



Teknoloji Fakültesi

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# Derin öğrenme ve Makine öğrenme ile veri tabanlarında aykırılıklar tespit etme ve erken müdahale sistemi

Aisan KHEIRI

Kaan Kürşat KILIÇ

"

Bitirme Projesi Raporu

Bilgisayar Mühendisliği Bölümü

DANIŞMAN

Doç. Dr. Buket DOĞAN

İSTANBUL, 2025

"

# İÇİNDEKİLER

<b>1. GİRİŞ</b>	<b>8</b>
<b>1.1. Proje Çalışmasının Amacı ve Önemi</b>	<b>8</b>
▪ 1.1.1 Çalışmanın Amacı	8
▪ 1.1.2 Çalışmanın Önemi	9
▪ 1.1.3 İlgili Çalışmalar	10
<b>1.2. Benchmark Çalışmaları</b>	<b>14</b>
▪ 1.2.1. Patent Araştırması	14
▪ 1.2.2. Patent Araştırma Sonuçları	15
▪ 1.2.3. Sonuç ve Değerlendirme	15
<b>2. MATERYAL VE YÖNTEM</b>	<b>16</b>
<b>2.1. Veri Seti ve Özellikleri</b>	<b>16</b>
▪ 2.1.1. Veri Setinin İçeriği	16
▪ 2.1.2. Veri Ön İşleme ve Temizleme	19
▪ 2.1.3. Veri Setinin Kullanım Amacı	20
▪ 2.1.4 Veri Seti Örneği	20
<b>2.2. Makine Öğrenmesi ve Derin Öğrenme Algoritmaları</b>	<b>21</b>
▪ 2.2.1 Makine Öğrenmesi Algoritmaları ve Hiperparametreler	21
▪ 2.2.2 Kullanılan Makine Öğrenmesi Sonuçları	24
▪ 2.2.3 Kullanılan Derin Öğrenme Algoritmaları	25
▪ 2.2.4 Kullanılan Derin Öğrenme Hiper Parametreleri	26
▪ 2.2.5 Kullanılan Derin Öğrenme Sonuçları	26
<b>3. WEB SİTESİ GELİŞTİRME VE KULLANILAN TEKNOLOJİLER</b>	<b>28</b>
<b>3.1. Web Sitesinin Genel Yapısı</b>	<b>28</b>
<b>3.2. Kullanılan Teknolojiler ve Araçlar</b>	<b>38</b>

<b>3.3. Veri Tabanı ve Tablolar</b>	<b>40</b>
<b>3.4. Web Sitesine Ait Ekran Kaydı</b>	<b>41</b>
<b>4. BULGULAR VE TARTIŞMA</b>	<b>41</b>
<b>5. SONUÇ</b>	<b>43</b>

## ÖZET

Bu çalışma, veri tabanı sistemlerinde ortaya çıkabilecek olası sorunları önceden tespit ederek, otomatik müdahale mekanizmaları geliştirmeyi hedeflemektedir. Bu doğrultuda, makine öğrenimi ve derin öğrenme tabanlı anomali tespit yöntemleri kullanılmaktadır. Araştırma kapsamında sürekli toplanan telemetri verileri detaylı bir şekilde analiz edilerek ön işleme aşamalarından geçirilmiş ve değerlendirilmektedir. Veri tabanlarında oluşabilecek hataları erken tespit edebilmek için yerli bir yazılım çözümü geliştirilmesi hedeflenmiştir.

Ayrıca, bu sistemin bir web sitesi üzerinden erişilebilir olması planlanarak kullanıcıların model eğitim sürecine dahil edilmesi amaçlanmaktadır. Kullanıcılar, hiperparametreleri belirleyerek model eğitimi gerçekleştirebilecek ve sonuçları değerlendirebilecektir. Çalışma kapsamında veri toplama, ön işleme, analiz ve model değerlendirme süreçleri ayrıntılı bir şekilde ele alınmış, farklı makine öğrenimi ve derin öğrenme algoritmalarının performans karşılaştırmaları yapılmıştır.

Sonuç olarak, bu çalışma, veri tabanı yönetim süreçlerini optimize ederek verimliliği artırmayı ve gelecekteki otomasyon sistemlerine katkı sağlamayı amaçlamaktadır. Projenin uzun vadeli hedefleri arasında, veri tabanlarını daha verimli hale getirerek sorgu maliyetlerini en aza indirmek ve sistemin, sunucu içindeki gerekli tüm aksiyonları otomatik olarak gerçekleştirebilir duruma gelmesini sağlamak yer almaktadır. Ayrıca, elde edilen analiz sonuçlarının raporlanarak sistematik bir şekilde saklanması planlanmaktadır.

Anahtar Kelimeler: Veri Tabanı Yönetimi, Anomali Tespiti, Telemetri Analizi, Veri Madenciliği, Büyük Veri, Makine Öğrenmesi Modelleri, Derin Öğrenme Algoritmaları, Hiperparametre Optimizasyonu, Yapay Zeka Tabanlı Analiz, Veri Ön İşleme Teknikleri, Model Performans Ölçümü, Otomatik Anomali Algılama, Web Tabanlı Veri Analizi, Sunucu Sağlık Kontrolü, Akıllı Veri Yönetimi.

**Haziran, 2025**

**Aisan KHEIRI, Kaan Kürşat KILIÇ**

## **ABSTRACT**

This study aims to detect potential issues in database systems in advance and develop automated intervention mechanisms. In this context, anomaly detection methods based on machine learning and deep learning are utilized. As part of the research, continuously collected telemetry data is thoroughly analyzed, preprocessed, and evaluated. A domestic software solution is intended to be developed to detect possible database errors at an early stage.

Additionally, making this system accessible via a website is planned, allowing users to participate in the model training process. Users will be able to define hyperparameters, train models, and evaluate the results. The study extensively covers data collection, preprocessing, analysis, and model evaluation processes while comparing the performance of various machine learning and deep learning algorithms.

As a result, this study aims to optimize database management processes, enhance efficiency, and contribute to future automation systems. The long-term objectives of the project include improving database efficiency, minimizing query costs, and enabling the system to autonomously execute all necessary actions within the server. Furthermore, it is planned that the obtained analysis results will be systematically recorded and reported.

**Keywords:** Database Management, Anomaly Detection, Telemetry Analysis, Data Mining, Big Data, Machine Learning Models, Deep Learning Algorithms, Hyperparameter Optimization, AI-Based Analysis, Data Preprocessing Techniques, Model Performance Evaluation, Automated Anomaly Detection, Web-Based Data Analysis, Server Health Monitoring, Intelligent Data Management.

**June, 2025**

**Aisan KHEIRI, Kaan Kürşat KILIÇ**

## KISALTMALAR

<b>DBMS</b>	: Database Management System
<b>DBA</b>	: Database Administrator
<b>LLM</b>	: Large Language Model
<b>ML</b>	: Machine Learning
<b>DL</b>	: Deep Learning
<b>AI</b>	: Artificial Intelligence
<b>SQL</b>	: Structured Query Language
<b>RDBMS</b>	: Relational Database Management System
<b>API</b>	: Application Programming Interface
<b>GPU</b>	: Graphics Processing Unit
<b>TPU</b>	: Tensor Processing Unit
<b>ORM</b>	: Object-Relational Mapping
<b>FPDF</b>	: Free PDF Library for Python
<b>RNN</b>	: Recurrent Neural Network
<b>LSTM</b>	: Long Short-Term Memory
<b>SVM</b>	: Support Vector Machine
<b>KNN</b>	: K-Nearest Neighbors
<b>Auto ML</b>	: Automated Machine Learning

## ŞEKİL LİSTESİ

Şekil 2.1: <i>Veri Setinin İlk Beş Satırı</i> .....	20
Şekil 3.1: <i>Kullanıcı Giriş ve Kayıt Ekranı</i> .....	28
Şekil 3.2: <i>Şirket Gurubunun Belirlenmesi</i> .....	29
Şekil 3.3: <i>Dosya Yükleme ve Algoritma Seçme Alanı</i> .....	31
Şekil 3.4: <i>Seçilen Algoritmaya Göre Hiper parametre Girişi</i> .....	31
Şekil 3.5: <i>Eğitim Sürecinin İlerlemesi Adımı</i> .....	32
Şekil 3.6: <i>Veri Seti İçeriği Tablo Halinde</i> .....	32
Şekil 3.7: <i>Grafiğin ChatGpt Tarafından Yorumlanması</i> .....	33
Şekil 3.8: <i>Veri Setinin İstatiksel Özellikleri</i> .....	33
Şekil 3.9: <i>Veri Setinin İlişkiler Tablosu</i> .....	33
Şekil 3.10: <i>PLE Değeri için Etkili Kolonlar ve ChatGpt Yorumu</i> .....	34
Şekil 3.11: <i>Veri Setinin Sütun Bazında İstatiksel Özeti</i> .....	34
Şekil 3.12: <i>Veri Setinin PLE değerlerine Ait Pasta Grafiği</i> .....	34
Şekil 3.13: <i>Modelin Performans Ölçümü ve Karmaşıklık Matrisi</i> .....	35
Şekil 3.14: <i>Tüm Sonuçların Raporlandığı Pdf Butonu</i> .....	35
Şekil 3.15: <i>Pdf Raporuna Ait Görsel</i> .....	36
Şekil 3.16: <i>Kullanıcının Oluşturduğu Tüm Raporların Tutulduğu Ekran</i> .....	36
Şekil 3.17: <i>Telegram Bot ve Anlık Bildirim Sistemi</i> .....	36
Şekil 3.18: <i>Yeni Kullanıcı İçin Oluşturulan Kullanım Kılavuzu Örneği</i> .....	37
Şekil 3.19: <i>Adminlerin Gözüktüğü Sekme</i> .....	37
Şekil 3.20: <i>Kullanıcıların Gözüktüğü Sekme</i> .....	38
Şekil 3.21: <i>Kullanıcıların Oluşturduğu Raporlar Sekmesi</i> .....	38

## TABLO LİSTESİ

Tablo 1.1: <i>Literatür Tarama</i> .....	14
Tablo 2.1: <i>Üç Algoritmaya Ait Model Eğitim Sonuçları</i> .....	24
Tablo 2.2: <i>İki Algoritmaya Ait Model Parametreleri</i> .....	26
Tablo 2.3: <i>Derin Öğrenmede İki Algoritmaya Ait Model Eğitim Sonuçları</i> .....	27



## 1. GİRİŞ

Günümüzde veri tabanı yönetim sistemleri, büyük ölçekli şirketlerden kamu kuruluşlarına kadar birçok alanda kritik bir rol oynamaktadır. Veri tabanlarının kesintisiz ve hatasız çalışması, veri güvenliği, sistem performansı ve iş sürekliliği açısından büyük önem taşımaktadır. Ancak, sistemlerde meydana gelebilecek anormal durumlar (aykırılıklar), performans düşüşlerine, veri kayıplarına veya güvenlik tehditlerine yol açabilmektedir. Geleneksel yöntemler genellikle statik kural tabanlı tespit sistemlerine dayanmakta olup, yeni ve bilinmeyen anomalileri algılamakta yetersiz kalmaktadır. Bu nedenle, makine öğrenmesi (ML) ve derin öğrenme (DL) tabanlı otomatik tespit ve müdahale mekanizmalarına duyulan ihtiyaç giderek artmaktadır.

Bu çalışmada, makine öğrenmesi ve derin öğrenme teknikleri kullanılarak, veri tabanlarında meydana gelen anormalliklerin tespit edilmesi ve otomatik müdahale mekanizmalarının geliştirilmesi amaçlanmaktadır. Sistem, veri tabanı telemetri verilerini sürekli olarak analiz ederek, anomalileri erken aşamada tespit edecek ve proaktif önlemler alacaktır. Makine öğrenmesi algoritmaları, geçmiş verilere dayalı istatistiksel analiz yaparak anomali modellerini belirlerken; derin öğrenme teknikleri, daha karmaşık ve öngörülemeyen anomalileri tespit etmek için gelişmiş yapay sinir ağları ile çalışacaktır.

### 1.1. Proje Çalışmasının Amacı ve Önemi

#### ▪ 1.1.1 Çalışmanın Amacı

Bu çalışma, makine öğrenmesi (ML) ve derin öğrenme (DL) tekniklerini kullanarak veri tabanlarında meydana gelen anomalileri tespit etmek ve erken müdahale mekanizmaları geliştirmek amacıyla yürütülmektedir. Veri tabanı sistemleri, günümüz dijital dünyasında büyük ölçekli işletmelerden kamu kurumlarına kadar birçok alanda kritik öneme sahiptir. Ancak, veri tabanlarında oluşabilecek aykırılıklar (anomaliler), veri bütünlüğünü, performansı ve sistem güvenilirliğini tehdit edebilir. Geleneksel veri tabanı yönetim sistemleri (DBMS), statik kural tabanlı yöntemlere dayanırken, dinamik ve değişen veri yapıları karşısında esneklikleri sınırlıdır.

Bu çalışmada, makine öğrenmesi ve derin öğrenmenin birlikte kullanıldığı hibrit bir sistem geliştirilerek, veri tabanlarında otomatik anomali tespiti ve erken müdahale mekanizmaları

oluşturulması hedeflenmektedir.

- Makine öğrenmesi algoritmaları ile geçmiş verilere dayalı olarak veritabanı anormalliklerini öngörmek,
- Derin öğrenme modelleri ile karmaşık ve öngörülemeyen anomalileri daha yüksek doğrulukla tespit etmek,

#### ▪ 1.1.2 Çalışmanın Önemi

Veri tabanlarında meydana gelen anomaliler ve sistem hataları, sistem kesintilerine, veri kaybına ve güvenlik açıklarına yol açabilmektedir. Geleneksel veri tabanı yönetim sistemleri, bu tür sorunları tespit etmek ve müdahale etmek için kural tabanlı yaklaşımlar ve manuel DBA (Database Administrator) müdahalelerine bağımlıdır. Ancak, günümüzün büyük ölçekli veri sistemleri için manuel yönetim süreçleri zaman alıcı ve yetersizdir.

Bu çalışmanın önemi aşağıdaki faktörlerle özetlenebilir:

1. Veri tabanı sistemlerinde gerçek zamanlı anomali tespiti: Geleneksel sistemler genellikle anomalileri geç tespit eder ve çoğu zaman müdahale edemez. Bu sistem, otomatik ve hızlı bir şekilde anomalileri tespit edip müdahale etme yeteneğine sahip olacaktır.
2. Makine öğrenmesi ve derin öğrenme tekniklerinin hibrit kullanımı: Çalışma, ML tabanlı sınıflandırma modelleri ile DL tabanlı zaman serisi analiz yöntemlerini entegre ederek daha yüksek doğruluk oranına sahip bir tespit mekanizması sunacaktır.
3. Endüstri ve akademik alanda katkı sağlaması: Geliştirilecek sistem, kurumsal veri tabanı yönetim sistemleri, finans, sağlık, e-ticaret ve büyük veri yönetimi gibi birçok alanda kullanılabilir. Ayrıca, yapay zeka destekli veri tabanı yönetimi alanında akademik katkı sunacaktır.
4. Bu çalışmanın çıktıları, günümüzün büyüyen veri sistemlerinin daha güvenli, ölçeklenebilir ve otonom hale gelmesine katkı sağlayacak ve gelecekteki yapay zeka destekli veri tabanı yönetim sistemlerine temel oluşturacaktır.

### ▪ 1.1.3 İlgili Çalışmalar

Veri tabanı yönetim sistemlerinde aykırılık tespiti, hata tahmini ve optimizasyon konularında birçok çalışma yapılmış olup, bu çalışmalar genellikle makine öğrenmesi, derin öğrenme ve büyük dil modelleri (LLM) gibi yöntemleri içermektedir. Bu bölümde, önerilen çalışmaya benzer araştırmalar ele alınarak kullanılan yöntemler, temel bulgular ve eksik yönler tartışılacaktır.

Makale Başlığı	Yazar(lar)	Yayın Tarihi	Yayın Künyesi	Özet
D-Bot: Database Diagnosis System using Large Language Models [1]	Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., ... & Zeng, G.	2023	Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., ... & Zeng, G. (2023). D-bot: Database diagnosis system using large language models. arXiv preprint arXiv:2312.01454.	Bu çalışma, büyük dil modellerinin (LLM) veritabanı teşhis süreçlerinde kullanılmasını amaçlayan D-Bot adlı bir sistem önermektedir. D-Bot, otomatik teşhis ipuçları üreterek, kök neden analizleri yaparak ve işbirlikçi teşhis mekanizmaları kullanarak veri tabanı yönetim süreçlerini optimize eder. Gerçek dünya verileriyle yapılan testler, D-Bot'un geleneksel DBA yöntemlerine kıyasla çok daha hızlı ve doğru sonuçlar verdiğini göstermektedir.
A Machine Learning Approach to Database Failure Prediction [2]	Karakurt, İ., Özer, S., Ulusinan, T., & Ganiz, M. C.	2017	Karakurt, İ., Özer, S., Ulusinan, T., & Ganiz, M. C. (2017). A machine learning approach to database failure prediction. In 2017	Bu çalışmada, Oracle veri tabanı sistemlerindeki olası hataların önceden tahmin edilmesi için makine öğrenmesi

			International Conference on Computer Science and Engineering (UBMK) (pp. 1030-1035). IEEE.	teknikleri kullanılmıştır. Destek Vektör Makineleri (SVM), Naïve Bayes, Lojistik Regresyon, Random Forest ve AdaBoost gibi algoritmalar test edilmiş ve Random Forest %98.37 doğruluk ile en iyi performansı göstermiştir. Çalışma, veri tabanı log verilerinden anormal olayları tahmin ederek şirketlerin arıza yönetimini iyileştirmesine katkı sağlamaktadır.
DB-GPT: Large Language Model Meets Database [3]	Zhou, X., Sun, Z., & Li, G.	2024	Zhou, X., Sun, Z., & Li, G. (2024). Db-gpt: Large language model meets database. Data Science and Engineering, 9(1), 102-111.	DB-GPT, büyük dil modellerinin veri tabanı sorgu optimizasyonu, indeks ayarı ve sorgu yeniden yazımı gibi görevlerde nasıl kullanılabileceğini araştırmaktadır. GPT-3.5 tabanlı DB-GPT modeli, PostgreSQL'in yerleşik optimizasyon mekanizmasına kıyasla %22.4 daha iyi sorgu optimizasyonu ve %9.8 daha iyi indeks optimizasyonu

				sağlamıştır. Bu çalışma, LLM'lerin veri tabanı yönetiminde nasıl kullanılabileceğini ortaya koyarak yeni nesil otomatik veritabanı optimizasyon sistemlerinin temelini atmaktadır.
Future Trends in SQL Databases and Big Data Analytics [4]	Islam, S.	2024	Islam, S. (2024). Future Trends In SQL Databases And Big Data Analytics: Impact of Machine Learning and Artificial Intelligence. Available at SSRN 5064781.	Bu makale, SQL veri tabanları ile büyük veri analitiğinin kesişim noktalarını ve yapay zeka (AI) ile makine öğrenmesi (ML) tekniklerinin veri yönetimi süreçlerini nasıl dönüştürdüğünü incelemektedir. Makine öğrenmesi destekli sorgu optimizasyonu, AI tabanlı güvenlik sistemleri, SQL ve NoSQL hibrit entegrasyonları gibi konular ele alınmıştır. Çalışma, büyük veri çağında SQL sistemlerinin daha ölçeklenebilir ve verimli hale gelmesi için yeni teknolojilerin nasıl entegre edilebileceğini göstermektedir.
LLM for Data Management	Li, G., Zhou, X., & Zhao,	2024	Li, G., Zhou, X., & Zhao, X. (2024).	Bu çalışma, büyük dil modellerinin

[5]	X.		LLM for Data Management. Proceedings of the VLDB Endowment, 17(12), 4213-4216.	(LLM) veri yönetimi süreçlerinde nasıl kullanılabileceğini analiz etmektedir. Geleneksel makine öğrenmesi yöntemlerine kıyasla LLM'lerin bağlamsal anlama yetenekleri sayesinde veri analitiği, sorgu optimizasyonu ve hata teşhisi gibi alanlarda daha başarılı sonuçlar verebileceği belirtilmektedir. Önerilen sistem, RAG (Retrieval-Augmented Generation) teknikleri ve vektör veri tabanları ile desteklenerek, LLM'lerin doğruluk oranını artırmayı hedeflemektedir.
Machine Learning for Database Management and Query Optimization [6]	Fahima, M. M. F., Sreen, A. S., Ruksana, S. F., Weihena, D. T. E., & Majid, M. H. M.	2024	Fahima, M. M. F., Sreen, A. S., Ruksana, S. F., Weihena, D. T. E., & Majid, M. H. M. (2024). Machine Learning for Database Management and Query Optimization. Elementaria: Journal of Educational Research, 2(1), 96-108.	Makine öğrenmesi tekniklerinin veri tabanı yönetimi ve sorgu optimizasyonu alanında nasıl uygulanabileceğini araştıran bu çalışma, iş yükü yönetimi, indeksleme ve veri kalitesi sağlama gibi konulara odaklanmaktadır. Derin öğrenme, pekiştirmeli öğrenme ve doğal

				dil işleme tekniklerinin, veri tabanı yönetim sistemlerinde verimliliği artırma potansiyeli vurgulanmaktadır. Çalışma, özellikle dinamik sorgu optimizasyonu ve otonom indeksleme tekniklerine odaklanarak, geleneksel yöntemlerden daha iyi sonuçlar elde edilebileceğini göstermektedir.
--	--	--	--	--

Tablo 1.1: *Literatür Tarama*

## 1.2. Benchmark Çalışmaları

### ▪ 1.2.1. Patent Araştırması

Bu çalışmada, veri tabanı yönetimi ve yapay zeka konularında mevcut patentlerin varlığını araştırmak amacıyla çeşitli ulusal ve uluslararası patent veri tabanlarında aramalar gerçekleştirilmiştir. Araştırmada, aşağıdaki anahtar kelimeler kullanılmıştır:

- "database and deep learning"
- "database optimization"
- "database and machine learning"
- "database artificial intelligence"
- "veri tabanı ve yapay zeka"
- "veri tabanı"
- "veri tabanı ve derin öğrenme"
- "veri tabanı ve makine öğrenmesi"

Araştırma, ABD Patent ve Ticari Marka Ofisi (USPTO), Espacenet (Avrupa Patent Ofisi), Google Patents ve Türk Patent ve Marka Kurumu veri tabanlarında gerçekleştirilmiştir.

### ▪ 1.2.2. Patent Araştırma Sonuçları

<b>USPTO</b>	<b>Patent</b>	<b>Veri</b>	<b>Tabanı:</b>
USPTO'nun Artificial Intelligence Patent Dataset (AIPD) 2023 güncellenmiş veri seti incelenmiş ve yukarıda belirtilen anahtar kelimelerle ilgili doğrudan bir patente rastlanmamıştır. Ancak, AI tabanlı veritabanı yönetimi ve optimizasyon süreçleri ile ilgili bazı genel patentler bulunmaktadır.			

<b>Espacenet</b>	<b>Patent</b>	<b>Veri</b>	<b>Tabanı:</b>
Espacenet üzerinden yapılan aramalarda, "database and deep learning", "database and machine learning" gibi anahtar kelimeler kullanılarak yapılan sorgular sonucunda doğrudan bir veri tabanı ve yapay zeka entegrasyonu üzerine alınmış patent tespit edilememiştir.			

<b>Google</b>	<b>Patents</b>	<b>Veri</b>	<b>Tabanı:</b>
Google Patent üzerinde yapılan aramalarda, veri tabanı optimizasyonu ve yapay zeka tabanlı veri tabanı yönetimi ile ilgili bazı başvuruların olduğu görülmüş, ancak bu başvuruların doğrudan büyük dil modelleri (LLM) veya derin öğrenme tabanlı veri tabanı teşhisi ve optimizasyonu ile ilgili olmadığı anlaşılmıştır.			

<b>Türk</b>	<b>Patent</b>	<b>ve</b>	<b>Marka</b>	<b>Kurumu:</b>
Türk Patent ve Marka Kurumu veri tabanında "veri tabanı ve yapay zeka", "veri tabanı ve makine öğrenmesi" gibi Türkçe ve İngilizce anahtar kelimelerle araştırma yapılmış, ancak konuya ilişkin doğrudan bir patent bulunamamıştır.				

### ▪ 1.2.3. Sonuç ve Değerlendirme

Yapılan benchmark çalışmaları sonucunda, veri tabanı yönetim sistemleri ve yapay zeka entegrasyonu ile ilgili patentlerin sınırlı olduğu görülmektedir. Mevcut patentler genellikle genel veri tabanı optimizasyon teknikleri ve yapay zeka ile ilgili genel çözümler üzerine yoğunlaşmıştır. Derin öğrenme (deep learning) ve büyük dil modelleri (LLM) ile veri tabanı teşhisi, optimizasyonu ve yönetimi konularında henüz kapsamlı bir patent başvurusunun olmadığı tespit edilmiştir.



## 2. MATERYAL VE YÖNTEM

### 2.1. Veri Seti ve Özellikleri

Bu çalışmada kullanılan veri seti, Data Platform şirketinin gerçek verilerini içermektedir ve veri tabanı performans analizine yönelik çeşitli metrikleri kapsamaktadır. Veri seti, zaman serisi tabanlı ölçümler içerdiğinden, veri tabanı yönetimi süreçlerinin performansını değerlendirmek ve optimizasyon stratejileri geliştirmek için analiz edilmiştir.

#### ▪ 2.1.1. Veri Setinin İçeriği

Veri setinde 36 farklı değişken bulunmaktadır. Bu değişkenler, veri tabanının çalışma durumu, performans metrikleri ve yönetim süreçleri ile ilgili bilgiler içermektedir. Öne çıkan bazı değişkenler şunlardır:

Sütun Adı	Açıklama
InstanceID	SQL Server sisteminden alınan verilerin hangi sunucu (veya örnek/instance) üzerinden geldiğini belirtir. Sistemleri birbirinden ayırmak için kullanılır.
CreateDate	Bu satırdaki verilerin hangi tarih ve saatte toplandığını belirtir. Yani sistem performansı o anda nasılmış, bu bilgiyi verir.
page life expectancy	SQL Server'ın belleğinde (RAM) bulunan verilerin ne kadar süre bellekte tutulduğunu gösterir. Düşükse, RAM yetersizdir ve sistem daha yavaş çalışabilir.
% privileged time	CPU'nun işletim sistemi seviyesinde (yani kullanıcı değil sistem modunda) ne kadar süre çalıştığını gösterir. Yüksekse donanım sıkıntısı veya sürücü problemi olabilir.
transactions/sec	Sistemde saniyede kaç işlem gerçekleştiğini gösterir. Bu, genel işlem yoğunluğunu belirtir.

Sütun Adı	Açıklama
<b>write transactions/sec</b>	Saniyede kaç tane yazma işlemi (insert, update, delete gibi) yapıldığını gösterir. Yüksek değerler diske yazma yoğunluğu olduğunu gösterir.
<b>logical connections</b>	Veri tabanına anlık olarak kaç bağlantı olduğunu belirtir. Yani kaç kullanıcı veya uygulama bağlanmış.
<b>dbaHCDOC</b>	HCDOC adlı bir kontrol veya sağlık kontrol prosedürünün çalıştırılıp çalıştırılmadığını gösterir.
<b>dbaShrinkLog</b>	Log dosyalarının "shrink" işlemi (boyut küçültme) ile temizlenip temizlenmediğini belirtir. Disk alanı kazanımı için yapılır.
<b>dbaHCDOCexportTxt</b>	HCDOC verilerinin dışa metin (.txt) olarak aktarılıp aktarılmadığını gösterir.
<b>dbaCheckPrimaryNode</b>	Sunucu kümeleme (AlwaysOn gibi) yapısında, sistemin birincil (primary) düğümde olup olmadığını kontrol eder.
<b>dbaTürkiyeFinansEventLogDelete</b>	Türkiye Finans sistemine özel olarak olay (event) loglarının silinip silinmediğini gösterir.
<b>dbaBackupLog</b>	Veri tabanı log dosyalarının yedeklenip yedeklenmediğini belirtir. Loglar düzenli yedeklenmezse veri kaybı riski vardır.
<b>dbaSessionKiller</b>	Uzun süre işlem yapmadan açık kalan bağlantıların otomatik olarak sonlandırılıp sonlandırılmadığını belirtir. Performansı korumak için kullanılır.
<b>dbafullbackup</b>	Veri tabanının tam (full) yedeğinin alınıp alınmadığını gösterir. Günlük veya haftalık

<b>Sütun Adı</b>	<b>Açıklama</b>
	yapılması gereken temel bir bakım işlemidir.
<b>dbaMaintenanceReIndex</b>	Veri tabanındaki index'lerin yeniden düzenlenip düzenlenmediğini belirtir. Performansı artırmak için yapılır.
<b>Maintenance</b>	Genel bakım işlemlerinin (yedeğe, index temizliği, log temizliği gibi) planlandığı şekilde çalışıp çalışmadığını gösterir.
<b>dbaHCDOCcontrol</b>	HCDOC işlemlerinin kontrol ve doğrulama aşamasının başarıyla tamamlanıp tamamlanmadığını belirtir.
<b>dbaHCDOCHadrControl</b>	HCDOC işlemlerinin yüksek erişilebilirlik sistemleri (AlwaysOn, HADR) ile entegre kontrolünün yapılma durumunu belirtir.
<b>syspolicy_purge_history</b>	SQL Server'ın kendi güvenlik ve yapılandırma politikalarına ait geçmiş log kayıtlarının temizlenip temizlenmediğini belirtir.
<b>dbaCheckListenerStatus</b>	SQL Server'ın dinleyici (listener) servisinin çalışıp çalışmadığını kontrol eder. Bu servis dış bağlantılar için gereklidir.
<b>dbaFullAylıkBackup</b>	Veri tabanının aylık tam yedeğinin alınma durumunu belirtir. Stratejik yedekleme politikalarının bir parçasıdır.
<b>dbaFibabankEventLogDelete</b>	FibaBank'a ait sistemlerde olay loglarının temizlenip temizlenmediğini belirtir.
<b>dbaFibadataShrink</b>	FibaBank sisteminde veri/log dosyalarının küçültme (shrink) işleminin yapılıp yapılmadığını

Sütun Adı	Açıklama
	belirtir.
<b>dbaCheckDBSync</b>	Veri tabanı replikasyonu veya yedek sunucular ile senkronizasyonun kontrol edilip edilmediğini belirtir.

Aşağıdaki sütunlar tabloda yer alan başlıca alanlardır:

- **InstanceID**
- **CreateDate**
- **Page Life Expectancy**
- **% Privileged Time**
- **Transactions/sec**
- **Write Transactions/sec**
- **Logical Connections**

Bu alanların dışındaki sütunlar ise, ilgili tarih ve saatteki job'ın başarılı bir şekilde çalışıp çalışmadığını göstermektedir.

#### ▪ 2.1.2. Veri Ön İşleme ve Temizleme

Bu proje kapsamında veri ön işleme adımları çok katmanlı olarak gerçekleştirilmiştir. Aşağıda sıralanan teknik adımlar sayesinde, hem makine öğrenmesi hem de derin öğrenme modelleri için dengeli ve anlamlı veri sağlanmıştır:

- Ham veri, dbaHCDOC, dbaHCDOCexportTxt gibi stored procedure'ler ile DOCperformanceMonitor, DOCjobStatus, DOCintegration tablolarından toplanmıştır.
- 40.000'den fazla sıfır değer içeren sütunlar temizlenerek noise reduction uygulanmıştır.
- Eksik timestamp'ler ve hatalı tarih formatları ayıklanmıştır.
- FP-Growth ve Apriori algoritmalarıyla iş parçacığı ilişkileri modellenmiş, yeni özellikler türetilmiştir.
- Page Life Expectancy (PLE) değerine göre [0–300], [301–5000], [5001+] aralıklarında

etiketleme yapılmıştır.

- Sınıf dengesini sağlamak için oversampling ve undersampling teknikleri kullanılmıştır.
- LSTM ve RNN gibi modeller için tüm sayısal veriler 0–1 aralığına normalize edilmiştir (MinMaxScaler).
- Eğitim-test ayrımı %80-%20 oranında yapılmış, lookback penceresi 240 olarak belirlenmiştir.

Bu kapsamlı ön işleme süreci, sistem performans tahminlerinin doğruluğunu ve model genel başarımını artırmaya yöneliktir.

### ▪ 2.1.3. Veri Setinin Kullanım Amacı

Bu veri seti, aşağıdaki amaçlarla analiz edilmiştir:

#### 1. Veritabanı Performans Analizi:

- o İşlem yükü (transactions/sec) ve CPU kullanım oranlarının korelasyonu analiz edilerek, sistem üzerindeki yoğunluk dönemleri belirlenmiştir.
- o Page Life Expectancy değişkeni ile bellek kullanım verimliliği değerlendirilmiştir.

#### 2. Anomali Tespiti:

- o Normalden sapmalar gösteren CPU ve bellek kullanımları tespit edilerek, sistem anormalliklerini belirlemek için Makine Öğrenmesi tabanlı anomali tespiti uygulanmıştır.

### ▪ 2.1.4 Veri Seti Örneği

InstanceID	CreateDate	page life expectancy	% privileged time	transactions/sec	write transactions/sec	logical connections	dbaKDOC	VobWadesidolmuslarisil	dbaShrinklog	...	syspolicy_purge_history	dbaCheckintenerStatus	SP_1b11ic6unlukInsertData	tbl11ic7arih1el
0	297 2024-06-20 00:00:00	12167	10.15	254.72	166.82	638	0	0	0	...	0	0	0	0
1	297 2024-06-20 00:00:15	12167	3.32	420.75	336.80	658	0	0	0	...	0	0	0	0
2	297 2024-06-20 00:00:30	12167	2.54	217.77	174.81	655	0	0	0	...	0	0	0	0
3	297 2024-06-20 00:00:45	12167	4.09	159.67	124.74	652	0	0	0	...	0	0	0	0
4	297 2024-06-20 00:01:00	12167	3.12	411.09	190.58	652	0	0	0	...	0	0	0	0

5 rows • 36 columns

Şekil 2.1: Veri Setinin İlk Beş Satırı

## 2.2. Makine Öğrenmesi ve Derin Öğrenme Algoritmaları

Bu çalışmada, veri tabanı performans analizi, anomali tespiti ve tahmine dayalı modelleme süreçlerinde farklı makine öğrenmesi ve derin öğrenme algoritmaları kullanılmıştır. Veri setindeki örüntüleri belirlemek, veri tabanı üzerindeki işlemleri optimize etmek ve anomali tespiti yapmak amacıyla aşağıdaki algoritmalar tercih edilmiştir.

### ▪ 2.2.1 Makine Öğrenmesi Algoritmaları ve Hiperparametreler

Bu çalışmada, verilerin sınıflandırılması ve tahmin edilmesi amacıyla üç farklı makine öğrenmesi algoritması uygulanmıştır. Bu algoritmaların her birine ilişkin kullanılan hiperparametreler aşağıda detaylı olarak sunulmuştur.

#### • 1. Random Forest Algoritması:

Random Forest, birden fazla karar ağacının birleşiminden oluşan bir öğrenme yöntemidir. Bu algoritma, güçlü tahmin performansı ile bilinmekte olup, yüksek doğruluk oranlarına ulaşabilmektedir. Random Forest algoritmasında kullanılan belirli hiper parametreler şunlardır:

- **n\_estimators:** 100  
Modeldeki karar ağaçlarının sayısı belirlenmiştir. 100, makul bir model doğruluğu elde etmek için seçilen optimum değeri temsil etmektedir.
- **max\_depth:** 5  
Karar ağaçlarının maksimum derinliği sınırlandırılmıştır. Bu, modelin aşırı öğrenmesini (overfitting) engellemek için kullanılan bir parametredir.
- **min\_samples\_split:** 2  
Bir iç düğümün iki alt düğüme ayrılması için gereken minimum örnek sayısını ifade etmektedir. Bu parametre, modelin daha ince ayrıntılara odaklanmasına olanak verir.
- **criterion:** Gini  
Karar ağacı inşasında, düğüm bölme kararlarını almak için kullanılan kriterdir. Burada, Gini saflığı kullanılmıştır.
- **Test Size:** 0.2  
Verilerin %20'lik kısmı test seti olarak ayrılmıştır.

- **Random**

**State:**

42

Rastgeleliğin tekrarlanabilirliğini sağlamak amacıyla belirlenen sabit bir sayı.

Sınıflama yönteminin temel amacı; veri tabanı sistemlerinde oluşabilecek performans anomalilerini önceden belirlemek, sistemleri üç sınıfa ayırmak ve erken müdahale mekanizmalarını başlatmaktır.

Veri sınıflandırması, Page Life Expectancy (PLE) değişkenine göre şu şekilde yapılandırılmıştır:

- 0–300: Düşük (Aykırı – Kritik Durum)

- 301–5000: Orta (İzlenmeli – Alarm)

- 5001+: Yüksek (Normal)

Bu eşikler istatistiksel analiz (ortalama  $\approx 6400$ , mod  $\approx 13500$ ) ve alan uzmanlarının görüşlerine dayanmaktadır. Bu yapı, hem model eğitimi hem de sistem uyarı mekanizması için temel teşkil etmektedir.

- **2. SVM (Support Vector Machine) Algoritması:**

Destek Vektör Makinesi (SVM), verilerin en iyi şekilde sınıflandırılabilmesi için bir hiper düzlem (hyperplane) bulmaya çalışan bir algoritmadır. SVM algoritmasında kullanılan hiper parametreler ise şu şekildedir:

- **C:**

2.00

C parametresi, modelin sınıflandırma hatalarını nasıl cezalandıracağını belirler. Daha büyük C değerleri, modelin hata toleransını azaltır.

- **Kernel:**

Sigmoid

SVM'nin doğrusal olmayan sınıflandırmalar için kullanılan çekirdek fonksiyonudur. Burada, sigmoid fonksiyonu tercih edilmiştir.

- **Sütun Aralığı:** 2 (page life expectancy) - 7 (dbaHCDOC)

Kullanılan özellikler arasındaki ilişkiyi tanımlar. Bu aralık, SVM modeline sağlanan veri sütunlarını belirlemektedir.

- **Test**

**Size:**

0.2

Test verisinin oranını belirler. Verilerin %20'lik kısmı test için ayrılmıştır.

- **Random State:** 42  
Modelin çalıştırılabilirliğini ve rastgeleliğinin sabitliğini sağlamak amacıyla belirlenen değer.
- **Operator:** >  
Sınıflandırma için kullanılan karşılaştırma operatörüdür.
- **Hedef Etiket:** 300  
Sınıflandırılacak hedef etiket değeri.
- **Hedef Etiket Tanımı:**  $y = df["page \ life \ expectancy"] > 300$   
Hedef etiket, verinin "page life expectancy" sütununun 300'den büyük olup olmadığına göre belirlenmiştir.
- **3. KNN (K-Nearest Neighbors) Algoritması:**

K-Nearest Neighbors (KNN) algoritması, en yakın komşulara dayalı bir sınıflandırma yöntemidir. KNN, etiketlerin öğrenilmesinde yakınlık ölçütlerini kullanır. Kullanılan hiper parametreler aşağıda verilmiştir:

- **n\_neighbors:** 5  
KNN modelinde, sınıflandırma kararı verirken dikkate alınacak komşu sayısını belirler.
- **metric:** Euclidean  
Komşuluk mesafesinin hesaplanmasında kullanılan metriktir. Burada, Öklid mesafesi tercih edilmiştir.
- **weights:** Uniform  
Komşu ağırlıklarının tüm komşular için eşit olduğu belirtilmiştir. Bu, her komşunun sınıf kararına eşit katkı sağlamasını sağlar.
- **Sütun Aralığı:** 2 (page life expectancy) - 7 (dbaHCDOC)  
KNN modeline sağlanan veri sütunlarının seçildiği aralık.
- **Test Size:** 0.2  
Test verisinin oranını belirler. Verilerin %20'si test için ayrılmıştır.
- **Random State:** 42  
Modelin rastgeleliğini kontrol altına almak için sabit bir değer belirlenmiştir.



- **Operator:** >  
KNN algoritmasında kullanılan karşılaştırma operatörüdür.
- **Hedef Etiket:** 300  
KNN algoritmasında sınıflandırılması gereken hedef etiket değeri.
- **Hedef Etiket Tanımı:**  $y = df["page \ life \ expectancy"] > 300$   
Hedef etiket, verinin "page life expectancy" sütununun 300'den büyük olup olmadığına göre belirlenmiştir.

### 2.2.2 Kullanılan Makine Öğrenmesi Sonuçları

Algorithm	Confusion Matrix	Accuracy	F1-Score	Recall
<b>SVM</b>	[[1864, 137], [104, 11889]]	0.9828	0.9900	0.9913
<b>Random Forest</b>	[[1593, 408, 0], [0, 5482, 0], [0, 0, 6511]]	0.9708	0.9708	0.9708
<b>KNN</b>	[[2004, 0], [25, 1978]]	0.9938	0.9937	0.9875

Tablo 2.1: Üç Algoritmaya Ait Model Eğitim Sonuçları

- **Tablo 2.1'e Ait Açıklamalar:**
- **Confusion Matrix:** Her algoritmanın, doğru ve yanlış sınıflandırdığı örneklerin sayısını göstermektedir. Satırlar gerçek sınıfı, sütunlar ise tahmin edilen sınıfı ifade eder. Örneğin, SVM algoritması için [1864, 137] ilk satırda, [104, 11889] ikinci satırda yer alıyor, bu da doğru ve yanlış sınıflandırmalar hakkında bilgi verir.
- **Accuracy (Doğruluk):** Modelin doğru tahminlerinin, toplam tahminlere oranıdır. Yüksek bir accuracy, modelin genel doğruluğunu gösterir. KNN algoritması, en yüksek doğruluğa sahip olup, %99.38 doğruluk oranına ulaşmıştır.
- **F1-Score:** Precision ve Recall değerlerinin harmonik ortalamasıdır. F1-Score, dengesiz sınıf dağılımlarında daha faydalıdır. SVM ve KNN algoritmaları, yüksek F1-Score değerleri ile dikkat çekmektedir.

- **Recall:** Gerçek pozitif sınıfların doğru bir şekilde sınıflandırılma oranını ifade eder. Yüksek bir recall değeri, modelin doğru pozitif sınıflandırmalarını artırma yeteneğini gösterir. SVM, en yüksek recall değerine sahip algoritmadır.

### ▪ 2.2.3 Kullanılan Derin Öğrenme Algoritmaları

Veri tabanı yönetimi ve performans optimizasyonunda zaman serisi analizi ve tahminleme için derin öğrenme modelleri kullanılmaktadır.

Zaman serisi verilerindeki performans tahmininde klasik yöntemler yetersiz kaldığı için bu projede hem RNN hem de LSTM mimarileri kullanılmıştır.

- RNN, kısa vadeli bağımlılıkları öğrenmekte başarılıdır fakat uzun dönemli ilişkilerde “vanishing gradient” sorunu yaşayabilir.
- LSTM, bellek hücreleri sayesinde geçmişteki önemli bilgileri daha uzun süre koruyarak hem kısa hem uzun vadeli bağımlılıkları öğrenebilir.
- LSTM mimarisiyle eğitilen modelde:
  - 1 LSTM katmanı (4 nöron), dropout, dense çıkış katmanı
  - MinMaxScaler ile normalize edilmiş giriş
  - Eğitim RMSE  $\approx$  796, Test RMSE  $\approx$  720 sonuçları elde edilmiştir.

Bu derin öğrenme modelleri sayesinde, sistem kaynaklarının geçmişe bağlı etkileri öğrenilmiş ve PLE gibi metrikler için doğru tahminler yapılabilmektedir.

- **Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNN):**
  - Önceki zaman adımlarını dikkate alarak ardışık verilerde analiz yapar.
  - Veri tabanındaki işlem yükü değişikliklerini zaman serisi bazında analiz etmek için kullanılmıştır.
  - Ani performans değişikliklerini öngörerek gelecekteki anormallikleri tespit edebilme yeteneğine sahiptir.
- **Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM):**
  - RNN modelinin geliştirilmiş versiyonudur ve uzun vadeli bağımlılıkları öğrenme kapasitesine sahiptir.

- Veri tabanı işlemleri üzerindeki zaman serisi tahminlerinde ve performans optimizasyonunda kullanılmıştır.
- Bellek mekanizması sayesinde geçmiş işlemleri daha etkili bir şekilde hatırlayarak gelecek tahminleri yapabilmektedir.

#### ▪ 2.2.4 Kullanılan Derin Öğrenme Hiper Parametreleri

Hiperparametre	LSTM Modeli	RNN Modeli
Nöron Sayısı	4	4
Dropout Oranı	0.2	0.2
Epoch Sayısı	2	2
Batch Boyutu	32	64
Aktivasyon Fonksiyonu	-	<i>tanh</i>
Optimizör	Adam	Adam

Tablo 2.2: İki Algoritmaya Ait Model Parametreleri

#### ▪ 2.2.5 Kullanılan Derin Öğrenme Sonuçları

Metrik	LSTM (Test)	RNN (Test)	Yorum
<b>MAE</b>	<b>656.69</b>	1149.70	LSTM, ortalama mutlak hata açısından çok daha düşük hata üretmiş ve daha doğru tahminler yapmıştır.
<b>MSE</b>	<b>985.50</b>	1525.89	LSTM, daha küçük kare hata ile tahminlerde daha az büyük hata yapmıştır.
<b>RMSE</b>	<b>31.39</b>	39.06	LSTM, hata dağılımında da daha başarılıdır.
<b>MAPE</b>	<b>26.21%</b>	77.89%	LSTM'nin yüzdelik hatası çok daha düşüktür; RNN'ye göre tahminlerde büyük ölçüde daha isabetlidir.

<b>R<sup>2</sup></b>	<b>0.97</b>	0.93	Her iki model de yüksek doğrulukla tahmin yapmıştır ancak LSTM daha fazla varyans açıklamaktadır.
----------------------	-------------	------	---

Tablo 2.3: Derin Öğrenmede İki Algoritmaya Ait Model Eğitim Sonuçları

- **Kullanılan Hata Metrikleri ve Anlamları**

1. **MAE (Mean Absolute Error)**  
Ortalama mutlak hata, tahmin edilen değerler ile gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasıdır. Daha düşük değerler, daha yüksek doğruluğu gösterir.
2. **MSE (Mean Squared Error)**  
Ortalama kare hata, hata değerlerinin karesinin ortalamasıdır. Büyük hatalara daha fazla ceza verdiği için duyarlıdır. Küçük MSE değeri, modelin genel hatasını daha düşük tuttuğunu gösterir.
3. **RMSE (Root Mean Squared Error)**  
MSE'nin kareköküdür ve birimsel olarak hedef değişkenle aynı ölçeği taşır. Genellikle yorumlaması daha kolaydır. Düşük RMSE, daha iyi performans anlamına gelir.
4. **MAPE (Mean Absolute Percentage Error)**  
Ortalama mutlak yüzdelik hata, tahmin hatalarının yüzde cinsinden ortalamasıdır. Gerçek değer sıfıra yakın olduğunda sorun yaratabilir ama genellikle sezgiseldir: %26 MAPE, tahminlerin ortalama %26 sapma içerdiğini gösterir.
5. **R<sup>2</sup> (R-Kare - Determination Coefficient)**  
Modelin bağımlı değişkenin varyansını ne kadar açıkladığını gösterir. 1'e ne kadar yakınsa, modelin o kadar iyi olduğunu gösterir. 0.93 ile 0.97 arasındaki R<sup>2</sup> değerleri oldukça yüksek kabul edilir.

Yapılan değerlendirmeler göstermektedir ki LSTM modeli, tüm hata metriklerinde RNN modeline kıyasla daha iyi performans sergilemiştir. Özellikle MAPE değerindeki dramatik fark, LSTM'nin gerçek değerlere çok daha yakın tahminlerde bulunduğunu ortaya koymaktadır. R<sup>2</sup> değerleri her iki modelde de yüksek olmakla birlikte, LSTM'nin

açıklayıcılık gücü daha yüksektir.

Bu sonuçlar, zaman serisi verileriyle çalışılan durumlarda LSTM'nin, belleği uzun süreli olarak daha etkili kullanabildiği için daha uygun bir derin öğrenme modeli olduğunu doğrulamaktadır.

### 3. WEB SİTESİ GELİŞTİRME VE KULLANILAN TEKNOLOJİLER

Bu çalışmada, yapay zekâ destekli bir web sitesi geliştirilerek, kullanıcıların veri analizi yapmasına, makine öğrenmesi modelleri eğitmesine ve sonuçları raporlamasına olanak sağlayan bir platform oluşturulmuştur. Web sitesi, React ve Flask gibi modern teknolojileri kullanarak veri tabanı yönetimi, analiz, model eğitimi ve raporlama işlemlerini entegre eden bir sistem olarak tasarlanmıştır.

#### 3.1. Web Sitesinin Genel Yapısı

Web sitesi, kullanıcı dostu bir arayüze sahip olup, veri yükleme, model seçimi, eğitim süreci, sonuçların görüntülenmesi ve raporlanması gibi adımları içermektedir. Platformun temel özellikleri şu şekildedir:

- **Kullanıcı Girişi ve Model Eğitimi:** Kullanıcılar sisteme giriş yaptıktan sonra, makine öğrenmesi veya derin öğrenme modellerini eğitmek için veri yükleyebilir. Model eğitimi tamamlandığında, kullanıcılar görselleştirilmiş sonuçları analiz edebilir ve seçtikleri metrikler doğrultusunda şirket kategorisini belirleyebilir.

Şekil 3.1: Kullanıcı Giriş ve Kayıt Ekranı

Şekil 3.2: Şirket Gurubunun Belirlenmesi

- **Kriter Tespiti ve Firma Kategorilendirme Aracı**

Şekil 3.1.2’de, SQL Server tabanlı veri tabanı sistemlerinin performans özelliklerine göre firma büyüklüğünü kategorize etmek amacıyla geliştirilmiştir. Hazırlanan bu sistemde, kullanıcıdan aşağıdaki işlemleri gerçekleştirmesi beklenmektedir:

1. **Adım 1 – Kriter Betiğini Çalıştırma:**  
Kullanıcı, tarafımızca sağlanan kriter.sql adlı SQL betiğini kendi veri tabanı ortamında çalıştırır. Bu betik, sistemin çeşitli performans göstergelerini (işlem sayısı, bağlantı yoğunluğu, RAM, CPU kullanımı vb.) analiz eder ve sonuçları tek satırlık bir tablo formatında üretir.
2. **Adım 2 – Sonuçların Girdisi:**  
Betik çalıştırdıktan sonra kullanıcı, elde ettiği değerleri bu ekranda ilgili metrik alanlarında işaretler. Her metrik için "Büyük", "Orta" veya "Küçük" kategorilerinden biri seçilir.
3. **Adım 3 – Firma Sınıflandırması:**  
Yapılan işaretleme sonucunda sistem, firmanın hangi ölçekte faaliyet gösterdiğini (örneğin büyük, orta ya da küçük veri işleme kapasitesine sahip bir sistem) belirler. Bu sınıflandırma; kaynak yönetimi, kapasite planlaması ve sistem iyileştirme stratejileri için temel girdi olarak kullanılabilir.

- **Kriter.sql Betiği İşleyiş Özeti**

Kriter. Sql betiği, SQL Server tabanlı sistemlerin performansını değerlendirmek amacıyla geliştirilmiştir. Kodun başında, önceki çalışmalardan kalan geçici tablo (#TotalRowsTemp) varsa silinir ve toplam satır sayısını tutacak şekilde yeniden oluşturulur. Bu adım, veri temizliği ve tutarlılığı sağlamak içindir.

Ardından, betik sırasıyla aşağıdaki 11 metriği sistemden dinamik SQL ve sistem görünümleri aracılığıyla toplar:

- **@MaxTransactions:** DOCperformanceMonitor tablosundan maksimum işlem sayısı alınır.
- **@MaxProcessorTime:** En yüksek işlemci kullanımı yüzdesi hesaplanır.
- **@TotalSessions:** Anlık oturum sayısı sys.dm\_exec\_connections ve sys.dm\_exec\_sessions kullanılarak belirlenir.
- **@SqlServerEdition:** SERVERPROPERTY fonksiyonu ile SQL Server sürümü tespit edilir.
- **@TotalRAM:** sys.dm\_os\_sys\_info görünümünden fiziksel RAM bilgisi çekilir.
- **@TotalRows:** Tüm kullanıcı veri tabanları dinamik SQL ile dolaşarak toplam satır sayısı hesaplanır.
- **@AvgLogBackupSize:** backupset ve backupmediafamily tabloları üzerinden son 50 log yedeğinin ortalama boyutu alınır.
- **@AvgTransactionsPerSec** ve **@AvgTransactions:** DOCperformanceMonitor üzerinden işlem yoğunlukları ölçülür.
- **@MaxConnectionCount** ve **@AvgConnectionCount:** Maksimum ve ortalama bağlantı sayıları hesaplanır.

Elde edilen tüm sonuçlar, @Results adlı tablo değişkenine tek bir satır olarak eklenmekte ve SELECT \* FROM @Results komutu ile kullanıcıya sunulmaktadır. Son adımda geçici tablo silinerek sistem temizliği sağlanır.

Bu yapı, sistem kaynaklarının durumu hakkında bütüncül ve güncel bir görünüm sunarak, veri tabanı altyapısının performansını nesnel biçimde analiz etmeyi mümkün kılar.

- **Dosya Yükleme ve İşleme:** Kullanıcılar veri setlerini yükleyerek model eğitimine başlayabilir ve arka planda çalışan algoritmalar sayesinde veri ön işleme adımlarını gerçekleştirebilir.

Şekil 3.3: Dosya Yükleme ve Algoritma Seçme Alanı

Şekil 3.4: Seçilen Algoritmaya Göre Hiper parametre Girişi

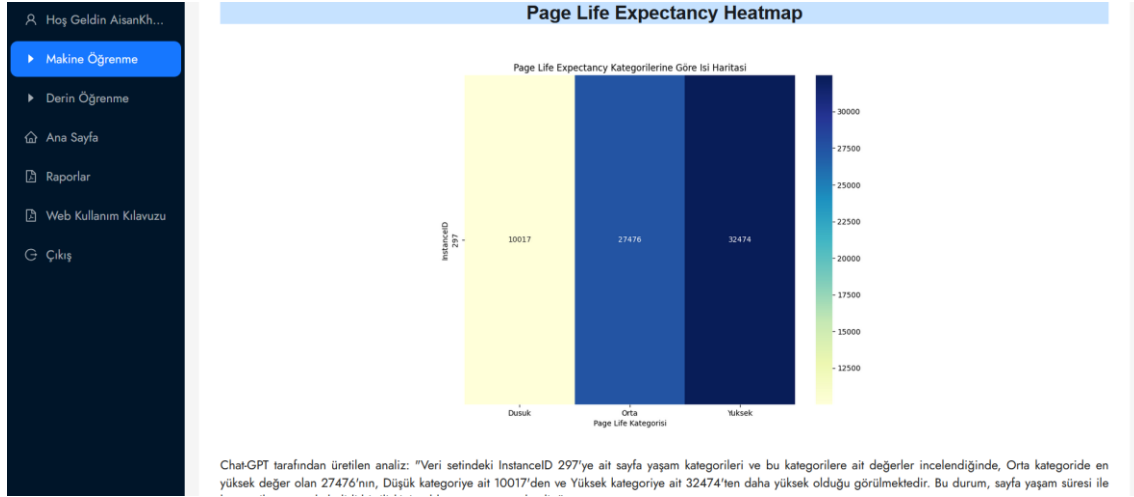


Şekil 3.5: Eğitim Sürecinin İlerlemesi Adımı

- **Gerçek Zamanlı Sonuç Analizi:** Yapay zeka destekli anlık yorumlama özelliği sayesinde, model eğitimi sonucunda oluşan grafikler ChatGPT tarafından analiz edilerek kullanıcıya detaylı açıklamalar sunulmaktadır.

InstanceID	CreateDate	page life expectancy	% privileged time	transactions/sec	write transactions/sec	logical connections	dbaHCDOC
297	2024-06-20 00:00:00	12167	10.15	254.72	166.82	638	0
297	2024-06-20 00:00:15	12167	3.32	420.75	336.8	658	0
297	2024-06-20 00:00:30	12167	2.54	217.77	174.81	655	0
297	2024-06-20 00:00:45	12167	4.09	159.67	124.74	652	0
297	2024-06-20 00:01:00	12167	3.12	411.09	190.58	652	0
297	2024-06-20 00:01:15	12167	4.09	267.42	166.64	654	0
297	2024-06-20 00:01:30	12167	3.12	250.91	187.93	657	0
297	2024-06-20 00:01:45	12167	13.46	481.46	253.72	657	0
297	2024-06-20 00:02:00	12167	12.09	607.91	281.5	656	0
297	2024-06-20 00:02:15	12167	9.17	484.47	265.71	653	0

Şekil 3.6: Veri Seti İçeriği Tablo Halinde



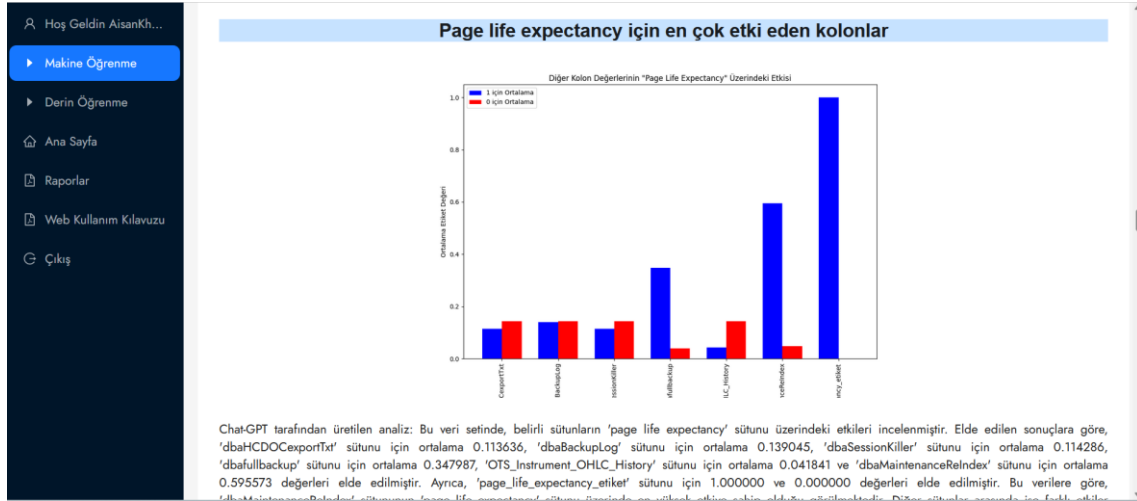
Şekil 3.7: Grafiğin ChatGpt Tarafından Yorumlanması

İstatistiksel Özetler						
Özellik	Mod	Medyan	Standart Sapma	Ortalama	Minimum	Maksimum
page life expectancy	13547.00	4375.00	6041.443132	6423.278374	0.00	24110.00
% privileged time	3.51	5.26	3.897389	6.232782	0.39	46.98
transactions/sec	101.87	421.43	1193.552874	799.743985	4.99	12576.29
write transactions/sec	58.92	194.84	212.895878	264.981632	0.00	2348.42
logical connections	589.00	626.00	114.495493	655.545714	341.00	978.00
dbaHCDOC	0.00	0.00	0.091134	0.008375	0.00	1.00

Şekil 3.8: Veri Setinin İstatistiksel Özellikleri

İlişkiler		
Sütun	1 için Ortalama	0 için Ortalama
dbaHCDOCexportTxt	0.113636	0.143205
dbaBackupLog	0.139045	0.143826
dbaSessionKiller	0.114286	0.143182
dbafullbackup	0.347987	0.039749
OTS_Instrument_OHLC_History	0.041841	0.143515
dbaMaintenanceRelIndex	0.595573	0.048068
page_life_expectancy_etiket	1.000000	0.000000

Şekil 3.9: Veri Setinin İlişkiler Tablosu

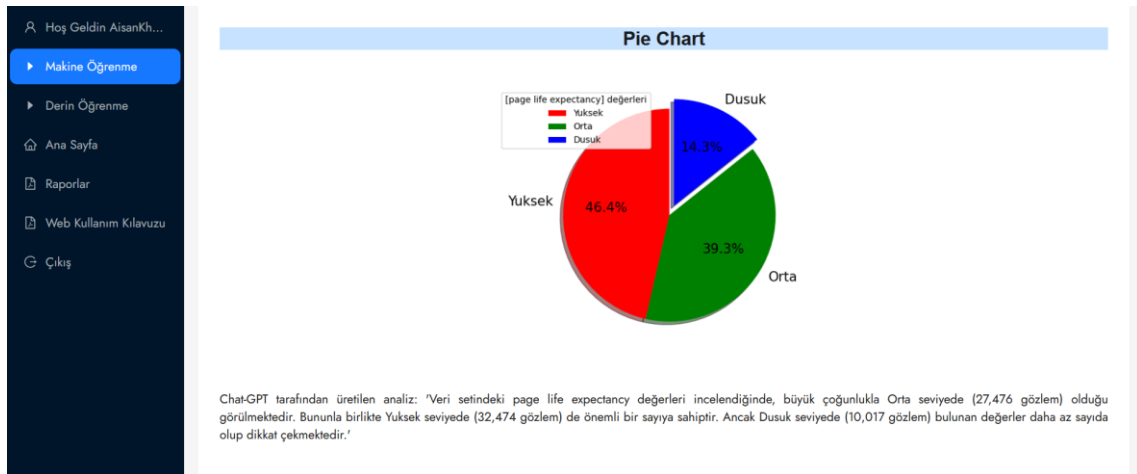


Şekil 3.10: PLE Değeri için Etkili Kolonlar ve ChatGpt Yorumu

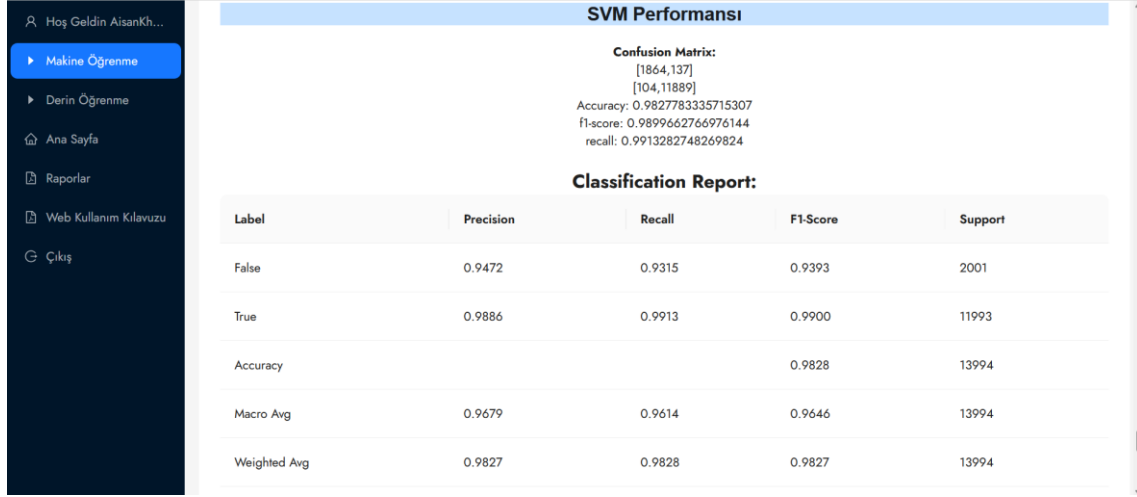
Veri Çerçevesinin Sütun Bazında İstatistiksel Özeti (Transpoze Edilmiş)

row	Count	Mean	Std	Min	25%	50%	75%	Max
InstanceID	69967.0	297.000000	0.000000	297.00	297.00	297.00	297.00	297.00
CreateDate	69967.0	6423.278374	6041.443132	0.00	1324.00	4375.00	10447.00	24110.00
page life expectancy	69967.0	6.232782	3.897389	0.39	3.51	5.26	7.79	46.98
% privileged time	69967.0	799.743985	1193.552874	4.99	244.76	421.43	727.68	12576.29
transactions/sec	69967.0	264.981632	212.895878	0.00	123.80	194.84	343.82	2348.42
write transactions/sec	69967.0	655.545714	114.495493	341.00	563.00	626.00	775.00	978.00
logical connections	69967.0	0.008375	0.091134	0.00	0.00	0.00	0.00	1.00
dbaHcDoc	69967.0	0.000029	0.005346	0.00	0.00	0.00	0.00	1.00
VbVadesiDolmuslariSil	69967.0	0.000029	0.005346	0.00	0.00	0.00	0.00	1.00
dbaShrinkLog	69967.0	0.001258	0.035443	0.00	0.00	0.00	0.00	1.00
dbaHcDocExportTxt	69967.0	0.000000	0.000000	0.00	0.00	0.00	0.00	0.00
dbaCheckPrimaryNode	69967.0	0.000000	0.000000	0.00	0.00	0.00	0.00	0.00

Şekil 3.11: Veri Setinin Sütun Bazında İstatistiksel Özeti

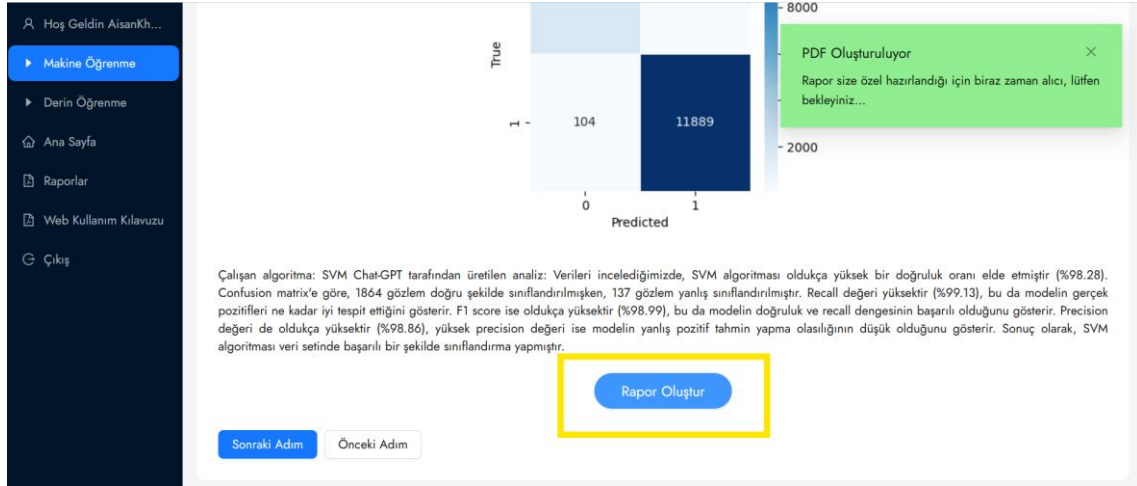


Şekil 3.12: Veri Setinin PLE değerlerine Ait Pasta Grafiği



Şekil 3.13: Modelin Performans Ölçümü ve Karmaşıklık Matrisi

- **Raporlama ve PDF Oluşturma:** Kullanıcılar, eğitim sonuçlarını PDF formatında rapor olarak kaydedebilir ve her oluşturulan rapor, kullanıcının hesabına otomatik olarak kaydedilmektedir.



Şekil 3.14: Tüm Sonuçların Raporlandığı Pdf Butonu

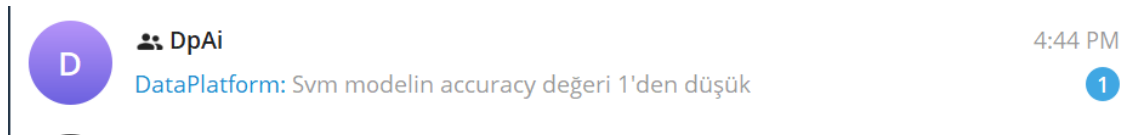


Şekil 3.15: Pdf Raporuna Ait Görsel

AI Model Adı	Rapor Oluşturma Tarihi	Rapor
svm model	2025-05-22 16:49:04	<a href="#">İndir</a>
svm model	2024-10-14 12:01:29	<a href="#">İndir</a>
svm model	2024-10-11 14:22:03	<a href="#">İndir</a>
svm model	2024-10-11 10:57:05	<a href="#">İndir</a>
svm model	2024-10-10 17:00:46	<a href="#">İndir</a>
svm model	2024-10-03 10:26:17	<a href="#">İndir</a>
lstm model	2024-10-02 17:58:55	<a href="#">İndir</a>
random forest model	2024-10-02 17:17:21	<a href="#">İndir</a>

Şekil 3.16: Kullanıcının Oluşturduğu Tüm Raporların Tutulduğu Ekran

- **Telegram Bot Bildirim Sistemi:** Model eğitimi sırasında, Telegram botu üzerinden anlık doğruluk (accuracy) değeri bildirimleri gönderilmekte, böylece kullanıcı modelin ilerleme durumunu gerçek zamanlı olarak takip edebilmektedir.

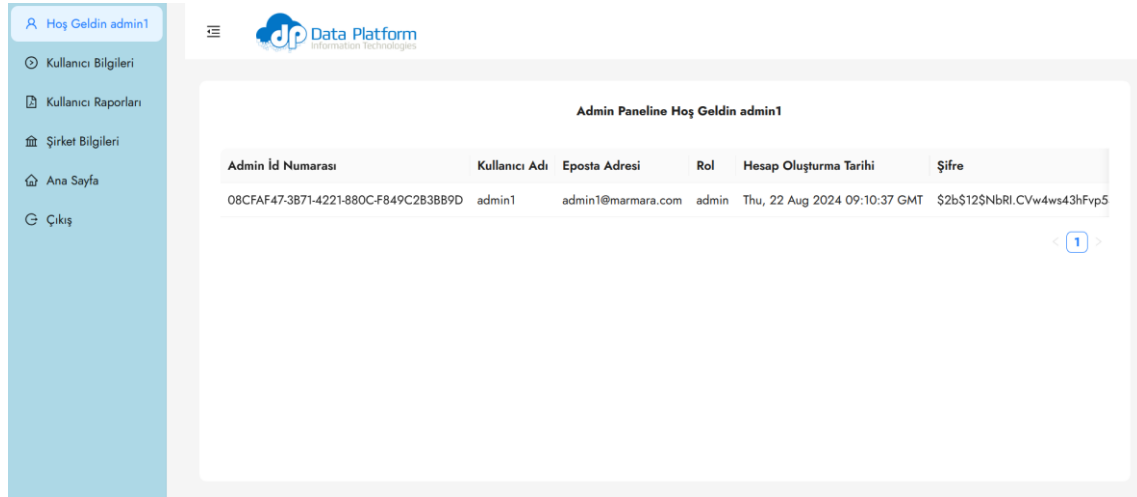


Şekil 3.17: Telegram Bot ve Anlık Bildirim Sistemi

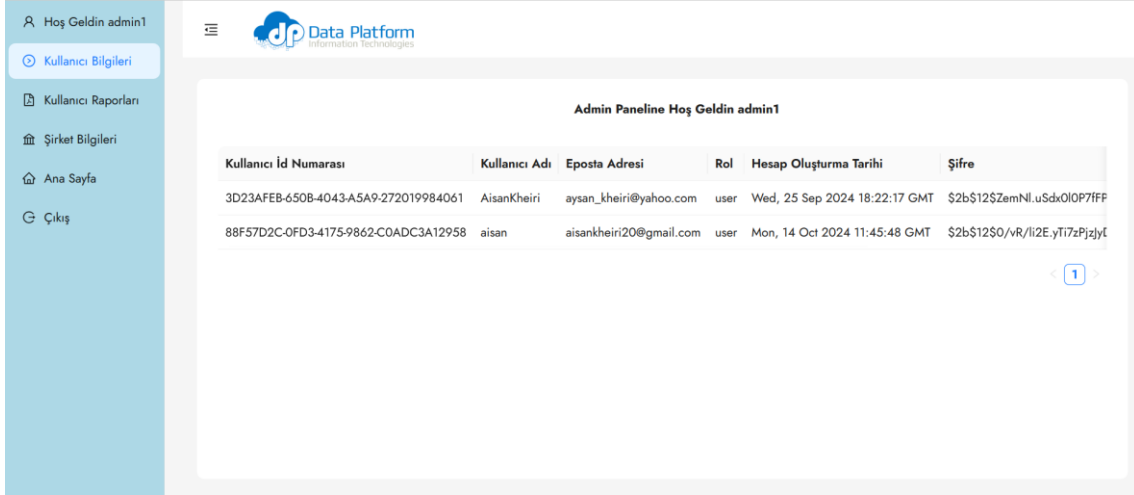


Şekil 3.18: Yeni Kullanıcı İçin Oluşturulan Kullanım Kılavuzu Örneği

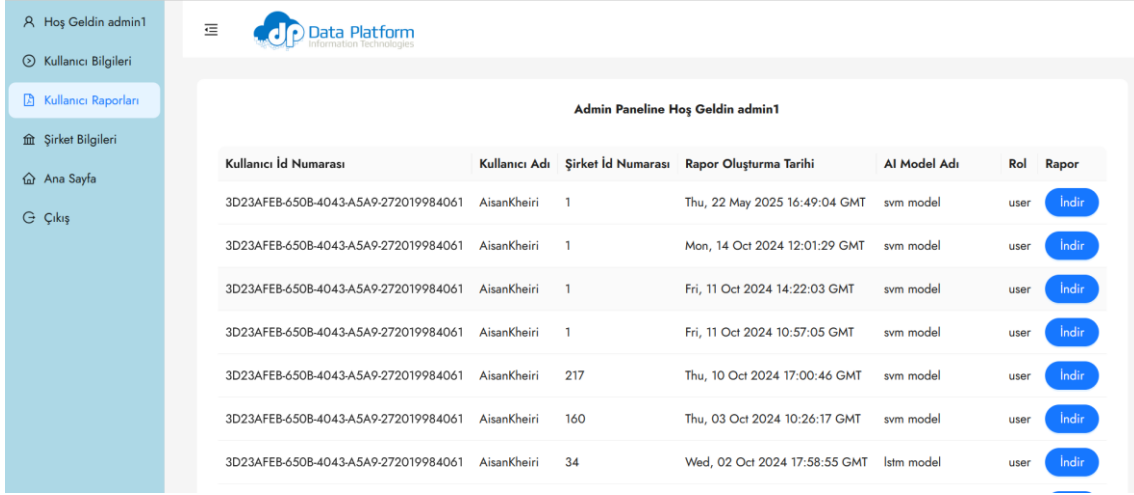
- **Admin Paneli ve Yönetim:** Web sitesi üzerinde bir admin paneli oluşturulmuş olup, admin kullanıcıları diğer kullanıcıların gerçekleştirdiği model eğitimlerini, raporlamalarını ve genel sistem istatistiklerini takip edebilmektedir.



Şekil 3.19: Adminlerin Gözüktüğü Sekme



Şekil 3.20: Kullanıcıların Gözüküğü Sekme



Şekil 3.21: Kullanıcıların Oluşturduğu Raporlar Sekmesi

### 3.2. Kullanılan Teknolojiler ve Araçlar

Web sitesinin geliştirilme sürecinde, modern frontend ve backend teknolojileri kullanılarak kullanıcı dostu, ölçeklenebilir ve yüksek performanslı bir sistem oluşturulmuştur. Kullanılan temel araçlar şunlardır:

- **Backend (Sunucu Tarafı)**
- **Python & Flask:**
  - Web API'lerinin geliştirilmesi, veri tabanı yönetimi ve model eğitimi süreçleri için kullanılan hafif ve ölçeklenebilir bir Python web çerçevesidir.
  - Kullanıcı doğrulama, veri yükleme, model eğitimi ve sonuç

raporlamalarını yönetmek için Flask ile RESTful API'ler oluşturulmuştur.

- **MSSQL Server:**
  - Veri yönetimi ve saklama süreçleri için kullanılan ilişkisel veri tabanı yönetim sistemi (RDBMS).
  - Kullanıcı işlemleri, model eğitim sonuçları ve analiz verileri MSSQL üzerinde saklanmaktadır.
- **Frontend (Kullanıcı Arayüzü - UI/UX)**
- **React.js:**
  - Kullanıcıların etkileşimli ve dinamik bir deneyim yaşamasını sağlamak için bileşen tabanlı modern JavaScript kütüphanesi kullanılmıştır.
  - Veri yükleme, model seçimi, eğitim süreci ve sonuçların görselleştirilmesi React bileşenleri aracılığıyla yönetilmektedir.
- **Ant Design (AntD):**
  - Web arayüzünün modern ve profesyonel bir görünüme sahip olması için kullanılan React tabanlı UI bileşen kütüphanesidir.
  - Kullanıcı girişi, veri yükleme, grafik gösterimi, butonlar ve form elemanları gibi bileşenler için Ant Design bileşenleri kullanılmıştır.
- **Bildirim ve Otomasyon Sistemleri**
- **Telegram Bot API:**
  - Model eğitimi sırasında doğruluk (accuracy) değeri ve diğer önemli istatistikler Telegram botu aracılığıyla kullanıcıya gerçek zamanlı olarak bildirilecektir.
  - Kullanıcılar, model eğitimi tamamlandığında Telegram üzerinden otomatik güncellemeler alabileceklerdir.
- **Raporlama ve Dosya Yönetimi**
- **FPDF (Python PDF Kütüphanesi):**
  - Kullanıcıların eğitim sonuçlarını ve analizlerini PDF formatında raporlamasına olanak sağlayan kütüphane kullanılacaktır.



- Kullanıcı, sonuçları analiz ettikten sonra "Rapor Oluştur" butonuna tıklayarak PDF formatında çıktı alabilecektir.
- FPDF ile oluşturulan PDF dosyaları, kullanıcının hesabına kaydedilecek ve istenildiğinde indirilebilecektir.

### 3.3. Veri Tabanı ve Tablolar

Tablo Adı	Açıklama
UserTable	Sistemdeki kullanıcı bilgilerini (UserID, UserName, Email, Password, Role, CreateDate) saklar. Bu tabloya kullanıcı ekleme, silme, güncelleme ve doğrulama işlemleri yapılır.
CompanyTable	Şirket bilgilerini (CompanyID, CompanyName) içerir. Şirket ekleme, silme ve güncelleme işlemleri gerçekleştirilir.
UserLogTable	Kullanıcı işlem geçmişi ve raporlarını (AI modeli, kullanıcı rolü, tarih, şirket bilgisi ve rapor dosyası) saklar. Kullanıcının yaptığı işlemler burada izlenebilir.

- **Stored Procedure'lerin Rolü**

#### Kullanıcı Yönetimi (UserTable)

sp\_AddUser: Yeni kullanıcı ekler

sp\_UpdateUser: Kullanıcı bilgilerini günceller

sp\_DeleteUser: Kullanıcıyı siler

sp\_CheckUserLogin: Kullanıcı adı ve şifre kontrolü yapar

sp\_CheckUserExists: Kullanıcının var olup olmadığını kontrol eder

#### Şirket Yönetimi (CompanyTable)

sp\_InsertCompany: Yeni şirket ekler

sp\_UpdateCompany: Şirket adını günceller

sp\_DeleteCompany: Şirketi siler

### **Kullanıcı Log Kayıtları (UserLogTable)**

sp\_InsertUserLog: Kullanıcının yaptığı işlemlerin kaydını ekler

sp\_UpdateUserLog: Kullanıcıya ait işlem geçmişini günceller

sp\_DeleteUserLog: Kullanıcıya ait işlem geçmişini siler

### **3.4. Web Sitesine Ait Ekran Kaydı**

Geliştirilen web sitesinin işlevselliğini ve kullanıcı arayüzünü daha etkili bir şekilde sunmak amacıyla bir ekran kaydı alınmıştır. Bu kayıt, sistemin çalışma prensibini ve uygulamanın kullanıcı ile etkileşimini görsel olarak desteklemektedir. İlgili ekran kaydına aşağıdaki bağlantıdan erişilebilir:

#### **Video Bağlantısı:**

<https://drive.google.com/file/d/1MP7VI-LmR7-OjoF7gMRPKUPKU74YJdVE/view?usp=sharing>

## **4. BULGULAR VE TARTIŞMA**

Yapılan modelleme sonucunda, sistemdeki aykırılıklar Page Life Expectancy gibi metriklerle başarılı şekilde sınıflandırılmış ve kritik eşiklerde ( $PLE < 300$ ) otomatik uyarı sistemleri oluşturulmuştur.

LSTM ve RNN gibi zaman serisi odaklı modellerin performansı karşılaştırıldığında LSTM'nin genel hata (MSE, RMSE, MAE) oranlarında daha üstün olduğu görülmüştür. Bu, geçmişe dayalı performans tahminlerinde LSTM'nin daha isabetli olduğunu göstermektedir.

Ayrıca FP-Growth ile iş parçacığı etkileri modellenmiş, örneğin 'dbaMaintenanceReIndex' job'ı çalıştığında logical connections ile PLE arasındaki bağ net biçimde analiz edilmiştir.

Bu bulgular, projenin amacına uygun olarak karar destek sistemine katkı sağlamış ve otonom uyarı mimarisinin temelini oluşturmuştur.

### **4.1. Gerçekleştirilen Çalışmalar**

#### **• 4.1.1. Veri Toplama ve Ön İşleme**

Proje kapsamında, Data Platform şirketine ait gerçek veriler kullanılmıştır. Bu veriler, veri tabanı yönetimi, sistem performansı takibi ve anomali tespiti amaçlarına hizmet edecek

şekilde ön işleme tabi tutulmuştur. Eksik veriler doldurulmuş, aykırı değerler temizlenmiş ve veri seti analiz süreçlerine uygun hale getirilerek optimize edilmiştir.

- **4.1.2. Web Sitesinin Geliştirilmesi**

Kullanıcı deneyimini ön planda tutan, etkileşimli bir web arayüzü React.js ve Ant Design kütüphanesiyle oluşturulmuştur. Backend tarafı ise Flask ve Python ile geliştirilmiş; bu yapı MSSQL veri tabanı ile tam uyumlu şekilde entegre edilmiştir. Web sitesi üzerinden veri analizleri, model çıktıları ve raporlamalar dinamik olarak sunulmaktadır.

- **4.1.3. Makine Öğrenmesi Modelleri ile Anomali Tespiti**

Anomali tespiti ve sistem performans değerlendirmeleri amacıyla çeşitli klasik makine öğrenmesi algoritmaları (Random Forest, KNN, SVM vb.) uygulanmıştır. Bu modellerin çıktıları, sistem davranışlarını analiz etmek ve olağan dışı durumları tespit etmek için başarıyla kullanılmıştır.

- **4.1.4. Model Performans Analizi**

Model eğitim süreçlerinde; doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoru gibi performans metrikleri kullanılarak değerlendirmeler yapılmıştır. Uygulanan modellerin karşılaştırmalı analizleri sonucunda, her bir modelin hangi senaryolarda daha verimli çalıştığı belirlenmiştir.

- **4.1.5. Gelişmiş Özelliklerin Entegrasyonu**

Projenin gelişmiş sürümünde aşağıdaki yenilikçi özellikler başarıyla uygulanmıştır:

- **Otomatik Model Seçimi (Auto ML):** Sistem, en uygun modelin parametrelerini otomatik olarak belirleyerek kullanıcı müdahalesine gerek kalmadan optimum sonuçlar sunmaktadır.
- **Telegram Bot API Entegrasyonu:** Model eğitim süreçlerinin çıktıları, Telegram Bot API aracılığıyla anlık olarak kullanıcılara iletilmektedir. Bu entegrasyon sayesinde doğruluk oranları ve diğer kritik metrikler gerçek zamanlı olarak takip edilebilmektedir.
- **Otomatik PDF Raporlama (FPDF):** Eğitim sonuçları ve analiz çıktıları, FPDF kütüphanesi kullanılarak otomatik şekilde PDF formatında raporlanmakta ve

kullanıcıya sunulmaktadır. Bu özellik, raporlama sürecini sistematik ve kullanıcı dostu bir hale getirmiştir.

## 5. SONUÇ

Proje çıktıları akademik açıdan zaman serisi temelli anomali tespiti için veritabanı performans yönetiminde başarılı biçimde uygulanabileceğini göstermiştir. Özellikle gerçek verilerle yapılan testlerde, derin öğrenme modellerinin sistem davranışını doğru tahmin edebildiği gözlemlenmiştir.

Endüstriyel olarak bu sistem:

- DBA iş yükünü azaltmakta
- SLA yönetimini güçlendirmekte
- Otonom sistem optimizasyonuna zemin hazırlamaktadır.

Yerli yazılım yaklaşımıyla geliştirilen bu sistem, Türkiye'deki veri merkezleri, bankacılık sistemleri ve ERP altyapılarında uygulanabilir, ölçeklenebilir ve sürdürülebilir bir yapıda tasarlanmıştır.

Bu çalışmada gerçekleştirilen tüm uygulamalar, veri tabanı yönetimi süreçlerinin yapay zekâ teknikleriyle nasıl etkin bir şekilde optimize edilebileceğini ortaya koymuştur. Geliştirilen web tabanlı sistem, kullanıcıların veri ön işleme, model eğitimi, anomali tespiti ve performans analizi gibi işlemleri otomatik ve etkileşimli bir şekilde gerçekleştirebilmesini mümkün kılmaktadır.

Özellikle Auto ML entegrasyonu, Telegram ile anlık bildirim sistemi ve otomatik PDF raporlama modülü sayesinde sistem, kullanıcı dostu ve gerçek zamanlı karar destek sunabilen bir yapıya kavuşmuştur.

Gelecek çalışmalarda, sistemin daha büyük ve çeşitli veri kümeleri üzerinde test edilmesiyle model doğruluğunun artırılması; aynı zamanda altyapının ölçeklenebilirliğinin ve kullanıcı arayüzünün daha da geliştirilmesi hedeflenmektedir. Bu doğrultuda, tam otomatik çalışan, esnek, güvenilir ve yüksek doğruluklu bir analiz platformu oluşturulması amaçlanmaktadır.

## KAYNAKLAR

- [1] X. Zhou et al., “D-Bot: Database Diagnosis System using Large Language Models,” arXiv.org, 2024. <https://arxiv.org/abs/2312.01454>
- [2] Ismet Karakurt, S. Ozer, Taner Ulusinan, and Murat Can Ganiz, “A machine learning approach to database failure prediction,” 2017 International Conference on Computer Science and Engineering (UBMK), pp. 1030–1035, Oct. 2017, doi: <https://doi.org/10.1109/ubmk.2017.8093426>.
- [3] X. Zhou, Z. Sun, and G. Li, “DB-GPT: Large Language Model Meets Database,” Data science and engineering (Internet), Jan. 2024, doi: <https://doi.org/10.1007/s41019-023-00235-6>.
- [4] S. Islam, “FUTURE TRENDS IN SQL DATABASES AND BIG DATA ANALYTICS: IMPACT OF MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE,” vol. 1, no. 4, pp. 47–62, Aug. 2024, doi: <https://doi.org/10.62304/ijse.v1i04.188>.
- [5] G. Li, X. Zhou, and X. Zhao, “LLM for Data Management,” Proceedings of the VLDB Endowment, vol. 17, no. 12, pp. 4213–4216, Aug. 2024, doi: <https://doi.org/10.14778/3685800.3685838>.
- [6] “View of Machine Learning for Database Management and Query Optimization,” Elementaria.my.id, 2024. <https://elementaria.my.id/index.php/e/article/view/66/23>