

**BAŞKENT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
SAVUNMA TEKNOLOJİLERİ VE SİSTEMLERİ ANABİLİM DALI  
SAVUNMA PLATFORMLARI TEZLİ YÜKSEK LİSANS PROGRAMI**

**YAZILIM PROJELERİNDE EFOR ANALİZİ VE TAHMİNİ**

**HAZIRLAYAN**

**CEREN ÖZGE ATMACA**

**YÜKSEK LİSANS TEZİ**

**ANKARA - 2024**



**BAŞKENT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
SAVUNMA TEKNOLOJİLERİ VE SİSTEMLERİ ANABİLİM DALI  
SAVUNMA PLATFORMLARI TEZLİ YÜKSEK LİSANS PROGRAMI**

**YAZILIM PROJELERİNDE EFOR ANALİZİ VE TAHMİNİ**

**HAZIRLAYAN**

**CEREN ÖZGE ATMACA**

**YÜKSEK LİSANS TEZİ**

**TEZ DANIŞMANI**

**PROF. DR. FARUK ELALDI**

**TEZ EŞ DANIŞMANI**

**DR. ÖZGÜN ÇÖLLÜOĞLU GÜLEN**

**ANKARA - 2024**

**BAŞKENT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

Savunma Teknolojileri ve Sistemleri Anabilim Dalı Savunma Platformları Tezli Yüksek Lisans çerçevesinde Ceren Özge ATMACA. tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Yüksek Lisans olarak kabul edilmiştir.

Tez Savunma Tarihi: 27/05/ 2024

Tez Adı: Yazılım Projelerinde Efor Analizi ve Tahmini

**Tez Jüri Üyeleri ( Unvanı, Adı - Soyadı, Kurumu )**

**İmza**

Prof. Dr. Faruk ELALDI, Başkent Üniversitesi

.....

Dr.Öğr. Üyesi Ahmet Anıl MÜNGEN, OSTİM Teknik Üniversitesi

.....

Dr. Öğr. Üyesi Didem ÖLÇER, Başkent Üniversitesi

.....

**ONAY**

Prof. Dr. Faruk ELALDI

Enstitüsü Müdürü

Tarih: ... / ... / .....

**BAŞKENT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**YÜKSEK LİSANS / DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU**

Tarih: 21/05/2024

Öğrencinin Adı, Soyadı: Ceren Özge ATMACA

Öğrencinin Numarası: 22020201

Anabilim Dalı: Savunma Teknolojileri Ve Sistemleri

Programı: Savunma Platformları Tezli Yüksek Lisans

Danışmanın Unvanı/Adı, Soyadı: Prof. Dr. Ömer Faruk ELALDI, Dr. Özgün ÇÖLLÜOĞLU  
GÜLEN

Tez Başlığı: Yazılım Projelerinde Efor Analizi ve Tahmini

Yukarıda başlığı belirtilen Yüksek Lisans/Doktora tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 80 sayfalık kısmına ilişkin, 21 / 05 / 2024 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 7'dir. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimedenden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:.....

**ONAY**

Tarih: .... / .... / .....

Öğrenci Danışmanı

Prof. Dr. Ömer Faruk ELALDI

## TEŐEKKÜR

Tez alıőmam sũresince bana yol gũsterici ve destek olan deęerli danıőman hocalarım Prof. Dr. Őmer Faruk ELALDI ve Dr. Őzgũn ŐLLŐOęLU GŐLEN'e en iten teőekkũrlerimi sunarım. Sizlerin bilgi, birikim ve tecrũbeleriniz sayesinde, tezimde ilerlemek ve geliőmek iin gerekli rehberlięi buldum. Sizlerin destekleri ve motivasyonu, tez alıőmamın her aőamasında beni cesaretlendirdi ve ilerlememi saęladı.

Ayrıca, yakın arkadaőlarıma ve sevgili aileme de teőekkũr etmek isterim. Sizlerin anlayıőınız, sabrınız ve desteęiniz olmadan bu yolculuk ok daha zor olurdu. Her zaman yanımda olduęunuz iin minnettarım.



# ÖZET

**Ceren Özge ATMACA**

**YAZILIM PROJELERİNDE EFOR ANALİZİ VE TAHMİNİ**

**Başkent Üniversitesi Fen Bilimleri Enstitüsü**

**Savunma Teknolojileri ve Sistemleri Anabilim Dalı**

**2024**

Bu çalışmada, yazılım proje yönetimindeki efor tahmini konusunda sınıflandırma tabanlı bir yaklaşımın nasıl uygulandığı incelenmiştir. Yazılım projelerinde çoğunlukla manuel ve uzman yargısına dayalı olarak yapılan efor tahmininin daha hızlı, tutarlı ve tekrar edilebilir bir şekilde gerçekleştirilmesini amaçlayan bu çalışma, yazılım firmaları için zaman ve maliyet açısından tasarruf edilebilecek, aynı zamanda tutarlı ve güvenilir tahminler elde edilebilecek bir sistem önerisinde bulunmuştur.

Araştırma kapsamında, yazılım efor tahmini için lojistik regresyon algoritması ile iki farklı sınıflandırma modeli geliştirilmiştir. Oluşturulan modele ilişkin uyum iyiliği istatistikleri ve modelin sınıflandırma başarısı hesaplanarak araştırma sonuçları analiz edilmiş, açıklama oranı ve başarı oranı yüksek sonuçlar elde edildiği görülmüştür. Bununla birlikte oluşturulan iki model ile veri ön işleme çalışmalarının sınıflandırma modeline etkileri incelenmiş, aynı veri seti ile literatürde yapılan çalışmalarla karşılaştırma sağlanmıştır.

**ANAHTAR KELİMELEER:** Yazılım efor tahminleme, sınıflama algoritması, lojistik regresyon.

# ABSTRACT

**Ceren Özge ATMACA**

**EFFORT ANALYSIS AND ESTIMATION IN SOFTWARE PROJECTS**

**Başkent University Institute of Science**

**Department of Defense Technologies and Systems**

**2024**

This study examines how a classification-based approach is applied to effort estimation in software project management. Aimed at achieving faster, more consistent, and repeatable effort estimation in software projects, which are mostly reliant on manual and expert judgment, this study proposes a system where software companies can save time and costs while obtaining consistent and reliable estimates.

Within the scope of the research, two different classification models were developed for software effort estimation using the logistic regression algorithm. The goodness-of-fit statistics of the constructed model and the classification success of the model were analyzed, revealing high explanatory and accuracy rates. Additionally, the effects of data preprocessing on the classification model were investigated, and comparisons were made with studies in the literature using the same dataset.

**KEYWORDS:** Software effort estimation, classification algorithm, logistic regression

# İÇİNDEKİLER

TEŞEKKÜR .....	vi
ÖZET .....	ii
ABSTRACT .....	iii
İÇİNDEKİLER .....	iv
TABLolar LİSTESİ .....	vi
ŞEKİLLER LİSTESİ .....	x
SİMGELER VE KISALTMALAR LİSTESİ .....	xi
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>1.1. Amaç .....</b>	<b>2</b>
<b>1.2. Literatür Taraması .....</b>	<b>2</b>
<b>1.2.1. Lojistik regresyon ile ilgili çalışmalar .....</b>	<b>2</b>
<b>1.2.2. Yazılım efor tahmini ile ilgili çalışmalar .....</b>	<b>4</b>
<b>1.2.3. SEERA veri seti ile ilgili çalışmalar .....</b>	<b>8</b>
<b>2. GENEL KISIMLAR .....</b>	<b>11</b>
<b>2.1. Yazılım Proje Yönetimi .....</b>	<b>11</b>
<b>2.2. Yazılım Projelerinde Efor Tahmininin Önemi .....</b>	<b>11</b>
<b>2.3. Yazılım Projelerinde Efor Tahmin Yöntemleri .....</b>	<b>12</b>
<b>3. GEREÇ VE YÖNTEM .....</b>	<b>13</b>
<b>3.1. Ön Bilgiler .....</b>	<b>13</b>
<b>3.1.1. Bağımlı Değişken .....</b>	<b>13</b>
<b>3.1.2. Bağımsız Değişken .....</b>	<b>13</b>
<b>3.2. Çalışmada Kullanılan Yöntemler .....</b>	<b>13</b>
<b>3.2.1. Lojistik Regresyon .....</b>	<b>14</b>
<b>3.2.2. Uyum İyiliği İstatistikleri .....</b>	<b>15</b>

3.2.3. Bağımsız Örneklem T-test .....	15
3.2.4. Tek Yönlü Anova Testi.....	16
3.2.5. Pearson Korelasyon Analizi.....	16
3.2.6. Normallik Analizi .....	17
3.2.7. Box Plot (Kutu Grafik) Yöntemi.....	17
3.2.8. Ki-Kare Analizi.....	18
3.3. Veri Seti.....	18
3.3.1. Verilerin Dağılımının İncelenmesi .....	22
3.3.2. Veri Setinin Hazırlanması.....	23
4. UYGULAMA VE BULGULAR .....	38
4.1. Sürekli Bağımlı Değişken ile Sınıflandırma Modeli (Model-1) .....	38
4.1.1. Sürekli bağımsız değişkenlere ait hesaplamalar .....	39
4.1.2. Sürekli bağımlı değişken ile bağımsız değişkenlerin karşılaştırılması .....	40
4.1.3. Sürekli bağımlı değişkenin kategorik hale getirilmesi .....	43
4.1.4. Sürekli bağımlı değişken ile sınıflandırma modeli bulguları.....	43
4.2. Kategorik Bağımlı Değişken ile Sınıflandırma Modeli (Model-2) .....	49
4.2.1. Bağımlı değişken ile bağımsız değişkenlerin karşılaştırılması .....	50
4.2.2. Sürekli Bağımlı Değişken İle Sınıflandırma Modeli Bulguları .....	51
4.3. Model-1 ve Model-2 Sonuçlarının Değerlendirilmesi.....	55
5. SONUÇ.....	58
KAYNAKLAR.....	60

## TABLÖLAR LİSTESİ

Tablo 3.1: Veri Seti Öznitelikleri.....	20
Tablo 3.2: Verilerin Dağılımı.....	22
Tablo 3.3: Projelerin Organizasyon Tipi Özelliklerine Ait Bulgular.....	23
Tablo 3.4: Projelerin Organizasyonel Rol Özelliklerine Ait Bulgular.....	24
Tablo 3.5: Projelerin Müşteri Organizasyon Tipi Özelliklerine Ait Bulgular .....	24
Tablo 3.6: Projelerin Geliştirme Türü Özelliklerine Ait Bulgular .....	25
Tablo 3.7: Projelerin Uygulama Alanı Özelliklerine Ait Bulgular .....	25
Tablo 3.8: Projelerin Devlet Politikası Etkisi Özelliklerine Ait Bulgular.....	26
Tablo 3.9: Projelerin Geliştirici İşe Alım Politikası Özelliklerine Ait Bulgular.....	26
Tablo 3.10: Projelerin Geliştirici Eğitimi Özelliklerine Ait Bulgular.....	27
Tablo 3.11: Projelerin Önceki Sisteme İlişkin Üst Yönetimin Görüşü Özelliklerine Ait Bulgular.....	27
Tablo 3.12: Projelerin Kullanıcı Bilgisayar Deneyimi Özelliklerine Ait Bulgular.....	27
Tablo 3.13: Projelerin Proje Yöneticisi Deneyimi Özelliklerine Ait Bulgular .....	28
Tablo 3.14: Projelerin Danışman Kullanılabilirliği Özelliklerine Ait Bulgular.....	28
Tablo 3.15: Projelerin VTYS Uzmanı Kullanılabilirliği Özelliklerine Ait Bulgular.....	28
Tablo 3.16: Projelerin Yazılım Aracı Deneyimi Özelliklerine Ait Bulgular .....	29

Tablo 3.17: Projelerin Ekip Seçimi Özelliklerine Ait Bulgular .....	29
Tablo 3.18: Projelerin Gelir Memnuniyeti Özelliklerine Ait Bulgular .....	30
Tablo 3.19: Projelerin Takvim Kalitesi Özelliklerine Ait Bulgular.....	30
Tablo 3.20: Projelerin Yazılım Metodolojisi Özelliklerine Ait Bulgular .....	30
Tablo 3.21: Projelerin Kullanılan Programlama Dili Özelliklerine Ait Bulgular .....	31
Tablo 3.22: Projelerin Kullanılan VTYS Özelliklerine Ait Bulgular .....	32
Tablo 3.23: Projelerin Açık Kaynak Yazılımı Kullanımı Özelliklerine Ait Bulgular .....	32
Tablo 3.24: Projelerin Dış Kaynak Kullanım Düzeyi Özelliklerine Ait Bulgular .....	32
Tablo 3.25: Projelerin Yazılımın Yeniden Kullanım Derecesi Özelliklerine Ait Bulgular .....	33
Tablo 3.26: Projelerin Standartların Kullanımı Özelliklerine Ait Bulgular.....	33
Tablo 3.27: Projelerin Gereksinim Doğruluk Düzeyi Özelliklerine Ait Bulgular .....	34
Tablo 3.28: Projelerin Teknik Dokümantasyon Özelliklerine Ait Bulgular .....	34
Tablo 3.29: Projelerin Kod Açıklamaları Özelliklerine Ait Bulgular .....	35
Tablo 3.30: Projelerin Kullanım Kılavuzu Özelliklerine Ait Bulgular .....	35
Tablo 3.31: Projelerin Tekrar Kullanılabilirlik Gerekliliği Özelliklerine Ait Bulgular .....	35
Tablo 3.32: Projelerin Ürün Karmaşıklığı Özelliklerine Ait Bulgular.....	36
Tablo 3.33: Projelerin Güvenilirlik Gereksinimleri Özelliklerine Ait Bulgular .....	36

Tablo 3.34: Projelerin Donanım Gerekliliği Özelliklerine Ait Bulgular .....	37
Tablo 4.1: Araştırmaya Ait Betimsel Bulgular .....	39
Tablo 4.2: Kategorik Değişkenler ile Sürekli Bağımlı Değişken Arasındaki Farklılık Analizi Sonuçları .....	40
Tablo 4.3: Sürekli Değişkenler ile Bağımlı Değişken Arasındaki İlişkiye Ait Sonuçlar.....	42
Tablo 4.4: Bağımlı Değişkenin Kategorize Edilmiş Sonucu .....	43
Tablo 4.5: Referans Noktasının Belirlenmesi .....	45
Tablo 4.6: Ki-Kare Değeri Tablosu - Sıfıncı Adıma İlişkin .....	45
Tablo 4.7: Model-1 Katsayılarına İlişkin Omnibus Testi .....	46
Tablo 4.8: Model-1 Uyum İstatistiği Değerleri.....	46
Tablo 4.9: Model-1'e Ait Katsayı Tahminleri .....	47
Tablo 4.10: Model-1'in Sınıflandırma Başarısı .....	48
Tablo 4.11: Bağımsız Değişkenler ile Kategorik Bağımlı Değişken Arasındaki İlişkinin Anlamli Olan Değişkenler .....	50
Tablo 4.12: Referans Noktasının Belirlenmesi .....	51
Tablo 4.13: Ki-Kare Değeri Tablosu - Sıfıncı Adıma İlişkin .....	51
Tablo 4.14: Model-2 Katsayılarına İlişkin Omnibus Testi .....	52
Tablo 4.15: Model-2 Uyum İstatistiği Değerleri.....	52

Tablo 4.16: Model-2'ye Ait Katsayı Tahminleri ..... 53

Tablo 4.17: Model-2'nin Sınıflandırma Başarısı ..... 55

Tablo 4.18 : Sınıflama Modellerinin Karşılaştırması..... 57



## ŞEKİLLER LİSTESİ

Şekil 3.1: Box Plot Grafiği..... 17

Şekil 3.2: Aykırı Değerlere Ait Box Plot Grafiği..... 22



## SİMGELER VE KISALTMALAR LİSTESİ

ANOVA	Analysis of Variance
BOA	Bayes Optimizasyon Algoritması
CMMI	Capability Maturity Model Integration
COCOMO	Constructive Cost Model
F	Tek Yönlü Anova Değeri
ISBSG	International Software Benchmarking Standards Group
Max.	Maksimum
Min.	Minimum
MLP	Multi Layer Perseptron
MLR	Multinomial Lojistik Regresyon
n	Sıklık
NASA	National Aeronautics and Space Administration
Ort.	Ortalama
p	Olasılık
PCA	Principal Component Analysis
R	Pearson Korelasyon Analizi Değeri
SEERA	Software Engineering in Sudan
SPSS	Statistical Package for the Social Sciences
S.S	Standart Sapma
T	Bağımsız Örneklem T-testi Değeri
YEEM	Yıldız Effort Estimation Metrics
X <sup>2</sup>	Ki Kare Değeri

# 1. GİRİŞ

Yazılım projelerinde efor tahmini, yazılım geliştirme yaşam döngüsünün kritik bir adımıdır. Efor tahmini yapmak proje yöneticilerine yeni bir proje için teklif verme, planlama ve kaynak tahsisi gibi önemli kararlar alırken rehberlik sağlamaktadır. Projelerin mümkün olduğunca belirgin olması, potansiyel riskleri azaltmakta, bu sebeple yüksek doğruluk ve sağlamlığa sahip yazılım efor tahminleme modelleri, projelerin başarılı olabilmesinde önemli rol oynamaktadır.

Hızla gelişen yazılım endüstrisinde yoğun rekabetin önüne geçmek amacıyla doğru bir şekilde yazılım geliştirme eforunu tahmin etmek büyük bir öneme sahiptir. Yanlış tahminler, maliyet ve kalite gibi birçok konuda kayıplara neden olabilmektedir. Yazılım geliştirme projelerinde bugünün en yaygın olarak kullanılan efor tahmini yöntemleri, çoğunlukla çalışanların subjektif yargılarına dayanan yaklaşımlardır. Ancak, proje başarısı üzerinde büyük etkisi olan bu kritik konunun, birçok değişkenden etkilenebilecek ve güvenilirliği sınırlı olabilecek insan tahminlerine bağlı olması uygun görülmemektedir. Ek olarak, tahmin kalemlerinin çeşitlenmesiyle birlikte bu yöntemlerin uygulanması giderek karmaşık hale gelerek iş yükünü artırabilmektedir.

Yazılım efor tahmini, bir yazılım mühendisliği projesi için gerekli olan tüm kaynakların önceden tahmin edilmesi işlemidir. Yazılım efor tahmini kavram olarak basit görünse de, pratikte oldukça zor ve karmaşıktır. Bu nedenle birçok yazılım projesi, öngörülen zamanda tamamlanamamış veya maliyet açısından beklenenden daha yüksek maliyetlerle karşılaşmıştır. Bu zorluklar, projenin özelliklerinden kaynaklanmakla birlikte, bilgi eksikliği, öznel yorumlar, maliyet ayırım hataları ve proje risklerinin tam olarak tahmin edilememesi gibi faktörlerden de kaynaklanabilmektedir. Bir yazılım projesine başlarken, proje süresi, maliyeti ve çalışacak personel sayısı gibi faktörlerin önceden tahmin edilerek dikkate alınması proje başarısı üzerinde önemli rol oynamaktadır.

## **1.1. Amaç**

Yazılım projelerinde efor tahmini, geliştirme sürecinin başlangıcında yapılan ve etkili proje yönetimi için kritik bir unsurdur. Bu sürecin amacı, projenin ilerleyişi ve aşamaları hakkında daha iyi bir gelecek görüşü sağlamaktır. Doğru tahminler, proje yöneticisinin maliyetleri, zamanı ve kaynakları daha iyi tahmin etmesine yardımcı olacaktır. Yazılım projesi eforunu doğru şekilde tahmin etme yeteneği, proje kabul veya projenin reddi gibi yönetim kararlarını da etkilemektedir. Yanlış tahminler, kaynak israfına veya teslimat süresinin gecikmesine neden olabilirken, eksik tahminler proje personelinin az veya çok çalışmasına ve bütçe aşımalarına yol açabilmektedir. Bu durum, proje sonuçlarının zamanında, belirlenen bütçede ve kaliteli bir şekilde teslim edilmesini olumsuz etkilemektedir.

Yazılım efor tahmini, hala karmaşık bir konudur ve daha doğru sonuçlar elde etmek için sürekli geliştirilmekte ve araştırılmaktadır. Son yıllarda, teknolojinin ilerlemesiyle birlikte, büyük verilerin analizi ve makine öğrenimi yöntemlerinin kullanımı yazılım efor tahmin sürecini iyileştirmekte önemli bir rol oynamıştır.

Bu araştırmanın temel amacı, doğruluk oranı yüksek ve nesnel yazılım efor tahmini sağlayabilecek bir model geliştirmektir. Proje eforunu etkileyen unsurları belirlemek, bu değişkenleri ele alarak proje başarısını öngöreceği bir model kurmak üzerine odaklanılmıştır. Yazılım mühendisliğinde başarılı bir proje, proje kapsam, zaman ve maliyet hedeflerini başarıyla gerçekleştiren projedir. Bu tez çalışması, daha doğru bir yazılım efor tahmini ile projelerin başarıyla tamamlanmasına yardımcı olacak bütüncül bir yazılım efor sınıflama modeli sunmaktadır.

## **1.2. Literatür Taraması**

Bu araştırma ile geliştirilen yazılım efor sınıflama modeli literatürde yer alan lojistik regresyon algoritması ve efor tahmini çalışmalarından ve belirlenen veri setinden yola çıkarak oluşturulmuştur. Bu bölümde bu konulara ilişkin makaleler açıklanmıştır.

### **1.2.1. Lojistik regresyon ile ilgili çalışmalar**

Lojistik regresyon yöntemi, yazılım efor tahmin çalışmalarında farklı amaçlarla kullanılmıştır. Literatürde yer alan ilgili çalışmalar aşağıda özetlenmiştir.

Sentas ve arkadaşlarının yaptığı çalışmada [1], yazılım efor tahmininde yaygın bir sorun olan eksik verilerin manipülasyonu konusu ele alınmıştır. Çalışmalarda genellikle, eksik gözlemleri göz ardı etmek gibi basit yöntemler kullanılmakta, ancak bu değerli bilgilerin kaybına neden olabilmektedir. Bu nedenle, bu çalışma ile kategorik kayıp değerlerin tahmin edilmesi için bir yöntem araştırılmıştır. Çoklu Doğrusal Regresyon'un (MLR) bu amaca hizmet edebileceği düşünülmüş ve ISBSG veri tabanındaki projeler üzerinde test edilmiştir. MLR'nin diğer eksik veri teknikleriyle karşılaştırılması, özellikle yüksek eksik veri yüzdesi durumlarında etkili olduğu tespit edilmiştir. Bu araştırma, tamamen kategorik bir yöntemin diğer popüler yöntemlerle en azından eşit derecede iyi performans gösterebileceğini ortaya koymuştur.

2005 yılında Takagi ve arkadaşları tarafından yapılan çalışmada, yazılım geliştirme projelerinde karşılaşılan risklerin belirlenmesi ve karakterize edilmesi için bir yaklaşım önerilmiştir. Bu yaklaşım, gereksinimler, tahminler, planlama, ekip organizasyonu ve proje Yönetim aktiviteleri olarak beş farklı konuyu içeren bir anket kullanarak projelerin karışıklık düzeyini belirlemekte ve lojistik regresyon analiziyle bu karışıklık düzeyi ile projelerin özellikleri arasındaki ilişkiyi incelemektedir. Yapılan deneysel çalışmalar, önerilen yöntemin projeleri doğru şekilde sınıflandırmada etkili olduğunu göstermektedir. Ayrıca, bu yaklaşımın riskli projeleri tespit etme konusunda da geçerli olduğu sonucuna varılmıştır. Araştırma, yazılım geliştirme projelerinde karşılaşılan sorunların istatistiksel analizlerle adım adım incelenmesiyle, projelerin sonuçları için basit bir tahmin modeli oluşturulabileceğini göstermektedir [2].

2015 yılında Christiansen ve arkadaşlarının yaptığı çalışmada, yazılım geliştirme projelerindeki risklerin tahmin edilmesi amacıyla çoklu lojistik regresyon yöntemi kullanılmıştır. Bu yöntem, yazılım geliştirme sürecinin kontrolü amacıyla proje risklerinin sınıflandırması ile nedensel risk faktörleri analizlerini içermektedir. Lojistik regresyon ile birleştirilmiş faktör analizi, yazılım geliştirme süreçlerinde ortaya çıkabilecek risk faktörlerinin belirlenmesi, riskin önceden tahmin edilmesi ve en aza indirilmesi hedeflenmiştir. Analitik sonuçların, stratejilerin oluşturulması ve geliştirilmesi ile birlikte hata risklerinin yönetilmesi, kontrol edilmesi ve azaltılması açısından önemli sorunların vurgulanmasına ve geliştirilmesine yol açabileceği düşünülmektedir. Çalışma kapsamında yazılım geliştirme risk analizine ilişkin oluşturulan sınıflandırma sonuçları % 90 oranında genel tahmin doğruluğu göstermiştir. Bu çalışma ile yazılım geliştirme proje başarısını

üzerinde önemli bir etken olan risk yönetimi ele alınmış ve risk faktörlerinin sınıflandırılması, tahmin edilmesi ve azaltılması ile ilgili önerilerde bulunulmuştur [3].

### **1.2.2. Yazılım efor tahmini ile ilgili çalışmalar**

Yazılım efor tahmini ile ilgili literatürde çeşitli yöntem ve farklı veri setleri kullanılarak çok fazla çalışma yapılmıştır. Literatürde yer alan ilgili çalışmalar aşağıda özetlenmiştir.

2007 yılında Ayyıldız tarafından hazırlanan çalışma ile yapay sinir ağı kullanarak yeni bir yazılım efor tahmini modeli geliştirilmiştir. Model, Türkiye'den bir firma tarafından sağlanan veriler ve özel bir ölçüt kümesi olan YEEM (Yıldız Effort Estimation Metrics) ile oluşturulmuştur. Literatürdeki çalışmaların çoğunda COCOMO (Constructive Cost Model) 81 veri kümesi kullanılmıştır, ancak bu çalışmada YEEM'in daha başarılı sonuçlar verdiği görülmüştür. MLP (Multi Layer Perseptron) ve Elman yapay sinir ağı modelleri kullanılarak yapılan tahminlerde, Elman modelinin daha başarılı olduğu tespit edilmiştir. Ancak, daha büyük veri kümelerinin kullanılmasıyla daha iyi sonuçlar elde edilebileceği öne sürülmüştür. Ölçüt kümesinin doğruluğu ve çeşitliliği, yapay sinir ağlarının başarısını önemli ölçüde etkilemektedir. Bu çalışma, gelecekteki araştırmalara yönelik öneriler sunmakta ve ölçüt kümesinin daha geniş bir şekilde kullanılması gerektiğini vurgulamaktadır [4].

2008 yılında Sezer tarafından yapılan çalışma, yazılım projelerinin efor tahminini yapmak için çok katmanlı ileri beslemeli yapay sinir ağı yöntemini ele almaktadır. Çalışmada sadece yazılımın büyüklüğü, satır kod sayısı, çalışan kişi sayısı ve yazılımdaki bozulma oranı kullanılmıştır. Yazılım eforunu etkileyen faktörlerin belirlenmesi için farklı yöntemler incelenmiş ve gerçekleştirilmiş yazılım projelerinden veriler toplanmıştır. Bu veriler önceden işlenmiş ve eğitim ile test verileri olarak ayrılmıştır. Yapay sinir ağı yöntemi, COCOMO 2000 ile karşılaştırıldığında daha iyi sonuçlar vermiştir. Ancak, daha fazla faktörün dikkate alınması için daha geniş veri setlerine ihtiyaç duyulduğu ortaya konulmuştur. Yazılım firmaları, gerçekleştirilen ve tahmin edilen yazılım projelerinin verilerini saklayarak ve yöneterek kendi tahmin yöntemlerini oluşturması önem arz etmektedir. Ayrıca, yazılım efor tahmininin yanı sıra kod satırı tahmini yaparak zaman yönetimini iyileştirmelidirler. Başarılı bir yazılım projesi iyi tahminlerin yanı sıra doğru gereksinim analizi ve planlama da gereklidir [5].

2008 yılında Bakır ve arkadaşları yazılım efor tahmininin çoğunlukla regresyon problemi olarak ele alınarak nokta tahmini olarak ilerlediğini, aşırı/eksik tahminleri engellemek adına, tam değerler yerine tahmin aralıklarını öngörmek daha pratik olabileceğini öne sürmüşlerdir. Bu sebeple çalışmalarında, yazılım efor tahmini yöntemi olarak sınıflandırma yaklaşımı sunulmuşlardır. Bu yaklaşım, her biri belirli bir çaba aralığına karşılık gelen çeşitli yazılım projelerini sınıflandırmayı amaçlamaktadır. Küme analizi, efor aralıklarını belirlemek için kullanılırken, farklı sınıflandırma algoritmaları, ilgili efor sınıflarını tanımlamak için ele alınmıştır. Bu önerilen yaklaşım, yedi kamuya açık veri kümesine uygulanmıştır ve deneysel sonuçlar, çaba tahmini için elde edilen başarı oranlarının %90-100 aralığında olduğunu göstermiş, literatürdeki nokta tahmini ile ilerlenen çalışmalarla uyumlu sonuçlar elde edildiği belirtilmiştir [6].

2011 yılında O. Fedotova, L. Teixeira ve H. Teixeira tarafından yazılan makalede, yazılım geliştirme efor tahmini konusunda, Çoklu Doğrusal Regresyon (MLR) tekniğinin kullanımını ve etkinliğini incelenmiştir. Bu çalışma, yazılım geliştirme organizasyonlarında resmi bir efor tahmin yönteminin gerekliliğini CMMI kapsamında değerlendirmiştir. Yazılım geliştirme ve test ekipleri için geliştirilen modelin kullanımıyla elde edilen sonuçlar ile uzman görüşleri kıyaslanmış ve model ile elde edilen sonuçların daha iyileştirici olduğu belirtilmiştir. Ancak, yazılım geliştirme ekibi için uygun bir model elde etme konusunda zorluklar yaşanmış ve gelecekteki tahminler için belirli değişkenlerin tanımlanması ve sınıflandırılması önerilmiştir [7].

2012 yılında Nassif, Ho ve Capretza, yazılım tahmininin zorluklarına ve geliştirilmesine yönelik yeni modellere üzerine odaklanmıştır. Çalışmada kullanım senaryosu diyagramlarına dayalı bir log-lineer regresyon modeli ve çok katmanlı algılayıcı sinir ağı modeli, yazılım çabasını tahmin etmek için önerilmiştir. Yapılan deneylerde, log-lineer regresyon modelinin büyük projeler için, çok katmanlı algılayıcı sinir ağı modelinin ise küçük projeler için daha iyi sonuçlar verdiği ortaya konulmuştur [8].

Bir başka araştırma, yazılım efor tahmininde makine öğrenmesi modellerinin kullanımını sistemli bir şekilde analiz etmeyi amaçlamıştır. Son yirmi yılda yayınlanan 84 çalışmayı inceleyerek, makine öğrenmesi modellerinin çeşitli yönlerini, teknik türlerini, tahmin doğruluğunu, karşılaştırmalarını ve bağlamlarını incelenmiştir. Bulgular, makine öğrenmesi modellerinin genel olarak kabul edilebilir bir tahmin doğruluğuna sahip olduğunu ve makine öğrenmesi olmayan modellere kıyasla daha iyi performans gösterdiğini

göstermiştir. Farklı makine öğrenmesi modellerinin güçlü ve zayıf yönleri olduğu ve bu nedenle farklı tahmin bağlamlarını desteklediği görülmüştür. Araştırmacılara, makine öğrenmesi tabanlı yazılım efor tahmini üzerinde daha fazla ampirik çalışma yapmaları ve endüstride derinlemesine örnek olay çalışmaları veya anketler yürütmeleri önerilmiştir. Uygulayıcılar için ise, makine öğrenmesi modellerinin mantığını anlamak, güçlü yönleri ve tahmin bağlamlarıyla eşleşecek modelleri seçmek, özel proje veri setlerini paylaşarak çalışmalarda kullanılmasına ön ayak olmak gibi öneriler sunulmuştur [9].

2017 yılında Pospieszny ve arkadaşları tarafından yazılan makalede, ISBSG veri kümesi ve akıllı veri hazırlama kullanılarak üç farklı makine öğrenimi algoritmasıyla (Destek Vektör Makineleri, Sinir Ağları ve Genelleştirilmiş Doğrusal Modeller) efor ve süre tahmin modelleri oluşturulmuş ve bunların kuruluşlar için karar destek araçları olarak kullanılması hedeflenmiştir. Geleneksel tahmin tekniklerinin pratikte yetersiz kalması, makine öğrenimi algoritmalarının kullanımının önemini artırdığı, fakat bu yöntemlerin karmaşıklığı ve bireysel algoritmaların etkinliğine odaklanması nedeniyle organizasyonlar içinde henüz yaygın olarak kullanılmadığı vurgulanmıştır. Bu nedenle, makale, efor ve süre tahmini için bütünsel ve pratik bir yaklaşım sunarak bu alandaki boşluğu doldurmayı amaçlamıştır [10]

2018 yılında ele alınan bu makale ile yazılım projelerinin efor tahmininde çevresel faktörlerin önemini vurgulamak için 31 efor tahmini veri setini analiz ediyor. Veri setlerinin çoğu ABD ve Avrupa'daki organizasyonları temsil ederken, çevresel faktörlerin (kültürel, toplumsal, teknik) yeterince temsil edilmediği görülmüştür. Bu durum, genel veri setlerinin veya niteliklerinin ülkeye özgü faktörleri yeterince yansıtamayacağını göstermektedir. Bu nedenle, uluslararası düzeyde genel veri kümeleri veya nitelikler değerlendirilirken, çevresel faktörlerin ve bunların efor tahminine etkilerinin dikkate alınması önemlidir. Bu faktörlerin etkisinin doğru bir şekilde değerlendirilmesi için daha fazla araştırmaya ihtiyaç vardır [11].

2019 yılında Gültekin tarafından yapılan çalışma ile yazılım efor tahminini geliştirmek için üç farklı modelin ortaya konulduğu bir tez öne sürmüştür. Her bir model, farklı veri setleri ve yazılım metodolojileri üzerinde uygulanmıştır. Birinci model, regresyon tabanlı bir yaklaşım kullanırken, ikinci model yapay sinir ağlarına dayanmaktadır. Üçüncü model ise yazılım metodolojileri dikkate alınarak makine öğrenme algoritmaları kullanılmıştır. Bu modellerin uygulanması sonucunda, yazılım efor tahmininde daha doğru sonuçlar elde edildiği görülmüştür. Önerilen modellerin hem eksiklikleri hem de pozitif yönleri

tartışılarak, yapay sinir ağları tabanlı modelin daha iyi sonuçlar verdiği belirtilmiştir. Bu modelin, %9'luk bir sapma oranıyla efor tahmininde başarılı olduğu ve benzer çalışmalara göre daha iyi bir performans sergilediği belirtilmiştir [12].

2020 yılında Olgun tarafından yapılan çalışma ile çevik yaklaşımlı yazılım projelerinde efor tahmini hesaplaması için iki makine öğrenmesi modeli karşılaştırılarak en düşük hata oranlarını veren model belirlenmiştir. Veri seti öncelikle bir dizi ön işleme sürecinden geçirilerek, öğrenme aşamasında daha verimli kullanılacak hale getirilmiştir. Yazılım efor tahmini aşamasında, Derin Sinir Ağı Regresyon Modeli ve Güçlendirilmiş Karar Ağacı Regresyon Modeli, 10 katlı Çapraz Doğrulama Yöntemi ile değerlendirilerek Ortalama Mutlak Hata değerleri üzerinden karşılaştırılmıştır. Karşılaştırma sonucunda Derin Sinir Ağı Regresyon modelinin, efor tahmininde çok daha iyi bir performans sergilediği gözlemlenmiştir [13].

2021 yılında Al-Khazraji tarafından hazırlanan çalışmada, makine öğrenmesi tekniklerine dayalı yeni bir yazılım efor tahmin yaklaşımı sunulmuştur. Önerilen yöntem, özellik seçimi ve etkilenmeyen özelliklerin çıkarılması için faktör analizi kullanılmıştır. Ardından, seçilen özelliklerin sınıflandırılması için Naive Bayes sınıflandırıcı uygulanmıştır. Bu yöntem, NASA yazılım veri kümesi üzerinde test edilmiş ve %94,15 kesinlik sağlamıştır. Bayes Optimizasyon Algoritması (BOA)'nın kullanımı ile diğer klasik makine öğrenimi yöntemlerine göre daha yüksek hassasiyet sağladığı ifade edilmiştir. Gelecekteki çalışmalarda, BOA stratejisi ve Naive Bayes parametrelerini kullanarak faktör analizinin daha fazla araştırılması önerilmiştir. Bu yöntemin, karmaşık problemlerin çözümünde ve tahmin sorunlarında kullanılabilmesi gözlemlenmiştir [14].

2021 yılında Kara ve Şamlı tarafından ele alınan çalışmada ise, COCOMO81, COCOMONASA ve COCOMONASA2 veri setleri kullanılarak 27 farklı algoritma test edilmiş ve performansları çeşitli ölçütlerle değerlendirilmiştir. Sonuçlar, algoritmaların farklı veri setleri üzerinde farklı sonuçlar üretebildiğini ve bazı algoritmaların belirli veri setlerinde daha iyi çalıştığını göstermiştir. Özellikle, veri setlerindeki özniteliklerin tahmin sonuçlarını etkilediği belirlenmiştir. Bazı algoritmaların başka bir sınıflandırıcı ve onun parametrelerini alabilen algoritmalarından daha iyi performans gösterdiği görülmüştür [15].

Derya tarafından 2023 yılında ele alınan çalışma ile Finlandiya, Kemerer, Çin ve Maxwell veri kümeleri kullanılarak modeller oluşturulmuş ve bu modellerin performansı

analiz edilmiştir. Modeller dokuz farklı algoritma kullanılarak oluşturulmuş ve 10 kat çapraz doğrulama tekniği kullanılmıştır. Hibrit özellik seçimi yaklaşımlarıyla optimize edilen modellerin, farklı algoritmalar ve özellik seçme teknikleri kullanılarak elde edilen başarılı sonuçlara ulaştığı gözlemlenmiştir. Özellik seçimi ile elde edilen modellerin, orijinal veri seti modellerine göre daha başarılı olduğu belirlenmiştir [16].

### **1.2.3. SEERA veri seti ile ilgili çalışmalar**

Çalışmada kullanılan SEERA (Software Engineering in Sudan) [17] efor tahmini veri seti ile ilgili yapılan çalışmalar aşağıda anlatılmıştır.

2020 yılında Mustafa ve Osman tarafından Haziran 2019 ile Şubat 2020 tarihleri arasında veriler toplanarak bu veri seti oluşturulmuştur. Veri seti, Sudan'daki 42 kuruluşu temsil eden 120 yazılım geliştirme projesinden oluşmakta olup, 76 öznitelik içermektedir. Ayrıca mevcut efor tahmini veri setlerinin aksine, meta veriler, alt nitelikler ve orijinal ham verilerle zenginleştirilmiştir. Bu veri seti, gelişmekte olan ülkelere ve kısıtlı ortamlardaki yazılım endüstrilerine daha uygun veriler sağlama konusunda mevcut boşluğu doldurması hedeflenmiştir.

Veri toplama süreci, veriye dahil edilene kuruluşlar ve proje özelliklerine ilişkin bilgiler derlenerek makale yazılmıştır [18]. Bahse konu makale yerel faktörlerin yazılım projesi eforu üzerindeki etkilerini göstermek ve veri seti kalitesini ortaya koymak için farklı çalışmalarda kullanılan veri setleri ile karşılaştırma ve analizleri içermektedir. Mustafa ve Osman devam eden süreçte bu veri setini kullanarak çevresel ve sosyoekonomik faktörlerin teknik maliyet faktörleri üzerindeki etkisini araştırmak, efor tahmin modeli geliştirmek ve efor faktörlerinin yaygınlığını ve büyüklüğünü ilgili PROMISE veri kümeleriyle karşılaştırmayı planlamışlardır.

2023 yılında Aslan proje yönetiminde makine öğrenmesi algoritmalarını değerlendirmek amacıyla 4 farklı veri seti, 6 farklı makine öğrenme algoritması ve 3 farklı veri işleme yöntemi ile analiz gerçekleştirmiştir. Bu çalışmada SEERA veri seti de kullanılmıştır. Makine öğrenmesi algoritmaları ile yazılım proje yönetimi için farklı yöntemlerin kapsamlı bir değerlendirmesi yapılmış, en iyi sonuçları veren veri seti, yöntem ve algoritma eşleştirmeleri belirlenmiştir. PCA analizi ile elde edilen sonuçların, bu araştırmada kullanılan alternatif yöntemlere kıyasla üstün sonuçlar elde etmedeki etkinliği vurgulanmış, bu yöntem ile Rastgele Orman ve En Yakın Komşular algoritmaları, usp05,

usp05-ft ve SEERA veri kümeleri birlikte kullanıldığında sürekli olarak en iyi sonuçlar alındığı tespit edilmiştir. Bu çalışma ile makine öğrenimi modellerinin performansını ve doğruluğunu artırmak için çeşitli özellik seçme tekniklerini, boyut azaltma yöntemlerini ve veri ön işleme stratejilerini araştırmaya odaklanılacağı belirtilmiştir [19].

2024 yılında Mustafa ve Osman 2020 yılında geliştirdikleri veri seti ile proje başlangıç döneminde proje eforunu ve süresini tahmin etmek için Rasgele Orman algoritmasını kullanarak bir model geliştirmiştir. SEERA veri setinin geniş sayıda öznitelige sahip heterojen bir veri seti olması, eksik değerler ile aykırı değerler içermesi, özniteliklerin kategorik ve sürekli olarak farklılık göstermesi sebebiyle veri ön işleme yöntemlerine ihtiyaç duyulmuştur. Çalışmada bağımsız değişkenler ile gerçekleşen efor ve süre özellikleri arasındaki ilişkinin derecesini belirlemek için bir korelasyon analizi yapılmıştır. Çabayı ve süreyi tahmin etmek için, özniteliklere en uygun modelin Rastgele Orman algoritması olduğuna karar verilmiştir [20]. Rastgele Orman algoritması ile kurulan model sonucunda “Gerçek Efor” ile “Ekip Boyutu”, “Özel Ekip Üyeleri” ve “Nesne Sayısı” değişkenlerinin anlamlı ilişkiye sahip olduğu tespit edilmiştir.

Bu çalışmada aşırı örnekleme, öznitelik seçimi, heterojenliğin ve yerel çevresel faktörlerin model doğruluğu üzerindeki etkileri ele alınmıştır. Deney sonuçları, kurulan modelin uzman kararından ortalama %60 daha doğru olduğunu ve daha güvenilir olduğunu, literatürde bildirilenlere benzer doğrulukta sonuçlar sağladığını göstermiştir.

Yapılan literatür araştırmaları göz önünde bulundurularak yazılım efor tahminleme çalışmaları değerlendirildiğinde;

- Açık kaynak kodlu veri seti sayısının sınırlı olması sebebiyle çalışmaların benzer veri setleri ile yapıldığı,
- Kullanılan veri setlerinin bir çoğunun güncelliğini kaybetmiş olması, eksik veri ve aykırı değerler içermesi sebebiyle veri kalitesi sorunlarının yaşandığı,
- Veri setlerinde yer alan öznitelik sayısı kısıtı sebebiyle yapılan çalışmaların sınırlandığı,
- Yazılım efor tahmini yöntemi metriği olarak genellikle kod satır sayısının ele alındığı ve farklı doğruluk ölçütlerinin daha az benimsendiği,
- Yazılım efor tahmininde genellikle nokta tahmini ile ilerlendiği, yöntem olarak çoğunlukla regresyon ve yapay sinir ağları yöntemlerinin kullanıldığı,

görülmektedir.

Bahse konu hususlar göz önünde bulundurularak veri seti araştırması yapıldığında SEERA (Software Engineering in Sudan) [17] efor tahmini veri setinin çalışmada kullanımına karar verilmiştir. Bu veri seti öznitelik çeşitliliğinin yüksek olması, çevresel ve sosyoekonomik özelliklere yer vermesi, verilerin meta veriler ve alt niteliklerle zenginleştirilmiş olması, farklı sektörlerdeki yazılım projelerini ele alınması sebebiyle araştırmaya iyi bir altlık sağlayacağı değerlendirilerek seçilmiştir.

Yazılım efor tahminleme alanında yapılan literatür taramaları neticesinde yazılım efor tahmininde nokta tahmini ile ilerlendiği ve çoğunlukla regresyon ile yapay sinir ağları yöntemlerinin kullanıldığı tespit edilmiştir. Regresyon tabanlı modeller giriş değerlerine bağlı olarak çalışmakta ve sonuç üretmektedir. COCOMO modelleri gibi kod satır sayısını giriş değeri olarak ele alan modellerde giriş değerine bağımlı sonuçlar elde edilmekte, hatta gelişen programlama dilleri sebebiyle kod satır sayısı tespiti göreceli hale gelebilmektedir. Yeni yaklaşımli çalışmalarda genellikle bu sınırlılıkları farklı makine öğrenmesi modelleri ile aşmaya çalışıldığı gözlemlenmektedir.

Literatürdeki çalışmalar ele alındığında yazılım efor tahmini yöntemi olarak sınıflandırma yaklaşımının noktasal tahmin yapmak üzerine yoğunlaştığı görülmektedir. Bu sebeple çalışmada birçok özelliği ile literatürdeki veri setlerinden daha kapsamlı ve yeni olan bir veri seti kullanılması, yazılım efor tahmin çalışmalarının ise sınıflandırma modeli ile oluşturulması kararlaştırılmıştır.

## **2. GENEL KISIMLAR**

### **2.1. Yazılım Proje Yönetimi**

Yazılım proje yönetimi, özel hedeflere yönelik bir sistem yönetimi yaklaşımıdır. Bu yaklaşım, projeyi başlatma, planlama, yürütme ve uygulama, izleme ve kontrol etme ile proje kapanışı gibi çeşitli aşamalarda gerçekleştirilir. Temel görevler arasında uygulama planının geliştirilmesi, kaynakların organize edilmesi, kalite yönetimi ve maliyet yönetimi bulunur. Bu süreç, yazılımın tanımlanmasını, geliştirilmesini ve bakımını içerir. Yazılım proje yönetiminin amacı, belirlenen çizelgelere, maliyetlere ve diğer gereksinimlere uygun şekilde projeyi ilerletmek ve tamamlanmasını sağlamaktır. Başarılı proje, belirlenen kapsamı, belirlenen kalitede, zamanında ve hesaplanan maliyet içinde tamamlanan projedir.

Proje yönetimi, çok çeşitli değişkenleri içeren dinamik bir süreçtir. Bu nedenle, tek bir yöntemle bir projeyi yönetmenin mutlak bir yolu yoktur. Farklı projeler farklı metodolojiler ile yönetilebilmekte, proje yönetim metodolojisi veya yazılım süreç modelleri, projelerin etkin bir şekilde yönetilmesini desteklemektedir.

### **2.2. Yazılım Projelerinde Efor Tahmininin Önemi**

Yazılım efor tahmini, bir projenin başarılı bir şekilde yönetilmesi ve tamamlanması için kritik bir öneme sahiptir. Doğru bir efor tahmini, proje yöneticilerine projenin boyutunu, karmaşıklığını ve kaynak ihtiyaçlarını önceden belirleme konusunda kılavuzluk eder. Bu, yeni projeler için bütçe tahminlerinin ve planlamalarının yapılmasını sağlar, kaynakların etkin bir şekilde tahsis edilmesini mümkün kılar ve proje sürecinin yönetimini kolaylaştırır. Ayrıca, doğru efor tahmini, projenin tüm paydaşlarına güven verir ve proje süresince ortaya çıkabilecek potansiyel riskleri önceden belirleme ve önleme konusunda yardımcı olur. Böylece, yazılım projelerinin başarıyla tamamlanması ve hedeflenen sonuçlara ulaşılması için kritik bir adım olan efor tahmini, yazılım endüstrisinde rekabet avantajı elde etmek ve maliyetleri minimize etmek için hayati bir öneme sahiptir.

### 2.3. Yazılım Projelerinde Efor Tahmin Yöntemleri

Yazılım projelerinde efor tahmin yöntemleri, bir projenin geliştirme sürecinde gereken kaynakları (insan gücü, zaman, para vb.) tahmin etmek için kullanılan farklı teknikleri ifade eder. Bu yöntemler genellikle projenin boyutu, karmaşıklığı, gereksinimleri ve diğer faktörler dikkate alınarak belirlenir. Yazılım efor tahmin yöntemleri üç ana kategoriye ayrılır.

**Uzman Tahmini:** Proje yöneticileri ve deneyimli ekip üyeleri, geçmiş projelerine dayanarak subjektif olarak efor tahminleri yapabilirler. Bu yöntem, deneyime dayalı olarak oldukça hızlı olabilir, ancak güvenilirlik ve doğruluk açısından değişkenlik gösterebilir. Mevcut veya önceki projelerin benzerliklerine dayanarak efor tahminlerinin yapıldığı benzerlik temelli tahminler de uzman tahmini olarak ele alınabilir. Bu yöntem, benzer projelerin tarihçesinden elde edilen verilere dayanarak tahminlerin daha doğru olabileceği düşüncesine dayanır.

**Algoritmik Yöntemler:** Algoritmik yöntemler, bağımsız değişkenleri, bağımlı değişkenlerle (efor, maliyet gibi) ilişkilendirmek için matematiksel formüller içeren yaklaşımlardır. Bu yöntemlerde belirli parametreler kullanılarak bağımlı değişken çıktı olarak elde edilir. COCOMO modeli, yazılım projelerinin boyutu, karmaşıklığı ve diğer faktörlerin bir fonksiyonu olarak eforu tahmin etmek için kullanılan algoritmik bir yöntemdir. Temel, orta ve gelişmiş seviyelerdeki projeler için farklı denklemler sağlar [13].

**Algoritmik Olmayan Modeller:** Son yıllarda, bilim insanları, efor tahminlerini sağlamak amacıyla yapay zekaya dayalı metotları incelerken, yazılım geliştirme projelerindeki eforların öngörülmesinde makine öğreniminin gücünden yararlanan yeni bir yaklaşım geliştirmeye odaklanmışlardır. Bu yaklaşım, yazılım projelerindeki eforları tahmin etmek için yapay zeka disiplinlerinden olan makine öğrenimi yöntemlerini içerir [13].

### 3. GEREÇ VE YÖNTEM

Çalışmada elde edilen verilerin analizi için SPSS 25.0 paket programından faydalanılmıştır. Analizlerde önem düzeyi  $p=0.05$  ve  $p=0.01$  olarak kabul edilmiştir.

#### 3.1. Ön Bilgiler

##### 3.1.1. Bağımlı Değişken

Değişkenler arasında teorik olarak var olan sebep sonuç ilişkisi yapısının konu alındığı araştırmalarda araştırmacının amacını tanımlayan bağımlı değişken (sonuç değişkeni) olarak tanımlanmaktadır.

Çalışmada kullanılan SEERA (Software Engineering in Sudan) [17] veri setinde yer alan “Gerçek Efor” değişkeni bağımlı değişken olarak ele alınmıştır, sürekli bir değişkendir.

##### 3.1.2. Bağımsız Değişken

Bir deney veya araştırma sürecinde bağımlı değişkeni etkileyen, bağımlı değişkendeki değer değişmelerine neden olan bağımsız değişken olarak tanımlanmaktadır.

Çalışmada kullanılan SEERA (Software Engineering in Sudan) [17] veri setindeki “Gerçek Efor” değişkeni dışındaki tüm öznitelikler bağımsız değişken olarak ele alınmıştır. Tablo 3.1’de verilen bağımsız değişkenler, sürekli ve kategorik olarak iki farklı gruba ayrılmaktadır.

#### 3.2. Çalışmada Kullanılan Yöntemler

Bağımlı değişkenin kategorik olması nedeniyle araştırma hipotezinin incelenmesinde lojistik regresyon yöntemi kullanılmıştır. Araştırmada bağımsız değişkenlerin bağımlı değişken (proje eforu) üzerindeki etkisinin ortaya konması ve bu doğrultuda etkisi fazla olan değişkenler kullanılarak sınıflandırma modeli oluşturmak hedeflenmiştir. Elde edilen verilerin dağılımının incelenmesi için normal dağılım analizleri ve merkezi eğilim ölçümleri ile basıklık-çarpıklık değerlerinden yararlanılmış olup katılımcıların her bir ölçüme katılım düzeyleri ise ortalama ve standart sapma değerleri ile incelenmiştir. Projelerin özelliklerinin incelenmesinde frekans ve yüzde analizi yapılmıştır. Bağımlı değişkenin bağımsız değişkenlere göre farklılıkların incelenmesi için iki farklı yöntem kullanılmış ve bu doğrultuda iki farklı sınıflandırma modeli oluşturulmuştur.

İlk modelde sürekli bağımlı değişken ile bağımsız değişkenlere göre farklılıkların incelenmesi için ikili grupların karşılaştırılmasında Bağımsız Örneklem T-testi, ikiden fazla gruplar arasındaki farkın incelenmesi için Tek Yönlü Anova Analizi kullanılmış, sürekli değişkenler ile yapılan analizlerde Pearson Korelasyon Analizi uygulanmıştır. Bağımlı değişkenin kategorik iki grup haline getirilmesi sonucu bağımsız değişkenlerin bağımlı değişken üzerindeki etkisinin analizi ve sınıflandırılma modelinin kurulması için Lojistik Regresyon algoritması kullanılmıştır.

İkinci model kurulurken öncelikle bağımlı değişken kategorik hale getirilmiştir. Bağımlı değişkenin kategorik hali ile bağımsız değişkenler arasındaki ilişkilerin incelenmesinde Bağımsız Örneklem T-testi ve Ki-Kare Analizi uygulanmıştır. Yapılan analizler neticesinde ilişkileri anlamlı olan bağımsız değişkenler Lojistik Regresyon modeline dâhil edilerek alternatif sınıflandırma modeli geliştirilmiştir.

### **3.2.1. Lojistik Regresyon**

Regresyon analizi, istatistiksel bir metot olup, deneysel ve gözlemsel verilerin analizinde kullanılır. Temelde, bağımlı değişken ile bağımsız değişken arasındaki ilişkiyi inceleyerek bir tahminde bulunacak bir denklem oluşturmayı sağlar. Regresyon analizi, problemi belirleme, değişkenleri ayırt etme, veri toplama, model belirleme, parametre tahmini yapma ve sonuçları yorumlama gibi aşamaları içerir. Bu analizde kullanılan veriler deneysel veya gözlemsel, nicel veya nitel olabilir ve parametre tahmini genellikle en küçük kareler veya maksimum olabilirlik yöntemleri ile gerçekleştirilir. Temel amaç, değişkenler arasındaki ilişkiyi anlamak ve verilerin çevresi hakkında mümkün olduğunca çok bilgi edinmektir [21].

Lojistik regresyon, ağırlıklı olarak kategorik bağımlı değişkenlerin tahmininde kullanılan bir gözetimli makine öğrenimi algoritmasıdır. Bu teknik, bağımsız değişkenlerin etkisini değerlendirerek sonuç değişkeninin olasılığını tahmin etmek için kullanılır ve genellikle üstünlük oranı gibi ölçümlerle bu etkiyi açıklar. Lojistik regresyon, ikili, çoklu ve sıralı olmak üzere üç ana sınıfa ayrılır. İkili lojistik regresyon, sadece iki olası sonuç için kullanılırken, çoklu lojistik regresyon birden fazla sonuç değeri için uygundur. Sıralı lojistik regresyon ise hedef değişkeninin sıralı tercihlerini tahmin etmek için kullanılır ve genellikle üç veya daha fazla sonuç değerine sahip durumlar için kullanılır [21].

### 3.2.2. Uyum İyiliği İstatistikleri

Uyum İyiliği İstatistikleri, istatistiksel bir modelin veri setine ne kadar iyi uyduğunu değerlendirmek için kullanılan ölçütlerdir. Bu istatistikler, modelin sonuç değişkenini tanımlamakta ne kadar etkili olduğunu değerlendirmek için kullanılır [22]. Lojistik regresyon modellerinin uyumunu değerlendirmek için aşağıda tanımlaması yapılan uyum istatistikleri kullanılır ve her birinin farklı yönleri dikkate alınır. Daha düşük -2 log likelihood değerleri ve daha yüksek Cox & Snell R Kare veya Nagelkerke R Kare değerleri, bir modelin daha iyi uyduğunu göstermektedir.

**-2 Log likelihood:** Bir modelin uyumunu değerlendirmek için kullanılan bir istatistiktir. Log likelihood, bir modelin gözlemlenen veriler üzerinde ne kadar iyi uyduğunu ölçer. -2 log likelihood, modelin log likelihood değerinin iki katıdır ve modelin uyumunun değerlendirilmesinde sıklıkla kullanılır. Daha düşük -2 log likelihood değerleri, modelin daha iyi uyduğunu gösterir.

**Cox & Snell R Square:** Lojistik regresyon modellerinde kullanılan bir uyum iyiliği istatistiğidir. R-Kare'ye benzer bir ölçüdür ancak binom dağılımı gibi lojistik regresyonun özelliklerini dikkate alarak hesaplanır. 0 ile 1 arasında olan Cox & Snell R Kare değeri, 1 değerine ne kadar yakın olursa, modelin verilere o kadar iyi uyduğu kabul edilir.

**Nagelkerke R Square:** Cox & Snell R Square'ye benzer şekilde, modelin varyansının açıklama yüzdesini ölçer ancak daha yüksek değerler üretir. Nagelkerke R Kare, Cox & Snell R Kare'nin maksimum değerine göre normalleştirilmiştir ve genellikle Cox & Snell R Kare'nin iyileştirilmiş bir versiyonu olarak kabul edilir.

### 3.2.3. Bağımsız Örneklem T-test

Bağımsız örneklem t-testi, iki bağımsız grup arasında ortalamalara bakarak istatistiksel olarak anlamlı bir fark olup olmadığını test etmek için kullanılır. Bu testin uygulanabilmesi için bağımlı değişkenin sürekli, bağımsız değişkenin ise ikili gruplu ve kategorik olması gerekmektedir. Bu analiz öncesi veri seti aykırı değerlerden ayrıştırılmalı ve bağımlı değişken normal dağılıma sahip olmalıdır.

### 3.2.4. Tek Yönlü Anova Testi

ANOVA, “analysis of variance” ifadesinin kısaltması olup, “varyans analizi” anlamına gelmektedir. Tek Yönlü Anova Testi, bir bağımlı değişkenin gruplar arasındaki ortalamalarının farklı olup olmadığını belirlemek için kullanılır. Bu test, gruplar arasındaki varyansın farklılığını karşılaştırarak gruplar arasında istatistiksel olarak anlamlı bir fark olup olmadığını belirler. Tek Yönlü Anova Testinin uygulanabilmesi için, bağımlı değişkenin sürekli, bağımsız değişkenin ise üç ile üstü gruplu ve kategorik olması gerekmektedir.

Tek Yönlü Anova Testi sonucunda elde edilen p-değeri, gruplar arasında istatistiksel olarak anlamlı bir fark olup olmadığını belirlemek için kullanılır. Eğer p-değeri belirlenen anlamlılık düzeyinden (genellikle 0.05) daha küçükse, gruplar arasında anlamlı bir fark olduğu kabul edilir ve gruplar arasında en az birinin diğerlerinden istatistiksel olarak farklı olduğu sonucuna varılır.

### 3.2.5. Pearson Korelasyon Analizi

Pearson Korelasyon Analizi, sürekli değişkenler arasındaki ilişkiyi ölçmek için kullanılan istatistiksel bir yöntemdir. Bu analiz, iki değişken kullanılarak uygulanır ve analizde Pearson korelasyon katsayısı olarak adlandırılan bir ölçü kullanır. Pearson korelasyon katsayısı, değişkenler arasındaki doğrusal ilişkinin gücünü ve yönünü ölçer. Değerler -1 ile 1 arasında olabilir:

- 1'e yaklaşan bir korelasyon katsayısı, negatif bir ilişkiyi gösterirken,
- 1'e yaklaşan bir korelasyon katsayısı, pozitif bir ilişkiyi gösterir.
- 0'a yakınsa, değişkenler arasında bir ilişki olmadığı anlaşılır.

Pearson korelasyon analizi, özellikle iki değişken arasındaki ilişkinin doğrusal olduğu ve değişkenlerin normal dağılıma sahip olduğu durumlarda kullanılır. Ancak, değişkenler arasında doğrusal olmayan bir ilişki varsa, bu analizin sonuçları güvenilir olmayabilir. Pearson korelasyon analizi, genellikle korelasyonun anlamlılığını test etmek için de kullanılır.



**Çeyrekler:** Veri setinin alt ve üst çeyrekleridir. Alt çeyrek, veri setinin alt %25'lik kısmını ifade ederken, üst çeyrek, veri setinin üst %25'lik kısmını ifade eder.

**Çeyrekler Arası Aralık:** İlk ve üçüncü çeyrek arasında yer alan değerler bu aralığa girmektedir. Veri setinin ortasındaki yüzde 50'lik aralığı ifade eder. Aykırı değerlerin tespitinde kullanılabilir.

**Alt Sınır:** Aykırı değerlerin tespitinde kullanılan sınırlardır. Bu sınırların dışında kalan değerler genellikle aykırı olarak kabul edilir.

**Aykırı Değerler:** Box Plot grafiğinde dışarıda bulunan noktalar, alt ve üst sınırların dışında kalan değerleri temsil eder. Bu noktalar genellikle aykırı olarak kabul edilir.

### 3.2.8. Ki-Kare Analizi

Ki-kare analizi, iki veya daha fazla kategorik değişkenin birbirinden bağımsız olup olmadığının araştırılması için kullanılan istatistik yöntemidir. Bu test bağımsızlığı ölçmekle birlikte değişkenler arasında ilişki varsa bu ilişki hakkında yeterli bilgi sağlamaz. Ki-kare testi, örneklem büyüklüğüne karşı duyarlı bir yöntemdir. Gözlem sayısı arttıkça ki-kare değeri ve dolayısıyla anlamlı fark gözlemeleme oranı artmaktadır.

Testin yapılabilmesi için en önemli ön koşul verilerin kategorik olmasıdır. Çünkü ki-kare testi frekanslar üzerinden çalışır ve çapraz tablolar oluşturur. Ayrıca gruplar birbirinden bağımsız olmalı, bir gözlem birden fazla grup/kategori/değişken altında yer almamalıdır.

### 3.3. Veri Seti

Bu çalışmada SEERA (Software Engineering in Sudan) [17] efor tahmini veri seti kullanılmıştır. SEERA veri seti, Sudan'daki 42 kuruluştan 120 yazılım geliştirme projesi verisinin toplanması sonucu elde edilmiştir. SEERA veri seti, 76 öznitelik içerir ve mevcut efor tahmini veri kümelerinin aksine, kodlama öncesinde ham verileri içeren meta veriler ve alt niteliklerle zenginleştirilmiştir. Bu veri seti, gelişmekte olan ülkelere ve kısıtlı ortamlardaki yazılım endüstrilerine daha uygun veriler sağlama konusunda mevcut boşluğu doldurmaktadır. Ayrıca, araştırmacıların çevresel ve sosyoekonomik faktörlerin teknik maliyet faktörleri üzerindeki etkisini araştırmak ve daha uygun yeni teknikler geliştirmesine olanak sağlayacak güncel ve çeşitli veriler sağlamaktadır [18] .

Veri setinde yer alan 76 öznitelik aşağıdaki tabloda belirtilmiştir. Çalışmada “Gerçek Efor” değişkeni bağımlı değişken olarak ele alınmış olup, diğer tüm öznitelikler bağımsız değişken olarak kullanılmıştır. Çalışmada, Tablo 3.1’de belirtilen bağımsız değişkenler ile bağımlı değişken arasındaki korelasyon analizinin yapılması, en iyi uyuma sahip olacak ve en az değişken kullanılacak tanımlamayı yapabilen bir sınıflandırma modeli kurmak hedeflenmiştir.

Veri seti oluşturulurken “Gerçek Efor” değişkeni hesaplanması, aşağıdaki formül ile yapılmıştır. Bir projenin gerçekleşen eforu hesaplanırken; projede görev alan ekip sayısı, proje gerçekleşme süresi ve aylık çalışma saati verileri kullanılmıştır.

$$\text{“} = ((\text{Gerçek Süre} * \text{Özel Ekip Üyeleri}) + ((\text{Ekip Boyutu} - \text{Özel Ekip Üyeleri}) * (50/100) * \text{Gerçek Süre})) * \text{Aylık Çalışma Saati ” [17]}$$

Tablo 3.1: Veri Seti Öznitelikleri

Öznitelik Grupları ve Tanımları							
Genel Bilgiler	Boyut	Efor	Çevre	Kullanıcılar	Geliştiriciler	Proje	Ürün
Proje Numarası	Nesne Sayısı	Tahmini Efor	Sözleşme Olgunluğu	Üst Yönetimin Desteği	Proje Yöneticisi Deneyimi	Takvim Kalitesi	Gereksinim Doğruluk Düzeyi
Proje Yılı	Boyutlandırma Yöntemi (Diğer)	<b>Gerçek Efor</b>	Devlet Politikası Etkisi	Önceki Sisteme İlişkin Üst Yönetimin Görüşü	Danışman Kullanılabilirliği	Geliştirme Ortamı Yeterliliği	Teknik Dokümantasyon
Organizasyon Numarası	Tahmini Boyut		Ekonomik İstikrarsızlık Etkisi	Sistem Anlaşılabilirliği	VTYS Uzmanı Kullanılabilirliği	Araç Kullanılabilirliği	Kod Açıklamaları
Organizasyon Tipi			Organizasyon Yönetim Yapısının Anlaşılabilirliği	Kullanıcı Direnci	Emsalsizlik	Yazılım Metodolojisi	Kullanım Kılavuzu
Organizasyonel Rol			Geliştirici İşe Alım Politikası	Kullanıcı Bilgisayar Deneyimi	Yazılım Aracı Deneyimi	# Çoklu Programlama Dilleri	Tekrar Kullanılabilirlik Gerekliliği
Organizasyon Boyutu			Geliştirici Teşvik Politikası	Kullanıcı Kararlılığı	Programlama Dili Deneyimi	Kullanılan Programlama Dili	Performans Gereksinimleri
BT Bölümü Boyutu			Geliştirici Eğitimi	Gereksinim Kararlılığı	Programcı Yeteneği	Kullanılan VTYS Dili	Ürün Karmaşıklığı

Müşteri Organizasyon Tipi			Geliştirme Ekibi Yönetimi	Gereksinim Esnekliği	Analist Yeteneği	Teknik Dayanıklılık	Güvenlik Gereksinimleri
Tahmini Süre					Ekip Seçimi	Açık Kaynak Yazılımı Kullanımı	Güvenilirlik Gereksinimleri
Gerçek Süre					Ekip Boyutu	Dış Kaynak Kullanım Düzeyi	Donanım Gerekliliği
% Proje Kazancı (Zararı)					Özel Ekip Üyeleri	Dış Kaynak Etkisi	
Geliştirme Türü					Günlük Çalışma Saati	Yazılımın Yeniden Kullanım Derecesi	
Uygulama Alanı					Ekip Sözleşmeleri	Risk Yönetimi Düzeyi	
					Ekip Sürekliliği	Standartların Kullanımı	
					Ekip Uyumu	Standartların Kullanım Düzeyi	
					Gelir Memnuniyeti	Yeniden Yapılandırma Süreci	

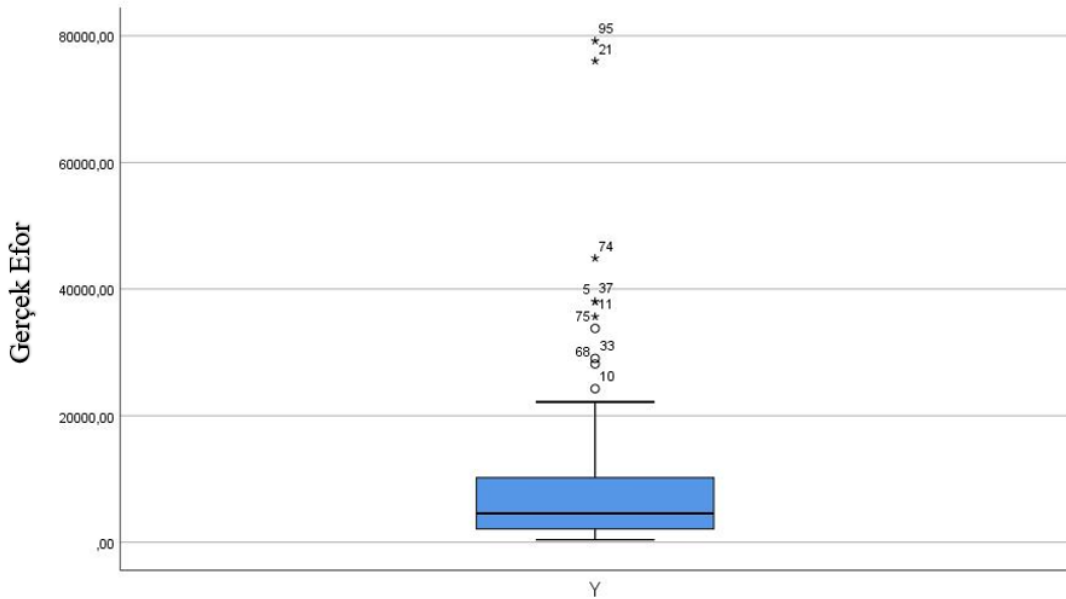
### 3.3.1. Verilerin Dağılımının İncelenmesi

Araştırmanın bu kısmında veri dağılımı incelenmiş ve veride aykırı değerler olduğu tespit edilmiştir. “Aykırı değerler, bir veri setindeki diğer değerlerle karşılaştırıldıklarında belirgin bir şekilde daha büyük veya daha küçük olan değerlerdir. Bu değerler, istatistiksel analizler üzerinde hiçbir etkiye sahip olmayabilecekleri gibi, bazen çok büyük bir etki de yaratabilirler” [23]. Veri tutarlılığını (doğruluğunu) artırmak amacıyla aykırı değerlerin veri setinden çıkarılarak verinin düzenlenmesi araştırma sonucu için önem arz etmektedir.

Tablo 3.2: Verilerin Dağılımı

Ölçümler	Merkezi Eğilim			Basıklık-Çarpıklık	
	Ort.	S.S	Medyan	Basıklık	Çarpıklık
Aykırı Değerler Varken	9060,92	12598,48	4576,00	3,345	14,067
Aykırı Değerler Kaldırıldığında	6002,55	5402,76	4224,00	1,333	0,457

Bağımlı değişkenin dağılımı incelendiğinde verilerde ilk aşamada verilerin dağılımının normal dağılımdan gelmediği tespit edilmiş, dağılımı bozan aykırı değerlerin incelenmesi için box plot grafiğine bakılmış, grafik sonucu 10 adet aykırı değer olduğu görülmüştür (Şekil 3.2).



Şekil 3.2: Aykırı Değerlere Ait Box Plot Grafiği

Şekil 2 incelendiğinde 95, 75, 74, 68, 37, 33, 21, 11, 10 ve 5 nolu projelerden gelen verilerin dağılımı bozduğu tespit edilmiş ilgili projeler örneklem dışı bırakıldığı zaman verilerin normal dağıldığı görülmüştür (Tablo 3.2). Tablo 3.2’deki Normal dağılım analizi sonucunda incelenen merkezi eğilim ölçümlerinden ortalama-medyanın birbirine yakınlığı ve basıklık ile çarpıklığın  $\pm 2$  arasında olması nedeniyle elde edilen verilerin normal dağılımdan geldiği belirlenmiştir [24]. Aynı zamanda araştırmaya dâhil olan proje sayısı yeterli olduğu için ( $n \geq 30$ ) merkezi limit teoreminden hareketle istatistiksel açıdan daha güçlü olan parametrik yöntemlere başvurulmuştur [25].

### 3.3.2. Veri Setinin Hazırlanması

Araştırmanın bu kısmında veri setinde incelenen değişkenleri betimlemek için frekans ve tanımlayıcı bulgulara ait analiz sonuçlarına yer verilmiştir.

Tablo 3.3: Projelerin Organizasyon Tipi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Organizasyon Tipi	Kamu Yazılım Şirketi	4	3,6
	Üniversite	8	7,3
	Federal Bakanlıklar	6	5,5
	Federal Müdürlükler	23	20,9
	Özel Yazılım Şirketi	58	52,7
	Kurumsal BT Departmanı	2	1,8
	Serbest Çalışan	5	4,5
	Telekomünikasyon Şirketi	4	3,6
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.3 incelendiğinde, projelerde en çok %52,7 ( $n=58$ ) ile “Özel Yazılım Şirketi” grubunun görüldüğü belirlenmiştir.

Tablo 3.4: Projelerin Organizasyonel Rol Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Organizasyonel Rol	Geliştirici	29	26,4
	Proje Yöneticisi	71	64,5
	Şirket Müdürü	4	3,6
	Teknik Danışman	5	4,5
	Sistem Yöneticisi	1	0,9
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.4 incelendiğinde, projelerde en çok %64,5 (n=71) ile “Proje Yöneticisi” rolünün görüldüğü belirlenmiştir.

Tablo 3.5: Projelerin Müşteri Organizasyon Tipi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Müşteri Organizasyon Tipi	Üniversite Departmanı	5	4,5
	Özel Şirket Departmanı	10	9,1
	Banka	15	13,6
	Federal Bakanlıklar	11	10,0
	Federal Bakanlık Departmanı	7	6,4
	Fabrika	2	1,8
	Hastane Departmanı	1	0,9
	Televizyon Kanalı	1	0,9
	Kar Amacı Gütmeyen Kuruluş	1	0,9
	Özel Şirket	23	20,9
	Hastane	1	0,9
	Şirket İçi Geliştirme	21	19,1
	Federal Müdürlükler	6	5,5
	Özel Okul	1	0,9
	Banka Departmanı	5	4,5
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.5 incelendiğinde, projelerde en çok %20,9 (n=23) ile “Özel Şirket” türünün görüldüğü belirlenmiştir.

Tablo 3.6: Projelerin Geliştirme Türü Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Geliştirme Türü	Yeni yazılım geliştirme	85	77,3
	Mevcut yazılımın yükseltilmesi	8	7,3
	Mevcut yazılımın değiştirilmesi	9	8,2
	İthal yazılımın özelleştirilmesi	8	7,3
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.6 incelendiğinde, projelerde en çok %77,3 (n=85) ile “Yeni yazılım geliştirme” türünün görüldüğü belirlenmiştir.

Tablo 3.7: Projelerin Uygulama Alanı Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Uygulama Alanı	Banka Sistemleri	13	11,8
	ERP	26	23,6
	Mobil Uygulamalar	6	5,5
	Finansal ve Yönetimsel	18	16,4
	Web Uygulamaları	10	9,1
	Özel Uygulamalar	37	33,6
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.7 incelendiğinde, projelerde en çok %33,6 (n=37) ile “Özel Uygulamalar” domain türünün görüldüğü belirlenmiştir.

Tablo 3.8: Projelerin Devlet Politikası Etkisi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Devlet Politikası Etkisi	Çok Olumlu Etki	11	10,0
	Olumlu Etki	3	2,7
	Etkisiz	40	36,4
	Olumsuz Etki	37	33,6
	Çok Olumsuz Etki	19	17,3
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.8 incelendiğinde, projelerde en çok %36,4 (n=40) ile “Etkisiz” görüldüğü belirlenmiştir.

Tablo 3.9: Projelerin Geliştirici İşe Alım Politikası Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Geliştirici İşe Alım Politikası	İşe alma standartları ve başvuru sahibi değerlendirmeleri mevcuttur ve uygulanmaktadır	50	45,5
	İşe alma standardı yok ancak başvuru sahibi değerlendirmesi uygulanıyor	23	20,9
	İşe alma standartları ve aday değerlendirmeleri mevcut ancak uygulanmıyor – tanıdıkların işe alınması	6	5,5
	Belirli bir politika izlenmiyor	31	28,2
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.9 incelendiğinde, projelerde en çok %45,5 (n=50) ile “İşe alma standartları ve başvuru sahibi değerlendirmeleri mevcuttur ve uygulanmaktadır” politikasının görüldüğü belirlenmiştir.

Tablo 3.10: Projelerin Geliştirici Eğitimi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Geliştirici Eğitimi	Kuruluş, kullanılan periyodik eğitimleri sağlar	23	20,9
	Geliştiriciler ilgili proje için özel olarak eğitilir	24	21,8
	Eğitim desteği verilmez	63	57,3
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.10 incelendiğinde, projelerde en çok %57,3 (n=63) ile “Eğitim desteği verilmez” görüldüğü belirlenmiştir.

Tablo 3.11: Projelerin Önceki Sisteme İlişkin Üst Yönetimin Görüşü Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Önceki Sisteme İlişkin Üst Yönetimin Görüşü	Hayır	42	38,2
	Evet	68	61,8
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.11 incelendiğinde, projelerde en çok %61,8 (n=68) ile “Evet” görüşünün görüldüğü belirlenmiştir.

Tablo 3.12: Projelerin Kullanıcı Bilgisayar Deneyimi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Kullanıcı Bilgisayar Deneyimi	Müşteri organizasyonunda önceki bilgisayar sisteminin mevcut olması (kullanıcıların bilgisayarlarla çalışma deneyiminin olması)	54	49,1
	Temel bilgilere aşına	49	44,5
	Bilgisayar deneyimi yok	7	6,4
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.12 incelendiğinde, projelerde en çok %49,1 (n=54) ile “Müşteri organizasyonunda önceki bilgisayar sisteminin mevcut olması (kullanıcıların bilgisayarlarla çalışma deneyiminin olması)” tecrübe seviyesinin görüldüğü belirlenmiştir.

Tablo 3.13: Projelerin Proje Yöneticisi Deneyimi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Proje Yöneticisi Deneyimi	Benzer yazılım sistemlerinde geçmiş deneyime sahip olmak	75	68,2
	Benzer olmayan yazılım sistemlerinde geçmiş deneyime sahip olmak	25	22,7
	Deneyime sahip olmamak	10	9,1
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.13 incelendiğinde, projelerde en çok %68,2 (n=75) ile “Benzer yazılım sistemlerinde geçmiş deneyime sahip olmak” tecrübe seviyesinin görüldüğü belirlenmiştir.

Tablo 3.14: Projelerin Danışman Kullanılabilirliği Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Danışman Kullanılabilirliği	Teknik ve proje yönetimi konularında danışman tavsiyesi alındı	14	12,7
	Teknik konularında danışman tavsiyesi alındı	29	26,4
	Danışman tavsiyesi alınmadı	67	60,9
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.14 incelendiğinde, projelerde en çok %60,9 (n=67) ile “Danışman tavsiyesi alınmadı” uygunluk seviyesinin görüldüğü belirlenmiştir.

Tablo 3.15: Projelerin VTYS Uzmanı Kullanılabilirliği Özelliklerine Ait Bulgular

Değişken	Grup	n	%
VTYS Uzmanı Kullanılabilirliği	Hayır	34	30,9
	Evet	76	69,1
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.15 incelendiğinde, projelerde en çok %69,1 (n=76) ile “Evet” uygunluk seviyesinin görüldüğü belirlenmiştir.

Tablo 3.16: Projelerin Yazılım Aracı Deneyimi Özelliklerine Ait Bulgular

<b>Değişken</b>	<b>Grup</b>	<b>n</b>	<b>%</b>
Yazılım Aracı Deneyimi	4 yıl üstü	46	41,8
	2 - 3 yıl	29	26,4
	1- 2 yıl	19	17,3
	Tecrübesiz	16	14,5
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.16 incelendiğinde, projelerde en çok %41,8 (n=46) ile “4 yıl üstü” tecrübe seviyesinin görüldüğü belirlenmiştir.

Tablo 3.17: Projelerin Ekip Seçimi Özelliklerine Ait Bulgular

<b>Değişken</b>	<b>Grup</b>	<b>n</b>	<b>%</b>
Ekip Seçimi	Uygulama türündeki deneyime dayalı	6	5,5
	İş uzmanlığına dayalı olarak (analistler, tasarımcılar, vb.)	27	24,5
	Mevcut geliştiricilere dayalı olarak	77	70,0
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.17 incelendiğinde, projelerde en çok %70,0 (n=77) ile “Mevcut geliştiricilere dayalı olarak” seçiminin görüldüğü belirlenmiştir.

Tablo 3.18: Projelerin Gelir Memnuniyeti Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Gelir Memnuniyeti	Çok Memnun	11	10,0
	Memnun	42	38,2
	Normal	31	28,2
	Memnun değil	26	23,6
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.18 incelendiğinde, projelerde en çok %38,2 (n=42) ile “Memnun” memnuniyet seviyesinin görüldüğü belirlenmiştir.

Tablo 3.19: Projelerin Takvim Kalitesi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Takvim Kalitesi	Takvim oluşturuldu ve periyodik değerlendirmeye takip edildi	56	50,9
	Takvim oluşturuldu ve periyodik değerlendirmeye takip edilmedi	28	25,5
	Takvim oluşturuldu, fakat uyulmadı	26	23,6
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.19 incelendiğinde, projelerde en çok %50,9 (n=56) ile “Takvim oluşturuldu ve periyodik değerlendirmeye takip edildi” kalite seviyesinin görüldüğü belirlenmiştir.

Tablo 3.20: Projelerin Yazılım Metodolojisi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Yazılım Metodolojisi	Şelale	38	34,5
	Çevik	24	21,8
	Hibrit Metodoloji	38	34,5
	Prototip	10	9,1
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.20 incelendiğinde, projelerde en çok %34,5 (n=38) ile “Şelale” ve “Hibrit Metodoloji” metodolojisinin görüldüğü belirlenmiştir.

Tablo 3.21: Projelerin Kullanılan Programlama Dili Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Kullanılan Programlama Dili	PHP	14	12,7
	Android	2	1,8
	ASP.net	4	3,6
	C#	4	3,6
	Java, Android	1	0,9
	Java	10	9,1
	Oracle Developer	35	31,8
	Visual Basic 6	13	11,8
	Python/Java	6	5,5
	Python	10	9,1
	Visual Basic .Net	3	2,7
	C++	1	0,9
	Python/HTML/PHP	1	0,9
	Javascript/PHP	1	0,9
	ASP.NET/ JavaScript/C#	1	0,9
	CH	2	1,8
	Java , C#, PHP	1	0,9
	Java/Android/C#	1	0,9
<b>Toplam</b>	<b>110</b>	<b>100,0</b>	

Tablo 3.21 incelendiğinde, projelerde en çok %31,8 (n=35) ile “Oracle Developer” dilinin kullanıldığı belirlenmiştir.

Tablo 3.22: Projelerin Kullanılan VTYS Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Kullanılan VTYS	MySQL	25	22,7
	Oracle	48	43,6
	Microsoft Server	21	19,1
	PostgreSQL	16	14,5
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.22 incelendiğinde, projelerde en çok %43,6 (n=48) ile “Oracle” kullanıldığı belirlenmiştir.

Tablo 3.23: Projelerin Açık Kaynak Yazılımı Kullanımı Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Açık Kaynak Yazılımı Kullanımı	Hayır	76	69,1
	Evet	34	30,9
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.23 incelendiğinde, projelerde en çok %69,1 (n=76) ile “Hayır” görüşünün olduğu belirlenmiştir.

Tablo 3.24: Projelerin Dış Kaynak Kullanım Düzeyi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Dış Kaynak Kullanım Düzeyi	Geliştirme ekibiyle aynı deneyim ve yeteneklere sahip bir kuruluşa dış kaynak sağlanması	11	10,0
	Dış kaynak kullanılmadı	99	90,0
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.24 incelendiğinde, projelerde en çok %90,0 (n=99) ile “Dış kaynak kullanılmadı” olduğu belirlenmiştir.

Tablo 3.25: Projelerin Yazılımın Yeniden Kullanım Derecesi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Yazılımın Yeniden Kullanım Derecesi	Önceki yazılım sistemindeki modülleri yeniden kullanın/satın alın	20	18,2
	Önceki bir yazılım sisteminin tasarımını yeniden kullanma	26	23,6
	Önceki yazılım sisteminin teknik özelliklerini yeniden kullanın	24	21,8
	Yeniden kullanım yok	40	36,4
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.25 incelendiğinde, projelerde en çok %36,4 (n=40) ile “Yeniden kullanım yok” derecesi olduğu belirlenmiştir.

Tablo 3.26: Projelerin Standartların Kullanımı Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Standartların Kullanımı	Yazılım geliştirme yaşam döngüsünün belirli aşamaları için standartların kullanılması	20	18,2
	Herhangi bir standart kullanılmaması	90	81,8
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.26 incelendiğinde, projelerde en çok %81,8 (n=90) ile “Herhangi bir standart kullanılmaması” standart seviyesinin kullanıldığı belirlenmiştir.

Tablo 3.27: Projelerin Gereksinim Doğruluk Düzeyi Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Gereksinim Doğruluk Düzeyi	Yazılım sistemini geliştirmek için kullanılan doğru gereksinim spesifikasyonları	69	62,7
	Gereksinim spesifikasyonlarının hatalı olması ve yazılım gereksinimlerinin yeniden analiz edilmesini gerektirmesi	10	9,1
	Gereksinim spesifikasyonlarının hatalı olması ve yazılım sisteminin yeniden tasarlanmasını gerektirmesi	9	8,2
	Hatalı gereksinim spesifikasyonları ve yazılım sisteminin yeniden programlanmasının gerekli olması	22	20,0
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.27 incelendiğinde, projelerde en çok %62,7 (n=69) ile “Yazılım sistemini geliştirmek için kullanılan doğru gereksinim spesifikasyonları” seviyesinde olduğu belirlenmiştir.

Tablo 3.28: Projelerin Teknik Dokümantasyon Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Teknik Dokümantasyon	Dokümantasyon yok	28	25,5
	Geliştirme yaşam döngüsünün büyük bir kısmı kapsamıyor	30	27,3
	Geliştirme yaşam döngüsünün minimum kısımları kapsamıyor	28	25,5
	Tüm aşamalar belgelendi	24	21,8
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.28 incelendiğinde, projelerde en çok %27,3 (n=30) ile “Geliştirme yaşam döngüsünün büyük bir kısmı kapsamıyor” dokümantasyon olduğu belirlenmiştir.

Tablo 3.29: Projelerin Kod Açıklamaları Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Kod Açıklamaları	Açıklama yok	24	21,8
	Programcıya kolaylık sağlayacak açıklamalar var	30	27,3
	Diğer programcıların anlaması için bazı modüllerde açıklamalar var	48	43,6
	Kod içinde ayrıntılı açıklamalar var	8	7,3
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.29 incelendiğinde, projelerde en çok %43,6 (n=48) ile “Diğer programcıların anlaması için bazı modüllerde açıklamalar var” açıklamalar olduğu belirlenmiştir.

Tablo 3.30: Projelerin Kullanım Kılavuzu Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Kullanım Kılavuzu	Kullanım Kılavuzu yok	33	30,0
	Kullanım kılavuzu tüm yazılım sistemini kapsamaz	19	17,3
	Teknik terminolojiyle yazılmış, anlaşılır olmayan kullanım kılavuzu	58	52,7
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.30 incelendiğinde, projelerde en çok %52,7 (n=58) ile “Teknik terminolojiyle yazılmış, anlaşılır olmayan kullanım kılavuzu” olduğu belirlenmiştir.

Tablo 3.31: Projelerin Tekrar Kullanılabilirlik Gerekliliği Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Tekrar Kullanılabilirlik Gerekliliği	Yeniden kullanıma gerek yok	26	23,6
	Bazı modüller yeniden kullanılır	38	34,5
	Başka bir yazılım sistemi geliştirmek için tüm yazılım sisteminin yeniden kullanılması	9	8,2
	Diğer müşterilere satılacak yazılım sisteminin özelleştirilmesi	37	33,6
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.31 incelendiğinde, projelerde en çok %34,5 (n=38) ile “Bazı modüller yeniden kullanılır” olduğu belirlenmiştir.

Tablo 3.32: Projelerin Ürün Karmaşıklığı Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Ürün Karmaşıklığı	Açık ve basit	32	29,1
	Açık ve biraz karmaşık	51	46,4
	Karmaşık	14	12,7
	Çok karmaşık algoritmalar içerir ve anlaşılması zordur	13	11,8
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.32 incelendiğinde, projelerde en çok %46,4 (n=51) ile “Açık ve biraz karmaşık” olduğu belirlenmiştir.

Tablo 3.33: Projelerin Güvenilirlik Gereksinimleri Özelliklerine Ait Bulgular

Değişken	Grup	n	%
Güvenilirlik Gereksinimleri	Kullanıcı memnuniyetsizliği ve rahatsızlığı	27	24,5
	Küçük maddi kayıplar hafifletilebilir	34	30,9
	Orta düzeyde parasal kayıp hafifletilebilir	17	15,5
	Büyük maddi kayıp	32	29,1
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.33 incelendiğinde, projelerde en çok %30,9 (n=34) ile “Küçük maddi kayıplar hafifletilebilir” olduğu belirlenmiştir.

Tablo 3.34: Projelerin Donanım Gerekliliği Özelliklerine Ait Bulgular

<b>Değişken</b>	<b>Grup</b>	<b>n</b>	<b>%</b>
Donanım Gerekliliği	Gerekli değil	61	55,5
	Zamanında temin edilebilen özel donanım gerekli ve donanım konusunda önceden deneyimimiz var	40	36,4
	Zamanında temin edilebilen özel donanım gerekli ancak donanım ile ilgili önceden deneyimimiz yok	9	8,2
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 3.34 incelendiğinde, projelerde en çok %55,5 (n=61) ile “Gerekli değil” olduğu belirlenmiştir.

## 4. UYGULAMA VE BULGULAR

Çalışmada yazılım projelerindeki efor tahmin etmek amacıyla sınıflandırma algoritmaları kullanılarak bir model oluşturmak ve bu model ile proje başlangıcında projenin yüksek veya düşük efor ile sonuçlanacağını tahmin etmek hedeflenmiştir. Bağımlı değişkenin kategorik olması nedeniyle lojistik regresyon yöntemi ile iki farklı sınıflandırma modeli oluşturulmuştur. Bağımlı değişken olan “Gerçek Efor” değişkeni sürekli bir değişkendir, bu değişkenin çalışmada farklı tiplerde (sürekli, kategorik) ele alınması sağlanarak sınıflandırma modelinde değişken tipinin etkilerinin incelenmesi hedeflenmiştir.

Kurulan ilk modelde (Model-1) sürekli bağımlı değişken ile bağımsız değişkenlere göre farklılıkların incelenmesi için ikili grupların karşılaştırılmasında Bağımsız Örneklem T-testi, ikiden fazla gruplar arasındaki farkın incelenmesi için Tek Yönlü Anova Analizi kullanılmış, Pearson Korelasyon Analizi ile sürekli değişkenler analiz edilmiştir. Bağımlı değişkenin kategorik iki grup haline getirilmesi sonucu bağımsız değişkenlerin bağımlı değişken üzerindeki etkisinin analizi ve sınıflandırılma modelinin kurulması için lojistik regresyon algoritması kullanılmıştır.

İkinci model (Model-2) kurulurken öncelikle bağımlı değişken kategorik hale getirilmiştir. Bağımlı değişkenin kategorik hali ile bağımsız değişkenler arasındaki ilişkilerin incelenmesinde Bağımsız Örneklem T-testi ve Ki-Kare Analizi uygulanmıştır. Yapılan analizler neticesinde ilişkileri anlamlı olan bağımsız değişkenler lojistik regresyon modeline dâhil edilerek alternatif sınıflandırma modeli geliştirilmiştir.

### 4.1. Sürekli Bağımlı Değişken ile Sınıflandırma Modeli (Model-1)

Çalışma kapsamındaki ilk modelde (Model-1) bağımlı değişken olan “Gerçek Efor” değişkeni sürekli olarak ele alınmış olup, bağımsız değişkenlerle farklılık analizleri yapılmıştır. Bu analizlerde “Gerçek Efor” bağımlı değişkeni ile farklılaşan bağımsız değişkenler sınıflandırma modeline dahil edilmek üzere alınmış, farklılığı anlamlı olmayan diğer değişkenlerin ise çalışma kapsamından çıkarılmıştır. Bağımlı değişken kategorik hale getirilmiş ve lojistik regresyon yöntemi kullanılarak belirlenen bağımsız değişkenler ile Model-1 oluşturulmuştur. Oluşturulan modele ilişkin uyum iyiliği istatistikleri ve modelin sınıflandırma başarısı hesaplanarak araştırma sonuçları analiz edilmiştir.

#### 4.1.1. Sürekli bağımsız değişkenlere ait hesaplamalar

Araştırmanın bu kısmında veri setindeki sürekli bağımsız değişkenler ele alınmış ve değişkenlere ilişkin ortalama ve standart sapma değerleri Tablo 4.1’deki gibi hesaplanmıştır.

Tablo 4.1: Araştırmaya Ait Betimsel Bulgular

Değişkenler	Ort.	S.s.	Min.	Max.
Organizasyon Boyutu	8,08	6,32	1,00	17,00
BT Bölümü Boyutu	4,34	4,21	1,00	12,00
Tahmini Süre	5,40	3,64	0,25	20,00
Gerçek Süre	9,40	8,02	0,50	48,00
Nesne Sayısı	167,15	180,37	7,00	750,00
Gerçek Efor	6002,55	5402,77	396,00	22176,00
Kullanıcı Kararlılığı	3,09	1,36	1,89	5,00
Gereksinim Kararlılığı	2,59	1,10	1,00	4,81
Gereksinim Esnekliği	2,52	1,36	1,00	5,00
Ekip Boyutu	4,67	3,11	1,00	25,00
Özel Ekip Üyeleri	3,17	2,91	0,00	20,00
Günlük Çalışma Saati	8,28	1,67	4,00	14,00
Ekip Sürekliliği	10,38	19,32	0,00	100,00
Yeniden Yapılandırma Süreci	2,13	1,19	1,00	5,00

Bulgular incelendiğinde “Organizasyon Boyutu” değerinin ortalamasının  $8,08\pm 6,32$  olduğu, “BT Bölümü Boyutu” değerinin ortalamasının  $4,34\pm 4,21$  olduğu, “Tahmini Süre” değerinin ortalamasının  $5,40\pm 3,64$  olduğu, “Gerçek Süre” değerinin ortalamasının  $9,40\pm 8,02$  olduğu, “Nesne Sayısı” değerinin ortalamasının  $167,15\pm 180,37$  olduğu, “Gerçek Efor” değerinin ortalamasının  $6002,55\pm 5402,77$  olduğu, “Kullanıcı Kararlılığı” değerinin ortalamasının  $3,09\pm 1,36$  olduğu, “Gereksinim Kararlılığı” değerinin ortalamasının  $2,59\pm 1,10$  olduğu, “Gereksinim Esnekliği” değerinin ortalamasının  $2,52\pm 1,36$  olduğu, “Ekip Boyutu” değerinin ortalamasının  $4,67\pm 3,11$  olduğu, “Özel Ekip Üyeleri” değerinin ortalamasının  $3,17\pm 2,91$  olduğu, “Günlük Çalışma Saati” değerinin ortalamasının  $8,28\pm 1,67$  olduğu, “Ekip Sürekliliği” değerinin ortalamasının  $10,38\pm 19,32$  olduğu, “Yeniden Yapılandırma Süreci” değerinin ortalamasının  $2,13\pm 1,19$  olduğu belirlenmiştir.

#### 4.1.2. Sürekli bağımlı değişken ile bağımsız değişkenlerin karşılaştırılması

Araştırmanın bu kısmında araştırmada kullanılan sürekli bağımlı değişkenin kategorik bağımsız değişkenler ile farklılıklarının incelenmesi için ikili grupların karşılaştırılmasında Bağımsız Örneklem T-testi, ikiden fazla grupların karşılaştırılmasında Tek Yönlü Anova Analizi yapılmıştır. Sürekli olan bağımsız değişkenler ile sürekli bağımlı değişken arasındaki ilişkinin incelenmesi için Pearson Korelasyon Analizi yapılmıştır.

Tablo 4.2: Kategorik Değişkenler ile Sürekli Bağımlı Değişken Arasındaki Farklılık Analizi Sonuçları

Değişkenler	Gerçek Efor	
	F	P
Organizasyon Tipi	1,211	0,304
Müşteri Organizasyon Tipi	1,856	0,041*
Geliştirme Türü	1,954	0,125
Uygulama Alanı	5,097	0,001**
Devlet Politikası Etkisi	1,329	0,264
Geliştirici İşe Alım Politikası	1,705	0,170
Geliştirici Eğitimi	1,083	0,342
Kullanıcı Bilgisayar Deneyimi	1,115	0,332
Proje Yöneticisi Deneyimi	0,457	0,634
Danışman Kullanılabilirliği	0,282	0,755
Yazılım Aracı Deneyimi	0,437	0,727
Ekip Seçimi	2,252	0,110
Gelir Memnuniyeti	0,863	0,463
Takvim Kalitesi	0,135	0,874
Yazılım Metodolojisi	1,850	0,143
Kullanılan Programlama Dili	1,294	0,214
Kullanılan VTYS	0,204	0,893
Dış Kaynak Etkisi	0,088	0,917
Yazılımın Yeniden Kullanım Derecesi	0,487	0,692
Gereksinim Doğruluk Düzeyi	2,388	0,073
Teknik Dokümantasyon	1,125	0,342
Kod Açıklamaları	1,632	0,186
Kullanım Kılavuzu	2,540	0,084

Tekrar Kullanılabilirlik Gerekliliği	2,537	0,061
Ürün Karmaşıklığı	0,295	0,829
Güvenilirlik Gereksinimleri	0,504	0,681
	t	P
Önceki Sisteme İlişkin Üst Yönetimin Görüşü	2,499	0,014*
VTYS Uzmanı Kullanılabilirliği	-	0,981
Açık Kaynak Yazılımı Kullanımı	1,496	0,138
Dış Kaynak Kullanım Düzeyi	1,878	0,063
Standartların Kullanımı	-	0,957
Donanım Gerekliliği	0,414	0,665

t:Bağımsız örneklem t-testi; F: Tek yönlü Anova Testi; \*p<0.05; \*\*p<0.01

Tablo 4.2 incelendiğinde; kategorik değişkenlerden “Müşteri Organizasyon Tipi”in (F:1.856; P=0.041<0.05), “Uygulama Alanı” (F:5.097; P=0.001<0.01), “Önceki Sisteme İlişkin Üst Yönetimin Görüşü” (t:2.499; P=0.014<0.05) değişkenlerinin bağımlı değişken ile farklılaştığı diğer değişkenlerin ise farklılığının anlamlı olmadığı belirlenmiştir. Bu durumda ileride kurulacak Lojistik Regresyon modeline bağımsız değişkenler olarak kategorik değişkenler içerisinde sadece “Uygulama Alanı”, “Önceki Sisteme İlişkin Üst Yönetimin Görüşü” ve “Müşteri Organizasyon Tipi” değişkenleri eklenmesine karar verilmiştir.

Tablo 4.3: Sürekli Değişkenler ile Bağımlı Değişken Arasındaki İlişkiye Ait Sonuçlar

Değişkenler	Gerçek Efor	
	r	P
Organizasyon Boyutu	0,058	0,545
BT Bölümü Boyutu	0,093	0,333
Tahmini Süre	0,515	0,001**
Gerçek Süre	0,545	0,001**
Nesne Sayısı	0,287	0,002**
Kullanıcı Direnci	0,168	0,079
Kullanıcı Kararlılığı	0,232	0,015*
Gereksinim Kararlılığı	0,311	0,001**
Gereksinim Esnekliği	0,050	0,601
Ekip Boyutu	0,557	0,001**
Özel Ekip Üyeleri	0,531	0,001**
Günlük Çalışma Saati	0,070	0,466
Ekip Sürekliliği	0,003	0,974
Yeniden Yapılandırma Süreci	0,100	0,299

\*p<0.05; \*\*p<0.01; r: Pearson korelasyon analizi

Tablo 4.3 kapsamında; sürekli değişkenlerden “Tahmini Süre” (p:0.001<0.01), “Gerçek Süre” (p:0.001<0.01), “Nesne Sayısı” (p:0.002<0.01), “Gereksinim Kararlılığı” (p:0.001<0.01), “Ekip Boyutu” (p:0.001<0.01), “Özel Ekip Üyeleri” (p:0.001<0.01) ve “Kullanıcı Kararlılığı” (p=0.015<0.05) ile “Gerçek Efor” arasındaki ilişkinin anlamlı olduğu belirlenmiştir. Bu sonuç ileride eklenecek Log. Regresyon modeli için sürekli değişkenlerden bağımsız değişkenler olarak yukarıda anlamlı çıkan değişkenlerin alınacağını göstermektedir.

#### 4.1.3. Sürekli bağımlı değişkenin kategorik hale getirilmesi

Araştırmanın bu kısmında araştırmada kullanılan bağımlı değişkenin kategorik hale getirilerek ilk sınıflandırma modelinin oluşturulması amaçlanmıştır. Bağımlı değişkenin kategorik hale getirilirken ortalama ve standart sapma değerlerinden faydalanılmıştır. İlk aşamada bağımlı değişkenin ortalamasından 1 standart sapma altında olan değerler için düşük efor, 1 standart sapma üzerinde olan değerler için yüksek efor, arasında kalan değerler için orta düzey efor olarak gruplandırılmış ancak bu durumda düşük grupta kalan veri sayısı sadece 4 olduğundan Sıralı Lojistik Regresyon sonucu paralellik varsayımını karşılanmayacağı için bağımlı değişken 2 gruba çevrilmesine karar verilmiştir. Bu durumda ortalama (6002.5) altında kalanlar düşük efor (bekleneni karşılamayan) ortalama üzerinde olanlar ise yüksek efor (bekleneni karşılayan) şeklinde sınıflandırılmış olup oluşturulan grup sonucu Tablo 4.4’de verilmiştir.

Tablo 4.4: Bağımlı Değişkenin Kategorize Edilmiş Sonucu

Değişken	Grup	n	%
Efor	Bekleneni karşılamayan (Düşük)	71	64,5
	Bekleneni karşılayan (Yüksek)	39	35,5
<b>Toplam</b>		<b>110</b>	<b>100,0</b>

Tablo 4.4 incelendiğinde sınıflandırma sonucunda incelenen projelerden %64.5’inin beklenen eforun altında kaldığı %35.5’inin ise beklenen eforu karşıladığı belirlenmiştir.

#### 4.1.4. Sürekli bağımlı değişken ile sınıflandırma modeli bulguları

Araştırmanın sınıflandırma modelinde, proje sonucu gerçekleşen performans düzeyi yüksek ve düşük olma durumu üzerinde, “Müşteri Organizasyon Tipi”, “Uygulama Alanı”, “Önceki Sisteme İlişkin Üst Yönetimin Görüşü”, “Tahmini Süre”, “Gerçek Süre”, “Nesne Sayısı”, “Gereksinim Kararlılığı”, “Ekip Boyutu” ve “Özel Ekip Üyeleri” değişkenlerinin etkisinin belirlenmesi amaçlanmıştır. Araştırmanın modelinin sınanması için sıralı lojistik regresyon analizi yapılması planlanmış ama gruplar arasında paralellik sağlanmadığından bağımlı değişken düşük ve yüksek performans şeklinde iki gruba indirgenerek ikili lojistik regresyon yapılmıştır.

Model-1 kurulmadan önce, lojistik regresyona ilişkin aşağıdaki varsayımlar ve kısıtlar incelenmiştir.

"Birinci varsayım analize giren değişkenlerdeki denek oranları ile ilgilidir. Eğer ele alınan grupta yordayıcı (bağımsız) değişkenlerin sayısına göre çok az sayıda denek varsa, bazı problemler ortaya çıkabilir. Lojistik regresyon, özellikle de kategorik değişkenlerin kombinasyonu sonucu oluşan hücreler (gözenekler) düşünülürse, hiçbir deneğin olmadığı hücre sayısının çok olduğu durumlarda oldukça geniş parametre kestirimleri ve standart hata üretebilir. Böyle bir durum ortaya çıkarsa, araştırmacılara kategorik değişkenlere ilişkin kategori birleştirmelerine gitmeleri, eğer uygunsa sorunlu kategoriye dışarıda bırakmaları ya da analiz açısından çok önemli değil ise, bu değişkeni tamamen çıkartmaları önerilir [26], Araştırmacılar için diğer bir seçenek de, denek sayısının artırılarak boş kalan hücrelerin doldurulmasını sağlamaya yönelik girişimdir. Bu araştırmada hiçbir deneğin olmadığı hücre sayısının olmamasından dolayı kategori birleştirme yapılmasına gerek duyulmamış, model için ele alınan bağımsız değişkenlerin birinci varsayıma uyduğu belirlenmiştir."

"İkinci varsayım, lojistik regresyon model-veri uyumunu değerlendirmenin bir aracı olarak uyum iyiliği testlerine dayalıdır. Uyum iyiliği testleri kategorik değişkenlerin kombinasyonu ile oluşan veri matrisinde, her bir hücre için beklenen frekansların değerlerini içerir. Eğer hücrelerin herhangi birinde beklenen frekans çok küçük ise (frekans değerinin 5'in altında olması) ve bu oran gözenek sayısının %20'sini geçiyorsa analizin gücü çok düşük olur. Eğer böyle bir durumla karşılaşırsa araştırmacı yapacağı analizin gücünün düşük düzeyde olacağını kabul etmeli veya ikiden fazla kategorisi bulunan değişkenlerde birleştirmelere gitmeli ya da toplam hücre sayısını azaltmak için bu kategorik değişkeni dışarıda bırakmalıdır [26]. Bu araştırmada hem bağımsız değişkenlerin kendi içerisinde hem de bağımlı ile bağımsız değişken arasındaki hücrelerde beklenen değerlerin 5'in altında ve bu 5'in altında olan değerlerin gözenek sayısının %20'si geçme durumu incelenmiş, 5'in altında olan güven değişkenlerindeki oranın %20'nin üstünde olmadığı belirlendiğinden, grup birleştirmesi yapılmasına gerek duyulmamıştır."

"Üçüncüsü varsayım, çoklu regresyonun tüm türlerinde olduğu gibi, lojistik regresyonun da bağımsız değişkenler arasındaki yüksek korelasyona oldukça duyarlı olmasıdır. Bu durum bağımsız değişkenler arasında çoklu bağlantı (multicollinearity) sorunu ile sonuçlanır. Eğer analize giren değişkenler arasında çoklu bağlantı problemi varsa bu sorunu devre dışı bırakabilmek için, bir ya da daha fazla değişkenin modelden çıkartılması tavsiye edilir [26]." Bağımlı değişkenin kategorize edilmemiş hali ile bağımsız değişkenler arasında yapılan çoklu doğrusal regresyon sonucunda "Nesne Sayısı", "Kullanıcı Kararlılığı", "Gereksinim Kararlılığı" değişkenlerin etkisi anlamsız olduğu için modelden çıkarılmasına karar verilmiş, geri kalan değişkenler için herhangi bir bağlantı problemi olmadığı VIF değerinin 10'un altında ve Tolerans değerinin 0.20'nin üzerinde olduğu belirlenmiştir.

"Dördüncüsü varsayım ise yordayıcı değişkene ilişkin uç değerler ile alakalı olup uç lojistik regresyon uç değerlere de oldukça duyarlı davranmaktadır. Standardize edilmiş hatanın (standardized residuals) uç

değerleri belirlemek üzere dikkatle incelenmelidir. Belirlenen herhangi bir uç değer (>|3|) standart yöntemler (veri girişini kontrol etme, örneklemeden çıkarma gibi) kullanılarak elenmelidir [27].” Bu çalışmada, standartlaştırılmış uç değerlerin -3 ile +3 aralığında olduğu tespit edilmiştir. Dolayısıyla, lojistik regresyon analizinin gerçekleştirilmesine engel teşkil eden bir durum bulunmamaktadır.

Tablo 4.5: Referans Noktasının Belirlenmesi

Gerçekleşen Performans	
Orijinal Değer	İç Değer
Yüksek	0
Düşük	1

Tablo 4.5 incelendiğinde gerçekleşen performansın yüksek olmasının referans noktası olarak belirlenmiştir.

Tablo 4.6: Ki-Kare Değeri Tablosu - Sıfıncı Adıma İlişkin

	Değişkenler	Skor	Sd	p
	Tahmini Süre	24,855	1	0,001**
	Gerçek Süre	20,974	1	0,001**
Adım	Ekip Boyutu	25,019	1	0,001**
0	Özel Ekip Üyeleri	22,646	1	0,001**
	Önceki Sisteme İlişkin Üst Yönetimin Görüşü	17,200	1	0,001**
	X <sup>2</sup>	74,458	5	0,001*
**p<0.01				

Ki-kare değerinin anlamlı olması sonucunda analize devam edilmesine bir engel olmadığı belirlenmiştir. Ancak, değişkenlerden anlamlı olmayanların modele dahil edilmesi, modelde anlamlı bir katkı sağlamayacağı gibi, diğer bağımsız değişkenlerin anlamlılığını da olumsuz etkileyebilir. Bu nedenle, anlamlı olmayan değişkenlerin modele dahil edilmemesi daha uygun olacaktır. [28]. Araştırmada “Müşteri Organizasyon Tipi” ve “Uygulama Alanı” değişkenlerinin p değerleri 0.05’den büyük olduğu için ve modele dahil edilse bile model için faydalı olmayacağı değerlendirilmiştir. Belirtilen bağımsız değişkenlerin modele dahil

edilmesi diğer bağımsız değişkenleri olumsuz etkileyebileceği için ele alınmamış ve modelin bu değişkenler dahil edilmeden kurulmasına karar verilmiştir.

Tablo 4.6’da Model-1 için geri kalan bağımsız değişkenlerin yer aldığı ki-kare değerinin anlamlı olduğu görülmektedir ( $X^2 = 74,458$ ,  $p < 0,01$ ,  $p = 0,001$ ).

Tablo 4.7: Model-1 Katsayılarına İlişkin Omnibus Testi

Adımlar	X <sup>2</sup>	Sd	p
Adım	114,751	5	0,001**
Blok	114,751	5	0,001**
Model	114,751	5	0,001**
**p<0.01			

Tablo 4.7 değerlendirildiğinde, Model-1’in ki-kare değerine ait p değerinin anlamlı olması, bağımlı değişken ile bağımsız değişkenlerin kombinasyonu arasında bir ilişki bulunduğunu göstermektedir. Bu da bağımsız değişkenler modele dahil edildiğinde, modelin yalnızca sabit terimi baz alan başlangıç modelinden daha iyi bir tahmin yapacağını ifade etmektedir.

Tablo 4.8: Model-1 Uyum İstatistiği Değerleri

-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
28,296	0,648	0,890

Uyum istatistiği değerlerinden -2 log likelihood değerlerinin düşük olması beklenirken, 0 ile 1 arasında olan Nagelkerke R Square ve Cox & Snell R Square değerleri, 1 değerine ne kadar yakın olursa, modelin verilere o kadar iyi uyduğu kabul edilir. Tablo 4.8 incelendiğinde Model-1’deki bağımsız değişkenlerin bağımlı değişkeni tahmin etmedeki açıklama oranı, bir diğer ifadeyle Model-1 sonucunda belirlenen bağımsız değişkenlerin yüksek efor performansını tahmin edilmesinin %89’unu açıkladığı belirlenmiştir. (Nagelkerke R Square= 0,890).

Tablo 4.9: Model-1'e Ait Katsayı Tahminleri

Değişkenler	$\beta$	Standart Hata	Wald	sd	P	Exp(B)
Tahmini Süre	0,019	0,14	0,019	1	0,889	1,02
Gerçek Süre	-0,594	0,196	9,181	1	0,002**	0,552
Ekip Boyutu	-1,593	0,554	8,26	1	0,004**	0,203
Özel Ekip Üyeleri	-0,891	0,397	5,027	1	0,025*	0,41
Önceki Sisteme İlişkin Üst Yönetimin Görüşü (Hayır→Evet)	3,479	1,164	8,935	1	0,003**	32,427
Sabit	15,177	4,565	11,054	1	0,001**	3901401

\*\*p<0.01; \*p<0.05

Tablo 4.9 incelendiğinde, bağımsız değişkenlerden sadece “Tahmini Süre” değişkeninin gerçekleşen yüksek performansı sınıflandırma üzerinde etkisi olmadığı ( $p=0.889>0.05$ ) belirlenirken diğer değişkenlerin ise anlamlı olduğu belirlenmiştir ( $p<0.01$ ;  $p<0.05$ ). Anlamlı etkisi olan değişkenler incelendiğinde, “Gerçek Süre” değişkeninin yüksek performans üzerinde 0.594'lük negatif bir etkisi olduğu ( $\beta= -0.594$ ;  $p=0.002<0.01$ ), “Ekip Boyutu” değişkeninin yüksek performans üzerinde 1.593'lük negatif bir etkisi olduğu ( $\beta= -1.593$ ;  $p=0.004<0.01$ ), “Özel Ekip Üyeleri” değişkeninin yüksek performans üzerinde 0.891'lik negatif bir etkisi olduğu ( $\beta= -0.891$ ;  $p=0.025<0.05$ ) belirlenirken “Önceki Sisteme İlişkin Üst Yönetimin Görüşü<sup>(Hayır→Evet)</sup>”nün alınmadığı projelerin alınanlara göre yüksek performanslı sonuçlanma olasılığı 32.427 kat daha fazla olduğu belirlenmiştir ( $\text{Exp}(B)=32.427$ ).

Sonuç olarak önceki sistem hakkında üst yönetimin görüşünün alınmadığı projelerde önceki sistem hakkında üst yönetimin görüşünün alındığı projelere göre yüksek performanslı sonuçlanma olasılığı 32.427 kat daha fazla, projenin gerçekleşme süresi artıkcça yüksek performanslı sonuçlanmasının azaldığı, projedeki ekip boyutu artıkcça yüksek performanslı sonuçlanmasının azaldığı ve projelerdeki özel ekip üye sayısının artırılması projenin yüksek performanslı sonuçlanmasını azalttığı belirlenmiştir. “Gerçek Süre” değişkeni proje başında bilinmeyeceği için efor tahmininde kullanımı anlamlı olmamakla birlikte, bu değişkenin

SEERA (Software Engineering in Sudan) [17] veri setindeki efor denkleminin bir parçasını oluşturması sebebiyle kurulan modelde etkisi yüksek bağımsız değişkenler arasında çıktığı görülmektedir.

Lojistik regresyon yöntemi ile oluşturulan Model-1 sonucu aşağıda verilmiştir.

$$Z=15.177-0.019*\text{Tahmini Süre} - 0,594*\text{Gerçek Süre} - 1,593* \text{Ekip Boyutu} - 0,891*\text{Özel Ekip Üyeleri} + 3.479* \text{Önceki Sisteme İlişkin Üst Yönetimin Görüşü (Hayır)}$$

Araştırma sonucu oluşturulan sınıflandırma modelinin (Model-1) performansının belirlenmesi için model olmadan sınıflandırma tahminlemesine bakılmış ve daha sonra modele bağlı sınıflandırma tahminlemesi yapılarak performansı gruplandırılması konusunda doğru tahmin oranı incelenmiştir. Model-1'in doğru sınıflandırma düzeyleri Tablo 4.10'da verilmiştir.

Tablo 4.10: Model-1'in Sınıflandırma Başarısı

	Gerçek Efor	Tahmin Edilen Efor		
		Yüksek	Düşük	Başarı
Model Öncesi	Yüksek	0	39	0
	Düşük	0	71	100
<b>Başarı Yüzdesi</b>				<b>64,5</b>
	Gerçek Efor	Tahmin Edilen Efor		
		Yüksek	Düşük	Başarı
Model Sonrası	Yüksek	34	5	87,2
	Düşük	2	69	92,7
<b>Başarı Yüzdesi</b>				<b>93,6</b>

Model-1 oluşturulmadan tüm eforları düşük olarak sınıflandırıldığında %64.5'lik doğru sınıflandırma yapılıyorken, Lojistik regresyon ile kurulan Model-1 sonucunda ise; Gerçekte yüksek eforlu projelerin 34'ünü doğru sınıflandırırken 5 projeyi düşük efor grubuna atamış ve %87.2'lik oranla yüksek eforlu gerçekleşen projeleri doğru tahmin etmiştir.

Gerçekte düşük eforlu projelerin 69'unu doğru sınıflandırırken 2 projeyi yüksek eforlu olarak tahmin etmiş ve düşük eforlu olan projelerin %92.7'sini doğru tahmin etmiştir. Genel doğru sınıflandırma yüzdesine bakıldığında Model-1 110 projenin 103'ünü model sayesinde doğru tahmin ederek %93.6'lık başarıya ulaşmıştır. Normalde model olmadan doğru sınıflandırma yüzdesi %64.5 iken model sayesinde %93.6'lık doğru sınıflandırma yapılabilmektedir.

Yeni bir projeye başlandığında, Model-1'deki değişkenler yerine konulduğu zaman yeni projenin %93.6 oranında yüksek performanslı mı yoksa düşük performanslı mı olacağı sınıflandırılabilir.

Örneğin yeni bir projede: "Tahmini Süre" değeri 6, "Gerçek Süre" değeri 8, "Ekip Boyutu" değeri 7 kişi, "Özel Ekip Üyeleri" değeri 7 kişi ve "Önceki Sisteme İlişkin Üst Yönetimin Görüşü" için "Hayır" olduğunu düşünelim. Bu durumda yeni gelen bu projenin yüksek mi yoksa düşük performanslı mı olacağını sınıflandıralım.

$$Z=15.177-0.019*6-0.594* 8-1.593*7-0.891* 7+3.479* 0$$

$$Z=-7.077$$

$$1/1+e^{-(-7.077)}=0.00084<0.5 \text{ olduğundan performans "yüksek" olarak tahmin edildi.}$$

Bu değerler veri setindeki 6 numaralı projeye ait veriler olup veri setindeki gerçek sınıflandırılması da "yüksek" grubunda olduğu belirlenmiştir.

#### **4.2. Kategorik Bağımlı Değişken ile Sınıflandırma Modeli (Model-2)**

Çalışma kapsamındaki ikinci modelde (Model-2) bağımlı değişken olan "Gerçek Efor" değişkeni kategorik olarak ele alınmış olup, bağımsız değişkenlerle arasındaki ilişkilerin incelenmesinde bağımsız örneklem t-testi ve Ki-kare analizi kullanılmıştır. Yapılan analizler neticesinde ilişkileri anlamlı olan değişkenlerle lojistik regresyon yöntemi kullanılarak Model-2 oluşturulmuştur. Oluşturulan modele ilişkin uyum iyiliği istatistikleri ve modelin sınıflandırma başarısı hesaplanmıştır.

Bağımlı değişkenin kategorik hale getirilmesine ilişkin açıklamalar çalışmanın 4.1.3 bölümünde ele alınmıştır.

#### 4.2.1. Bağımlı değişken ile bağımsız değişkenlerin karşılaştırılması

Araştırmanın bu kısmında araştırmada kullanılan bağımlı değişkenin kategorik hali ile bağımsız değişkenler arasındaki ilişkilerin incelenmesinde bağımsız örneklem t-testi ve ki-kare analizi ile yapılmıştır.

Tablo 4.11: Bağımsız Değişkenler ile Kategorik Bağımlı Değişken Arasındaki İlişkinin Anlamlı Olan Değişkenler

<b>Bağımsız Değişkenler</b>	<b>t</b>	<b>P Değeri</b>
Tahmini Süre	-5,615	0,001**
Gerçek Süre	-5,044	0,001**
Nesne Sayısı	-4,441	0,001**
Kullanıcı Kararlılığı	-2,738	0,007**
Gereksinim Kararlılığı	-3,139	0,002**
Ekip Boyutu	-5,639	0,001**
Özel Ekip Üyeleri	-5,291	0,001**
Yeniden Yapılandırma Süreci	-2,082	0,040*
<b>Bağımsız Değişkenler</b>	<b>X<sup>2</sup></b>	<b>P Değeri</b>
Uygulama Alanı	28,924	0,001**
Dış Kaynak Kullanım Düzeyi	4,242	0,039*
Kullanım Kılavuzu	6,545	0,038*

\*\*p<0.01; \*p<0.05; t: Bağımsız örneklem t-testi; X<sup>2</sup>: Ki-kare analizi

Tablo 4.11’de bağımsız değişkenlerden sürekli değişkenlerin bağımlı kategorik değişken ile farklılığının anlamlılığına bağımsız örneklem t-testi ile bakılmış olup kategorik bağımsız değişkenlerin bağımlı kategorik değişken ile ilişkisine ki-kare analizi ile bakılmıştır. Yapılan analizler sonucu anlamlı çıkan ve Tablo 4.11’de yer alan değişkenlerin Model-2’ye bağımsız değişken olarak eklenmesine karar verilmiştir.

#### 4.2.2. Sürekli Bağımlı Değişken İle Sınıflandırma Modeli Bulguları

Tablo 4.12: Referans Noktasının Belirlenmesi

Gerçekleşen Performans	
Orijinal Değer	İç Değer
Yüksek	0
Düşük	1

Tablo 4.12 incelendiğinde gerçekleşen performansın yüksek olmasının referans noktası olarak belirlendiği görülmektedir.

Tablo 4.13: Ki-Kare Değeri Tablosu - Sıfıncı Adıma İlişkin

Değişkenler	Skor	sd	p
Tahmini Süre	24,855	1	0,001**
Gerçek Süre	20,974	1	0,001**
Nesne Sayısı	16,983	1	0,001**
Kullanıcı Kararlılığı	7,139	1	0,008**
Gereksinim Kararlılığı	9,197	1	0,002**
Ekip Boyutu	25,019	1	0,001**
Özel Ekip Üyeleri	22,646	1	0,001**
Yeniden Yapılandırma Süreci	4,244	1	0,039*
Uygulama Alanı	28,924	5	0,001**
Uygulama Alanı	20,823	1	0,001**
Dış Kaynak Kullanım Düzeyi	4,242	1	0,039*
Kullanım Kılavuzu	6,545	2	0,038*
X <sup>2</sup>	75,412	16	0,001**

Tablo 4.13’de Model-2 için geri bağımsız değişkenlerin yer aldığı ki-kare değerinin (residual chi-square) anlamlı olduğu görülmektedir (X<sup>2</sup> =75,412, p<01, p=0.001).

Tablo 4.14: Model-2 Katsayılarına İlişkin Omnibus Testi

Adımlar	X <sup>2</sup>	sd	p
Adım	120,251	16	0,001**
Blok	120,251	16	0,001**
Model	120,251	16	0,001**

\*\*p<0.01

Tablo 4.14 incelendiğinde, Model-2'nin ki-kare değerine ilişkin p değerinin anlamlı olması, bağımlı değişken ile bağımsız değişkenler kombinasyonu arasında ilişkinin olduğunu göstermektedir. Yani bağımsız değişkenler modele dahil edildiğinde, ilk aşamada modelin sadece sabit terimi baz alarak başlangıç modeldeki tahmininden daha iyi bir tahmin yapacağı anlamına gelmektedir.

Tablo 4.15: Model-2 Uyum İstatistiği Değerleri

-2 likelihood	Log Cox & Snell R Square	R Nagelkerke Square	R
22,797	0,665	0,914	

Uyum istatistiği değerlerinden -2 log likelihood değerlerinin düşük olması beklenirken, 0 ile 1 arasında olan Nagelkerke R Square ve Cox & Snell R Square değerleri, 1 değerine ne kadar yakın olursa, modelin verilere o kadar iyi uyduğu kabul edilir. Tablo 4.15 incelendiğinde Model-2'deki bağımsız değişkenlerin bağımlı değişkeni tahmin etmedeki açıklayıcısı yani model sonucunda kullanılan bağımsız değişkenlerin yüksek efor performansını tahmin edilmesinin %91.4'ünü açıkladığı belirlenmiştir. (Nagelkerke R Square= 0,914).

Tablo 4.16: Model-2'ye Ait Katsayı Tahminleri

Değişkenler	$\beta$	Standart Hata	Wald	sd	P	Exp(B)	
Tahmini Süre	0,173	0,184	0,885	1	0,347	1,189	
Gerçek Süre	0,622	0,263	5,595	1	0,018*	1,862	
Nesne Sayısı	0,002	0,009	0,040	1	0,841	1,002	
Kullanıcı Kararlılığı	0,996	0,793	1,578	1	0,209	2,708	
Gereksinim Kararlılığı	1,253	0,943	1,767	1	0,184	3,501	
Ekip Boyutu	1,720	0,724	5,639	1	0,018*	5,587	
Özel Ekip Üyeleri	1,813	0,728	6,199	1	0,013*	6,126	
Yeniden Yapılandırma Süreci	1,056	0,811	1,694	1	0,193	2,875	
Uygulama Alanı			2,109	5	0,834		
Uygulama Alanı (ERP)	2,569	3,472	0,547	1	0,459	13,048	
Uygulama Alanı (Mobil Uygulamalar)	0,408	2,731	0,022	1	0,881	1,503	
Uygulama Alanı (Finansal ve Yönetimsel)	-	17,377	12924,909	0,000	1	0,999	0,000
Uygulama Alanı (Web Uygulamaları)	2,113	1,800	1,378	1	0,240	8,274	
Uygulama Alanı (Özel Uygulamalar)	2,466	3,165	0,607	1	0,436	11,776	
Dış Kaynak Kullanım Düzeyi (Dış kaynak kullanılmadı)	0,012	2,010	0,000	1	0,995	1,013	
Kullanım Kılavuzu			0,813	2	0,666		
Kullanım Kılavuzu (Kullanım kılavuzu tüm yazılım sistemini kapsamaz)	-0,134	2,054	0,004	1	0,948	0,874	
Kullanım Kılavuzu (Teknik terminolojiyle yazılmış, anlaşılır olmayan kullanım kılavuzu)	1,841	2,285	0,649	1	0,421	6,300	
Sabit	-	33,091	12,046	7,546	1	0,006*	0,000

Tablo 4.16 incelendiğinde, bağımsız değişkenlerden “Gerçek Süre” değişkeninin gerçekleşen yüksek performans sınıflandırma üzerinde anlamlı ve 0.622’lik pozitif etkisi olduğu ( $\beta= 0.622$ ;  $p=0.018<0.05$ ), “Ekip Boyutu” değişkeninin yüksek performans üzerinde 0.1.720’lik pozitif bir etkisi olduğu ( $\beta= 1.720$ ;  $p=0.018<0.05$ ) ve “Özel Ekip Üyeleri” değişkeninin yüksek performans üzerinde 1.813’lük pozitif bir etkisi olduğu ( $\beta= 1.813$ ;  $p=0.013<0.01$ ) belirlenmiştir. “Gerçek Süre” değişkeni proje başında bilinmeyeceği için efor tahmininde kullanımı anlamlı olmamakla birlikte, bu değişkenin SEERA (Software Engineering in Sudan) [17] veri setindeki efor denkleminin bir parçasını oluşturması sebebiyle kurulan modelde etkisi yüksek bağımsız değişkenler arasında çıktığı görülmektedir.

Lojistik regresyona yöntemi ile oluşturulan Model-2’ye ait sonuç aşağıda verilmiştir.

$$Z=-33.091 + 0,173*\text{Tahmini Süre} + 0,622*\text{Gerçek Süre} + 0,002*\text{Nesne Sayısı} + 0,996*\text{Kullanıcı Kararlılığı} + 1,253*\text{Gereksinim Kararlılığı} + 1,72*\text{Ekip Boyutu} + 1,813*\text{Özel Ekip Üyeleri} + 1,056*\text{Yeniden Yapılandırma Süreci} + 2,569*\text{Uygulama Alanı(1)} + 0,408*\text{Uygulama Alanı(2)} - 17,377*\text{Uygulama Alanı(3)} + 2,113*\text{Uygulama Alanı(4)} + 2,466*\text{Uygulama Alanı(5)} + 0,012*\text{Dış Kaynak Kullanım Düzeyi(1)} - 0,134*\text{Kullanım Kılavuzu (1)} + 1,841*\text{Kullanım Kılavuzu (2)}$$

Araştırma sonucu oluşturulan Model-2’nin performansının belirlenmesi için model olmadan sınıflandırma tahminlemesine bakılmış ve daha sonra modele bağlı sınıflandırma tahminlemesi yapılarak performansı gruplandırılması konusunda doğru tahmin oranı incelenmiştir. Model-2 doğru sınıflandırma düzeyleri Tablo 4.17’de verilmiştir.

Tablo 4.17: Model-2'nin Sınıflandırma Başarısı

	Gerçek Efor	Tahmin Edilen Efor		
		Yüksek	Düşük	Başarı
Model Öncesi	Yüksek	0	39	0
	Düşük	0	71	100
<b>Başarı Yüzdesi</b>				<b>64,5</b>
	Gerçek Efor	Tahmin Edilen Efor		
		Yüksek	Düşük	Başarı
Model Sonrası	Yüksek	38	1	97,4
	Düşük	1	70	98,6
<b>Başarı Yüzdesi</b>				<b>98,2</b>

Model-2 oluşturulmadan tüm eforları düşük olarak sınıflandırıldığında %64.5'lik doğru sınıflandırma yapıyorken,

Lojistik regresyon ile kurulan Model-2 sonucunda ise; Gerçekte yüksek eforlu projelerin 38'ini doğru sınıflandırırken 1 projeyi düşük efor grubuna atamış ve %97,4 oranla yüksek eforlu gerçekleşen projeleri doğru tahmin etmiştir.

Gerçekte düşük eforlu projelerin 70'ini doğru sınıflandırırken 1 projeyi yüksek eforlu olarak tahmin etmiş ve düşük eforlu olan projelerin %98,6'sını doğru tahmin etmiştir. Genel doğru sınıflandırma yüzdesine bakıldığında Model-2 110 projenin 108'ini model sayesinde doğru tahmin ederek %98.2'lik başarıya ulaşmıştır. Normalde model olmadan doğru sınıflandırma yüzdesi %64.5 iken model sayesinde %98.2 doğru sınıflandırma yapılabilmektedir.

### 4.3. Model-1 ve Model-2 Sonuçlarının Değerlendirilmesi

Yazılım efor tahmini üzerine iki farklı sınıflandırma modeli oluşturulmuştur. Kullanılan veri seti bir dizi hazırlık sürecinden geçirilmiştir. Veri setindeki aykırı değerler belirlenerek ilgili projeler veri setinden çıkarılmış, veri seti normal dağılıma sahip hale getirilmiştir. Bununla birlikte sürekli bağımsız değişkenlerin ortalama hesaplamalarının yapılması, bağımlı değişken ile kategorik ve sürekli bağımsız değişkenler ile farklılıklarının

incelenmesi sağlanmıştır. Bağımlı değişkenin sürekli ve kategorik olarak ele alınması ile iki farklı sınıflandırma modeli oluşturulmuştur. Bu iki model ile veri ön işleme çalışmalarının sınıflandırma modeline etkilerinin incelenmesi sağlanmıştır.

Çalışma kapsamındaki ilk modelde (Model-1) bağımlı değişken olan “Gerçek Efor” değişkeni sürekli olarak ele alınmış olup, bağımsız değişkenlerle farklılık analizleri yapılmıştır. Bu analizlerde “Gerçek Efor” bağımlı değişkeni ile farklılaşan bağımsız değişkenler belirlenmiş, bu değişkenler arasında uyum iyiliği testi yapılarak modele anlamlı bir katkı yapmayacağı düşünülen anlamlı olmayan değişkenler çıkarılmıştır. Bağımlı değişken kategorik hale getirilmiş ve lojistik regresyon yöntemi kullanılarak belirlenen bağımsız değişkenler ile Model-1 oluşturulmuştur. “Gerçek Efor” bağımlı değişkeni ile “Ekip Boyutu”, “Özel Ekip Üyeleri”, “Gerçek Süre” ve “Önceki Sisteme İlişkin Üst Yönetimin Görüşü (Hayır)” değişkenlerinin anlamlı ilişkiye sahip olduğu tespit edilmiştir.

Araştırma sonucu oluşturulan sınıflandırma modeli performansının belirlenmesi için model olmadan sınıflandırma tahminlemesine bakılmış ve daha sonra modele bağlı sınıflandırma tahminlemesi yapılarak performansı gruplandırılması konusunda doğru tahmin oranının arttığı tespit edilmiştir. Bu çalışma ile oluşturulan Model-1 sınıflama modeli sayesinde veri setinde bulunan birçok değişken analiz edilmiş, efor üzerinde etkili olan değişkenler kullanılarak yazılım eforunu %89'luk açıklama oranı ve %93.6'lık başarı oranı ile tahminleyebilen bir model oluşturulmuştur.

Çalışma kapsamındaki ikinci modelde (Model-2) bağımlı değişken olan “Gerçek Efor” değişkeni kategorik olarak ele alınmış olup, bağımsız değişkenlerle arasındaki ilişkilerin incelenmesinde bağımsız örneklem t-testi ve Ki-kare analizi kullanılmıştır. Yapılan analizler neticesinde ilişkileri anlamlı olan değişkenlerle lojistik regresyon yöntemi kullanılarak Model-2 oluşturulmuştur. “Gerçek Efor” bağımlı değişkeni ile “Ekip Boyutu”, “Özel Ekip Üyeleri” ve “Gerçek Süre” değişkenlerinin anlamlı ilişkiye sahip olduğu tespit edilmiştir.

Araştırma sonucu oluşturulan sınıflandırma modeli performansının belirlenmesi için model olmadan sınıflandırma tahminlemesine bakılmış ve daha sonra modele bağlı sınıflandırma tahminlemesi yapılarak performansı gruplandırılması konusunda doğru tahmin oranının arttığı tespit edilmiştir. Bu çalışma ile oluşturulan Model-2 sınıflama modeli sayesinde veri setinde bulunan birçok değişken analiz edilmiş, efor üzerinde etkili olan

değişkenler kullanılarak yazılım eforunu %91.4'lük açıklama oranı ve %98.2'lik başarı oranı ile tahminleyebilen bir model oluşturulmuştur.

Sınıflandırma modelleri kıyaslandığında Tablo 4.18'de görüleceği üzere, Model-2 ile yazılım eforunu açıklama oranı %89'dan %91.4'e; model başarı oranı ise %93.6'dan %98.2'ye çıkarılmıştır. Bu sonuçlar, değişkenler üzerinde yapılan ön işleme çalışmalarının model üzerinde olumlu etkilerinin olduğunu göstermektedir.

Tablo 4.18 : Sınıflama Modellerinin Karşılaştırması

	Model Öncesi		Model Sonrası
	Başarı Oranı	Başarı Oranı	Açıklama Oranı
Model-1	64,5	93,6	0,890
Model-2	64,5	98,2	0,914

Bununla birlikte literatürde SEERA veri seti ile Rasgele Orman sınıflandırma algoritması kullanılarak oluşturulan yazılım eforu tahminleme çalışmasında [20] elde edilen sonuçlar ile Model-1 ve Model-2 sonuçları kıyaslanmıştır. Mustafa ve Osman tarafından ele alınan bu çalışmada Rastgele Orman algoritması ile kurulan model sonucunda “Gerçek Efor” ile “Ekip Boyutu”, “Özel Ekip Üyeleri” ve “Nesne Sayısı” değişkenlerinin anlamlı ilişkiye sahip olduğu tespit edilmiştir. Bu değişkenlerin kurulan Model-1 ve Model-2'de yer aldığı ve bağımlı değişken ile anlamlı ilişkisi olduğu görülmektedir.

## 5. SONUÇ

Proje eforunun doğru tahmin edilmesi, proje yönetiminin etkinliğini doğrudan etkilemektedir. Yanlış efor tahminleri, projenin sonunda firma için itibar kaybı ve maliyet artışı gibi olumsuz sonuçlara neden olabilmektedir. Bu nedenle, proje efor tahmininde kullanılan tahmin yöntemi büyük önem taşımaktadır. Mevcut tahminler genellikle sezgisel bir yaklaşımla yapıldığı için tutarlılık oranları düşük olabilmektedir. Bu çalışmada, sınıflandırma modeli yaklaşımı önerilerek otomatik bir efor tahmin sistemi geliştirilmiştir.

Bu çalışmada, sınıflandırma tabanlı bir tahmin yönteminin, proje yönetimindeki efor tahmini sorunlarına nasıl uygulandığı incelenmiştir. Çalışmada, veri seti üzerinde analizler yapılmış, gerçek efor üzerinde etkisi yüksek değişkenler belirlenmiş ve lojistik regresyon yöntemi ile iki farklı yazılım eforu sınıflama modeli oluşturulmuştur. Oluşturulan Model-1 sınıflama modeli sayesinde yazılım eforunu %89'luk açıklama oranı ve %93.6'lık başarı oranı ile tahminleyebilen bir model; Model-2 sınıflama modeli sayesinde ise yazılım eforunu %91.4'lük açıklama oranı ve %98.2'lik başarı oranı ile tahminleyebilen iki farklı model oluşturulmuştur. Model-2 ile açıklama oranı ve başarı oranı daha yüksek sonuçlar elde edilmiş ve bu sayede veri ön işleme çalışmalarının sınıflandırma modeline olumlu etkileri olduğu tespit edilmiştir.

Literatürde yapılan çalışmalar çoğunlukla yazılım eforunu noktasal olarak tahminlemeyi hedeflediği için doğrudan karşılama sağlanamamakla birlikte, çalışmada başarı oranı yüksek sonuçların alındığı görülmüştür. Bununla birlikte literatürde SEERA veri seti ile Rasgele Orman sınıflandırma algoritması kullanılarak oluşturulan yazılım eforu tahminleme çalışmasında [20] elde edilen sonuçlar ile Model-1 ve Model-2 sonuçları kıyaslanmış, gerçek efor ile anlamlı ilişkiye sahip değişkenlerin benzer olduğu tespit edilmiştir. Neticede yazılım eforu tahminleme çalışmalarının sınıflandırma algoritmaları kullanılarak etkili şekilde kullanılabilmesi ve literatürdeki çalışmalarla uyumlu sonuçların elde edilebileceği görülmüştür.

Bu çalışmanın bir sonraki adımında, farklı sınıflandırma algoritmaları ile kapsamlı veri setleri kullanılarak yazılım efor tahmin çalışması yapılması ve literatürdeki çalışmalarla karşılaştırılması hedeflenmektedir. Bunun dışında veri ön işleme aşamasında farklı

yöntemler uygulanarak veri seti ile algoritma uyumlulukları ele alınıp tahmin oranları üzerindeki etkileri incelenebilecektir.



## KAYNAKLAR

- [1] P. Sentas, L. Angelis, I. Stamelos ve G. L. Bleris, "Multiple Logistic Regression as Imputation Method Applied on Software Effort Prediction," Bilişim Bölümü, Selanik Aristoteles Üniversitesi, Yunanistan, 2011.
- [2] Y. Takagi, O. Mizuno ve T. Kikuno, "An Empirical Approach to Characterizing Risky Software Projects Based on Logistic Regression Analysis," cilt 10, no. 495-515, 2005.
- [3] T. Christiansen, P. Wuttidittachotti, S. Prakancharoen ve S. . A.-o. Vallipakorn, "Prediction of Risk Factors of Software Development Project by Using Multiple Logistic Regression," *ARPJ Journal of Engineering and Applied Sciences*, cilt 10, no. 3, ISSN 1819-6608, 2015.
- [4] M. Ayyıldız, "Yazılım Projeleri Ölçüm Sonuçları Veritabanının Oluşturulması Ve Yeni Yazılım Projelerinin Maliyet Tahmininde Kullanımı," Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul, 2007.
- [5] A. Sezer, "Yazılım Projelerinde Yapay Sinir Ağı Uygulaması İle Maliyet Tahmini," Bilgisayar Mühendisliği Bölümü, Haliç Üniversitesi, İstanbul, 2008.
- [6] A. Bakır, B. Turhan ve A. Bener, "Software Effort Estimation As A Classification Problem," Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, İstanbul, 2008.
- [7] O. Fedotova, L. Teixeira ve H. Teixeira, "Software Effort Estimation With Multiple Linear Regression: Review And Practical Application," *Journal of Information Science and Engineering*, cilt 29, no. 5, 2011.
- [8] A. B. Nassifa, D. Hob ve L. F. Capretza, "Towards An Early Software Estimation Using Log-linear Regression And A Multilayer Perceptron Model," *The Journal of Systems and Software*, cilt 86, no. 144-160, 2012.

- [9] J. Wena, S. Li, Z. Lin, Y. Hu ve C. Huang, "Systematic Literature Review Of Machine Learning Based Software Development Effort Estimation Models," *Information and Software Technology*, cilt 54, no. 41-59, 2012.
- [10] P. Pospieszny, B. Czarnacka-Chrobot ve A. Kobylinski, "An Effective Approach For Software Project Effort And Duration Estimation With Machine Learning Algorithms," *The Journal of Systems & Software*, 2017.
- [11] E. I. Mustafa ve R. Osman, "An Analysis of the Inclusion of Environmental Cost Factors in Software Cost Estimation Datasets," %1 içinde *2018 IEEE International Conference on Software Quality, Reliability and Security Companion*, 2018.
- [12] M. Gültekin, "Makine Öğrenmesi Tabanlı Yazılım Maliyet Tahmini Yöntemlerinin Karşılaştırmalı Analizi," Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul, 2019.
- [13] S. Olgun, "Yazılım Projelerinin Yönetiminde Maliyet Tahmini İçin Derin Öğrenme Tabanlı Yeni Bir Yaklaşım," Endüstri Mühendisliği Bölümü, İstanbul Üniversitesi, İstanbul, 2020.
- [14] M. T. H. Al-Khazraji, "New Software Cost Estimation Approach By Using Machine Learning Based Feature Extraction Techniques," Elektrik ve Bilgisayar Mühendisliği Bölümü, Altınbaş Üniversitesi, İstanbul, 2021.
- [15] Ş. E. Kara ve R. Şamlı, "Yazılım Projelerinin Maliyet Tahmini için WEKA'da Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Analizi," *Avrupa Bilim ve Teknoloji Dergisi*, cilt 23, no. 415-426, 2021.
- [16] D. Derya, "Multi-Objective Software Project Cost Estimation Using Recent Machine Learning Approaches," Bilgisayar Mühendisliği Bölümü, Çankaya Üniversitesi, Ankara, 2023.
- [17] E. I. Mustafa ve R. Osman, "The SEERA Software Cost Estimation Dataset," 29 Ağustos 2020. [Çevrimiçi]. Available: [zenodo.org/records/3987969](https://zenodo.org/records/3987969).

- [18] . E. I. Mustafa ve R. Osman, "SEERA: A Software Cost Estimation Dataset For Constrained Environments," %1 içinde *In Proceedings of the 16th International Conference on Predictive Models and Data Analytics in Software Engineering*, Sacramento, California, USA, 2020.
- [19] A. Aslan, "Detailed Analysis Of MI Algorithms In Project Management Applications," Bilgisayar Mühendisliği Bölümü, Bahçeşehir Üniversitesi, İstanbul, 2023.
- [20] E. I. Mustafa ve R. Osman, "A Random Forest Model For Early-Stage Software Effort Estimation For The SEERA Dataset," *Information and Software Technology*, cilt 169, no. 107-413, 2024.
- [21] A. Atalar, "Lojistik Regresyon Analizi ve Makine Öğrenmesi Uygulamaları," Matematik Anabilim Dalı, Haliç Üniversitesi, İstanbul, 2022.
- [22] S. Şenel ve B. Alatlı, "Lojistik Regresyon Analizinin Kullanıldığı Makaleler Üzerine Bir İnceleme," *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, no. 1309-6575, 2014.
- [23] A. A. Köprü, "Yüksek Boyutlu Büyük Verilerde Aykırı Değerleri Tanılama Teknikleri," İstatistik Anabilim Dalı, Ankara Üniversitesi, Ankara, 2023.
- [24] D. George ve P. Mallery, *SPSS for Windows Step by Step: A Simple Guide and Reference*, 17.0 update, Boston: Pearson, 2010.
- [25] A. Ghasemi ve S. Zahediasl, "Normality Tests for Statistical Analysis: A Guide for Non-Statisticians," *International Journal of Endocrinology and Metabolism*, cilt 10, no. 2, 486-489, 2012.
- [26] B. G. Tabachnick, ve L. S. Fidell , *Using Multivariate Statistics*, Boston: Pearson, 1996.
- [27] C. A. Mertler ve R. A. Vannatta, *Advanced and Multivariate Statistical Methods: Practical Application and Interpretation*. 3rd Edition, 2005.

[28] A. Field, *Discovering statistics using SPSS* (2nd edition), 2005.



