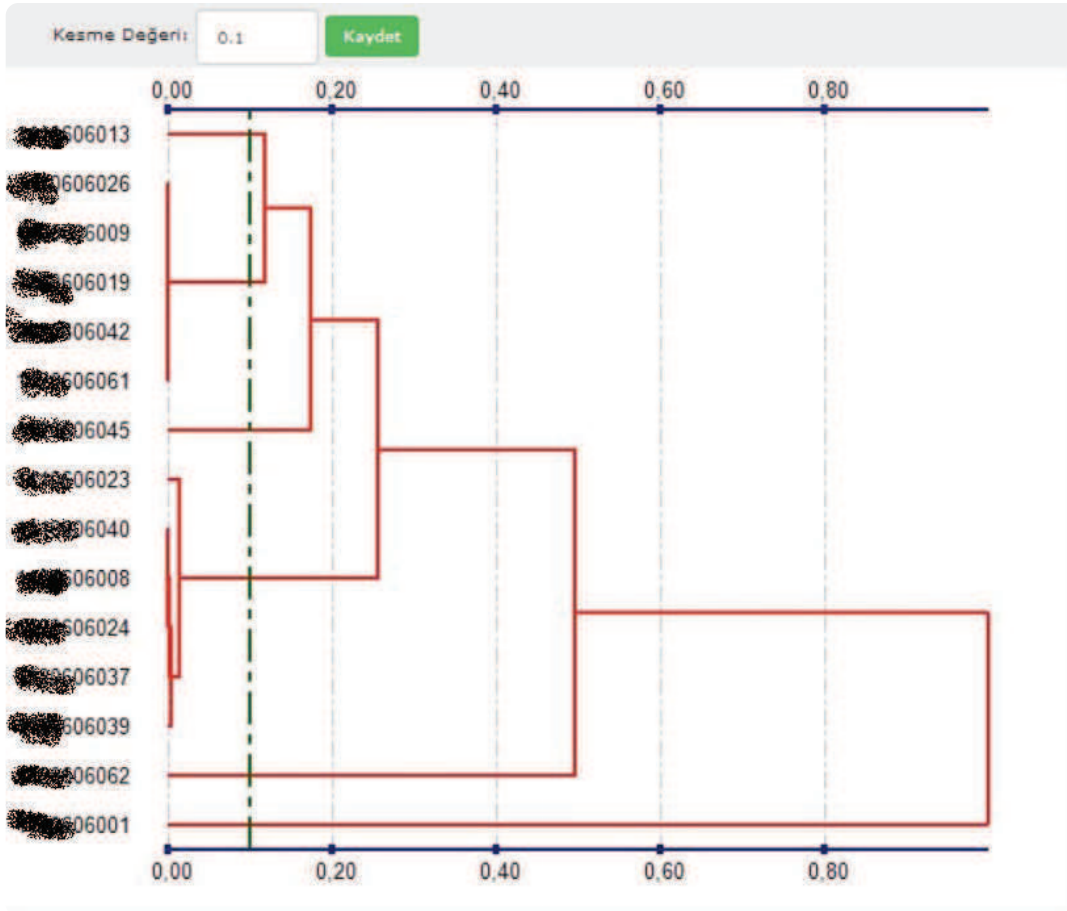
## Uygulama ve Ödev Değerlendirme

Uygulama/Ödev değerlendirme hiyerarşik kümeleme yöntemi kullanılmıştır. Görsel olması açısından dendrogram ile gösterim yapılmıştır. Dendrogramdaki kesme değerine göre öğrenciler kümelenecek bir şekilde gösterilir. Öğrenci ödevlerini kendi bilgisayarınıza indirebilirsiniz.



## Dendrogram

Dendrogram, hangi öğrencinin hangi öğrenciye yakın uygulama/ödev yaptığı ve benzerlik seviyesini gösterir. 0 (sıfır) değerine yakınlık benzerliğin çok olduğunu gösterir. Örneğin aşağıdaki örnekte (026, 009, 019, 042, 061) 5 öğrenci aynı ödevi gönderirken 013 bu öğrencilere yakın bir ödev hazırlamıştır. İşte bu noktada ödevin türüne göre 013 benzer ödevde yapmış olabilir, benzer olmayan bir ödevde yapmış olabilir. Bu noktada seçim akademisyene kalmaktadır. Akademisyen, bir kesme değeri girerek küme sayısını belirler ve sol bölümden öğrenci noktalarını verebilir. Öğrenci noktaları yazıldıktan sonra kaydet ikonuna basmayı unutmayın. Ayrıca, "Beni Hatırla" yı tıklamadıysanız 1200 saniye sonra sayfanın kapatılacağını unutmayın. "Beni hatırla" tıkladığınızda oturumuz hep açık kalır, güvenlik açısından kendi bilgisayarınız hariç bu sekmeyi kullanmanız önerilmez. Veya işiniz bittiğinde "Oturum Kapat" ile oturumu sonlandırabilirsiniz.



## Asistan Seçimi

Asistan seçimi bölümünde dersinize asistan atadığınızda asistanınızda sizinle aynı işlemleri yapabilir. Ders notu yükleyebilir, uygulama veya ödev sonuçlarını görebilir.

## Dönem Öğrencileri

Dönem öğrencileri otomatik olarak yüklenir. Ancak dönem içinde değişiklikler olabilir. Dersi dışarıdan bir öğrenciyi dahil etmek isteyebilirsiniz. Bu durumda, dönem öğrencileri bölümünden öğrenci ekleyebilir veya silebilirsiniz. Dönemin

ilk haftalarında öğrenciler sıklıkla ders değişimi yapabilmektedir. Bu durumda, bu bölümde altta yer alan "Öğrencileri NKU Web Servisi üzerinden Güncelle" ile ders alan öğrencileri güncelleyebilirsiniz.

# İletişim

---

Önerileriniz ve sorularınız için erdincuzun (at) nku.edu.tr ile irtibata geçebilirsiniz.

---

© 2015 - NKÜ Bilgisayar Mühendisliği Bölümü