www.tubiad.org
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 3, No: 2, 2016 (329-334)

El-Cezerî Journal of Science and Engineering
Vol: 3, No: 2, 2016 (329-334)

ECJSE

Makale / Research Paper

Veri Madenciliği İle Yazılım Hata Tespiti

Züleyha AKGÜN[1]

[1] Akgün Yazılım, Etimesgut, Ankara – Türkiye

Geliş/Received: 11.09.2015 **Düzeltme/Revised:** 04.01.2016 **Kabul/Accepted:** 04.01.2016

Özet: Bilişim teknolojilerinde ilerleme ile hayatımız kolaylaşmış, özellikle kamu kuruluşları ile yaptığımız işleri bilgisayar ekranında tamamlama hatta imzalar evden elektronik ortamda atılmaya başlanmıştır. Bütün bu yazılımları yapan şirketlerin sayısı son yıllarda tüm dünyada olduğu gibi ülkemizde de hızla atmaya başlamıştır. Yazılım mühendisliği sadece kod yazmak olmayıp, projenin planlaması, tasarımı ve sınanmasını da içeren bir mühendislik disiplini olmuştur. Bu araştırma kapsamında yazılım esnasında yapılan hataların en aza indirgenmesi amacı ile veri madenciliği yöntemleri kullanılarak yazılım hataları ve bu hataların en çok yaşandığı durumlar tespit edilmiştir. Elde edilen sonuçlar göstermektedir ki yazılım mühendisinin mezun olduğu okul, tecrübesi, medeni durumu, yazılım dillerindeki tecrübesi ve ilgili sektör ve firma tecrübesi yazılım hata sayısına etki etmektedir.

Anahtar kelimeler: Yazılım mühendisliği, Veri madenciliği, Apriori algoritması, Hata.

Software Error Detection Using Data Mining

Abstract: Our life has been getting easier with information technologies. We use computers and internet in our daily life, even we sign documents using electronic signature in our home. In recent years number of software companies has been increased in all over the world. Software Engineering includes requirements, planning, design, coding and testing levels. All companies aim to decrease number of mistake in coding. In this research work we worked on data mining techniques to classfy coding mistakes and mistake reasons. Experimental results show that engineer graduated school, experience in coding, company and sector effects number of mistakes.

Keywords: Software engineering, Data mining, Apriori algorithm, Error.

1. GİRİŞ

Bilişim teknolojisindeki hızlı ilerleme ile birlikte yazılım mühendisliği önem arz eden bir meslek haline gelmiştir. Sağlık, eğitim, savunma sanayii, eğlence ve devlet sektörlerinde her geçen gün hayatımızı kolaylaştıran yazılımlar kullanılmaya başlamaktadır. Yazılım projesi geliştirmede en önemli safhalar çözümleme, tasarım, kodlama, sınama ve bakımdır. Bu aşamaların bir kısmında yapılan hatalar, eksiklikler projenin tamamına etki eder; projenin bakım maliyetini artırır. Yazılım firmaları bu geliştirme esnasında hata sayısının en az olmasını isterler. Yazılım kalitesini etkileyen unsurlar şu şekilde sıralanabilir [1,2]:

- Güvenirlik

Bu makaleye atıf yapmak için
Akgün, Z., "*Veri Madenciliği İle Yazılım Hata Tespiti*", El-Cezerî Fen ve Mühendislik Dergisi, 2016, 3(2); 329-334.

How to cite this article
Akgün, Z., "*Software Error Detection Using Data Mining*", El-Cezerî Journal of Science and Engineering, 2016, 3(2); 329-334.

- Doğruluk
- Verimlilik
- Kullanışlılık
- Hata bulma kolaylığı
- Esneklik
- Tekrar kullanılabilirlik

Bir yazılım projesinde en fazla karşılaşılabilecek hataları ise şu şekilde sıralayabiliriz:

- Arabirim hataları
- Veri tabanı erişiminde hatalar
- Çok sayıda niteleyici veri gerektirmesi
- Komut girişi halinde sistemin kesilmesi
- Yetersiz veya eksik test
- Hatalı veya eksik gereksinim belirleme
- Temel tasarım hataları

Organizasyonların ürettiği ürünün kalitesini ve projelerin beklenen zamanda ve kapsamda tamamlanmasını belirleyen en önemli unsurlardan biri de hatasız geliştirme yapmaktır. Hatalı ürün çıktıları aynı zamanda tekrar çalışmaları arttırmakta olup aynı iş ürünü üzerinde tekrar çalışmalara sebep olmaktadır. Hatalı üretilen ürün bileşenleri üzerinde yazılım uzmanları tekrar düzenleme ve ek geliştirme yapmakta olduğu gibi test uzmanları da aynı iş ürünü üzerinde tekrar test gerçekleştirmektedir. Bu ise hem zaman hem müşteri nezdinde güven ve hem de ticari işletmenin görünür görünmez birçok maliyetleri olarak ortaya çıkmaktır [1,3].

Yazılım geliştiricilere baktığımızda ise her birinin üretmiş olduğu ürün bileşenlerinin hata oranları değişkenlik göstermektedir. Bu durum yazılım geliştiricinin eğitim düzeyine, geliştirme yapmakta olduğu ürünün ait olduğu sektör kapsamındaki bilgisine, toplam çalışma tecrübesine, sektör tecrübe süresine, mezun olduğu üniversiteye, yaşına, geliştirme yapılan programlama diline, uzmanlık alanlarına gibi etmenlere bağlı olabilir ve değişkenlik gösterebilir. Bu bildiride sunulan çalışma kapsamında geliştirilen yazılım projelerinden toplanan veriler veri madenciliği yöntemleri kullanılarak hataların tespiti, yazılımcı-hata ilişkilerini içeren kurallar çıkartılması, sınıflandırmalar ve kümeleme yöntemleri uygulanmıştır. Kullanılan veriler genel olarak şu şekilde açıklanabilir [2,3,4]:

- Test sonuçlarına göre toplanan veriler; Test Edilen İş Ürün/Ürün Bileşeni Adı, Test Türü (Fonksiyonel, Entegrasyon, Birim Test), Test Edilen Toplam Senaryo Sayısı, Toplam Test Adım Sayısı, Hata Sayısı, Personel Bazlı Hata Yüzdesi.
- Personel kapsamında toplanan veriler; Mezun Olduğu Bölüm, Mezun Olduğu Üniversite, Tecrübe Yılı, Çalışmış Olduğu Projenin dahil olduğu alana ait Tecrübesi (Sağlık, Eğitim vb.), Medeni Durumu, Yaşı, Cinsiyeti, Projesinde Kullandığı Programlama Diline Ait Tecrübe Yılı.

Bu çalışma kapsamında veri madenciliğinde aşağıdaki yöntemler kullanılmıştır:

- Karar Ağacı
- Apriori Algoritması
- Naive Bayes Yöntemi
- Yapay Sinir Ağları

2. Verilerin Toplanması Ve Temizlenmesi

Çalışma kapsamında veri madenciliği yöntemleri kullanılarak hataların tespiti, yazılımcı-hata ilişkilerini içeren kurallar çıkartılması amacıyla geliştirilen yazılım projelerinden toplanan veriler aşağıdaki gibi olup; bu verilerin toplanması aşamasında kullanılan sistemler ve yöntemler takip eden paragraflarda açıklanmaktadır;

1. Yazılım Uzmanı Kişisel Bilgileri
    a. Adı, Soyadı
    b. Eğitim Seviyesi(Lisans,Doktora Vb.)
    c. Görev Yeri
    d. Birimi
    e. Unvanı
    f. Mezun Olduğu Üniversite
    g. Mezun Olduğu Bölüm
    h. Cinsiyeti
    i. Medeni Durumu
    j. Doğum Tarihi
    k. Üniversite Mezuniyet Yılı
    l. Toplam Çalışma Tecrübe Yılı
    m. Akgün'de Çalışma Tecrübe Yılı
    n. Proje Sektör Bilgisi Ve Sektörel Tecrübe Süresi (Ay)
2. Projenin Geliştirme/Üretim Hata Bilgisi (Personelin Toplam Hata Sayısı ve Hata Payı(%))
3. Projede Kullanılan Teknolojiler
    a. Veritabanı (Oracle, MySQL, SQLServer)
    b. Yazılım Geliştirme Ortamı (Eclipse, Intellıj Idea, Netbeans, Adobe Flash Builder, Microsoft Visual Studio)
    c. Kodlama (JAVA, PLSQL, C#, .NET, XML, HTML, DELPHI, JAVASCRIPT, ACTIONSCRIPT, C++, ORACLE FORMS)
    d. Çatılar (Spring, Hıbernate/JPA)
    e. Tasarım (EXT-JS, JQUERY,QT, FLEX, GXT)
    f. Web Servisleri (RESTFUL, AXIS, JAXWS, WCF)
    g. Raporlama (Oracle Reports, Java Raporlama, Itext, Jasper, OO, BIRD, Fast Report)
    h. Toolkit (DCM4CHEE, VTK, DCMTK)
    i. Sektörel Standartlar (HL7, DICOM, IHE, SAĞLIK NET, MKYS, MEDULA)
4. Personelin Projede Kullanılan Teknoloji Bazlı Tecrübe Süresi (ay)
5. DÖF Sayısı, Gözden Geçirme (Peer Review) Verileri (Uygunsuzluk Sayısı, GG Sayısı, GG Bazlı Uygunsuzluk Sayısı)

Geliştirme ekibinin kişisel bilgileri insan kaynakları birimince personelin özgeçmişi incelenerek ve gerekli durumlarda ilgililerle görüşerek elde edilmiştir. Projelerin kullanmakta oldukları veri tabanı, programlama dili, raporlama aracı vb. teknoloji ve ortamlar ile ilgili bilgiler ise; ilgili Proje/Birim Yöneticileriyle birlikte görüşerek ortaya çıkarılmıştır. Her bir proje üyesinin projede kullanılan teknolojiler kapsamındaki tecrübe süresi, personel ile gerçekleştirilen birebir görüşmelerde belirlenmiştir. Organizasyonda operasyona devrolmuş işler kapsamında müşterilerden ve firma içerisinden yazılım geliştirme ürünlerindeki değişiklik talepleri, yeni istekler, hatalar, öneriler vb.

dahilinde kurumsal ERP ve JIRA araçları kullanılmaktadır. Bu kapsamda, ürünlere ve ürün bileşenlerine gelen hatalar da bu sistemler üzerinden kayıt altına alınmaktadır. Ek olarak, yazılım üretim ürünlerimiz dahilinde gerçekleştirilen testler test senaryoları başarılı ve başarısız adımlar olarak kayıt altına alınmakta ve ürün/ ürün bileşenleri kapsamındaki hatalar izlenebilmektedir [5,6,7].

Tablo 1. Hata Payı Yüzdesi Hesaplama Bilgileri ve Tablo Formatı

Sorumlu Olduğu Ürün/Modül Listesi	**Hata Payı**	**Ürün/ Modül Hata Sayısı**	**Modül Tekrar Sayısı**	**Ağırlıklı Hata**	**Ağırlıklı Tekrar**	**Kişi Bazlı Toplam Hata Sayısı**	**Ürün Toplam Hata Sayısı**	**Hata Payı (%)**
X				(Modül Hata Sayısı* Hata Payı)	(Modül Tekrar Sayısı*Hata Payı)	(Ağırlıklı Hata + Ağırlıklı Tekrar)	(Ekibin Toplam Hata Sayısı)	(Kişinin Toplam Hata Sayısı/ Ekibin Toplam Hata Sayısı)*100
Y								
Z								
T								

Organizasyondaki yazılım üretim projelerinde ve operasyona devrolmuş işlerde ortaya çıkan hataların, aktif olarak kullanılmakta olan ve test hatalarının yer aldığı sistemlerden toplanması aşamasında aşağıdaki adımlar izlenmiştir;

1. Çağrı/Talep Türü, "Hata" olan işler filtrelenmiştir.
2. Her bir ürün/ürün bileşeni dâhilinde ortaya çıkan hata sayısı; kurumsal veri ambarı olarak kullanmakta olunan İş Zekası ürününden sayısal olarak elde edilmektedir.
3. Ürün/ürün bileşenlerinin geliştirilmesinden/bakım idamesinden sorumlu Yazılım Uzmanlarının bilgisi her bir ürün/ürün bileşeni/modül bazlı toplanmıştır.
4. Bazı ürün/ürün bileşenleri kapsamında birden fazla sorumlu Yazılım Uzmanı bulunmakta olup hangi ürün/ürün bileşeninden hangi Yazılım Uzmanlarının % kaç sorumlu oldukları bilgisi elde edilmiştir.
5. Eğer ürün/ürün bileşenleri dâhilinde birden fazla yazılım uzmanı sorumlu ise; ilgili ürün/ürün bileşeni kapsamında ortaya çıkan hata sayısı yazılım uzmanının sorumlu olduğu hata payı oranı ile çarpılmıştır. Modül Hata Sayısı ile Hata Payı Oranının çarpılması sonucunda Ağırlıklı Hata verisi elde edilmiştir.
6. Ek olarak, modüller kapsamında ortaya çıkan hataların tek seferde giderilmemesi sonucu tekrar işler ve tekrar hatalı durumlar ortaya çıkmakta ve bu tekrar işlere sebep olan hatalar ise İş Zekası üzerinden ürün/ürün bileşeni/modül bazlı tekrar sayısı olarak belirlenmektedir. (Tekrar sayısı hata türündeki bir işin test biriminden tekrar ilgili yazılım uzmanına geri dönüş sayısı olarak hesaplanmaktadır.)
7. Ürün/ürün bileşeni/modül bazlı ortaya çıkan tekrar sayısı ise; ilgili Yazılım Uzmanının ürün/ürün bileşeni/modül kapsamındaki hata payı oranı ile çarpılmaktadır.
8. Ürün/ürün bileşeni tekrar sayısı ile hata payı oranının çarpımı sonucunda Ağırlıklı tekrar verisi elde edilmiştir.
9. Ağırlıklı Hata ve Ağırlıklı tekrar verilerinin toplamı kişi bazlı Toplam Hata Sayısı verisini vermektedir.

10. (Kişinin Toplam Hata Sayısı/ Ürün Bazlı Toplam Hata Sayısı)*100 formülü kişinin Hata Payı Yüzdesini vermektedir.

Yukarıdaki maddelerde bahsedilen veriler aşağıdaki tabloya her yazılım geliştirici için kaydedilmiş ve hata payları hesaplanmıştır.

3. Yöntem

Veri madenciliği büyük veri setlerinden önemli bilgileri elde ettiğimiz bir süreçtir. Bankacılıkta risk analizlerinde; sigortacılıkta usulsüzlüklerin önlenmesinde; tıp alanında teşhiste; endüstride kalite kontrole kadar oldukça geniş bir uygulama alanı vardır. Veri madenciliği aşağıdaki süreçlerden oluşur [2,4,8,9]:

1. Verilerin temizlenmesi
2. Verilerin birleştirilmesi
3. Gerekli verilerin seçilmesi
4. Verilerin uygun hale dönüştürülmesi
5. Veri madenciliği
6. Elde edilen örüntülerin analizi

Veri madenciliği yöntemleri kural çıkarma, sınıflandırma ve kümeleme olmak üzere üç genel başlık altında toplanabilir. Kural çıkarma yöntemi olarak en çok kullanılan Apriori yöntemidir. Sınıflandırma yöntemleri Karar Ağacı (KA), Bayesian Sınıflandırma(BS) ve Yapay Sinir Ağları (YSA); Kümeleme yöntemleri ise K-ortalama, Genetik Algoritma (GA) önemli yöntemlerdir [3,10].

Karar Ağaçları yönteminde entropi hesaplayarak her bir özellik için kazanç hesabı yapıldıktan sonra elde edeceğimiz ağacın ilk düğümü belirlenir, ve diğer özellikler için veri tabanı güncellenerek sınıflar ortaya çıkana kadar devam edilip karar ağacı oluşturulur. Bayesien ve Naive Bayes Sınıflandırma yöntemlerinde ise ihtimal hesapları kullanılarak ortaya çıkabilecek her bir olasılık için bir oran belirlenir. Yeni bir veri ortaya çıktığında her bir sınıf için ihtimal yüzdeleri hesaplanır ve verinin en yüksek sınıfa ait olduğu sonucu çıkarılır. Bu çalışma sonucunda elde edilen sonuçlar şöyledir:

Tablo 2. Veri madenciliği yöntemi ile elde edilen başarı oranları

Veri Madenciliği Yöntemi	Doğruluk Oranı
Karar Ağacı	%82,4
Yapay Sinir Ağları	%89,6
Naive Bayes Sınıflandırma	%91,3
Apriori	%67,5

4. Sonuçlar

Bu çalışma sonucunda elde edilen sonuçlar Naive Bayes Sınıflandırma yöntemine göre %90 üzerinde başarı vermektedir. Geliştirilen çalışma kritik projelerde personel konumlandırılmasında, işe alımlar esnasında personelin değerlendirilmesinde ve seçiminde, kritik önem düzeyindeki projelerde tasarım ve kodlama yapacak yazılım uzmanlarının belirlenmesinde kullanılabilir.

Bu çalışma kapsamında elde edilen kazanç; kritik takvime sahip olan projelerde ve ürün bileşenlerinin geliştirilmesinde testlerden ortaya çıkan hataların düzeltilmesi esnasında ortaya çıkan hataların azaltılarak projelerin, işlerin vb. zamanında, beklenen kalite ve bütçe sınırları içerisinde

teslimatını sağlayarak işletmelerin verimlilik ve karlılığına katkı sağlamaktır. Kritik önem düzeyindeki veya takvimi net olan ve zamanında yetiştirilmediğinde kaldırılamayacak riskleri olan projelerde personel konumlandırılmasında bu verilerden yararlanılacaktır ve böylece riskleri de minimize etmek hedeflenmiştir.

Kaynaklar

[1] Kalıpsız O., Buharalı A. ve Biricik G., (2012). Sistem analizi ve tasarımı, Papatya Yayıncılık.
[2] Elbasi E. (2014).Veri Madenciliği Ders Notları, Çankaya Üniversitesi.
[3] J.W. Fawcett ve M.K. Gungor, (2005). Software development risk model. Applied to data from open-source Mozilla Project, International Conference on Software Engineering Research and Practice, Las Vegas, NV, USA.
[4] S.L. Pfleeger ve S.A. Bohner, (1990). A framework for software maintenance metrics, IEEE Transactions on Software Engineering, pp. 320-327.
[5] Kudyba, S. (2004). Managing Data Mining. Cy-berTech Publishing,146-163.
[6] Han, J., ve Kamber, M., (2001). Data Mining Concepts and Techniques, Morgan Kaufmann Publishers.
[7] Kaur., H., ve Wasan., S. (2006). Empirical Study on applications of Data Mining Techniques in Healthcare, Journal of Computer Science, 2(2).
[8] Chen., Y., ve Wu., S. (2003). Exploring Out-Pa-tient Behaviors in Claim Database: A Case Study Using Association Rules, AMIA Sym-posium Proceedings.
[9] Nagadevara., V., Application of Neural Prediction Models in Healthcare.
[10] Carino., C., Jia., Y., Lambert., B., West., P., Yu., C. (2005). Mining Officially Unrecognized Side Effects of Drugs by Combining Web Search and Machine Learning, CIKM'05,Bremen, Germany.