Metin Madenciliği Kullanarak Yazılım Kullanımına Dair Bulguların Elde Edilmesi

Deniz Kılınç[1], Fatma Bozyiğit[1], Akın Özçift[1], Fatih Yücalar[1], Emin Borandağ[1]

[1] Celal Bayar Üniversitesi, Hasan Ferdi Turgutlu Teknoloji Fakültesi
Yazılım Mühendisliği Bölümü, Manisa, Türkiye
deniz.kilinc@cbu.edu.tr, fatma.bozyigit@cbu.edu.tr,
akin.ozcift@cbu.edu.tr, fatih.yucalar@cbu.edu.tr,
emin.borandag@cbu.edu.tr

Özet. Yazılım teknolojileri hızla ilerlemekte ve buna paralel olarak hem kamu alanında hem de özel sektörde gerçekleştirilen yazılım projelerinin sayısı artmaktadır. Yazılım otomasyon projelerinden elde edilen en büyük çıktılardan birisi kuşkusuz ki üretilen verilerdir. Yüksek boyutlu, anlaşılması güç bu verilerin işlenerek, daha anlamlı ve yönlendirici verilere dönüştürülmesi önemli bir ihtiyaçtır. Elde edilen veriler, veri ve metin madenciliği teknikleri işe analiz edilebilmektedir. Analiz sonuçlarına göre mevcut yazılım sistemlerinin daha kullanışlı hale gelmesi sağlanabilir. Bu çalışmada, Celal Bayar Üniversitesi (CBU) tarafından kullanılan Üniversite Bilgi Sisteminin (UBS) yazılım otomasyonu ele alınmıştır. Veri kaynağı olarak, öğrencilerin UBS sistemi aracılığı ile dönemsel ders kayıtlanmalarını yaparken ortaya çıkan sorunları danışman hocalarına bildirmek için kullandıkları mesajlaşma modülünden faydalanılmıştır. Her bir öğrenciye ve danışmana ait toplam 110,192 adet mesaj, metin madenciliği teknikleri (kelimelerin birlikte geçme durumları, K-means kümeleme algoritması) kullanılarak anlamlı hale getirilmiş ve kayıtlanma dönemlerinde en çok hangi sorunlarla karşılaşıldığı çıkarımı yapılarak alınabilecek önlemler tartışılmıştır.

Anahtar Kelimeler. Yazılım kullanımı, metin madenciliği, kümeleme, K-means

1 Giriş

Sürekli gelişen ve gün geçtikçe kendini yenilemekte olan teknoloji ile birlikte yazılım alanında gerçekleştirilen uygulamalar da çoğalmaktadır. Kullanılmakta olan uygulamaların değişen kullanıcı ihtiyaçlarını devamlı olarak karşılıyor olması ve sorunsuz bir şekilde çalışması beklenmektedir. Bu amaçla mevcut sistemlerin sürekli olarak güncelliğinin sağlanması, etkinliğinin arttırılarak dinamikliğinin sağlanması gereklidir.

Yazılım projelerinin etkin kullanımı ile birlikte veri tabanlarında bu sistemlere ait depolanan veri miktarları da büyük boyutlara ulaşmaktadır. Uygulamalarda kullanıcıdan alınan girdiler sistemi değerlendirmeye yarayan bildirimler olarak

değerlendirilebilir. Metodolojik yöntemlerle mevcut durum analizi yaparak sistemlerin güçlü ve zayıf yönlerinin belirlenmesi, toplanan veriler sayesinde sağlanabilmektedir. Böylelikle elde edilen veriler sayesinde sistemin durumu incelenebilir, iyileştirilmesi veya güncellenmesi gereken kısımlar netleştirilebilir.

Yazılım kullanımına dair elde edilen bulguların incelenmesi, geliştirilmiş yazılım sistemlerinin değerlendirilmesi için önemli ihtiyaçtır. Bu ihtiyacı karşılamak amacı ile elde edilen büyük veri yığınlarından gizli örüntüleri ve ilişkileri ortaya çıkarmaya yarayan veri madenciliği tekniklerinden yararlanılmaktadır. Toplanan veriler metin tabanlı ise eldeki yapısal olmayan bilgiler veri madenciliğindeki gibi alıp doğrudan kullanılamaz. Metin madenciliği, yapısal olmayan bu verileri analiz eder ve veri madenciliğinde kullanılabilecek formata dönüştürür. Böylelikle sisteme dışarıdan gelen veriler üzerinden mevcut yazılımı test etmeye yarayan önemli girdiler elde edilebilir.

Bu çalışmada, Celal Bayar Üniversitesi (CBÜ)[1] tarafından kullanılan Üniversite Bilgi Sistemi (UBS)[2] yazılım otomasyonu ve ürettiği metin tabanlı veriler analiz edilmek üzere ele alınmıştır. Tüm üniversitelerde olduğu gibi CBÜ'de de öğrencilere, öğretim hayatları boyunca "Danışmanlık Hizmeti" verilmektedir. Öğretim üyeleri tarafından verilen bu hizmetin önemli adımlarından bir tanesi, öğrencilerin CBÜ UBS üzerinden dönemsel ders kayıtlanmalarını yaparken kullandıkları mesajlaşma modülü aracılığı ile gerçekleşmektedir. Öğrenciler, kayıtlanma modülünde karşılaştıkları teknik problemleri ve ders seçimi, çakışma vb. konulardaki sorunlarını, metin tabanlı mesajlaşma uygulaması kullanarak öğretim üyelerine iletmekte ve öğretim üyeleri de bu sorunları öğrencileri yönlendirerek çözmektedirler. Bu adımların tamamında metin içerikli mesajlar karşılıklı olarak gönderilmektedir. Çalışmanın temel amacı, mesajlaşma metinlerinin metin madenciliği teknikleri kullanılarak analiz edilmesi ve UBS kayıtlanma modülünde iyileştirilebilecek noktaların saptanmaya çalışılmasıdır.

Bildirinin devamında, ikinci bölümde çalışmanın yöntemi ile ilgili ayrıntılı bilgi verilmiştir. Üçüncü bölümde deneysel veri setleri ve bu veri seti üzerine uygulanan metotlar (teknik detaylar, dikkat edilen yazılım geliştirme metrikleri) ele alınmıştır. Dördüncü bölümde sonuç ve tartışmaya yer verilirken, gelecekte yapılması planlanan çalışmalardan da bahsedilmiştir.

2 Materyal ve Metotlar

2.1 Veri ve Metin Madenciliği

Yazılım Mühendisliği süreçleri için veri, bir uygulama üzerinde çalışmak ve yeni uygulamalar geliştirmek için çok önemlidir. Elde edilen verilerden yapılacak analizlerde değerlendirmek üzere önemli ipuçları elde etmek ve bu doğrultuda veriyi işe yarar hale getirmek için Veri Madenciliği yöntemlerinden faydalanılmaktadır. Veri madenciliği büyük hacimli veriler içerisinde değerli bilgiyi keşfetme işlemidir [3].

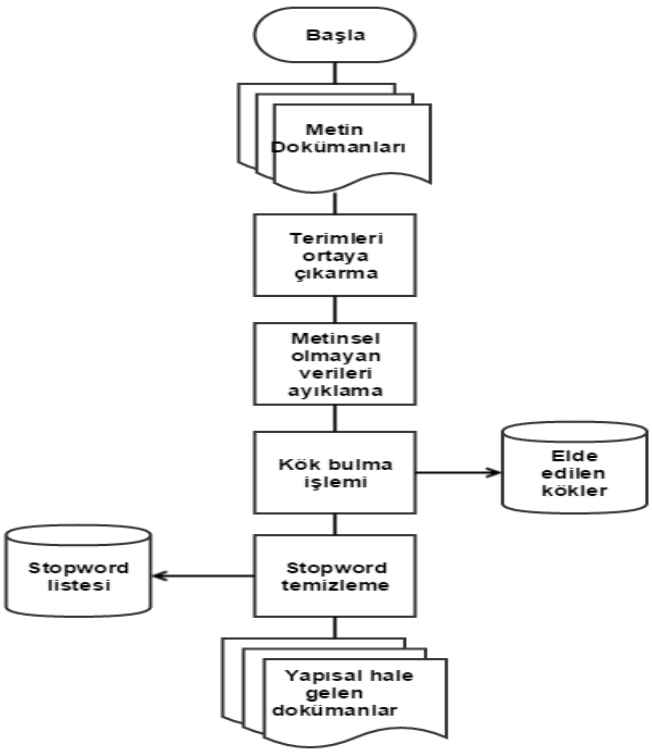
Yapısal veri, bir yapı içerisinde organize edilebilen ve bundan dolayı tanımlanabilen veri için kullanılan bir terimdir [4]. Bu tip veriler üzerinden çıkarımlar yapmak ve bilgi edinmek veri madenciliği tekniklerinin kullanımı ile

gerçekleştirilmektedir. Ancak verinin tipi analiz işlemine uygun yapıda veya uygun olmayan yapıda bir metin şeklinde olabilir.

Metin madenciliği genellikle yapısal halde olmayan metinlerden ilgi çekici bilgi ve anlam çıkarma işlemi olarak tanımlanır [5]. Metin halinde bir verinin üzerinden bilgi çıkarımı yapabilmek için ise bazı işlemlerin gerçekleştirilmesi gerekmektedir. Bu işlemler ile birlikte yapısal olmayan metinsel veriler yapısal bir hale dönüşür. Böylelikle metinsel veriler veri madenciliği tekniklerinin uygulanabileceği uygun formata dönüştürülmüş olur.

2.2 Ön İşleme

Metinsel veriler üzerinde analiz ve çıkarımlar yapabilmek için ilk olarak bu veriler üzerinde birtakım işlemlerin yapılması gerekmektedir. Bu işlemler uygulanırken metin madenciliğinde önemli bir yere sahip olan ön işleme tekniklerinden faydalanılmaktadır. Genellikle doğal dil kullanılarak yazılmış dokümanlar üzerinde çalışılan metin madenciliği alanında ön işleme aşaması, veri temizlemesinin yanında veriyi uygun formata getirme işlemini de gerçekleştirmektedir [6]. Şekil 1'de metinsel verileri yapısal hale getirmek için kullanılan ön işleme adımları gösterilmiştir. Metin madenciliği tekniklerini kullanarak ön işlemden geçirilen ve yapısal hale gelen veriler, veri madenciliği tekniklerinde kullanılmaya hazır haldedir.



Şekil. 1. Ön işleme aşamaları

Information Retrieval (IR) [7] uygulamalarında temel işlemlerden biri, frekansı yüksek olan ve doküman ayırt etmede bir etkisi olmayan kelimeleri elimine etmektir [8]. Şekil 1'de de görüldüğü gibi ön işlemede genellikle ilk olarak metinsel olmayan noktalama işaretleri, boşluklar ve numerik değerler gibi bazı stop word'leri ayıklama işlemi yapılır. Daha sonra doküman içerisindeki kelime grupları elde edilir. Kelimeleri elde etmek ve bu kelimeleri köklerine ayırmak için stemming, lemmatization gibi işlemler tercih edilebilmektedir. Son aşamada ise doküman içerisinde işe yaramayacak terimler belirlenir ve bu terimlerin listesi yapılır. Hazırlanan listeye göre dokümanda bulunması istenmeyen gereksiz veriler temizlenir. Bu kelimeler İngilizce'de "or", "and", "am/is/are" gibi tek başına bir anlam belirtmeyen terimler olabilirken, Türkçede "ve", "veya", "ama" gibi kelimelerdir.

2.3 Matrislerin Oluşturulması

Bu modelde ilk olarak, metin dokümanlarına ait her bir terimin ağırlığı hesaplanır. Bir terim doküman içerisinde geçmiyor ise ağırlık değeri 0, geçiyor ise 0 dan farklıdır [9]. Ağırlık hesaplama işleminde IR tekniklerinden olan $TF \times IDF$ methodundan faydanılabilmektedir.

TF bir terimin dokuman içerisinde geçme frekansını verir iken, IDF değeri bu terimin diğer dokümanlarda geçme sıklığını bildirmektedir. TF ve IDF değerlerinin denklemleri Formül 1 ve Formül 2 de gösterilmiştir.

$$TF(t) = \frac{t \text{ teriminin bir dokümanda geçme sayısı}}{\text{Dokumandaki toplam terim sayısı}} \quad (1)$$

$$IDF(t) = log_e \frac{Vektor\ modelindeki\ Toplam\ dokuman\ sayısı}{\text{İçerisinde } t \text{ terimini bulunduran toplam dokuman sayısı}} \quad (2)$$

Her terimin ağırlığı hesaplandıktan sonra doküman terim matrisleri oluşturulmaktadır. Şekil 2'de örneği verilen doküman terim matrisinde D veri seti içerisindeki dokümanları, T dokümanlarda yer alan terimleri, d ise bu terimlerin ağırlıklarını göstermek için kullanılmıştır [10]. Bu matrisler üzerinde gerekli veri madenciliği algoritmaları kullanarak veriler üzerinden anlam çıkarma işlemi yapılmaktadır.

$$\begin{array}{c|cccc} & T_1 & T_2 & \dots & T_3 \\ D_1 & d_{11} & d_{12} & \dots & d_{1t} \\ D_2 & d_{21} & d_{22} & \dots & d_{2t} \\ \dots & \dots & \dots & \dots & \dots \\ D_n & d_{n1} & d_{n2} & \dots & d_{nt} \end{array}$$

Şekil 2. Doküman terim matris gösterimi

2.4 Kümeleme

Kümeleme analizi bir veri kümesindeki bilgileri belirli yakınlık kriterlerine göre gruplara ayırma işlemidir [11]. K-means algoritması kümeleme işlemlerinde kullanılan denetimsiz yöntemlerden birisidir. Algoritmanın amacı özellik çıkarımı yapılmış verilerin her birinin hangi kümeye dâhil olduğunu bulmaktır. Bu doğrultuda n adet veri k adet kümeye dağıtılmaktadır. Böylece her verinin benzerlik oranının en çok olduğu veriler ile aynı kümede olması sağlanır.

Kümeleme algoritmalarına göre öncelikle verilerin dâhil edileceği kümeler ve kümelerin merkez değerleri belirlenir. Hangi kümeye ait olduğuna karar verilmesi gereken verilerin her bir küme merkezine olan uzaklıkları bulunur. Bu uzaklık değerleri hesaplanırken genellikle Euclidian Uzaklık formülü kullanılmaktadır. Formül 3'te (x, y) ve (a, b)noktaları arasındaki uzaklık değerini hesaplamaya yarayan Euclidian Uzaklık denklemi görülmektedir.

$$uzaklık\big((x,y),(a,b)\big) = \sqrt{(x-a)^2 + (y-b)^2} \tag{3}$$

Elde edilen uzaklık değerlerine göre uygun kümelere yerleştirilen nesnelerin ortalama değerleri bulunur ve içinde bulundukları kümenin merkez noktası bu değer ile değiştirilir. Merkez nokta sabitleninceye kadar bu işlem tekrarlanır. Son aşamada kararlı küme merkez noktaları ve küme elemanları elde edilmiş olur.

K-means yönteminin doğruluğu test edilirken yaygın olarak karesel hata ölçütü SSE kullanılır [12]. Bu değeri elde etmek için, nesnelerin bulundukları kümenin merkez noktalarına olan uzaklıklarının karelerinin toplamı Formül 4'te görülen eşitlik ile hesaplanmaktadır [13].

$$SSE = \sum_{k=1}^{K} \sum_{x \in G_k} uzaklık^2(x, c_k) \tag{4}$$

Bu kriterleme sonucu, k tane kümenin olabildiğince yoğun ve birbirinden ayrı sonuçlanması hedeflenmeye çalışılır [12]. Sonuç olarak, belirlenen kümelerdeki küme içi benzerlik oranlarının büyük, kümeler arası benzerlik oranlarının düşük olup olmadığını bu hata değeri ile kontrol edilmiş olur. Hata değerinin küçük olması K-means yöntemine göre beklenen sonucun elde edildiğini ve kümeleme işleminin tutarlı yapıldığını bildirmektedir.

3 Deneysel Çalışmalar

3.1 Veri Seti ve Ön İşleme

Bu çalışmada kullanılmak üzere CBÜ UBS sisteminde yer alan, 2014-2015 güz dönemi kayıtlanmasına ait 110,192 adet mesaj kullanılarak bir veri seti oluşturulmuştur. Bu mesajlar öğrencilerin sistem üzerinden öğretim üyelerine soru sormak için gönderdikleri ve bu sorulara gelen cevaplardan oluşmaktadır. Bu mesajlar üzerinde aşağıda belirtilen ön işlemler uygulanarak, veri temizliği yapılmış ve analiz edilecek mesaj sayısı 98,575'e düşmüştür.

- Küçük harfe dönüştürme: Tüm metinler küçük harfe dönüştürülmüştür.
- Boşluk ve noktalama işareti silme: İçerisinde sadece boşluk karakteri veya noktalama işaretleri geçen mesajları silinmiştir.
- Karakter yer değiştirme: CBÜ UBS sisteminin cep telefonu gibi akıllı cihazlar üzerinden yoğun bir kullanımı olduğu için Türkçe ve İngilizce karakterlerin karışık olarak kullanıldığı görülmüş ve tüm metinlerdeki Türkçe karakterler, İngilizce karaktere dönüştürülmüştür.
- Minimum kelime uzunluğu elemesi: Kelimelerin minimum uzunluğu 3 olarak belirlenmiştir. Metin içerisindeki bir ve iki harf uzunluğunda olan kelimeler silinmiştir.
- Stopword olan kelimeleri eleme: Özel bir stopword sözlüğü oluşturularak, stopword'ler atılmıştır. (merhaba, iyi akşamlar, günaydın, hayırlı olsun)
- Gövdeleme: Türkçenin sondan eklemeli bir dil olması sebebi kelimede belirli bir uzunluğu temel alan Fixed Prefix Metot'un diğer gövdeleme işlemleri ile benzer başarı oranı verdiği gözlemlenmiştir [8]. Bu sebeple, çalışmada özel bir gövdeleme algoritması kullanılmamıştır. Gövdeleme işlemi için metinlerdeki kelimelerin ilk 7 karakteri alınmıştır.

3.2 Uygulanan Yöntemler ve Sonuçlar

Ön işleme adımının tamamlanmasının ardından dokuman-kelime ve kelime-dokuman matrisleri oluşturulmuştur. Bu matrisler kullanılarak, kelime geçme sıklığı bulunması, kelime-kelime-ilişki matrisinin oluşturulması ve K-means kümeleme algoritması gibi metin madenciliği teknikleri uygulanabilir hale gelmiştir. İlk olarak, yüksek frekanslı kelimeler bulunmuş ve bu bilgiden yola çıkarak, yüksek frekanslı kelimelerin meydana getirdiği kelime bulutu oluşturulmuş ve Şekil 3'teki görünüm elde edilmiştir. Ön işleme sonucunda 98,575 adet mesaj içeren veri setinde bir kelimenin yüksek frekanslı olarak seçilebilmesi için tüm mesajlarda en az 20 kez geçmesi kuralı uygulanmıştır.

Şekil 3. Ön işleme ardından oluşan kelime bulutu

Şekil 3'teki kelime bulutu görünümünden yola çıkılarak kelimeler arası birlikte geçme sıklıkları üzerinde denemeler yapılmış ve kelime-kelime-ilişki matrisi oluşturulmuştur. Tablo 1, 2 ve 3'te yüksek frekanslı 3 kelime ve bu kelimelere en yakın kelimeler, yakınlık oranları ile birlikte gösterilmektedir.

Tablo 1. "ders" kelimesi ile birlikte geçen kelimeler ve yakınlık oranları

"ders" Kelimesi	**Yakınlık oranı**
Secimi	0.39
Seçilme	0.17
Seçmeli	0.15
Eksik	0.14

Tablo 1'de "ders" kelimesine en fazla oranda yakınlık gösteren 4 kelime gösterilmiştir. Bu kelimeleri art arda sıraladığımızda "ders secimi eksik", "ders seçilme eksik", "seçmeli ders eksik" gibi farklı içeriklere sahip mesajlar üretilebilmektedir. Bu tablodan yola çıkılarak, seçmeli derslerin seçimi ile ilgili öğrencilerin problem yaşadığı ve/veya öğrencilerin karşısına eksik seçmeli derslerin geldiği gibi yorumların yapılması mümkündür.

Tablo 2. "cakisan" kelimesi ile birlikte geçen kelimeler ve yakınlık oranları

"cakisan" Kelimesi	**Yakınlık oranı**
Dersler	0.29
Var	0.20
Napcaz	0.14
Gönderi	0.12

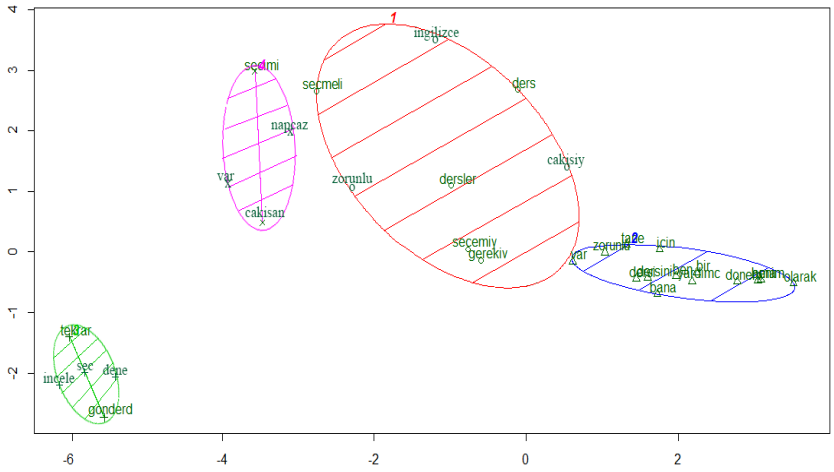
Tablo 2'de "cakisan" kelimesine en fazla oranda yakınlık gösteren 4 kelime görülmektedir. Tablo 1'in yorumlanma mantığı aynı şekilde Tablo 2'ye uygulandığında, "cakisan dersler var napcaz" gibi çarpıcı ve muhtemelen öğrenciler tarafından kullanılan bir mesaj içeriği üretilebilmektedir. Tablo 2'den yola çıkılarak, öğrencilerin ders çakışması problemi ile fazlaca karşılaştıkları ve bu durum karşısında ne yapacaklarını bilmedikleri yorumu yapılabilir.

Tablo 3. "tekrar" kelimesi ile birlikte geçen kelimeler ve yakınlık oranları

"tekrar" Kelimesi	**Yakınlık oranı**
Sec	0.39
Dene	0.17
Bak	0.15
Düzenle	0.14
İncele	0.13
Gönder	0.12

Tablo 3'te de "tekrar" kelimesi için Tablo 1 ve 2'deki yorumlama mantığı aynı şekilde uygulandığında "tekrar sec düzenle gonder", "tekrar gonder", "tekrar sec" gibi derslerin tekrar seçilmesi ve düzenlenip geri gönderilmesi gibi işlemlerin öğrenciler tarafından sıkça gerçekleştirildiği şeklinde yorum yapılabilir.

Kelime-kelime-ilişki matrisinin oluşturulmasına ve en sık geçen kelimelerin yakınlık incelenmesi ek olarak K-means kümeleme analizi de yapılmıştır. Şekil 4'te veri setinde bulunan öğrenci ve öğretim üyeleri mesajlaşmalarına ait dokümanları analizi sonucu oluşan 4 kümeye ait bilgiler grafiksel olarak gösterilmiştir. Kümeler incelendiğinde Tablo 1, 2 ve 3'teki ile tutarlı yakınlık oranına sahip kelimelerin aynı kümelerde toplandığı gözlemlenmiştir.



Şekil 4. K-means kümele analiz sonucu

4 Tartışma ve Öneriler

Bölüm 3'teki Tablo 1, 2 ve 3'teki analizler yüksek frekanslı tüm kelimeler üzerinde yapıldıktan sonra aşağıdaki yorumlamalar yapılmış ve gelecek çalışmaları önerilmiştir.

- Ders çakışması ile ilgili UBS yazılımı daha yönlendirici olabilir. Kayıt aşamasında öğrenciye önerilerde bulunabilir.
- Yanlışlıkla eksik seçmeli ders seçerek, danışmana onaya yollama adımında iyileştirme yapılabilir.
- Seçmeli havuzların fazla ve karmaşık olması, danışman ve öğrenci arasında fazla sayıda mesajlaşmaya neden olmaktadır. Seçmeli derslerle ilgili üniversite genelinde standartlar gözden geçirilebilir.
- Yapılan analizleri doğrulamak amacıyla öğrencilere analiz sonuçları ve yazılım kullanılabilirliği ile ilgili anket düzenlenebilir.
- Gövdeleme işlemi için kelimelerin ilk 7 karakterinin seçilmesi yerine Türkçe gövdeleme işlemini destekleyen bir kütüphane kullanılabilir.

Kaynaklar

1. Celal Bayar Üniversitesi Tanıtımı, http://www.cbu.edu.tr
2. Celal Bayar Üniversitesi Bilgi Sistemi, https://ubs.cbu.edu.tr
3. Azzalini, A., Scarpa, B., Walton, G.: Data Analysis and Data Mining: An Introduction. Oxford University Press, New York (2012).
4. Dolgun, M., Özdemir, T., Oğuz, D.: Veri madenciliğinde yapısal olmayan verinin analizi: Metin ve Web madenciliği. İstatistikçiler Dergisi, 2, 48-59 (2008).
5. Hotho, A., Nurnberger, A., Paaß, G.: A Brief Survey of Text Mining. LDV Forum - GLDV Journal for Computational Linguistics and Language Technology 20(1), 19-62 (May 2005).
6. Fieldman, R., Sanger J.: The text mining handbook advanced approaches in advanced analyzing unstructured data. Cambridge University Press (December 2006).
7. Kılınç, D., Bozyiğit, F., Kut, A., Kaya, M.: Overview of Source Code Plagiarism in Programming Courses. International Journal of Soft Computing and Engineering (IJSCE), 5(2), 79-85 (May 2015).
8. Can F., Kocberber S., Balcik E., Kaynak C., Ocalan HC., and Vursavas OM.: Information retrieval on Turkish texts. Journal of the American Society for Information Science and Technology, 59(3), 407-421 (2008).
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for information retrieval. Journal of the American Society for Information Science, 18(11), 613-620 (1975).
10. Bozyiğit F.: Analyzing source code and detecting similarities. MSc. Thesis, Dokuz Eylül University (2015).
11. Sarıman, G.: Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 15(3), 192-202 (2011)
12. Işık, M., Çamurcu, A.Y.: K-Means, K-Medoids Ve Bulanık C-Means Algoritmalarının Uygulamalı Olarak Performanslarının Tespiti. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, s. 31-45 (2007).
13. Pang-Ning, T., Michael, S., and Kumar, V. :Introduction to Data Mining. Pearson Addison Wesley (2005).