

**T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**METİN MADENCİLİĞİ TABANLI BİLDİRİM TAKİP
SİSTEMİ**

YÜKSEK LİSANS TEZİ

MEHMET ALİ KÜÇÜK

DENİZLİ, AĞUSTOS - 2019

**T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**



**METİN MADENCİLİĞİ TABANLI BİLDİRİM TAKİP
SİSTEMİ**

YÜKSEK LİSANS TEZİ

MEHMET ALİ KÜÇÜK

DENİZLİ, AĞUSTOS - 2019

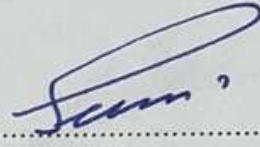
KABUL VE ONAY SAYFASI

MEHMET ALİ KÜÇÜK tarafından hazırlanan "Metin Madenciliği Tabanlı Bildirim Takip Sistemi" adlı tez çalışmasının savunma sınavı 20.08.2019 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği ile Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

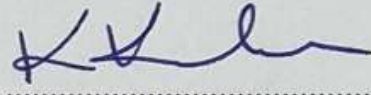
Jüri Üyeleri

İmza

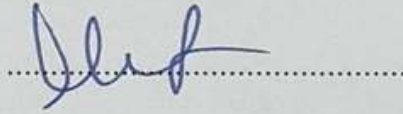
Danışman
Prof. Dr. Sezai TOKAT



Üye
Prof. Dr. Kadir KAVAKLIOĞLU



Üye
Dr. Öğr. Üyesi Mahmut SİNECEN



Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
28/08/2019 tarih ve 34/22 sayılı kararıyla onaylanmıştır.



Prof. Dr. Uğur YÜCEL

Fen Bilimleri Enstitüsü Müdürü ✓.

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde bilimsel etiğe ve akademik kurallara özenle riayet edildiğini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiğe uygun olarak kaynak gösterildiğini ve alıntı yapılan çalışmalara atfedildiğine beyan ederim.



MEHMET ALİ KÜÇÜK

ÖZET

METİN MADENCİLİĞİ TABANLI BİLDİRİM TAKİP SİSTEMİ
YÜKSEK LİSANS TEZİ
MEHMET ALİ KÜÇÜK
PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: PROF. DR. SEZAI TOKAT)

DENİZLİ, AĞUSTOS - 2019

Bildirim takip sistemleri kapsamlı yazılım projelerinin önemli bir bileşenidir. Modern yazılım endüstrisi, geliştiricilere çok sayıda sorun takip ve proje yönetim aracı sunmaktadır. Bu takip sistemlerinin temel amacı ekiplerin ve kullanıcıların çalışmalarını basitleştirmektir. Bildirim takip yazılımları kullanmak, ekip üyelerinin iletişim kurmasına ve koordinasyonuna yardımcı olmaktadır. Proje çok sayıda kullanıcıya hitap ediyorsa, bir bildirim takip yazılımı, kullanıcı istek ve beklentilerinin durumu veya bu konudaki değişiklikler hakkında hem geliştiricileri hem de kullanıcıları hızlı ve esnek bir yapıda bilgilendirebilir.

Farklı birçok modüller barındıran Pamukkale Üniversitesi Pusula Bilgi Sisteminde her modül için bir bildirim takip sistemine gerek duyulmaktadır. Geliştirilen F1 Bildirim Takip Sistemi tüm modüller için bildirimlerin tek çatı altında toplanmasını sağlayan Pamukkale Üniversitesi Pusula Bilgi Sisteminin önemli gelişmelerinden birisidir. F1 Bildirim Takip Sistemine girilen talepler metinsel web formları vasıtasıyla alınmaktadır. Metinsel veri madenlerinde bilgi arayan metin madenciliği, F1 Bildirim Takip Sistemi bildirimleri içindeki kelime veya kelime grupları ile bildirimlerin özellikleri arasındaki ilk bakışta görülemeyen bağlantıları çıkarmaktadır.

Bir bildirim takip sistemi, gelen taleplerin ilgili talep değerlendiriciye atanmasını gerektirir. Taleplerin doğru şekilde atanması bildirim takip sisteminin başarısı için önemli bir parametredir. İlgili proje seçim işleminin başlık satırından doğrudan alınması veya kullanıcının kendisinin seçmesi durumunda seçimler, talep metnindeki konudan farklı olduğu için sorunlar yaşanmaktadır. Bu tez çalışmasında, Pamukkale Üniversitesi Pusula Sistemi F1 Bildirim Takip Sistemine kullanıcılar ve geliştiriciler tarafından girilen taleplerin projeleri metin madenciliği yöntemlerinden Birliktelik Analizi ile otomatik olarak atanması amaçlanmaktadır. Böylece, Pamukkale Üniversitesi Pusula Sistemi F1 Bildirim Takip Sisteminde kullanılan manuel atama işlemi ortadan kaldırılmış, zaman ve iş yükü azaltılmış olacaktır.

ANAHTAR KELİMELEER:Bildirim Takip, Metin Madenciliği, Birliktelik Analizi

ABSTRACT

**TEXT MINING BASED ISSUE TRACKING SYSTEM
MSC THESIS
MEHMET ALİ KÜÇÜK
PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE
COMPUTER ENGINEERING**

(SUPERVISOR:PROF. DR. SEZAI TOKAT)

DENİZLİ, AUGUST 2019

Issue tracking systems are an important component of comprehensive software projects. The modern software industry offers numerous problem tracking and project management tools for developers. The main purpose of these tracking systems is to simplify the work of teams and users. Using an issue tracking software helps team members communicate and coordinate. If the project is addressed to a large number of users, an issue tracking software can quickly and flexibly inform both developers and users about the status or changes in user requests and expectations.

Pusula System that means “compass” in Turkish and contains many different modules, requires a ticket system for each module. Developed F1 Issue Tracking System is one of the important developments of Pamukkale University Pusula System, which provides the collection of issues under one roof for all modules. Requests entered into F1 Issue Tracking System are received through textual web forms. Text mining looking for information in textual data mines appears invisible links between words or phrases within F1 Issue Tracking System tickets and the characteristics of tickets.

An issue tracking system requires that incoming requests be assigned to the relevant request evaluator. The correct assignment of requests is an important parameter for the success of the issue tracking system. If the project selection process is taken directly from the title line or if the user selects it himself, problems arise because the selections are different from the subject in the request text. In this thesis, it is aimed to automatically assign the issues entered by Pamukkale University Pusula System F1 Issue Tracking System to the relevant request evaluator by using text mining methods, Conjoint Analysis. Thus, manual assignment process used in Pamukkale University Pusula System F1 Issue Tracking System will be eliminated and time and work load will be reduced.

KEYWORDS:Issue Tracker, Text Mining, Conjoint Analysis

İÇİNDEKİLER

Sayfa

| | |
|---|-----------|
| ÖZET..... | i |
| ABSTRACT | ii |
| İÇİNDEKİLER | iii |
| ŞEKİL LİSTESİ | iv |
| TABLO LİSTESİ | v |
| KISALTMALAR | vi |
| ÖNSÖZ..... | vii |
| 1. GİRİŞ..... | 1 |
| 1.1 Tezin Amacı | 4 |
| 1.2 Tezin Akışı | 4 |
| 2. BİLDİRİM TAKİP SİSTEMLERİ VE METİN MADENCİLİĞİ | 5 |
| 2.1 Bildirim Takip Sistemi | 5 |
| 2.1.1 Bildirim Takip Sistemleri Kullanım Amaçları | 9 |
| 2.1.2 Bildirim Takip Sistemlerinde Karşılaşılan Problemler..... | 10 |
| 2.1.3 Bildirim Takip Sistemleri ve Yığın İzleri | 11 |
| 2.2 Veri Madenciliği..... | 12 |
| 2.2.1 Veri Madenciliği Teknikleri | 13 |
| 2.2.2 Birliktelik Analizi | 14 |
| 2.3 Metin Madenciliği | 16 |
| 2.3.1 Metin Madenciliği Süreçleri | 18 |
| 2.3.1.1 Veri Toplama | 18 |
| 2.3.1.2 Temizleme ve Dönüştürme | 18 |
| 2.3.1.3 Model Belirleme ve Uygulama | 19 |
| 2.3.1.4 Tahminleme ve Raporlama | 19 |
| 2.3.2 Metin Madenciliği ve Veri Madenciliği | 19 |
| 2.4 Doğal Dil İşleme | 22 |
| 2.4.1 Türkçe Dili | 23 |
| 2.4.2 Zemberek | 26 |
| 2.5 Literatür Araştırması | 27 |
| 3. UYGULAMA | 29 |
| 3.1 F1 Bildirim Takip Sistemi | 33 |
| 3.1.1 Gereksiz kelimelerin atılması | 36 |
| 3.1.2 Metin Dönüştürme | 37 |
| 3.1.3 Kelime Köklerinin Çıkarılması..... | 39 |
| 3.2 Birliktelik Analizinin Uygulanması | 43 |
| 3.3 Birliktelik Analizinin Kişilere Uygulanması..... | 47 |
| 3.4 Tahmin İçin Uygun Değerlerin Belirlenmesi | 47 |
| 3.5 Bildirime Ait Projenin Tahmini | 51 |
| 4. SONUÇ ve KARŞILAŞTIRMA | 64 |
| 5. KAYNAKLAR..... | 67 |
| 6. ÖZGEÇMİŞ..... | 75 |

ŞEKİL LİSTESİ

Sayfa

| | |
|---|----|
| Şekil 2.1 : Bilgi Yönetim Grafiği..... | 16 |
| Şekil 3.1 : Bildirim Talep Ekranı | 34 |
| Şekil 3.2 : Örnek Bildirim Görüntüsü..... | 36 |
| Şekil 3.3 : Yazılım Akış Diyagramı | 41 |
| Şekil 3.4 : Uygun Güven Düzeyinin Belirlenmesi (Gereksiz Kelimeler ile).... | 48 |
| Şekil 3.5 : Uygun Güven Düzeyinin Belirlenmesi (Gereksiz Kelimeler Hariç)48 | |
| Şekil 3.6 : Eğitim Doğruluk Oranı | 50 |

TABLO LİSTESİ

Sayfa

| | |
|--|----|
| Tablo 2.1: Türkçe Sondan Ekleme Örneği | 24 |
| Tablo 2.2: Yapım Ekleri Örnekleri | 24 |
| Tablo 2.3: Türkçe Çekim Ekleri Örneği | 25 |
| Tablo 2.4: Türkçe Ses Türemesi ve Düşmesi..... | 25 |
| Tablo 3.1: Projelere Göre Bildirim Sayıları..... | 34 |
| Tablo 3.2: Gereksiz Kelimeler | 37 |
| Tablo 3.3: Dönüşüm Tablosu..... | 39 |
| Tablo 3.4: Yazılım Kod Örneği | 40 |
| Tablo 3.5: Kelime Tablosu..... | 40 |
| Tablo 3.6: Bildirim Örneği..... | 42 |
| Tablo 3.7: Açıklama Kelime Tablosu | 42 |
| Tablo 3.8: Başlık Kelime Tablosu | 42 |
| Tablo 3.9: Kelime Öneri Tablosu..... | 43 |
| Tablo 3.10: Örnek “Dil” Kelimesi İçin Açıklama Alanında Güven Değer Tablosu | 44 |
| Tablo 3.11: Örnek “Dil” Kelimesi İçin Başlık Alanında Güven Değer Tablosu | 45 |
| Tablo 3.12: Örnek “Dil” Kelimesi için Kelime Ağırlığı..... | 46 |
| Tablo 3.13: Örnek Bir Kullanıcıya Ait Güven Değeri..... | 47 |
| Tablo 3.14: ‘Ve’ Kelimesi Güven Değeri..... | 49 |
| Tablo 3.15: Eğitim Oranları | 50 |
| Tablo 3.16: Bildirimin Projelere Göre Ağırlıkları | 52 |
| Tablo 3.17: PBS Bildirim Örneği..... | 53 |
| Tablo 3.18: PBS Bildirimi Başlık Kelimeleri | 53 |
| Tablo 3.19: PBS Bildirimi Açıklama Kelimeleri..... | 52 |
| Tablo 3.20: PBS Bildirim Proje Tahmini..... | 54 |
| Tablo 3.21: ENBS Bildirim Örneği | 54 |
| Tablo 3.22: ENBS Bildirimi Başlık Kelimeleri | 55 |
| Tablo 3.23: ENBS Bildirimi Açıklama Kelimeleri..... | 55 |
| Tablo 3.24: ENBS Bildirimi Proje Tahmini | 56 |
| Tablo 3.25: EYS Bildirimi Örneği | 57 |
| Tablo 3.26: EYS Bildirimi Başlık Kelimeleri..... | 57 |
| Tablo 3.27: EYS Bildirimi Açıklama Kelimeleri | 58 |
| Tablo 3.28: EYS Bildirimi Proje Tahmini | 59 |
| Tablo 3.29: EDO Bildirimi Örneği | 60 |
| Tablo 3.30: EDO Bildirimi Başlık Kelimeleri | 60 |
| Tablo 3.31: EDO Bildirimi Açıklama Kelimeleri..... | 61 |
| Tablo 3.32: EDO Bildirimi Proje Tahmini | 62 |
| Tablo 3.33: Bildirim Tahmin Oranı | 63 |

KISALTMALAR

| | | |
|---------------|---|-------------------------------------|
| BTS | : | Bildirim Takip Sistemi |
| F1 | : | F1 Bildirim Takip Sistemi |
| OBS | : | Öğrenci Bilgi Sistemi |
| OBİS | : | Öğrenci İşleri Bilgi Sistemi |
| ENBS | : | Enstitü Bilgi Sistemi |
| HBS | : | Hazırlık Bilgi Sistemi |
| BTS | : | Bildirim Takip Sistemi |
| PBS | : | Personel Bilgi Sistemi |
| EBS | : | Eğitim Öğretim Bilgi Sistemi |
| BİYS | : | Bukalemun İçerik Yönetim Sistemi |
| TFBS | : | Tıp Fakültesi Bilgi Sistemi |
| EYS | : | Eğitim Yönetim Sistemi |
| ERBS | : | Erasmus Bilgi Sistemi |
| YOBS | : | Yurt Dışı Öğrenci Bilgi Sistemi |
| EDO | : | Ek ders Otomasyonu |
| FBS | : | Formasyon Bilgi Sistemi |
| SBS | : | Sağlık Spor ve Kültür Bilgi Sistemi |
| ABS | : | Altyapı Bilgi Sistemi |
| MBS | : | Mezun Bilgi Sistemi |
| SYBS | : | Stratejik Yönetim Bilgi Sistemi |
| SYS | : | Süreç Yönetimi Sistemi |
| EDS | : | Eğitim Destek Sistemi |
| PRBS | : | Proje Bilgi Sistemi |
| PAÜ | : | Pamukkale Üniversitesi |
| Pusula | : | Pusula Bilgi Sistemi |

ÖNSÖZ

Rekabetçi modern işletmeler dünyasında önemli bir yeri olan yazılımlar, yazılım ekibi ve kullanıcıların geliştirme süreci boyunca verdikleri geri beslemeler ile sürekli gelişen ve büyüyen bir yapıda olmaktadır. Pamukkale Üniversitesinde kullanılmakta olan Pusula Bilgi Sistemi, eklenen yeni modüller ve projeler ile her geçen gün büyümektedir. Bu büyüme ile orantılı olarak yazılımların geliştirme süreçleri karmaşıklaşmakta, yazılım gereksinimleri çeşitlenerek ve değişerek artmaktadır.

Pamukkale Üniversitesi Pusula Bilgi Sisteminde yazılımların geliştirme süreci ilk zamanlarda birebir iletişim yoluyla ve günlük veya haftalık toplantılarla ilerlemekteydi. Yazılımın yeni özellikleri kullanıcılara anlatılıp gerekli düzeltmeler için notlar alınmaktaydı. Bu düzeltmeleri veya yeni işleri faaliyete almak sonraki toplantılara kalmaktadır. Modül ve proje sayısı arttıkça birebir iletişim ve toplantı sayıları yetersiz kalmaya başlamıştı. Ayrıca modüllerin ve süreçlerin kayıt altına alınması gerekli olmuştur. Son kullanıcılar ise bu durumdan çok memnun değillerdi. İşleyişi hızlandırmak, kişiler arası iletişimi sağlamak ve yazılım süreçlerini takip etmek için bir bildirim takip sistemi gerekmektedir. Pusula Bilgi Sistemini tasarlayan bilgi işlem yazılım ekipleri, yaptıkları kurum-içi toplantıda artık sistemin bir bildirim takip sistemine gereksinimi olduğunu tartışarak kendi bildirim takip yazılımını yazmaya karar verildi ve bu gereksinimler sonucunda aktif olarak kullanılmakta olan F1 Bildirim Takip Sistemi ortaya çıktı.

Metin madenciliği metinler içerisinde bilinmeyen karmaşık örüntüler çıkarmaya çalışan bir tekniktir. Günümüzde işletmeler, pazar analizi yapmak ve gelişmeleri takip etmek için veri ve metin madenciliğine büyük önem vermekte ve kullanmaktadır.

Pusula Bilgi Sistemindeki F1 Bildirim Takip Sistemi ve metin madenciliği konularının öneminden yola çıkarak, çalışmakta olduğum Pamukkale Üniversitesi Bilgi İşlem Daire Başkanlığında geliştirilen Pusula Bilgi Sisteminin önemli bir parçası olan F1 Bildirim Takip Sistemini iyileştirmek için metin madenciliğinden yararlanmak amacı ile bir tez çalışması yapmaya karar verdik. Böylece, F1 Bildirim Takip Sistemi üzerinde çalışmalara başladım ve tezimi hazırladım. İlk anlamlı kelimeler çıkarıldığında tahminleme yüzdesi çok düşük çıkmıştı.

Analizler sonucunda bildirimlerden çıkarılan kelime sayısının beklediğimizden az olduğunu gördük. Çünkü anlamsız kelimeleri ele almamıştık; kullanıcılar kelimeleri tam olarak Türkçedeki doğru karşılığı ile yazmıyor, kısaltmaları farklı kullanabiliyordu. Ayrıca dönem, tarih, öğrenci numarası gibi kalıpların farklı biçimlerde girildiği ettik. En büyük problemlerden biri de cümlelerin ve kelimelerin bir kısmının imla kurallarına uygun yazılmamış olmasıydı. Bütün bu konuları bir arada değerlendirerek Pamukkale Üniversitesi Pusula Bilgi Sistemi F1 Bildirim Takip Sistemi için girilen bildirimlerin ilgili oldukları projeleri otomatik olarak belirleyen bir sistemi veri madenciliği yöntemlerini kullanarak gerçekleştirme düşüncesi ile yola çıktım.

Yüksek lisans tez çalışmamda bana her konuda yol gösteren, aradığımda her daim vaktini bana ayıran değerli hocam Prof. Dr. Sezai TOKAT' a verdiği bilgiler ve desteğinden dolayı teşekkürlerimi sunarım.

Analiz kısmında bana desteklerini esirgemeyen iş arkadaşım Günay KILIÇ'a, beni sürekli teşvik eden, moral veren arkadaşım İbrahim BUDAK'a ve Mehmet Ulaş KOYUNCUOĞLU' na teşekkür ederim.

1. GİRİŞ

Pamukkale Üniversitesi F1 Bildirim Takip Sistemi yazılımı kullanıcılar tarafından girilen açıklama kısmında, kişinin karşılaştığı durum ile ilgili açılmış olan taleplerin ait olduğu projenin belirlenmesi, sorumlu yazılımcıya bildirim atanması ve sorumlu kişinin problemi çözdükten veya gereksinimleri karşıladıktan sonra işlemi sonlandırması süreçlerini karşılayan bir yazılımdır. Bu süreçler zaten bütün bildirim ve iş takip sistemlerinin yapması beklenen temel işlemlerdir. F1 Bildirim Takip Sisteminde başlık ve açıklama alanları metin olarak son kullanıcılar, test yapan kişiler veya analistler tarafından girilmektedir. Pamukkale Üniversitesinde yazılım ekipleri proje tabanlı ayrılmaktadır. Bu yüzden çözümün hızlandırılması için bildirim projenin belirlenmesi en önemli noktalardan birisidir. Projenin yanlış belirlenmesi yanlış kişiye veya kişilere atanması demektir. Bu durumda çözüme tekrardan proje seçimi ile başlamak gerekmektedir. Bildirimler kategori bilgisi de barındırmasına rağmen, farklı projeler benzer kategorileri içerdiğinden çözüme hızlı erişilebilmesi için önce projenin belirlenmesi gerekmektedir.

Günümüzde yazılım sektörü hızla gelişmekte, dünya çapında çok-kullanıcılı, çok-uluslu, çok-dilli uygulamaların kullanımı her geçen gün artmaktadır. İlgili paydaşların dünya üzerinde uzak noktalara dağılmasıyla, özellikle yazılım gereksinimlerinin yönetimi karmaşık hale gelmekte ve kullanımı zorlaşmaktadır (Prause ve diğ. 2008). Bir işletme yüksek teknoloji üretmiyor veya kullanmıyor olsa bile, yazılım proje yöneticileri ve geliştirme ekipleri işletmenin başarısında kritik bir rol oynamaktadır. Bilgi teknolojileri ve yazılım geliştirme süreçlerindeki başarısızlıkların farklı sebepleri olmasına rağmen üzerinde az durulan sebeplerden biri de işletmenin kaynaklarını harcayan fakat amacına asla ulaşamayan projelerdir (Keil 1995). Bu süreçleri yöneten uzmanların yazılım geliştirme, iyileştirme ve bakım süreçleri hakkındaki bilgileri örgütsel başarı veya başarısızlığı belirlemektedir (Futrell ve diğ. 2001).

Bireysel, takıma bağlı, örgütsel birçok farklı kaynağın yönetimini gerektiren karmaşık yapıdaki yazılım projelerinin yaşam döngüleri boyunca pek çok konuyla ve problemle uğraşılmak zorundadır. Bu durumda uzmanlar proje yönetim araçlarından

yararlanırlar (Mishra and Mishra 2013). Proje yönetim araçlarında bulunması gereken önemli özelliklerden biri de bildirim takip yazılımlarıdır.

Hata izleme (bug tracer) olarak da bilinen bildirim takip yazılımları, basit bir posta-listesi uygulamasından farklı olarak yazılımın geliştirilmesini kolaylaştıracak, çözecek ve hataların her ayrıntısına ilişkin düzenlemelerin yapılmasına izin veren yazılım uygulamalarıdır (Dall'Olio ve diğ. 2010). Bu yazılımlar, bir yazılım paketindeki hataların daha hızlı çözüme ulaştırılmasını ve bulunan çözümler üzerinde veya çözüm bulma yöntemleri üzerinde bir tartışma ortamı oluşturulmasını sağlayarak üzerinde çalışılan yazılımın geliştirilmesini kolaylaştırır, yazılım paketinin geliştirilmesi sırasında durum belirleme ile aktif veya pasif olan yazılım proje modüllerinin belirlenebilmesine yardımcı olmaktadır (Dall'Olio ve diğ. 2010). Bir yazılım paketinde, bir proje modülüne ait çok sayıda bildirim talebi bulunuyorsa, bu durum, ilgili proje modülünün etkin bir topluluk tarafından kullanıldığı anlamına gelir. Yazılımcılar bildirim taleplerine ve sorularına hızlı bir şekilde karşılık verebiliyorsa, yazılım geliştirme sürecinin aktif olarak devam ettiği ve kodun kullanıcı talep ve gereksinimleri de göz önüne alınarak üretildiği anlaşılmaktadır (Dall'Olio ve diğ. 2010). Bildirim takip yazılımları, bu açıdan, kullanıcıların davranışlarını belirlemede de yol gösterici olmaktadır.

Pamukkale Üniversitesi Pusula Bilgi Sistemi (Pusula) 2008 yılında Pamukkale Üniversitesi (PAÜ) Bilgi İşlem Daire Başkanlığı bünyesindeki yazılım ekipleri tarafından geliştirilmeye başlanmış ve halen geliştirilmekte olan, üniversitenin yazılım otomasyonu gerektiren tüm uygulamalarını barındıran kapsamlı bir yazılım projesidir. Pusula, bünyesinde dâhil olduğu yazılımların iş takibinin yapılabileceği F1 Bildirim Takip Sisteminin de dâhil olduğu 2019 Temmuz ayı itibariyle 23 adet alt yazılım projesinden oluşmaktadır. Her bir yazılım projesi için kullanıcılar tarafından F1 Bildirim Takip Sistemi üzerinden iyileştirme, hata ve öneri talepleri açılmaktadır. Bu taleplerde başlık, açıklama ve tip alanları seçilerek giriş yapılmaktadır. İlgili bildirim için ilgili yazılım projesine ve doğru yazılımcıya ulaştırılabilmesi için kullanıcılar tarafından girilen başlık ve açıklamalardaki metinlerin doğru değerlendirilmesi gerekmektedir. İş takip sistemlerinde metin içeriği yazılım analistleri yardımı ile yapılabildiği gibi metin madenciliği sayesinde bilgisayarlar yardımı ile de yapılabilir.

Metin madenciliđi, daha önce bilinmeyen bir bilginin bilgisayar ve yazılım teknolojileri yardımı ile metinlerden otomatik olarak elde edilmesidir. Temel anahtar, geleneksel deneyerek öğrenme yöntemleriyle daha uzun sürecek metinden bilgiyi anlama ve ayırıştırma süreci ile hipotezler oluşturma süreci için, çıkarımların bir araya getirilmesidir. Metin madenciliğinde amaç, daha önce bilinmeyen bilgileri keşfetmektir. Metin madenciliđi, büyük veri tabanlarından ilginç örüntüler bulmaya çalışan bir tekniktir (Hearst 2003).

Metin madenciliđi ile ilgili gelişmeler metinleri kategorize etmek ile başlamış ve doğal dil işleme teknikleri ile devam etmiştir. Metin madenciliğinde amaç çok karmaşık olan insan beyninin metinleri anlama, ilişkilendirme, sorgulama becerilerini yaklaşık olarak taklit edebilecek makineler oluşturmak veya sistemler geliştirmektir. (Miner, ve diğ. 2012).

Metin madenciliğinin uygulanması eđer metinler belirli karakterlerle veya standart kalıplarla düzenli biçimde ayrılmış ise çok kolaydır. Fakat metinler daha karmaşık cümle veya cümle gruplarından oluşuyorsa metin madenciliğinin uygulanması zorlaşmaktadır. Bu amaçla birçok dil için o dile özel olarak geliştirilmiş olan doğal dil işleme teknikleri kullanılmaktadır. Bu teknikler ile paragraflar cümlelere ayrılır. Daha sonra cümleler kelime ve kısaltmalara bölünür. Belirlenen kelimelerden de kelime kökleri bulunmaktadır. Her dilde paragraf ve cümleler belirli standartlarla başlayıp bitmesine rağmen, cümle ve kelime yapıları farklıdır. Bu yüzden her dil için farklı teknikler geliştirilmiştir.

Bu çalışmada, metin madenciliđi tekniklerinden birliktelik analizi yöntemi ile Türkçe dil desteđi bulunan, Java tabanlı bir doğal dil işleme kütüphanesi olan Zemberek Kütüphanesi kullanılmıştır. Türkçe dil desteđi sağlayan Zemberek Kütüphanesi yardımı ile bildirimlerin başlık ve açıklama alanındaki metinlerin içerdiđi kelimeler köklerine ayrılmıştır. Ortaya çıkan kelimeler ve kısaltmaların projeler ile ilişkileri veri madenciliđi tekniklerinden olan birliktelik analizi ile belirlenmiştir. Birliktelik kurallarına göre belirlenen bu ilişkiler ile bir karar destek sistemi oluşturulmuştur. Bu karar destek sistemi verilen bildirim başlık ve açıklama alanındaki metnin kelimelerine ve bildirim açan kişiye göre bildirim ilgili olduđu proje içeriđini belirleyebilmektedir.

1.1 Tezin Amacı

Bildirim takip sistemlerinin amacı, yapılması gerekli işlemin bir an önce işi gerçekleştirecek kişiye iletilmesi ve işin en hızlı şekilde çözülmesine yardımcı olmaktır. Büyük ölçekli yazılım projelerinde ilgili işi ilgili projeye yönlendirmesi zaman almaktadır. Tez çalışmasında, Pamukkale Üniversitesi Bilgi İşlem Daire Başkanlığında geliştirilen Pusula Bilgi Sistemi'ne ait bildirim takip yazılımı olan F1 Bildirim Takip Sistemi ele alınarak, bildirim ve iş takip sisteminin başlatıldığı günden bu yana girilen bildirim talepleri ön-işleme süreçleri ve gerekli metin madenciliği yöntemleri ile analiz edilmiş uzman analistler tarafından manuel işlem yapılmasının önüne geçecek yapay zekaya dayalı bir sistem geliştirilmesi amaçlanmıştır. Böylece Pamukkale Üniversitesi Bilgi İşlem Daire Başkanlığında zaman ve iş-gücü maliyetlerinin azaltılması ve bildirim proje yönlendirmelerinin otomatik olarak yapılması hedeflenmektedir.

1.2 Tezin Akışı

Tezin akışı: Tezin ilk bölümünde, tez konusuyla ilgili ön bilgiler verilmiş, ikinci bölümde bildirim takip sistemleri ve metin madenciliği konuları tanıtılmış, tezde kullanılan birliktelik analizi ve doğal dil işleme üzerinde durulmuştur. Üçüncü bölümde, Pamukkale Üniversitesi Pusula Bilgi Sisteminde yer alan F1 Bildirim Takip Sisteminden elde edilen verilerin doğal dil işleme araçları yardımıyla düzenlenmesiyle oluşturulan veri kümelerine birliktelik analizi uygulanması üzerinde durulmuş ve proje tahmin başarımları analiz edilmiştir. Dördüncü bölümde ise yapılan çalışma değerlendirilmiş ve gelecekte yapılabilecek çalışmalar konusunda bilgi verilmiştir.

2. BİLDİRİM TAKİP SİSTEMLERİ VE METİN MADENCİLİĞİ

2.1 Bildirim Takip Sistemi

Bilgisayarın icadından bu yana dört işlemle ve delikli kartlarla başlayan bilgi işleme ve kaydetme süreci günümüzde farklı cihazlarla milyonlarca farklı işlem yapılarak değişik ortamlarda ve biçimlerde devam etmektedir. Teknolojilerde pek çok gelişim ve değişim olmuştur. Kişiler için bu teknolojik gelişmelerle birlikte yazılım, çalışma, belgeler, ürünler vs. üzerinde ortak çalışabilmek ve iletişim içerisinde kalabilmek çok önemlidir. Teknoloji kullanılan pek çok alanda iletişimi sürdürmek, problemleri gidermek, yeni değerler oluşturabilmek ve bunları kayıt altına almak için Bildirim Takip Sistemleri (BTS) geliştirilmiştir.

Tüm kurumlarda çalışanlar bazı görevler üzerinde çalışır, ekip içinde işbirliği yapar ve belirli süreçleri uygulayarak iş takibini yapmaktadırlar. BTS Kullanıcıları, görevlerini takip edebilir, diğer ekip üyeleriyle paylaşabilir, amaçlanan ayrıntıları ekleyebilir, ekleri belirleyebilir, belirlenen özellikleri ayarlayabilir, yapılan işlemle ilgili sorunun durumunu belirleyebilir. Pek çok kurum sistemi sadece iş takibi için kullanmak istememektedir, aynı zamanda bir ürünün test aşamasında hata raporlaması için, proje yönetimi ve isteğe bağlı kurumsal süreçleri içinde barındırmasını istemektedir (Wu ve diğ. 2011).

BTS yazılım alanında önemli bir yere sahiptir. Küçük, orta veya büyük ölçekli pek çok yazılım projesinde kullanılmaktadır. BTS, özellikle çok kullanıcıli sistemlerde kullanıcı ile yazılımcı arasındaki iletişimi ve problemi çözmeyi sağlamaktadır. BTS'lerde bildirimler sadece kişiler tarafından oluşturulmakla kalmamakta, yazılımın kendisi tarafından kritik hatalarda da otomatik olarak da oluşturulabilmektedir. Otomatik açılan bu bildirimlere özelleştirilebilir bilgiler de eklenebilmektedir. Bu bildirimlere aşağıda bahsedilecek olan yığın izi raporları yerleştirilebilir. BTS sayesinde artık evden veya uzak mesafelerden problemleri takip etmek veya yazılımcı grubuyla problemi çözmek için bir tartışma platformu

oluşturmak mümkün olmaktadır. BTS'ler projenin geçmişi ile ilgili bilgi de sunabileceği gibi projenin geleceğini tahmin etmek için de veri sunabilir. BTS'ler gelen isteklerin ve yapılan işlerin sayılabilirliğini sağladığı gibi, yazılımla ilgili önemli hata ve yeniliklerin ne zaman, nasıl ve kimin tarafından yapıldığı ile ilgili bilgi de vermektedir.

BTS'lerin faydalarından bahsetmek gerekirse öncelikli olarak kullanıcı odaklı, kaliteli bir yazılım geliştirmesine yardımcı olmaktadır. Son kullanıcı veya müşteri memnuniyetini arttırmakla beraber, son kullanıcılar ve müşteri ile yazılım ekibi arasındaki iletişimi de geliştirmektedir. Masraflar azaltılarak yazılım ekibinin üretkenliği arttırılmaktadır (Janak 2009).

BTS temelde, yazılım ve/veya donanım geliştirilen projelerin tasarımı esnasında, çıkan teknik hataların raporlanması ve takibini içeren bir yazılım uygulamasıdır. BTS'ler 1970'den beri kullanılmakta olup geliştirilen sistemlerin kullanıcıları veya testçileri tarafından bir hata ile karşılaşılması durumunda, form doldurulması şeklinde çalışır. Doldurulan formlar genellikle, hatayı açıklayan bir özet, hatanın oluşma şekli, tekrardan oluşturulabilmesi için adımlar ve diğer tanımlama bilgilerini içerir (Lotufo ve diğ. 2012).

BTS'ler yeni özellik isteklerini, bir defalık görevleri, istenmeyen yamaları, farklı başlangıç ve bitiş durumlarına sahip olan her şeyi izlemek için de sıkça kullanılır. Bu nedenle, BTS'ler aynı zamanda sorun izleyici, kusur izleyici, yapay izleyici, istek izleyici veya sorunlu bilet sistemi vb. olarak da adlandırılabilir (Fogel 2005).

BTS, özellikle açık kaynak kodlu projeler olmak üzere herhangi bir yazılım projesinin kritik bir parçasıdır. Proje geliştirme sürecinin merkezi bir parçası ve geliştiricilerin her biri için günlük bir bileşen konumundadır. Bir yazılım projesinin ana bilgisayarlarında veya üçüncü taraflarca barındırdığı bir sisteme kadar çeşitli BTS çözümleri bulunmaktadır. BTS'ler, özellikler açısından büyük ölçüde farklılık gösterir. Bazıları az sayıda geliştiriciye sahip küçük basit projeler için tasarlanmıştır. Bazıları oldukça karmaşık, işlevselliği sınırsız şekilde özelleştirilebilen yapıdadırlar. Bir BTS'yi seçmeden önce gereksinimler değerlendirmelidir. Yalnızca yazılımdaki bir hata listesini tutmak için basit bir BTS'ye gereksinim olabileceği gibi, birden çok proje, ekip, otomasyon ve diğer özelliklerle karmaşık bir BTS çözümüne de

gereksinim duyulabilir. Gereksinimler ne olursa olsun, bir BTS seçiminde her zaman kilit bir gereksinim olarak kullanılabilirliği korunması gerekmektedir (Bacon 2012).

BTS'ler, kuruluşların sorun bildirme, atama, izleme, çözünürlük ve arşivlemeyi yönetmelerine yardımcı olur. Bununla birlikte, bildirim takibi temelde sosyal bir süreçtir ve bu nedenle sistemin tasarımını ve kullanımını bu açıdan anlamak önemlidir. Bir BTS sadece hataları, özellikleri ve sorunları takip etmek için bir veri tabanı değil, aynı zamanda yazılım ekibinin içindeki ve dışındaki birçok paydaş için iletişim ve koordinasyon noktası olmaktadır. Müşteriler veya son kullanıcılar, proje yöneticileri, kalite güvence personeli ve programcıların hepsi, BTS içinde var olan paylaşılan bilgilere ve kalıcı iletişime katkıda bulunmaktadır (Bertram ve diğ. 2010).

Baysal ve diğ. (2013) BTS'lerin en çok yazılım projelerinde kullanıldığından bahsetmektedir. BTS'ler geliştiricilerin, yöneticilerin ve son kullanıcıların hata ve özellik istekleri göndermelerini sağlamakta ve ayrıca hata düzeltme ve kod inceleme gibi diğer görevlerde de kullanılmaktadır. Bu sistemler proje yönetimi görevlerinde de kullanıldığından, geliştirme-bakım faaliyetleri ve proje destekleyici araçlarla entegre olabilmektedir (Baysal ve diğ. 2013).

BTS'lerin çalışma prensipleri temel olarak yapılacaklar ('Todo'), yapılanlar ('Doing'), test aşaması ('Test') ve bitenler ('Done') olarak dörde ayrılır. Elbette BTS'lerin çoğu özelleştirilebilir şekilde tasarlanmaktadır. Bunlara ek olarak daha sonra yapılacaklar ('BackLog'), belki iterasyon planı yaparak ('Iteration') bölümleri eklenebilmektedir veya kurumsal organizasyonlarda kullanılan başka bir durum eklenip çıkarılabilmektedir. 'Todo' kısmında son kullanıcılardan, proje yöneticilerinden veya test uzmanlarından gelen bildirimler yer almaktadır. İlk değerlendirme burada yapılır. Kurumsal çözümlere göre iptal veya ret işlemleri, süreci kısaltma adına burada yapılabilmektedir. İlk programcı, analist, son kullanıcı ve proje yöneticisi ile iletişim burada başlamaktadır. 'Todo'da projesi veya kategorisi, nasıl çözümleneceği belirlenen bildirim, çözümü yapacak kişiye atanır. Bu işlemlerle beraber bildirim 'Doing' aşamasına geçmiş olur. Bu aşamada bildirim soruna yönelik çözüm veya istenilen özelliğe dönük geliştirmeler uygulanmaktadır. Çözüm veya geliştirmeye yönelik standart testler yapıldıktan sonra ek testlerin yapılabilmesi için 'Test' durumuna geçirilir. Burada test uzmanları

tarafından ek testler yapıldıktan ve sorunun çözüldüğü veya özelliğın eklendiğı bilgisi bildirimini açana haber verildikten sonra ‘Done’ bölümüne aktarılarak işlem tamamlanır.

Diao ve diğ. (2009) tarafından önerilen BTS’de son kullanıcılar, İnternet aracılığıyla bilet oluşturma ve erişimini koordine eden biletleme sistemi yoluyla bildirim açarlar. Bildirim kayıtları veri tabanında saklanır ve kullanıcı adı, platform, hata açıklamaları ve önem kodu gibi olay ayrıntılarını içerir. Sistem yöneticileri bildirimleri sistemlerinden alır, olaylara göre sorunları tespit eder, çözer ve belgelenmiş çözümlerle bildirimleri kapatır (Diao ve diğ. 2009).

Baysal ve diğ. (2013) tarafından yapılan çalışmada, bildirim takip sistemlerinin kişiselleştirilmesi üzerine durumsal farkındalıktan bahsedilmiştir. Bireysel yazılımcılar kendilerine özgü bildirim takip sistemlerine gerek duymuşlardır. Ayrıca, yazılımcının sık sık gerçekleştirdikleri belirli görevlere göre uyarlanmış görüşler sunarak, yazılımcının çalışma bağlamındaki durumsal farkındalığını artıran “Extending Bugzilla” adlı bir prototip önerilmiştir. Mozilla geliştiricilerinden gelen girdilerle bu prototip aktif olarak kullanılmıştır (Baysal ve diğ. 2013).

Sureka ve diğ. (2011) tarafından yapılan çalışmada, bir yazılım projesinde risk, tehdit ve hata analizi için takip sistemlerine sistematik bir yaklaşım sunulmuştur. BTS’leri, versiyonlama sistemleri ve e-posta listeleri ve yazılım havuzları gibi pratik olarak çok sayıda bilgi içermektedir. Bu yönüyle, bildirim takip sistemlerinden türetilen bilgilerle sosyal ağ analiz teknikleri kullanılmıştır. Mozilla Firefox projesinin hata rapor verileri üzerinde deneysel analizler yapılmıştır (Sureka ve diğ. 2011).

Pandey ve diğ. (2017) yaptıkları çalışmada test yapanların ve müşterilerin kullanmış oldukları BTS’de bildirimleri sınıflandırmayı amaçlamışlardır. BTS yazılımlarından biri olan JIRA programı kullanılan bir işletmenin açmış olduğu bildirimler üzerinde analizler yapılmıştır. BTS’de Hata, İyileştirme, Yeni İstek, Belge, Kod Gözden geçirme ve Diğer olmak üzere 6 adet kategorisi olan bildirimler vardır. Bu kategorileri Hata (Bug) ve Hata olmayan (Non-Bug) olarak iki sınıfa ayırarak sınıflandırma algoritmaları uygulamışlardır. Metin işlemede metinler küçük harflere dönüştürülmüştür. Noktalama işaretleri, sayılar ve parolalar kaldırılmıştır.

Gereksiz kelimelerde sınırlandırmalara giderek bir kısmı çıkarılmıştır. Sonucunda bildirimlerin metinsel verilerini Bug ve Non-Bug olarak sınıflandırmışlardır (Pandey ve diğ. 2017).

2.1.1 Bildirim Takip Sistemleri Kullanım Amaçları

Çevik yazılım geliştirme, basit prensiplere dayalı yazılım geliştirme metotlarının genel adıdır. Yazılım geliştirme sürecine de çevik süreçler denir. Çevik süreçler açısından bakıldığında ayrıntılara yardımcı olmak için bildirim takibi kullanır. Bildirimler yeni işleri, hata ve sorunları temsil eder. Ekip aldığı bildirimleri Post-it notları şeklinde tahtaya yerleştirir. BTS' lerde bu post-it tahtasının web uygulamasıdır. Böyle olunca BTS'ler daha iyi bir ayrıntı sunar ve uzaktan çalışma imkânı sağlamaktadır. Ayrıca ekipler görevleri tartışmak istediğinde her zaman uygun olma avantajı sağlamaktadır. Bütün görevleri arşivlemeyi gerçekleştirmektedir (Verona ve diğ. 2016).

BTS uygulaması sorunları oluşturmak ve yönetmek için kullanılmaktadır. Bir BTS, aynı anda farklı projeleri yönetmemize yardımcı olmaktadır. İyi bir BTS, sorun çözme sürecinin 360 derecelik bir görünüm sağlamamıza yardımcı olmaktadır. Gönderilen sorunun durumu kolayca takip edilebilmektedir. Sorun gönderildikten sonra çözüm için uygun kaynaklara atanmaktadır (Pal ve diğ. 2012).

İşletmelerde işbirliği mekaniği çok önemlidir, çünkü hangi konuşmaların gerçekleştiğini anladığımızda, insanların nasıl sohbet ettiklerini optimize edebilirsiniz. Yazılım insanlar tarafından yazılır, insanlar hata yapar. Çoğu yazılım projesi, hata görüşmesi için BTS adı verilen özel bir yazılım kullanır. Bir BTS hata dosyası isterken, başka BTS hata hakkında daha fazla bilgi istemektedir. Böylelikle BTS hatadan etkilenenler için ipuçları ve bilgiler sunmaktadır. Başka bir BTS sadece düzeltme almaktadır. İnsanlar düzeltmeyi dener ve geri bildirimde bulunurlar. Konuşmanın özü aynıdır: paylaşılan bir dil, paylaşılan bir konu (hatayı araştırmak ve düzeltmek) ve bir iletişim kanalı (Bacon 2012).

Ortak çevrimiçi projeler, gelişmek için çeşitli destek araçlarına gereksinim duyar. En basit topluluk bile iletişim, iş saklama ve bilgi paylaşımı için araçlar

kullanmaktadır. Birçok topluluk, topluluktaki farklı işlevler için çeşitli araç ve yapılar kullanır. Farklı katkı sağlayan işletmelerin katkı türleri için farklı ihtiyaçları vardır. Geliştiricilerin hata izleyicilere, yama sistemlerine ve sürüm kontrolüne ihtiyacı vardır; dokümantasyon yazarlarının, web tarayıcıları üzerinden yeni sayfalar oluşturmaya, sayfalarda düzenlemeler yapmasına ve bu sayfaları birbirine bağlamaya olanak sağlayan bir yazılım altyapısı kullanan web sitelerine ihtiyacı vardır; çevirmenlerin çeviri araçlarına ihtiyacı vardır; test cihazlarının test takımlarına ihtiyacı vardır. Böyle bir toplulukta herkesin de birbiriyle iletişim kurması gerekir ve bu iletişim için etkili ve bütünlük bir araç kullanmak, birbirleri ile bağlantısı olmayan yapılar kullanmaya göre çok daha faydalı olacaktır (Bacon 2012).

Bir BTS geliştiricilerin BTS'ler üzerinde çalışmalarını takip etmelerini, birbirleriyle koordine olmalarını ve sürümleri planlamalarını sağlar. Herkesin hata durumunu sorgulamasını ve belirli hatalarla ilgili bilgileri kaydetmesini sağlar. Yalnızca hataları değil, görevleri, sürümleri, yeni özellikleri vb. izlemek için de kullanılabilir (Fogel 2005).

2.1.2 Bildirim Takip Sistemlerinde Karşılaşılan Problemler

Çoğu BTS iyi niyetli ancak deneyimsiz veya kötü bilgilendirilmiş kullanıcılar tarafından sunulan, çoğaltılmış veya geçersiz sayılan hataların bildirilmesi sorunu ile yüzleşmektedir. Bununla mücadelede ilk adım, genellikle BTS'nin ön sayfasına belirgin bir şekilde uyarıda bulunmak, bir hatanın gerçekten hata olup olmadığını nasıl söyleyeceğinizi, daha önce girilmiş olup olmadığını, nasıl arayacağınızı ve nihayetinde etkin bir şekilde nasıl raporlayacağınızı açıklamaktır. Biri hala yeni bir hata olduğunu düşünüyorsa bildirim girmelidir. Bu, bir süre için gürültü seviyesini azaltacaktır, ancak kullanıcı sayısı arttıkça, problem sonunda geri dönecektir. Hiçbir bireysel kullanıcı bunun için suçlanmamalıdır. Kullanıcıları gelecekte daha iyi sorun bildirmeleri ve projenin devamlılığı için ilk bildirimleri yararlı olmasa da teşvik edilmelidir. Bu sorunu önlemek için BTS'yi izleyen kişilerin, sorunları geçersiz kılmak için yeterli bilgiye sahip olan veya kabul edilecek bir bildirim olduğundan emin olmak ve diğer kullanıcıların açtıkları bildirimlerinden farklı olduğundan emin olmalıdır (Fogel 2005).

BTS'nin karşılaştığı sonraki problem ise bildirim kategorisinin yanlış seçilmesidir. Tabii ki, kullanıcılar bazen bir bildirim kategorisini yanlış tanımlayabilir, sonuçta bildirim yanlış kişiye gönderilmesi sonucu onu yeniden yönlendirmeleri gerekmektedir. Bununla birlikte, önemli olan, yükün hala paylaşıldığıdır. BTS'de bildirimler hala geliştiriciler arasında eşit bir şekilde dağıtılıyorsa, her bir bildirim zamanında yanıt alabilmektedir (Fogel 2005).

2.1.3 Bildirim Takip Sistemleri ve Yığın İzleri

BTS'ler yalnızca kullanıcılar ve test uzmanlarından bildirim almazlar. Aynı zamanda geliştirilen yazılımlar da kod parçacıkları düzenlenerek kritik hata durumlarında veya belirlenmiş durumlarda web servisleri aracılığı ile sisteme bildirim açabilmektedir. Bu bildirimlere isteğe bağlı metinler veya ek bilgiler verilebilmektedir. Yazılım geliştirilirken en çok kullanılan yöntem ise yığın izlerinin (stack traces) raporlarını bildirim metin olarak eklemektir.

Yığın izleri hata veya sorun ayıklama işlerinde ve çözümlerinde yazılım geliştiricilerini destekleyen programlama yapısıdır. Yazılım hata ayıklaması zordur ve çoğu zaman bir hatanın nedenini belirlemek için milyonlarca satırdan oluşan bir arama yapmak gerekmektedir. Ancak yığın izleri, hata ayıklamayı hızlandırmak için hata içerebilecek aday dosyalarının listesini potansiyel olarak daraltabilir (Schroter 2010). Yığın izleri yazılımın herhangi bir anında yazılımın hangi noktasında olduğu ile ilgili bilgileri vermektedir. Bunlar o andaki dosya, satır ve fonksiyon adı gibi bilgileri içermektedir.

Yığın izleri, bir programın yürütülmesi üzerine çağrılan kod yığınları içindeki aktif kodla ilgili raporları ve problemin kaynağına dair ipuçları vererek hata ayıklamaya yardımcı olmak için yaygın olarak kullanılmaktadır (Bettenburg ve Premraj 2008).

Schroter (2010), yaptığı çalışmada yazılımın bildirdiği hata raporlarının yaklaşık %60'nın sabit olduğunu bildirmektedir. Bildirime gönderilen yığın izi raporu, hatayı düzeltmek için gerekli kodun raporda belirtilen izlerde veya yakınlarında olduğunu belirtmektedir. Moreno ve diğ. yığın izlerinden gelen metin

ile yazılımın kaynak kodu arasında hatanın yerini bulma yaklaşımını geliştirmek için bir çalışma yapmışlardır. Çalışmalarında hata analizlerini başlık ve açıklama alanlarından, kod sınıfları için yazılımdaki metin açıklamaları ve yorumlardan almışlardır. Metin ön işlemleri yaparak anahtar kelimeleri çıkarmışlardır (Moreno ve diğ. 2014).

Jiang ve diğ. (2010) yaptıkları çalışmada Java programında yanlış atamadan dolayı oluşan hataların giderilmesi için program bölümlendirme ve yığın izlerini kullanmışlardır. Program bölümlendirme aracı ile arama alanını daraltıp yığın izlerindeki bilgileri takip ederek yanlış atamanın olduğu yeri tespit ederek hatayı gidermek için yaklaşımda bulunmuşlardır. Çalışmalarına tek başına olmamakla birlikte yığın izlerinin önemli katkısı olmuştur (Jiang ve diğ. 2010).

2.2 Veri Madenciliği

Veri madenciliği, anlamlı kalıpları ve kuralları keşfetmek için büyük miktardaki verilerin araştırılması ve analizidir. Veri madenciliğinin amacı, bir müşterinin davranışını daha iyi anlayarak bir şirketin pazarlama, satış ve müşteri destek işlemlerini iyileştirmesine yardımcı olunması anlamına da gelmektedir. Veri madenciliği tekniklerinin ve araçlarının; astronomi, tıp ve endüstriyel işlem kontrolüne kadar çeşitli alanlarda geçerliliğini sürdürmektedir. Veri madencisi; istatistiklerden, bilgisayar bilimlerinden ve makine öğrenimi araştırmalarından elde edilen bir kepçe tekniği kullanmaktadır. Belirli bir durumda uygulanacak belirli bir teknik kombinasyonunun seçimi; veri madenciliği görevinin, mevcut verilerin niteliğine ve veri madencisinin beceri ve tercihlerine bağlıdır. Veri madenciliğinde kullanılan modeller, analiz sonuçlarını değerlendirmede skor üretmek için kullanılmaktadır. Skor, bir modelin bulgularını tek bir sayıyla ifade etmenin bir yoludur (Berry ve Linoff 1997).

Veri tabanlarında, depolanan bilgi bakımından zengin verilerden ziyade, karar vericilerin sezgilerine dayanarak önemli kararlar alınmaktadır. Karar vericinin, çok miktarda verinin içine yerleştirilmiş değerli bilgiyi elde etmek için araçları bulunmamaktadır. Ayrıca, bilgiyi manuel olarak bilgi tabanlarına girmek için kullanıcılara veya etki alanı uzmanlarına dayanan mevcut uzman sistem

teknolojilerini göz önünde bulundurmak gerekmektedir. Bu durum, hatalara daha fazla eğilimli olmakla birlikte son derece zaman alıcı ve masraflıdır. Veri analizi yapan veri madenciliği araçları, önemli veri kalıplarını açığa çıkarabilir. Böylelikle Analistler; iş stratejilerine, bilgi tabanlarına, bilimsel ve tıbbi araştırmalara büyük katkı sağlayabilir. Bu yönüyle, bilgiyi 'altın külçelerine' dönüştürecek veri madenciliği araçlarının sistematik bir şekilde geliştirilmesine katkısı olacaktır (Han, Pei ve Kamber 2011).

Veri madenciliğindeki tipik bir örnek, hangi ürünlerin raflara yakın yerleştirileceğini tahmin etmek veya benzerleri için kuponlar sunarak tüketici satın alma modellerini kullanmaktır. Örneğin, bir el feneri alırsanız, onunla birlikte pil satın almanız da olasıdır. İlgili bir uygulama, kredi kartı kullanımında olduğu gibi sahtekârlığın otomatik tespitidir. Analistler, normal harcama modellerinden sapmaları bulmak için çok sayıda kredi kartı kaydına bakarlar. Klasik bir örnek, denizaşırı bir uçağın uçuşundan sonra az miktarda benzin almak için bir kredi kartının kullanılmasıdır (Hearst 2003).

2.2.1 Veri Madenciliği Teknikleri

Her veri madenciliği tekniği, modelleme hedefine bağlı olarak farklı bir amaca hizmet etmektedir. En yaygın iki modelleme sınıflandırma ve tahmindir. Sınıflandırma modelleri kategorik etiketleri (ayrık, sırasız) değerlendirirken, öngörü (tahmin) modelleri sürekli değişken değerleriyle ilgilenmektedir. Veri madenciliğinde Karar Ağaçları, Regresyon, Birliktelik Kuralı, Kümeleme algoritmaları gibi teknikler kullanılmaktadır (Palaniappan ve Awang 2008).

Karar Ağacı, mevcut tüm özellikleri aynı anda kullanan ve her piksel için tek bir üyelik kararı veren geleneksel istatistiksel ve sinirsel sınıflandırıcıların aksine, karar ağacı etiket ataması sorununa çok aşamalı veya sıralı bir yaklaşım kullanmaktadır. Etiketleme işleminin, tek ve karmaşık bir karardan ziyade sıralı testlerin sonuçlarına dayanan basit kararlar zinciri olduğu düşünülmektedir. Karar dizileri ise, düğümlerde uygulanan testlerle karar ağacının dallarını oluşturur (Pal ve Mather 2003).

Bağımlı değişkenler arasındaki ilişkileri araştıran regresyon analizi birçok araştırma projesinin temel parçasıdır. Örneğin; sınıf kapasitesinin değişiminin öğrencilerin başarısına etkisinin araştırılması, bir gayzerin bir sonraki patlamasının önceki patlamalardan tahmin edilmeye çalışılması, diyetteki değişimlerin yaş, cinsiyet ve egzersiz miktarına bağlı olarak kolesterol seviyesine etkisinin araştırılması, kişi başına gelire göre ülkelerin doğum oranlarının analiz edilmesi gibi çalışmalarda ilk akla gelecek analiz yöntemi olarak regresyonu analizi örnek verilebilir (Weisberg 2005).

Kümeleme, veri tabanına kaydedilmiş veri setlerinin veya belirli bir grubu oluşturmuş olan nesnelerin birbirleriyle karşılaştırıldığında yüksek benzerliğe sahip olacak şekilde oluşturulmasına, ancak diğer kümelerdeki nesnelere çok benzememesine denilmektedir (Han ve diğ. 2011). Bir başka deyişle kümeleme, bir veri setindeki bilgileri bazı yakınlık kıstaslarına göre gruplara ayırmaktır. Bu gruplar arasında benzerlik oranı en az grup içinde benzerlik oranı en çok olmalıdır. Kümelemede amaç, ele alınan örnekte var olduğu bilinen bireylerin veya nesnelerin, arasındaki benzerliklere dayanan farklı sayıdaki özel grupları oluşturmak, daha sonra bu grup elemanlarının profilini ortaya çıkarmaktır (PN Tan ve diğ. 2006).

2.2.2 Birliktelik Analizi

Birliktelik analizi, mağazaların işlem veri tabanlarından ilişkilendirme yapmak veya ortak oluşumlar çıkararak beraber kullanılma modellerini keşfetmede yararlı bir yöntemdir. Birçok alanda kullanılan birliktelik analizinin en sık kullanıldığı alanlar ise pazarlama, satış, servis ve işletme alanlarıdır. Birliktelik analizi yapılarak müşteri davranışları ortaya çıkartılmaya çalışılmaktadır. Analizden elde edilen bilgiler bu alanların stratejilerinin oluşturulmasında kullanılabildiğinden son derece önemlidir. Bununla birlikte, mevcut yöntemler, ürünlerin tüm mağazalarda her zaman rafta olduğu varsayımı nedeniyle, çok mağazalı bir ortamda önemli satın alma modellerini bulamayabilir. Bu yönüyle de, birliktelik analizi mevcut yöntemlerden oldukça farklıdır (Chen ve diğ. 2005). En çok kullanılan alanlar pazarlama, satış, servis ve işletme olsa da daha pek çok alanda kullanımı bulunmaktadır. Bu çalışmada kelimelerin projeler ile birlikteliğine bakılmaktadır.

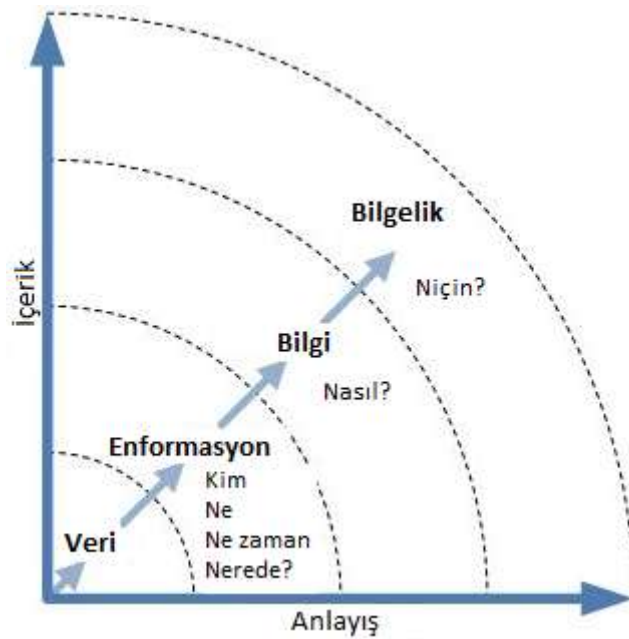
Birliktelik analizi, müşterilerin çok katmanlı ürünler veya hizmetler arasındaki değişimlerini ölçmek için pazar araştırmasında en çok kullanılan yöntemler arasında yer almaktadır. Bu yöntem, tüketicilerin bir ürünün veya hizmetin toplam değerini, özelliklerinin her birinin bireysel değerini birleştirerek değerlendirdiğini varsayar. Bu nedenle birliktelik analizi, tüketiciler için potansiyel ürün veya hizmetleri temsil eden özellikler kümesi hakkında değerli bilgiler sağlar ve pek çok araştırmacı ve uygulayıcı, yeni ürün fırsatlarını, pazar bileşenlerini belirlemek ve en uygun fiyat ve reklam stratejilerine karar vermek için kullanmaktadır. Özellikle şirketlerin fayda bileşenlerini oluşturmalarına ve ayrıca çeşitli özellikler arasında takaslar konusunda karar vermelerine izin vermektedir. Ayrıca, ürün tasarımlarının belirli bir rekabet ortamı için satışları ve karları en üst düzeye çıkarmak için en uygun ürün yapılandırılmalarını geliştirmelerine yardımcı olmaktadır (Yoon ve Park 2007).

Horiuchi ve diğ. 2014'te birliktelik analizini Japonya 2014 Temsilciler Meclisi seçimlerinde kullanmışlardır. Partilerin seçim manifestolarını kullanılarak hazırlanmış olan seçim anketleri yapılmıştır. Seçmenlerin anket sorularına verdikleri cevaplar politik tercihlerini de ortaya koymaktadır. Gerçek parti manifestolarına dayanan seçim anketinde birliktelik analizinin, seçmenlerin temsili demokrasilerdeki çok boyutlu politika tercihlerini anlamada daha iyi bir yaklaşım olduğuna inanmışlar ve yaklaşımlarının gelecekteki seçim analizlerine dahil edilmesi gerektiğini savunmuşlardır (Horiuchi ve diğ. 2014).

Almario ve diğ. birliktelik analizi kullanarak, biyolojik terapiye karar vermenin etkenlerinin ülseratif kolit ve chorn hastalığı olan (bağırsaklarda kronik iltihaplı bir hastalık) hastalar arasında değişiklik gösterdiğini tespit etmişlerdir. Bu çalışma biyolojik terapiye karar vermenin oldukça bireysel olduğunu vurgulamaktadır. Tedavi tercihlerini belirlerken hastaların kişisel ve demografik özelliklerinin yanı sıra klinik değerlerinin etkili olmadığı birliktelik analizi kullanılarak gösterilmiştir (Almario ve diğ. 2018).

2.3 Metin Madenciliği

Veriler, nesnelerin ve olayların özelliklerini temsil eden sembollerdir. Enformasyon, verilerin bir takım işlemlerden geçirilerek daha kullanılabilir hale getirilmesiyle oluşur. Örneğin, Nüfus Müdürlüğü veri toplar, bu verileri işler, İstatistiksel olarak birçok tabloda sunulan bilgilere dönüştürür. Veriler gibi, bilgiler de nesnelerin ve olayların özelliklerini temsil eder, ancak verilerden daha karmaşık ve kullanışlıdır. Veri ile enformasyon arasındaki fark yapısal değil işlevseldir. Enformasyon; kimin, neyin, ne zaman, nerede ve kaç olduğu gibi soruların cevaplarında bulunur. Bilgi ise nasıl yapılır sorusunun cevabıdır. Bilgelik ise niçin sorusunun cevabıdır; enformasyon ve bilgi ile beraber verimliliği arttırmamızı sağlar. Bu kavramlar bilgi bilimin yapı taşlarını oluşturmaktadır. Veriler anlaşıldıkça bilgiye yaklaşım artmaktadır (Ackoff 1989). Bu kavramlar arasındaki ilişki Şekil 2.1’de verilmiştir (Web-1).



Şekil 2.1 : Bilgi Yönetim Grafiği (Web-1)

Veri, basit herhangi bir şekilde var olan, varlığın ötesinde bir anlam taşımayan, ham olarak bulunan sembollerdir. Enformasyon veriler arasında ilişkisel bağlantı yoluyla anlamlandırma, kullanılabilir olan veya olmayan veri setleridir. Bilgi fayda sağlayan enformasyon topluluğudur. Bilgisayarda bulunan uygulamaların

çoğunda (modelleme, simülasyon vb.) bir çeşit depolanmış bilgi kullanılır (Bellinger 2004).

Metin madenciliği, yapılandırılmamış metin verilerinden anlamlı bilgiler çıkarmaya çalışan çok yeni olmayan bir teknolojidir. Metin madenciliği, veri madenciliğinin metinsel veriye bir uzantısıdır. Çok sayıda metinsel belgeden yararlı bilgileri hızlıca toplamak için, otomatik bilgisayar tekniklerini kullanmak zorunlu hale gelmiştir. Metin madenciliği, metin dosyaları, HTML dosyaları, sohbet mesajları ve e-postaların içeriği gibi yapılandırılmamış metin verilerinden faydalı modeller, trendler, desenler veya kurallar bulmaya odaklanmaktadır. Otomatik bir teknik olarak, metin madenciliği “metinlerden bilgiyi verimli ve sistematik olarak belirlemek, çıkarmak, yönetmek, entegre etmek ve kullanmak” için kullanılabilir. Birçok araştırmacı; işletme, sağlık bilimleri ve eğitim alanlarındaki büyük miktarda metin verilerini analiz etmek için metin madenciliği tekniklerini başarıyla kullanmıştır. He ve diğ. (2013) dijital kütüphanedeki belgelerden meta verileri çıkarmak ve metindeki uygun öğeleri işaretleyerek belgeleri zenginleştirmek için metin madenciliği tekniklerini kullanmışlardır (He ve diğ. 2013).

Bilgi depolamanın en doğal şekli metin olduğundan, metin madenciliğinin, veri madenciliğinden daha yüksek bir ticari potansiyele sahip olduğuna inanılmaktadır. Aslında, son zamanlarda yapılan bir çalışma, bir şirketin bilgisinin % 80'inin metin belgelerinde bulunduğunu göstermiştir. Bununla birlikte, metin madenciliği, doğal olarak yapılandırılmamış ve bulanık olan metin verilerini ele almayı içerdiğinden (veri madenciliğinden) çok daha karmaşık bir iştir. Metin madenciliği, bilgi alma, metin analizi, bilgi çıkarma, kümeleme, sınıflandırma, görselleştirme, veri tabanı teknolojisi, makine öğrenmesi ve veri madenciliğini içeren çok disiplinli bir alandır (Tan 1999).

Metin madenciliği algoritmaları, temel belgelerin kendisinde değil belgelerin özellik tabanlı temsilleri üzerinde çalıştığından, genellikle iki önemli hedef arasında bir denge vardır. İlk hedef, bir belgenin anlamını doğru bir şekilde göstermek için özelliklerin anlamsal ve anlamsal seviyelerinin doğru kalibrasyonunu sağlamaktır. İkinci hedefte kalıpları pratik bir şekilde tanımlamaktır. Ayrıca kelime setleri, sözlükler, eş anlamlılar veya ontolojiler ile desteklenebilir. Metinleri temsil etmek için pek çok özellik kullanılabilse de en çok dört özellik kullanılmaktadır. Karakterler, kelimeler, terimler ve kavramlar. Karakter, diğer özelliklerin yapı taşı olup harf, sayı veya özel sembollerden oluşur. Metin madenciliğinde karakterler bazı

özel durumlar haricinde pek kullanılmazlar. Kelime, anlamsal zenginliğin temel seviyesi olarak kabul edilen karakterler topluluğudur. Terim, belli bir alana özgü özel kelime veya kelime gruplarına denir. Kavram, belge veya metinlerde bulunmayan fakat kategori olarak birebir ilişkisi olan çapraz referanslama yapılabilen kelime veya kelime gruplarıdır (Feldman ve Sanger 2007).

Akıllı metin analizi, metin içindeki bilgi keşfi olarak da bilinen metin madenciliği, genellikle bir metinden ilgi çekici bilginin çıkarılması sürecini ifade eder. Metin madenciliği bir takım süreçlerden oluşur. Metin kümeleme ve sınıflandırma, bilgi keşfi, adlandırılmış varlık tanımlama, doğal dil işleme, soru cevaplama ve görselleştirme alanlarını kapsamaktadır (De Bruijn ve Martin 2002).

2.3.1 Metin Madenciliği Süreçleri

2.3.1.1 Veri Toplama

Metin madenciliği ve veri madenciliğindeki ilk adım farklı ortamlarda ve şekillerde bulunan metinlerin ve verilerin toplanmasıdır. Her işletme verilerini farklı şekillerde ve ortamda tutmaktadır. Analiz için sadece tek bir işletmeden alınacak veriler yeterli olmayabilir. Farklı işletmelerden de almak gerekebilir (Tang ve MacLennan 2005).

2.3.1.2 Temizleme ve Dönüştürme

Metin madenciliğindeki ikinci adım verilerin temizlenmesi ve dönüştürülesidir. Veri temizliği değersiz verilerin düzenlenmesi, gürültülerin giderilmesi ve tutarsız verilerin düzenlenmesidir. Değersiz veriler değerlendirilerek hiç dikkate alınmayabilir veya uygun bir değer belirlenip atama yapılarak temizleme yapılmaktadır. Gürültü ise veri tipine uygun değerlerin gelmemesi olarak ortaya çıkmaktadır. Bu değerlere en yakın komşu değer, hesaplanabileceksen hesaplanarak veya grubun en düşük, yüksek, ortalama değerlerinden biri verilerek temizlenmektedir. Tutarsız veri, almış olduğu değerlerin gelme ihtimali hiç olmayan veri demektir. Bunu temizlemek elle yapılabilir, tasarlanmış rutinlerle veya bilgi

mühendisliği araçları ile düzeltilebilir. Temizleme işlemi yapılırken veya sonrasında normalizasyon, yumuşatma, genelleme ve ortalama işlemleri isteğe göre uygulanarak dönüşüm işlemi yapılabilmektedir (Han ve diğ. 2011).

2.3.1.3 Model Belirleme ve Uygulama

Veriler temizlendikten ve dönüşüm yapıldıktan sonra, modeller oluşturmaya başlanır. Herhangi bir model oluşturmadan önce, metin madenciliği projesinin amacını ve metin madenciliği görevinin türünü anlamak gereklidir. Bu proje bir sınıflandırma görevi, bir atama görevi veya bir bölümlendirme görevi midir? Bu aşamada, alan bilgisine sahip iş analistleriyle takım oluşturmak gereklidir. Doğru yaklaşım, farklı algoritmalar kullanarak birden fazla model oluşturmak ve daha sonra bir sonraki adımda bazı araçlar kullanarak bu modellerin doğruluğunu karşılaştırmaktır (Tang ve MacLennan 2005).

2.3.1.4 Tahminleme ve Raporlama

Metin ve veri madenciliği projesinde kalıpları ortaya çıkarmak işin yarısıdır asıl amaç eğitim modellerini uygulayarak tahminleme yapmaktır. Tahminleme yapmak için eğitilmiş bir model ve yeni veriler gerekmektedir. Raporlama metin madenciliği sonucu üretilen bilgiler veya tahminler için önemli bir dağıtım kanalıdır. Birçok işletme yöneticilerine rapor sunmaktadır. Çoğu metin veya veri madenciliği aracı, kullanıcıların metinsel veya grafik çıktıları olan madencilik modellerinden önceden tanımlanmış raporlar oluşturmasını sağlayan raporlama özelliklerine sahiptir (Tang ve MacLennan 2005).

2.3.2 Metin Madenciliği ve Veri Madenciliği

Metin ve veri madenciliği, makine tarafından okunan materyallerden bilgi üretme işlemidir. Büyük miktardaki materyaller kopyalanır ve verileri çıkarılır. Çıkarılan örnekleri tanımlamak için yeniden birleştirmeler yapılır.

Van Driel ve diğ. (2006) tarafından yapılan çalışmada insan hastalık genetiğinin modüler yapısını yansıttığı için hastalıklara yönelik aday genlerin yanı sıra genler ve proteinler arasındaki fonksiyonel ilişkileri tahmin etmek için kullanılabilme özelliğine sahip fenotip haritalama konusu üzerinde durmuşlardır (Van Driel ve diğ. 2006). Çalışmada 5000'in üzerinde insan fenotipi metin madenciliği yöntemleri ile sınıflandırılmış ve fenotipler arasındaki benzerlikler ortaya koyulmaya çalışmıştır.

He ve diğ. (2013) tarafından yapılan çalışmada, pizza endüstrisindeki firmaların metin madenciliği ile değerlendirmesi yapılmıştır. Çok sayıda kullanıcı tarafından oluşturulan içerik, sosyal medya sitelerinde serbestçe kullanılabilir. Bu içerikler, şirketlerin bir sosyal medya rekabet analizini nasıl gerçekleştireceklerini anlamalarına ve sosyal medya verilerini karar vericiler ve e-pazarlamacılar için bilgiye dönüştürmelerine yardımcı olmaktadır. Facebook ve Twitter'daki yapılandırılmamış metin içeriğinin en büyük üç pizza zinciri olan Pizza Hut, Domino's Pizza ve Papa John's firmaları için bir vaka çalışması yapılmıştır. Şirketlerin sosyal medya rekabet analizi stratejilerini geliştirmelerine yardımcı olması için önerilerde bulunulmuştur (He ve diğ. 2013).

Nasukawa ve Nagano (2001) tarafından yapılan çalışmada, çok miktardaki metin verilerinden bilgi elde etmek için metin madenciliği kullanılmıştır. Çalışma, kullanıcının gereksinimlerini ve ilgi alanlarını karşılayan belgeleri seçmesine izin veren bilgi alma teknolojilerinden veya belgeleri düzenleyen belge kümeleme teknolojilerinden farklı olarak, metin içindeki belirli konulardaki eğilimleri ve önemli özellikleri gösteren önemli desenleri ve kuralları bulmaya odaklanmıştır. Önerilen prototip bir sistem ile metin veri tabanlarına uygulayarak ürün hatalarını otomatik olarak tespit edilebilmektedir. Bunun yanı sıra, herhangi bir metni okumadan, yardım merkezi verimliliğini ve belirli bir ürünü içeren müşterilerin davranışlarındaki değişiklikler analiz edilebilir (Nasukawa ve Nagano 2001).

Mittermayer (2004) tarafından yapılan çalışmada, metin madenciliği teknikleri ile gün içi hisse senedi fiyat eğilimlerinin tahmini yapılmıştır. Basın bültenlerinin yayınlanmasından hemen sonra hisse senedi fiyat eğilimlerini tahmin etmek için uygulanan bir sistem olan NewsCATS oluşturulmuştur. NewsCATS, başlıca üç bileşenden oluşmaktadır. İlk bileşen, metin ön işleme tekniklerinin uygulanması yoluyla basın bültenlerinden alakalı bilgileri alır. İkinci bileşen, basın açıklamalarını önceden tanımlanmış kategorilere ayırır. Son olarak bulgu, basın

bültenlerinin sınıflandırılmasının, hisse senedi fiyat eğilimlerini tahmin etmek için kullanılabilecek ek bilgiler sağlayabildiği gözlemlenmiştir (Mittermayer 2004).

Li ve Wu (2010) tarafından yapılan çalışmada, çevrimiçi forumlardan etkin bağlantıyı tespit etmek için duygu analizi ve metin madenciliği tekniklerinden destek vektör makineleri (SVM) kullanılmıştır. Duygu analizi cümlelerin kelimelerinin değerlendirilerek cümlelerin olumlu veya olumsuz şekilde analiz edilmesidir. İlk olarak, duygu analizi ile çevrim içi forumlardaki metinlerin polaritesi belirlenmiştir. Ardından, SVM ile tahmin yapılmıştır. Yapılan çalışmada, 31 farklı konum ve 220.053 mesaj incelenmiştir. Kullanılan yöntemlerin başarılı sonuçlar verdiğine değinilmiştir (Li ve Wu 2010).

Raorane ve diğ. (2012) tarafından yapılan çalışmada, birliktelik analizi kullanılarak bilgi çıkarımı yapılmıştır. Bu çalışmanın amacı, büyük miktarda veriyi analiz etmek ve tüketici davranışından faydalanmak ve rakipler üzerinde rekabet üstünlüğüne yol açan doğru kararı vermektir. Kullanılan yöntemin, müşterilerin bazı belirli ürünleri satın alırken yaptıkları işlemlerin dinamik davranışlarıyla ilişkili kalıpları optimize etmek için etkili bir şekilde kullanılabileceğini göstermektedir. Sonuç olarak yöntem, ürünler arasında belirli ilişkiler olduğunu ortaya çıkarmıştır (Raorane ve diğ. 2012).

Aguinis ve diğ. (2012) tarafından yapılan çalışmada, birliktelik analizini yönetim araştırmalarında kullanılması amaçlanmıştır. Çalışmada kullanılan analiz, yönetim alanında dağınık olarak var olan ve kullanılmayan verilerin değerlendirilmesine olanak sağlamaktadır. Ayrıca analiz, dinamik teoriler geliştirmeye yardım edebilen ve analiz birimleri arasında ilişki kurmak için uygundur. Birliktelik analizi, insan kaynakları yönetimi, örgütsel davranış, girişimcilik ve stratejik yönetim gibi yönetim alanlarında geleceğe yönelik bir bakış açısı sağladığı vurgulanmıştır (Aguinis ve diğ. 2012).

Kaur ve Kang (2016) tarafından yapılan çalışmada, birliktelik analizi kullanılarak piyasa verilerinin değişen eğilimleri belirlenmiştir. Birliktelik analizi, satıcıya doğru karar vermede yardımcı olabilecek ve alıcının satın alma davranışını anlama konusunda perakendeciye bilgi sağlamaktadır. Fakat bu durumun analizinde çeşitli algoritmalar kullanılmaktadır. Çalışma, AIS algoritması, Apriori Algoritması, FP-Ağaç Algoritması (Frequent Pattern-Tree Algorithm), RARM (Rapid Association Rule Mining) Algoritması hakkında bilgi sağlamaktadır. Birliktelik analizinin geliştirmekte ve uygulamasının kolay olduğundan söz edilmiştir (Kaur ve Kang 2016).

Cunningham ve Frank (1999) tarafından yapılan çalışmada, kütüphane dolaşım verileri kullanılarak birliktelik analizi yapılmıştır. Birliktelik analizi, tüketici satın alma modellerinin analizinde, özellikle sık sık satın alınan ürünlerin tespitinde yaygın olarak kullanıldığı görülmüştür. Apriori algoritması ile bir üniversite kütüphanesinden ödünç alınan kitapların sınıflandırma kategorilerini tespit etmesine yönelik uygulama gerçekleştirilmiştir. Elde edilen bilgiler, kullanıcıları bilgi gereksinimlerine uygun belgeler içerebilecek koleksiyonun ek bölümlerine yönlendirmede ve bir kütüphanenin fiziksel düzenini belirlemede faydalı olacağı düşünülmüştür. Bunun yanı sıra, sonuçlardan çıkarılan sınıflandırma şemasının farklı bir görüş sağladığına değinilmiştir (Cunningham ve Frank 1999).

Trnka (2010) tarafından yapılan çalışmada, birliktelik analizini Altı Sigma metodolojisine uygulanmıştır. Altı Sigma'nın istatistiksel birkaç yöntem kullandığı düşünülerek birliktelik analizi kullanılmıştır. Birliktelik Analizi, Altı Sigma'ya (bir aşamasına) uygulanmasıyla sonuçları iyileştirebilir ve sürecin Sigma performans seviyesini değiştirebilmektedir. Araştırmada, GRI (General Rule Induction) algoritmasını kullanarak veriler arasında ilişki kurulmuştur. Analizin son aşamasında C5.0 algoritması kullanılarak kural tabanlı profiller oluşturulmuştur. Birliktelik analizi kullanılarak, Altı Sigma performans seviyesi dolaylı olarak yükseltilebileceği vurgulanmıştır (Trnka 2010).

2.4 Doğal Dil İşleme

İşletmeler, devlet kurumları ve bireyler; çalışma ve yaşam için kritik önem taşıyan, fakat potansiyeli yüksek olan, sakladıkları değerleri elde etmek için yeterince iyi anlaşılmayan çok miktarda metinle giderek artan bir şekilde karşı karşıya kalmaktadırlar (Manning 1999). Metinlerin anlaşılması insanlar için kolay olsa da tüm metnin anlaşılması ve bilgi çıkarımı çok zaman alacaktır. Bu yüzden metinleri bilgisayara anlatmak gerekmektedir. Bilgisayarın insanları ve metinleri anlamalarını sağlamak için birçok problem vardır. Birincisi belirsizlik: kelimelerin ve cümlelerin anlamlarının çok olması nedeniyle ne anlama geldiği konusundaki kafa karışıklığı. İkincisi Düzensizlik: düşünceler bazen belirsiz ve yanlış terimlerle ifade edilir. Üçüncüsü eksiklik: tüm fikir sunulmaz ve dinleyiciden “satırlar aralarını” okuması beklenir. Dördüncüsü Yanlışlık: heceleme, noktalama işaretleri ve dilbilgisi

problemleri anlamı gizleyebilir. Bu sorunları azaltmak için doğal dil işleme teknikleri geliştirilmiştir. Böylece metnin söz dizimi ve anlamı analiz edilmeye çalışılmaktadır (Web-2).

“Doğal dil” insanlar tarafından günlük iletişimde kullanılan dillerdir. Programlama dilleri ve matematiksel gösterimler gibi yapay dillerin aksine, doğal diller kuşaktan kuşağa geçerken gelişti ve açık kurallarla saptamak zor olmaktadır. Herhangi bir doğal dilin bilgisayar manipülasyonunu kapsaması için doğal dil işleme kullanılmaktadır. Bir uçta doğal dil işleme, farklı yazma stillerini karşılaştırmak için kelime frekanslarını saymak kadar basit olabilir. Diğer uçta ise, en azından onlara yararlı cevaplar verebilecek ölçüde, insan söylemlerini tamamlamayı “anlamayı” içerir. Doğal dil işleme bireysel çalışmalarda, hesaplamalı dilbilimlerinde, yapay zeka, metin madenciliği veya korpus dilbiliminde kullanılabilir (Bird ve diğ. 2009).

Doğal dil işleme teknikleri her dilin gramer yapısına uygun olarak kelimeleri cümleleri ayrıştıran düzelter, düzenleyen ve bilgisayarlar için anlamlı hale getiren süreçlerdir. Bu süreçler için geliştirilmiş olan pek çok kütüphane bulunmaktadır. Bunlardan bir tanesi de Java’da geliştirilmiş olan ve Türkçe içinde desteği bulunan Zemberek’tir.

2.4.1 Türkçe Dili

İnsanlar çevrelerini anlama ve yorumlama sürecini ana dilleri ile yaparlar. Dillerin amacı insanlar arası iletişimi sağlamaktır. Dünya üzerinde kullanılmakta olan pek çok anadilden farklı bir yapısı olan kullanılmakta olduğumuz Türkçe Dili’dir. Türkçe Türkiye Cumhuriyetine mensup olan vatandaşların konuştuğu dildir. Türkçe 11. Yüzyılda Anadolu’yu fethedip üç kıtada devletler kuran Oğuz Türklerinin dilidir (Tekin ve Ölmez 1999). Bir dilin özelliklerinin belirlenmesi doğal dil işleme Sistemlerinin geliştirilmesi için büyük önem arz etmektedir (Aşlıyan ve diğ. 2006).

Türkçe Ural-Altay dil grubu içerisindedir ve sondan eklemeli bir dildir. Türkçe dünya en çok konuşulan yedi dil arasında olup iki yüz milyondan fazla kişi konuşabilmektedir (İşcan 2011). Tablo 2.1’de Türkçe bir kelimeye ait sondan ekleme

örneđi verilmiřtir. Türkçe dilinde iki adet ünlü uyumu bulunmaktadır. Büyük Ünlü Uyumu; Türkçede kelime ilk hecesinde kalın bir ünlü (a, ı, o, u) varsa diđer hecelerinde bunlarla devam etmesi, eđer ilk hecede ince bir ünlü (e, i, ö, ü) varsa diđer hecelerinde bunlarla devam etmesi kuralıdır. Küçük Ünlü Uyumu ise kelimedeyuvarlak ünlülerin (o, ö, u, ü) ardından dar yuvarlak ünlü (u, ü) veya düz geniş ünlü (a, e) gelmesi veya düz ünlülerin (a, e, ı, i) ardından düz ünlülerin gelmesi kuralıdır (TDK 1983).

Türkçede sona gelen ekler iki ana gruba ayrılır. Bunlar Yapım Ekleri ve Çekim Ekleridir. Çekim ekleri kelimenin farklı biçimlerini ifade ederken, yapım ekleri ile yeni bir kelime oluřturmaktadır (Erdem, 2011). Yapım Ekleri isimden isim, isimden fiil, fiilden isim, fiilden fiil olmak üzere dört gruba ayrılır. Çekim Ekleri ise İsim Çekimleri ve Fiil Çekimleri olarak ikiye ayrılır. Tablo 2.2’de Yapım Eklerine örnekler verilmiřtir. Tablo 2.3’te ise Çekim Ekleri ve alt grupları için örnekler vardır.

Tablo 2.1: Türkçe Sondan Ekleme Örneđi

| Türkçe | İngilizce |
|---------------|------------------------------------|
| Göz | eye |
| Gözlük | eyeglasses |
| Gözlükçü | optician:someone who sells glasses |
| Gözlükçülük | the business of selling glasses |

Tablo 2.2: Yapım Ekleri Örnekleri

| Kelime | Yapım Eki | Yeni Kelime | Yapım Türü |
|---------------|------------------|--------------------|-------------------|
| Tuz | -luk | Tuzluk | İsimden isim |
| Sanat | -çı | Sanatçı | İsimden isim |
| Kan | -a | Kanamak | İsimden fiil |
| Az | -al | Azalmak | İsimden fiil |
| Asmak | -kı | Askı | Fiilden isim |
| Seçmek | -im | Seçim | Fiilden isim |
| Kovmak | -ala | Kovalamak | Fiilden fiil |
| Atmak | -ıl | Atılmak | Fiilden fiil |

Tablo 2.3: Türkçe Çekim Ekleri Örneği

| Kelime | Çekim Eki | Yeni Kelime | Çekim Eki Türü |
|---------------|---|---|-----------------------|
| Kuş | -lar, -ler | Kuşlar | Çokluk eki |
| Ev | -i, -e, -de, -den | Evi, Eve, Evde, Evden | Hal Eki |
| Bebek | -in, -in, -ün , -un | Bebeğin | İlgi Eki |
| Ben | -m, -n, -miz, -mız | Benim | İyelik Eki |
| Aile | -çe, -ça, -ce, -ca | Ailece | Eşitlik Eki |
| Yapmak | -yor, -ecek, -tı ... vs zaman ekleri | yapıyor, yapacak, yaptı | Kip Eki |
| Gelmek | -m, -n, -k, -niz, -ler | geldim, geldin, geldik, geldiler ...vs | Şahıs Eki |

Türkçenin zor bir dil olmasının sebeplerinden biri de ünsüz yumuşamasıdır. ‘p, ç, t, k’ ünsüz harfleri ile biten kelimeler ünlü harf ile başlayan ek aldıklarında yumuşayarak ‘b, c, d, ğ’ harflerine dönüşür. Örneğin “bıçak” kelimesi sonuna ‘ı’ yönelme eki aldığında “bıçağı” olmaktadır (Kahraman, 2010). Benzer bir şekilde ünsüz benzeşmesi vardır. Buda Türkçe kelimeye gelen ekinin ünsüz harfinin kelimeye uyararak başka bir ünsüze dönüşmesidir. Buna örnek olarak –ki eki verilebilir; “bugün-ki” “bugünkü”, “dün-ki” “dünkü”. Türkçenin diğer bir özelliği ise ses düşmesidir veya türemesidir. Türkçede bazı kelimeler sonlarına aldıkları eklerle birlikte son hecedeki harflerden bazıları düşer veya kelimeye ek harfler gelir (Dursunoğlu 2010). Tablo 2.4’te bunlara örnekler verilmiştir.

Tablo 2.4: Türkçe Ses Türemesi ve Düşmesi

| Kelime | Kelime + Ek | Ek Almış Hali | Ses Olayı |
|---------------|--------------------|----------------------|------------------|
| ağız | ağız - ı | ağzı | Ses düşmesi |
| oğul | oğul - u | oğlu | Ses düşmesi |
| sızı | sızı - lamak | sızlamak | Ses düşmesi |
| az | az - cık | azıcık | Ses Türemesi |
| gelmek | gel - yor | geliyor | Ses Türemesi |

2.4.2 Zemberek

Zemberek Türkçe için geliştirilmiş açık kaynak kodlu bir doğal dil işleme kütüphanesidir. Bu kütüphane içerisinde yazım denetimi, çalışmamızda da kullanılan hatalı veya yanlış kelimeler için öneri, Türkçe harf düzeltme işlemi (geleceğim kelimesindeki ikinci g harfinin ğ ile değiştirilmesi gibi), fazlalık harflerin çıkartılması (çıkartmak kelimesindeki t harfinin birisin atılması gibi) gibi dil işlemlerini gerçekleştirebilmektedir. Java ile geliştirilmiş bir kütüphanedir. Zemberek metin kullanılan pek çok farklı projeye veya farklı ortamlara adapte edilerek kullanılabilir.

Akın ve Akın (2007) tarafından yapılan çalışmada, Türk Dili için açık kaynak kodlu olan Zemberek ile doğal dil işleme çerçevesinde bir değerlendirme yapılmıştır. Bilişim dünyasındaki doğal dil işleme çözümlerinin çoğu Hint-Avrupa dillerine dayanmaktadır. Dillerin kendine özgü zorlukları ve geniş kapsamlı, genel amaçlı bir NLP kütüphanesi için çözüm eksikliği söz konusudur. Zemberek projesi, Türkçe dili ile bu alanı doldurmanın yanında tüm Türk dilleri için esnek, açık kaynak kodlu, platformdan bağımsız bir doğal dil işleme çerçevesi sunmayı hedeflemektedir (Akın ve Akın 2007).

Gemci ve Peker (2013) tarafından yapılan çalışmada, Türkçe tweetleri konularına göre ayırmak için LDA (Latent Dirichlet Allocation) kullanılmıştır. Tweet metinlerini analize uygun duruma getirmek için ilgili veri setinde olan kelimeler köklerine ayrılmıştır. Bu işlem gerçekleştirilirken Zemberek Kütüphanesi kullanılmıştır. Tweet'lerde kullanılan dil ve yaygın yazım hataları metin analizini güçleştirmektedir. Zemberek yardımıyla yeni kelimeler eklenerek sorunlu olan kelimeler ortadan kaldırılmıştır. Sonuç olarak Zemberek Kütüphanesinin yapılacak analiz için son derece önemli olduğu vurgulanmıştır (Gemci ve Peker 2013).

Alparslan ve Bahsi (2009) tarafından yapılan çalışmada, Türkçe dilinde yazılan gizli belgelerin güvenlik seviyelerinin sınıflandırması yapılmıştır. Çeşitli güvenlik seviyelerine sahip (sınıflandırılmamış-kısıtlı-gizli) TÜBİTAK UEKAE'nin belgeleri, SVM ve Bayes algoritmaları kullanılarak sınıflandırılmıştır. Belgelerdeki ilişkiyi görmek için bir ağırlık matrisi olan TF-IDF (Term Frequency-Inverse Document Frequency) metriği seçilmiştir. Türkçe dili, eklemeli diller grubuna

girmektedir ve İngilizceye kıyasla farklı zorluklar içeren bir doğal dil işleme sorunu oluşturduğu için Zemberek kullanılarak kelimeler köklerine ayrılmıştır (Alparslan ve Bahsi 2009).

2.5 Literatür Araştırması

Zinner ve diğ. (2015), bir Alman firmasına ait bildirim takip sistemindeki açılmış olan bildirimler ile çalışma yapmışlardır. Çalışılan firmada küçük ve ortak ölçekli şehirlerde bulunan 400 şubede 20.000 çalışan günlük 1000 adete kadar bildirim girmektedir. Firmada açılan bildirimleri sınıflandırmak için metin madenciliği yöntemlerini kullanarak otomatik kategorizasyon ile ilgili çalışma yapılmıştır (Zinner ve diğ. 2015).

Altıntaş ve Tantug (2014) yaptıkları çalışmada İstanbul Teknik Üniversitesinde kullanılan Bildirim Takip Sistemindeki bildirimleri kullanmışlardır. Açılan bildirimlere otomatik olarak kategori ve alt kategori atamak için doğal dil işleme kullanarak kelimeleri çıkartmışlardır. Bu kelimelerde metin madenciliği yöntemlerini kullanıp deneysel bir çalışma yaparak bildirimlerin otomatik kategorize edilebileceğini vurgulamışlardır (Altıntaş ve Tantug 2014).

Antoniol ve diğ. (2008) BTS'lerin bir yazılımın bakım faaliyetlerini yönetmede değerli varlıklar olduğunu belirtmiştir. BTS'ler açık kaynaklı projelerde ve yazılım endüstrisinde yaygın olarak kullanılır ve arıza onarım talepleri, iyileştirmeler ve yeniden yapılanma faaliyetleri gibi pek çok farklı türde sorun toplamaktadır. Antoniol ve diğ. (2008), BTS'lerde yayınlanan sorunların metninin, düzeltici bakım ve diğer tür faaliyetlerde sınıflandırma yapmak için yeterli olup olmadığını araştırmışlardır. Alternatif karar ağaçları, Naive Bayes sınıflandırıcıları ve lojistik regresyonun bildirimleri diğer sorun türlerinden doğru bir şekilde ayırt etmek için kullanılabilirliğini göstermişlerdir. Mozilla, Eclipse ve JBoss ile ilgili konularda yapılan deneysel çalışmalardan elde edilen sonuçlar, sorunların %77 ile %82 arasında doğru bir şekilde sınıflandırılabilirliğini göstermiştir.

Komiya ve Nakamura (2019) çalışmasında insanlar kelimeleri çeşitli şekillerde anlar ve aynı kelimeler kişinin bilgisine, deneyimine ve duyarlılığına bağlı

olarak farklı şekilde yorumlanmasından bahsetmiştir. Diğer yandan, imgeleri kelimelerle ilişkilendirerek anlayış geliştirilebileceğini öne sürmüşlerdir. Kumarhanelerin yapmış olduğu kampanyalardaki kelimelerin insanlar üzerinde yapmış olduğu olumlu, olumsuz etkiyi ve tercihlerini nasıl etkilediğini metin madenciliği ve birliktelik analizi yaparak araştırmışlardır (Komiya ve Nakamura 2019).

Lee ve diğ (2016) Android İşletim Sisteminin geliştirilmesi aşamasında açılmış olan bildirimler üzerinde doğal dil işleme tekniklerini kullanarak metin madenciliği yapmışlardır. Popüler özellik isteklerinin en önemli değer olmadığını ortaya koymuşlardır. Revizyonları etkilemeye yönelik bir strateji, yeniden işleme dahil olan insani çabaların maliyetini, bu çabalara yatırımda bulunmanın geri dönüşünü, diğer rekabet önceliklerini, sorundan kaç kullanıcının etkilendiğini dikkate almaları yönünde tavsiyede bulunmuşlardır. “Hangi grup paydaşların konudan en çok etkilendiği gibi şikayetlerin sayısı düşünüldüğünde, belki de bu tür şikayetlerin diğerlerini nasıl etkilediğini düşünen bir strateji de faydalı olacaktır.” demişlerdir (Lee ve diğ. 2016).

3. UYGULAMA

Türkiye’de 2019 yılı itibari ile on teknik üniversite, bir yüksek teknoloji enstitüsü ve iki güzel sanatlar üniversitesinin yanı sıra Millî Savunma Üniversitesi ve Polis Akademisi dahil olmak üzere toplam 130 devlet üniversitesi ve 73 vakıf üniversitesi olmak üzere toplam 203 adet üniversite bulunmaktadır.

Denizli’de üniversite kurulmasından önce, 1982 yılında Denizli Eğitim Yüksekokulu ve Denizli Mühendislik Fakültesi 9 Eylül Üniversitesine bağlanmıştır. Pamukkale Üniversitesinin temel altyapısı oluşturan da bu fakülteler olmuştur. 1985 sonrası kentte üniversite kurulmasına dair oluşumlar hız kazanmış ve 1987 yılında dönemin valisi Necati Bilican’ın öncülüğünde 40 kurucu üye ile Denizli Yüksek Öğrenim Vakfı ve aynı yıl Tıp Fakültesi 9 Eylül Üniversitesi bünyesinde kurulmuştur. Denizli Yüksek Öğrenim Vakfının da büyük uğraşları sonucunda Pamukkale Üniversitesi 1992 yılında 11/7/1992 tarih 3837 sayılı Kanun ile kurulmuştur. Denizli’de üniversite, diğer Anadolu kentleriyle karşılaştırılınca nispeten geç kurulmasına rağmen, kent halkının desteğini de arkasına alarak 27 yılda hızla yapılanmış 8 yüksekokul, 16 fakülte, 14 adeti farklı ilçelerde yer alan 17 meslek yüksekokulu ve 6 enstitüye ulaşmıştır. 2019 yılı itibariyle 53296 öğrencisi ile birlikte 2120 akademik ve 1456 idari personeli ile Pamukkale Üniversitesi, Türkiye’nin önde gelen en büyük eğitim kurumlarından birisidir (Yavuzçehre 2016).

Çalışmaya konu olan yazılım PAÜ Bilgi İşlem Daire Başkanlığı profesyonel yazılım ekibi tarafından geliştirilmiş Pusula Sistemidir. Pusula bilgi sistemi üniversitenin öğrenci, akademik personel ve idari personellerine yazılım ile ilgili tüm konularda farklı modüller üzerinden hizmet vermektedir.

Pusula yazılımların birkaç yazılım projesi hariç diğer tüm projeler kendi bünyesinde istihdam edilen birimler tarafından iyileştirme ve düzenleme çalışmaları yapılmaktadır. Tüm yazılımların birleştiği ortak platformun adı Pusula Bilgi Sistemidir (Pusula). Pusula tüm sistemin çekirdeğini oluşturmaktadır. Giriş kontrollerinin yapıldığı, Yetkili sayfaların giriş esnasında düzenlendiği ve diğer gerekli işlemlerin yapıldığı merkezdir. Pusula bünyesinde 23 adet yazılım proje

barındırmasına rağmen bu projelerden 20 tanesi için Bildirim Takip Sisteminden bildirim açılmıştır. Çalışmanın bu kısmında Pusula bünyesinde bildirim açılan projelere kısaca değinilmiştir.

Öğrenci Bilgi Sistemi (OBS) : Öğrencilerin ders kayıt, not, staj ve öğrenci danışmanlık işlemlerinin yapıldığı, bu işlemlere ait belge ve raporlarının bulunduğu bilgi sistemidir. Öğrenci anketlerinin düzenlenmesi ve doldurulması, öğrenci kataloglar işlemleri de bu bilgi sisteminde yapılmaktadır. Üniversite öğrenci, idari ve akademik personelleri tarafından aktif bir şekilde kullanılan sistem en çok bildirim oluşturulduğu bilgi sistemidir.

Öğrenci İşleri Bilgi Sistemi (OBİS) : Bildirim sayısı olarak OBS'yi takip eden bu sistemde Fakülte ve Merkez Öğrenci işlerine ait olan öğrenci işlemleri, Yönetim Kurulu Kararlarının girilmesi ve güncellenmesi, harçlar, askerlik, transkript gibi işlemler yapılmaktadır. Bu kısımda ayrıca derslerin açılması, ders görevlendirmeleri, İlk kayıt işlemleri, Çift anadal ve yandal işlemleri, yatay geçiş ve nüfus, akademik takvim işlemleri de yer almaktadır.

Personel Bilgi Sistemi (PBS) : Personele ait olan atama, kıdem, terfi, izin ve görevlendirme işlemleri ve süreçleri bulunmaktadır. Bu işlemlerin yanında personelin eğitim, askerlik, disiplin ceza ve ödül, hizmet cetveli, unvan vs.. gibi bilgileri yer almaktadır. PBS' de Türkiye Cumhuriyeti Sosyal Güvenlik Kurumu (SGK) ve Doküman Yönetim Sistemi (DYS) ile web servisleri aracılığı ile her daim bağlantı içerisinde olup personel bilgi alış verişi yapılmaktadır. Bu sistemdeki bilgiler diğer projelerdeki personel işlemleri ile bütünleşik çalışmaktadır.

Eğitim Öğretim Bilgi Sistemi (EBS) : Bu sistemde Bologna süreci kapsamında müfredat için ders teklifleri, onayları ve kaldırılması, eğitim öğretim kılavuzlarının girilmesi, koşullu derslerin düzenlenmesi, Avrupa Kredi Transfer Sisteminin (AKTS) düzenlenmesi, Bilgi paketleri için program bilgilerinin düzenlenmesi yer almaktadır.

Enstitü Bilgi Sistemi (ENBS) : Enstitü işlemlerinin yapıldığı bölümdür. Burada yüksek lisans ve doktora öğrencileri için başvuru ve mülakat işlemlerinin

yanı sıra öğrencinin not ve danışmalık işlemleri yapılmaktadır. Aynı zamanda enstitüye ait her türlü rapor ve belgeler burada yer almaktadır.

Bukalemun İçerik Yönetim Sistemi (BİYS) : Pamukkale Üniversitesi idari ve akademik tüm birim ve kişilerin bilgilerini paylaşmak ve tanıtım yapmak için herhangi yazılımcı yardımı olmadan web sayfası oluşturmak ve düzenlemek için kullandıkları sistemdir. Etkinlik, haber ve duyurular burada yapılabilmektedir.

Eğitim Yönetim Sistemi (EYS) : Haftalık ders programlarının girilmesi, Ek derslere onay verilmesi, Telafi ders işlemleri, Sınav yer ve zaman işlemlerinin yapılması, sınav kağıdı okuma bilgilerini girilmesi burada yer almaktadır. Ders programlarının zaman çizelgeleri ve bunlarla ilgili raporlar bu sistemde yer almaktadır.

Erasmus Bilgi Sistemi (ERBS) : Erasmus kapsamında üniversiteler arası antlaşmalar ve özelliklerinin girildiği, bu kapsama programlardaki kontenjanlarının belirlendiği, öğrencilerin başvuru bilgileri ve sınav not işlemlerinin, anlaşmalı üniversitelere geçiş takibinin ve koordinatörlük işlemlerin tutulduğu sistemdir.

Yurt Dışı Öğrenci Bilgi Sistemi (YOBS) : Erasmus haricinde yurt dışından eğitim almak için yabancı uyruklu öğrenci sınavı(YÖS) ile gelen öğrenciler için program kontenjanları, yeterlilik sınav not takibi, not işlemleri, burs bilgilerinin düzenlendiği ve bu öğrencileri için yapılacak diğer işlemlerin olduğu sistemdir.

Hazırlık Bilgi Sistemi (HBS) : Pusulada bulunan Yabancı Dil ve İlahiyat hazırlık okuyacak olan öğrencilerin için hazırlık not ve devamsızlık işlemlerinin, muafiyetlerinin, hazırlık ders kur işlemlerinin, ders ve sınıf oluşturma işlemlerinin yapıldığı bölümdür.

Sağlık Kültür ve Spor Bilgi Sistemi (SBS) : Öğrencilerin yemek burs bilgilerini, part time zamanlı çalışma çizelgelerini barındırır. Ayrıca öğrenci toplulukları ve etkinlik bilgileri tutulmaktadır. Genel etkinlikler ve öğrenci toplulukları için salon ve kongre merkezi rezervasyon işlemleri yapılmaktadır.

Alt Yapı Bilgi Sistemi (ABS) : Üniversitenin bütün taşınmaz mülklerin tamamının bütün özellikleri ile beraber tutulduğu ve Yüksek Öğretim Mekanları

Yatırım Karar Destek Sistemi ile web servis aracılığı ile bilgi alışverişinin yapıldığı sistemdir. Bütün derslik planları burada yapılmaktadır. Taşınmazlar ile ilgili arıza ve bakım işlemleri de bu sistemde yürütülmektedir.

Tıp Fakültesi Bilgi Sistemi (TFBS) : Pamukkale Üniversitesi Tıp Fakültesinin ders ve notlandırma işlemleri diğer fakültelerden farklıdır. Bu yüzden Tıp fakültesi dersleri ve öğrencilik işlemleri için derslerin açılması, görevlendirilmesi ve not girişleri ve takibi bu sistemden yapılmaktadır.

Proje Bilgi Sistemi (PRBS) : Akademik personelin Teknoloji Transfer Ofisi(TTO), Döner Sermaye, Dış Kaynaklı proje süreçleri ve Mentör- Mentee başvuruları bu sistem sayesinde yürütülmektedir.

Eğitim Destek Sistemi (EDS) : Derslerle ilgili kaynak ve bilgilere erişim sağlayan, ödev, sınav gibi etkinlik oluşturulabilen OBS ile birlikte bütünleşik çalışan uzaktan eğitim sistemidir.

Stratejik Yönetim Bilgi Sistemi (SYBS) : Strateji Daire Başkanlığı tarafından yürütülen sistemde üniversite ve diğer birimlere ait stratejik planlarının yapılması, bu planların bütçe girişleri, gerçekleştirme oranlarının işlenmesi ve stratejik plan süreçlerinin raporları yer almaktadır.

Süreç Yönetim Sistemi (SYS) : Üniversite bünyesinde bulunan tüm süreç ve iş akışlarının tanımlanması ve düzenlenmesi, iş analizlerinin yapılması, üniversite organizasyon ve unvan şemalarının düzenlenmesi, bu işlemlere ait rapor ve belgeler bu sistemde yer almaktadır.

Formasyon Bilgi Sistemi (FBS) : Eğitim Fakültesi tarafından yürütülen sistemde Pedagojik Formasyon programlarının belirlenmesi, açılması, şartlar, kabuller ve kayıt işlemleri yapılmaktadır. Bu programlara ait ders ve şube işlemleri, görevlendirme ve öğrenci not işlemleri yer almaktadır.

Ek Ders Otomasyonu (EDO) : Aylık olarak Akademik personelin izin ve görevlendirmeleri, maaş karşılığı dersler ve resmi tatiller çıkarıldıktan sonra geriye kalan örgün ve ikinci öğretimde verilen ek derslerin hesaplanması, onay süreçleri ve

raporlanması yapılmaktadır. Bu sistem daha sonra EYS sistemi ile birleştirilerek EYS sisteminin altına alınmıştır.

Mezun Bilgi Sistem (MBS) : Pamukkale Üniversitesi'nden mezun olan öğrencilerin kariyer bilgilerini paylaşabildiği, yeni gelişmeler için anket doldurabildiği sistemdir. MBS sayesinde gerekli analizler yapılarak müfredatlar tekrardan gözden geçirilebilmektedir.

3.1 F1 Bildirim Takip Sistemi

Yazılım geliştirme ekiplerinin en büyük yardımcısı iş takip sistemleridir. PAÜ Pusula sistemi de yazılım ekiplerine yardımcı olmak amacı ile bünyesinde F1 bildirim takip sistemini bulundurmaktadır. Pusula sistemine 2008 yılında başlanmasına karşın, F1 16.05.2012 tarihinde hizmet alınmıştır. F1'de son kullanıcılar karşılaştığı problemleri veya iyileştirme isteklerini bir ara yüz sayesinde Pusula sistemine bildirmektedir.

Bildirimler bildirim tarihinden oluşturulan benzersiz bir referans numarası ile saklanmaktadır. Bu bildirimler başlık ve açıklama giriş alanları doldurularak ve bildirim konusu seçilerek yapılmaktadır. Çalışma bildirim konuları "Pusula Bilgi Sistemi İçindeki Yazılım İle İlgili Talepler" ve "Web Sitesi Talepleri" konusu ile sınırlandırılmıştır.

Şekil 3.1'te görüldüğü gibi kişiler problem ve isteklerini Başlık ve Açıklama alanlarını doldurarak ifade etmektedir. Belirlenen konularda 04.04.2019 tarihine kadar toplam 14596 adet bildirim girilmiştir. Bu tarihe kadar girilen bildirimlerin Pusula projelerine göre dağılımı Tablo 3.1'de verilmiştir.

Bildirim Tipi :

Bildirim Konusu :

E-posta Talepleri/Problemleri
Kamera Arızası
Bariyer Problemleri
Akıllı Kart/OGS Geçiş Sistemleri
Sunucu Talepleri
Yemek Bursu Tanımlama/Iptal
Misafir Kullanıcı İnternet İşlemleri
FTP Alanı Talebi
BOTT - El Terminali Problemleri
Ağ(Network)/İnternet Problemleri

Seç

Pusula Bilgi Sistemi İçindeki Yazılımlarla İle İlgili Talepler
Web Sitesi Talepleri

Seç

Başlık :

Açıklama :

Email :

Eklenti :

Şekil 3.1 Bildirim Talep Ekranı

Tablo 3.1: Projelere Göre Bildirim Sayıları

| Proje Kodu | Proje Adı | Bildirim Sayısı | Oran |
|------------|----------------------------------|-----------------|-------|
| OBS | Öğrenci Bilgi Sistemi | 3223 | 0,221 |
| OBİS | Öğrenci İşleri Bilgi Sistemi | 2363 | 0,162 |
| EYS | Eğitim Yönetim Sistemi | 1959 | 0,134 |
| PBS | Personel Bilgi Sistemi | 1698 | 0,116 |
| BİYS | Bukalemun İçerik Yönetim Sistemi | 1439 | 0,099 |
| ENBS | Enstitü Bilgi Sistemi | 1262 | 0,086 |
| EBS | Eğitim Öğretim Bilgi Sistemi | 728 | 0,050 |
| ERBS | Erasmus Bilgi Sistemi | 362 | 0,025 |
| HBS | Hazırlık Bilgi Sistemi | 362 | 0,025 |
| YOBS | Yurtdışı Öğrenci Bilgi Sistemi | 184 | 0,013 |
| F1 | F1 Bildirim Takip Sistemi | 130 | 0,009 |
| SBS | SKS Bilgi Sistemi | 123 | 0,008 |
| ABS | Altyapı Bilgi Sistemi | 123 | 0,008 |
| PUSULA | Pusula Bilgi Sistemi | 122 | 0,008 |
| TFBS | Tıp Fakültesi Bilgi Sistemi | 100 | 0,007 |
| PRBS | Proje Bilgi Sistemi | 89 | 0,006 |
| EDS | Eğitim Destek Sistemi | 82 | 0,006 |
| SYBS | Stratejik Yönetim Bilgi Sistemi | 62 | 0,004 |
| SYS | Süreç Yönetim Sistemi | 60 | 0,004 |
| FBS | Formasyon Bilgi Sistemi | 56 | 0,004 |
| EDO | Ek Ders Otomasyonu | 50 | 0,003 |
| MBS | Mezun Bilgi Sistemi | 19 | 0,001 |

Tablo 3.1'e göre bildirimlerin 3223 tanesi OBS ile ilgilidir. OBS'yi 2363 bildirim ile OBİS ve 1959 bildirim ile EBS takip etmektedir. Son sıralarda ise 19 bildirim ile MBS ve 50 bildirim ile EDS yer almaktadır.

Bildirimler PAÜ Bilgi İşlem Daire Başkanlığında görevli ve yazılım proje ekibinde bulunan alanında uzman analistler tarafından başlık ve açıklama alanlarına göre değerlendirilip ilgili projenin yazılımcısına yönlendirilmektedir. Şekil 3.2'de analistler tarafından yazılımcıya yönlendirilip sonuçlandırılan PBS'ye ait bildirim ekran görüntüsü verilmiştir. Şekil 3.2'de son kullanıcının giriş yaptığı başlık ve açıklama alanının yanı sıra kullanıcıya analistler tarafından yapılan geri dönüşte görülmektedir. Uzman analistler tarafından her bir bildirim okunması, sorgulanması ve değerlendirilmesi süreçlerinin 1 dakika sürdüğü düşünülecek olursa analiz için kullanılan tüm bildirimler için bu süre 14.596 dakika yapmaktadır. En kötü durumda bütün analistlerin her bildirim okursa dört kişi için bu süre 58.388 dakika olacaktır. En iyi durum her bir bildirim ilgili projesi ile alakalı analist okur ve gerekli atamayı yapacak olursa 14.596 dakika olacaktır. En iyi durum senaryosunda mesai saati olarak analistler bu bildirimler için 30 iş günü harcamışlardır. En kötü senaryo için ise bu 120 iş günü olmaktadır.

| Bildirim Detayı | |
|-------------------------------|---|
| Referans No | : 2019030423785 |
| Bildirim Tarihi | : 04.03.2019 09:18 |
| Bildirim Türü | : Hata |
| Bildirim Sahibi | : |
| Başlık | : Görevlendirme talebi sorunu |
| Açıklama | : Pusula bilgi sisteminde görevlendirme talebi yapmaya çalıştığımda, sistem hata veriyor ve talebi tamamlamama izin vermiyor. Acilen düzeltilmesi hususunda gereğinin yapılmasını arz ederim. |
| Bildirim Giren | : |
| Öncelik | : Düşük |
| Proje | : Personel Bilgi Sistemi |
| Planlanan Bitiş Tarihi | : 04.03.2019 |
| Onaylayan | : |
| Onaylama Tarihi | : 04.03.2019 11:47 |
| Son İşlem Tarihi | : 04.03.2019 15:29 |
| Bildirim Sonucu | : Merhaba; Görevlendirme ile ilgili talebiniz sistemde kaydedilmiş olarak görünüyor. Aldığınız ekran görüntüsü önceki görevlendime tarihi ile çakıştığı durumda görüntülenebilmektedir. İyi çalışmalar. |

Şekil 3.2: Örnek Bildirim Görüntüsü

3.1.1 Gereksiz kelimelerin atılması

Gereksiz kelimelerin atılması (stop-word removing) metin madenciliği sürecinde ön işleme adımlarından biridir. En basit yaklaşımda, kelime bir belgedeki 1 boşlukla ayrılmış herhangi bir sözcük öbeğidir. Ancak, edatlar ve bağlaçlar gibi durma sözcükleri olarak adlandırılan çok sayıda bilgilendirici olmayan kelime vardır. Bu nedenle, durma listesi genellikle belge temsil sürecinde filtrelenmesi gereken kelimelerden oluşur, çünkü kategoriler arasında ayırım potansiyeli yoktur. Durma listesine dahil edilecek olan kelimeler dile ve göreve bağlıdır. Ancak, var, ve, veya, gibi, hemen vs tüm durumlar için durma sözcükleri olarak kabul edilebilecek genel bir sözcük kümesi vardır (Silva ve Ribeiro 2003).

Bildirimlerde ayırt edici özellik içermeyen, cümleleri veya kelimeleri birbirine bağlayan kendi başına anlam ifade etmeyen kelimeler belirlenmektedir. Bu

kelimler edatlar, sıfatlar veya bağlaçlar olabilir. Gereksiz kelime kümesinden ilk 20 kelime örneği Tablo 3.2’de verilmiştir.

Tablo 3.2: Gereksiz Kelime Örnekleri

| KelimeID | Kelime |
|-----------------|---------------|
| 3 | ve |
| 28 | ile |
| 37 | için |
| 40 | gibi |
| 41 | bir |
| 48 | iyi |
| 70 | çok |
| 85 | veya |
| 90 | var |
| 98 | sonra |
| 109 | aynı |
| 119 | da |
| 142 | ama |
| 151 | merhaba |
| 171 | değil |
| 173 | lütfen |
| 174 | her |
| 177 | nin |
| 188 | sadace |

Tablo 3.2’de gösterilen örnekler gibi uzman denetimiyle 85 adet gereksiz kelime kümesi oluşturulmuştur. Bu kelimler edatlar bağlaçlar ve kesme işareti ile ayrılmış tek başına anlama ifade etmeyen ve bütün bildirimlerde kullanılan kelimelerdir.

3.1.2 Metin Dönüştürme

Metin madenciliğinin önemli adımlarından birisi de metinlerin dönüştürülmesidir (text transformation). Metin içinde ortak anlam ifade eden birbirinden benzersiz veriler aynı kelime olarak kabul edilmelidir. Çalışmanın bu kısmında Türkiye Cumhuriyeti kimlik numaraları (T.C No) gibi benzersiz olan ama aynı anlama gelen kelimelerin düzenlenmesinden bahsedilmiştir.

Daha önce açılmış bir bildirim ile alakalı yeni bir bildirim açılacak ise, bu bildirimde daha önceki bildirimde 13 haneli referans numarası başlık veya açıklama alanına eklenebilmektedir. Metin dönüşümü yapılırken 13 haneli sayısal veriler bildirim referans numarası olarak kabul edilmiş ve “refno” olarak isimlendirilmiştir.

Bildirimlerde hata oluşan kişi veya öğrencilerden bahsedilirken tekillik oluşturduğu 11 haneli Türkiye Cumhuriyeti Kimlik numarası “tcNo”, yüksek öğrenim durumundaki öğrenciler için 9 haneli öğrenci numarası “yüksekÖgrNo” ve lisans durumundaki öğrenciler için 8 haneli öğrenci numarası “lisansÖgrNo” olarak adlandırılmıştır.

Bazı bildirimlerde ise çeşitli tiplerde ayrıntılı tarihler de girilmiştir. “dd.MM.yyyy”, “dd.MM.yy” veya “dd/MM/yyyy” vs. formatlarda girilmiş olan tarih bilgileri “ayrıntıTarih” kelimesi ile eşleştirilmiştir. Bildirim içerisinde eğitim öğretim döneminden bahsedilirken “YYYY-YYYY”, “YYYY-YY” veya “YYYY/YYYY” vs. gibi formatlarda kullanımları mevcuttur. Bunlar “dönemNumara” kelimesi ile isimlendirilmiştir. Rezervasyon işlemlerinde saat kullanımı mevcuttur. “hh.mm” veya “hh:mm” şeklinde olan metinlere “saatNumara” kelimesi verilmiştir.

Bildirimler içerisinde e-posta adresleri ve linkler de bulunabilmektedir. E-postalar “eposta” kelimesine linkler ise “webUrl” kelimesine dönüştürülmüştür. Öğrenci ve Enstitü işlemlerini kapsayan yazılımların bildirimlerinde ise üç veya dört harften ve üç rakamdan oluşan ders kodlarına “dersKod” kelimesi verilmiştir. Özellikle ek derslerle ilgili yazılımlarda çok geçen kelime grubu “ek ders” ’tir. Bu kelime grubu “ek” ve “ders” olarak ayrılmaktadır. “Ders” kelimesi diğer yazılımların bildirimlerinde yoğun bir şekilde kullanıldığından ve “ek” kelimesi genel bir kelime olduğu için “ek ders” kelime grubunun ayırt edici bir özelliği olması gerekli görülmüştür. Pek çok bildirimlerde rastlanılan “Yönetim kurulu kararı” kelime grubu kısaltmaya gidilerek yerine açıklama ve başlık kısımlarında “ykk” kelimesi ile kullanılmıştır. Türkçede “ykk” kısaltması kullanılmamaktadır. Çalışmada “ykk” geçen kelimelerde kökünü “ykk” olarak kabul edilmiştir.

Bütün bu dönüşümler yapıldıktan sonra geriye kalan ve tekillik oluşturan sayısal işlemler ve “-”, “’”, “,”, “,”, “(”, “)” gibi karakterler metinlerden temizlenmiştir. Yapılmış olan bütün dönüşümler Tablo 3.3’ de verilmiştir.

Tablo 3.3: Dönüşüm Tablosu

| Örnek veri | Kelime |
|-------------------------------------|--------------|
| 2019030423785 | refNo |
| 12345678910 | tcNo |
| 123456789 | yüksekÖgrNo |
| 12345678 | lisanÖgrNo |
| 12.04.2018 12/04/18 | ayrıntıTarih |
| 2011-2012 2010/11 2009 –Bahar | dönemNumara |
| 13:10 14.15 | saatNumara |
| malik@pau.edu.tr | eposta |
| http://pusula.pau.edu.tr | webUrl |
| Ek derslerde Ek dersi | ekders |
| Ykklarda Ykk Ykksı | ykk |
| 12,5,596 “_” ’ ‘ “,” (“,”),” ’ “ | |

3.1.3 Kelime Köklerinin Çıkarılması

Netbeans IDE’si Java tabanlı proje ve uygulama geliştirmeye yarayan bir framework’tür. Zemberek kütüphanesi Java tabanlı bir kütüphanedir. En uygun uygulama yazılımının da Java uygulaması olacağına karar verilmiştir. Metin dönüşümündeki formatlara karar verildikten sonra bu düzenlemeleri gerçekleştirecek hem de Zemberek yardımı ile kelime köklerini çıkaracak (stemming) uygulama yazılmıştır. Bu uygulamada önce başlık ve açıklama metinlerine gerekli dönüşümler yapılmıştır. Daha sonra bu metinler paragraflardan cümlelere dönüştürülmüştür. Bu cümlelerin Türkçe kelime analizi yapılmıştır.

```
TurkishMorphology morphology = TurkishMorphology.createWithDefaults();  
morphology.analyzeSentence(sentence);
```

Bu analiz sonucunda kelime kökleri çıkarılmaktadır. Fakat en kök kelimeler çıkarılmaktadır. Mesela “kalemi” kelimesi “kale” köküne, “m” yapım eki ve “i” hal eki olarak ayrılmaktadır. Cümlede geçen kelimenin en iyi kökü alması amaçlanmaktadır. İlk analiz sonucunda kelimeler bellidir. Analiz edilen kelimeler

cümle ile beraber belirgenleştirme işlemine sokulup tekrar analiz edilmesi sağlanmaktadır.

```
SentenceAnalysis result = morphology.disambiguate(sentence, analysis);  
    SingleAnalysis anlys : result.bestAnalysis();
```

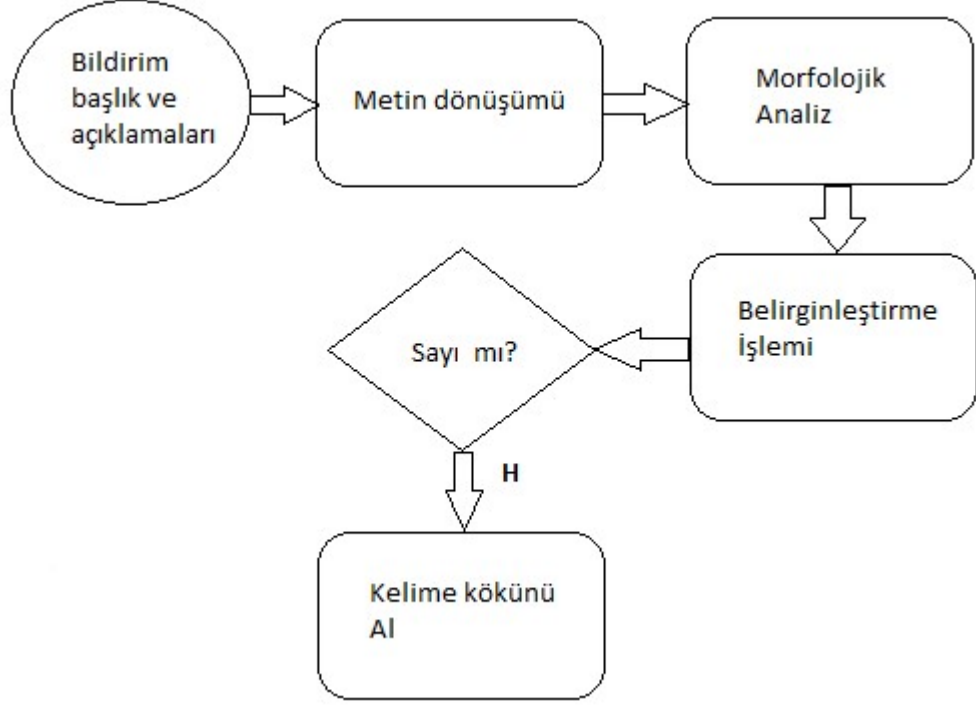
Analizi tamamlanan kelimeler Tokenizer kütüphanesi ile kelime tipi belirlenir.

```
Token token : tokenizer.tokenize(anlys.getDictionaryItem().lemma);
```

Tokenizer kütüphanesi değişik tiplerdeki kelimeleri tanımlamaya yardımcı olmaktadır. Bu tipler sözcük (word), bilinmeyen (unknown), hem karakter hemde rakam içeren (alphanumerical), rakam (number), tarih (date), zaman (time), Roma rakamı (RomanNumeral)... vs. gibi şeylerdir. Bilinmeyen kelimeler genellikle yazım yanlışlığından kaynaklanan veya farklı dilde yazılan kelimelerdir. Eğer kelime tipi sözcük (Word), kısaltma (abbreviation), hem rakam hem de karakter içeren (alphanumerical) veya bilinmeyen (unknown) ise veri tabanına kaydedilmektedir. Yazılımın bir kod örneği Tablo 3.4'te verilmiştir. Şekil 3.3'te ise yazılım akış diyagramı yer almaktadır. Yazılım çalıştıktan sonra çıkan kelimeleri tekil bir şekilde "lu_kelime" adında veri tabanında tabloya kaydedilmektedir. Bu tabloda en çok kullanılan 20 kelimenin örneği Tablo 3.5'te verilmiştir.

Tablo 3.4: Yazılım Kod Örneği

```
String senten = klm.RenameWords(rs.getString("girisAciklama"));  
List<String> sentences = extractor.fromParagraph(senten);  
for (String sentence : sentences) {  
    List<WordAnalysis> analysis = morphology.analyzeSentence(sentence);  
    SentenceAnalysis result = morphology.disambiguate(sentence, analysis);  
    for (SingleAnalysis anlys : result.bestAnalysis()) {  
        TurkishTokenizer tokenizer = TurkishTokenizer.ALL;  
        if (anlys.getStem().length() <= 40) {  
            for (Token token : tokenizer.tokenize(anlys.getDictionaryItem().lemma)) {  
                if (token.getType() == TurkishLexer.Word || token.getType() ==  
TurkishLexer.WordAlphanumerical || token.getType() ==  
TurkishLexer.WordWithSymbol ||  
token.getType() == TurkishLexer.UnknownWord || token.getType() ==  
TurkishLexer.Abbreviation) {  
                    db.InsertWord(anlys.isUnknown() ? anlys.getStem() :  
anlys.getDictionaryItem().normalizedLemma());  
                    db.InsertBildirimKelime(anlys.isUnknown() ?  
anlys.getStem() : anlys.getDictionaryItem().normalizedLemma(), bildirimId);  
                } } } } }
```



Şekil 3.3: Yazılım Akış Diyagramı

Tablo 3.5: Kelime Tablosu

| ID | Kelime | Toplam Kullanım |
|-----|-------------|-----------------|
| 43 | olmak | 14391 |
| 33 | öğrenci | 14029 |
| 56 | ders | 13433 |
| 36 | yapmak | 8450 |
| 30 | girmek | 7537 |
| 159 | dersKod | 6947 |
| 23 | etmek | 6478 |
| 84 | sistem | 5933 |
| 29 | not | 5163 |
| 265 | bilgi | 4989 |
| 31 | lisansÖgrNo | 4846 |
| 35 | dönem | 4815 |
| 185 | gerekmek | 4054 |
| 7 | sayfa | 3939 |
| 63 | vermek | 3775 |
| 83 | tarih | 3754 |
| 49 | çalışmak | 3639 |
| 45 | program | 3378 |
| 158 | almak | 3347 |
| 8 | bölüm | 3271 |

Kelimeler veri tabanına kaydettikten sonra bildirimlerin ID'leri ile bildirim açıklama alanında geçen kelimelerin ID'leri "tx_bildirimKelime" adında bir tabloya, bildirim başlığında geçen kelimeleri ise "tx_kelimeBildirimBaslik" adında bir başka bir tabloya kaydedilmektedir. Tablo 3.6'da girilmiş olan bir bildirim örneği verilmektedir.

Tablo 3.6: Bildirim Örneği

| BildirimID | Başlık | Açıklama |
|------------|--------------|--|
| 24056 | staj notu | 12345678 ÖğrenciAd ÖğrenciSoyad ÖĞRENCİNİN STAJ NOTLARI GİRİLDİ ANCAK 2.STAJ NOTU GÖRÜNÜYOR. 1. STAJ NOTU GÖRÜNÜMÜYOR. |

Bu bildirim açıklama alanı kelimelere ayrıldığında Tablo 3.7'deki gibi olmaktadır. Bildirim başlık alanının tablosunun bir örneği de Tablo 3.8'de verilmiştir.

Tablo 3.7: Açıklama Kelime Tablosu

| BildirimID | KelimeID | Kelime |
|------------|----------|--------------|
| 24056 | 31 | lisansÖgrNo |
| 24056 | 73 | ÖğrenciAd |
| 24056 | 74 | ÖğrenciSoyad |
| 24056 | 33 | öğrenci |
| 24056 | 72 | staj |
| 24056 | 29 | not |
| 24056 | 30 | girmek |
| 24056 | 75 | ancak |
| 24056 | 72 | staj |
| 24056 | 29 | not |
| 24056 | 76 | görünmek |
| 24056 | 72 | staj |
| 24056 | 29 | not |
| 24056 | 76 | görünmek |

Tablo 3.8: Başlık Kelime Tablosu

| BildirimID | KelimeID | Kelime |
|------------|----------|--------|
| 24056 | 72 | staj |
| 24056 | 29 | not |

Bütün kelimeler çıkarıldıktan sonra oluşan tablomuzda hala tam anlamlı olmayan kelimeler kalmaktadır. Bu kelimeler Türkçe kurallarına uymayan

kelimelerdir. Örneğin “tesekür” kelimesi Türkçe kurallarına uymamaktadır. Doğal dil işleme programının ilk etapta bunun kökünü bulması imkânsızdır. Bu yüzden Zemberek kütüphanesinin TurkishSpellChecker objesini kullanarak bu türdeki kelimelere suggestForWord(word) fonksiyonu ile öneri kelime vermesi sağlanmıştır. Örnekler Tablo 3.9 ‘da verilmiştir.

Tablo 3.9: Kelime Öneri Tablosu

| Kelime | Öneri |
|-------------------|--------------|
| başladığımda | Başlamak |
| örneğin | Örnek |
| çalışıyordum | Çalışmak |
| karsılasıyorum | Karşılaşmak |
| fakültesi | Fakülte |
| detaytranskriptte | transkript |
| öğrencisinin | Öğrenci |
| gönderirmenizin | Göndermek |
| bulunmasını | Bulmak |
| düşünmekteyim | Düşünmek |

Öneri sunulmuş olan kelimeler kontrol edilerek yanlış öneri sunulan kelimeler düzeltilmektedir. Bu düzeltmeler kontrol edilirken öneri sunulmamış bazı kelimeler içinde öneri eklenmektedir. Daha sonra öneri sunulmuş olan bütün kelimeler yeni kök kelimelerine dönüştürülmüştür.

3.2 Birliktelik Analizinin Uygulanması

Birliktelik veri madenciliğinde kullanılan analizlerden biridir. Birliktelik analizinde kullanılan iki adet değer yer almaktadır. Bunlardan biri Destek değeridir. Destek değeri; bir yazılım projesinin bildirimlerinde proje ile kelime ilişkisinin hangi oranda tekrarlandığını belirtir. Destek değerinin genel fonksiyonu eşitlik (1)’de gösterilmiştir.

$$Destek(X \rightarrow Y) = \frac{Frekans\left(\frac{x}{y}\right)}{N} \quad (1)$$

Çalışmamıza bu fonksiyon uygulandığında eşitlik (2) oluşmaktadır.

$$Destek(Kelime \rightarrow Proje) = \frac{Frekans(Kelime,Proje)}{Toplam\ Bildirim\ Sayısı} \quad (2)$$

Bu fonksiyona göre bir yazılım projesinde kullanılan kelimelerin her bir projeye göre destek değerleri çıkartılabilmektedir. Güven değeri; bir kelimenin bir yazılım projesinde kullanılma olasılığını ortaya çıkartır. Güven değeri genel fonksiyonu eşitlik (3) ‘teki gibidir.

$$Güven(X \rightarrow Y) = Frekans(X, Y) / Frekans(Y) \quad (3)$$

Çalışmada kullanılan fonksiyon eşitlik (4) gösterilmiştir.

$$Güven(Kelime \rightarrow Proje) = Frekans(Kelime, Proje) / Frekans(Kelime) \quad (4)$$

Eşitlik (4) kullanılarak bütün kelimelerin yazılım projelerinde açıklama alanında ve başlık alanında güven değerleri bulunmuştur. “Dil” kelimesi ile ilgili açıklama alanı için güven değeri Tablo 3.10’da, başlık için güven değeri Tablo 3.11’de örnek gösterimi verilmiştir.

Tablo 3.10: Örnek “Dil” Kelimesi İçin Açıklama Alanında Güven Değer Tablosu

| Proje | Kelimenin Projede Kullanılma Sıklığı | Güven Değeri |
|----------------------------------|--------------------------------------|--------------|
| Hazırlık Bilgi Sistemi | 42 | 0,046 |
| Eğitim Öğretim Bilgi Sistemi | 70 | 0,076 |
| Erasmus Bilgi Sistemi | 178 | 0,193 |
| Öğrenci İşleri Bilgi Sistemi | 128 | 0,139 |
| Eğitim Yönetim Sistemi | 84 | 0,091 |
| Tıp Fakültesi Bilgi Sistemi | 9 | 0,010 |
| Personel Bilgi Sistemi | 58 | 0,063 |
| Altyapı Bilgi Sistemi | 3 | 0,003 |
| Öğrenci Bilgi Sistemi | 166 | 0,180 |
| Eğitim Destek Sistemi | 3 | 0,003 |
| Bukalemun İçerik Yönetim Sistemi | 80 | 0,087 |
| Ek Ders Otomasyonu | 6 | 0,007 |
| Enstitü Bilgi Sistemi | 85 | 0,092 |
| Formasyon Bilgi Sistemi | 1 | 0,001 |
| Yurtdışı Öğrenci Bilgi Sistemi | 10 | 0,011 |
| TOPLAM | 923 | |

Tablo 3.11: Örnek “Dil” Kelimesi İçin Başlık Alanında Güven Değer Tablosu

| Proje | Kelimenin Projede Kullanılma Sıklığı | Güven Değeri |
|----------------------------------|--------------------------------------|--------------|
| Öğrenci Bilgi Sistemi | 28 | 0,173 |
| Enstitü Bilgi Sistemi | 12 | 0,074 |
| Hazırlık Bilgi Sistemi | 12 | 0,074 |
| Eğitim Öğretim Bilgi Sistemi | 2 | 0,012 |
| Erasmus Bilgi Sistemi | 58 | 0,358 |
| Personel Bilgi Sistemi | 20 | 0,124 |
| Eğitim Yönetim Sistemi | 1 | 0,006 |
| Bukalemun İçerik Yönetim Sistemi | 16 | 0,099 |
| Yurtdışı Öğrenci Bilgi Sistemi | 2 | 0,012 |
| Öğrenci İşleri Bilgi Sistemi | 11 | 0,068 |
| TOPLAM | 162 | |

Güven aralığı bulunduktan sonra kelimelere ağırlık verilmesi gerekmektedir. Kelimelere ağırlık vermek için eşitlik (5)’teki formül kullanılmaktadır.

$$w_i = tf_i \times \log \frac{D}{df_i} \quad (5)$$

Eşitlik (5)’e göre kelimenin ağırlığı proje içerisinde geçme sayısı ile doğru orantılı, bütün projelerde geçme sıklığı ile ters orantılıdır (İlhan ve diğ. 2008). Eşitlik (5)’de tf_i bir kelimenin bir projede geçme sıklığını, D toplam bildirim sayısını, df_i kelimenin kaç adet bildirimde geçtiği sayı olarak belirlenmektedir. Buna göre 14596 adet bildirimden 439 tanesinde kullanılan “dil” kelimesinin ağırlık tablosu Tablo 3.12’de verilmiştir.

Tablo 3.12: Örnek “Dil” Kelimesi için Kelime Ağırlığı

| Proje | Kelimenin Projede Geçme Sayısı | Güven Değeri | Ağırlık | Normalize [0-1] |
|----------------------------------|--------------------------------|--------------|---------|-----------------|
| Hazırlık Bilgi Sistemi | 42 | 0,046 | 63,893 | 0,236 |
| Eğitim Öğretim Bilgi Sistemi | 70 | 0,076 | 106,488 | 0,393 |
| Erasmus Bilgi Sistemi | 178 | 0,193 | 270,785 | 1,000 |
| Öğrenci İşleri Bilgi Sistemi | 128 | 0,139 | 194,722 | 0,719 |
| Eğitim Yönetim Sistemi | 84 | 0,091 | 127,786 | 0,472 |
| Tıp Fakültesi Bilgi Sistemi | 9 | 0,010 | 13,691 | 0,051 |
| Personel Bilgi Sistemi | 58 | 0,063 | 88,233 | 0,326 |
| Altyapı Bilgi Sistemi | 3 | 0,003 | 4,564 | 0,017 |
| Öğrenci Bilgi Sistemi | 166 | 0,180 | 252,530 | 0,933 |
| Eğitim Destek Sistemi | 3 | 0,003 | 4,564 | 0,017 |
| Bukalemun İçerik Yönetim Sistemi | 80 | 0,087 | 121,701 | 0,449 |
| Ek Ders Otomasyonu | 6 | 0,007 | 9,128 | 0,034 |
| Enstitü Bilgi Sistemi | 85 | 0,092 | 129,307 | 0,478 |
| Formasyon Bilgi Sistemi | 1 | 0,001 | 1,521 | 0,006 |
| Yurtdışı Öğrenci Bilgi Sistemi | 10 | 0,011 | 15,213 | 0,056 |
| TOPLAM | 923 | | | |

Tablo 3.12 incelendiğinde eşitlik (4) ile hesaplanan güven değerleri ve eşitlik (5) ile hesaplanan ağırlık değerleri görülmektedir. Güven değerleri ve ağırlıklar [0-1] arasında normalize edildiğinde aynı olduğu görülmüştür. Çalışmada kelime ağırlıkları olarak güven değerleri kullanılmıştır.

3.3 Birliktelik Analizinin Kişilere Uygulanması

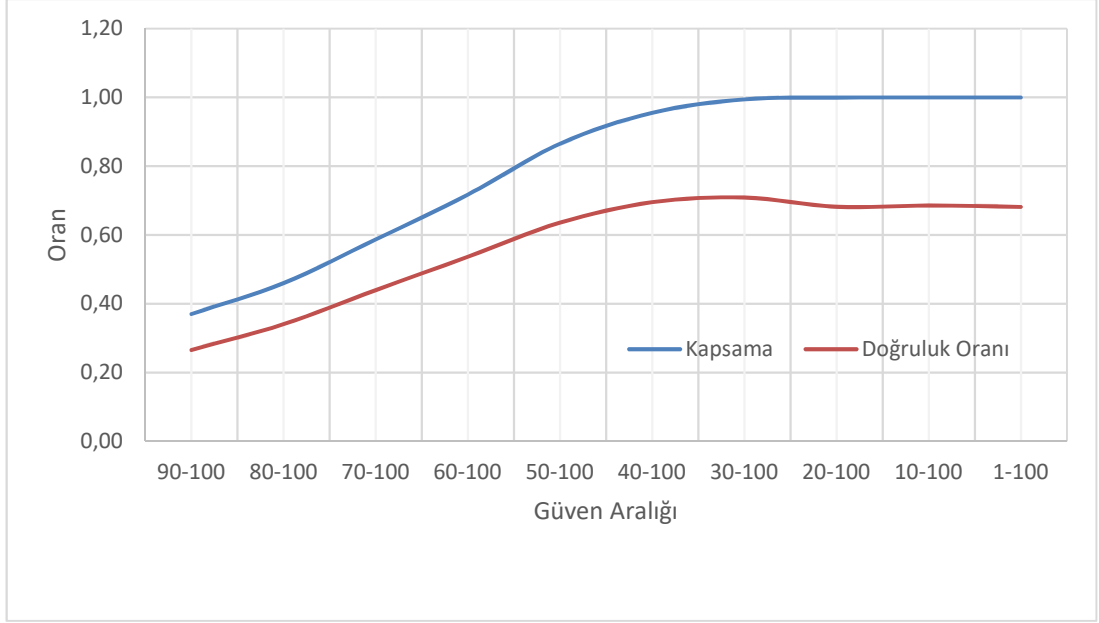
Pamukkale Üniversitesi Pusula sistemindeki yazılımlar birimlere göre değişiklik göstermektedir. Öğrenci İşleri Daire Başkanlığında genellikle öğrencilerle ilgili yazılımlar için bildirim açılmaktadır. Personel Daire Başkanlığında ise personel ile ilgili yazılımlar için bildirim açılmaktadır. Bu yüzden bildirim açan kişi de açıklama ve başlık kadar değer kazanmaktadır. Bir kişinin her proje için birliktelik analizi yapılarak elde edilen güven değeri Tablo 3.13'te verilmiştir.

Tablo 3.13: Örnek Bir Kullanıcıya Ait Güven Değeri

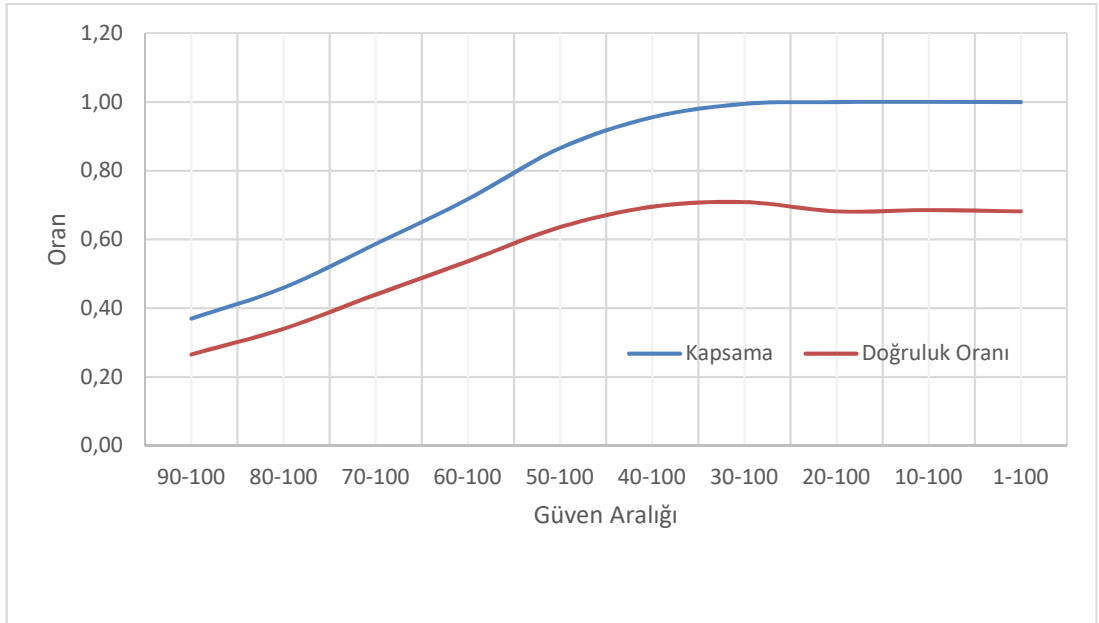
| Proje | Açılan Bildirim Sayısı | Güven Değeri |
|---------------------------|-------------------------------|---------------------|
| Enstitü Bilgi Sistemi | 1 | 0,029 |
| Personel Bilgi Sistemi | 16 | 0,471 |
| F1 Bildirim Takip Sistemi | 11 | 0,324 |
| Pusula Bilgi Sistemi | 1 | 0,029 |
| Süreç Yönetim Sistemi | 5 | 0,147 |
| TOPLAM | 34 | |

3.4 Tahmin İçin Uygun Değerlerin Belirlenmesi

Bir bildirim içerdiği kelimelerinden tahmin edileceği durumda, güven düzeyi en yüksek kelimeleri tercih etmenin projeyi belirlemede etkin olacağı düşünülebilir. Sadece yüksek güven değerine sahip kelimeler kullanıldığında beraberinde bu kelimelerin kapsadığı bildirim sayılarının azalması gibi bir problem ile karşılaşmaktadır. Bu nedenle güven düzeyini bütün bildirimleri kapsayacak şekilde belirlenmelidir. Güven düzeyi en yüksekten (%90) başlayarak en düşüğe doğru (%1) azaltılmış ve her bir güven düzeyine göre doğru tahmin oranları ve bildirim kapsama oranları Şekil 3.4 ve Şekil 3.5'te verilmiştir.



Şekil 3.4:Uygun Güven Düzeyinin Belirlenmesi (Gereksiz Kelimeler ile)



Şekil 3.5: Uygun Güven Düzeyinin Belirlenmesi (Gereksiz Kelimeler Hariç)

Şekil 3.4 ve 3.5'teki grafikten görüleceği gibi %30 -%100 güven değeri aralığında tüm bildirimleri kapsamakta ve doğruluk oranı en üst seviyededir. Güven değer aralığı arttırıldığında kapsama alanı değişmemektedir, fakat doğruluk oranı azalmaktadır. Bu oranın azalmasının nedeni sık kullanılan kelimelerin sonuca az miktarda olsa da olumsuz bir etkisinin bulunmasıdır. Şekil 3.4 ve 3.5'e bakıldığında gereksiz kelimelerin sonuca etki etmediği görülmektedir. Çünkü gereksiz kelimeler

her bir bildirimde kullanıldığından projelerdeki ağırlıkları düşük çıkmaktadır ve etkisi az olmaktadır. “Ve” gereksiz kelimesine ait güven değer tablosu Tablo 3.14’te verilmiştir.

Tablo 3.14: ‘Ve’ Kelimesi Güven Değeri

| Proje | Kullanılma Sayısı | Güven Değeri |
|----------------------------------|-------------------|--------------|
| Öğrenci Bilgi Sistemi | 2939 | 0,245 |
| Öğrenci İşleri Bilgi Sistemi | 1972 | 0,164 |
| Eğitim Yönetim Sistemi | 1709 | 0,142 |
| Enstitü Bilgi Sistemi | 1108 | 0,092 |
| Bukalemun İçerik Yönetim Sistemi | 1084 | 0,090 |
| Personel Bilgi Sistemi | 1031 | 0,086 |
| Eğitim Öğretim Bilgi Sistemi | 930 | 0,077 |
| Erasmus Bilgi Sistemi | 231 | 0,019 |
| Hazırlık Bilgi Sistemi | 212 | 0,018 |
| Süreç Yönetim Sistemi | 138 | 0,012 |
| Yurtdışı Öğrenci Bilgi Sistemi | 98 | 0,008 |
| SKS Bilgi Sistemi | 83 | 0,007 |
| Eğitim Destek Sistemi | 77 | 0,006 |
| F1 Bildirim Takip Sistemi | 69 | 0,006 |
| Altyapı Bilgi Sistemi | 69 | 0,006 |
| Pusula Bilgi Sistemi | 52 | 0,004 |
| Proje Bilgi Sistemi | 51 | 0,004 |
| Tıp Fakültesi Bilgi Sistemi | 50 | 0,004 |
| Ek Ders Otomasyonu | 32 | 0,003 |
| Stratejik Yönetim Bilgi Sistemi | 27 | 0,002 |
| Formasyon Bilgi Sistemi | 25 | 0,002 |
| Mezun Bilgi Sistemi | 21 | 0,002 |
| TOPLAM | 12008 | |

Tablo 3.14’ten görüleceği gibi gereksiz kelimelerin güven değerleri çok düşüktür ve en yüksek değeri de 2939 adet kullanımıyla OBS’de görülmektedir. 1972 adet kullanımıyla OBİS, 1709 adet ile EYS takip etmektedir. Tablo 3.14’e bakıldığında gereksiz kelimelerin ancak %20 ile % 1 aralığında dikkate alınmaktadır. Şekil 3.3 ve 3.4’e bakıldığında ise gereksiz kelimelerin oran üzerinde olumlu veya olumsuz bir etkisi olmadığı görülmektedir.

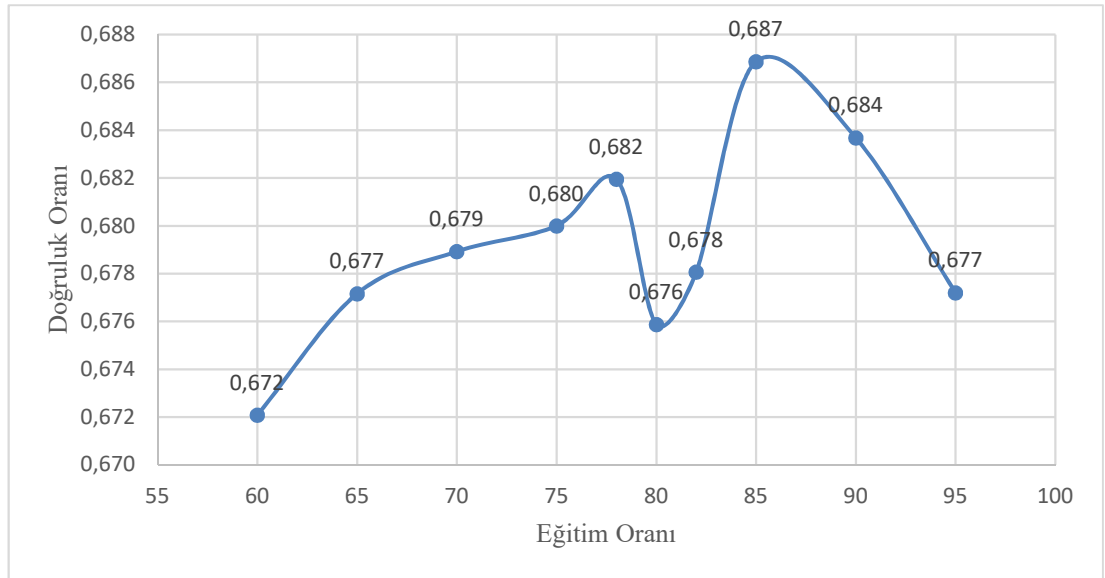
Eğitim için kullanılacak olan veri seçimi de büyük önem arz etmektedir. Eğitim için seçilecek olan veri kümesinin ne kadar çok olursa doğru sonuca yaklaşıacağı düşünülebilir. Eğitim kümesi seçilirken her bir küme elemanları

değiştirilerek 10 kez tekrarlanmıştır. Sonuçlar ise bu tekrarlarını ortalaması olarak alınmıştır. Bu değerler Tablo 3.15'te gösterilmiştir.

Tablo 3.15: Eğitim Oranları

| Deney | Eğitim Oranları | | | | | | | | | |
|-----------------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | %60 | %65 | %70 | %75 | %78 | %80 | %82 | %85 | %90 | %95 |
| 1 | 0,672 | 0,681 | 0,670 | 0,679 | 0,675 | 0,670 | 0,681 | 0,692 | 0,665 | 0,679 |
| 2 | 0,667 | 0,682 | 0,690 | 0,698 | 0,694 | 0,669 | 0,686 | 0,697 | 0,694 | 0,665 |
| 3 | 0,679 | 0,676 | 0,673 | 0,678 | 0,677 | 0,675 | 0,679 | 0,679 | 0,682 | 0,679 |
| 4 | 0,667 | 0,680 | 0,687 | 0,679 | 0,681 | 0,670 | 0,680 | 0,694 | 0,706 | 0,683 |
| 5 | 0,672 | 0,663 | 0,681 | 0,672 | 0,677 | 0,679 | 0,677 | 0,679 | 0,697 | 0,689 |
| 6 | 0,666 | 0,674 | 0,672 | 0,681 | 0,674 | 0,675 | 0,680 | 0,697 | 0,696 | 0,676 |
| 7 | 0,670 | 0,675 | 0,666 | 0,682 | 0,692 | 0,681 | 0,655 | 0,683 | 0,675 | 0,668 |
| 8 | 0,681 | 0,677 | 0,688 | 0,668 | 0,692 | 0,671 | 0,677 | 0,669 | 0,678 | 0,671 |
| 9 | 0,668 | 0,683 | 0,680 | 0,685 | 0,671 | 0,676 | 0,683 | 0,686 | 0,678 | 0,696 |
| 10 | 0,678 | 0,681 | 0,681 | 0,678 | 0,686 | 0,694 | 0,683 | 0,693 | 0,665 | 0,667 |
| Ortalama | | | | | | | | | | |
| | 0,672 | 0,677 | 0,679 | 0,680 | 0,682 | 0,676 | 0,678 | 0,687 | 0,684 | 0,677 |

Tablo 3.15'i incelediğimizde eğitim kümesi belirlediğimiz aralıklar içerisinde en fazla fark %0,022 olmuştur. Eğitim kümesini bu oranlardan hangisini alırsak alalım sonucumuzda çok anlamlı rakamsal farklılıklar oluşmayacaktır. Yani belirlemiş olduğumuz metodoloji kararlı ve güvenilirdir denilebilmektedir. Şekil 3.6'da Eğitim kümesinin doğruluk oranına etkisini gösteren bir grafik verilmiştir.



Şekil 3.6: Eğitim Doğruluk Oranı

Şekil 3.6'te gösterildiği gibi eğitim kümesi veri sayısı artırıldıkça doğruluk oranı artmaktadır. Fakat belli bir düzeyden sonra eğitime ayrılan veriler artsa da doğruluk oranı düşmektedir. %85 eğitim kümesi ile bütün kelimelerin tamamen öğrenildiği ortaya çıkmaktadır. Çalışmada en iyi proje tahminlemesinin veri setinin %85'i eğitim için,%15'i de tahmin için kullanıldığında elde edildiği görülmüştür.

3.5 Bildirime Ait Projenin Tahmini

Çalışmamızda belirlemiş olduğumuz %85 eğitim kümesini kullanarak tüm kelimelerin güven değerleri dolayısıyla ağırlıkları bulunmuştur. Bildirimi açan kişiler içinde ağırlıklar hesaplanmıştır. Proje tahminlemesi bildirim başlık ve açıklama alanındaki kelimelerin ağırlıklarının toplamı her proje için hesaplanmaktadır.

Kişilere ait her proje için ağırlıkları bulunduktan sonra bulunan bu değerler [0-1] arasında normalize edilmektedir. Normalize edilmiş bu değerlerin toplamı projeye ait değeri vermektedir. Değeri en yüksek olan proje tahmin edilen proje olmaktadır. Öğrenci Bilgi Sistemine ait olan 24056 ID'li bildirim için örnek hesaplama yapılmıştır. Tablo 3.6 verilen örneğin hesaplanması Tablo 3.16'da gösterilmiştir.

Tablo 3.16: Bildirimin Projelere Göre Ağırlıkları

| Proje | Açıklama | Başlık | Kişi | Toplam |
|----------------------------------|----------|--------|-------|--------|
| Öğrenci Bilgi Sistemi | 1,000 | 1,000 | 0,053 | 2,053 |
| Öğrenci İşleri Bilgi Sistemi | 0,500 | 0,190 | 1,000 | 1,691 |
| Enstitü Bilgi Sistemi | 0,108 | 0,075 | 0,000 | 0,183 |
| Eğitim Yönetim Sistemi | 0,051 | 0,019 | 0,000 | 0,070 |
| Eğitim Öğretim Bilgi Sistemi | 0,035 | 0,032 | 0,000 | 0,067 |
| Hazırlık Bilgi Sistemi | 0,038 | 0,021 | 0,000 | 0,059 |
| Erasmus Bilgi Sistemi | 0,025 | 0,013 | 0,000 | 0,039 |
| Tıp Fakültesi Bilgi Sistemi | 0,014 | 0,009 | 0,000 | 0,022 |
| Personel Bilgi Sistemi | 0,021 | 0,001 | 0,000 | 0,022 |
| Bukalemun İçerik Yönetim Sistemi | 0,010 | 0,003 | 0,000 | 0,013 |
| Yurtdışı Öğrenci Bilgi Sistemi | 0,009 | 0,001 | 0,000 | 0,010 |
| Formasyon Bilgi Sistemi | 0,007 | 0,002 | 0,000 | 0,009 |
| Mezun Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| Pusula Bilgi Sistemi | 0,002 | 0,000 | 0,000 | 0,000 |
| Süreç Yönetim Sistemi | 0,002 | 0,000 | 0,000 | 0,000 |
| Ek Ders Otomasyonu | 0,001 | 0,000 | 0,000 | 0,000 |
| Altyapı Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| Proje Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| SKS Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| F1 Bildirim Takip Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| Stratejik Yönetim Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |
| Eğitim Destek Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |

Tablo 3.16’da gösterilen normalize edilmiş değerlere bakıldığında bildirim açan kişi en çok Öğrenci İşleri Bilgi Sistemi’nde bildirim açmıştır. Fakat açıklama ve başlık alanındaki kelimelerin ağırlıkları Öğrenci Bilgi Sistemine ait olduğunu göstermektedir. Açıklama alanın kelimeleri Öğrenci İşleri Bilgi Sisteminden iki kat daha yüksek bir oranla Öğrenci Bilgi Sistemini işaret etmektedir. Başlığın kelimeleri ise ikinci en yakın projeye beş kat daha yüksek bir oranla yine Öğrenci Bilgi Sistemini göstermektedir. Toplamlar doğrultusunda birinci sırada Öğrenci Bilgi Sistemi, ikinci sırada Öğrenci İşleri Bilgi Sistemi, çok zayıf oranlarla üçüncü sırada Enstitü Bilgi Sistemi gelmektedir. Dördüncü ve beşinci sırada sırasıyla Eğitim Yönetim Sistemi ve Eğitim Öğretim Bilgi Sistemi bulunmaktadır.

Tablo 3.17’de Personel Bilgi Sistemi için açılmış olan bir bildirim örneği bulunmaktadır. Tablo 3.18’de bu bildirim başlık kelimeleri, Tablo 3.19’da ise bildirim açıklama alanının kelimeleri bulunmaktadır.

Tablo 3.17: PBS Bildirim Örneği

| BildirimID | Başlık | Açıklama |
|-------------------|-----------------------|---|
| 68 | SGK öğrenim işlemleri | Öğrenim bilgilerinin Hitap programına göre uyarlanması gerekmektedir. |

Tablo 3.18: PBS Bildirimi Başlık Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 68 | 2997 | sgk | 1 |
| 68 | 145 | öğrenim | 1 |
| 68 | 122 | işlem | 1 |

Tablo 3.19: PBS Bildirimi Açıklama Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 68 | 145 | öğrenim | 1 |
| 68 | 265 | bilgi | 1 |
| 68 | 513 | hitap | 1 |
| 68 | 45 | program | 1 |
| 68 | 532 | göre | 1 |
| 68 | 7199 | uyarlamak | 1 |
| 68 | 185 | gerekmek | 1 |

Tablo 3.18 ve 3.19'daki çıkan kelimelerin ve bildirim açan kişinin ağırlıkları neticesinde Tablo 3.20'deki normalize edilmiş değerler ortaya çıkmaktadır. Tablo 3.20 değerlendirildiğinde açıklama ve başlık kelimelerinin Personel Bilgi Sistem'i için ağırlığının yüksek olduğu görülmektedir. Bildirimi açan kişinin de Personel Bilgi Sistemi'nde daha çok bildirim açtığı belirtilmektedir. Dolayısıyla Personel Bilgi Sistemi tahmini büyük farkla birinci olarak çıkmaktadır. İkinci sırada beş kat daha düşük oranla Erasmus Bilgi Sistemi, üçüncü sırada altı kat daha düşük oranla Öğrenci Bilgi Sistemi yer almıştır. Dördüncü sırada Öğrenci İşleri Bilgi Sistemi, beşinci sırada Eğitim Yönetim Sistemi yer almışlardır.

Tablo 3.20: PBS Bildirim Proje Tahmini

| Proje | Açıklama | Başlık | Kişi | Toplam |
|----------------------------------|----------|--------|-------|--------|
| Personel Bilgi Sistemi | 1,000 | 1,000 | 1,000 | 3,000 |
| Erasmus Bilgi Sistemi | 0,104 | 0,006 | 0,518 | 0,628 |
| Öğrenci Bilgi Sistemi | 0,174 | 0,294 | 0,048 | 0,516 |
| Enstitü Bilgi Sistemi | 0,366 | 0,047 | 0,012 | 0,425 |
| Öğrenci İşleri Bilgi Sistemi | 0,228 | 0,106 | 0,000 | 0,334 |
| Eğitim Yönetim Sistemi | 0,259 | 0,052 | 0,000 | 0,310 |
| Eğitim Öğretim Bilgi Sistemi | 0,179 | 0,091 | 0,000 | 0,270 |
| Yurtdışı Öğrenci Bilgi Sistemi | 0,004 | 0,003 | 0,181 | 0,188 |
| Bukalemun İçerik Yönetim Sistemi | 0,054 | 0,005 | 0,024 | 0,083 |
| Hazırlık Bilgi Sistemi | 0,040 | 0,011 | 0,000 | 0,051 |
| Pusula Bilgi Sistemi | 0,000 | 0,002 | 0,024 | 0,026 |
| Ek Ders Otomasyonu | 0,013 | 0,002 | 0,000 | 0,015 |
| Tıp Fakültesi Bilgi Sistemi | 0,005 | 0,003 | 0,000 | 0,009 |
| Altyapı Bilgi Sistemi | 0,002 | 0,006 | 0,000 | 0,008 |
| Proje Bilgi Sistemi | 0,004 | 0,001 | 0,000 | 0,005 |
| Stratejik Yönetim Bilgi Sistemi | 0,001 | 0,002 | 0,000 | 0,003 |
| SKS Bilgi Sistemi | 0,000 | 0,001 | 0,000 | 0,002 |
| Süreç Yönetim Sistemi | 0,001 | 0,001 | 0,000 | 0,002 |
| Eğitim Destek Sistemi | 0,003 | 0,000 | 0,000 | 0,000 |
| Formasyon Bilgi Sistemi | 0,001 | 0,000 | 0,000 | 0,000 |

Tablo 3.21’de Enstitü Bilgi Sistemi için açılmış bir bildirim bulunmaktadır. Tablo 3.22’de bu bildirim başlık kelimeleri, Tablo 3.23’te ise açıklama kelimeleri bulunmaktadır. Tablo 3.24’te bildirim sonuçları görülmektedir.

Tablo 3.21: ENBS Bildirim Örneği

| BildirimID | Başlık | Açıklama |
|------------|-----------------------|--|
| 188 | Enstitü Bilgi Sistemi | Enstitü Bilgi Sistemi'ne yeni eklenen sayfaların eğitiminin verilmesi ve bundan sonraki yol haritası hakkında bilgilendirme yapılması. |

Tablo 3.22: ENBS Bildirimi Başlık Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 188 | 205 | enstitü | 1 |
| 188 | 265 | bilgi | 1 |
| 188 | 84 | sistem | 1 |

Tablo 3.23: ENBS Bildirimi Açıklama Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 188 | 205 | enstitü | 1 |
| 188 | 265 | bilgi | 2 |
| 188 | 84 | sistem | 1 |
| 188 | 1017 | ne | 1 |
| 188 | 219 | yeni | 1 |
| 188 | 435 | eklemek | 1 |
| 188 | 7 | sayfa | 1 |
| 188 | 26 | eğitim | 1 |
| 188 | 63 | vermek | 1 |
| 188 | 3 | ve | 1 |
| 188 | 284 | bu | 1 |
| 188 | 98 | sonra | 1 |
| 188 | 183 | yol | 1 |
| 188 | 10210 | harita | 1 |
| 188 | 136 | hak | 1 |
| 188 | 36 | yapmak | 1 |

Tablo 3.24: ENBS Bildirimi Proje Tahmini

| Proje | Açıklama | Başlık | Kişi | Toplam |
|----------------------------------|-----------------|---------------|-------------|---------------|
| Enstitü Bilgi Sistemi | 0,668 | 1,000 | 0,243 | 1,911 |
| Eğitim Yönetim Sistemi | 0,264 | 0,101 | 1,000 | 1,365 |
| Öğrenci Bilgi Sistemi | 0,173 | 0,228 | 0,910 | 1,311 |
| Bukalemun İçerik Yönetim Sistemi | 1,000 | 0,061 | 0,009 | 1,070 |
| Hazırlık Bilgi Sistemi | 0,038 | 0,068 | 0,441 | 0,547 |
| Öğrenci İşleri Bilgi Sistemi | 0,103 | 0,158 | 0,234 | 0,495 |
| Personel Bilgi Sistemi | 0,079 | 0,176 | 0,225 | 0,480 |
| Proje Bilgi Sistemi | 0,005 | 0,029 | 0,324 | 0,358 |
| Eğitim Öğretim Bilgi Sistemi | 0,110 | 0,120 | 0,108 | 0,338 |
| Altyapı Bilgi Sistemi | 0,003 | 0,085 | 0,225 | 0,313 |
| Süreç Yönetim Sistemi | 0,012 | 0,043 | 0,171 | 0,226 |
| Pusula Bilgi Sistemi | 0,010 | 0,044 | 0,162 | 0,215 |
| Stratejik Yönetim Bilgi Sistemi | 0,007 | 0,062 | 0,081 | 0,150 |
| F1 Bildirim Takip Sistemi | 0,005 | 0,025 | 0,108 | 0,138 |
| SKS Bilgi Sistemi | 0,007 | 0,022 | 0,108 | 0,137 |
| Tıp Fakültesi Bilgi Sistemi | 0,003 | 0,015 | 0,117 | 0,135 |
| Ek Ders Otomasyonu | 0,010 | 0,002 | 0,081 | 0,094 |
| Erasmus Bilgi Sistemi | 0,036 | 0,054 | 0,000 | 0,090 |
| Mezun Bilgi Sistemi | 0,002 | 0,026 | 0,045 | 0,073 |
| Yurtdışı Öğrenci Bilgi Sistemi | 0,011 | 0,020 | 0,000 | 0,031 |
| Eğitim Destek Sistemi | 0,003 | 0,012 | 0,000 | 0,015 |
| Formasyon Bilgi Sistemi | 0,003 | 0,003 | 0,000 | 0,006 |

Tablo 3.24 incelendiğinde bildirim açıklama kelimelerinin Bukalemun İçerik Yönetim Sistemine, başlığın kelimelerinin Enstitü Bilgi Sistemine yakın, bildirim açan kişinin ise en çok Eğitim Yönetim Sisteminde bildirim açtığı görülmektedir. Fakat genel toplamda ise Enstitü Bilgi Sistemi'nin projesi olduğu ve doğru tahminlendiği görülmüştür. Açıklama kelimeleri ikinci en yüksek olarak Enstitü Bilgi Sistemini, Başlık kelimeleri ikinci en yüksek oran Öğrenci Bilgi Sistemi, Kişi oranı ise ikinci en yüksek oran Öğrenci Bilgi Sistemine olmuştur. Toplamlara bakıldığında birinci sırada Enstitü Bilgi Sistemi, ikinci Eğitim Yönetim

Sistemi, üçüncü olarak ta Öğrenci Bilgi Sistemi yer almıştır. Dördüncü sırada Bukalemun İçerik Yönetim Sistemi, beşinci sırada düşük bir oranla Hazırlık Bilgi Sistemi vardır.

Tablo 3.25'te Eğitim Yönetim Sistemi'nde açılmış olan bir bildirim örneği gösterilmektedir. Tablo 3.26 ve 3.27'de bu bildirim için kelimeler verilmiştir. Tablo 3.28'de Eğitim Yönetim Sistemi bildirimlerinin ağırlık sonuçları görünmektedir.

Tablo 3.25: EYS Bildirimi Örneği

| BildirimID | Başlık | Açıklama |
|-------------------|--|--|
| 3415 | Ders bitimindeki danışmanlıkların iptali | Dersler 5 ocakta bitiyor. Tatillerde ders danışmanlıkları ödenmiyor, sadece öğrenci tezde ise uzm.alan dersi, tez danışmanlığı ödeniyor. Ders danışmanlıklarının otomatikmen ücrete yansımaması için iptali gerekiyor. |

Tablo 3.26: EYS Bildirimi Başlık Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 3415 | 56 | ders | 1 |
| 3415 | 3334 | bitim | 1 |
| 3415 | 155 | danışman | 1 |
| 3415 | 341 | iptal | 1 |

Tablo 3.27: EYS Bildirimi Açıklama Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 3415 | 56 | ders | 3 |
| 3415 | 1088 | ocak | 1 |
| 3415 | 628 | bitmek | 1 |
| 3415 | 995 | tatil | 1 |
| 3415 | 155 | danışman | 1 |
| 3415 | 943 | ödemek | 2 |
| 3415 | 211 | sadece | 1 |
| 3415 | 33 | öğrenci | 1 |
| 3415 | 213 | tez | 1 |
| 3415 | 838 | ise | 1 |
| 3415 | 770 | uzman | 1 |
| 3415 | 99 | otomatik | 1 |
| 3415 | 576 | ücret | 1 |
| 3415 | 966 | yansımak | 1 |
| 3415 | 37 | için | 1 |
| 3415 | 341 | iptal | 1 |
| 3415 | 185 | gerekmek | 1 |

Tablo 3.28: EYS Bildirimi Proje Tahmini

| Proje | Açıklama | Başlık | Kişi | Toplam |
|----------------------------------|-----------------|---------------|-------------|---------------|
| Eğitim Yönetim Sistemi | 1,000 | 0,418 | 1,000 | 2,418 |
| Personel Bilgi Sistemi | 0,249 | 1,000 | 0,000 | 1,249 |
| Enstitü Bilgi Sistemi | 0,457 | 0,551 | 0,000 | 1,008 |
| Öğrenci İşleri Bilgi Sistemi | 0,352 | 0,631 | 0,000 | 0,983 |
| Öğrenci Bilgi Sistemi | 0,366 | 0,525 | 0,000 | 0,890 |
| Eğitim Öğretim Bilgi Sistemi | 0,091 | 0,089 | 0,000 | 0,180 |
| Bukalemun İçerik Yönetim Sistemi | 0,052 | 0,112 | 0,000 | 0,163 |
| Ek Ders Otomasyonu | 0,035 | 0,026 | 0,037 | 0,098 |
| SKS Bilgi Sistemi | 0,022 | 0,052 | 0,000 | 0,074 |
| Hazırlık Bilgi Sistemi | 0,024 | 0,027 | 0,000 | 0,052 |
| Altyapı Bilgi Sistemi | 0,000 | 0,001 | 0,037 | 0,039 |
| Erasmus Bilgi Sistemi | 0,013 | 0,017 | 0,000 | 0,030 |
| Tıp Fakültesi Bilgi Sistemi | 0,011 | 0,009 | 0,000 | 0,020 |
| Pusula Bilgi Sistemi | 0,007 | 0,006 | 0,000 | 0,013 |
| Formasyon Bilgi Sistemi | 0,007 | 0,004 | 0,000 | 0,011 |
| Eğitim Destek Sistemi | 0,007 | 0,003 | 0,000 | 0,010 |
| Mezun Bilgi Sistemi | 0,000 | 0,000 | 0,000 | 0,000 |
| Proje Bilgi Sistemi | 0,017 | 0,000 | 0,000 | 0,000 |
| F1 Bildirim Takip Sistemi | 0,009 | 0,000 | 0,000 | 0,000 |
| Yurtdışı Öğrenci Bilgi Sistemi | 0,005 | 0,000 | 0,000 | 0,000 |
| Süreç Yönetim Sistemi | 0,003 | 0,000 | 0,000 | 0,000 |
| Stratejik Yönetim Bilgi Sistemi | 0,002 | 0,000 | 0,000 | 0,000 |

Tablo 3.28 'e bakıldığında açıklama alanındaki kelimelerin ağırlığının Eğitim Yönetim Sisteminde, başlığın Personel Bilgi Sistemi yönünde olduğu görülmektedir. Bildirim açan kişinin ise yine Eğitim Yönetim Sisteminde daha çok bildirim açtığı tablodan okunmaktadır. Toplam ağırlıktan da yine bildirim Eğitim Yönetim Sistemi projesi olarak doğru tahmin edilmiştir. Açıklama kelimelerinin ikinci en yakın projesi yarı oranla Enstitü Bilgi Sistemi, Başlık kelimeleri yarıdan bir daha büyük bir oranla ikincisi Öğrenci İşleri Bilgi Sistemi olmuştur. Toplamda ise ikinci sırada

Personel Bilgi Sistemi, üçüncü olarak Enstitü Bilgi Sistemi, dördüncü olarak Öğrenci İşleri Bilgi Sistemi, beşinci olarak ta Öğrenci Bilgi Sistemi yer almıştır.

Tablo 3.29’da Ek Ders Otomasyonu için girilmiş olan bildirim bilgileri yer almaktadır. Tablo 3.30 ve 3.31’de bu bildirim için olan başlık ve açıklama alanlarının kelimeleri verilmektedir.

Tablo 3.29: EDO Bildirimi Örneği

| BildirimID | Başlık | Açıklama |
|------------|---|---|
| 2381 | Sisteme girilen ders hocanın ücret formunda görülmemektedir . | Sosyal Bilgiler Eğitimi Anabilim Dalına ait bahar yarıyılındaki tüm derslerinin hepsini sisteme girmeme rağmen Doç. Dr. AD SOYAD girdiği NÖ Sosyal proje geliştirme dersi hocanın ücret formunda gözükmemektedir. hocanın diğer dersleri görülmektedir. bu durumun acil olarak düzeltilmesi istenmektedir. iyi çalışmalar |

Tablo 3.30: EDO Bildirimi Başlık Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|------------|----------|--------|--------|
| 2381 | 84 | sistem | 1 |
| 2381 | 30 | girmek | 1 |
| 2381 | 56 | ders | 1 |
| 2381 | 209 | hoca | 1 |
| 2381 | 576 | ücret | 1 |
| 2381 | 577 | form | 1 |
| 2381 | 78 | görmek | 1 |

Tablo 3.31: EDO Bildirimi Açıklama Kelimeleri

| BildirimID | KelimeID | Kelime | Sayısı |
|-------------------|-----------------|---------------|---------------|
| 2381 | 947 | sosyal | 1 |
| 2381 | 265 | bilgi | 1 |
| 2381 | 26 | eğitim | 1 |
| 2381 | 156 | anabilim | 1 |
| 2381 | 157 | dal | 1 |
| 2381 | 393 | ait | 1 |
| 2381 | 326 | bahar | 1 |
| 2381 | 685 | yarıyıl | 1 |
| 2381 | 886 | tüm | 1 |
| 2381 | 56 | ders | 3 |
| 2381 | 1714 | hepsi | 1 |
| 2381 | 84 | sistem | 1 |
| 2381 | 30 | girmek | 2 |
| 2381 | 88 | rağmen | 1 |
| 2381 | 404 | doç | 1 |
| 2381 | 11 | dr | 1 |
| 2381 | 2301 | AD | 1 |
| 2381 | 3547 | SOYAD | 1 |
| 2381 | 1340 | nö | 1 |
| 2381 | 947 | sosyal | 1 |
| 2381 | 783 | proje | 1 |
| 2381 | 220 | gelişmek | 1 |
| 2381 | 209 | hoca | 2 |
| 2381 | 576 | ücret | 1 |
| 2381 | 577 | form | 1 |
| 2381 | 147 | gözükmek | 1 |
| 2381 | 208 | diğer | 1 |
| 2381 | 78 | görmek | 1 |
| 2381 | 284 | bu | 1 |
| 2381 | 143 | durum | 1 |
| 2381 | 1342 | acil | 1 |
| 2381 | 43 | olmak | 1 |
| 2381 | 92 | düzelme | 1 |
| 2381 | 38 | istemek | 1 |
| 2381 | 48 | iyi | 1 |
| 2381 | 49 | çalışmak | 1 |

Tablo3.32’de Ek Ders Otomasyonu bildirim tahmin sonuçları yer almaktadır. Tablo 3.32 incelendiğinde açıklama ve başlık kelimelerinin ağırlığının Eğitim Yönetim Sistemine yönelik olduğu görülmektedir. Fakat bildirim açan kişi ise sadece Ek Ders Otomasyonuna bildirim açmıştır. Sadece kişi ağırlığının yeterli olmadığı

görülmüştür. Bu yüzden proje tahmini Eğitim Yönetim Sistemi olarak yanlış tahminleme yapılmıştır. Açıklama kelimelerinin ikinci en yüksek oranı Öğrenci Bilgi Sistemi, Başlık kelimelerinin ikinci en yüksek oranı ise yine Öğrenci Bilgi Sistemi olmuştur. Toplamda ise birinci sırada Eğitim Yönetim Sistemi, ikinci sırada Öğrenci Bilgi Sistemi, üçüncü sırada ise birinci sırada çıkması beklenen Ek Ders Otomasyonu yer almıştır. Dördüncü ve beşinci sıralarda daha düşük oranlarla sırasıyla Öğrenci İşleri Bilgi Sistemi ve Personel Bilgi Sistemi vardır. Daha sonra yapılabilecek çalışmalarda bir takım ön koşullandırma yapılarak veya açıklama, başlık alanlarının toplama etki oranları değiştirilerek daha iyi tahmin yüzdeleri çıkartılabilir.

Tablo 3.32: EDO Bildirimi Proje Tahmini

| Proje | Açıklama | Başlık | Kişi | Toplam |
|----------------------------------|----------|--------|-------|--------|
| Eğitim Yönetim Sistemi | 1,000 | 1,000 | 0,000 | 2,000 |
| Öğrenci Bilgi Sistemi | 0,648 | 0,761 | 0,000 | 1,410 |
| Ek Ders Otomasyonu | 0,040 | 0,113 | 1,000 | 1,153 |
| Öğrenci İşleri Bilgi Sistemi | 0,454 | 0,259 | 0,000 | 0,713 |
| Enstitü Bilgi Sistemi | 0,265 | 0,380 | 0,000 | 0,645 |
| Personel Bilgi Sistemi | 0,168 | 0,229 | 0,000 | 0,397 |
| Eğitim Öğretim Bilgi Sistemi | 0,225 | 0,110 | 0,000 | 0,335 |
| Bukalemun İçerik Yönetim Sistemi | 0,172 | 0,115 | 0,000 | 0,286 |
| Hazırlık Bilgi Sistemi | 0,058 | 0,169 | 0,000 | 0,227 |
| Proje Bilgi Sistemi | 0,089 | 0,048 | 0,000 | 0,136 |
| Erasmus Bilgi Sistemi | 0,026 | 0,056 | 0,000 | 0,082 |
| F1 Bildirim Takip Sistemi | 0,010 | 0,042 | 0,000 | 0,052 |
| Pusula Bilgi Sistemi | 0,018 | 0,028 | 0,000 | 0,046 |
| Altyapı Bilgi Sistemi | 0,005 | 0,038 | 0,000 | 0,043 |
| Süreç Yönetim Sistemi | 0,011 | 0,026 | 0,000 | 0,036 |
| SKS Bilgi Sistemi | 0,009 | 0,018 | 0,000 | 0,027 |
| Tıp Fakültesi Bilgi Sistemi | 0,012 | 0,013 | 0,000 | 0,025 |
| Stratejik Yönetim Bilgi Sistemi | 0,004 | 0,020 | 0,000 | 0,024 |
| Eğitim Destek Sistemi | 0,007 | 0,009 | 0,000 | 0,016 |
| Yurtdışı Öğrenci Bilgi Sistemi | 0,005 | 0,008 | 0,000 | 0,014 |
| Mezun Bilgi Sistemi | 0,001 | 0,011 | 0,000 | 0,012 |
| Formasyon Bilgi Sistemi | 0,004 | 0,006 | 0,000 | 0,010 |

Tahmin için ayrılan tüm bildirimler için bu işlemler gerçekleştirilmiştir. Belirlediğimiz oranlarda eğitim kümesindeki bildirimler her defasında değiştirilerek 10 kez tekrarlanmıştır. Sonuçlar Tablo 3.33'te gösterilmiştir.

Tablo 3.33: Bildirim Tahmin Oranı

| | Eğitim Kümesi (%)85 – Güven Aralığı (%) 30-100 (Gereksiz Kelimeler ile) | Eğitim Kümesi (%)85 – Güven Aralığı (%) 30-100 (Gereksiz Kelimeler Hariç) |
|---|--|--|
| 1 | 0,706 | 0,713 |
| 2 | 0,724 | 0,703 |
| 3 | 0,711 | 0,714 |
| 4 | 0,716 | 0,704 |
| 5 | 0,729 | 0,723 |
| 6 | 0,715 | 0,699 |
| 7 | 0,693 | 0,712 |
| 8 | 0,715 | 0,700 |
| 9 | 0,720 | 0,727 |
| 10 | 0,714 | 0,706 |
| Ortalama | 0,714 | 0,710 |
| Eğitim İçin Kullanılan Bildirim Sayısı | 2205 | 2205 |

Tablo 3.33’te görüldüğü gibi tahminleme için ayrılan bildirim sayısı 2205’tir. 14.596 adet bildirimden 12.391 tanesi ise eğitim için kullanılmıştır. Bu bildirimlerden gereksiz kelimeler ile tahmin edilen en iyi oran %72,42 olurken en kötü oran ise %69,34 olmuştur. Gereksiz kelimeler çıkarıldıktan sonra en iyi oran %72,33 en kötü oran %69,88 olmuştur. Bu sonuçların ışığında gereksiz kelimelerin çıkarılması veya çıkarılmaması arasında çok küçük fark var bunun sebebi eğitim kümesinin her ölçümde değişmesi neticesinde olmaktadır. Çünkü gereksiz kelimelerin hesaba alınabilmesi için güven değerini %20 altında bir seviyeden başlatmak gerekmektedir. Bu yüzden seçilen güven aralığında (%30 - %100) gereksiz kelimeler gelmemektedir.

4. SONUÇ ve KARŞILAŞTIRMA

Bu çalışmada Pamukkale Üniversitesi Bilgi İşlem Daire Başkanlığı tarafından geliştirilen entegre bilgi yönetim sistemine, kullanıcılar ve geliştiriciler tarafından girilen yeni talep, öneri ve hata bildirimlerinin metin madenciliği ile değerlendirilmesi hedeflenmiştir. Öğrenci işleri ve personel işleri olmak üzere birçok projenin entegre çalıştığı Pusula Bilgi Sistemine yapılan web tabanlı bildirim talepleri F1 Bildirim Takip Sistemi aracılığıyla alınmaktadır. F1 Bildirim Takip Sistemine test kullanıcısı ve son kullanıcılar tarafından bildirim talepleri girilmektedir. Bildirimler sisteme bildirim sahibinin bilgileriyle kaydedilmektedir. Diğer taraftan bildirim zamanı, ek belgeleri, iç mesajlaşması gibi bilgilere ulaşılabilir.

Sisteme girilen bildirim talepleri başlık ve açıklama alanlarına göre proje yöneticileri/yazılım analistleri tarafından değerlendirilerek Pusula Bilgi Sistemi üzerindeki bir projeye ve bir ve/veya daha fazla kategoriyle ilişkilendirilerek işleme alınmaktadır. Sistemdeki projeler için uzmanlaşmış personele bildirim ataması da bu aşamada gerçekleştirilmektedir. Bildirimler aciliyet durumuna göre değerlendirilerek iş planına dâhil edilmektedir. Yapılan işlemler neticesinde bildirim test edilmekte ve yapılan işlemler sonuç kısmına yazılarak bildirim kapatılmaktadır.

Bu tez çalışması kapsamında F1 Bildirim Takip Sistemindeki veriler kullanılmıştır. Sistemde kullanılan iş yükünü ve zamanı azaltmak için bildirimlerin ilgili olduğu yazılım projesinin otomatik olarak tahmin edilebilmesi amaçlanmıştır. Çünkü bildirimlerin projelerinin belirlenip ilgili yazılımcıya aktarılması zaman ve ciddi bir iş yükü getirmektedir. Bu görev için harcanan zamanın, özellikle yoğun çalışma dönemlerinde, öncelikli işlere yönlendirilmesi personel performansının ön çıkarılması açısından önem arz etmektedir.

Tez çalışması kapsamında, öncelikle gelen her bir bildirim için veri setinin düzenlenmesi yapılmıştır. Bu kapsamda gereksiz kelimeler listesine eklenen kelimeler bildirim açıklamasından atılmış, metin dönüştürme ve kelime kök çıkarma işlemleri yapılmıştır. Bu aşamada Zemberek kütüphanesi kullanılarak kelime köklerine ulaşılmıştır. Daha sonra yazılımcılar veya kullanıcılar tarafından girilmiş

olan tüm bildirimlerin ön işlemden geçirilmiş içeriğinden kelime güven değerleri çıkarılmıştır. Veri madenciliği analiz türü olan birliktelik analizi kullanılarak destek değerleri ve güven değerleri hesaplanmıştır. Kelime ağırlıkları ve güven değerleri hesaplamasında normalizasyon işlemi uygulanmış olup değerleri eşit çıktığı için güven değeri kelime ağırlığı olarak alınmıştır. Aynı şekilde bildirim sahipleri için projelere göre kişilerin ağırlıkları güven değeri olarak alınmıştır.

Eğitim için kullanılacak veri setinin seçimi yapıldıktan sonra eğitim kümesinin doğruluk oranına etkisi araştırılmıştır. Eğitim kümesi veri sayısı arttıkça doğruluk oranının arttığı gözlemlenmiştir. Optimum eğitim kümesi oranı %85, tahmin kümesi oranı ise %15 olarak elde edilmiştir. Bir bildirim projelere göre ağırlıkları açıklama, başlık ve kişiye göre normalize edilmiş ve hangi projeye ait olduğu tahmin edilmiştir. Eğitim kümesi ve tahmin kümesi oranına göre bütün bildirimler analiz edildiğinde yaklaşık %72 gibi bir doğru tahminleme yapıldığı görülmektedir.

Metin madenciliği kullanarak yaptığımız bu çalışma bildirimde kullanılan kelimeler ile bildirim projesi arasında yüksek bağlantı olduğu tespit edilmiştir. Kökler çıkarıldıktan sonra farklı yöntemler ile daha iyi sonuçların alınabileceğini bu çalışma göstermektedir.

Çalışma sırasında; ilk anlamlı kelimelerin çıkartılmasından kaynaklı tahminleme yüzdesinin düşük olması, bildirimdeki kelimelerin tam Türkçe dil kurallarına uygun olmaması, sayısal ve alfa numerik terimlerin bildirim başlık ve açıklamasından çıkarılması ve imla kurallarını ihlal eden cümlelerin tam olarak analiz edilememesi gibi sorunlar ile karşılaşmıştır. Anlamsız ve Türkçe dil kurallarına uymayan kelimelere alternatif öneri sunulması ve benzersiz yapıda olan sayısal ve alfa numerik terimler metin dönüştürme yapılarak çözüm üretilmiştir.

Bu çalışmanın devamı olarak farklı bildirim takip sistemlerinde metin olarak girilen bildirimlerin projesi, kategorisi ve atanacak uzmanlaşmış personel tahmin edilebilir. Çünkü bildirim takip sistemleri özelleştirilebilmektedir. Tek bir projeye ait olan farklı kategorideki bildirimler içinde bu çalışma yapılabilir. Çalışmada lineer bir yaklaşım ve Zemberek kütüphanesi ile birliktelik analizi yapıldığından sonraki çalışmalarda Word2Vec kütüphanesi yardımıyla vektörel yaklaşım ile daha iyi

sonular alınabilir. Word2vec, kelimelerin dşük boyutlu vektör temsillerini ıkarmak için yaygın olarak kullanılan bir algoritmadır (Ji ve diğ. 2019).

5. KAYNAKLAR

Ackoff, R.L., “From data to wisdom”, *Journal of Applied Systems Analysis*, 16(1), 3-9, (1989).

Aguinis, H., Forcum, L.E., and Joo, H. “Using market basket analysis in management research”, *Journal of Management*, 39(7), 1799-1824, (2013).

Akın, A. A., and Akın, M.D. “Zemberek, an open source NLP framework for Turkic languages”, *Structure*, 10, 1-5, (2007).

Almario, C.V., Keller, M.S., Chen, M., Lasch, K., Ursos, L., Shklovskaya, J., Melmed G.Y. and Spiegel, B.M. “Optimizing selection of biologics in inflammatory bowel disease: development of an online patient decision aid using conjoint analysis”, *The American Journal of Gastroenterology*, 113(1), 58, (2018).

Alparslan, E. and Bahsi, H. “Security level classification of confidential documents written in Turkish”, *Proceedings of the International Conference on User Centric Media*, Springer, Berlin, Heidelberg, (2009).

Altintas, M., and Tantug, A.C. “Machine learning based ticket classification in issue tracking systems”, *Proceedings of the International Conference on Artificial Intelligence and Computer Science (AICS)*, (2014).

Antoniol, G., Ayari, K., Di Penta, M., Khomh, F., and Guéhéneuc, Y.G., “Is it a bug or an enhancement?: a text-based approach to classify change requests”, *Proceedings of the 18th Annual International Conference on Computer Science and Software Engineering*, 8, 304-318, (2008).

Aşlıyan, R., Korhan G., ve Filiz A. “Türkçe otomatik heceleme sistemi ve hece istatistikleri”, (2006).

Bacon, J., “The art of community: Building the new age of participation”, *O'Reilly Media, Inc.*, (2012).

- Baysal, O, Reid Holmes, and Godfrey, M.W. “Situational awareness: personalizing issue tracking systems”, *Proceedings of the 2013 International Conference on Software Engineering*, 1185-1188, (2013).
- Bellinger, G., Castro D., and Mills, A., “Data, information, knowledge, and wisdom”, <http://www.systems-thinking.org/dikw/dikw.htm>, (2004).
- Berry, M. J., and Linoff, G., “Data mining techniques: for marketing, sales, and customer support”, *John Wiley & Sons, Inc.* (1997).
- Bertram, D., Volda, A., Greenberg, S., and Walker, R., “Communication, collaboration, and bugs: the social nature of issue tracking in small, collocated teams”, *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 291-300, (2010).
- Bettenburg, N., Premraj, R., Zimmermann, T., and Kim, S., “Extracting structural information from bug reports”, *Proceedings of the ACM International Working Conference on Mining Software Repositories*, 27-30, (2008).
- Bird, S., Ewan K., and Edward L., “Natural language processing with Python: analyzing text with the natural language toolkit”, *O'Reilly Media Inc.*, (2009).
- By, A., Tokar, S., and Markus, M., “Precipitation-runoff modeling using artificial neural networks and conceptual models”, *Journal of Hydrologic Engineering*, 4, 156-161, (2000).
- Chen, Y. L., Tang, K., Shen, R. J., and Hu, Y. H., “Market basket analysis in a multiple store environment”, *Decision Support Systems*, 40(2), 339-354, (2005).
- Cunningham, Sally Jo, and Eibe Frank, “Market basket analysis of library circulation data”, *Proceedings of the IEEE 6th International Conference on Neural Information Processing*, 2, 825-830, (1999).
- Dall'Olio, G.M., Bertranpetit, J., and Laayouni, H. “The annotation and the usage of scientific databases could be improved with public issue tracker software”, *Database: The Journal of Biological Databases and Curation*, 2010, 1-6, (2010).
- De Bruijn, B. and Martin, J. “Getting to the (c)ore of knowledge: mining biomedical literature”, *International Journal of Medical Informatics*, 67(1-3), 7-18, (2002).

- Demir, N., and Yılmaz, E., “Türkçe ses bilgisi”. *Anadolu Üniversitesi Yayınları*, No.2362, (2011).
- Diao, Y., Jamjoom, H., and Loewenstern, D. “Rule-based problem classification in it service management”, *Proceedings of the IEEE International Conference on Cloud Computing*, 221-228, (2009).
- Dursunoğlu, H., “Türkiye Türkçesinde konuşma dili ile yazı dili arasındaki ilişki”, *Atatürk Üniversitesi Türkiyat Araştırmaları Enstitüsü Dergisi*, 30, 1-21, (2006).
- Erdem, M. “Türkçede çekim ve yapım eklerinin özellikleri ve sınırları”, *Bilig*, 58, 71-90, (2011).
- Erhardt, R.A.A., Schneider, R. and Blaschke, C. “Status of text-mining techniques applied to biomedical text”, *Drug Discovery Today*, 11(7-8), 315-325, (2006).
- Feldman, Ronen, and James Sanger, “The text mining handbook: advanced approaches in analyzing unstructured data”, *Cambridge University Press*, (2007).
- Fogel, K. “Producing open source software: How to run a successful free software Project”, *O'Reilly Media, Inc.*, (2005).
- Futrell, R.T., Schafer, L.I., and Donald F., “Quality software project management”, *Prentice Hall PTR*, Upper Saddle River, NJ, (2001).
- Gemci, F., and Peker, K.A. “Extracting Turkish tweet topics using LDA”, *Proceedings of the IEEE 8th International Conference on Electrical and Electronics Engineering (ELECO)*, 531-534, (2013).
- Han, J., Kamber, M., and Pei, J. “Data mining: concepts and techniques”, 3rd edition, Morgan Kaufmann, (2011).
- He, Wu, Zha, S. and Li, L. “Social media competitive analysis and text mining: a case study in the pizza industry”, *International Journal of Information Management*, 33(3), 464-472, (2013).
- Hearst, M., “What is text mining”, *SIMS, UC Berkeley*, (2003).

Horiuchi, Y., Smith, D.M., and Yamamoto, T. “Measuring voters’ multidimensional policy preferences with conjoint analysis: application to Japan’s 2014 election”, *Political Analysis*, 26(2), 190-209, (2018).

İşcan, A. “Use of suggestopedia in Turkish teaching as foreign language”, *Turkish Studies-International Periodical for the Languages, Literature and History of Turkish or Turkic*, 6(1),1317-1332, (2011).

Janak, J., “Issue tracking systems”, PhD Thesis, Masarykova Univerzita, Fakulta Informatiky, (2009).

Ji, S., Satish, N., Li, S., & Dubey, P., “Parallelizing word2vec in shared and distributed memory”, *IEEE Transactions on Parallel and Distributed Systems*, (2019).

Jiang, S., Zhang, H., Wang, Q., & Zhang, Y., “A debugging approach for Java runtime exceptions based on program slicing and stack traces”, *Proceedings of the 10th IEEE International Conference on Quality Software*, 393-398, (2010).

Kahraman, T., “Dilbilgisi öğretiminin durumu ve sonuçları üzerine küçük bir araştırma”, *Dokuz Eylül Üniversitesi Buca Eğitim Fakültesi Dergisi*, 27, (2010).

Kaur, M., and Kang, S., “Market basket analysis: identify the changing trends of market data using association rule mining”, *Procedia Computer Science*, 85, 78-85, (2016).

Keil, M., “Pulling the plug: software project management and the problem of project escalation”, *MIS Quarterly*, 19(4), 421-447, (1995).

Komiya, N., and Nakamura, J. “Exploring the types of casinos preferred in Japan via conjoint analysis of relevant words.”, *Advances in Human-Computer Interaction*, Article ID 8632892, 10 pages, (2019).

Türkçe sözlük. Türk Dil Kurumu, Ankara, (1983).

Lee, C.W., Licorish, S.A., Savarimuthu, B.T.R., and MacDonell, S.G., “Augmenting text mining approaches with social network analysis to understand the complex relationships among users' requests: a case study of the Android operating system”,

Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS), 1144-1153, (2016).

Li, N., and Wu, D. D., “Using text mining and sentiment analysis for online forums hotspot detection and forecast”, *Decision Support Systems*, 48(2), 354-368, (2010).

Lotufo, R., Passos, L., and Czarnecki, K., “Towards improving bug tracking Systems with game mechanisms”, *Proceedings of the IEEE 9th Working Conference on Mining Software Repositories*, 2–11, (2012).

Manning, C.D., and Schütze, H., “Foundations of statistical natural language processing”, *MIT press*, (1999).

Miner, G., Elder I.V., J., Fast, A., Hill, T., Nisbet, R., and Delen, D., “Practical text mining and statistical analysis for non-structured text data applications”, *Academic Press*, (2012).

Mishra A., Mishra D., “Software project management tools: a brief comparative view”, *ACM SIGSOFT Software Engineering Notes*, 38(3), (2013).

Mittermayer, M. A., “Forecasting intraday stock price trends with text mining techniques”, *Proceedings of the IEEE 37th Annual Hawaii International Conference on System Sciences*, 1-10, (2004).

Moreno, L., Treadway, J. J., Marcus, A., and Shen, W., “On the use of stack traces to improve text retrieval-based bug localization”, *Proceedings of the IEEE International Conference on Software Maintenance and Evolution*, 151-160, (2014).

Nasukawa, T., and Nagano, T., “Text analysis and knowledge mining system”, *IBM Systems Journal*, 40(4), 967-984, (2001).

Pal, M., and Mather, P. M., “An assessment of the effectiveness of decision tree methods for land cover classification”, *Remote Sensing of Environment*, 86(4), 554-565, (2003).

Pal, Mahender, “Microsoft Dynamics CRM 2011 application design”, *Packt Publishing Ltd*, (2012).

- Palaniappan, S., and Awang, R., “Intelligent heart disease prediction system using data mining techniques”, *Proceedings of the IEEE/ACS international conference on computer systems and applications*, AICSSA 2008, 108-115, (2008).
- Pandey, N., Sanyal, D. K., Hudait, A., and Sen, A., “Automated classification of software issue reports using machine learning techniques: an empirical study”, *Innovations in Systems and Software Engineering*, 13(4), 279-297 (2017).
- Prause, C. R., Scholten, M., Zimmermann, A., Reiners, R., and Eisenhauer, M., “Managing the iterative requirements process in a multi-national project using an issue tracker”, *Proceedings of the IEEE International Conference on Global Software Engineering*, 151-159, (2008).
- Raorane, A. A., Kulkarni, R. V., and Jitkar, B. D., “Association rule–extracting knowledge using market basket analysis”, *Research Journal of Recent Sciences ISSN*, 2277, 2502, (2012).
- Schröter, A., Bettenburg, N., and Premraj, R., “Do stack traces help developers fix bugs?”, *Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR)*, 118-121, (2010).
- Silva, C., and Ribeiro, B., “The importance of stop word removal on recall values in text categorization”, *Proceedings of the IEEE International Joint Conference on Neural Networks*, 3, 1661-1666, (2003).
- Srivastava, A.N., and Sahami, M. “Text mining: classification, clustering, and applications”, *Chapman and Hall/CRC*, (2009).
- Sureka, A., Goyal, A., and Rastogi, A., “Using social network analysis for mining collaboration data in a defect tracking system for risk and vulnerability analysis”, *Proceedings of the 4th ACM India Software Engineering Conference*, 195-204, (2011).
- Tan, A. H., “Text mining: the state of the art and the challenges”, *Proceedings of the PAKDD Workshop on Knowledge Discovery from Advanced Databases*, 8, 65-70, (1999).

- Tan, P.N., Steinbach, M., and Kumar, V., “Classification: basic concepts, decision trees, and model evaluation”, *Introduction to data mining*, 1, 145-205, (2006).
- Tang, ZhaoHui, and Jamine Maclellan, “Data mining with SQL Server 2005”, *John Wiley & Sons*, (2005).
- Tekin, T. ve Ölmez, M. “Türk dilleri: giriş”, *Simurg*, 156, (1999).
- Trnka, A., “Market basket analysis with data mining methods”, *Proceedings of the IEEE International Conference on Networking and Information Technology*, 446-450, (2010).
- Van Driel, M.A., Bruggeman, J., Vriend, G., Brunner, H.G., and Leunissen, J.A., “A text-mining analysis of the human genome”, *European Journal of Human Genetics*, 14 (5), 535, (2006).
- Verona J., “Practical DevOps”, *Packt Publishing Ltd*, (2016).
- Vural, H. ve Böler, T. “Ses ve şekil bilgisi”, Kesit Yayınları, Ankara, (2011).
- Web-1 <http://www.theitsmreview.com/2016/04/dikw-model-knowledge-management> (01.06.2019).
- Web-2 <https://libereurope.eu/wp-content/uploads/Text%20and%20Data%20Mining%20Factsheet.pdf>, (01.01.2019)
- Weisberg, S., “Applied linear regression”, *John Wiley & Sons*, 528, (2005).
- Wu, L.L., Xie, B., Kaiser, G.E., and Passonneau, R., “BugMiner: software reliability analysis via data mining of bug reports”, *Columbia University Computer Science Technical Reports*, CUCS-024-11, (2011).
- Wu, R., Zhang, H., Kim, S., and Cheung, S.C., “Relink: recovering links between bugs and changes”, *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, 15-25, (2011).
- Yavuzçehre, P.S., “Üniversitelerin kentlerine etkileri: Denizli Pamukkale üniversitesi örneği”, *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 235-250, (2016).

Yoon, B., and Park, Y. “Development of new technology forecasting algorithm: hybrid approach for morphology analysis and conjoint analysis of patent information.”, *Proceedings of the IEEE Transactions on Engineering Management*, 54(3), 588-599, (2007).

Zinner, T., Lemmerich, F., Schwarzmann, S., Hirth, M., Karg, P., and Hotho, A., “Text categorization for deriving the application quality in enterprises using ticketing systems”, *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery*, Springer, Cham, 325-336, (2015).

6. ÖZGEÇMİŞ

| | |
|-------------------------------|---|
| Adı Soyadı | :Mehmet Ali Küçük |
| Doğum Yeri ve Tarihi | :Buldan 20.04.1981 |
| Lisans Üniversite | :Selçuk Üniversitesi |
| Elektronik posta | :malik@pau.edu.tr |
| İletişim Adresi Başkanlığı | :Pamukkale Üniversitesi Bilgi İşlem Daire |