İlişkisel Veri Tabanlarında Mükerrer Kayıtların Makine Öğrenmesiyle Tespiti
Near Duplicate Detection in Relational Databases

Ahmet Tuğrul Bayrak[1], Aykut İnan Yılmaz[1], Kemal Burak Yılmaz[1],Remzi Düzağaç[1]
Veri Bilimi ve Analitik Bölümü[1]
ETSTUR[1]
İstanbul, Türkiye[1]
tugrul.bayrak, inan.yilmaz, burak.yilmaz,remzi.duzagac@etstur.com[1]
Olcay Taner Yıldız[2]
Işık Üniversitesi Bilgisayar Mühendisliği Bölümü[2]
olcay.yildiz@isikun.edu.tr[2]

***Özetçe* —Veri miktarının artışına paralel olarak, ilişkisel veri tabanlarında mükerrer kayıtlar da artmaktadır. Artan bu kayıtlar kullanıldıkları rapor veya analizlerde tutarsızlığa sebep olabilmektedir. Bu sorunu en aza indirgemek için yaptığımız çalışmada, kayıtların birbirlerine olan benzerlikleri ve alan uzmanlık bilgisiyle belirlenen ağırlıklar, öznitelik olarak kullanılarak makine öğrenmesi algoritmaları ile mükerrer kayıtların bulunması hedeflenmiştir. Yapılan işlem sonucunda 9301467 satır veride 28412 mükerrer çift tespit edilmiştir. Bulunan bu mükerrer kayıtlar veri kaynağından temizlenerek verinin daha tutarlı hale gelmesi sağlanmaktadır.**

Anahtar Kelimeler—Makine Öğrenmesi, Benzerlik Fonksiyonları, Mükerrer Kayıt Tespiti

***Abstract*—While data amount increases, number of duplicate records in relational databases increase gradually. The duplicate records might cause inconsistency on reports and analyzes. To reduce the effects of this problem, we aim to detect duplicate records using machine learning algorithms with features that are produced by similarity of the records. We achieved to detect 28412 duplicate records in 9301467 records. The detected duplicate rows are removed from the data source and the data become more consistent.**

Keywords—Machine Learning, Similarity Functions, Duplicate Record Detection

I. GİRİŞ

Teknolojinin hızla ilerlemesi, bilgisayar kullanımının yaygınlaşması ve veri üreten ortamların artması, saklanan veri miktarını arttırmıştır. Gün geçtikçe büyüyen veriler çeşitli şekillerde işlenmekte ve bu verilerden farklı sonuçlar elde edilebilmektedir. Bu yüzden, kaynak sistemlerden her türlü veri alınıp veri tabanlarında saklanmaktadır. Saklanan bu verilerin uygun şekilde depolanması ve temizliği son derece önemlidir. Günümüzde veri tabanlarındaki en büyük sorunlardan biri olan veri kirliliğinin sebeplerinden bir tanesi, veri kümesi içerisinde tekrarlayan kayıtların olmasıdır. Bu durum verinin gereksiz yer kaplamasına, kalitesinin düşmesine ve hedef veriye ulaşılmasında sorunlara yol açmaktadır.

Depolanan farklı veri kayıtları birebir mükerrer olabileceği gibi, bu kayıtların sadece bazı değerlerinin aynı olması da kayıtların aynı varlığı temsil etmesine sebep olabilir. Tüm değerleri aynı olan mükerrer kayıtların tespiti daha kolay bir problem olup, bu kayıtlar standart veri tabanı işlemleriyle kolayca temizlenebilmektedir. Öte yandan, bazı değerleri aynı olan kayıtların mükerrer olup olmadığını yorumunu yapmak ise daha karmaşık bir problemdir.

Mükerrer kayıtların tespit edilmesi problemi ilk olarak Newbombe tarafından farklı zamanlarda alınan, birebir aynı ya da benzer medikal kayıtlar için tanımlanmıştır. Bu yöntemde kayıtlar karşılaştırılırken, soy isimler ilk olarak *soundex* kod olarak bilinen, ismin ilk harfi ve üç basamaklı bir sayıyla sesçil kodlara dönüştürülmüş, karşılaştırma bu kodlara göre yapılmıştır [1]. Hernandez ve Stalfo [2] potansiyel adayları limitlemek için uzaklık hesaplaması gerektiren sıralanmış komşuluk yöntemini geliştirmişken, McCallum kayıtlatın benzerlikleri ölçüldükten sonra, bu benzerliklere göre uygulanan bir öbekleme algoritması önermiştir [3]. Cohen ve Richman mükerrer kayıt tespiti için, benzerliklerin beraber kullanıldığı

978-1-5386-1501-0/18/$31.00

bir çatı önermişlerdir [4].

Bu çalışmamızda, ilişkisel veri tabanı tablolarına farklı kanallardan gelen, birebir aynı olan ya da birbirine çok benzeyen ve aynı varlığı temsil eden kayıtların tespiti yapılmıştır. Bu işlem sırasında, alan uzmanlık bilgisine göre yapılan işaretlemeler, çeşitli benzerlik fonksiyonlarıyla zenginleştirilip işaretlendikten sonra, benzer durumdaki mükerrer kayıtlar makine öğrenmesi yöntemleri kullanılarak bulunmuştur.

Çalışmada, II. bölümün veri ön işleme kısmında eldeki verinin hangi işlemlerden geçirildiğinden bahsedilmiştir. Verinin hazırlanması kısmında, mükerrer olduğu düşünülen aday ikililerin nasıl eşleştirildiği açıklanmıştır. III. bölümde, ilk olarak verideki benzerliğin nasıl ölçüldüğü ve makine öğrenmesi adımı için gerekli verilerin nasıl seçildiği anlatılmıştır. Sonrasında ise, hazırlanan bu veri üzerinden makine öğrenmesiyle aday çiftlerin bulunma yönteminden bahsedilmiştir.

II. VERI

A. Veri Ön İşleme

Büyük sistemlerde kullanılan canlı veri genel olarak belirli miktarda eksik, tutarsız veya gürültülü kayıtlar içermektedir. Bu yüzden, eldeki veriler kullanılmadan önce bazı ön işlemlerden geçirilip temizlenmeli ve standart hale getirilmelidir [5]. Böylece veri kalitesi arttırılarak doğru çıktılar üretilir ve yapılacak çalışmaların daha sağlıklı sonuç vermesi hedeflenir.

Bu çalışmada ön işleme adımları olarak; kategorik verilerdeki harfler büyütülüp, satır sonu boşluğu gibi özel karakterler temizlenmiştir. Tarih bilgisinin içerdiği saat bilgisi atılarak bu öznitelik standart hale getirilmiştir. Telefon verisinde boşluklar temizlenmiş, ülke kodları eklenerek veri standart hale getirilmiştir. Belirlenen bu kurallara uymayan satırlardaki hücre değerleri doğrulamadan geçemeyip silinmiştir (TABLO I).

TABLO I: ÖN İŞLEME ADIMLARI

Öznitelik	Orjinal Veri	Ön İşlemeden Geçmiş Veri
CEP TELEFONU NUMARASI	+90 555 555 55 55	905555555555
CEP TELEFONU NUMARASI	(0544)-444-44-44	905444444444
DOĞUM TARİHİ	17/02/1987 12:00:00	17.02.1987
DOĞUM TARİHİ	01-01-1970	01.01.1970
İSİM	Tuğrul bayrak	TUGRUL BAYRAK
İSİM	Kemal Burak YILMAZ	KEMAL BURAK YILMAZ
ŞEHİR	KADIKÖY	ISTANBUL
ŞEHİR	AADANA	ADANA
ÜLKE	İstanbul	TURKIYE
ÜLKE	TR	TURKIYE

Şehir ve ülke bilgisinde yazım yanlışları, şehir yerine ilçe bilgisi yazılması gibi durumlar için değerler, benzerlik oranlarına bakılarak düzeltilmiş, benzemeyen değerlere ise atılmıştır. Bu işlem için şehirlerin doğru yazımlarının olduğu ilişkisel veri tabanı tablosu, verideki şehir bilgileri gruplandıktan sonra kartezyen çarpım ile aynı satırda yan yana getirilmiştir. Daha sonra, *Levenshtein* ve *cosine* benzerlikleri kullanılarak, bozuk olan şehir verileri en yüksek eşleşme sağlanan şehir bilgisiyle güncellenmiştir. Her iki benzerlikten de alt sınır olan 0.80 puanı alamayan şehir bilgileri ise silinmiştir (TABLO II). Aynı özelliklere sahip ülke bilgisi için de bu işlem tekrarlanmıştır.

TABLO II: ŞEHİR DÜZELTME İŞLEMİ

Şehir Adı	Aday Şehir	Doğru Şehir Değeri	*Levenshtein* Benzerlik Paunı	*Cosine* Benzerlik Puanı	Kullanılan Değer
AADANA	ADANA	ADANA	0,83	0,99	ADANA
ABANT	ABANT	BOLU	1,00	1,00	BOLU
ADABA	ADANA	ADANA	0,80	0,90	ADANA
AFYO	AFYON	AFYON KARA-HİSAR	0,80	0,89	AFYON KARA-HİSAR
ANTALLYA	ANTALYA	ANTALYA	0,87	0,97	ANTALYA
AAAAA	ADANA	ADANA	0,60	0,90	
AARAU	AKSARAY	AKSARAY	0,57	0,89	
ACC	CATALCA	ISTANBUL	0,40	0,80	
AVUSTRALYA	ANTALYA	ANTALYA	0,60	0,83	

B. Verinin Hazırlanması

Yapılan çalışmada 9301467 satırlık bir ilişkisel veri tabanı tablosunun verileri kullanılmıştır. Bu veriler müşteri izniyle kullanılıp, gizli tutulmaktadır. Ayrıca, TABLO I sembolik müşteri verilerinden oluşmaktadır. Bu veri kümesi içerisinde isim, soy isim, doğum yeri, doğum tarihi, cep telefonu numarası, ev telefonu numarası, şehir, ülke, cinsiyet gibi kolonlarla beraber, her bir satır için mükerrer olmayan müşteri numaraları bulunmaktadır. Bahsedildiği gibi, eldeki veride tüm müşteri numaraları tekil olarak bulunsa da bazı müşteri verileri farklı sistemlerden gelirken yeni müşteri numaraları ile birden fazla kez kaydedilebilmektedir.

İlişkisel veri tabanında, tablo yapısındaki mükerrer kayıtları bulmak amacıyla tüm kayıtları birbirleriyle karşılaştırmak için, ideal durumda her satır bir diğeriyle kartezyen çarpıma sokulmalıdır [6]. Ancak, eldeki verinin büyüklüğü ve kartezyen çarpım sonucunda 9385165^2 sayıda veri üretileceği düşünüldüğünde, tüm satırların birbiriyle karşılaştırılması zaman ve hafıza maliyetinin fazla olmasından dolayı mümkün olmamaktadır. Bu yüzden kartezyen çarpım yerine veriler benzerlik belirten kolonlar bazında birleştirilmiştir. Birleştirme işlemi için; cep telefonu numarası, tam isim gibi verideki önemli öznitelikler ayrı ayrı anahtar olarak alınıp, veri kendisiyle

kolon bazında birleştirilip sonuçlar satır bazında alt alta eklenmiştir. İsim alanlarının bulunduğu kolonların anahtar olarak verildiği birleştirmelerde, isim alanlarındaki harflerin yanlış sırayla yazıldığı durumlar düşünülerek bu alanlarda birleştirme öncesi sesli harfler atılıp kalan sessiz harfler alfabetik olarak sıralanarak birleştirme anahtarları oluşturulmuştur.

Farklı anahtarlarla yapılan birleştirme işlemleri sonucu aynı aday ikililer gelebileceği düşünülerek, birleştirilmiş veride birebir aynı olan aynı aday ikilileri içeren satırlar tekilleştirilmiştir. Böylece, n sayıdaki verinin kartezyen çarpımı ile oluşacak n^2 sayıdaki veri yerine, sadece önemli öznitelikleri eşleşmiş olan verilerden oluşan aday ikililer oluşturulmuştur. Bununla beraber, birleştirme işleminde (1) sorgusu ile içsel birleştirme yerine (2) sorgusu ile öz birleştirme kullanılmıştır.

```
SELECT A.oznitelik_02, B.oznitelik_02, A.oznitelik_01
FROM ornek_tablo A
INNER JOIN ornek_tablo B ON
A.oznitelik_01=B.oznitelik_01                    (1)
```

```
SELECT A.oznitelik_02, B.oznitelik_02, A.oznitelik_01
FROM ornek_tablo A, ornek_tablo B                (2)
WHERE A.satir_numarasi <> B.satir_numarasi
AND A.oznitelik_01 = B.oznitelik_01
```

Böylece, içsel birleştirmede tekil olmayan değerlerin kendileriyle olan eşleşmelerinden gelip, kontrol sonucu sonradan elenecek satırların ve aynı kayıtların kendileriyle eşleşmesinin önüne geçilmiştir.

TABLO III'teki örnek veri kümesi için *oznitelik_01* kolonu üzerinden yapılan içsel birleştirme sonucu, daha sonradan bir kısmı elenecek yirmi bir satır oluşurken, öz birleştirme sonucu on dört satır oluşmaktadır. Benzer şekilde, yapılan çalışmada oluşacak toplam 112765923 satır yerine, 10605319 satırlık analiz yapılacak veri oluşmuştur. Verinin büyüklüğü göz önüne alındığında içsel birleştirme yerine öz birleştirme kullanılmasının önemi anlaşılmaktadır.

TABLO III: ÖRNEK VERI KÜMESI

satır numarası	oznitelik_01	oznitelik_02
1	a	105
2	a	106
3	b	107
4	b	108
5	b	109
6	b	110
7	c	111

III. YÖNTEM

A. Verideki Benzerliğin Ölçülmesi

Verilerin bulunduğu tabloda, birleştirme işlemlerinden sonra aday mükerrer çiftlerin öznitelikleri kolon bazında yan yana getirilmiştir. Birleştirilmiş bu veriler alt alta eklenip bir araya getirildikten sonra tekilleştirilmiş olan tüm aday çiftler için öznitelikler arasındaki benzerliğe bakılmıştır. Cep telefonu numarası, kimlik numarası gibi öznitelikler için statik bir karşılaştırma ile eşit olup olmama durumlarına göre sırasıyla 1 ya da 0 puanı verilirken, kategorik olan isim özniteliğinde benzerlik fonksiyonları kullanılarak, benzerlik oranlarına göre [0, 1] aralığında puan verilmiştir. Benzerlik fonksiyonu olarak *Levenshtein* ve *cosine* benzerlikleri kullanılmıştır. Bu iki fonksiyonun beraber kullanılmasının sebebi, *Levenshtein* benzerliğinin karakter tabanlı, *cosine* benzerliğinin ise vektör tabanlı olmasıdır [7]. Benzerlik sonuçları [0, 1] arasında normalize edilerek kullanılmıştır (TABLO IV).

Öznitelik puanları, özniteliklerin bir kişiyi kendi başlarına temsil etme oranına göre bir ağırlıkla çarpılarak son hallerini almıştır. Ağırlıkların seçiminde alan uzmanlık bilgisi esas alınmıştır. Tam isim, cep telefonu numarası, şehir, doğum tarihi, cinsiyet öznitelikleri için verilen bu ağırlıklar sırasıyla; 8, 5, 3, 2, 1 şeklindedir. İsim özniteliğinin diğer özniteliklerden daha önemli olmasının sebebi, müşteri verilerinde isim dışındaki alanların diğer müşteri bilgilerine göre doldurulma ihtimalinin yüksek olmasıdır.

Problem gereği, tespit edilen aday ikililerin doğru olma oranının, yakalanan tüm doğru çiftlerin oranından daha önemli olması sebebiyle bahsedilen iki benzerlik fonksiyonu ayrı ayrı kullanılmış, isim öznitelikleri puanlanırken bu iki benzerlikten de en az 0.80 puan alması beklenmiştir. Yeterli isim puanını almayan aday ikililer, diğer özniteliklerden aldıkları puana bakılmaksızın elenmiştir. Buna kurala göre TABLO IV'te 1. kolondaki ikili elenirken, 2. kolondaki ikili aday olarak kalmıştır.

TABLO IV: ÖZNİTELİKLER

	KEMAL BURAK YILMAZ - BURAK YILMAZ	AHMET TUGRUL BAYRAKK - AHMET TUGRULBAYRAK
Tam isim benzerlik Levenshtein	0,636	0,944
Tam isim benzerlik cosine	0,807	0,984
İsim puanı	5,772	7,712
Cep telefonu numarası puanı	5	5
Şehir puanı	3	0
Doğum tarihi puanı	2	0
Cinsiyet puanı	1	1
Toplam puan	18,215	15,640

B. Makine Öğrenimiyle Mükerrer Kayıtların Bulunması

Aday çiftlerin toplam benzerlik puanları, özniteliklerden alınan puanlar kullanılarak Öklid uzaklığı ile ifade edilmiştir. Bir aday çiftin aynı kişi olduğunu belirleyen en az sayıda öznitelikten aldığı en yüksek puan, çiftin aynı kayıt olduğunun belirlenmesi için alt sınırı oluşturmaktadır. Bu kuralla, kayıtların mükerrer olma durumuna göre doğru ve yanlış kümesi için aday çiftler arasından kayıtlar seçilmiştir. Sistem doğası gereği yanlış olarak işaretlenmiş kayıtlar doğru olarak işaretlenmiş kayıtlardan daha fazla olduğundan, dengeli olmayan olan bu veriyi dengelemek için normal dağılıma göre doğru ve yanlış olarak belirlenen kayıtlardan birbirine yakın sayıda rasgele kayıtlar seçilmiştir. Bu işlem sayesinde iş birimi tarafından uzman bilgisiyle doğru ya da yanlış olarak işaretlenecek en uygun aday çiftler belirlenmiştir.

Yapılan manuel işaretleme sonrasında analiz için eğitim kümesi oluşturulmuştur. Eğitim kümesinde 29889 aday çiftin bulunduğu satır mükerrer, 30163 satır ise mükerrer değil olarak işaretlenmiştir. Bu kümede, isim özniteliklerinin farklı kombinasyonları için benzerlik puanları, isim dışındaki alanların ise puan alıp almamasına göre 1 ya da 0 değerleri yazılmıştır. 10545267 satırlık veri üzerinde sırasıyla naive Bayes, destekçi vektör makinesi, rassal ormanlar ve k en yakın komşu algoritmaları kullanılmıştır. Çapraz geçerleme sonrası uygulanan farklı algoritmalar için duyarlılık, kesinlik ve doğruluk değerleri aşağıdaki gibidir (TABLO V). Tabloda görüldüğü üzere, naive Bayes gibi basit bir algoritma fazla başarılı olmazken, karmaşık olan diğer algoritmalar daha başarılı sonuçlar vermiştir.

TABLO V: ALGORITMA PERFORMANSLARI

	Duyarlılık(%)	Kesinlik(%)	Doğruluk(%)
Naive Bayes	82,17	87,26	84,87
Destekçi Vektör Makinesi	99,65	99,49	99,62
Rassal Ormanlar	98,43	99,56	99,21
K En Yakın Komşu	98,71	98,49	98,54

Makine öğrenimi kullanılması sayesinde, puanlanmış örnek küme üzerinde iş birimi tarafından işaretlenmiş veriler kullanılarak, iki farklı kaydın mükerrer olma ihtimali hesaplanmıştır.

Algoritmalar Python'ın scikit-learn kütüphanesi kullanılarak uygulanmıştır. Algoritmalarda kullanılan parametrelerin optimizasyonda parametreler belirlenen değer aralıklarında denenerek algoritmaların performanslarına bakılmıştır. Rassal ormanlar algoritması için ağaç sayısı belirlenirken [100, 600] aralığına bakılmış, 300 olarak seçilmiştir. Destekçi vektör makinesi için *C*, *gamma* ve *kernel* parametreleri sırayla; 2154, 0.1 ve *rbf* olarak seçilmiştir. K en yakın komşu algoritmasında ise *n_neighbors* parametresi [0, 50] değer aralığından bakılarak 5 olarak belirlenmiştir.

IV. SONUÇ

Çalışmamız sonucunda, ilişkisel veri tabanında bulunan, tüm değerleri aynı olan mükerrer kayıtlarla birlikte, bazı değerleri farklı olsa da birbirine benzeyen ve aynı varlığı temsil eden kayıtların tespiti yapılmıştır. İlişkisel veri tabanında tablo yapısında bulunan veri, ön işlemeden geçirilip standart hale getirilmiştir. Daha sonra, isim, cep telefonu numarası, şehir bilgisi gibi öznitelikler kolon bazında ayrı ayrı birleştirilip, alt alta eklenerek aday mükerrer ikililer oluşturulmuştur. Bununla beraber, aday ikililerin kolon bazında yan yana getirilmesi işleminde, kartezyen çarpım kullanılmaması nedeniyle yakalanamayan aday ikililer bulunmaktadır. Bu durumda oluşacak satır sayısı ve hafıza gibi parametreler düşünüldüğünde, verilerin varlıkları temsil eden anahtarlarla ayrı ayrı birleştirilmesi daha verimlidir.

Alan uzmanlık bilgisi ve benzerlik fonksiyonları yardımıyla seçilip işaretlenen eğitim kümesi üzerinden makine öğrenmesi algoritmaları ile kurulan modeller, bahsedilen aday mükerrer ikililerden oluşan veri üzerinde uygulanmıştır. Bu işlemler sonucunda, veri tabanındaki aynı varlığı temsil eden kayıtlar yaklaşık %99 doğrulukla belirlenmiş, 9301467 satır veri içerisinde 28412 çift mükerrer kayıt başarılı bir şekilde tespit edilmiştir. Tespit edilen bu mükerrer kayıtlar veri tabanından temizlenerek, bu verilerin kullanıldığı rapor ve analizler daha tutarlı hale getirilmiştir.

KAYNAKLAR

[1] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James. Automatic linkage of vital records. Science, 130:954–959, 1959.

[2] M. A. Hernandez and S. J. Stolfo. The merge/purge problem for large databases. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD-95), pages 127–138, San Jose, CA, May 1995.

[3] A. K. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000), pages 169–178, Boston, MA, Aug. 2000.

[4] W. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002), Edmonton, Alberta, 2002.

[5] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Leaning", World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:1, No:12, 2007

[6] G. F. Morales, A. Gionis, "Streaming similarity self-join", Proceedings of the VLDB Endowment Journal, Vol:9, Issue:10, June 2016 Pages:792-803

[7] M. Bilenko, R. J. Mooney, "Adaptive Duplicate Detection Using Learnable String Similarity Measures", Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003), Washington DC, pp.39-48, August 2003