

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

STACK OVERFLOW PLATFORMUNDA DEVOPS
GÖNDERİLERİNDEKİ TRENDLERİN ETİKET VE KONU MODELLEME
ANALİZİ İLE BELİRLENMESİ

YÜKSEK LİSANS TEZİ

Burak BAKIRCI

TEMMUZ 2024

TRABZON



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

STACK OVERFLOW PLATFORMUNDA DEVOPS
GÖNDERİLERİNDEKİ TRENDLERİN ETİKET VE KONU MODELLEME
ANALİZİ İLE BELİRLENMESİ

Burak BAKIRCI

ORCID :

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“YAZILIM YÜKSEK MÜHENDİSİ”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 05 / 08 / 2024

Tezin Savunma Tarihi : 17 / 07 / 2024

Tez Danışmanı : Doç. Dr. Özcan ÖZYURT

ORCID :

Trabzon 2024

ÖNSÖZ

“Stack Overflow Platformunda DevOps Gönderilerindeki Trendlerin Etiket ve Konu Modelleme Analizi ile Belirlenmesi” isimli bu tez, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Anabilim Dalı, Tezli Yüksek Lisans Programı’nda hazırlanmıştır.

Tez çalışmam boyunca hiçbir desteğini esirgemeyen ve bilim insanı olma yolundaki ilk adımlarımı atarken kıymetli bilgileriyle ve tecrübesiyle yolumu aydınlatan danışman hocam Sayın Doç. Dr. Özcan ÖZYURT’a teşekkürü bir borç bilirim. Ayrıca, yapıcı eleştirileri ve önerileri ile tezime büyük katkıları bulunan saygıdeğer hocalarım Arş. Gör. Musa ASLAN, Arş. Gör. Hakan AYDIN ve Arş. Gör. İbrahim Uğur YILMAZ’a, teşekkür ederim. Tez çalışması sürecinde bana desteklerini esirgemeyen Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü’ndeki tüm değerli hocalarıma teşekkürlerimi sunarım.

Son olarak, doğduğum günden beri elimi hiç bırakmayan, hayatım boyunca aldığım tüm kararlarda bilgi ve tecrübelerinden faydalandığım ve her kararında arkamda olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen ve bugünlere gelmemde en büyük paya sahip olan annem Emine BAKIRCI ve babam Abdurrahman BAKIRCI’ya, ilk tanıdığım günden beri hep yanımda olup bana destek olan eşim Zeynep YANBAY BAKIRCI’ya ve tez çalışmamı yazarken çeşitli istekleri ile tatlı tebessümler ettiren biricik kızım Eylül Nisa BAKIRCI’ya çok teşekkür ederim.

Bu tez çalışmasının bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

Burak BAKIRCI
Trabzon 2024

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Stack Overflow Platformunda DevOps Gönderilerindeki Trendlerin Etiket ve Konu Modelleme Analizi ile Belirlenmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Doç. Dr. Özcan ÖZYURT’un sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 17 / 07 / 2024

Burak BAKIRCI

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ.....	X
SEMBOLLER VE KISALTMALAR DİZİNİ	XI
1. GENEL BİLGİLER	1
1.1. Giriş	1
1.2. DevOps	2
1.2.1. DevOps Yaşam Döngüsü	3
1.2.2. Sürekli Entegrasyon Kavramı	5
1.2.3. Dağıtım Otomasyonu Kavramı	5
1.2.4. Uygulama Performans Yönetimi Kavramı.....	6
1.2.5. Test Odaklı Geliştirme Kavramı	7
1.2.6. Sürekli Geri Bildirim Kavramı.....	7
1.2.7. Diğer DevOps Kavramları	8
1.2.8. DevOps Yaşam Döngüsü Aşamalarında Kullanılan Örnek Araçlar	8
1.3. Stack Overflow Soru-Cevap Sitesi	9
1.4. Konu Modelleme	11
1.4.1. Gizli Dirichlet Ayrımı	12
1.4.2. GDA Parametreleri.....	13
1.4.3. GDA Kullanım Alanları	14
1.5. İlişkili Çalışmalar	15
1.5.1. DevOps ile İlgili Çalışmalar.....	15
1.5.2. Stack Overflow Soru-Cevap Sitesi ile İlgili Çalışmalar	16
1.5.3. Konu Modelleme ve Yazılım Geliştirme Verileri Üzerindeki Uygulama ile İlgili Çalışmalar	17
1.6. Çalışmanın Amacı, Gerekçesi ve Önemi	19
1.6.1. Etiket Analizi	19
1.6.2. Konu Modelleme.....	19

1.6.3. DevOps Uygulamalarındaki Trendlerin Belirlenmesi	20
1.6.4. DevOps Topluluğunun Karşılaştığı Zorlukların Belirlenmesi.....	20
1.6.5. DevOps Alanındaki Gelecek Trendlerinin Tahmini	20
1.6.6. Eğitim Programlarının Güncellenmesi İçin Temel Bilgiler Sağlanması	20
2. YAPILAN ÇALIŞMALAR.....	22
2.1. Çalışmanın Ana Mimarisi	22
2.2. Veri Setinin Oluşturulması.....	23
2.3. Etiket Analizleri	25
2.4. Konu Modelleme Analizleri.....	27
3. BULGULAR VE TARTIŞMA	32
3.1. Etiket Analizleri Bulguları ve Tartışma	32
3.2. Konu Modelleme Analizleri Bulguları ve Tartışma.....	50
3.3. Etiket Analizleri ile Konu Modelleme Analizleri Bulgularının İlişkisi ve Tartışma.....	61
4. SONUÇLAR.....	64
4.1. Etiket Analizi Sonuçları	64
4.2. Konu Modelleme Analizi Sonuçları.....	64
4.3. Çalışmanın Genel Sonuçları.....	65
5. ÖNERİLER.....	67
5.1. DevOps Alanında Çalışanlara Yönelik Öneriler	68
5.2. Eğitim Kurumlarına Yönelik Öneriler	68
5.3. Politika Yapıcılara Yönelik Öneriler.....	69
6. KAYNAKLAR	70
ÖZGEÇMİŞ	

Yüksek Lisans Tezi

ÖZET

STACK OVERFLOW PLATFORMUNDA DEVOPS GÖNDERİLERİNDEKİ TRENDLERİN ETİKET VE KONU MODELLEME ANALİZİ İLE BELİRLENMESİ

Burak BAKIRCI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Özcan ÖZYURT
2024, 72 Sayfa

Bu tez çalışmasında, Stack Overflow (SO) platformundaki yazılım geliştirme ve operasyonların birleştirilmesiyle oluşan DevOps kavramı ile ilgili gönderiler üzerinde etiket analizi ve konu modelleme analizi yapılmıştır. Çalışmanın amacı, DevOps ile ilgili en çok tartışılan konuları, tarihsel gelişimi ve gelecekteki potansiyel eğilimleri belirlemektir. Veri seti, 2008-2023 yılları arasında SO platformunda “devops” etiketiyle paylaşılan 30.600 soru ve 34.327 cevaptan oluşmaktadır. Analiz sürecinde veriler, betimsel istatistikler ve Gizli Dirichlet Ayrımı (GDA) algoritması kullanılarak incelenmiştir. Çalışmanın bulguları “azure-devops” etiketinin 2014’ten itibaren belirgin bir artış gösterdiğini ortaya koymuştur. Bunun yanında DevOps ile ilgili gönderilerin toplam 24 ana konu altında kümelendiği ve “sürekli entegrasyon”, “dağıtım otomasyonu”, “uygulama performans yönetimi” ve “test odaklı geliştirme” gibi konuların öne çıktığı görülmüştür. Sonuçlar, DevOps topluluğunun genel olarak sürekli entegrasyon ve dağıtım süreçlerindeki karmaşıklıklar ile, araçların entegrasyonu ve performans yönetimi ile uğraştığını göstermiştir. Bu konuların üstesinden gelmek için işbirliği ve iletişimin artırılması, sürekli öğrenme ve iyileştirme süreçlerine odaklanılması önerilmektedir. Bu çalışma, DevOps pratiklerinin optimizasyonuna ve bu alandaki bilgi birikiminin derinleştirilmesine önemli katkılar sağlamaktadır.

Anahtar Kelimeler: DevOps, Stack Overflow, Konu modelleme, GDA

Master Thesis

SUMMARY

IDENTIFICATION OF TRENDS IN DEVOPS POSTS ON THE STACK OVERFLOW PLATFORM USING TAG AND TOPIC MODELING ANALYSIS

Burak BAKIRCI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Software Engineering Master Program
Supervisor: Assoc. Prof. Özcan ÖZYURT
2024, 72 Pages

In this thesis, a tag analysis and topic modeling analysis are conducted on posts related to the concept of DevOps, which combines software development and operations, on the Stack Overflow (SO) platform. The aim of the study is to identify the most discussed topics related to DevOps, its historical development, and potential future trends. The dataset consists of 30,600 questions and 34,327 answers posted on the SO platform under the “devops” tag between 2008 and 2023. During the analysis process, descriptive statistics and the Latent Dirichlet Allocation (LDA) algorithm were used to examine the data. The study’s findings revealed a significant increase in the use of the “azure-devops” tag starting from 2014. Additionally, DevOps-related posts were clustered under a total of 24 main topics, with themes such as “continuous integration,” “deployment automation”, “application performance management,” and “test-driven development” standing out. The results indicate that the DevOps community generally deals with complexities in continuous integration and deployment processes, tool integration, and performance management. To overcome these issues, it is recommended to enhance collaboration and communication, and focus on continuous learning and improvement processes. This study makes significant contributions to the optimization of DevOps practices and the deepening of knowledge in this field.

Key Words: DevOps, Stack Overflow, Topic modeling, LDA

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Yazılım projesinin en temel yaşam döngüsü.....	3
Şekil 2. Yazılım geliştirme yaşam döngüsü	4
Şekil 3. DevOps yaşam döngüsü	4
Şekil 4. SO web sitesi görünümü ve “devops” araması sonucu	10
Şekil 5. GDA için temsili grafiksel model (Blei vd., 2003)	12
Şekil 6. Çalışmanın ana mimarisi	22
Şekil 7. Yıllara göre soru ve cevap sayıları	24
Şekil 8. Konu modelleme analizi adımları	28
Şekil 9. GDA algoritmasının akış şeması	30
Şekil 10. Eğim değerlerine göre en iyi ilk 15 etiket	37
Şekil 11. Eğim değerlerine göre en kötü ilk 15 etiket	38
Şekil 12. Hacimlerin eğim hesabına göre en iyi ilk 15 etiket.....	40
Şekil 13. Hacimlerin eğim hesabına göre en kötü ilk 15 etiket.....	41
Şekil 14. Yıllara göre eğim sıralı ilk 15 etiketin gelecek tahmini	43
Şekil 15. Diğer etiketlere kıyasla eğim sıralı en iyi ilk 15 etiket.....	45
Şekil 16. Diğer etiketlere kıyasla eğim sıralı en kötü ilk 15 etiket.....	46
Şekil 17. Diğer etiketlere kıyasla eğim sıralı en iyi ilk 15 etiketin gelecek tahmini	48
Şekil 18. PyLDavis ile konu modellemenin görselleştirilmesi	50
Şekil 19. Konu isimlendirmeleri ve yüzdeleri	55
Şekil 20. Konu sayılarına göre eğim sıralı konu listesi	56
Şekil 21. Diğer yıllara kıyasla eğim sıralı konu listesi	57
Şekil 22. Diğer konulara kıyasla eğim sıralı konu listesi	58
Şekil 23. Diğer konulara kıyasla en iyi ilk 15 konunun gelecek 3 yıl tahmini.....	60

TABLÖLER DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Veri setine ilişkin bilgiler.....	25
Tablo 2. En çok kullanılan ilk 50 etiket, kullanım sayıları (n) ve yüzdeleri (f)	33
Tablo 3. En çok kullanılan ilk 20 ikili etiket, sayısı(n) ve yüzdesi(f)	34
Tablo 4. En çok kullanılan ilk 20 üçlü etiket, sayısı(n) ve yüzdesi(f).....	36
Tablo 5. Konu numarası, konu isimlendirmesi, konu terimleri ve yüzdeleri	52
Tablo 6. Yıllara göre konu sayıları, toplam adedi(n) ve yüzdesi(f)	54
Tablo 7. Pozitif performans gösteren ilk 15 etiketin konu dağılımı.....	62
Tablo 8. Negatif performans gösteren ilk 15 etiketin konu dağılımı.....	63

SEMBOLLER VE KISALTMALAR DİZİNİ

<i>AWS</i>	: Amazon Web Services
<i>CI/CD</i>	: Continuous Integration / Continuous Development
<i>DDİ</i>	: Doğal Dil İşleme
<i>DevOps</i>	: Development and Operations
<i>GDA</i>	: Gizli Dirichlet Ayrımı
<i>Git</i>	: Global Information Tracker
<i>IaC</i>	: Infrastructure as Code
<i>pyLDavis</i>	: Python LDA Sonuç Görselleştirme Kütüphanesi
<i>SO</i>	: Stack Overflow
<i>TF – IDF</i>	: Term Frequency-Inverse Document Frequency
α	: Belgelerin konu dağılımının homojen olmasının parametresi
β	: Konuların kelime dağılımının homojen olmasının parametresi
θ	: Her bir belgenin konu dağılımı
w	: Kelimenin kendisini temsil eder
z	: Kelimenin atandığı konuyu temsil eder
M	: Toplam doküman sayısı
N	: Dokümanın boyutu
c_v	: coherence değeri

1. GENEL BİLGİLER

1.1. Giriş

Bilgisayar teknolojileri alanında mevcut yazılım teknolojilerinin çeşitliliği, kullanım alanlarına göre belirlenmektedir. Sektörel ihtiyaçlar, bu teknolojilerin gelişim yönünü belirgin bir şekilde etkilemektedir. Bankacılık gibi dinamik sektörlerde, uygulamaların hem mobil işletim sistemlerinde hem de masaüstü internet tarayıcılarında güvenli bir şekilde işlev görmesi beklenmektedir. Bu gereksinim, yazılım geliştirme ortamlarının çeşitlenmesine neden olmuştur. Yazılım geliştirme süreçlerinde, geliştirme ve operasyon ekiplerinin farklı alanlarda çalışması yaygındır; burada geliştirme ekibi yazılımın geliştirilmesi ve test edilmesiyle, operasyon ekibi ise yapılandırma yönetimi, hata takibi ve yazılımın dağıtımı gibi konularla ilgilenir (Khan vd., 2023). Yazılım mühendisliği, günümüzde geniş bir teknoloji, araç, dil ve platform yelpazesine sahiptir (Barua vd., 2014). Takım çalışması, kod incelemeleri, yazılım testleri ve yayımlama gibi süreçlerin sürekli gelişim ve entegrasyonunu sağlamak amacıyla, geliştirme ve operasyon ekiplerinin birlikte çalışması gerekmektedir. Bu iş birliğine atıfta bulunmak için, İngilizce'deki "development" ve "operations" terimlerinin baş harflerinin birleşiminden "DevOps" terimi türetilmiştir. Yazılım ve bilişim teknolojileri sektörlerinde hız ve verimlilik, temel öncelikler arasında yer almaktadır (Ebert vd., 2016). DevOps, yenilikçi ürünlerin ve özelliklerin piyasaya daha hızlı sunulabilmesi amacıyla bir paradigma olarak gelişmiştir (Ebert vd., 2016). Bu kavram, yazılım endüstrisinde giderek daha fazla önem kazanmaktadır. DevOps'a yönelik yeni konferanslar ve sistematik incelemeler, yazılım mühendisliği araştırmacılarının bu alana büyük ilgi gösterdiğini ortaya koymaktadır (Jabbari vd., 2016). Yazılım sektöründe çalışan profesyonellerin DevOps özelliklerine sahip olması beklenmektedir, ve bu özellikler yazılım alanında kariyer hedefleyen genç bireyler için iş ilanlarında sıkça talep edilmektedir. DevOps'un yazılım endüstrisindeki uygulama alanlarını analiz etmek için kullanılan yöntemlerden biri, ilgili toplulukların incelenmesidir.

Yazılım geliştirme topluluğu, bilgi paylaşımında soru-yanıt sitelerini yoğun bir şekilde kullanmaktadır. Zaman içinde, bu tür siteler önemli yazılım mühendisliği bilgi kaynakları haline gelmiştir (Barua vd., 2014). Bu platformlar arasında Stack Overflow (SO) en büyük kullanıcı ve yorum hacmine sahip olanlardan biridir (Moutidis vd., 2021). SO,

programlama ile ilgili soruların ücretsiz bir şekilde sorulup yanıtlandığı, yanıtların soru sahibinin memnuniyetine göre oylanabildiği ve soruların konularına göre etiketlenebildiği bir platformdur (Baltadzhieva vd., 2015). SO’da yazılım ve bilişim teknolojilerine dair geniş bir yelpazede sorular yer almakta ve bu sorular ilgili teknolojilerle etiketlenmektedir. DevOps’un yazılımdaki güncel konumu hakkında bilgi edinmek, büyük veri hacmi nedeniyle zor olabilir. Bu tür analizler, metin madenciliği ve doğal dil işleme teknikleri kullanılarak gerçekleştirilebilir (Gurcan vd., 2021). Doğal dil işleme tekniklerinden biri olan Gizli Dirichlet Ayrımı (GDA), metin koleksiyonları gibi ayrık veri kümeleri için kullanılan bir olasılıksal modeldir (Blei vd., 2003). GDA algoritması, belgelerin altında yatan konuları belirleyerek, metin koleksiyonlarının daha anlamlı bir şekilde analiz edilmesini sağlamaktadır.

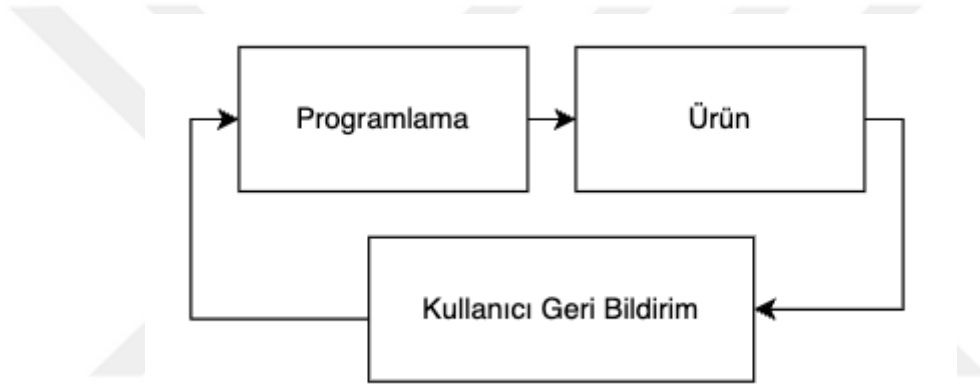
1.2. DevOps

DevOps, özünde çevik yazılım geliştirmenin bağlamında ortaya çıkmış olup, küçük sürümlerde sürekli çalışan yazılımın hızlı teslimatını mümkün kılan bir yaklaşımdır (Ghantous vd., 2019). Kuruluşlar, DevOps çalışma yöntemlerini benimseme konusunda önemli bir ilgi göstermektedirler (Bou Ghantous vd. 2017). Günümüzde yazılım geliştiriciler birçok projeyi farklı takımlar ile birlikte geliştirmektedirler. Müşteri bir yazılım şirketinden hizmet aldığı anda olası sistemsel geri dönüşlerin geliştiriciler tarafından iş olarak alınması ve yapılan düzenlemelerin belirli bir takvime göre halihazırda çalışan ortama aktarılması gerekmektedir. DevOps araçları yaygınlaşmadan önce yazılım şirketlerinde görev dağılımları geliştirme, operasyon, veri tabanı, güvenlik gibi ekipler tarafından yürütülürken DevOps kavramı ile sürekli geliştirme (Continuous Integration) ve sürekli teslimat (Continuous Delivery) tüm ekipler tarafından yapılabilir hale gelmektedir.

Amazon Web Services (AWS)’ye göre DevOps, kurumların yazılım geliştirme ve altyapı yönetim süreçlerini hızlandırarak, uygulama ve hizmetleri daha hızlı ve verimli bir şekilde sunmalarını sağlayan bir kültür, yöntem ve araçlar birleşimidir. Bu sayede onu kullanan kurumlara, geleneksel yöntemleri kullanan kurumlara göre daha hızlı ürün geliştirme ve iyileştirme imkanı sunmaktadır. (URL-2, 2022). Bir yazılım projesinin planlanmasından bakım aşamasına kadar tüm süreç DevOps araçları ile yürütülür.

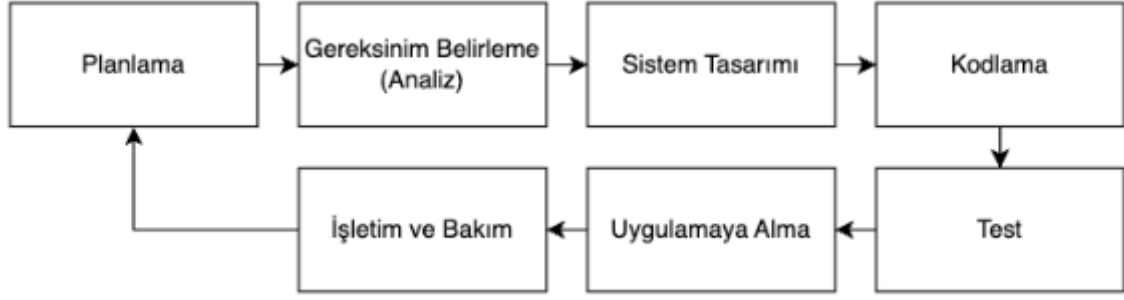
1.2.1. DevOps Yaşam Döngüsü

DevOps yaşam döngüsü, geleneksel yazılım geliştirme süreçlerini kapsayarak, tüm ekibin sürecin her aşamasına dahil olduğu, çeşitli yardımcı araçların kullanıldığı kapsamlı bir döngüdür. DevOps yaşam döngüsünü detaylandırmadan önce, temel yazılım projelerinin ve klasik yazılım geliştirme süreçlerinin yaşam döngülerini incelemek önemlidir. Yazılım, kullanıcılar tarafından kullanıma sunulduktan sonra, kullanıcıların karşılaştıkları sorunların geliştiricilere bildirilmesiyle başlayan ve sürekli olarak tekrar eden bir yeniden programlama ve geliştirme döngüsüne girer. Bu süreç, zamanla daha fazla detaylandırılarak geliştirilmiştir. Aşağıda sürecin en temel hali Şekil 1’de gösterilmektedir.



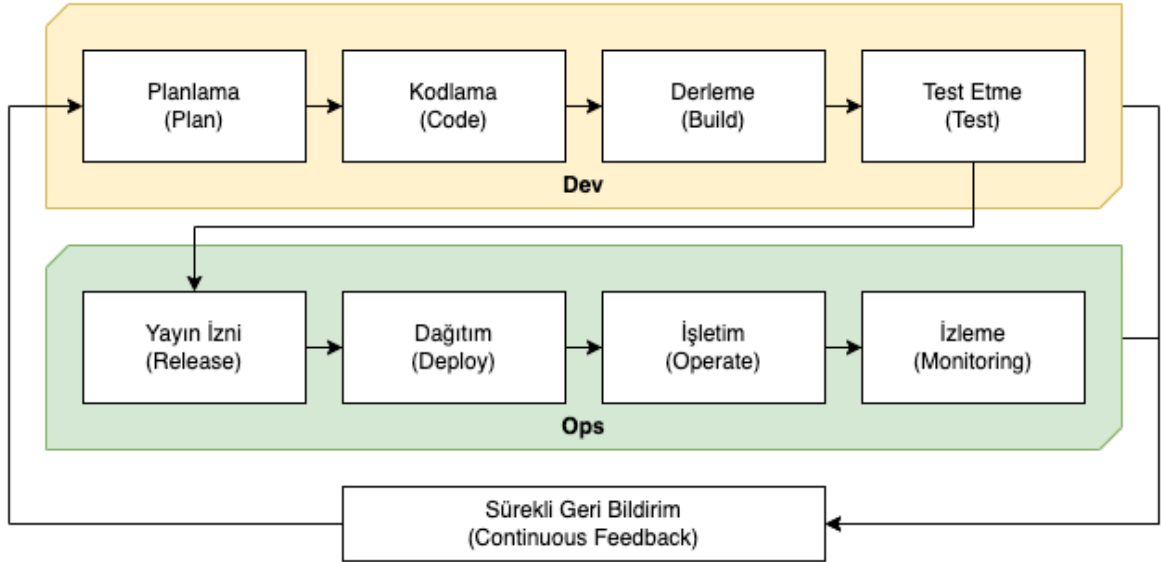
Şekil 1. Yazılım projesinin en temel yaşam döngüsü

Şekil 1’de bir yazılım projesinin en temel yaşam döngüsünü göstermektedir. Bu temelde programlama ile ürün ortaya çıkarılmaktadır ve ürünün kullanımı süresince kullanıcıların geri bildirimleri ile tekrar programlama yapılmaktadır. Fakat bu süreçte planlama olmadığından kullanıcı geri bildirimleri ürün ve bakım maliyetlerini oldukça artırmaktadır. Bu da sürekli yazılım geliştirmeyi gerektirmektedir. Sürekli yazılım geliştirme ihtiyacı, planlamanın yapılmasının elzem olduğu sonucunu doğurmuştur. Buradan hareketle yazılım geliştirme yaşam döngüsü genişletilmiştir. Yazılım geliştirme yaşam döngüsü; planlama, gereksinim belirleme, analiz, sistem tasarımı, kodlama, test, uygulamaya alma ve işletim ve bakım adımlarından oluşmaktadır (Yıldız, 2021). Bu adımlar sırası ile birbirini izlemekte ve son adım olan işletim ve bakım adımı tekrar planlama adımı ile bağlanmaktadır. Şekil 2’de yazılım geliştirme yaşam döngüsü şematize edilmiştir.



Şekil 2. Yazılım geliştirme yaşam döngüsü

Şekil 2’de görüldüğü gibi yazılım geliştirme yaşam döngüsü genel anlamı ile 7 aşamadan oluşmaktadır ve son aşama olan işletim ve bakım aşaması ile tekrar planlama adımı ile süreç devam etmektedir. Her bir adım bir önceki adıma bağlıdır. DevOps yaşam döngüsünde ise tüm adımlarda bir süreklilik söz konusudur. Şekil 3’te açık sarı zemin içerisinde bulunanlar geliştirme aşamalarını, açık yeşil zemin içerisinde bulunanlar ise operasyon adımlarını içermektedir ve tüm adımların herhangi birinden sürekli geri bildirim alındığı belirtilmektedir.



Şekil 3. DevOps yaşam döngüsü

Şekil 3’te DevOps yaşam döngüsü şematize edilmiştir. Geliştirme ve operasyon ekipleri planlama aşaması dahil tüm aşamalarda geri bildirim alabilmektedir. DevOps yaşam döngüsü adımlarının ilk dört aşaması planlama (plan), kodlama (code), derleme (build), test

etme (test) olarak geliştiricilerin üzerinde olan işlerdir ve bu dört aşamaya sürekli entegrasyon (continuous integration) denmektedir. Sonrasında ise ikinci bir dörtlü işlemler grubu vardır. Bunlar; yayın izni (release), dağıtım (deploy), işletim (operate) ve izleme (monitoring) olarak operasyonel ekip üzerinde olan işlerdir ve bu dörtlüye sürekli dağıtım (continuous delivery) denmektedir. Bu adımların tamamını kapsayan gerçek zamanlı iletişim ve yaşam döngüsü boyunca izlenebilirlik süreçleri ise sürekli geri bildirim (continuous feedback) olarak adlandırılmaktadır.

1.2.2. Sürekli Entegrasyon Kavramı

Sürekli entegrasyon, yazılım geliştirme ekiplerinin, genellikle günlük olarak en az bir kez olmak üzere, sık sık kodlarını bir araya getirdikleri bir uygulamadır. Bu sık entegrasyonlar, her bir birleştirmede otomatik olarak gerçekleştirilen derlemeler (testler dahil) ile desteklenir, böylece entegrasyon hataları olabildiğince erken tespit edilir (Fowler vd., 2006). Yazılım projeleri, değişikliklerin ekip üyeleri tarafından kolayca izlenebilmesi ve potansiyel hataların erken aşamada tespit edilip, deneyimli yazılımcılar tarafından onaylanabilmesi için doküman yönetim sistemleri kullanılarak geliştirilmektedir. Bu alanda en yaygın kullanılan araç versiyon kontrol sistemi olan Git'tir ve diğer sistemler farklı isimlere sahip olsa da, temelde Git komutlarını kullanmaktadır. Yazılım geliştirme ortamlarının çoğu Git desteği sağlamakta olup, sürekli entegrasyonun vazgeçilmez bir aracı haline gelmiştir. Ekip üyeleri, birden fazla yazılım projesinde küçük değişiklikler ve geliştirmeler yaparlar. Bu değişiklikler, otomatik bir sistem tarafından denetlenir ve onaylanır. Bu da kullanıcıların büyük hatalarla karşılaşmalarını önler.

1.2.3. Dağıtım Otomasyonu Kavramı

Dağıtım otomasyonu (deployment automation), yazılım geliştirme sürecinin bir aşamasıdır ve işin en güncel versiyonunun canlı ortama başarıyla taşınmasını sağlar. Gelişen eklentiler sayesinde yapılan değişiklikler temel mantıksal testlere yazılımsal olarak otomatikleştirilerek uygulanabilmektedir. Örneğin bir web sayfasının üst menü alanının arka zemin renginin değişikliği istendiğinde bu değişiklik kısa bir sürede yapılabilir. Bu küçük değişiklik yazılım akışını bozmayacağından kıdemli bir yazılımcının onaylamasını gerektirecek bir durum da olmayacaktır. Bu tür değişiklikler onay aşamasında, kıdemli yazılımcının sadece hangi dosyalarda değişiklik yapıldığına bakmasını gerektirir. Bu da

zamandan tasarruf etmemizi sağlayacaktır. Aynı zamanda dağıtım operasyonunda uzak bir sunucu çalıştırılıyor ve bu sunucu sürekli hizmet veriyor ise bu sunucunun da hizmeti durdurulmadan bu dağıtımın yapılması gerekmektedir. Olası hataların test edilip bir hata yok ise doküman yönetim sisteminde onaylanan değişikliklerin ve varsa sunucuların hizmetini kesmeden otomatik olarak yazılımın en son halinin devreye alınması işi dağıtım otomasyonu işini kapsamaktadır. Sürekli teslimat, geliştiricilerin sık sık değişiklikleri ana geliştirme hattına entegre ettiği ve yazılımlarının sürümlerini ürettiği bir çevik yazılım geliştirme uygulamasıdır (Vassallo vd. 2016). DevOps içerisinde sürekli teslimat kavramı ile doğrudan ilişkili olan dağıtım otomasyonu çevik proje geliştirmede çok önemli bir bölümdür.

1.2.4. Uygulama Performans Yönetimi Kavramı

Uygulamanın performans yönetimi (application performance management) DevOps kavramları içerisinde sürekli geri bildirim konusu ile bağlantılıdır. Performans, zamanında tamamlama yeteneği ve kaynak kullanımı gibi sistem özelliklerini tanımlar ve birçok uygulamada performansın ölçüldüğü değerler arasında yanıt süresi, verimlilik ve kaynak kullanımı bulunur (Brunnert vd., 2015). Yazılımların internete bağımlı olması, donanım kullanımları, kullanıcıların sayısı gibi faktörler yazılım çalışmasını etkileyen faktörlerdir. Sürekli entegrasyon ve sürekli geri bildirim olan bir yazılıma aşağıdaki yetenekler kazandırılmalıdır:

- Sayısal veriler (kullanıcı sayısı, istek sayısı, en yoğun istek alınan saatler gibi)
- Loglama (yapılan her bir isteğin ve cevabının tarihsel olarak kaydının tutulması)
- İzleme (sayısal verileri izlenebilir grafiklere dönüştürme)
- Uyarı verme (izleme esnasında müdahale gerektirecek bir durumda geliştiricilere bildirim yapılması)

Bu yeteneklerin kazandırıldığı bir uygulamada geri bildirimlere yapılacak olan müdahaleler daha hızlı bir şekilde gerçekleştirilir.

1.2.5. Test Odaklı Geliştirme Kavramı

Test odaklı geliştirmede (test driven development) kod yazılmadan önce testler yazılmaktadır. Bu konuda yazılım mühendisliği alanında test mühendisliği özel uzmanlık alanı olarak gelişmektedir. Kod yazılmadan önce analiz adımlarından sonra testler yazılmaktadır. Yazılım testleri konularından birim testleri, entegrasyon testleri, uçtan uca sistem testleri, kabul testleri gibi testler yapılmaktadır. Maximilien ve arkadaşları (2003) test odaklı geliştirme yaklaşımı kullanılarak oluşturulan bir sistem ile geçici birim testi yaklaşımı kullanılarak oluşturulan benzer bir sisteme kıyasla kusur oranlarını yaklaşık yüzde 50 azaltmışlardır. Test odaklı geliştirmenin amacı neyin nerede hataya yol açabileceğini önceden keşfetmek ve doğru bir şekilde tüm sistemin çalışıyor olacağını bilmektir.

1.2.6. Sürekli Geri Bildirim Kavramı

DevOps içinde sürekli geribildirim, yazılım geliştirme ve işletme süreçlerinin entegre edilmesi ve otomatikleştirilmesi çerçevesinde, ürünün kalitesini ve teslimat süreçlerinin verimliliğini artırmak için kritik bir rol oynar. DevOps yaklaşımı, sürekli entegrasyon, sürekli teslimat ve sürekli izleme gibi uygulamaları içerir ve bu uygulamaların her biri sürekli geri bildirim mekanizmalarına dayanır. Sürekli entegrasyonda geri bildirimi; geliştiricilerin kod değişikliklerini sık sık ve düzenli olarak ana kod tabanına birleştirmelerini gerektirir. Her birleştirme işlemi otomatikleştirilmiş derlemeler ve testlerle desteklenir, bu da kodun kalitesi ve entegrasyonun başarısı hakkında anında geri bildirim sağlar. Bu sürekli geribildirim, hataların erken tespit edilmesini ve düzeltilmesini sağlayarak ürün kalitesinin artırılmasına yardımcı olur. Sürekli teslimatta geri bildirim; yazılımın herhangi bir zaman diliminde güvenli ve otomatik bir şekilde üretim ortamına teslim edilebilir durumda tutulmasını sağlar. Bu süreç, yazılımın kullanıcıya teslim edilmeden önce çeşitli test ve onay aşamalarından geçmesini içerir, bu da geliştirme ve işletme ekiplerine yazılımın üretim ortamına hazır olup olmadığı hakkında sürekli geri bildirim sağlar. Sürekli izlemede geri bildirim; üretim ortamındaki yazılımın performansını, kullanılabilirliğini ve güvenliğini izlemeyi içerir. Bu izleme, gerçek zamanlı veri toplama ve analizi aracılığıyla gerçekleştirilir ve potansiyel sorunların veya performans düşüklüklerinin erken tespit edilmesine olanak tanır. Bu geri bildirim, ekiplerin proaktif bir şekilde sorunları çözmesine ve sürekli iyileştirmeler yapmasına yardımcı olur. Sürekli geribildirim, DevOps kültürünün

temel bir unsuru olarak, ekipler arası işbirliğini ve iletişimi artırır, süreçlerin şeffaflığını sağlar ve hızlı adaptasyon ile inovasyonu teşvik eder. Bu sürekli geri bildirim döngüsü, ekiplerin daha hızlı öğrenmesine, süreçleri daha etkin bir şekilde yönetmesine ve müşteri memnuniyetini artırarak rekabet avantajı sağlamasına olanak tanır.

1.2.7. Diğer DevOps Kavramları

Geliştirilen yazılımın büyüklüğüne, kullanıcı sayısına, kullanıcı kitlesine, üçüncü dereceden servis kullanımlarına göre yazılım ürünleri için DevOps araçları farklılaşmaktadır. Bu farklılaşmalar dikkate alındığında aşağıdaki bazı kavramlar da DevOps kavramı içerisinde değerlendirilmektedir ve yazılım mühendisleri açısından dikkate alınması gereken kavramlardır. Bu kavramlardan bazıları; kod kapsamı (code coverage), bilgisayar destekli yazım denetimi (linting), geçici ortamlar (ephemeral environments), sürekli evreleme tanımı (continuous staging definition), sanal makineler ve konteynerler (virtual machines and containers), sürekli dağıtımlar (rolling deployments), mavi/yeşil dağıtımlar (blue/green deployments), gökkuşağı dağıtımlar (rainbow deployments), kanarya dağıtımlar (canary deployments), otomatik ölçeklendirme (autoscaling), hizmet keşfi (service discovery), günlük toplama (log aggregation), hayati ürün metrikleri (vital product metrics) gibi kavramlardır.

1.2.8. DevOps Yaşam Döngüsü Aşamalarında Kullanılan Örnek Araçlar

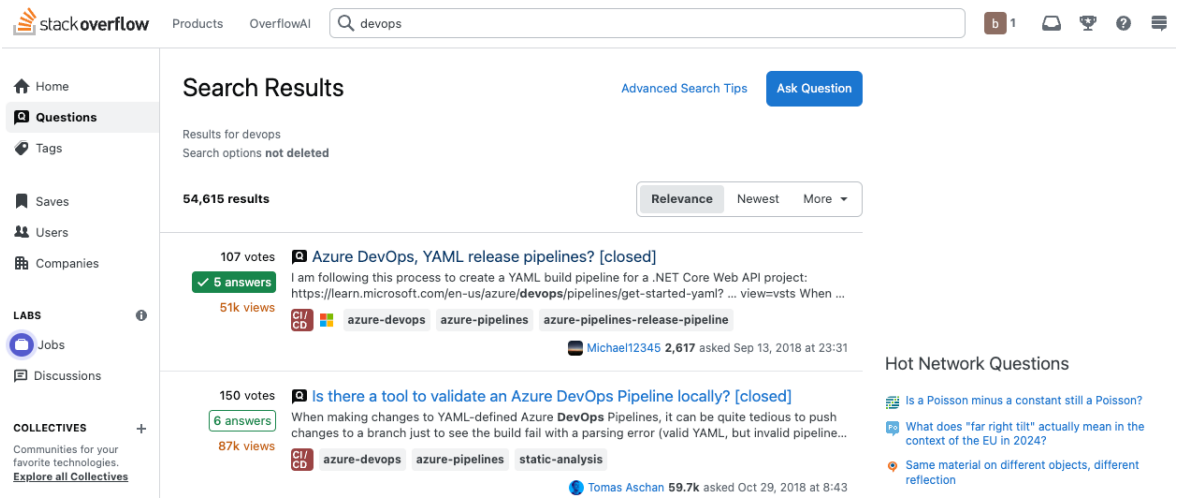
Devops yaşam döngüsü aşamalarında kullanılan örnek araçlar, yazılımın geliştirilmesinden dağıtımına kadar olan süreçlerde kullanılır ve her birinin, süreçleri daha verimli hale getirme potansiyeli bulunmaktadır. Örneğin, versiyon kontrol sistemleri kod değişikliklerini yönetirken, sürekli entegrasyon ve dağıtım araçları yazılımın sorunsuz bir şekilde kullanıcıya ulaşmasını sağlamaya yardımcı olur. Konfigürasyon yönetimi araçları, altyapının tutarlı ve güvenilir bir şekilde yönetilmesine katkıda bulunurken, konteyner teknolojileri uygulamaların taşınabilir ve ölçeklenebilir olmasını sağlar. İzleme ve loglama araçları ise sistem performansını ve sağlığını sürekli olarak takip ederek potansiyel sorunların erken tespit edilmesine olanak tanır. Bir organizasyonun teknik yeteneklerini ve kültürel ortamını güçlendirmek, yazılım teslimat performansı ve dolayısıyla genel organizasyonel performans üzerinde önemli bir etkiye sahip olabilir ve bu nedenle, her

aracın seçimi ve uygulanması, belirli bir organizasyonun hedefleri ve mevcut pratikleriyle uyumlu olmalıdır (URL-4, 2023). Araçlara verilebilecek örnekler şu şekilde listelenebilir:

- Versiyon kontrol sistemi araçları : Git, Gitlab, Bitbucket
- Sürekli entegrasyon/sürekli dağıtım araçları : Jenkins, Gitlab CI/CD, CircleCI, LayerCI, TravisCI
- Konfigürasyon yönetimi araçları : Ansible, Chef, Puppet
- Konteyner araçları : Docker, Kubernetes
- İzleme ve loglama araçları : Prometheus, Grafana, ELK Stack(Elasticsearch, Logstash, Kibana)

1.3. Stack Overflow Soru-Cevap Sitesi

SO, 2008 yılında kurulmuş bir soru-cevap sitesidir (URL-1, 2022). SO, farklı eğitim ve seviyedeki yazılım geliştiricilerin ve programcıların yardımlaşmasını sağlamak amacıyla geliştirilmiş olan bir soru-cevap sitesidir (Squire, 2015). Veri olarak odaklandığı alan yazılım sektörü ve buna bağlı olarak yazılımla ilgilenen herkeştir. SO’nun kuruluş amacı, yazılım geliştiricilerin karşılaştıkları programlama sorunlarına çözüm bulabilmeleri, bilgi paylaşımında bulunabilmeleri ve en iyi uygulamaları tartışabilmeleri için bir platform sağlamaktır. SO, yazılım geliştiricilerin karşılaştıkları teknik zorlukları aşmalarına yardımcı olmak ve yazılım geliştirme topluluğunda bilgi paylaşımını teşvik etmek amacıyla tasarlanmış, yazılım geliştirme ile ilgili soruların sorulup cevaplandığı, kullanıcıların soruları oylayıp yorum yapabildikleri, ayrıca konulara göre etiketleyebildikleri bir topluluk odaklı bir soru-cevap platformudur. Sitenin genel görünümü ve “devops” araması sonucu Şekil 4’te verilmiştir.



Şekil 4. SO web sitesi görünümü ve “devops” araması sonucu

Şekil 4’te SO soru-cevap sitesinde yapılan “devops” kelimesinin arama sonucunda gelen bilgilerde sorular ve cevaplar ayrı ayrı sıralanmaktadır. Aranılan bir kelimenin sonuçlarının getirilmesi için kelimenin site üzerindeki soruların başlığında, içeriğinde veya anahtar kelimelerinde olması yeterlidir. Arama sonucunda gelen bilgiler aşağıda sıralanmıştır:

- Soru başlığı: Sorunun içeriğini özetleyen başlıktır.
- Soru metni: Sorunun detaylı açıklamasıdır.
- Soruya verilen cevaplar: Soruya toplamda verilen cevapların listesidir.
- Sorunun görüntülenme sayısı: Sorunun ne kadar görüntülendiğini belirten sayısal değerdir.
- Soruya verilen cevaplardan kabul edilen cevap veya cevapların varlığı: Soru sahibi veya moderatörler tarafından kabul edilen cevapların varlığını belirten bilgidir.
- Sorunun aldığı skor: Sorunun toplam oy sayısı veya puanıdır.
- Sorunun etiketleri: Sorunun kategorize edilmesine yardımcı olan etiketlerin listesidir.
- Soruya yapılan yorumlar: Soruya gelen yorumların listesidir.
- Yorumlara verilen cevaplar: Yapılan yorumlara verilen cevapların listesidir.
- Soruya verilen cevaplara yapılan yorumlar: Soruya verilen cevaplara yapılan yorumların listesidir.
- Soruya verilen cevaplara yapılan yorumlara verilen cevaplar: Soruya verilen cevaplara yapılan yorumlara verilen cevapların listesidir.

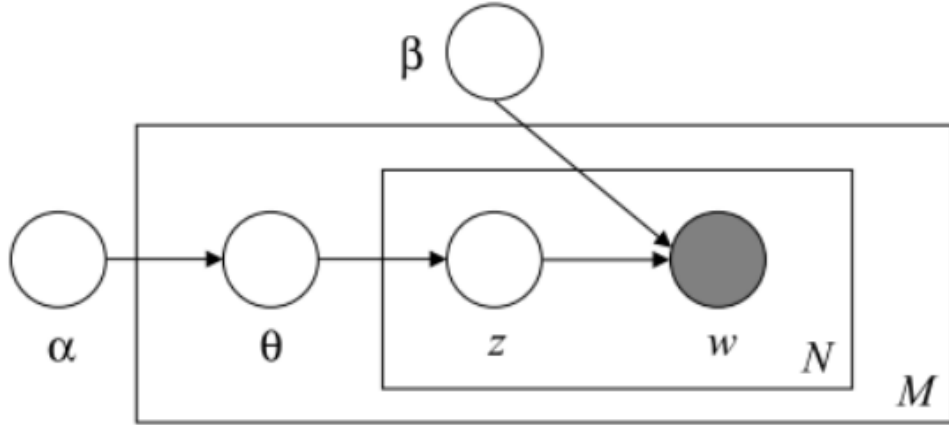
- Sorunun oluřturulma tarihi.
- Sorunun gncelleme tarihi.
- Sorunun aık bir soru mu yoksa kapatılmıř bir soru mu olduėu bilgisi: Sorunun czlp czlmediėini belirten bilgidir.
- Soruyu soran kullanıcı profil bilgileri: Soruyu soran kullanıcıya iliřkin profil bilgileridir.
- Soruyu soran kullanıcının istatistikleri: Soruyu soran kullanıcının site zerindeki aktivitelere iliřkin istatistiklerdir (rneėin ka soru sorduėu, ka yanıt verdiėi, aldıėı toplam skor vb.).

1.4. Konu Modelleme

Metin madenciliėi, doėal dil metninden anlamlı bilgiler toplamayı hedefleyen, geliřmekte olan bir alanı temsil etmektedir ve bu, belirli amalar iin faydalı bilgileri ıkarmak zere metni analiz etme sreci olarak tanımlanmaktadır (Witten, 2004). Konu modelleme yntemleri, kelimeler zerindeki olasılık daėılımlarını kullanarak, belgelerdeki konuların rastgele bir araya gelmesi ilkesine dayanmaktadır (Stevyers vd. 2007). Konu modelleme, metin madenciliėi ve Doėal Dil İřleme (DDİ) disiplinlerinde kilit bir denetimsiz ėrenme tekniėi olarak grlmektedir. Bu yntem, veri setlerinde etiketleme veya sınıflandırma ihtiyaını ortadan kaldırarak, algoritmaların veri ierisindeki gizli temaları veya yapıları baėımsız bir řekilde tanımlamasına olanak tanımaktadır. zellikle byk ve yapılandırılmamıř metin kmelerinde, konu modelleme teknikleri metinler arasındaki benzerlikleri ve farklılıkları belirleyerek onları tematik gruplara ayırmaktadır. Bu iřlemler, metinlerin ierdiėi bilgilerin daha iyi anlařılmasına ve sınıflandırılmasına yardımcı olmaktadır. Denetimsiz ėrenme yaklařımı olarak konu modelleme, belirli bir girdiye karřılık gelmesi gereken ıktılar olmaksızın veri ierisindeki desenleri ve iliřkileri keřfetmeyi amalamaktadır. Konu modelleme, belgelerin altında yatan tematik yapıları ortaya ıkarak metin koleksiyonlarının daha anlamlı bir řekilde analiz edilmesini saėlamaktadır.

1.4.1. Gizli Dirichlet Ayrımı

Makine öğrenmesi ve metin madenciliği uygulamalarında büyük önem taşıyan GDA, doküman gibi ayrık verileri modellemek ve bu dokümanların içerdiği konuları belirlemek için kullanılan temel ve popüler bir konu modelleme yöntemidir (Blei vd., 2003). GDA, bu tür verileri analiz etmek için kullanılan bir üretici grafiksel modeldir (Ekinci vd., 2017). Burada “üretici grafiksel model” ifadesi gözlemlenen verilerin altında yatan olasılık dağılımını değişkenler arasındaki ilişkileri ve bağımlılıkları temsil etmektedir. Başlıktaki "gizli" kelimesine yüklenen ifade, dokümanı oluşturan gizli konuları keşfederek dokümanın içeriğindeki derin anlamları ve temaları ortaya çıkarmayı amaçlayan süreçtir (Jadhav, 2014). GDA, belgelerin belirli konuların karışımı olarak modellenebileceği varsayımına dayanır ve her konunun belirli kelimelerin olasılık dağılımıyla karakterize edildiğini öne sürer (Blei vd., 2003). GDA'nın esnek yapısı, büyük metin veri kümelerinde derinlemesine tematik analizler yapılmasını sağlamaktadır. GDA için grafiksel model Şekil 5'te verilmiştir.



Şekil 5. GDA için temsili grafiksel model (Blei vd., 2003)

Şekil 5, GDA'nın belge-konu ve konu-kelime dağılımlarını nasıl modellediğini ve bu dağılımların birbirleriyle nasıl ilişkili olduğunu göstermektedir. Her bir belge, belirli konuların bir karışımı olarak (α) ve her bir konu, belirli kelimelerin bir karışımı olarak (β) temsil edilir. GDA, belgelerden (M) bu konuları (z) ve kelime dağılımlarını (w) öğrenerek, her belgenin hangi konuları içerdiğini (θ) ve her konunun hangi kelimelerle (β) temsil edildiğini belirlemektedir. Bu süreç, belge ve kelime dağılımlarını (α ve β gibi

hiperparametreler aracılığıyla) modelleyerek, belgelerin içeriklerini daha iyi anlamamıza yardımcı olmaktadır. GDA için kullanılan parametreler ayrıntılı açıklanmıştır.

1.4.2. GDA Parametreleri

Her belge için, belgedeki her kelimenin ait olabileceği konuların dağılımını tanımlayan bir dirichlet dağılımından rastgele konu atamaları yapılır. Bu, her belgedeki her kelime için rastgele bir konu ataması yapılması ve sonuç olarak her konunun belirli kelimelerle ilişkilendirilmesi anlamına gelir. Her belgedeki her kelime için, kelimenin mevcut konu ataması geçici olarak kaldırılır ve kelimenin her konuya ait olma olasılığı, o konunun belgedeki varlığı ve kelimenin o konudaki genel varlığına bağlı olarak hesaplanır. Daha sonra, bu olasılıklara dayanarak yeni bir konu ataması yapılır. Parametre kısaltmaları ve açıklamaları şu şekildedir:

- α : Bu, belgelerin konu dağılımları üzerine bir önsel (prior) dirichlet dağılımının parametresidir. α değeri, belgeler arasındaki konu çeşitliliğini ve her bir belgenin konu dağılımının ne kadar homojen olacağını etkiler. Düşük bir α değeri, belgelerin daha az sayıda konuya odaklanmasına neden olurken, yüksek bir α değeri belgelerin daha çok konuyu kapsamasına yol açmaktadır.
- β : Bu, konuların kelime dağılımları üzerine bir önsel dirichlet dağılımının parametresidir. β değeri, konuların kelime çeşitliliğini ve her bir konunun kelime dağılımının ne kadar homojen olacağını etkiler. Düşük bir β değeri, konuların daha az sayıda kelimeye odaklanmasına neden olurken, yüksek bir β değeri konuların daha çok kelimeyi kapsamasına yol açmaktadır.
- θ_d : Bu, d belgesinin konu dağılımını temsil eder ve dirichlet dağılımı kullanılarak α parametresine bağlı olarak örneklenir. θ_d her bir belgenin hangi konuları ne kadar yoğunlukta içerdiğini göstermektedir.
- $z_{d,n}$: Bu, d belgesindeki n 'inci kelimenin atandığı konuyu gösterir. $z_{d,n}$ değeri, θ_d konu dağılımına bağlı olarak çok nominal bir dağılımdan örneklenir ve belirli bir kelimenin hangi konuya ait olduğunu belirtmektedir.
- $w_{d,n}$: Bu, d belgesindeki n 'inci kelimenin kendisidir. $w_{d,n}$ değeri, ilgili $z_{d,n}$ konusunun kelime dağılımına (β parametresine bağlı olarak) ve seçilen konuya ($z_{d,n}$) bağlı olarak çok nominal bir dağılımdan örneklenmektedir.

- M: Toplam doküman sayısıdır.
- N: m. dokümanın boyutudur.

Bu değişkenler, GDA'nın grafiksel modelinin temel yapı taşlarını oluşturur ve algoritmanın belgeler içindeki gizli konu yapılarını nasıl modellediğini anlamamızı sağlar. LDA, bu değişkenleri kullanarak metin veri kümelerindeki tematik yapıları keşfeder ve belgelerin konu dağılımlarını ve konuların kelime dağılımlarını öğrenir.

1.4.3. GDA Kullanım Alanları

Metin sınıflandırma ve gruplandırma alanında; GDA, belgeleri benzer temaları paylaşan gruplara otomatik olarak sınıflandırmak için kullanılmaktadır. Haber makaleleri, akademik makaleler, blog yazıları gibi farklı metin kaynaklarını tematik olarak düzenlemeyi ve gruplamayı kolaylaştırmaktadır. Bilgi keşfi ve içerik özeti alanında; büyük metin koleksiyonlarında önemli temaları ve konuları keşfetmek için kullanılmaktadır. GDA, belgelerin ana fikirlerini ve konularını özetleyerek, kullanıcıların büyük veri kümeleri içinde hızlıca gezinmesine ve ilgilendikleri bilgilere ulaşmasına yardımcı olmaktadır. Öneri sistemleri alanında; kullanıcıların ilgi alanlarına ve geçmiş okuma davranışlarına dayanarak kişiselleştirilmiş içerik önerileri yapmak için kullanılmaktadır. GDA, kullanıcıların ilgilendiği konuları modelleyebilmekte ve bu bilgiyi benzer içerikleri önermek için kullanabilmektedir. Trend analizi ve pazar araştırması alanında; zaman içindeki konu trendlerini ve değişimlerini analiz etmek için kullanılmaktadır. GDA, sosyal medya gönderileri, müşteri yorumları veya haber makaleleri gibi kaynaklardan elde edilen verilerdeki tematik değişimleri izleyerek, toplumsal eğilimler, marka algısı ve pazar dinamikleri hakkında iç görüler sunabilmektedir. DDİ uygulamaları alanında; sentiment analizi, otomatik etiketleme ve metin tabanlı veri madenciliği gibi çeşitli DDİ uygulamalarında ön işleme adımı olarak kullanılmaktadır. GDA, metinlerdeki ana konuları belirleyerek bu tür uygulamaların daha etkili bir şekilde gerçekleştirilmesine yardımcı olmaktadır. Akademik araştırma ve eğitim alanında; akademik literatürdeki araştırma trendlerini ve disiplinler arası ilişkileri analiz etmek için kullanılmaktadır. GDA, büyük akademik veri tabanlarında disiplinler, konular ve araştırma alanları arasındaki ilişkileri keşfetmek için kullanılabilir.

1.5. İlişkili Çalışmalar

Bu bölümde, çalışmanın odak noktasını oluşturan DevOps kavramı ile ilişkili literatür incelenmiştir. İlk olarak, DevOps'a dair çalışmalar ele alınmıştır. Ardından, SO soru-cevap platformu hakkında yapılmış olan araştırmalar gözden geçirilmiştir. Son bölümde ise, özellikle SO platformundan elde edilen verilerin kullanıldığı çalışmalar detaylandırılmıştır. Bu üç kategorize edilmiş bölüm, mevcut literatürün kapsamlı bir özetini sunmayı amaçlamaktadır.

1.5.1. DevOps ile İlgili Çalışmalar

DevOps, yazılım geliştirme (Dev) ve bilgi teknolojileri operasyonları (Ops) arasındaki iş birliğini ve iletişimi artırarak yazılım teslimat süreçlerini hızlandırmayı, sürekli entegrasyon ve sürekli dağıtımı desteklemeyi amaçlayan bir kültür ve uygulama setidir (Bass vd., 2015). Literatürde DevOps konusu ile ilgili çeşitli çalışmalar mevcuttur. Bu çalışmalardan örnekler şu şekildedir.

Erich, Amrit ve Daneva, (2017) tarafından hazırlanan raporda, DevOps kavramının literatürdeki durumunu detaylı bir şekilde incelemiştir. Çalışmada, DevOps'ın tanımı, önemi, uygulama alanları ve bu kavramın yazılım geliştirme süreçlerine etkileri üzerine yayınlanmış akademik ve endüstriyel çalışmalar gözden geçirilmiştir. Araştırma, DevOps uygulamalarının organizasyonel performans üzerindeki olumlu etkilerini, sürekli entegrasyon, sürekli teslimat ve otomasyon gibi uygulama prensiplerinin önemini vurgulayarak, DevOps'un disiplinler arası bir yaklaşım olarak nasıl ortaya çıktığını ve evrildiğini ortaya koymuştur. Sonuç olarak, bu rapor, DevOps'un yazılım geliştirme ve işletim süreçlerini entegre etmede kritik bir rol oynadığını ve bu entegrasyonun, hız, verimlilik ve kalite gibi alanlarda önemli iyileştirmeler sağladığını belgelemiştir (Erich vd., 2017). Bir diğer çalışmada araştırmacılar, DevOps uygulamalarının başarılı olmasında kritik rol oynayan faktörleri derinlemesine incelemiş, DevOps'un etkin uygulanabilmesi için gerekli olan başarı faktörlerini belirlemek amacıyla, mevcut akademik yazınları kapsamlı bir şekilde gözden geçirmişlerdir. Çalışma, organizasyonel kültür, işbirliği, otomasyon, sürekli öğrenme ve sürekli iyileştirme gibi faktörlerin DevOps uygulamalarının başarısında merkezi bir öneme sahip olduğunu ortaya koymuştur. Sonuç olarak, bu inceleme, DevOps'un başarılı implementasyonu için kritik faktörlerin anlaşılmasına katkıda bulunarak, organizasyonların bu yaklaşımı daha etkili bir şekilde benimsemelerine ve uygulamalarına yardımcı olmayı

amaçlamaktadır (Azad, ve Hyrynsalmi, 2023). Diğer bir çalışmada ise, DevOps konusunda eğitim ve endüstri arasındaki potansiyel uyumsuzlukları ve boşlukları ele almaktadır. Araştırmacılar, DevOps becerilerinin ve bilgisinin akademik kurumlar tarafından nasıl öğretildiği ile endüstrinin bu alandaki beklenti ve gereksinimleri arasındaki ilişkiyi incelemiştir. Çalışma, eğitim programlarının güncel endüstri ihtiyaçlarını yansıtmada yetersiz kaldığını ve bu durumun mezunların iş piyasasına girişinde zorluklar yaratabileceğini ortaya koymuştur. Sonuç olarak, bu çalışma, eğitim kurumlarının ve endüstrinin daha yakın işbirliği içinde çalışması gerektiğini ve DevOps eğitiminin sürekli olarak endüstrinin değişen ihtiyaçlarına uyum sağlaması gerektiğini vurgulamaktadır (Sánchez-Cifo vd., 2023).

1.5.2. Stack Overflow Soru-Cevap Sitesi ile İlgili Çalışmalar

SO, yazılım geliştiricilerin karşılaştıkları sorunlara çözüm bulmalarına yardımcı olan en büyük soru-yanıt platformlarından biridir (Barua vd., 2014). SO ile ilgili olan bir çalışmada, SO platformunda sorulan soruların kalitesinin önceden tahmin edilmesine odaklanılmıştır. Araştırmacılar, makine öğrenimi tekniklerini kullanarak soruların alacağı yanıtların sayısı, aldığı oylar ve sorunun çözülüp çözülmediği gibi ölçütler temelinde kalitesini tahmin etmeye çalışmışlardır. Çalışma, soruların kalitesini etkileyen çeşitli özellikleri incelemiş ve bu özelliklerin tahmin modelinde nasıl kullanılabileceğini göstermiştir. Sonuç olarak, bu çalışma, SO gibi çevrimiçi soru-cevap platformlarının kalitesini artırmak için otomatik araçların geliştirilmesine katkı sağlayacak önemli bulgular sunmuştur (Baltadzhieva ve Chrupała, 2015). Bir diğer çalışmada ise, SO platformundan elde edilen bilgiler ile kod yeniden kullanımının Jupyter Notebook üzerindeki etkilerini keşfetmek amaçlanmıştır. Araştırmacılar, Jupyter Notebook kullanıcılarının SO'dan kod parçacıklarını nasıl aldıklarını ve bu kodların yeniden kullanımının kalite, performans ve güvenlik üzerindeki etkilerini analiz etmişlerdir. Çalışma, SO'dan alınan kodların genellikle doğrudan kopyalandığını ve bu durumun bazen hatalı veya verimsiz kod kullanımına yol açabildiğini ortaya koymuştur. Ayrıca, kod yeniden kullanımının, özellikle veri bilimi ve makine öğrenimi projelerinde, geliştirme sürecini hızlandırmanın yanı sıra belirli riskleri de beraberinde getirdiğini vurgulamaktadır. Bu keşifsel çalışma, kodun yeniden kullanımı konusunda farkındalığı artırırken, geliştiricilerin ve araştırmacıların dikkat etmesi gereken önemli noktaları belirlemiştir (Yang vd., 2023). Başka bir çalışmada araştırmacılar, SO

platformundaki etiket tahmin görevi için farklı metin temsili yöntemlerinin değerlendirilmesine odaklanmıştır. Araştırmacılar, soruların içeriğine en uygun etiketleri otomatik olarak tahmin etmek amacıyla çeşitli metin temsili tekniklerini karşılaştırmış ve test etmişlerdir. Çalışma, kelime gömme, TF-IDF (Terim Frekansı-Terim Nadirlik Frekansı) gibi geleneksel yöntemlerden derin öğrenme tabanlı yaklaşımlara kadar geniş bir yelpazedeki yöntemlerin performansını analiz etmiş ve bu metodolojilerin etiket tahmin başarımı üzerindeki etkilerini incelemiştir. Sonuç olarak, bu çalışma, SO gibi teknik forumlarda etiketleme sisteminin otomasyonu ve optimizasyonu için kullanılabilecek metin temsili yöntemlerinin etkinliğini ve sınırlılıklarını ortaya koymuştur (Skenderi vd., 2023).

1.5.3. Konu Modelleme ve Yazılım Geliştirme Verileri Üzerindeki Uygulama ile İlgili Çalışmalar

Konu modelleme, büyük metin koleksiyonlarından gizli temaları çıkarmak için kullanılan istatistiksel bir modelleme aracıdır. GDA, metin koleksiyonlarından konuları keşfetmek için yaygın olarak kullanılan bir yöntemdir (Blei vd., 2003). Yazılım mühendisliği alanında, Gurcan ve arkadaşları (2021), metin madenciliği ve doğal dil işleme tekniklerinin, yazılım geliştirme ile ilgili forumlardan elde edilen verilerin analizinde nasıl kullanılabileceğini göstermişlerdir. Joorabchi ve arkadaşları (2016), yapmış oldukları çalışmada, SO üzerinde yer alan soru ve cevapları analiz ederek bilgisayar bilimi öğrencilerinin karşılaştığı zorlukları ve konuları keşfetmişlerdir. Bu çalışma, öğrencilerin hangi konularda daha fazla zorlandığını belirleyerek eğitim materyallerinin ve öğretim yöntemlerinin geliştirilmesine katkı sağlamayı amaçlamaktadır. Başka bir çalışmada araştırmacılar, SO'da yazılım geliştirme ile ilgili paylaşılan bilgileri inceleyerek, bu platformun yazılım geliştiriciler için nasıl bir bilgi kaynağı olduğunu ve yazılım geliştirme sürecine nasıl katkı sağladığını araştırmışlardır. Bu tür çalışmalar, soru-cevap platformlarının tasarımı ve kullanımı konusunda iç görüşler sunarken, yazılım geliştiricilerin bu forumları daha etkin kullanmalarına yardımcı olabilecek bilgiler sağlamaktadır (Ahmad vd., 2018). Bir diğer çalışmada ise Rosen ve Shihab (2016), mobil geliştiricilerin SO platformunda ne tür sorular sorduğunu büyük ölçekli bir analizle incelemiştir. Araştırmacılar, mobil geliştirme ile ilgili soruları, konuları, zorlukları ve eğilimleri belirlemek için SO'daki veri setlerini derinlemesine analiz etmişlerdir. Çalışma, mobil geliştiricilerin en çok karşılaştıkları sorunların UI tasarımı, platform özgü uygulama geliştirme ve performans optimizasyonu gibi konular olduğunu ortaya koymuştur. Ayrıca,

mobil geliştirme alanında belirli platformlara (iOS, Android vb.) özgü zorlukların yanı sıra, çapraz platform geliştirme ile ilgili soruların da sıkça sorulduğunu belgelemiştir. Bu çalışma, mobil geliştirme topluluğunun ihtiyaçlarını ve zorluklarını daha iyi anlamak için önemli bir kaynak sağlayarak, eğitim materyalleri, araçlar ve platformların geliştirilmesine yönelik iç görüler sunmaktadır. Bir başka çalışma, soru-cevap arşivlerinden yazılım geliştirme yöntemleri hakkında iç görüler elde etmeye odaklanmaktadır. Araştırmacılar, çeşitli soru-cevap platformlarında bulunan büyük veri kümelerini analiz ederek yazılım geliştirme süreçleri, metodolojileri ve karşılaşılan zorluklar hakkında kapsamlı incelemeler yapmışlardır. Bu çalışma, agile, devops ve waterfall gibi farklı yazılım geliştirme yaklaşımlarının uygulanış şekillerini, popülerliğini ve bu yöntemlerle ilişkili problem alanlarını ortaya koymuştur. Bu analiz, yazılım geliştirme topluluğunun mevcut eğilimlerini, tercih edilen teknikleri ve karşılaşılan engelleri anlamak açısından önemli iç görüler sunmaktadır (Khan vd., 2023). Ülkemizde de SO üzerinden alınan veriler ile GDA kullanılarak yapılmış olan çalışmalar mevcuttur. Örneğin Gürcan ve Özyurt (2021), SO platformundaki gönderilerdeki trend konuları kelime frekans analizi yöntemi ile tespit etmeye odaklanmıştır. Araştırmacılar, belirli bir zaman aralığında SO’da yer alan gönderileri analiz ederek, teknoloji ve yazılım geliştirme alanındaki güncel trendleri ve popüler konuları ortaya çıkarmayı amaçlamışlardır. Çalışma, en sık kullanılan terimlerin ve konuların belirlenmesi yoluyla, teknoloji alanındaki değişimleri ve gelişimleri takip etmede önemli bir araç sunmuştur. Sonuçlar, belirli programlama dilleri, kütüphaneler, çerçeveler ve geliştirme teknikleri etrafında yoğunlaşan tartışmaların popülerliğini göstererek, yazılım geliştirme topluluğunun ilgi alanlarının ve ihtiyaçlarının anlaşılmasına katkıda bulunmuştur. Bu çalışma, eğitimcilerin, araştırmacıların ve geliştiricilerin, sektördeki güncel trendleri izlemeleri ve buna uygun stratejiler geliştirmeleri için değerli bir kaynak oluşturmaktadır. Diğer bir çalışma, yazılım odaklı soru-cevap sitelerinde bulunan veriler üzerine metin madenciliği tekniklerinin uygulanmasını konu almaktadır. Araştırma, bu platformlarda yer alan zengin metin verilerinden bilgi çıkarma ve anlamlandırma süreçlerini incelerken, yazılım geliştirme ile ilgili sorunlar, çözümler, trendler ve popüler konular hakkında derinlemesine analizler sunmaktadır. Şahin, algoritmalara dayalı metin analizi yöntemlerini kullanarak, yazılım geliştiricilerin karşılaştıkları yaygın sorunları, en çok tartışılan konuları ve teknolojik eğilimleri belirlemeyi amaçlamıştır. Çalışma, metin madenciliği uygulamalarının, yazılım geliştirme topluluğunun ihtiyaçlarını ve zorluklarını daha iyi anlamak için nasıl kullanılabileceğini göstermektedir (Şahin, 2022).

1.6. Çalışmanın Amacı, Gerekçesi ve Önemi

Yazılım geliştirme dünyası, hızla evrimleşen teknolojiler, artan karmaşıklık ve değişen iş ihtiyaçları ile karşı karşıya bulunmaktadır. Bu bağlamda, yazılım geliştirme ve operasyonların birleştirilmesini hedefleyen DevOps yaklaşımı, günümüzde büyük önem taşımaktadır. DevOps, sürekli entegrasyon, sürekli teslimat, otomasyon, güvenlik ve iş birliği gibi prensipleri benimseyerek, daha hızlı ve verimli bir yazılım geliştirme süreci vadetmektedir. Bu tez çalışması, DevOps'un güncel trendlerini, karşılaşılan zorlukları ve gelecekteki potansiyel gelişim alanlarını anlamak amacıyla, dünyanın en büyük programlama tabanlı soru-cevap platformlarından biri olan SO platformunun sunduğu verileri analiz etmektedir. SO, yazılım geliştiricilerinin bilgi paylaşımı, sorun çözümü ve deneyimlerini paylaşması için vazgeçilmez bir platform olarak kabul edilmektedir. Bu platformdaki “devops” etiketiyle işaretlenmiş gönderiler, sektördeki tartışmaları, trendleri ve topluluğun ilgi alanlarını yansıtmaktadır. Çalışma, bu zengin bilgi kaynağını iki farklı yaklaşımla inceleyerek, DevOps alanındaki trendleri, zorlukları ve gelecekteki potansiyel gelişim alanlarını daha iyi anlamak için kapsamlı bir analiz sunmaktadır.

1.6.1. Etiket Analizi

SO platformundaki gönderilere eklenen etiketler, o gönderinin konusu hakkında önemli ipuçları vermektedir. Bu çalışmada, “devops” etiketiyle işaretlenmiş gönderilerdeki etiketlerin sıklığı, zaman içindeki değişimleri ve birlikteliklerinin analizi gerçekleştirilmektedir. Bu analiz, DevOps alanında en popüler araçları, teknolojileri ve uygulamaları, sektörün hangi yönleri daha fazla ilgi gösterdiğini ve hangi alanlarda kullanımın azaldığını ortaya koymaktadır.

1.6.2. Konu Modelleme

Konu modelleme, bir veri kümesindeki gizli tematik yapıları ortaya çıkarmaya yönelik bir istatistiksel yöntemdir. Bu çalışmada, GDA algoritması kullanarak, “devops” etiketiyle işaretlenmiş gönderilerin altında yatan konular belirlenmektedir. Bu analiz, DevOps topluluğunun en çok hangi konuları tartıştığını, bu konuların zaman içinde nasıl değiştiğini ve gelecekteki potansiyel trendleri ortaya koymaktadır.

Bu iki analiz yönteminin birleştirilmesi, SO platformundaki “devops” gönderilerinin kapsamlı bir şekilde incelenmesini sağlayarak, şu gibi önemli sonuçlara ulaşılmasını hedeflemektedir.

1.6.3. DevOps Uygulamalarındaki Trendlerin Belirlenmesi

SO platformu, DevOps alanında kullanılan araçlar, teknolojiler ve uygulamalar konusunda zengin bir veri havuzu sunmaktadır. Etiket analizi, bu verilerden yola çıkarak, sektördeki en popüler araçları, teknolojileri ve uygulamaları ortaya koyarken, konu modelleme analizi ise, DevOps uygulamalarıyla ilgili en çok tartışılan temaları ve trendleri belirlemektedir.

1.6.4. DevOps Topluluğunun Karşılaştığı Zorlukların Belirlenmesi

Etiket analizi, “devops” etiketi altında sıkça yer alan sorunlara işaret eden etiketleri ortaya koymaktadır. Örneğin, “azure-devops” gibi belirli araçların kullanımına dair sorunlar veya “continuous-integration” ve “deployment” gibi temel konularda karşılaşılan zorluklar belirlenerek, DevOps topluluğunun ihtiyaç duyduğu çözüm ve iyileştirme alanlarına ışık tutulmaktadır.

1.6.5. DevOps Alanındaki Gelecek Trendlerinin Tahmini

Etiket ve konu modelleme analizleri ile elde edilen veriler, DevOps alanındaki gelecek trendleri ve kullanım alanlarını tahmin etmemize olanak sağlayacak, sektördeki öncü firmalar ve akademisyenler için yol haritası çıkarılmasına yardımcı olacaktır.

1.6.6. Eğitim Programlarının Güncellenmesi İçin Temel Bilgiler Sağlanması

DevOps alanında eğitim ve endüstri arasında var olan uyumsuzluk, mezunların iş dünyasına tam olarak entegre olmasını zorlaştırmaktadır. Bu tez çalışması, sektördeki güncel trendleri, popüler araçları ve karşılaşılan zorlukları belirleyerek, eğitim programlarının güncellenmesi ve endüstri ihtiyaçlarına daha uygun hale getirilmesi için temel bir kaynak sağlayacaktır.

Bu tez çalışmasının sonuçları, DevOps topluluğunun ihtiyaçlarını, trendleri ve gelecekteki potansiyel gelişim alanlarını daha iyi anlamamıza katkı sağlayarak, sektörün ve akademinin bu alandaki gelişimine önemli katkılar sunacaktır.

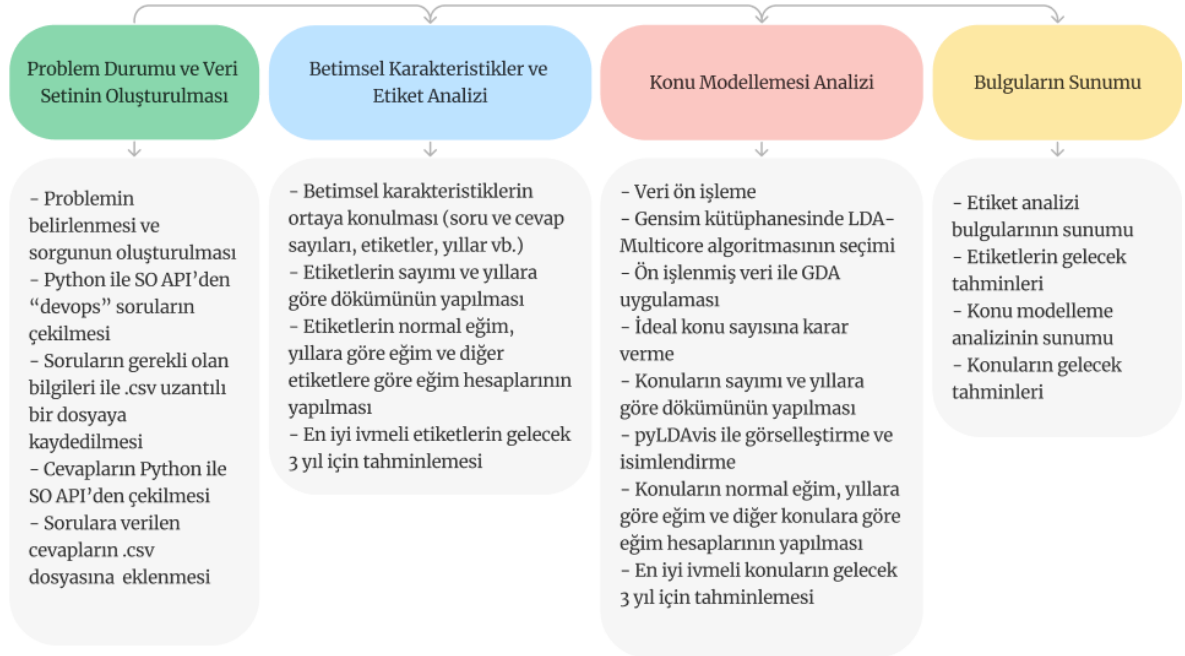


2. YAPILAN ÇALIŞMALAR

Bu bölümde, tezin amacı olan SO platformunda DevOps ile ilgili gönderilerdeki trendlerin etiket ve konu modelleme analizi ile belirlenmesi için takip edilen yöntemsel süreç ve mimari ele alınmaktadır. Takip eden alt bölümlerde, veri setinin oluşturulması, etiket analizleri ve konu modelleme analizleri olmak üzere üç ana aşama detaylı olarak incelenmektedir. Çalışmanın temel mimarisini oluşturan bu aşamalar, hem veri toplama ve işleme yöntemlerini hem de etiket ve konu modelleme analizlerinin gerçekleştirilme sürecini adım adım açıklamaktadır.

2.1. Çalışmanın Ana Mimarisi

Çalışma dört temel esas üzerine kurulmuştur. Bu dört aşama, “Problem Durumu ve Veri Setinin Oluşturulması”, “Betimsel Karakteristikler ve Etiket Analizi”, “Konu Modellemesi Analizi” ve “Bulguların Sunumu” şeklinde isimlendirilmiştir. Şekil 6’da bu dört aşamalı mimari her bir adımdaki iş ve işlemler ile birlikte gösterilmiştir.



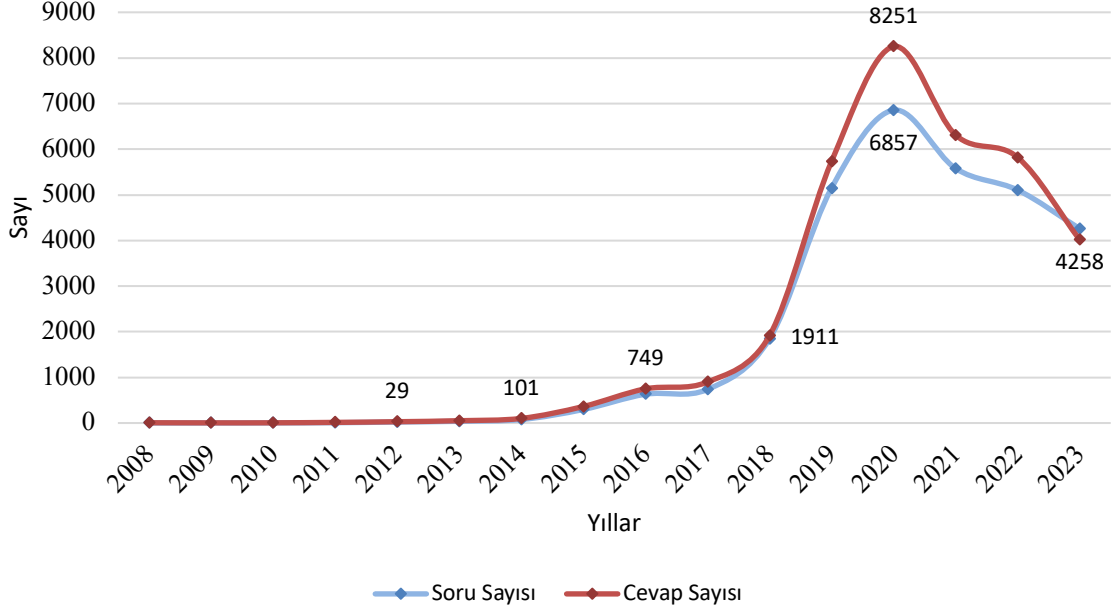
Şekil 6. Çalışmanın ana mimarisi

Şekil 6’da görüldüğü gibi ilk aşama “Problem durumu ve veri setinin oluşturulması” aşamasıdır. Bu aşamada veri setinin oluşturulması sürecinde, 2008 yılından 2023 yıl sonuna kadar SO soru-cevap platformu üzerinde paylaşılan “devops” ile ilgili veriler, Python programlama dili kullanılarak SO API servisi aracılığıyla bir CSV dosyasına kaydedilmiştir. Betimsel analiz kapsamında, bu dosyada yer alan veriler üzerinden soruların cevaplanma durumları kontrol edilmiş ve sorulara verilen cevaplar, mevcut ise, yeni bir sütunda ilgili sorunun satırına işlenmiştir. Her sorunun en az bir, en fazla beş olmak üzere ilgili olduğu zorunlu bir etiketleme durumu vardır. Bu etiketler soruları kategorize etmede kullanıldığı gibi, yıllara göre analizlerde de etkili çıktılar vermektedir. Etiket analizleri çerçevesinde, sorulara ait etiketlerin toplam sayısı ve yıllara göre dağılımı gibi genel bilgiler elde edilmiştir. Etiketlerin bu sayısal değerleri üzerinden normal eğim, yıllara göre eğim ve diğer etiketlere göre eğim hesapları yapılmıştır ve en iyi ivmeli etiketler için 2. dereceden lineer olmayan regresyon formülü hesabı ile gelecek tahmini yapılmıştır. Konu analizleri yapılmadan önce veri ön işleme yapılmıştır. Veri seti birtakım eklerden, kod parçalarından, ip numaralarından, durak kelimelerden arındırılmış, soru başlığı + soru metni + soruya verilen cevaplar şeklinde her bir satır bir dokümanı temsil edecek şekilde temizlenmiştir. Konu analizleri, “devops” ile ilgili soru ve cevaplardan oluşan metinler üzerinde, GDA algoritması kullanılarak yapılmıştır. Son olarak, konu analizlerinin görselleştirilmesi sürecinde, elde edilen sonuçlar çeşitli grafiklerle desteklenerek sunulmuştur. Tüm işlemler çalışmanın ilerleyen bölümlerinde detaylı olarak açıklanmaktadır. Bu süreçlerin tamamı, veri setinin hazırlanmasından başlayarak, analizlerin yapılmasına kadar olan aşamalar ve görselleştirmeler, bu çalışmanın içeriğini kapsamaktadır.

2.2. Veri Setinin Oluşturulması

İlk olarak SO üzerinde 2008 yılından 2023 yılı sonuna kadar paylaşılmış olan “devops” ile ilgili toplamda 30600 satır veri Python programlama dili ile SO API servisi kullanılarak çekilmiş ve .csv formatında kaydedilmiştir. Akabinde bu çekilen sorulara verilen cevapların olup olmadığı kontrol edilmiş (IsAnswered alanı = 1), kontrol sonucunda soruya verilen cevap veya cevaplar var ise tekrar API servisinden bu soruya verilen cevaplar yeni bir sütunda ilgili sorunun satırına işlenmiştir. Cevaplar işlenirken cevaplar birden fazla ise cevaplar arasına “#####” ifadesi eklenerek tek bir “cevaplar” sütunu elde edilmiş ve tabloya işlenmiştir. Metin ve konu modelleme analizlerinde bu “#####” ifadesi çıkarılmıştır. Veri

setinin olduğu .csv dosyasındaki sütunlardaki bilgilerden toplam soru adetleri, yıllara göre soru adetleri, toplam etiket sayısı, yıllara göre etiket sayıları gibi genel bilgiler oluşturulmuştur. Yıllara göre soru ve cevap sayıları Şekil 7’de grafiksel gösterimle verilmiştir.



Şekil 7. Yıllara göre soru ve cevap sayıları

Şekil 7’de görüldüğü gibi, veri seti incelendiğinde ilk kayıtlar 2008 yılında alınmıştır. 2008 yılından 2023 yılı sonuna kadar toplam soru sayısı 30600 adettir. Veri setinde bulunan 2008 yılından 2013 yılı sonuna kadar toplam soru sayısı 72’dir. Her yıl sorulan sorular ve bu soruların sayısı incelendiğinde 2014 yılından önce DevOps ile ilgili olduğu düşünülen soruların çevik yazılım geliştirme ile alakalı olduğu saptanmış ve bu çalışmada esas olarak verinin %99,76’sını oluşturan son 10 yıl (2014-2023) üzerinde tüm analizler yapılmıştır. Yıllara göre eğim ve ortalama büyüme analizlerindeki sonuçların daha anlaşılır olması için son 10 yıl dikkate alınmıştır. Oluşturulan nihai veri setine ilişkin bilgiler Tablo 1’de verilmiştir.

Tablo 1. Veri setine ilişkin bilgiler

Genel Bilgiler	Veri
Toplam soru adedi:	30600
Toplam cevap adedi:	34327
Tekil etiket sayısı:	5068
Toplam etiket kullanım sayısı:	102839
Son 10 yıldaki (2014-2023) soru sayısı	30528 (%99,76)
İlk 500 etiketin kullanım sayısı:	90198 (%87,90)
En çok kullanılan tekil etiket:	azure-devops (n=16921)
En çok kullanılan ikili etiket:	azure-devops & azure-pipelines (n=4890)
En çok kullanılan üçlü etiket:	azure & azure-devops & azure-pipelines (n=1294)
En çok kullanılan dördü etiket:	azure & azure-devops & azure-pipelines & devops (n=117)

2.3. Etiket Analizleri

Veri seti üzerinde ilk olarak etiket analizi yapılmıştır. SO'da etiketler soruyu soran kullanıcılar tarafından eklenmektedir ve bir sorunun en fazla 5 etiketi olabilir. Etiketler, soruyu soran kullanıcının sorunun kategorisini, hangi teknoloji ile ilgili olduğunu, aradığı cevabın hangi konu ile bağlantılı olduğunu belirtmek için kullanılabilir. Bu sebeple etiket analizi soruları ve dolayısıyla soruların bağlı bulunduğu kategorileri analiz etmekte oldukça değerli bilgiler taşımaktadır. Her soruda kullanılan en az 1, en fazla 5 etiket bulunmaktadır. Etiket analizleri yapılırken öncelikle birliktelik analizi yapılmıştır. Bu analiz yapılırken her sorunun taşıdığı etiketler kendi içerisinde alfabetik olarak sıralanmıştır. Bu sıralamadan sonra 2 ve üzerinde etiket sayısına sahip sorular için etiket ikilileri, üçlülere ve dördülere oluşturulmuştur. Frekansların bulunması için bütün bulunan tekli, ikili, üçlü ve dördü etiket kombinasyonları sayma fonksiyonu kullanılmıştır. Bu etiketler için sorunun oluşturulma tarihi bilgisi ile hangi yıla ait olduğu bulunduktan sonra ilgili yılda kaç kez kullanıldığı hesaplanmış ve ayrı ayrı tekli, ikili, üçlü, dördü etiketlerin sıralandığı tablolarda tutulmuştur. Bu şekilde en fazla kullanılan etiket listeleri yapılmıştır.

Etiketlerin yıllar içerisindeki değişimini analiz edebilmek için eğim fonksiyonu kullanılmıştır. Eğim fonksiyonu, bir veri setindeki bir dizi bağımsız değişkenin (x) ve buna karşılık gelen bir dizi bağımlı değişkenin (y) arasındaki doğrusal ilişkiyi hesaplamak için kullanılan bir fonksiyondur. Bu fonksiyon, bu iki değişken arasındaki doğrusal regresyon çizgisinin eğimini hesaplar. Bu çalışmadaki bağımsız değişkenler 2014-2023 yılları arasındaki etiket sayılarıdır. Bağımlı değişkenler ise yıllardır. Doğrusal regresyon çizgisi, x değerlerinin birim artışına karşılık gelen y değerlerinin tahmini artışını ifade eder. Eğim, istatistiksel analizlerde ve veri analizinde önemli bir ölçüdür. Bir veri setindeki değişkenler arasındaki ilişkiyi anlamak ve bu ilişkiyi açıklamak için kullanılır. Eğim, iki değişken arasındaki doğrusal ilişkiyi belirler. Bu ilişki, bağımsız değişkenin bir birim artışının bağımlı değişken üzerindeki etkisini ölçer. Eğim pozitif ise, değişkenler arasında doğrusal bir artış ilişkisi olduğunu gösterir; negatif ise, ters bir ilişki olduğunu gösterir. Eğim, veri setindeki trendleri ve desenleri tanımlamak için kullanılabilir. Kullanılan veri setine bakıldığında zaman serisi özelliği taşımaktadır. Özellikle zaman serisi verilerinde eğimi analiz etmek, gelecekteki trendleri tahmin etmek ve geçmiş performansı anlamak için önemlidir.

Veri setindeki etiket sayılarına bakıldığında örneğin “azure-devops” etiketi 2014-2023 için sırasıyla 1, 4, 10, 31, 809, 3393, 4295, 3284, 2857, 2236 kez toplamda son 10 yılda 16920 soruda kullanılmıştır. Yüzdesel olarak bakıldığında “azure-devops” etiketinin i.yıl / toplam etiket sayısı formülünden hesaplanan yüzdeleri sırasıyla 2014-2023 yılları için %0,01, %0,02, %0,06, %0,18, %4,78, %20,05, %25,38, %19,41, %16,89, %13,22’dir. Bu yüzdelere tüm etiketler için hesaplanmış ve her etiketin yıllara göre eğimi zaman serisindeki eğim hesabı formülüne göre hesaplanmıştır. Eğim hesabı,

$$\text{Eğim} = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad (1)$$

biçiminde tanımlanır. Denklem 1’de n, veri noktalarının sayısıdır. $\sum xy$, x ve y değerlerinin çarpımlarının toplamıdır. $\sum x$ ve $\sum y$ sırasıyla x ve y değerlerinin toplamlarıdır. $\sum x^2$ x değerlerinin karelerinin toplamıdır. Denklem 1’e göre yukarıdaki “azure-devops” etiketinin yüzde değerlerinin eğimi 2.573 olarak hesaplanmıştır. Etiketlerin 2014-2023 yılları kullanım sayısına göre yüzdelik dilimleri hesaplandıktan sonra eğim hesabı yapılarak en çok büyüme kaydeden ve en çok küçülme kaydeden etiketler saptanmıştır. Toplamda 5068 adet tekil etiket, son 10 yıldaki toplam kullanım sayısına göre sıralandığında ilk 500

etiketin kullanım sayısının toplam veri setinin %87,90'ı olduğu görülmüştür. İlk 500 etiket sonrasında gelen etiketlerin büyük bir çoğunluğu 2019 yılından önce kullanıldığı ve son 10 yılda toplam en fazla 16 kez kullanıldığı saptanmıştır. Bu durum bazı yıllarda etiketin kullanım sayısının 0 olup bazı yıllarda ise rakamsal olarak 3,4,5 gibi pozitif değerler olduğunda alınan yüzdelerin çok büyük veya çok küçük olması dolayısıyla eğim hesabında yanıltıcı sonuçlar üretmiştir. Bu sebeple veri setindeki kullanım sayısı olarak en büyük ilk 500 etiket için eğim hesabı yapılmıştır ve en iyi performans gösteren ilk 15 etiket bulunmuştur.

En iyi performans gösteren ilk 15 etiket için gelecek tahmini hesabı yapılmıştır. Bu tahminler için 2. dereceden lineer olmayan regresyon modeli kullanılmıştır. Regresyon modeli Python dili scikit-learn kütüphanesi kullanarak kodlanmıştır. 2. dereceden lineer olmayan regresyon hesabı,

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon \quad (2)$$

biçiminde hesaplanır. Denklem 2'de y , bağımlı değişkeni ifade eder (tahmin edilmek istenen değer). x bağımsız değişkeni temsil eder (yıllar). $\beta_0, \beta_1, \beta_2$ modelin katsayılarıdır; bu katsayılar, modelin verilere en iyi uyum sağlamasını sağlamak için eğitim sürecinde belirlenir. ϵ hata terimidir ve modelin tüm varyansı açıklayamamasından kaynaklanan rastgele sapmayı ifade eder. Ancak ileride, gerçek değerler farklı faktörlere bağlı olarak bu tahminlerden farklı olabilir.

2.4. Konu Modelleme Analizleri

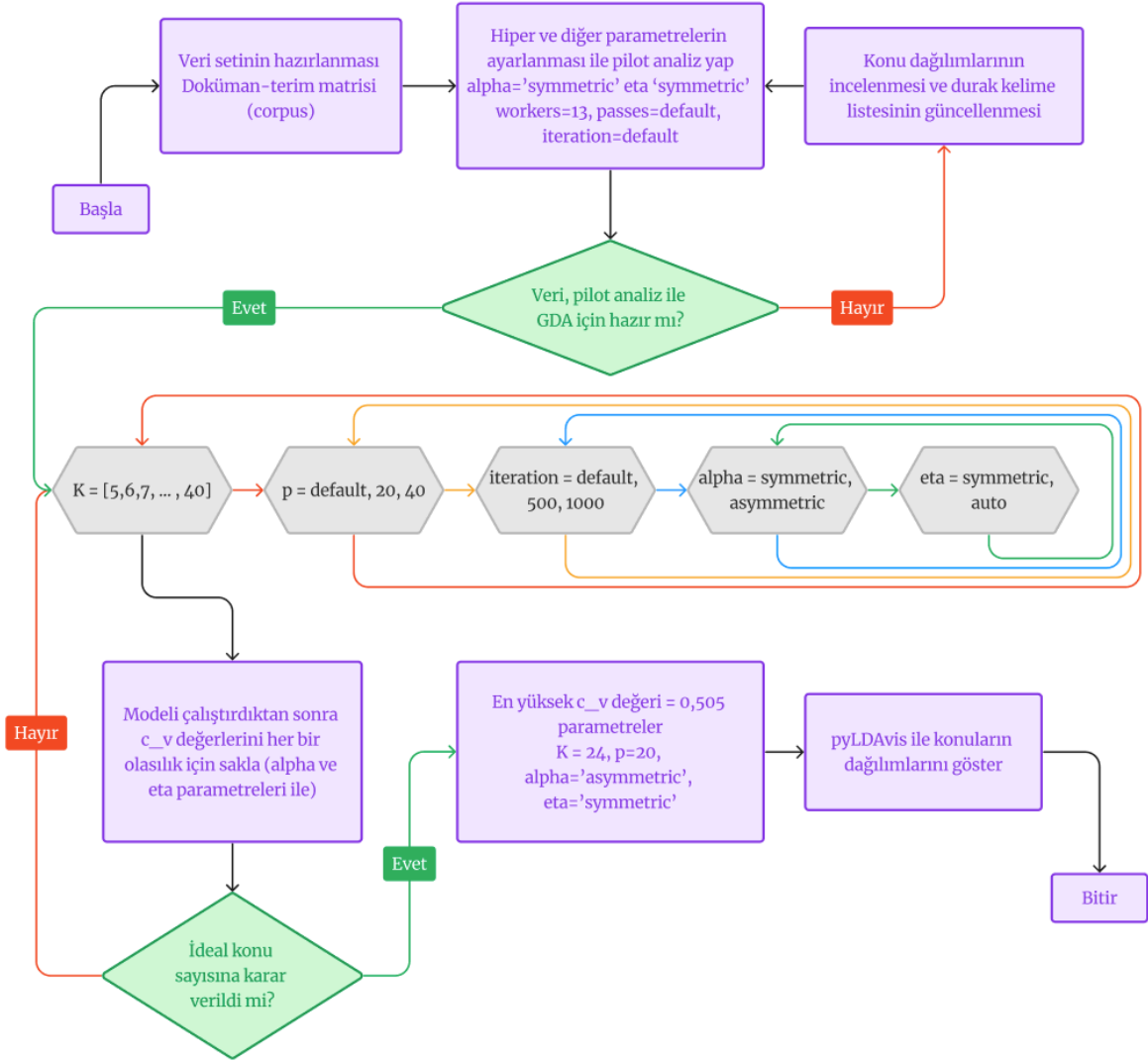
Etiket analizlerinin ardından çalışmanın amacı doğrultusunda ikinci aşamada konu modelleme analizi yapılmıştır. Bu aşamada veri seti üzerinde GDA tabanlı konu modelleme analizi yapılmıştır. Konu modelleme analizi, işlemcisi 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz, 14 çekirdekli ve 16 GB RAM özellikli diz üstü bilgisayarda, Python dili Gensim kütüphanesindeki paralelleştirilmiş GDA olan "LdaMulticore" kullanılarak yapılmıştır (URL-4, 2023). Model eğitimini paralelleştirmek ve hızlandırmak için tüm işlemci çekirdeklerini kullanan bu algoritma, çalışılan bilgisayardaki çekirdek sayısının bir düşüğü ile algoritmayı paralelleştirmektedir. Şekil 8'de GDA tabanlı konu modelleme analizinin adımları verilmiştir.



Şekil 8. Konu modelleme analizi adımları

Şekil 8’de, GDA tabanlı konu modelleme analizinin uygulanma süreci beş ana adımda ayrıntılı olarak açıklanmaktadır. Bu süreç, “Problem durumu” ile başlamakta ve bulguların toplanması ve sunumu ile son bulmaktadır. İlk adım olan “Problem durumu” aşamasında DevOps konusuna çalışılabilmesi için SO soru-cevap sitesinin incelenmesi ile başlanmaktadır ve sırasıyla “SO’dan veri alma işlemlerinin dokümantasyonu” ve “DevOps ile ilgili soruların ve cevapların nasıl SO’dan alınacağını tespit etme” ile sonlanmaktadır. “Veri toplama ve korpus oluşturma” aşamasında, veri seti oluşturulmuştur. Bu aşamada ham veriler indirilmiş ve .csv formatında saklanmıştır. Bu korpus, GDA modelinin uygulanacağı veriyi sağlamakta ve analizlerin temelini oluşturmaktadır. Üçüncü adım olan “Veri ön işleme” aşamasında, veriler küçük harflere dönüştürülüp temizlenir. Bu aşama, verilerin tutarlılığını ve analiz edilebilirliğini artırmaktadır. Ardından, kök kelimeler elde edilir ve durak kelimeler çıkarılır. Bu adımlar, verilerin daha anlamlı ve analize uygun hale getirilmesini sağlar. Son olarak, belge-terim matrisi oluşturulur. Bu matris, belgeler ve

terimler arasındaki ilişkileri gösterir ve GDA modelinin uygulanması için gereklidir. Dördüncü adım, “Ampirik analiz ve konuların keşfi” aşamasında, pilot bir GDA analizi gerçekleştirilir. Durak kelime listesi düzenlenir ve iteratif analizler yapılır. Bu işlemler modelin doğruluğunu artırmak için gereklidir. Ayrıca, GDA algoritmasının hiperparametreleri ayarlanır ve ideal konu sayısı belirlenir. Bu adım, modelin en iyi şekilde çalışmasını sağlamak ve anlamlı konuların keşfedilmesini mümkün kılmaktadır. Son adım, “Bulguların toplanması ve sunumu” aşamasında, konular pyLDavis ile konu numaraları ile görselleştirilmiştir. Sonrasında konular, ilk 15 terimine göre isimlendirilir. Bu isimlendirme, konuların daha anlaşılır ve anlamlı hale getirilmesini sağlamaktadır. Son olarak, konu trendleri görsel ve tablo halinde sunulmuştur. Bu işlemler, elde edilen bulguların etkili bir şekilde iletilmesini sağlamaktadır ve araştırmanın sonuçlarını net bir şekilde ortaya koymaktadır. Bu beş adım, GDA modelinin sistematik ve etkili bir şekilde uygulanmasını sağlamaktadır. Her adım, bir sonraki adım için gerekli temeli oluşturmakta ve araştırmanın bütünlüğünü korumaktadır. Bu süreç, çalışmanın, verileri anlamlı konulara ayırmasını ve bu konuları güvenilir, anlamlı ve etkili bir şekilde sunmasını sağlamaktadır. Şekil 9’da GDA tabanlı konu modelleme algoritmasının gerçekleştirilmesinin akış diyagramı verilmiştir.



Şekil 9. GDA algoritmasının akış şeması

Şekil 9’da, başlangıçta hazır bulunan doküman terim matrisi veri seti ile 13 çekirdek işlemci kullanılarak çalışan GDA modeli için hiperparametreleri ön tanımlı olarak belirlendi. Bu parametreler arasında alpha (‘symmetric’ ve ‘asymmetric’), eta (‘symmetric’ ve ‘auto’), iterasyon sayısı (default, 500, 1000), ve passes (default, 20, 40) olarak ayarlanmıştır. Ardından, pilot analiz yaparak verinin GDA için hazır olup olmadığı kontrol edilmiştir. Eğer veri hazır değilse, konu dağılımlarını inceleyip durak kelime listesi güncellenmiştir. Veri hazır olduğunda, GDA modeli konu sayısı (K) 5 konudan 40 konu aralığında tüm parametre ihtimalleri ile eğitilmiştir. Eğitim sonrasında, ideal konu sayısına karar verilmesi gereklidir. İdeal konu sayısına karar verirken c_v tutarlılık skoru önemli bir parametredir. Yüksek c_v değeri, daha tutarlı ve iyi tanımlanmış konuları gösterirken, düşük c_v değeri, konu dağılımında daha fazla gürültü veya belirsizlik olduğunu gösterir (Rüdiger vd., 2022). Konu

sayısına karar verilirken c_v tutarlılık değerinin yanında, uzman gözü ile konuların dağılımının korpusu temsil etme düzeyi de incelenmektedir. Bu bağlamda iterasyonel süreçte her bir model için tüm c_v değerleri ilgili konu sayısı karşılığına parametre değerleri ile birlikte kayıt altına alınmıştır. En yüksek değerli c_v değeri olan 0.505'e sahip parametrelerle ($K = 24$, p alpha = 'asymmetric', eta = 'symmetric') 24 konulu modelin ideal dağılım gösterdiği görülmüştür. GDA tabanlı konu modelleme analizi sonucu elde edilen 24 konulu model PyLDAvis kütüphanesi ile görselleştirilmiştir. PyLDAvis, GDA model sonuçlarını görselleştirmek için kullanılan bir Python kütüphanesidir. PyLDAvis, GDA modelin sonuçlarını etkileşimli bir görselleştirme aracılığıyla sunarak, modelin konularını ve bunların belgelerle ilişkisini anlamayı kolaylaştırmakta ve konuları isimlendirmekte yardımcı olmaktadır (Tran K., 2023). PyLDAvis yardımı ile görselleştirilen 24 konu isimlendirilmiştir. Her konu, konu isimlendirmesi, konu terimleri ve konunun toplamdaki yüzdesi verilerek genel anlamda en hacimli konuların neler olduğu gösterilmiştir. Sonrasında yıllara göre hangi konudan kaç adet tespit edildiği verilmiştir. Her konunun toplam konu sayısına oranları bulunarak hacimleri hesaplanmıştır. Konuların yıllara göre kullanım sayılarına bakarak eğim formülü uygulanmış ve en iyi ivmeye sahip konu isminden başlanarak liste halinde verilmiştir. Tüm konuların son 10 yıldaki kullanım sayılarına göre bulunan ivme değerlerinin tamamının pozitif olduğu görülmüştür. Konuların diğer yıllara göre son 10 yıldaki kendi içindeki yüzdeleri hesaplanmıştır. Bu yüzdelerle uygulanan eğim formülü çıktıları tüm konularda pozitif tespit edilmiştir. Sonrasında her bir yılı ve o yılın içindeki konuların kullanım sayılarını birbirine oranladığımızda konuların diğer konulara kıyasla o yılki yüzdeleri hesaplanmıştır. Bu şekilde her yıl için hesaplanan yüzde değerleri her bir konunun diğer konulara kıyasla ağırlığını ortaya koymuştur. Bu şekilde bulunan yüzde değerlerine uygulanan eğim formülü çıktılarına bakıldığında bazı konuların pozitif, bazı konuların ise negatif ivmelendiği görülmüştür. Bu sebepten ötürü bulunan sonuçlar anlamlıdır. Özellikle pozitif ivmelenen etiketlere gelecek tahminlemesi yapılmıştır. Bu tahminleme daha önceden belirtildiği üzere Python scikit-learn kütüphanesindeki 2. dereceden lineer olmayan regresyon formülü ile hesaplanmış değerler ile gösterilmektedir.

3. BULGULAR VE TARTIřMA

Bu bölümde mevcut çalışma kapsamında elde edilen bulgulara yer verilmiştir. Bulgular iki alt başlık altında sunulmuştur. İlk başlık altında etiket analizi bulgularına yer verilmişken ikinci alt başlık altında konu modelleme analizi bulgularına yer verilmiştir.

3.1. Etiket Analizleri Bulguları ve Tartışma

Etiket frekans analizine göre son 10 yılda en çok kullanılan ilk 50 etiketin isimleri, kullanım sayıları (n) ve yüzdeleri (f) Tablo 2’de gösterilmektedir. Bu etiketler çalışmanın toplam %63,06’sını oluşturmaktadır ve oldukça büyük bir kapsama sahiptir.

Tablo 2. En çok kullanılan ilk 50 etiket, kullanım sayıları (n) ve yüzdeleri (f)

Sıra	Etiket	n	f	Sıra	Etiket	n	f
1	azure-devops	16920	%16,49	26	.net-core	443	%0,43
2	devops	7875	%7,67	27	gitlab	438	%0,43
3	azure	6194	%6,04	28	linux	430	%0,42
4	azure-pipelines	5777	%5,63	29	java	421	%0,41
5	docker	1971	%1,92	30	nuget	397	%0,39
6	git	1645	%1,60	31	docker-compose	384	%0,37
7	azure-devops-rest-api	1306	%1,27	32	pipeline	381	%0,37
8	powershell	1195	%1,16	33	cicd	363	%0,35
9	jenkins	1095	%1,07	34	azure-functions	357	%0,35
10	amazon-web-services	1088	%1,06	35	continuous-deployment	354	%0,34
11	kubernetes	1084	%1,06	36	jenkins-pipeline	350	%0,34
12	c#	1040	%1,01	37	azure-devops-extensions	350	%0,34
13	continuous-integration	1005	%0,98	38	deployment	340	%0,33
14	yaml	976	%0,95	39	azure-devops-server-2019	334	%0,33
15	azure-pipelines-release-pipeline	860	%0,84	40	maven	323	%0,31
16	python	843	%0,82	41	azure-pipelines-build-task	304	%0,30
17	tfs	825	%0,80	42	gitlab-ci	303	%0,30
18	azure-pipelines-yaml	807	%0,79	43	azure-active-directory	302	%0,29
19	terraform	704	%0,69	44	asp.net-core	301	%0,29
20	node.js	538	%0,52	45	angular	291	%0,28
21	github	534	%0,52	46	bash	289	%0,28
22	visual-studio	483	%0,47	47	nginx	285	%0,28
23	.net	466	%0,45	48	npm	279	%0,27
24	ansible	452	%0,44	49	reactjs	278	%0,27
25	azure-web-app-service	447	%0,44	50	javascript	277	%0,27
Toplam						64704	%63,06

Tablo 2’de görüldüğü gibi üst sıralarda, “azure-devops” (f= %16,49) ve “devops” (f=%7,67) gibi otomasyon ve iş akışı yönetimine odaklanan etiketler görülmektedir. Bu etiketler, DevOps kültürünün ve bu alandaki araçların yazılım geliştirme süreçlerinde önemli bir yer tuttuğunu göstermektedir. Bulut hizmetleri ve altyapı yönetimi de “azure”, “amazon-web-services” ve “kubernetes” gibi etiketlerle temsil edilmektedir. Bu etiketlerin yüksek kullanım sayıları, bulut bilişimin ve mikro hizmetlerin popülerliğinin bir göstergesi olabilir. Programlama dilleri ve çerçeveler, “c#”, “python”, “.net”, ve “java” gibi etiketlerle temsil edilirken, bunların kullanımı genel kullanımın yaklaşık yüzde 1’ini oluşturmaktadır. Bu da

çeşitli dillerin ve teknolojilerin yazılım geliştirme ekosisteminde sağlam bir yer edindiğini göstermektedir. Sürekli entegrasyon ve dağıtım (CI/CD) pratikleri “continuous-integration”, “jenkins”, ve “gitlab-ci” gibi etiketlerle vurgulanmakta ve bu alanın geliştirme süreçlerindeki önemini ortaya koymaktadır. Araçlar ve geliştirme ortamları, “visual-studio”, “github”, ve “docker” gibi etiketlerle temsil edilmekte ve kullanım yüzdeleri bu araçların geliştiriciler arasında yaygın olarak kullanıldığını göstermektedir. Frontend ve backend geliştirme ile ilgili kütüphaneler ve çerçeveler ise “angular”, “reactjs”, “asp.net-core” gibi etiketlerle temsil edilerek, modern web geliştirmedeki çeşitliliği ve popüleriteyi yansıtmaktadır. Son olarak, “javascript” etiketinin listeye dahil olması, bu dilin web geliştirmedeki evrensel rolünü ve sürekli var olan popüleritesini vurgulamaktadır. Tablo 2, genel olarak, yazılım geliştirme ekosistemindeki güncel trendleri ve teknolojik yatırım alanlarını göstermektedir. Yüksek kullanım oranlarına sahip etiketler, şirketlerin ve bireysel geliştiricilerin üzerinde yoğunlaştığı alanları belirginleştirmektedir. Veri setinde soruların birlikte kullanılan etiketlerine bakıldığında en çok kullanılan ikili etiketlerden ilk 20 ikili etiket, Tablo 3’te gösterilmektedir.

Tablo 3. En çok kullanılan ilk 20 ikili etiket, sayısı(n) ve yüzdesi(f)

Sıra	Etiket	n	f	Sıra	Etiket	n	f
1	azure-devops & azure-pipelines	4890	%3,36	11	azure-devops & azure-pipelines-release-pipeline	753	%0,52
2	azure & azure-devops	4466	%3,07	12	azure-devops & yaml	737	%0,51
3	azure & azure-pipelines	1542	%1,06	13	amazon-web-services & devops	728	%0,50
4	devops & docker	1082	%0,74	14	azure-devops & azure-pipelines-yaml	691	%0,48
5	azure-devops & azure-devops-rest-api	1048	%0,72	15	devops & kubernetes	662	%0,46
6	azure-devops & devops	1046	%0,72	16	azure-devops & c#	576	%0,40
7	azure-devops & git	1032	%0,71	17	azure-devops & tfs	560	%0,39
8	azure & devops	964	%0,66	18	azure-pipelines & azure-pipelines-yaml	516	%0,36
9	devops & jenkins	796	%0,55	19	azure-devops & continuous-integration	483	%0,33
10	azure-devops & powershell	773	%0,53	20	continuous-integration & devops	451	%0,31
Toplam						23796	%16,37

Tablo 3’te görüldüğü gibi ilk sırada “azure-devops & azure-pipelines” etiketi, %3,36’lık bir kullanım oranıyla yer almaktadır ve bu yüksek oran, yazılım geliştirme

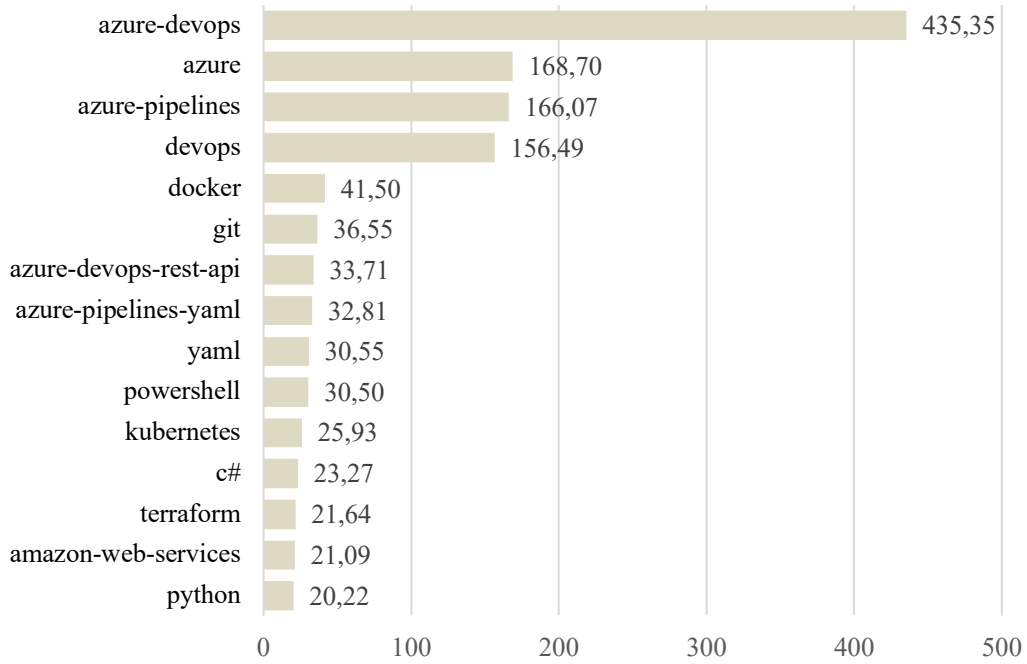
süreçlerinde bu iki hizmetin entegre bir şekilde ne kadar sık kullanıldığını göstermektedir. İkinci sırada yer alan “azure & azure-devops” etiketi de %3,07 kullanım oranıyla, Azure platformunun genel kullanımının ve Azure DevOps servisinin popüleritesini ortaya koymaktadır. DevOps kültürünün ve araçlarının popüleritesi, “devops & docker”, “azure-devops & devops” ve “azure-devops & git” gibi etiket kombinasyonlarında da görülmektedir. Bu kombinasyonlar, modern yazılım geliştirme pratiklerinde sürekli entegrasyon, dağıtım ve versiyon kontrol sistemlerinin önemini vurgulamaktadır. Kubernetes ve Docker gibi konteyner teknolojileri ve orkestrasyon araçları, “devops & kubernetes” ve “azure-devops & azure-pipelines-yaml” etiketleriyle temsil edilmektedir. Bu etiketlerin listeye girmesi, mikro hizmet mimarilerinin ve konteynerizasyonun giderek artan popüleritesini gösteriyor. “azure-devops & powershell”, “azure-devops & c#” ve “azure-devops & tfs” gibi kombinasyonlar, Microsoft teknolojileri ve dillerinin DevOps süreçleri içindeki entegrasyonuna dikkat çekmektedir. Yazılım otomasyonu ve yönetimine dair etiketler dışında, “amazon-web-services & devops” gibi etiketler de Amazon’un bulut hizmetlerinin DevOps ile nasıl bütünleştiğini göstermektedir. Genel olarak Tablo 3, yazılım geliştirme ve operasyonlar dünyasında Azure ve DevOps araçlarının baskın olduğunu, sürekli entegrasyonun ve otomasyonun sektördeki öneminin altını çizmektedir. Ayrıca, modern yazılım geliştirme ekosisteminin çoklu araçlar ve platformlar arasında nasıl bir entegrasyon içinde olduğunu göstermektedir. İkili etiketlerin değerlendirmesinden sonra Tablo 4’te bir soruda birlikte kullanılan etiketlerden en çok kullanılan ilk 20 üçlü etiket verilmiştir. Tablo 4, yazılım geliştirme ve operasyonları alanında sıklıkla birlikte kullanılan ilk 20 üçlü etiket setini, bu etiketlerin kullanım sayılarını (n) ve toplam içindeki yüzdesini (f) göstermektedir.

Tablo 4. En çok kullanılan ilk 20 üçlü etiket, sayısı(n) ve yüzdesi(f)

Sıra	Etiket	n	f
1	azure & azure-devops & azure-pipelines	1294	%1,27
2	azure-devops & azure-pipelines & azure-pipelines-yaml	444	%0,43
3	azure & azure-devops & devops	433	%0,42
4	azure-devops & azure-pipelines & yaml	363	%0,36
5	azure-devops & azure-pipelines & azure-pipelines-release-pipeline	361	%0,35
6	azure-devops & azure-pipelines & devops	298	%0,29
7	azure & azure-devops & powershell	228	%0,22
8	devops & docker & docker-compose	227	%0,22
9	azure & azure-devops & azure-devops-rest-api	222	%0,22
10	azure & azure-devops & git	205	%0,20
11	azure-devops & azure-pipelines & powershell	194	%0,19
12	azure & azure-devops & yaml	191	%0,19
13	devops & jenkins & jenkins-pipeline	189	%0,19
14	azure & azure-devops & azure-web-app-service	181	%0,18
15	azure & azure-devops & azure-pipelines-release-pipeline	176	%0,17
16	azure-devops & azure-pipelines & continuous-integration	175	%0,17
17	azure-devops & azure-pipelines & azure-pipelines-build-task	173	%0,17
18	azure-devops & azure-pipelines & git	155	%0,15
19	azure & azure-pipelines & devops	151	%0,15
20	azure-devops & azure-devops-rest-api & azure-pipelines	149	%0,15
Toplam		5809	%5,69

Tablo 4’te görüldüğü gibi “azure & azure-devops & azure-pipelines” etiket kombinasyonu, %1,27’lik bir kullanım oranı ile en popüler üçlü etiket kombinasyonu olarak elde edilmiştir. Azure bulut platformu, Azure DevOps hizmetleri ve Azure Pipelines sürekli entegrasyon/sürekli dağıtım araçları arasındaki bu bütünleşik kullanım, bulut tabanlı geliştirme ortamlarının ve DevOps pratiklerinin ne kadar iç içe geçtiğini vurgulamaktadır. İkinci sırada yer alan “azure-devops & azure-pipelines & azure-pipelines-yaml” kombinasyonu, YAML bazlı Azure Pipelines konfigürasyon dosyalarının kullanımının da popüler olduğunu göstermektedir. Üçüncü sıradaki “azure & azure-devops & devops” kombinasyonu ise, Azure’ın DevOps kültürü ve pratikleriyle ne kadar sık birlikte anıldığını ortaya koymaktadır. Docker ve docker-compose gibi konteyner teknolojileri, “devops & docker & docker-compose” etiketiyle temsil edilmekte ve bu kombinasyonun %0,22’lik bir kullanım oranı, konteynerizasyon ve mikro hizmetlerin DevOps süreçleri içindeki önemini yansıtmaktadır. Microsoft’un otomasyon ve komut satırı araçları olan PowerShell, “azure &

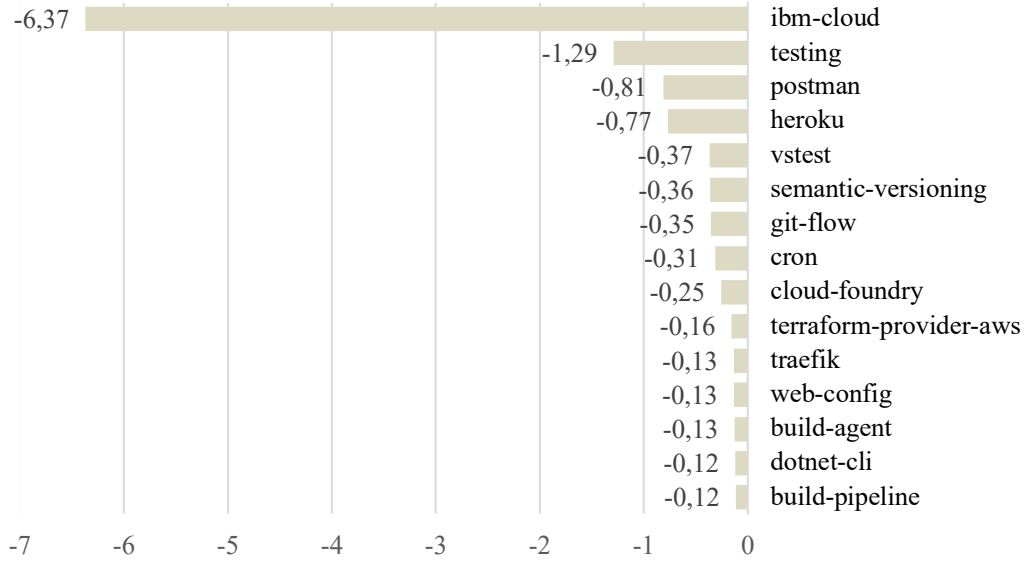
azure-devops & powershell” ve “azure-devops & azure-pipelines & powershell” kombinasyonları ile listelenmiştir. Bu da PowerShell’in, Azure DevOps ve Pipelines ile entegrasyonu ve otomasyon süreçlerindeki önemi hakkında bilgi vermektedir. Jenkins ve Jenkins pipeline, “devops & jenkins & jenkins-pipeline” etiketiyle, sürekli entegrasyon ve sürekli teslimat süreçlerindeki yaygın kullanımını göstermektedir. Genel olarak Tablo 4, Azure platformu ve DevOps araçları etrafında yoğunlaşmış bir kullanım modeli sergilemekte ve yazılım geliştirme ekosisteminin bu alanlarda ne kadar aktif olduğunu göstermektedir. Etiket kombinasyonlarının çeşitliliği, ekiplerin farklı araç ve teknolojileri bir araya getirerek daha verimli ve entegre çalışma ortamları oluşturduğunun kanıtı niteliğindedir. Çalışmanın etiket analizi bulgularının sayısal verilerinden yola çıkarak bazı hesaplamalar yapılmıştır. Eğitim hesabına göre en iyi performans gösteren (zamanla en çok kullanılan) ilk 15 etiket Şekil 10’da gösterilmektedir.



Şekil 10. Eğitim değerlerine göre en iyi ilk 15 etiket

Şekil 10’da görüldüğü gibi bulut hizmetleri kategorisinde, Microsoft’un bulut ve DevOps çözümleri ile ilgili etiketler dikkat çekmektedir. Nitekim “azure-devops” en yüksek eğitim değerine sahipken, “azure” ve “azure-pipelines” da önemli eğitim değerleri ile sıralamada üst pozisyonlarda yer almaktadır. Bu etiketler, Azure bulut platformuna ve

DevOps pratiklerinin entegrasyonuna olan ilginin yüksek olduğunu göstermektedir. Konteynerizasyon kategorisinde, “docker” ve “kubernetes” etiketleri öne çıkmaktadır. Docker, uygulama konteynerizasyonu için kullanılan bir araçken, Kubernetes, konteynerleri ölçeklendirmek ve yönetmek için kullanılmaktadır. Bu etiketlerin yüksek eğim değerleri, konteyner teknolojisinin ve mikro hizmet mimarilerinin popüleritesinin arttığını göstermektedir. Programlama dilleri ve araçları kategorisinde ise “git”, “yaml”, “powershell”, “c#”, “terraform”, “amazon-web-services” ve “python” etiketleri bulunmaktadır. Git, versiyon kontrol sistemleri içinde; YAML, yapılandırma dosyaları için; PowerShell, Windows tabanlı script yazımı ve otomasyon için; C#, Microsoft tarafından geliştirilen bir programlama dili; Terraform, altyapı kodu olarak bilinen IaC için; AWS, bulut hizmetleri sağlayıcısı; ve Python, genel amaçlı bir programlama dili olarak kullanılmaktadır. Bu etiketler, yazılım geliştirme ve sistem yönetimde yaygın olarak kullanılan diller ve araçların kapsamını temsil etmektedir. Bu çeşitlilik, teknoloji sektöründe birçok farklı araç ve dili kapsayan geniş bir uzmanlık alanı olduğunu göstermektedir. Eğim değerlerine göre en kötü performans gösteren ilk 15 etiket ise Şekil 11’de gösterilmektedir.

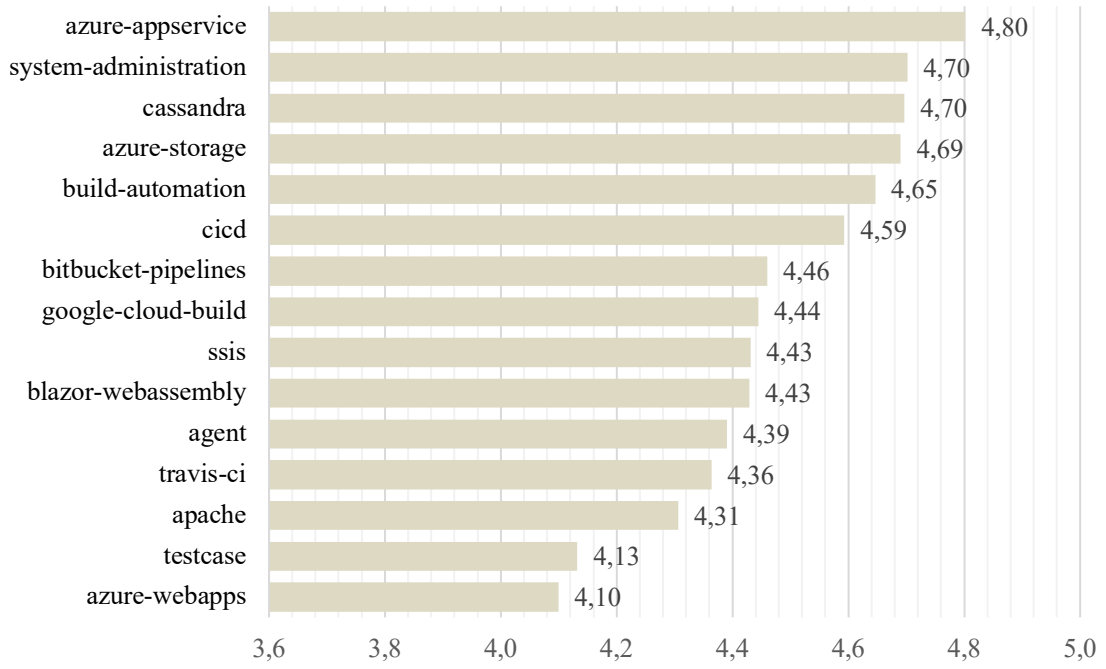


Şekil 11. Eğim değerlerine göre en kötü ilk 15 etiket

Şekil 11’de, eğimlerin azalışına göre (giderek kullanımı azalan) sıralanan 15 etiket gösterilmektedir. “ibm-cloud” etiketi, -6,37 ile en düşük eğim değerine sahiptir ve listenin başında yer almaktadır. Bu durum, IBM’in bulut servisi ile ilgili ilginin veya kullanımın

azaldığını gösterebilir. “testing” ve “postman” etiketleri sırasıyla -1,29 ve -0,81 eğim değerleriyle takip etmiştir. Bu da yazılım testi ve API test araçlarına olan ilginin azalabileceğine işaret edebilir. “heroku”, “vstest”, ve “semantic-versioning” gibi etiketler daha az negatif eğim değerleri gösterirken, bu durum bu teknoloji veya metodolojilere olan ilginin daha yavaş bir oranda azaldığını gösterebilir. “git-flow” ve “cron” etiketlerinin negatif eğim değerleri, bu iş akışları veya araçların kullanımında bir azalma olduğunu düşündürebilir. “cloud-foundry”, “terraform-provider-aws” ve “traefik” gibi altyapı ve bulut hizmetleriyle ilgili etiketlerin negatif eğim değerleri, bu özel araçların veya hizmetlerin popüleritesinin veya kullanımının azaldığını gösteriyor olabilir. En altta, “web-config”, “build-agent”, “dotnet-cli”, ve “build-pipeline” etiketleri de küçük negatif eğim değerlerine sahiptir. Bu durum, sürekli entegrasyon ve dağıtım süreçleriyle ilgili araç ve tekniklerin belirli yönlerinin popülerliğinin düşmekte olduğunu gösterebilir. Genel olarak bu grafik, teknoloji ve yazılım geliştirme alanında bazı araç, servis ve uygulamaların kullanım trendlerinin azalma yönünde olduğuna dair bir gösterge sağlamaktadır. Ancak, bu eğim değerleri yalnızca bir gösterge olduğundan, bunların sektördeki genel kullanım eğilimlerinin kesin bir yansıması olduğu sonucuna varılmamalıdır.

Her bir yılın etiket kullanım sayısına oranla ilgili bir etiketin o yılki kullanım sayısı, etiketin ilgili yıldaki hacmini göstermektedir. Buna göre son 10 yıldaki her bir etiketin toplam kullanım sayısının %100 olduğunu düşünürsek her yıl için bir hacim hesaplaması yapıldığında bu hesap her bir etiketin kendi son 10 yıllık performansı anlamına gelmektedir. Bu yüzdeliklerden çıkarılan eğim hesabına göre en fazla ivmelenen ilk 15 etiket Şekil 12’de gösterilmektedir.

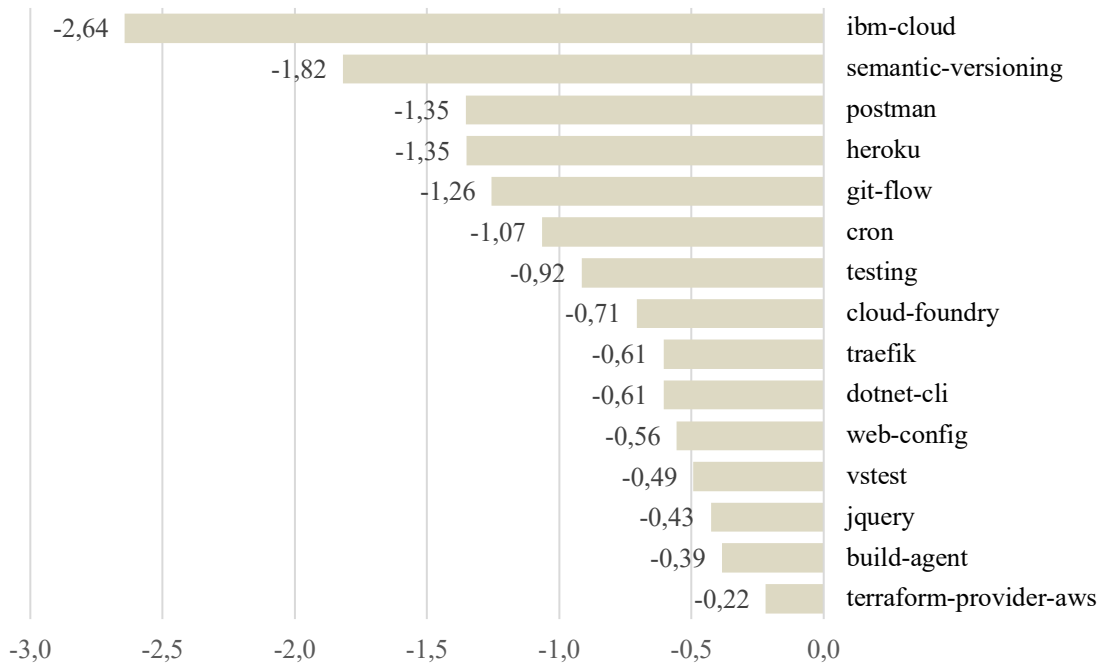


Şekil 12. Hacimlerin eğim hesabına göre en iyi ilk 15 etiket

Şekil 12’de, farklı teknolojiler veya hizmetler ile ilgili etiketler ve bunların zaman içindeki kullanım trendlerini gösteren eğim değerleri yer almaktadır. Göze çarpan ilk durum, bu etiketlerin çoğunun bulut bilişim, veri depolama, ve otomasyon gibi teknolojik konularla ilgili olduğudur. Örneğin, “azure-appservice” ve “azure-storage” etiketleri Microsoft’un Azure bulut platformunun iki farklı servisini temsil etmekte ve her iki etiketin de 4,70’lik bir eğim değerine sahip olması, bu hizmetlerin popülerliğinin giderek arttığını göstermektedir. Diğer etiketler arasında “cassandra” (veri depolama ve yönetimi için kullanılan bir NoSQL veritabanı sistemi), “ssis” (SQL Server Integration Services, veri entegrasyonu için bir platform) ve “blazor-webassembly” (web uygulamaları geliştirmek için bir framework) bulunmaktadır. Bu etiketlerin de 4,69 ve 4,43 gibi oldukça yüksek eğim değerlerine sahip olması, bu teknolojilerin kullanımının arttığını göstermektedir. Yazılım geliştirme ve otomasyonla ilgili etiketler de vardır. Bu etiketlere örnek; “build-automation”, “cicc”, “bitbucket-pipelines”, “google-cloud-build” ve “travis-ci” etiketleridir. Bunlar, yazılım geliştirme süreçlerinin otomatikleştirilmesi ve kodun kalitesinin sürekli olarak kontrol edilmesi alanlarında kullanılmaktadır. Eğim değerlerinin düşük olduğu etiketlere gelirsek, “testcase” ve “azure-webapps” etiketleri sırasıyla 4,13 ve 4,10 eğim değerlerine sahiptir. Bu, popülerliklerinin artış hızının diğer etiketlere göre biraz daha düşük olduğuna

işaret etmektedir. Genel olarak, grafikteki eğilimler, teknoloji dünyasında bulut bilişim, veri yönetimi ve otomasyonun öneminin artmaya devam ettiğini ve bu konularda faaliyet gösteren araçların ve hizmetlerin popülerliğinin arttığını göstermektedir.

Hacim oranlarına göre en fazla artış gösteren etiketlerin analizinin yanında bir de en az artış/en çok düşüş gösteren etiketler de analiz edilmiştir. Buna göre her etiketin zamanda kendi içerisinde kullanım hacminin azaldığı elde edilmiştir. Hacimlerin eğim hesabına göre en kötü performans gösteren ilk 15 etiket Şekil 13'te gösterilmektedir.



Şekil 13. Hacimlerin eğim hesabına göre en kötü ilk 15 etiket

Şekil 13'te görüldüğü gibi özellikle “ibm-cloud”, “semantic-versioning”, “postman” gibi etiketlerin başı çektiği listede, bu teknoloji veya pratiklerin eskisi kadar sık kullanılmadığını ya da yerlerini daha yeni ve gelişmiş araçlara bıraktıklarını düşündürebilir. “ibm-cloud” en üstte yer alarak en büyük negatif eğilime sahiptir. Bu, IBM bulut servislerinin kullanımının düşüşte olabileceğini veya bu etiketin daha az popüler hale geldiğini işaret ediyor olabilir. “semantic-versioning” ve “postman” da benzer bir eğilim göstermektedir; bunun sebebi bu araçların alternatifleriyle değiştirilmiş olması ya da farklı yaklaşımların benimsenmesi olabilir. “heroku”, “git-flow”, ve “cron” gibi araçlar da kullanım oranlarında düşüş göstermektedir. Bu, piyasaya yeni giren alternatiflerin etkisi

olabileceği gibi, sektördeki değişen ihtiyaçlar ve yaklaşımlar da etkili olmuş olabilir. “cloud-foundry”, “traefik” ve “dotnet-cli” gibi etiketlerin de düşük eğim değerlerine sahip olması, bu teknolojilerin veya uygulamaların popülerliklerinde azalma olduğunu gösterebilir. Bu, teknolojinin ilerlemesiyle daha eski araçların ve uygulamaların güncellenmesi veya yerlerini daha modern çözümlere bırakmaları sonucunda meydana gelmiş olabilir. Web geliştirmede kullanılan “jquery” gibi araçların eğim değerlerinin düşük olması da, modern frontend geliştirme pratiklerinin artık bu kütüphaneleri az kullandığını veya alternatiflerle değiştirdiğini göstermektedir. Son olarak, “terraform-provider-aws” etiketinin de listede yer alması, AWS için belirli Terraform sağlayıcılarının kullanımının azalabileceğini düşündürülebilir. Bu da kullanıcıların farklı konfigürasyon yönetimi araçlarına veya farklı bulut sağlayıcılarına geçiş yapmış olabileceğinin bir işareti olabilir. Bu tür negatif eğimler, belirli teknolojilerin ve uygulamaların kullanımındaki azalmaların yanı sıra teknolojik ilerleme ve sektördeki trend değişimlerinin bir göstergesi olabilir. Etiketin o yılki kullanım sayısının toplam kullanım sayısına göre yüzdesi alınarak, bulunan değerlere göre ivmelere bakıldığında en çok kullanılan etiketlerin ivmelerine göre tüm listede farklılıklar gözlenmektedir. Bu etiketler aslında diğer etiketlere göre daha iyi performans sergilediğinden bu etiketler için python scikit-learn kütüphanesindeki 2. dereceden lineer olmayan regresyon modeli ile gelecek üç yıl için gelecek tahmini yapılmıştır.

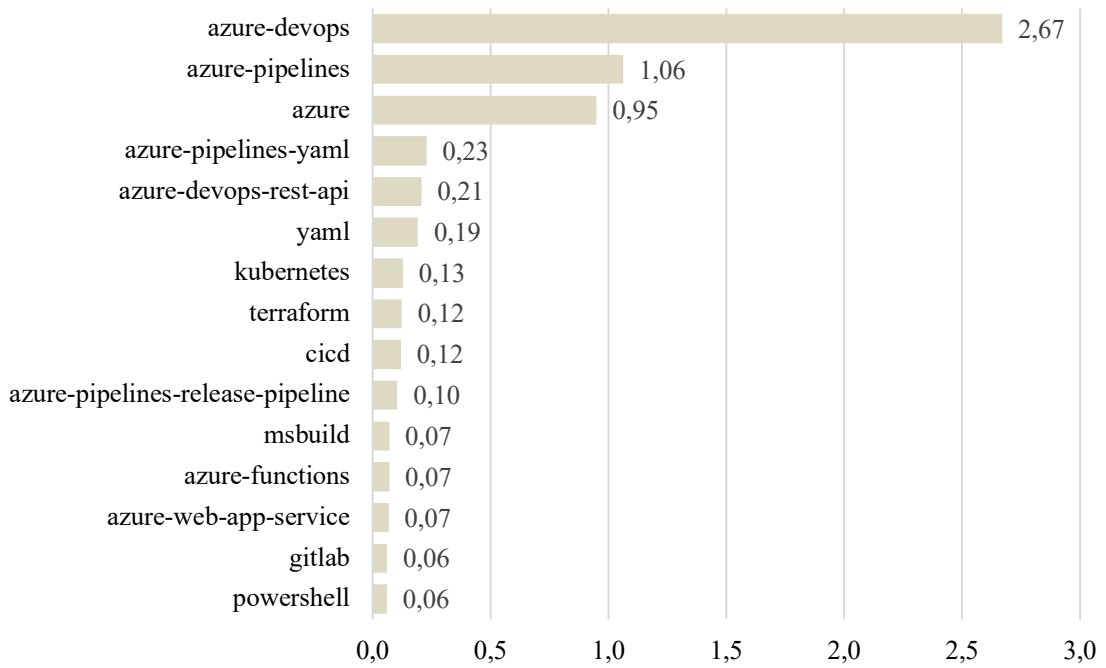
İlk 500 etiket içerisinde en iyi ivmelenen ilk 15 etiketin gelecek tahmini Şekil 14’te gösterilmiştir. Geçmiş verilere dayanarak gelecekteki eğilimleri tahmin etmek, etiketlerin nasıl değişeceğini ve gelişeceğini anlamak için önemlidir. Bu, gelecekte hangi konuların veya teknolojilerin daha popüler olacağını öngörmemizi sağlamaktadır. Özellikle teknoloji ve yazılım geliştirme alanlarında, hangi konuların veya araçların daha fazla ilgi göreceğini bilmek, stratejik planlamada yardımcı olabilir. Akademik ve endüstriyel araştırmalarda, hangi konuların daha fazla dikkat çekeceğini bilmek, araştırma projelerinin ve ürün geliştirme çalışmalarının odaklanacağı alanları belirlemek için kullanılabilir. Gelecek tahminleri 2024, 2025 ve 2026 yılları için hesaplanmıştır ve kesikli çizgiler ile gösterilmiştir. Veri setindeki gerçekleşen geçmiş değerler için düz çizgi kullanılmıştır.



Şekil 14. Yıllara göre eğim sıralı ilk 15 etiketın gelecek tahmini

Şekil 14’te son on yılda, bazı etiketlerin kullanımında istikrarlı bir artış göze çarpmaktadır. Özellikle, “azure-appservice”, “system-administration” ve “cicd” etiketleri, diğerlerine göre daha yüksek bir artış trendi göstermekte, bu da onların alanlarında daha fazla benimsendiğine işaret etmektedir. “cassandra” ve “apache” gibi etiketlerde ani zirveler mevcuttur; bu da belirli olaylar veya sürümlerle ilişkili geçici popülerlik dalgalanmalarına işaret ediyor olabilir. “blazor-webassembly” ve “agent” gibi etiketlerdeki düşüşler, teknolojideki değişikliklere veya bu teknolojilerin popülerliğinin azalmasına bağlı olabilir. “travis-ci” ve “azure-webapps” gibi etiketlerin kullanımı zaman içinde yavaş yavaş artarken, “google-cloud-build” ve “bitbucket-pipelines” gibi etiketlerin kullanımı daha değişken bir patika izlemiştir. Tahmini kullanım sayılarına göre, çoğu etiketin gelecek yıllarda kullanımının artacağı öngörülse de, bu tahminler mevcut trendlere ve veriye dayalı tahminlere dayanıyor ve her zaman kesin olmayabilir. Bu etiketler, belirli bulut hizmetleri, veri tabanları, otomasyon sistemleri ve geliştirme araçlarını ifade ediyor olabilir ve gözlemlenen trendler, yazılım geliştirme endüstrisindeki genel değişimleri ve tercihleri yansıtıyor olabilir.

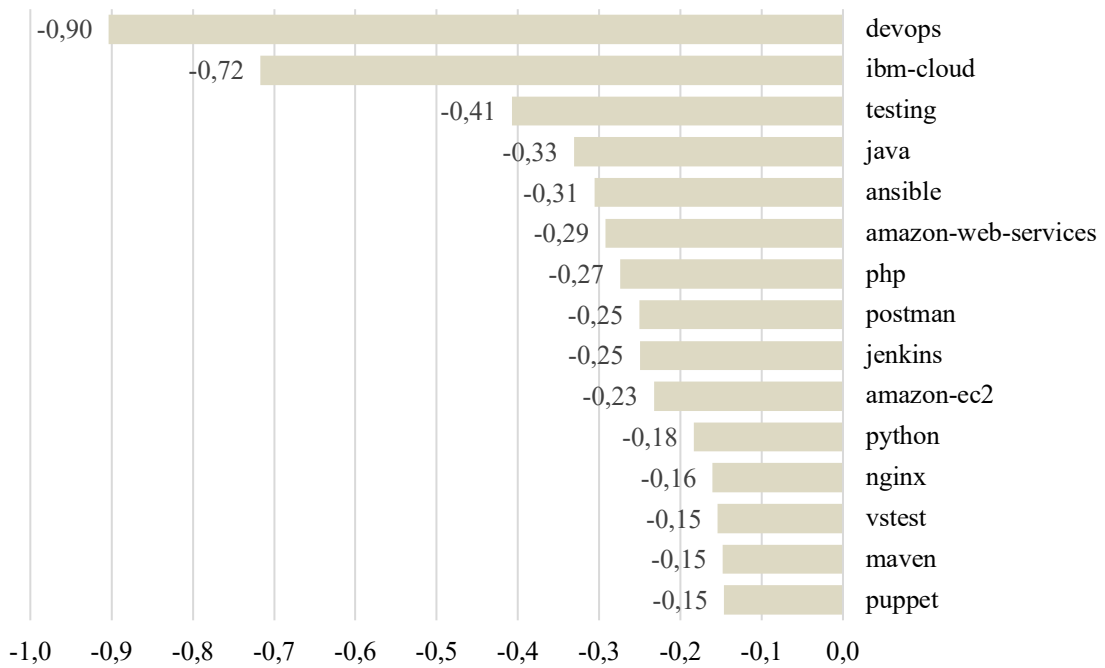
Veri setinde her yıl kullanılan etiket sayısı değişkenlik göstermektedir. Buna göre her bir yıldaki kullanım sayısının o yıldaki kullanım sayısına bölünmesi ile bulunan değer, etiketin diğer etiketlerin kullanımına göre değerlendirilmiştir. Bu veriler ışığında hesaplanan hacim yüzdelerine göre yapılan eğim hesabında en iyi performans gösteren ilk 15 etiket Şekil 15’te gösterilmektedir.



Şekil 15. Diğer etiketlere kıyasla eğitim sıralı en iyi ilk 15 etiket

Şekil 15’te görüldüğü gibi, en yüksek artış oranına sahip olan “azure-devops” etiketi, 2,67’lik bir değerle dikkat çekmektedir. Bu, Microsoft’un Azure DevOps platformunun popülerliğinin hızla arttığını göstermekte ve bu da sürekli entegrasyon, sürekli teslimat ve diğer DevOps pratiklerine olan artan ilgiden kaynaklanmaktadır. “azure-pipelines” ve “azure” etiketlerinin de sırasıyla 1,06 ve 0,95 gibi değerlere sahip olması, Azure’un genel bulut hizmetleri ve özellikle sürekli entegrasyon/dağıtım işlem hatları için kullanımının arttığını göstermektedir. Bu eğilimler, bulut tabanlı çözümlerin ve DevOps altyapısının giderek daha yaygın hale geldiğini işaret ediyor. “azure-pipelines-yaml”, “azure-devops-rest-api” ve “yaml” gibi etiketlerin daha düşük, fakat pozitif eğitim değerleri, bu teknolojilerin ve uygulamaların da popülerliklerini artırdığını fakat önceki etiketlere göre daha yavaş bir oranda olduğunu göstermektedir. “kubernetes” ve “terraform” etiketleri, konteyner orkestrasyonu ve altyapı olarak kod pratiklerinin sağlam bir artış gösterdiğini belirtmektedir. Bu eğilimler, yazılım geliştirme süreçlerinde otomasyon ve altyapı yönetiminin öneminin arttığını göstermektedir. “cicd”, “azure-pipelines-release-pipeline” ve “msbuild” etiketlerinin artış oranları da bu araç ve metodolojilerin kullanımının arttığını, ancak daha sınırlı bir oranda olduğunu göstermektedir. Son olarak, “azure-functions”, “azure-web-app-service”, “gitlab” ve “powershell” gibi etiketlerin de artış gösterdiği görülmektedir. Ancak

bu artış oranları oldukça düşük kalmıştır. Bu da onların hala popüler olduğunu, ancak yavaş bir büyüme oranına sahip olduklarını göstermektedir. Genel olarak bu grafik, Azure hizmetlerinin ve DevOps altyapısının giderek artan bir popülerlik kazandığını, konteyner teknolojileri ve altyapı otomasyon araçlarının da benzer bir yükseliş içinde olduğunu göstermektedir. Daha düşük eğim değerlerine sahip etiketler bile pozitif bir artışa sahipken bu durum, teknoloji sektörünün bu alanlardaki sürekli evrimini ve gelişimini yansıtmaktadır. Etiketlerin diğer etiketlere kıyasla hesaplanan hacimlerinin eğim hesabına göre en kötü ilk 15 etiket Şekil 16’da gösterilmiştir.

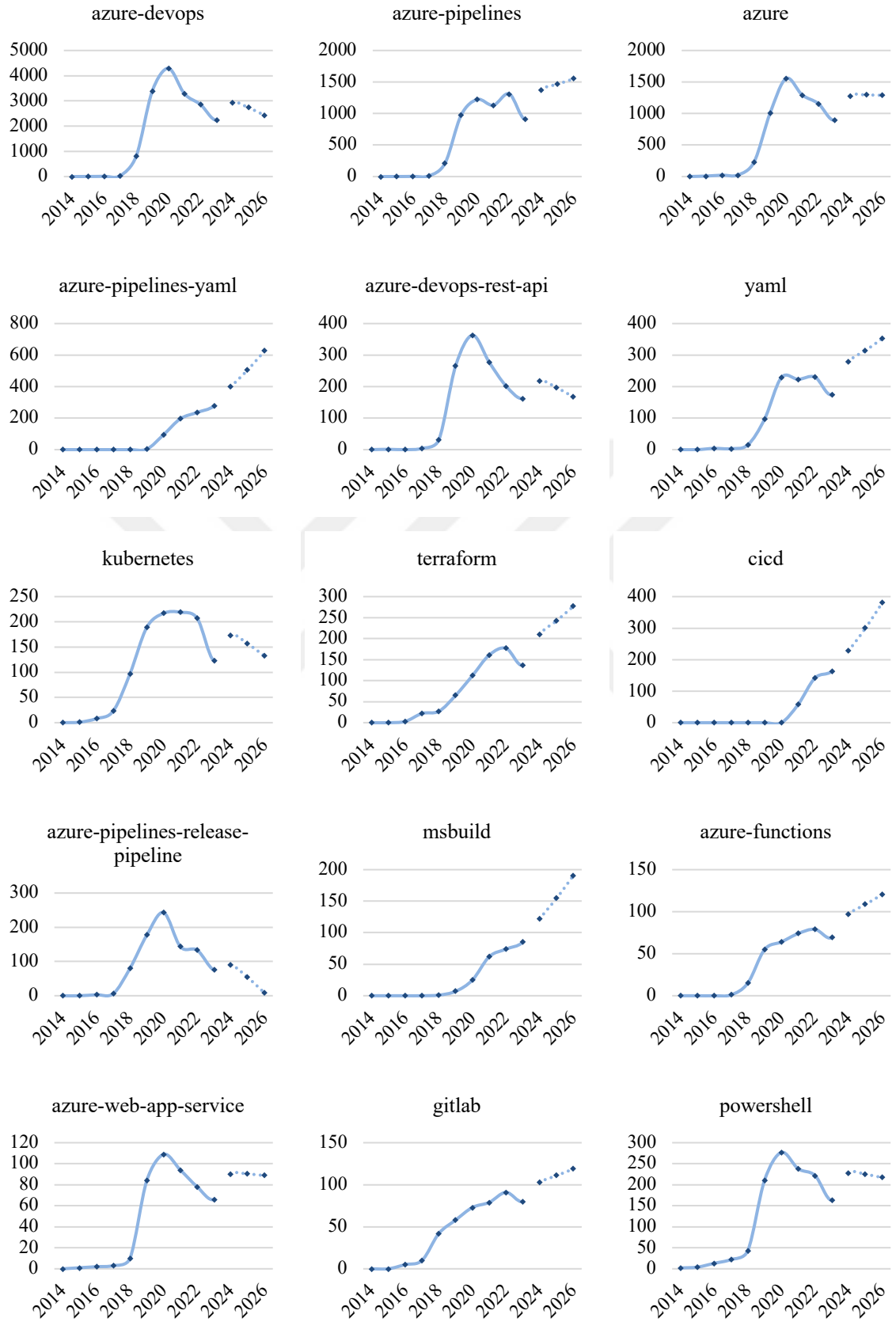


Şekil 16. Diğer etiketlere kıyasla eğim sıralı en kötü ilk 15 etiket

Şekil 16’da, DevOps kültürü ve uygulamalarıyla ilgili bazı etiketlerin son on yılda nispeten düşen kullanım oranları gösterilmektedir. En dikkat çeken etiket “devops” olup, bu büyük düşüş DevOps terimine ve yaklaşımlarına olan ilginin azalıyor olabileceğini ya da bu terimin daha geniş anlamda kabul görmesiyle spesifik DevOps etiketlerinin daha az kullanılır hale geldiğini işaret edebilir. “ibm-cloud” ve “testing” etiketlerindeki düşüş, IBM Cloud platformunun belki de diğer bulut hizmetleri karşısında rekabet gücünü yitiriyor olmasını veya yazılım testi yaklaşımlarının evrilerek farklı araç ve tekniklerle anılıyor olmasını gösterebilir. “java” ve “php” gibi programlama dilleri de azalan eğim değerleri

göstermektedir ve bu durum bu dillerin popülerliğinin modern alternatifler lehine azalmakta olduğuna işaret edebilir. Ancak bu diller hala geniş kullanım alanlarına sahip olduklarından, bu düşüş mutlaka bu dillerin kullanımdan kalktığını göstermez; belki de sektörde yeni ve trend olan dillerin ortaya çıkmasıyla ilişkilendirilebilir. “ansible”, “amazon-web-services”, ve “jenkins” gibi DevOps araçlarının kullanımındaki azalış, teknolojideki değişimlere ve yeni araçların yükselişine bağlı olabilir. Örneğin, AWS’nin geniş bir hizmet yelpazesine sahip olmasına rağmen, kullanıcılar belki de daha uzmanlaşmış veya yenilikçi çözümlere yöneliyor olabilir. “postman” ve “amazon-ec2” gibi etiketler, belirli bir teknolojinin veya hizmetin azalan spesifik kullanımını işaret ediyor olabilirken, “python” ve “nginx” gibi diğer etiketlerdeki hafif düşüşler, bu teknolojilerin hala oldukça popüler olmasına rağmen, kullanımlarının zaman içinde değişkenlik gösterdiğini göstermektedir. Son olarak, “vtest”, “maven”, ve “puppet” gibi etiketler, belki de değişen teknoloji yığınları ve yazılım geliştirme pratiklerine bağlı olarak azalan kullanım eğilimleri göstermektedir. Şekil 16, DevOps ve yazılım geliştirme alanlarında kullanılan teknolojilerin ve araçların nasıl değişebileceği konusunda iç görü sağlamak ve bazı araçların ve uygulamaların zamanla nasıl gözden düşebileceğine dair örnekler sunmaktadır.

Etiketlerin diğer etiketlere göre o yıl içindeki hacimleri düşünüldüğünde performanslarının eğim hesabı diğer etiketlere göre ayrılan etiketler olduklarını göstermektedir. Gelecek tahminleri için 2. dereceden lineer olmayan regresyon modelini bu eğim hesaplarına göre ilk 15 etiket için kullanarak Şekil 17’de gelecek üç yıl için kullanım sayısı tahminleri verilmiştir.

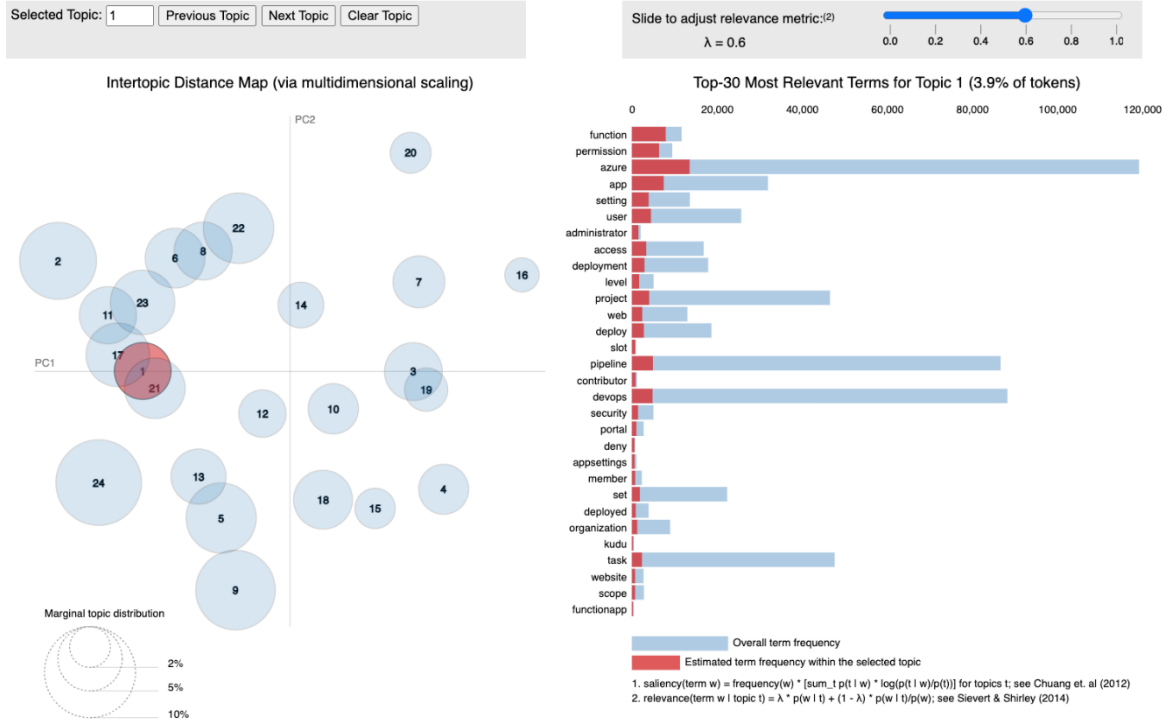


Şekil 17. Diğer etiketlere kıyasla eğim sıralı en iyi ilk 15 etiketin gelecek tahmini

Şekil 17’de, diğer etiketlere kıyasla ivme sıralı en iyi ilk 15 etiketin son on yıldaki gerçek kullanım verilerini (düz çizgiler) ve geleceğe dair tahminlerini (kesikli çizgiler) göstermektedir. İlk dikkat çeken grafikte, “azure-devops” etiketi özellikle 2018 ve 2019 yıllarında büyük bir kullanım artışı göstermiş, ancak sonrasında düşüşe geçmiştir. Bu düşüş, DevOps kültürünün evrimi veya “azure-devops” hizmetlerinin kullanım şekillerinin değişmesi gibi faktörlerle ilişkili olabilir. “azure-pipelines” ve “azure” etiketleri, daha kararlı bir artış göstererek, Azure’un sürekli entegrasyon ve dağıtım pipeline’ları için yaygın bir platform olarak kullanımının arttığını göstermektedir. “azure-pipelines-yaml” etiketinin sabit bir yükselişi, YAML konfigürasyon dosyalarının Azure DevOps işlem hatlarında giderek daha fazla kullanıldığını gösteriyor olabilir. “azure-devops-rest-api” etiketindeki önceki yıllardaki büyük zirve, REST API’lerinin bu dönemde önemli bir entegrasyon aracı olarak popüler olduğunu, ancak sonrasında kullanımın azalmış olabileceğine işaret etmektedir. “yaml” etiketinin genelde sabit bir artış sergilemesi, YAML’ın DevOps içerisinde sürekli bir popülerlik kazandığını göstermektedir. “kubernetes” ve “terraform” etiketlerindeki zirveler, bu araçların bulut altyapı yönetimi ve otomasyonu için kritik olduğunu ve zamanla önemli noktalarda kullanıldığını, ancak kullanım trendlerinde inişler ve çıkışlar olduğunu göstermektedir. “cicd” etiketinin istikrarlı artışı, sürekli entegrasyon ve sürekli dağıtımın DevOps pratiklerinde merkezi bir rol oynadığını doğrulamaktadır. “azure-pipelines-release-pipeline” ve “msbuild” gibi etiketler, belirli araçların ve iş akışlarının kullanımındaki değişiklikleri yansıtmaktadır. Ancak bu etiketlerdeki tahmini kullanım sayıları gelecekte bir artışa işaret etmektedir. “azure-functions” ve “gitlab” etiketlerindeki yavaş ve kararlı artış, bu teknolojilerin DevOps içindeki önemli rolünü ve büyüyen popülerliğini göstermektedir. Son olarak, “azure-web-app-service” ve “powershell” etiketlerindeki dalgalanmalar, teknoloji ve araç kullanımının zamanla nasıl değişebileceğine dair örnekler sunmaktadır. “powershell”, özellikle Windows merkezli otomasyon ve yapılandırma için kullanılırken, “azure-web-app-service” Microsoft’un bulut tabanlı web hizmetlerine olan ilginin dalgalı grafiği göstermektedir. Bu grafikler, DevOps kültüründe kullanılan teknolojilerin ve araçların geçmişte nasıl bir yol izlediğini ve önümüzdeki yıllarda hangi trendlerin devam edebileceğini veya ortaya çıkabileceğini tahmin etmede yardımcı olabilir. Her bir etiketin kullanımı, teknoloji dünyasındaki daha geniş eğilimlerin ve endüstrinin ihtiyaçlarının bir göstergesi olarak okunabilir.

3.2. Konu Modelleme Analizleri Bulguları ve Tartışma

Bu bölümde, veri seti üzerinde gerçekleştirilen GDA tabanlı konu modellemesi analizinden elde edilen bulgular çeşitli yönleri ele alınmaktadır. Analiz sonucunda 24 konulu model elde edilmiştir ve bu konular Şekil 18’de pyLDAvis kütüphanesi ile görselleştirilmiştir.



Şekil 18. PyLDAvis ile konu modellemenin görselleştirilmesi

Şekil 18’de PyLDAvis kütüphanesinin çıktılarını gösterirken lambda 0,6 olarak seçilmiştir. PyLDAvis’in lambda parametresi, bir topic modeldeki terimlerin önemini yorumlamak için kullanılan bir ayarlamadır. Lambda değeri, bir terimin konuya özgü ne kadar belirgin (discriminative) olduğu ile o konu genelinde ne kadar yaygın olduğunu dengeler. Lambda 0.6 gibi bir değer seçildiğinde, hem terimlerin konuya özgü bilgi taşıma gücü (belirginlik) hem de o terimin tüm dokümanlarda ne kadar yaygın olduğunu gösterir. Bu değer, lambdanın tam 0 (sadece belirginlik) veya 1 (sadece yaygınlık) olmaması anlamına gelir. Lambda 0,6 her iki özelliği de dikkate alan bir denge sağlamaktadır. Lambda değeri 1’e yaklaştıkça, terimlerin konu genelindeki yaygınlığı daha fazla vurgulanır. Lambda değeri 0’a yaklaştıkça, terimlerin o özel konu içindeki belirginliği daha fazla vurgulanır.

Lambda 0,6 kullanıcıya terimlerin hem konu belirginliğine hem de genel yaygınlığına bakma imkanı sunarak daha dengeli bir görsel sunum sağlamaktadır. Bu, kullanıcıların konuları yorumlamalarına yardımcı olacak bilgileri dengeli bir şekilde görmelerini sağlamaktadır.

Haritada konuların konumlarına ve dairelerin boyutlarına bakarak, her bir konunun veri setindeki ağırlıkları ve birbirleriyle olan ilişkileri çıkarılabilir. Haritada gösterilen çıktılara bakarak, konuların veri setindeki ağırlığı ve ilişkileri hakkında çeşitli gözlemler yapılabilir. Konu 1, 21 ve 17'nin büyük dairelerle temsil edildiğini ve birbirine yakın konumlandırıldıklarını görülmektedir. Bu, Azure hizmetleri, kaynak yönetimi ve DevOps uygulamaları hakkındaki konuların veri setinde oldukça önemli olduğunu ve birbiriyle ilişkili terimler içerdiğini göstermektedir. Bunlar, altyapı yönetimi ve otomasyonla ilgili birbirine bağlı konseptlerdir. Öte yandan, konu 24 (Dinamik Web Sayfaları) daha izole ve daha küçük bir daireyle temsil edilmektedir. Bu da bu konunun veri setinde diğer konulara kıyasla daha az yer kapladığını ve muhtemelen daha özelleşmiş, belirli programlama dilleriyle ilgili olduğunu işaret etmektedir. Konu 6 (Linux'ta Node.js Uygulamaları), 8 (Git ve Sürüm Kontrolü) ve 22 (Azure DevOps Sorunlarını Giderme) bir araya gruplandırılmış ve büyük dairelerle gösterilmiştir, bu da bu konuların teknik olarak iç içe geçmiş olduğunu ve veri setinde önemli bir yer tuttuğunu göstermektedir. Bunlar, Azure üzerinde DevOps sorunlarını giderme gibi modern yazılım geliştirme süreçlerinin önemli yönlerini temsil etmektedir. Daha küçük dairelerle temsil edilen ve haritanın farklı köşelerinde yer alan konulara bakıldığında, bunların daha spesifik konular olabileceği düşünülebilir. Örneğin, konu 9 (Azure Hizmetleri ve İzinleri) ve konu 13 (Azure Kaynak Şablonları) veri setinde belirli bir yere sahip olmasına rağmen, bu konuların dağılımı ve diğer konularla olan ilişkisi daha sınırlı olabilir. Bu harita, bir organizasyonun eğitim veya araştırma odaklarını belirlerken, hangi konuların birlikte ele alınabileceğine ve hangi konuların daha fazla vurgulanması gerektiğine dair stratejik kararlar almasına rehberlik edebilir. Ayrıca, hangi konuların daha fazla içerik veya kaynak geliştirmeye ihtiyaç duyduğunu göstermesi açısından da faydalı olabilir. Tablo 5'te GDA sonuçlarından konu numarası, konu isimlendirmesi, konu terimleri ve yüzdeleri verilmiştir. Bu bilgiler en yüksek kullanım sayısına istinaden yüzdelik hesabına göre en yüksek değerli konudan başlayarak azalan şekilde sıralanmıştır.

Tablo 5. Konu numarası, konu isimlendirmesi, konu terimleri ve yüzdeleri

Konu	Konu İsimlendirmesi	Konu Terimleri	Yüzde
1	Azure'da DevOps'u Uygulama	devops item azure work api get rest project team field query user list create view feature state board date process	%17,97
2	Azure'da İşlem Hatlarını Otomatikleştirme	variable task script pipeline azure powershell value build devops command run output file set name error get yaml environment bash	%13,54
3	Azure'da Dağıtımların Güvenliği	azure function app pipeline permission devops user project setting access deployment task deploy error web set level run microsoft security	%13,35
4	Docker, Nginx ve Maven ile Konteyner Uygulamaları	container docker server file maven run port app nginx version compose application proxy image php dependency service volume host project	%10,10
5	.NET Core Uygulamaları	build file project task artifact net version pipeline package microsoft dotnet folder publish nuget azure error solution core devops csproj	%7,95
6	Linux'ta Node.js Uygulamaları	node npm file install version error home bin command android run module build linux path usr err lib directory available	%6,41
7	Azure'da API Erişimi	token azure api secret devops access key get pat client vault user request value rest code response service credential header	%5,20
8	Git ve Sürüm Kontrolü	branch pull commit request git merge master feature main file commits policy push devops azure create ref code tag develop	%4,31
9	Azure Hizmetleri ve İzinleri	service azure connection app resource create application role devops error access account user subscription name deploy tenant permission network microsoft	%4,07
10	SSH ve Ansible ile Güvenli Sunucu	ssh key file ansible certificate host name user password server error app config local ssl path root var command cert	%3,08
11	Azure'da Git Depoları	git repository repo project azure devops visual studio code repos tfs github gitlab clone source file push remote server folder	%2,11
12	Azure'da Testleri Otomatikleştirme	test run azure devops code pipeline coverage case task report nuget file project unit build error xml suite plan sonarqube	%2,02
13	Azure Kaynak Şablonları	name parameter template value job variable type string environment resource arm stage json echo yaml env input yaml property output	%1,94
14	Azure'da Veritabanı Yönetimi	file sql database azure server error devops wiki system command microsoft task module script work net migration data program line	%1,42

Konu	Konu İsimlendirmesi	Konu Terimleri	Yüzde
15	AWS ve Terraform Kullanımı	aws terraform resource instance task name lambda gradle create amazon action plan module log ec bucket run error code provider	%1,25
16	Docker ve Python ile Konteynerleştirme	docker image package build run container version install python feed cache npm registry command azure error dockerfile file push latest	%1,03
17	Kendi Kendine Barındırılan Aracılar	agent azure server devops hosted machine run window version self microsoft pool pipeline job service install tool premise set upgrade	%0,90
18	Uçtan Uca İşlem Hattı	pipeline build release azure stage trigger devops run yaml deployment job environment deploy file branch create repository code dev repo	%0,75
19	Buluttaki Jenkins Örnekleri	jenkins time instance server data application node cpu load request job cloud metric process database memory limit take prometheus user	%0,66
20	Kubernetes Kümeleri	cluster kubernetes pod name namespace node ingres error kubect helm angular port app service yaml type nginx deployment spec io	%0,58
21	Web Uygulamalarında Hata Ayıklama ve Dağıtma	app debug build web displayname azure deploy task input zip file module static function python pipeline stage install site name	%0,42
22	Azure DevOps Sorunlarını Giderme	azure devops file error account extension work json email data code page get user organization click log microsoft notification problem	%0,37
23	Java ve Python Uygulamaları	java python table line site lib sonar file database eclipse jvm jetty package internal self core mymi base util error	%0,31
24	Dinamik Web Sayfaları	class tag html text function tcp const return var accept src import name type content google form anywhere value page	%0,26

Tablo 5’te ilk 5 isimlendirmeye bakıldığında “Azure’da DevOps’u uygulama”, çeşitli terimlerle ifade edilen bir dizi faaliyeti içermekte olup, bu alandaki çalışmaların %17,97’sini oluşturmaktadır. Bu faaliyetler arasında projeler, ekipler ve API’ler gibi öğeler bulunmaktadır. “Azure’da İşlem Hatlarını Otomatikleştirme” ise %13,54 oranında yer almakta ve değişkenler, komut dosyaları ve powershell komutları gibi terimler öne çıkmaktadır. “Azure’da Dağıtımların Güvenliği” %13,35 oranında önemli bir yer tutmakta ve bu konuda kullanıcı izinleri, proje ayarları ve güvenlik seviyeleri gibi terimler kullanılmaktadır. “Docker, Nginx ve Maven ile Konteyner Uygulamaları” %10,10 oranında ifade edilmekte ve sunucu dosyaları, portlar ve proxy gibi terimler içermektedir. Son olarak,

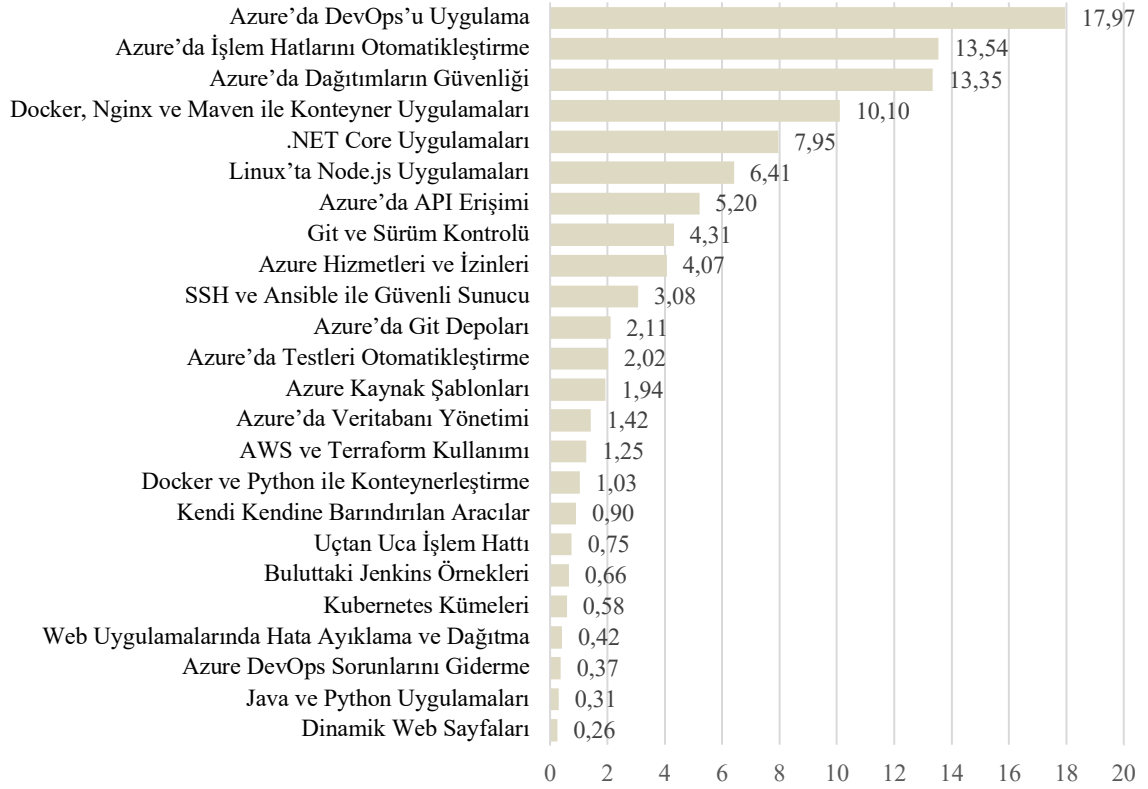
“NET Core Uygulamaları” %7,95 oranında temsil edilmekte olup, bu bağlamda yapılar, paketler ve nuget gibi terimler ön plana çıkmaktadır. Bu ilk 5 konu tüm konuların %62,91’ini kapsamaktadır. Tablo 6’da konuları oluşturan kelimeler, yıllara göre kaç adet konu geçtiği, son 10 yıldaki toplam konu sayısı(n) ve yüzdeleri(f) verilmiştir.

Tablo 6. Yıllara göre konu sayıları, toplam adedi(n) ve yüzdesi(f)

Konu	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	n	f
1	9	32	74	78	258	1034	1459	1017	860	664	5485	%17,97
2	3	18	46	70	187	643	973	808	792	593	4133	%13,54
3	5	14	38	46	233	716	915	804	746	558	4075	%13,35
4	27	78	170	183	281	449	594	501	424	376	3083	%10,10
5	2	4	19	21	177	572	582	402	371	277	2427	%7,95
6	7	39	71	86	140	293	383	344	338	256	1957	%6,41
7	1	10	17	21	82	251	352	284	270	300	1588	%5,20
8	1	12	15	21	70	209	286	255	243	204	1316	%4,31
9	3	31	49	39	69	204	246	222	203	175	1241	%4,07
10	10	26	55	63	76	133	158	157	131	130	939	%3,08
11	1	2	5	11	43	131	147	115	112	76	643	%2,11
12	1	5	11	12	38	111	137	116	86	101	618	%2,02
13	0	1	6	10	31	48	123	117	122	133	591	%1,94
14	1	3	15	11	18	69	95	84	67	72	435	%1,42
15	1	8	12	25	37	44	63	67	67	59	383	%1,25
16	0	1	3	6	15	41	63	66	51	68	314	%1,03
17	1	2	6	5	16	53	67	48	49	29	276	%0,90
18	0	0	0	2	10	32	70	40	39	37	230	%0,75
19	4	6	10	9	20	21	36	33	35	26	200	%0,66
20	0	1	3	4	21	29	32	31	42	15	178	%0,58
21	0	1	2	2	4	25	13	21	19	41	128	%0,42
22	0	1	1	1	6	25	30	14	8	28	114	%0,37
23	0	1	1	7	3	8	17	16	16	25	94	%0,31
24	0	0	2	3	7	6	16	17	14	15	80	%0,26
Toplam	77	296	631	736	1842	5147	6857	5579	5105	4258	30528	%100,00

Tablo 6’da son 10 yıldaki konu sayısına bakıldığında en hacimli ilk 3 konunun toplamda %44,86 ile neredeyse yarısını oluşturmaları önemli bir veridir. Hacim olarak en düşük 3 konunun toplamı ise %0,94 olarak görülmektedir. Dağılımların 24 konuya göre 16. konu da dahil yüzde 1’in üzerinde olması dikkat çekmektedir. Tabloda, farklı konulara ait yıllara göre gelişimleri ve istatistiksel sonuçlar gösterilmektedir. 2014 yılından başlayıp

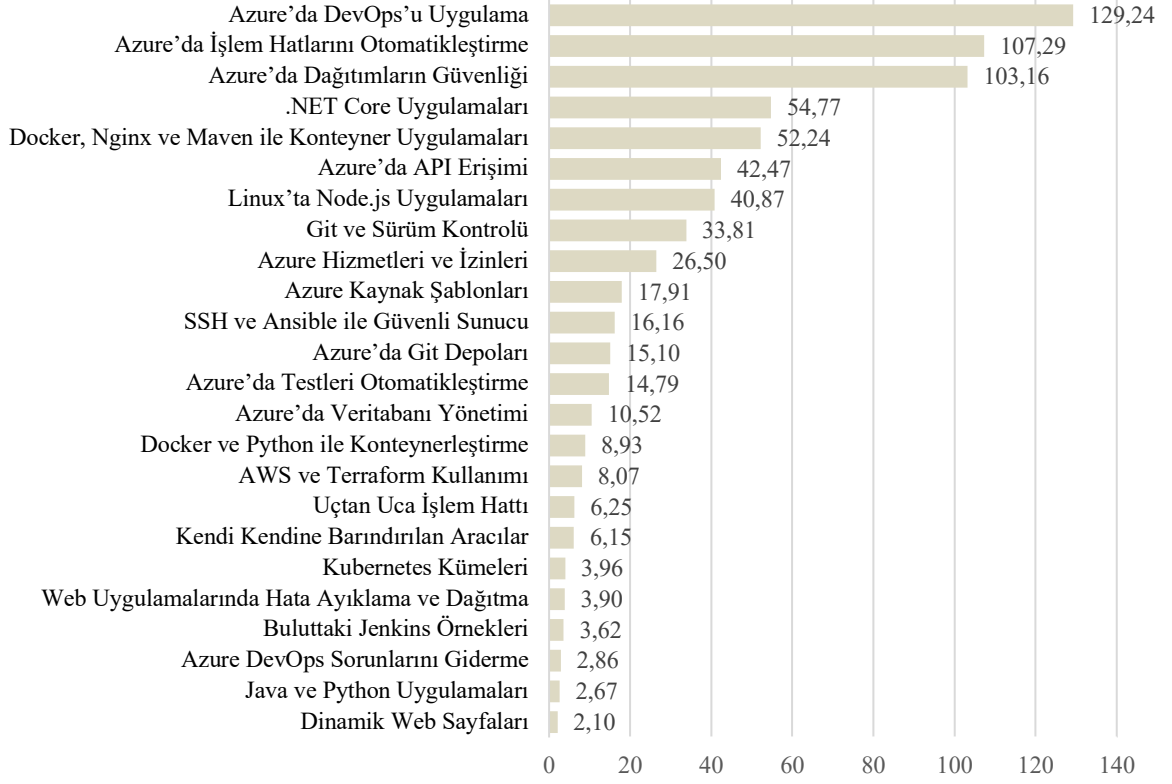
2023 yılına kadar uzanan veriler, her yıl için konu başlıklarına göre dağılımı ve her konunun genel toplam içerisindeki yüzdesel oranını göstermektedir. Özellikle 1 numaralı konu dikkat çekici bir artış sergilemiş ve toplamda %17,97 ile en yüksek paya sahiptir. Diğer taraftan, bazı konuların zaman içinde istikrarlı bir artış veya azalış gösterdiğini, bazılarının ise nispeten sabit kaldığı görülmektedir. Genel toplama baktığımızda, yıllar içinde toplam sayıda önemli bir artış olduğu ve bu artışın özellikle 2019 ve 2020 yılları arasında daha belirgin olduğu görülmektedir. Her bir konu, ilgili alanda meydana gelen değişimleri ve trendleri anlamak için değerli veriler sunmaktadır. Şekil 19'da konuların isimlendirmeleri ve hacim yüzdeleri sıralı bir içinde verilmiştir.



Şekil 19. Konu isimlendirmeleri ve yüzdeleri

Şekil 19'da görüldüğü gibi, DevOps ile ilgili çeşitli teknolojik konularda toplanan verilerin yüzdesel bir dağılımı gösterilmektedir. Grafikteki düzen, her konunun göreceli önemini ve alandaki araştırma veya uygulama sıklığını anlamada yardımcı olabilir. Bu veriler, bulut bilişim ve DevOps pratikleri konusunda gelişen trendleri ve ilgi alanlarını

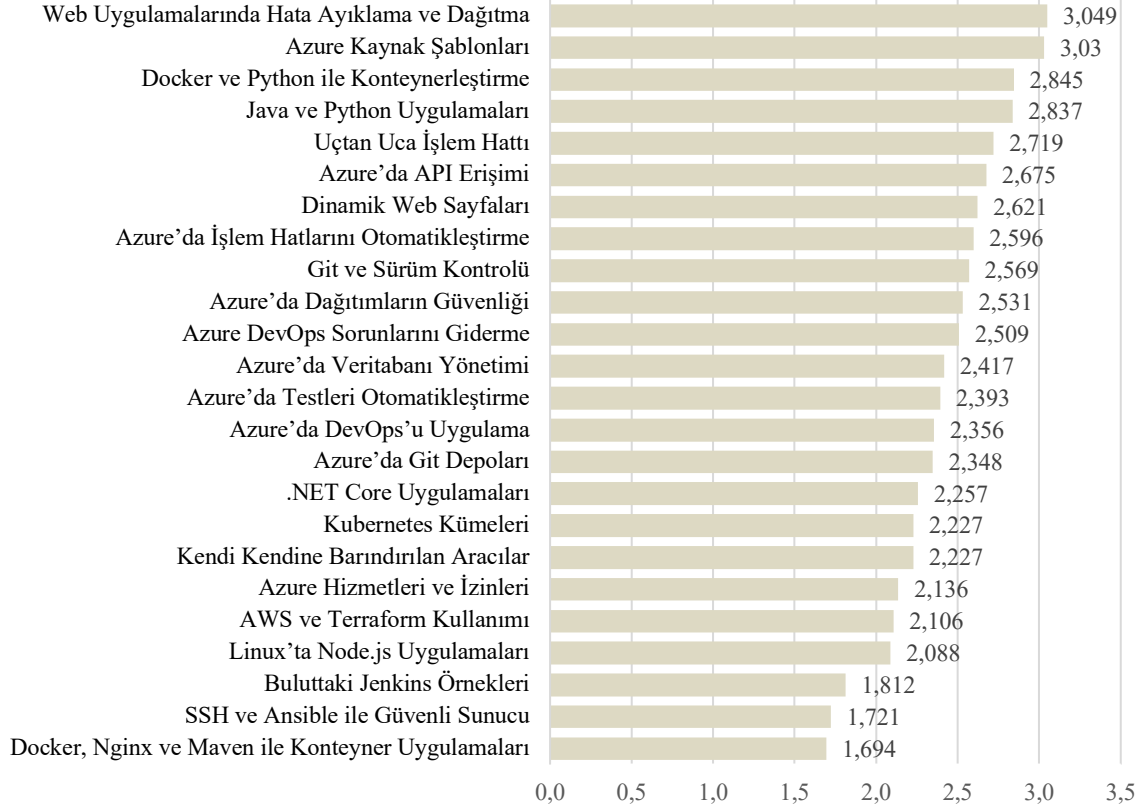
analiz etmek için kullanılabilecek bir temel sunmaktadır. Konuların yıllara göre sayılarındaki değişimlerin eğim hesabına göre sıralamaları Şekil 20’de gösterilmiştir.



Şekil 20. Konu sayılarına göre eğim sıralı konu listesi

Şekil 20’de görüldüğü gibi, aynı teknoloji konularının zaman içindeki yüzdelerinin eğimine göre sıralandığı bir görsel sunumdur ve bir önceki analizde sunulan yıllık dağılım verilerine dayanmaktadır. Göze çarpan bir değişiklik, “.NET Core Uygulamaları” konusunun, önceki analizde 5. en yüksek yüzdelerle paya sahip olmasına karşın, bu analizde eğim açısından 4. sıraya yükselmiş olmasıdır. Bu durum, bu konunun zaman içindeki göreceli popülaritesinde bir artışa işaret edebilir. Buna karşın, “Buluttaki Jenkins Örnekleri” konusunun eğimi artış göstererek, bu alandaki artan ilgi ve önemin bir göstergesi olarak değerlendirilebilir. “Azure’da Dağıtımların Güvenliği” konusu da 103,16 ile yüksek bir eğime sahiptir. Bu da bu konunun teknoloji alanında sürekli olarak önemini koruduğunu veya arttırdığını gösterebilir. Diğer yandan, “Java ve Python Uygulamaları” konusunun %2,67 gibi düşük bir eğimle listenin alt sıralarında yer alması, bu dillerin kullanımının veya popülaritesinin zaman içinde daha az artış gösterdiğini veya nispeten sabit kaldığını işaret

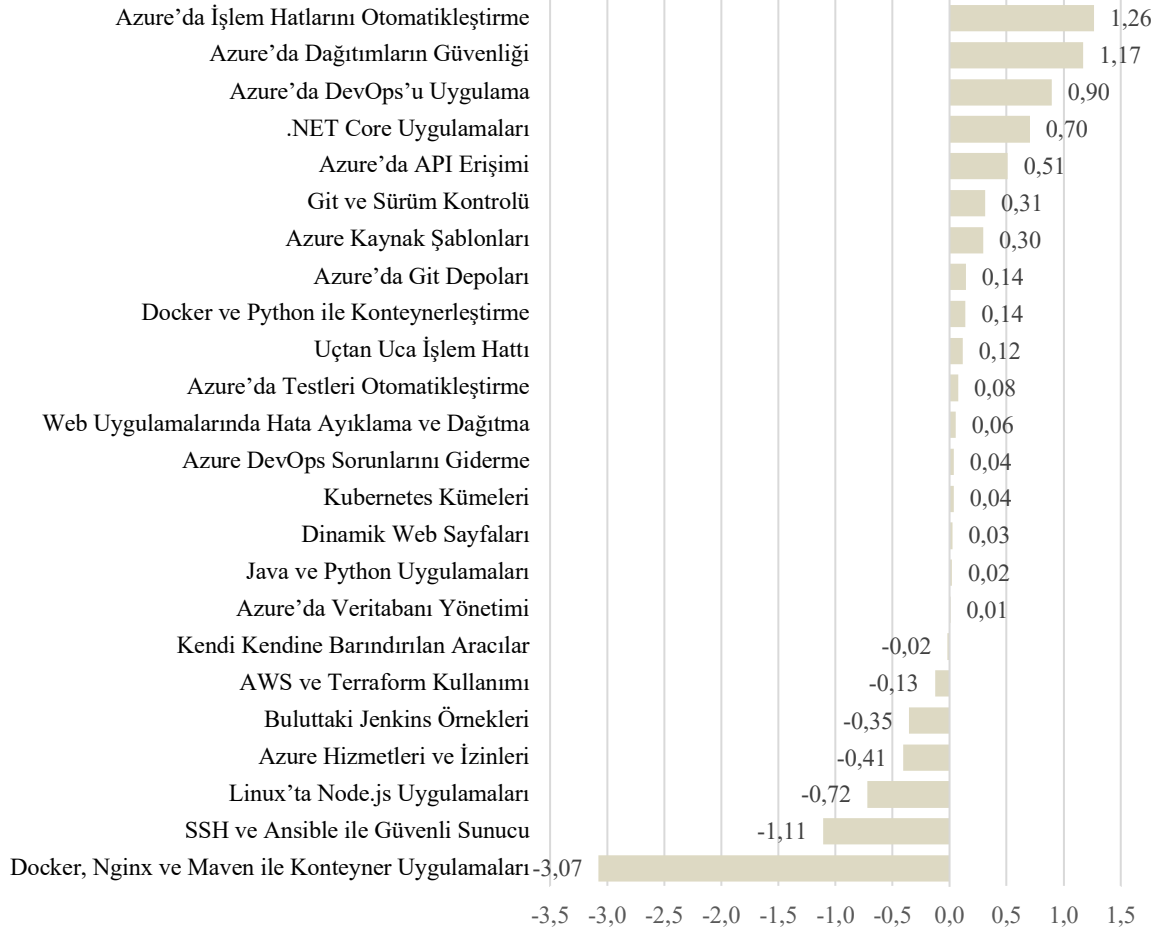
etmektedir. Konuların hacimlerinin yıllara göre hesaplanan değerlerinin eğimine göre en iyiden itibaren konular ve eğimleri Şekil 21’de gösterilmiştir.



Şekil 21. Diğer yıllara kıyasla eğitim sıralı konu listesi

Şekil 21’de yıllara göre konuların o yılki kullanım sayısının yüzdesi hesaplandıktan sonra eğitim değerleri hesaplanarak bu eğitim değerlerine göre sıralanan konular gösterilmiştir. Bu bar grafik konu sıralamalarına bakıldığında Şekil 20’deki konu sayılarına göre ivme sıralı konu listesi sıralamaları ile farklı pozitif değerler içermektedir. En yüksek eğitim değerine sahip “Web Uygulamalarında Hata Ayıklama ve Dağıtma” konusu genel listede 21. sırada iken bu listede 1. olarak dikkat çekmektedir. Orta sıralar çok farklılık göstermemekte fakat genel listede 4. sırada olan “Docker, Nginx ve Maven ile Konteyner Uygulamaları” konusu bu listede 24. sırada görülmektedir. Fakat bu durum bu konu için negatif bir yargı oluşturmamalıdır. Tüm konuların eğitimlerinin pozitif olması bu konuların DevOps ile ilgili ilerleyen yıllarda yine kullanılacakları anlamı taşıyabilir. Yani yıllara göre konular birbirine yakın bir şekilde artış veya azalış göstermiştir. Şekil 22’de konunun yıllara göre diğer

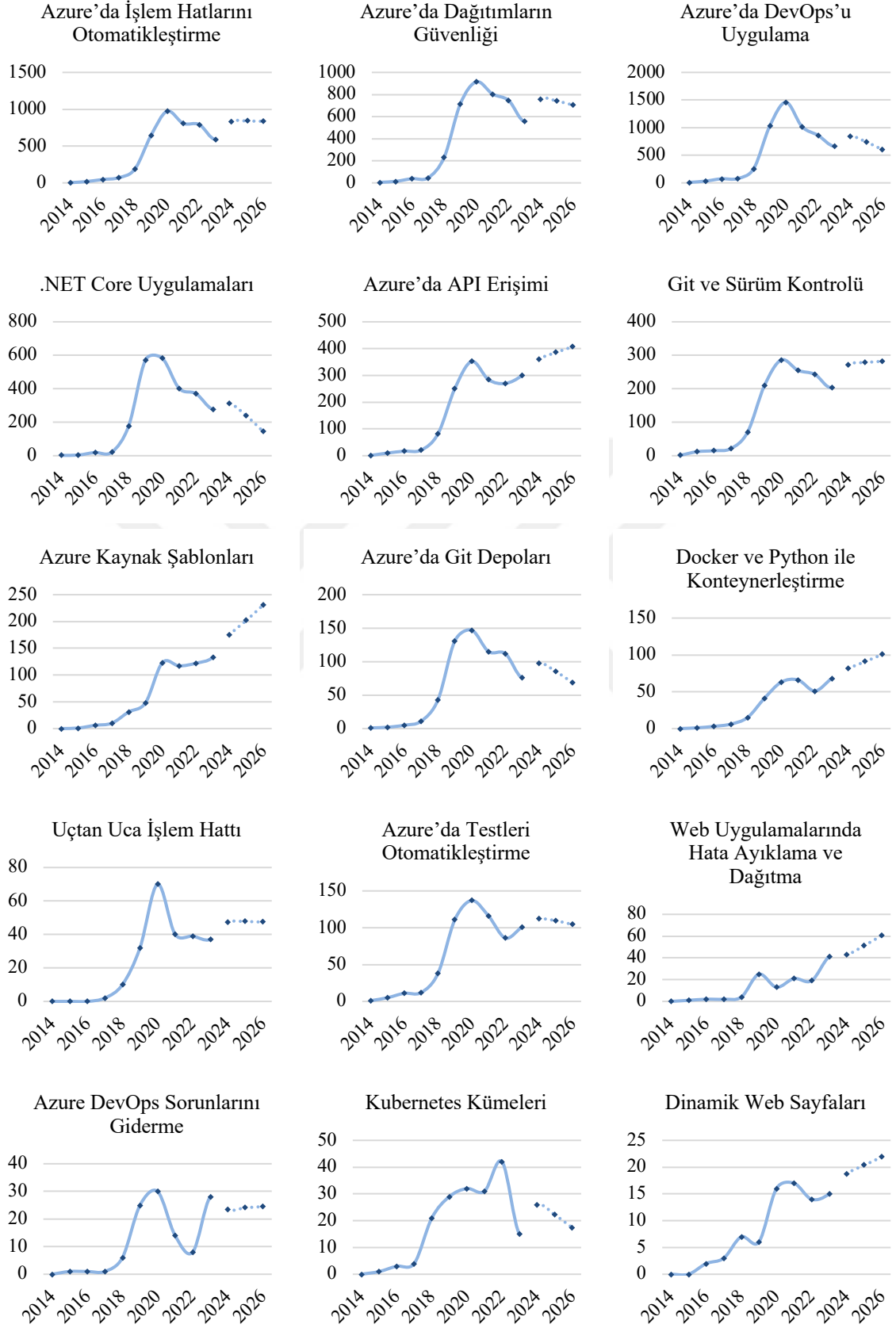
konulara kıyasla hacim oranının eğim hesabına göre konular en iyiden başlayarak sıralanmıştır.



Şekil 22. Diğer konulara kıyasla eğim sıralı konu listesi

Şekil 22'de görülen bar grafik, DevOps ile ilişkili teknoloji konularının 2014'ten 2023'e kadar olan süreçteki göreceli popülerite değişimlerini eğim değerleriyle ifade etmektedir. Analizde, "Azure'da İşlem Hatlarını Otomatikleştirme" konusunun zaman içinde artan bir eğimi (+1,26) koruyarak popülerliğini sürdürdüğü görülmektedir. "Azure'da Dağıtımların Güvenliği" konusu, gösterdiği (+1,17) değeri ile bu dönemde kayda değer bir ilgi artışı yaşamıştır. Bu, Azure'da dağıtımların sürekli gelişen teknolojiler olarak önemini koruduğunu ve hatta artırdığını göstermektedir. "Azure'da DevOps'u Uygulama" (+0,90), ".NET Core Uygulamaları" (+0,70) ve "Azure'da API Erişimi" (+0,51) konuları da pozitif bir eğime sahip olup, bu alanlarda sürekli bir ilgi ve gelişme olduğunu işaret etmektedir.

Diğer yandan, “Git ve Sürüm Kontrolü” ve “Azure Hizmetleri ve İzinleri” gibi otomasyon odaklı konular, az da olsa artan bir eğilim (+0,31 ve +0,30) gösterirken, “Azure’da Git Depoları” ve “Azure’da Testleri Otomatikleştirme” gibi konular sınırlı artışlar (+0,14) sergilemişlerdir. Ancak, “Docker, Nginx ve Maven ile Konteyner Uygulamaları” konusunda (-3,07) yaşanan büyük düşüş, bu teknolojilerin veya araçların kullanımında bir azalma veya diğer teknolojik konulara veya araçlara göre göreceli bir tercih değişikliğine işaret edebilir. Konunun yıllara göre diğer konulara kıyasla yüzdelerinin eğim hesabına göre konu listesindeki iyi göstergeli ilk 15 konu için gelecek tahmini yapılmıştır ve Şekil 23’te gösterilmiştir. Gelecek tahmini 2. dereceden lineer olmayan regresyon formülüne göre 2024, 2025 ve 2026 yılları için hesaplanan değerler ile yapılmıştır ve bu yıllar grafiklerde noktalı çizgi ile gösterilmiştir.



Şekil 23. Diğer konulara kıyasla en iyi ilk 15 konunun gelecek 3 yıl tahmini

Şekil 23'te yer alan grafikler, eğimleri diğer konulara göre daha iyi olan ilk 15 konunun geçmişten günümüze kadar olan gerçekleşen değerlerini ve bu değerlerin 2024, 2025 ve 2026 yılları için yapılan tahminleri göstermektedir. “Azure’da DevOps’u Uygulama”da belirgin bir artış gözlemlenirken, 2020 yılında doruk noktasına ulaştıktan sonra bir azalma eğilimi görülmekte ve tahminler bu düşüşün devam edeceğini işaret etmektedir. “Docker ve Python ile Konteynerleştirme” konusundaki ilgi artışı istikrarlı bir şekilde devam etmekte ve gelecek yıllar için de pozitif bir eğilim tahmin edilmiştir. “Web Uygulamalarında Hata Ayıklama ve Dağıtma” konusu ise 2020’de zirveye ulaştıktan sonra düşüşe geçmiş, ancak gelecek yıllar için yavaş bir toparlanma öngörülmüştür. “Uçtan Uca İşlem Hattı” konusu, 2018’de yaşanan büyük ilgi sonrası düşüş göstermiş ve bu eğilimin devam etmesi beklenmiştir. “Azure’da API Erişimi” ve “Azure Kaynak Şablonları” konularında gözlenen artışın, önümüzdeki yıllarda devam edeceği tahmin edilirken, “Git ve Sürüm Kontrolü” ve “Uçtan Uca İşlem Hattı” konularında kademeli bir artış öngörülmürken “.NET Core Uygulamaları”ndaki düşüşün önümüzdeki yıllarda devam etmesi öngörülmüştür. Son olarak, “Dinamik Web Sayfaları” alanında istikrarlı bir büyüme ve bu büyümenin tahmin edilen yıllarda da sürmesi beklenmektedir. Bu tahminler, mevcut eğilimlerin ve endüstri içindeki ilginin gelecekteki yönünü anlamak için değerli birer gösterge oluşturmaktadır. Bu tahminler birer gösterge olup DevOps’un değişen ve gelişen bir alan olması dolayısıyla gelecekteki değerler bu tahminlerden daha farklı sonuçlar olarak karşımıza çıkabilir.

3.3. Etiket Analizleri ile Konu Modelleme Analizleri Bulgularının İlişkisi ve Tartışma

Çalışmada etiket analizi bulguları ile konu modelleme analizi bulgularının ilişkisi incelenmiştir. Bunun için konuların diğer konulara göre artış gösteren konular ile azalış gösteren konuların hangi etiketler ile kullanıldıkları saptanmıştır. Bu bilgilere göre etiketlerin diğer etiketlere göre daha iyi performans gösteren ilk 15 etiketin ve daha kötü performans gösteren ilk 15 etiketin 24 konudaki kullanım sayıları ayrı ayrı bulunmuştur.

Tablo 7. Pozitif performans gösteren ilk 15 etiketın konu dağılımı

Etiket	Artan Konulardaki Adedi	Azalan Konulardaki Adedi	Fark
azure-devops	12891	4029	8862
azure-pipelines	4183	1594	2589
azure	4810	1384	3426
azure-pipelines-yaml	597	210	387
azure-devops-rest-api	1133	173	960
yaml	692	284	408
kubernetes	745	339	406
terraform	557	147	410
ci/cd	249	114	135
azure-pipelines-release-pipeline	701	159	542
msbuild	93	164	-71
azure-functions	311	46	265
azure-web-app-service	371	76	295
gitlab	297	141	156
powershell	1072	123	949

Tablo 7’de pozitif performans gösteren ilk 15 etiketın konu dağılımı gösterilmiştir. Artan konulardaki kullanım sayıları azalan konulardaki kullanım sayılarından oldukça fazla bulunmuştur. Aralarında “msbuild” etiketi ise azalan konularda daha fazla kullanılarak negatif bir sonuç doğurmuştur. Bu durum beklenen bir sonuç değildir. Diğer etiketlerin durumu ise beklenen sonuçtur. Artan performans gösteren konuların kullanım yüzdesi %74,19 ile oldukça yüksek olduğundan bu sonuçların ortaya çıkması doğaldır. Tablo 8’de ise negatif performans gösteren ilk 15 etiketın konu dağılımı gösterilmiştir.

Tablo 8. Negatif performans gösteren ilk 15 etiketın konu dağılımı

Etiket	Artan Konulardaki Adedi	Azalan Konulardaki Adedi	Fark
devops	6038	1837	4201
ibm-cloud	198	43	155
testing	121	19	102
java	363	58	305
ansible	390	62	328
amazon-web-services	841	247	594
php	144	6	138
postman	34	26	8
jenkins	806	289	517
amazon-ec2	231	44	187
python	597	246	351
nginx	272	13	259
vstest	41	33	8
maven	308	15	293
puppet	31	26	5

Tablo 8 incelendiğinde diğer etiketlere göre daha kötü performans gösteren ilk 15 etiketın tamamının artan konulardaki kullanım sayılarının azalan konulardaki kullanım sayılarından daha fazla olduğu görülmüştür. Bu durum beklenen bir sonuç olmadığı gibi bu değerlerin artan konuların kullanım yüzdesinin %74,19 ile oldukça yüksek olmasından kaynaklı olabilir. Sonuç olarak artan konulardaki artan etiketlerin daha fazla kullanılması pozitif olarak ilişkilendirilebilmektedir fakat azalan etiketlerin de artan konularda daha fazla kullanılması negatif ilişkilendirilebilir.

4. SONUÇLAR

Bu çalışma, SO platformunda "devops" etiketiyle işaretlenmiş sorular ve bu sorulara verilen cevaplar üzerinde gerçekleştirilen kapsamlı bir analiz sunmaktadır. Araştırmanın amacı, DevOps ile ilgili en çok tartışılan konuları, tarihsel gelişimi ve gelecekteki potansiyel eğilimleri belirlemektir. Bu amaçla, etiket analizi ve konu modelleme analizi yöntemleri kullanılmıştır.

4.1. Etiket Analizi Sonuçları

Etiket analizi sonuçları DevOps uygulamalarındaki en popüler teknolojileri ve araçları ortaya koymuştur. "azure-devops", "devops", "azure", "azure-pipelines" ve "docker" gibi etiketlerin sık kullanımı, bulut tabanlı çözümlerin ve konteyner teknolojilerinin DevOps dünyasında ne kadar önemli olduğunu göstermektedir. "azure-devops" etiketi, 2014'ten itibaren belirgin bir artış göstererek, Microsoft'un bulut tabanlı DevOps çözümlerinde lider konumunu pekiştirdiğini ortaya koymaktadır. Bu artış, Microsoft'un Azure platformuna ve DevOps araçlarına yönelik yatırımlarının etkisini yansıtmaktadır. "docker" etiketi de önemli bir artış göstererek, konteyner teknolojisinin DevOps uygulamalarında giderek daha popüler hale geldiğini ve yazılımların taşınabilirliği ve ölçeklenebilirliğini artırmak için kullanıldığını göstermektedir. Bunlara ek olarak, DevOps dünyasında bazı trendlerin değiştiğini de göstermektedir. "ibm-cloud" ve "testing" gibi etiketlerin zaman içinde azalan kullanımı, bu teknolojilerin yerini yeni ve daha gelişmiş çözümlere bırakmaya başladığını düşündürmektedir. Bu değişim, DevOps alanında sürekli bir gelişim ve yenilik olduğunu ve teknolojilerin hızla evrimleştiğini göstermektedir.

4.2. Konu Modelleme Analizi Sonuçları

Konu modelleme analizi sonuçları "devops" etiketli soruların altında yatan ana temaları belirlemiştir. Bu çalışmada belirlenen 24 ana konu arasında, "Azure'da DevOps'u Uygulama", "Docker ve Python ile Konteynerleştirme", "Web Uygulamalarında Hata Ayıklama ve Dağıtım", "Uçtan Uca İşlem Hattı" ve "Sürekli Entegrasyon" gibi konular öne çıkmaktadır. "Azure'da DevOps'u Uygulama" konusunun yüksek hacmi, Azure platformunun DevOps uygulamalarında ne kadar yaygın olarak kullanıldığının bir

göstergesidir. "Docker ve Python ile Konteynerleştirme" konusunun zaman içinde hızlı artışı, konteyner teknolojilerinin DevOps alanında hızla benimsendiğini göstermektedir. "Web Uygulamalarında Hata Ayıklama ve Dağıtım" ve "Uçtan Uca İşlem Hattı" konularının pozitif eğilimi ise, DevOps uygulamalarında hata ayıklama ve dağıtım süreçlerinin öneminin arttığını göstermektedir. Bununla birlikte, bazı konuların zaman içinde azalan kullanım oranları da gözlemlenmiştir. "Java ve Python Uygulamaları" konusunun azalan eğilimi, bu dillerin kullanımının daha yeni ve daha popüler alternatifler lehine azalmakta olduğunu düşündürmektedir. Bu durum, yazılım dünyasında sürekli bir evrim olduğunu ve teknoloji trendlerinin hızla değiştiğini göstermektedir.

4.3. Çalışmanın Genel Sonuçları

Bu çalışma, DevOps alanındaki eğilimlerin yazılım dünyasına ne tür katkılar sağlayabileceğini ve ne tür beklentiler yaratabileceğini açıkça göstermektedir. Azure platformunun artması ve konteyner teknolojilerinin yükselişi, şirketlerin yazılımları daha hızlı, güvenilir ve ölçeklenebilir bir şekilde teslim etmesini sağlayacaktır. "Sürekli Entegrasyon" ve "Dağıtım Otomasyonu" gibi konuların daha da gelişmesi, yazılım geliştirme süreçlerinin otomasyonunu artırarak, ekiplerin daha verimli çalışmasına olanak sağlayacaktır. Bu gelişmelerin, yazılım dünyası için önemli faydaları olacağı ön görülmüştür:

- Hızlı ve Güvenilir Yazılım Teslimi: Şirketler, yazılımları daha hızlı ve güvenilir bir şekilde teslim edebilecek, bu da rekabet güçlerini artıracaktır.
- Müşteri Memnuniyeti: Daha hızlı ve daha kaliteli yazılım teslimi, müşteri memnuniyetini artıracaktır.
- Optimum Yazılım Geliştirme Süreçleri: Otomasyon ve bulut tabanlı çözümler, yazılım geliştirme süreçlerini optimize edecek, ekiplerin daha verimli çalışmasına olanak tanıyacaktır.
- Daha Yüksek Üretkenlik: DevOps uygulamaları, yazılım geliştiricilerinin ve operasyon ekiplerinin daha yüksek bir üretkenlik seviyesine ulaşmasını sağlayacaktır.

Çalışmanın sonuçları, DevOps alanındaki eğilimlerin yazılım dünyasını nasıl değiştireceğine ve sektör profesyonellerine, eğitim kurumlarına ve politika yapıcılara ne tür

katkılar sağlayabileceğine ışık tutmaktadır. Bu çalışma, sektör profesyonellerine, eğitim kurumlarına ve politika yapıcılara bir dizi önemli bilgi sağlamaktadır. Sektör profesyonelleri bu çalışmanın sonuçlarını kullanarak, DevOps stratejilerini optimize edebilir, yeni teknolojileri benimseyebilir ve yazılım geliştirme süreçlerini daha verimli hale getirebilirler. Eğitim kurumları, DevOps eğitimi için daha güncel ve endüstri ihtiyaçlarına uygun müfredatlar oluşturabilirler. Politika yapıcılar ise, DevOps'un sektör gelişimine olan etkisini daha iyi anlayabilir ve bu alanda destekleyici politikalar geliştirebilirler.



5. ÖNERİLER

Bu çalışma, SO platformunda "devops" etiketiyle işaretlenmiş sorular ve bu sorulara verilen cevaplar üzerinde yapılan bir analizle, DevOps alanında önemli bilgiler sağlamaktadır. Ancak, her araştırma gibi, bu çalışma da bazı sınırlılıklara sahiptir. Bu sınırlılıklar, gelecekte yapılacak çalışmalara yön verebilir ve alanın daha da derinlemesine incelenmesine katkı sağlayabilir. Bu çalışmanın en belirgin sınırlılığı, veri setinin SO platformuna özgü olmasıdır. SO, DevOps alanında çalışan tüm bireyleri temsil etmediği için, çalışma sonuçları DevOps uygulamalarındaki genel eğilimleri tam olarak yansıtmayabilir. Ayrıca, veri setinin sadece sorular ve cevaplar içermesi, DevOps uygulamalarıyla ilgili daha geniş bir bilgi yelpazesini kapsayan başka veri kaynaklarının (örneğin, bloglar, haber makaleleri) incelenmesinin gerekliliğini ortaya koymaktadır. Çalışmada kullanılan etiket analizi ve konu modelleme yöntemleri de belirli sınırlılıklara sahiptir. Etiketler, soruların içeriğini tam olarak yansıtmayabilir ve farklı kullanıcılar tarafından aynı etiketi farklı anlamlarda kullanabilir. Konu modelleme ise, verilerde gizli kalmış birçok detayı ortaya çıkaramıyor olabilir ve karmaşık ilişkileri tam olarak yakalayamayabilir. Dolayısıyla, bu yöntemlere ek olarak, daha gelişmiş doğal dil işleme teknikleri, derin öğrenme temelli konu modelleme algoritmaları ve veri madenciliği yöntemleri kullanılarak DevOps alanındaki tartışmaların daha derinlemesine incelenmesi gereklidir. Bu sınırlılıklar göz önüne alındığında, gelecekte yapılacak çalışmalar için bazı öneriler sunulabilir:

- Çoklu Veri Kaynağı Entegrasyonu: Gelecekteki çalışmalar, Stack Overflow platformundan elde edilen verileri, DevOps ile ilgili blog yazıları, haber makaleleri ve diğer platformlardan elde edilen verilerle birleştirerek daha geniş bir veri seti oluşturabilir ve DevOps alanındaki tartışmaları daha geniş bir perspektiften incelemeyi sağlayabilir.
- Karma Yöntemli Yaklaşım: Gelecekteki çalışmalar, etiket analizi ve konu modelleme analizi gibi yöntemleri daha gelişmiş tekniklerle birleştirerek kullanabilir. Örneğin, doğal dil işleme teknikleri, metin verilerinden daha derin anlamlar çıkarabilir ve daha hassas sonuçlar elde edilmesini sağlayabilir.
- Boylamsal Çalışmalar: DevOps alanındaki değişimleri daha net bir şekilde anlamak için, uzun bir zaman aralığını kapsayan boylamsal çalışmalar yapılabilir. Bu

alışmalar, teknolojilerin zaman içindeki evrimini ve DevOps uygulamalarının nasıl deęiştirdini gösteren deęerli bilgiler sağlayabilir.

5.1. DevOps Alanında Çalışanlara Yönelik Öneriler

DevOps alanında çalışanlara yönelik öneriler ele alındığında, alanda çalışmak isteyen kişilerde olması gereken özelliklerden en belirginleri sürekli öğrenme becerisi, işbirliği ve iletişim kabiliyeti ve işlemleri otomatikleştirme becerisi olarak düşünülebilir. Bu kavramlar:

- Sürekli Öğrenme: DevOps alanının sürekli gelişen bir alan olduğundan çalışmada oldukça söz edilmektedir. Teknolojiler hızla deęişmekte ve her yıl yeni araçlar ortaya çıkmaktadır. Bu sebeple DevOps alanında çalışanlar, sadece bir tek alana odaklanmak yerine, sürekli öğrenmeye ve kendilerini güncel tutmaya özen göstermelidir. Yeni teknolojileri takip etmeli, sektördeki son gelişmeleri incelemeli ve eğitim programlarına katılarak bilgilerini artırmalıdır.

- İş Birliği ve İletişim: DevOps alanında çalışanların başarılı olması için, geliştirme ekipleri ve operasyon ekipleri arasında iyi bir iş birliği ve iletişim olmalıdır. Bu işbirliği sadece teknik alanla sınırlı kalmamalı, aynı zamanda ortak hedeflere yönelik bir kültür de oluşturulmalıdır. Ekipler arasındaki süreçlerin kolaylaştırılmasını ve sorunların daha hızlı bir şekilde çözülmesini sağlayacaktır. Ekipler arasında etkin iletişimi ve bilgi paylaşımını teşvik eden kültürel deęişimler desteklenmelidir.

- Otomasyon : Otomasyon, DevOps için esasen kritik bir unsurdur. Ekipler, sıkıcı ve tekrarlayan işleri otomatikleştirerek, daha verimli çalışabilir ve hata riskini azaltabilirler. Otomasyon sadece tek bir aşamalı bir işlem deęil, sürekli gelişmeyi ve iyileştirmeyi içeren bir süreçtir.

5.2. Eğitim Kurumlarına Yönelik Öneriler

Eğitim kurumlarına yönelik öneriler kapsamında DevOps konusunda düşünülmesi gereken konular güncel müfredat, uygulamalı eğitim ve endüstri ilişkileri konularıdır. Bu konular:

- Güncel Müfredat: Eğitim kurumları, müfredatlarını DevOps alanındaki son gelişmeleri yansıtabak şekilde güncellemelidir. Öğrencilerin, bulut tabanlı çözümler,

konteyner teknolojileri ve otomasyon konularında eğitim görmeleri önemlidir. Eğitim programları, pratik uygulamalara ve proje çalışmalarına daha fazla yer vermelidir.

- Uygulamalı Eğitim : Eğitim kurumları, öğrencilere pratik uygulamalar yaptırmaya özen göstermelidir. Bu, öğrencilerin DevOps prensiplerini gerçek dünya senaryolarında uygulamalarına ve bu alanda pratik deneyim kazanmalarına olanak tanıyacaktır. Öğrencilere DevOps araçlarını kullanmayı, otomasyon süreçleri tasarlamayı ve ekip çalışması yapmayı öğretmek önemlidir.

- Endüstri İlişkileri : Eğitim kurumları, DevOps alanında çalışan şirketlerle işbirliği yapmalı ve öğrencilere sektör ile bağlantı kurma imkanı sağlamalıdır. Bu işbirliği, staj programları, konuşmacı davetleri ve sektör uzmanları ile atölye çalışmaları şeklinde gerçekleştirilebilir.

5.3. Politika Yapıcılara Yönelik Öneriler

Politika yapıcılara yönelik öneriler olarak iki temel başlık halinde düşünülebilir. Bu başlıklar:

- Destekleyici Politikalar: Politika yapıcılar, DevOps uygulamalarını destekleyici politikalar geliştirmelidir. Bu politikalar, DevOps alanında araştırma ve geliştirme çalışmalarını desteklemeli, öğrencilere DevOps eğitimi imkanı sağlamalı ve sektörde DevOps uygulamalarını teşvik etmelidir.

- DevOps Eğitimi Teşvik: Politika yapıcılar, DevOps alanında eğitim ve gelişim programlarını desteklemeli ve bu alanda çalışan bireyleri teşvik etmelidir. DevOps eğitimi için kaynak ayırmak ve DevOps alanında çalışan bireyler için kariyer gelişim fırsatları yaratmak önemlidir.

Bu çalışma, DevOps alanındaki gelişmelere ışık tutmaktadır ve sektörün geleceği için değerli bilgiler sunmaktadır. Bununla birlikte, DevOps uygulamalarının hızlı bir şekilde yaygınlaşması beklentisiyle, sektör profesyonellerinin, eğitim kurumlarının ve politika yapıcıların bu alanda daha derinlemesine çalışmaları ve daha etkin stratejiler geliştirmeleri önemlidir.

6. KAYNAKLAR

- Ahmad, A., Feng, C., Ge, S., & Yousif, A. (2018). A survey on mining stack overflow: Question and answering (Q&A) community. *Data Technologies and Applications*, 52(2), 190-247.
- Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors—A systematic literature review. *Information and Software Technology*, 107150.
- Baltadzhieva, A., & Chrupała, G. (2015, September). Predicting the quality of questions on stackoverflow. In *Proceedings of the international conference recent advances in natural language processing* (pp. 32-40).
- Barua, A., Thomas, S. W., ve Hassan, A. E., 2014. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3), 619-654.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993-1022.
- Bou Ghantous, G., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *PACIS2017*.
- Brunnert, A., van Hoorn, A., Willnecker, F., Danciu, A., Hasselbring, W., Heger, C., ... & Wert, A. (2015). Performance-oriented DevOps: A research agenda. *arXiv preprint arXiv:1508.04752*.
- Chen, X., Ji, Z., Fan, Y., ve Zhan, Y., 2017. Restful API architecture based on laravel framework. In *Journal of Physics: Conference Series* (Vol. 910, No. 1, p. 012016). IOP Publishing.
- Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *Ieee Software*, 33(3), 94-100.
- Ekinci, E. ve Omurca, S. İ., 2017. Ürün Özelliklerinin Konu Modelleme Yöntemi ile Çıkartılması. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 9(1), 51-58.
- Erich, F., Amrit, C., & Daneva, M. (2014). Report: Devops literature review. University of Twente, Tech. Rep.
- Fowler, M., & Foemmel, M. (2006). Continuous integration.
- Ghantous, G. B., & Gill, A. Q. (2019). An agile-devops reference architecture for teaching enterprise agile. *International Journal of Learning, Teaching and Educational Research*, 18(7), 128-144.

- Gürcan, F., & Özyurt, Ö. (2021). Stackoverflow gönderilerinde tartışılan trend konuların kelime frekans analizi ile belirlenmesi. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 11(2), 357-368. <https://doi.org/10.17714/gumusfenbil.811123>
- Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In *Proceedings of the scientific workshop proceedings of XP2016* (pp. 1-11).
- Jadhav N., 2014. Topic Models for Sentiment analysis: A Literature Survey, *Teknik Rapor*, 2014, pp. 1-11.
- Joorabchi, A., English, M., & Mahdi, A. E. (2016). Text mining stackoverflow: An insight into challenges and subject-related difficulties faced by computer science learners. *Journal of Enterprise Information Management*, 29(2), 255-275.
- Khan, A. A., Khan, J. A., Akbar, M. A., Zhou, P., & Fahmideh, M. (2023). Insights into Software Development Approaches: Mining Q&A Repositories. *arXiv preprint arXiv:2305.01315*.
- Khan, M. S. A., Farabi, A. R., ve Iqbal, A., 2022. What Do Firebase Developers Discuss About? An Empirical Study on Stack Overflow Posts. In *2022 9th International Conference on Networking, Systems and Security* (pp. 63-74).
- Khan, S. U., Khan, A. W., Khan, F., Khan, J., & Lee, Y. (2023). Factors influencing vendor organizations in the selection of DevOps for global software development: an exploratory study using a systematic literature review. *Cognition, Technology & Work*, 1-16.
- Maximilien, E. M., & Williams, L. (2003, May). Assessing test-driven development at IBM. In *25th International Conference on Software Engineering*, 2003. *Proceedings*. (pp. 564-569). IEEE.,
- Moutidis, I., & Williams, H. T. (2021). Community evolution on stack overflow. *Plos one*, 16(6), e0253010.
- Rosen, C., & Shihab, E. (2016). What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering*, 21, 1192-1223.
- Rüdiger, M., Antons, D., Joshi, A. M., & Salge, T. O. (2022). Topic modeling revisited: New evidence on algorithm performance and quality metrics. *Plos one*, 17(4), e0266325.
- Sánchez-Cifo, M. Á., Bermejo, P., & Navarro, E. (2023). DevOps: Is there a gap between education and industry?. *Journal of Software: Evolution and Process*, e2534.
- Skenderi, E., Laaksonen, S. M., Stefanidis, K., & Huhtamäki, J. (2023, January). Assessing Text Representation Methods on Tag Prediction Task for StackOverflow. In *Hawaii International Conference on System Sciences* (pp. 585-594).
- Stevyers, M. ve Griffiths, T., 2007. Probabilistic Topic Models, *Handbook of Latent Semantic Analysis*. Editör: Landauer, T., McNamara, D.S. , Dennis, S., Kintsch W. Erlbaum, 2007.

- Squire, M., 2015. Should We Move to Stack Overflow? Measuring the Utility of SocialMedia for Developer Support. IEEE International Conference on Software Engineering Vol 2 (s. 219-228). IEEE.
- Şahin, A.S., 2022. Yazılım içerikli soru-cevap sitelerine metin madenciliğinin uygulanması.
- Tran, K. (2023, Ekim 24) pyLDavis: Topic Modelling Exploration Tool That Every NLP Data Scientist Should Know.: <https://neptune.ai/blog/pyldavis-topic-modelling-exploration-tool-that-every-nlp-data-scientist-should-know> adresinden alınmıştır.
- URL-1. (2022, Aralık 29) DevOps arama sonuçları.: <https://stackoverflow.com/search?q=devops> adresinden alınmıştır.
- URL-2. (2023, Mart 21) What is DevOps?: <https://aws.amazon.com/tr/DevOps/what-is-DevOps/> adresinden alınmıştır.
- URL-3. (2023, Mart 23) DevOps yaşam döngüsü.: <https://teknoreg.com/yazilim-harikalari-test-yazilimi-DevOps-nedir/> adresinden alınmıştır.
- URL-4. (2023, Eylül 13) parallelized Latent Dirichlet Allocation.: <https://radimrehurek.com/gensim/models/ldamulticore.html> adresinden alınmıştır.
- Vassallo, C., Zampetti, F., Romano, D., Beller, M., Panichella, A., Di Penta, M., & Zaidman, A. (2016, October). Continuous delivery practices in a large financial organization. In 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 519-528). IEEE.
- Watts, S. (2024, Ocak 21) Google/DORA's 2023 Accelerate State of DevOps Report.: https://www.splunk.com/en_us/blog/learn/state-of-devops.html adresinden alınmıştır.
- Witten, I. H., 2004. Text Mining.
- Yang, M., Zhou, Y., Li, B., & Tang, Y. (2023). On Code Reuse from StackOverflow: An Exploratory Study on Jupyter Notebook. arXiv preprint arXiv:2302.11732.
- Yıldız, D., 2022. Bilgi Sistemi Yazılım Geliştirme Yaşam Döngüsü Safhalarından Gereksinim Belirleme ve Sistem Tasarımında Kalite Odaklılık: Üç Proje İncelemesi. Bilişim Teknolojileri Dergisi, 15(1), 55-64.

ÖZGEÇMİŞ

İlköğrenimini Zonguldak'ta, lise eğitimini Rize'de tamamladı.

Akademik geçmişi:

- Marmara Üniversitesi, Bilgisayar ve Kontrol Öğretmenliği (İngilizce)
(2005-2010)
- Karadeniz Teknik Üniversitesi, Yazılım Mühendisliği
(2019-2021)
- Karadeniz Teknik Üniversitesi, Yazılım Mühendisliği Yüksek Lisans
(2021-2024)

İş Tecrübesi:

- 2010-2015 yılına kadar farklı alanlarda iş yapan teknoloji ve yazılım şirketlerinde yazılım geliştirme ve bakım onarım görevleri yaptı.
- 2015 yılından günümüze, Milli Eğitim Bakanlığı'nda Teknik Öğretmen kadrosunda Bilişim Teknolojileri öğretmenliği yapmaktadır.

İyi derece İngilizce bilmektedir.

Yayınlar

BAKIRCI, B., & ÖZYURT, Ö. StackOverflow Üzerinden Yapılan DevOps Paylaşımlarının Etiket ve n-Gram Analizi: Yazılım Geliştirme Süreçlerindeki Uygulama Alanları ve Etkileri: 4th International Symposium of Scientific Research and Innovative Studies (ISSRIS'24) (317-327).