Mühendislik Bilimleri ve Tasarım Dergisi
13(2), 528-544, 2025
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.1604190

Araştırma Makalesi Research Article

PROJE EFOR TAHMİNİ İÇİN MAKİNE ÖĞRENMESİ MODELLERİNİN GELİŞTİRİLMESİ VE SHAP YÖNTEMİ KULLANILARAK AÇIKLANMASI

Esma Nur KAYA[1], Yasin GÖRMEZ[2*]

[1] Sivas Cumhuriyet Üniversitesi, Sosyal Bilimler Enstitüsü, Yönetim Bilişim Sistemleri Anabilim Dalı, Sivas, Türkiye

[2] Sivas Cumhuriyet Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Yönetim Bilişim Sistemleri Bölümü, Sivas, Türkiye

Anahtar Kelimeler

Proje Yönetimi, Proje Efor Tahmini, Makine Öğrenmesi, Açıklamalı Yapay Zekâ, SHAP.

Öz

Günümüzde işletmeler, dijitalleşen dünyaya uyum sağlamak için başarılı bir proje yönetimine ihtiyaç duymaktadır. Özellikle yazılım projelerinin artışıyla birlikte, doğru efor tahmini yapmak kritik bir süreç haline gelmiştir. Efor tahmini, projenin tamamlanması için gereken zaman ve iş gücü miktarını tahmin ederek maliyetleri optimize etmeyi sağlamaktadır. Bu çalışmada, proje efor tahmini için rastgele orman, karar ağacı, doğrusal regresyon, yapay sinir ağı, GradientBoost ve AdaBoost yöntemleri geliştirilmiştir. china_original, cocomonasa_v1, humans2, nasa93, usp05 ve usp05-ft gibi 6 farklı veri seti üzerinde 50 tekrarlayan sınama yaklaşımı kullanılarak analizler yapılmış ve modeller ortalama mutlak hata, ortalama logaritmik kare hatası, belirleme katsayısı ve ortalama göreli büyüklük hatası metrikleri kullanılarak karşılaştırılmıştır. Analiz sonuçlarına göre yapay sinir ağı, rastgele orman, karar ağaçları ve GradientBoost modellerinin farklı veri setlerinde en başarılı modeller olduğu gözlemlenmiştir. Proje efor tahmini için ise en başarılı modelin karar ağacı olduğu kanısına varılmıştır. Çalışmada yapılan diğer bir analizde ise, geliştirilen modeller açıklamalı yapay zekâ modeli olan SHAP (SHapley Additive exPlanations) yöntemi kullanılarak açıklanmıştır. Yapılan açıklamalar doğrultusunda her bir veri seti için bazı özniteliklerin model karar alma sürecinde diğer özniteliklere göre daha etkili olduğu gözlemlenmiştir.

DEVELOPMENT OF MACHINE LEARNING MODELS FOR PROJECT EFFORT PREDICTION AND EXPLANATION USING SHAP METHOD

Keywords

Project Management, Project Effort Estimation, Machine Learning, Explainable Artificial Intelligence, SHAP.

Abstract

In today's digitalized world, successful project management has become essential for businesses, with accurate effort estimation emerging as a critical component due to the increasing prevalence of software projects. Effort estimation facilitates cost optimization by predicting the time and labor required for project completion. This study developed and evaluated six regression models—random forest, decision tree, linear regression, neural network, GradientBoost, and AdaBoost—for project effort estimation. Analyses were conducted on six datasets (china_original, cocomonasa_v1, humans2, nasa93, usp05, and usp05-ft) using 50 repeated holdout tests, and model performance was compared using metrics such as mean absolute error, mean squared logarithmic error, coefficient of determination, and mean relative magnitude error. The results demonstrated that artificial neural networks, random forest, decision trees, and GradientBoost models performed most effectively across the datasets, with the decision tree identified as the best-performing model for effort estimation. Furthermore, the study utilized the SHAP (Shapley Additive Explanations) method to interpret the models, revealing that specific attributes were more influential than others in the decision-making process across different datasets.

Alıntı / Cite

Kaya, E. N., Görmez, Y., (2025). Proje Efor Tahmini İçin Makine Öğrenmesi Modellerinin Geliştirilmesi ve SHAP Yöntemi Kullanılarak Açıklanması, Mühendislik Bilimleri ve Tasarım Dergisi, 13(2), 528-544.

Yazar Kimliği / Author ID (ORCID Number)

E. N. Kaya, 0000-0003-1688-3832
Y. Görmez, 0000-0001-8276-2030

Makale Süreci / Article Process	
Başvuru Tarihi / Submission Date	20.12.2024
Revizyon Tarihi / Revision Date	08.05.2025
Kabul Tarihi / Accepted Date	16.05.2025
Yayım Tarihi / Published Date	27.06.2025

[*] İlgili yazar / Corresponding author: yasingormez@cumhuriyet.edu.tr, +90 346 487 00 00

DEVELOPMENT OF MACHINE LEARNING MODELS FOR PROJECT EFFORT PREDICTION AND EXPLANATION USING SHAP METHOD

Esma Nur KAYA[1] Yasin GÖRMEZ[2†],
[1] Sivas Cumhuriyet University, Institute of Social Sciences, Department of Management Information Systems, Sivas, Turkey
[2] Sivas Cumhuriyet University, Faculty of Economics and Administrative Sciences, Department of Management Information Systems, Sivas, Turkey

Highlights

- The performance of regression models in project effort prediction was evaluated.
- Regression models were interpreted using the SHAP method.
- The effects of attributes from different datasets on model decision-making were analyzed.

Purpose and Scope

The study evaluates the performance of regression models in project effort estimation, aiming to identify the most suitable models for automated project effort estimation systems. Using diverse datasets, multiple evaluation metrics, and a repeated holdout approach, the study seeks to generalize findings by assessing model performance under varying conditions. Furthermore, through model interpretability analyses, the study investigates which attributes have the most significant impact on automatic project effort estimation.

Design/methodology/approach

Random Forest, Decision Tree, Artificial Neural Network, Gradient Boosting, AdaBoost, and Linear Regression methods were employed to achieve the objectives of this study. The models were compared using evaluation metrics, including mean absolute error, mean squared logarithmic error, coefficient of determination, and mean relative magnitude error. Analyses were conducted on six datasets (china_original, cocomonasa_v1, humans2, nasa93, usp05, and usp05-ft) and interpreted using the SHAP method. Beeswarm plots were generated to visualize the explanations, measuring the impact of each attribute on the model's estimation process.

Findings

Based on the analyses, it was observed that Artificial Neural Network, Random Forest, Gradient Boosting, and Decision Tree models were the most successful across different datasets. While both Random Forest and Decision Tree models performed best on the cocomonasa_v1 dataset, the Decision Tree model was the top-performing model on two datasets, Gradient Boosting excelled on two datasets, and Artificial Neural Network was the most successful on one dataset. Consequently, the Decision Tree model is considered the most effective among the models analyzed in this study for project effort estimation. Furthermore, explanation analysis revealed that certain attributes are significantly more influential than others during the model decision-making process.

Research limitations/implications

The number of regression models and datasets used in the study is limited and could be increased. In addition, the study did not perform hyperparameter optimization and used a single explainable artificial intelligence method. For future studies, it is suggested to diversify the methods used and to make comparisons for the explainable artificial intelligence models.

Originality

The innovative aspect of the study is the analysis of different regression models on different datasets. In addition to this situation, the fact that explainable artificial intelligence approaches have never been used for project effort estimation in the literature makes the study novel.

† Corresponding author: yasingormez@cumhuriyet.edu.tr, +90 346 487 00 00

1. Giriş (Introduction)

Günümüzde dijitalleşme, kurum ve kuruluşların gündemini oldukça meşgul etmektedir. Özellikle büyük ölçekli firmalar birçok iş sürecini dijital ortama taşımıştır ve taşımaya da devam etmektedir. Bu kapsamda firmalar, yazılım projeleri üreterek yeni ürün ya da sistem geliştirmeyi amaçlamaktadır. Planlanan projelerin başarılı bir şekilde tamamlanabilmesi için iyi bir proje yönetim sürecinin yürütülmesi gerekmektedir. Yetersiz bir proje yönetimi sonucunda hedeflere ulaşamama, kaynak israfı, takım içi çatışmalar, risklerin yönetilmemesi, gecikmeler, kalite problemleri ve iletişim sorunları gibi olumsuz durumlarla karşılaşılabilmektedir. Proje yönetim sürecinde birçok temel unsuru etkileyen efor tahmini (ET), proje yönetimi için kritik bir öneme sahiptir.

Proje efor tahmini (PET), bir projenin tamamlanması için gereken iş miktarının zaman ve kaynaklar açısından önceden belirlenmesini kapsayan bir süreç olarak tanımlanabilmektedir. İş birimlerinde genellikle adam/saat, adam/gün ya da adam/ay gibi ölçütlerle ifade edilen ET, projenin kapsamına, görevlerin karmaşıklığına, mevcut kaynaklara ve geçmiş deneyimlere dayanılarak yapılmaktadır. Bu durumlar dikkate alındığında ET, proje yönetim sürecinde birçok açıdan önem taşımaktadır. Projenin tamamlanması için gereken sürenin belirlenmesi, doğru ET ile başladığı için yanlış tahminler, gereksiz gecikmelere neden olabilmektedir. Projede kullanılacak kaynakların (personel, donanım, yazılım vb.) miktarı ve kullanım süresi de ET ile belirlenmektedir. Yanlış ET, gereğinden az ya da fazla kaynak ayrımına neden olabileceği için projenin maliyet ve zaman açısından başarısız olmasına sebep olabilmektedir. Maliyet planlamasının da temelini oluşturan ET'nin yanlış yapılması durumunda bütçe aşımları veya finansal kaynak yetersizlikleri meydana gelebilmektedir. Doğru ET proje aşamalarına ve kalite kontrol süreçlerine yeterli zaman ayırmayı sağlarken, yanlış ET aşırı iş yükü ve gecikmelere neden olacağı için hem kalite süreçlerinde hem de risk yönetiminde olumsuz sonuçlara neden olabilmektedir.

Proje yönetim sürecinde ET yapabilmek için farklı yöntem ve teknikler kullanılmaktadır. Efor tahmininde uzman görüşlerinden sıklıkla faydalanılmaktadır. Uzman görüşleri kullanılarak yapılan efor tahmininde tecrübeli profesyonellerin geçmiş deneyimlerine güvenilmektedir. Genellikle karmaşık projelerde uygulanan ve sezgisel bir yaklaşımla hızlı sonuç alınan uzman görüş yönteminde tahmin sonucu uzmanların yanlılıklarına ve bilgi seviyesine bağlı olarak değişebilmektedir (Jorgensen, 2005). Analog yöntemler kullanılarak yapılan ET, geçmiş projelerle olan benzerliklere dayanılarak yapılmaktadır. Veri tabanındaki tamamlanmış projeler arasından mevcut projeye benzer olanların seçilerek değerlendirme yapıldığı bu yaklaşım, özellikle benzer projelerin bulunduğu durumlarda yüksek doğruluk sağlamaktadır (Shepperd vd., 1996). Paramedik yaklaşımla yapılan proje efor tahmininde matematiksel formüller veya istatistiksel modeller kullanılmaktadır. Bu yaklaşımda kullanılacak olan parametreler projelerin büyüklüğüne, karmaşıklığına ve diğer ölçütlere göre belirlenmektedir (Tsunoda vd., 2012). PET için kullanılan bir diğer yaklaşım olan Delphi yöntemi, anonim uzmanlardan gelen tahminlerin birden fazla turda toplandığı ve ortalama bir sonuca ulaşıldığı sistematik bir süreçtir. Delphi yönteminin temel amacı grup düşüncesi etkisini azaltmak ve daha tarafsız sonuçlar almayı sağlamaktır (Erasmus & Daneva, 2013). Son zamanlarda PET için yapay zekâ ve makine öğrenmesi tabanlı yaklaşımlar sıklıkla kullanılmaktadır. Bu yaklaşımlarda kullanılan yöntemler, geçmiş verileri kullanarak otonom tahminler yapabilmektedir. Bu yöntemlerin büyük miktarlarda veriyi işleyebiliyor olması, başarılı sonuçları hızlı bir şekilde elde edebilmesindeki en önemli etkenlerden biridir (Ritu & Bhambri, 2023).

Bu çalışmada ilk olarak PET için makine öğrenmesi modellerinin performansını ölçmek amaçlanmıştır. Bu bağlamda rastgele orman (RO), karar ağacı (KA), doğrusal regresyon (DR), yapay sinir ağı (YSA), GradientBoost (GB) ve AdaBoost (AB) olmak üzere 6 farklı yöntem süreğen değerleri tahmin etmek üzere eğitilmiştir. Modelleri eğitmek ve performanslarını karşılaştırmak için china_original, cocomonasa_v1, humans2, nasa93, usp05 ve usp05-ft olmak üzere 6 farklı proje efor veri seti kullanılmıştır. Önerilen modellerin farklı koşullardaki performanslarının da ölçülebilmesi için tekrarlayan sınama yönteminden faydalanılmıştır. Çalışmanın diğer bir amacı ise veri setlerinde kullanılan özniteliklerin makine öğrenmesi modellerinin performansına olan etkisinin hesaplanmasıdır. Bu kapsamda, eğitilen makine öğrenmesi modelleri SHapley Additive exPlanations (SHAP) yöntemi kullanılarak açıklanmıştır. SHAP sayesinde eğitilmiş modellere hangi özniteliğin, modelin karar almasına ne kadar etki ettiği ölçülmüştür. Çalışmanın literatüre ilk katkısı, farklı makine öğrenmesi modellerinin PET üzerindeki etkisinin ölçülmesi olarak değerlendirilmektedir. Çalışmanın literatüre diğer bir katkısının ise PET için önerilen makine öğrenmesi modellerinin SHAP kullanılarak açıklanması olduğu düşünülmektedir. Bu yaklaşım sayesinde PET için önerilen makine öğrenmesi modellerinin karar alma süreçlerinin daha net bir şekilde yorumlanabileceği ön görülmektedir. Literatür incelendiğinde PET için açıklamalı yapay zekâ yaklaşımlarının daha önce kullanılmamış olması ise çalışmayı özgün olarak değerlendirmemizi sağlamıştır. Çalışmanın devamında literatür özeti yapılmış, materyal ve yöntem anlatılmış, deney sonuçları verilmiş ve sonuçlar özetlenmiştir.

2. Kaynak Araştırması (Literature Survey)

Günümüze kadar PET için oldukça fazla çalışma yapılmıştır. Anolog tabanlı yöntemlerin birbirleri ve doğrusal regresyon ile karşılaştırıldığı çalışmada, küçük bir veri kümesi sağlandığında insanların efor tahmininde araçlara göre daha iyi olduğu sonucuna varılmıştır. Bu durumun yanı sıra aynı çalışmada yazılım projeleri için efor tahmininde insanların ve araçların yeteneklerinin birleştirilmesinin daha verimli sonuçlar elde etmeye katkı sunacağı çıkarımı yapılmıştır (Walkerden & Jeffery, 1999). Vaka tabanlı akıl yürütme yöntemlerinin proje efor tahminindeki başarısını artırmak için genetik algoritma ile birleştirildiği çalışmada öznitelik seçim ve öznitelik ağırlıklandırma yapılmıştır. PROMISE veri tabanı kullanılarak yapılan deney sonuçlarına göre genetik algoritma kullanarak yapılan parametre iyileştirmesinin proje efor tahmin modeli performansını artırdığı gözlemlenmiştir (Hameed vd., 2023). Destek vektör makinaları, YSA, genelleştirilmiş doğrusal modeller ve topluluk yöntemlerinin ISBSG veri seti kullanılarak eğitildiği çalışmada proje efor tahmini için 0.19 ortalama mutlak hata, 0.07 minimum karesel hata ve 0.27 ortalama karekök sapması elde edilmiştir (Pospieszny vd., 2018). Özelleştirilmiş makine geliştirme projeleri için efor tahmini yapılan çalışmada sekiz özniteliği olan bir veri seti kullanılmıştır. İlgili çalışmada YSA modelinin gizli katmandaki nöron sayısı optimize edilerek 0.11 logaritmik ortalama karesel hata ve 0.94 R skoru elde edilmiştir (Şengüneş & Öztürk, 2023). Yazılım projelerinde efor tahmini analizi yapılan çalışmada, yerelleştirilmiş mahalle karşılıklı bilgi tabanlı sinir ağı, nöro-bulanık mantık, uyarlanabilir genetik tabanlı sinir ağ ve genetik fil güdümünün optimizasyonu tabanlı nöro bulanık ağ önerilmiştir. cocomo81, cocomonasa_2, cocomonasa_v1, Desharnais ve China veri setleri kullanılarak yapılan analizler sonucunda 88.23% ile 28.51% arasında tahmin doğruluk oranları elde edilmiştir (Sharma & Vijayvargiya, 2022). Destek vektör makinası ve radyal tabanlı sinir ağları kullanılarak önerilen hibrit model, endüstriyel ve öğrenci projelerinden toplanan çok sayıda örnek kullanılarak oluşturulan veri seti ile eğitilmiştir. Yapılan deney sonuçlarına göre önerilen model, tüm veri setlerinde kullanım durum noktaları yöntemi kullanılarak geliştirilen modellere göre daha iyi sonuçlar elde etmiştir (Azzeh & Nassif, 2016). GitHub platformundan toplanan verilerle gerçek projelerin efor veri setinin hazırlandığı çalışmada Adaboost ve sınıflandırma-regresyon ağacı yöntemleri kullanılmıştır. İlgili çalışmada yapılan analizler sonucunda personel faktörünün efor tahminini iyileştirmede etkili olduğu, önerilen modelin mevcut modellerle karşılaştırılabilir bir performans elde ettiği ve önerilen sınıflandırma-regresyon ağacı yöntemi sayesinde veri setindeki örneklerin çevrimiçi olarak artırılabileceği kanaatine varılmıştır (Qi vd., 2017). Nöro bulanık ağ ve genetik fil güdümü optimizasyonu kullanarak yazılım eforlarını maliyet açısından tahmin etmeyi amaçlayan model, endüstriyel projelerden elde edilen beş tarihsel ve kıyaslama veri seti kullanılarak değerlendirilmiştir. Yapılan analizler sonucunda önerilen modelin, destek vektör regresyonları, dalgacık YSA ve karar ağacı yöntemlerine göre daha iyi yazılım projesi efor tahmini skoru elde ettiği gözlemlenmiştir (Sharma & Vijayvargiya, 2023). Optimize edilen YSA modelinin 15 farklı projedeki efor tahmini için kullanıldığı çalışmada, önerilen YSA modelinin COCOMO modeline göre daha başarılı olduğu sonucuna varılmıştır (Mukherjee & Malu, 2014). DR, destek vektör regresyonu, YSA, KA, RO ve topluluk yöntemlerinin 499 örneğe sahip Çin veri seti kullanılarak karşılaştırıldığı çalışmada RO ve DT yöntemlerinin diğer yöntemlere göre önemli ölçüde daha iyi olduğu sonucuna varılmıştır (Sharma & Vijayvargiya, 2021). Çoklu eklemeli regresyon ağaçları yönteminin DR, radyal tabanlı fonksiyon sinir ağları ve destek vektör regresyon modelleriyle NASA veri seti kullanılarak karşılaştırıldığı çalışmada, çoklu eklemeli regresyon ağaçları modelinin diğer modellere üstünlük sağladığı gözlemlenmiştir (Elish, 2009). NASA veri setinin kullanıldığı bir çalışmada COCOMO modeli Naive Bayes, lojistik regresyon ve rastgele orman yöntemleri ile karşılaştırılmış ve doğruluk, hassasiyet, kesinlik ve ROC eğrisi altında kalan alan bakımından en başarılı yöntemlerin Naive Bayes ve rastgele orman yöntemleri olduğu görülmüştür (BaniMustafa, 2018). Bayes ağları yönteminin efor tahmini için önerildiği çalışmada sırasıyla 40, 50 ve 70 proje içeren veri kümeleri kullanılmış ayrıca, birinci ve ikinci veri kümeleri birleştirilerek dördüncü veri seti ve birinci, ikinci ve üçüncü veri kümeleri birleştirilerek beşinci veri seti oluşturulmuştur. Çalışma kapsamında yapılan analizler sonucunda veri setlerinde %90 ile %99.375 arasında doğruluk, 0.026 ile 0.1065 ortalama mutlak hata ve 3.29 ile 12.80 arasında bağıl hatanın ortalama büyüklüğü elde edilmiştir (Dragicevic vd., 2017). Regresyon ağaçları, model ağaçları, çok katmanlı algılayıcı, doğrusal regresyon ve destek vektör regresyonu yöntemlerinin torbalama yaklaşımı ile topluluk modeline dönüştürüldüğü çalışmada NASA veri seti kullanılmıştır. Yapılan deney sonuçlarına göre torbalama yaklaşımının modellerin performansını iyileştirdiği görülmüştür. Bu durumun yanı sıra önerilen model, literatürde kendisinden önce kullanılmış olan hem doğrusal regresyon hem de radyal temel fonksiyonundan daha iyi sonuçlar etmiştir (Braga vd., 2007). Yazılım projelerinde erken aşamada efor tahmini için açık erişimli SEERA veri setinin kullanıldığı çalışmada rastgele orman yöntemi ile %70'den daha iyi başarı oranı elde edilmiştir (Mustafa & Osman, 2024). Yazılım geliştirmede efor tahmini için önerilen tam bağlı YSA modelinin hiper parametrelerinin meta sezgisel gri kurt yöntemi ile optimize edildiği çalışmada 12 farklı veri seti üzerinde analizler yapılmıştır. Yapılan analizler doğrultusunda meta sezgisel gri kurt ile hem hiper parametre optimizasyonu yapmanın hem de yerel optimumlara düşmekten kaçınma yeteneğini geliştirmenin YSA modelinin performansında istatiksel olarak anlamlı iyileşmeler yaptığı gözlemlenmiştir (Kassaymeh vd., 2024). Yazılım efor tahmini için SVR tabanlı topluluk yönteminin önerildiği çalışmada farklı çekirdek fonksiyonları kullanan SVR modelleri birleştirilmiştir. Çalışma sonucunda SVR yönteminin tek başına kullanılması ile farklı çekirdek fonksiyonları yardımıyla topluluk yöntemine dönüştürülmesi arasında fark olmadığı gözlemlenmiştir (Hosni,

2024). Yapılan bir çalışmada proje yönetim sürecinde farklı kurum ve kuruluşların personel yeterliliklerine göre proje tanımlanması ve değerlendirilmesi amaçlanmıştır. Geçmiş projelerden elde edilen veriler, makine öğrenmesi süreçleriyle analiz edilerek proje başarısına etkileri incelenmiştir. Geliştirilen sistem, yapay zekâ tekniklerini kullanarak proje başarısını tahmin etmiş ve proje bileşenlerini değerlendirme kriterlerine göre puanlayarak başarıyla sonuçlandırmıştır (Özgür vd., 2023). Yapay zekâ tekniklerinden yararlanarak proje yönetiminde risk analizi için kavramsal bir çerçeve geliştirilen çalışmada yapay zekânın proje yönetimindeki dönüştürücü potansiyeli, risk analizi ve karar alma süreçlerindeki rolü üzerinde durulmuştur.

Tablo 1. Yazılım projesi efor tahmini çalışmalarının karşılaştırmalı analizi
(Comparative analysis of software project effort estimation studies)

Çalışma	Kullanılan Veri Setleri	Yöntemler	Metrikler	Sonuçlar
(Hameed vd., 2023)	PROMISE	Vaka tabanlı akıl yürütme + Genetik algoritma (özellik seçim/ağırlıklandırma)	----------	Genetik algoritma ile parametre iyileştirme, model performansını artırmıştır.
(Pospieszny vd., 2018)	ISBSG	Destek vektör makinaları (SVM), YSA, GLM, topluluk yöntemleri	MAE: 0.19, MSE: 0.07, RMSE: 0.27	En iyi sonuçlar topluluk yöntemleri ile elde edilmiştir.
(Şengüneş & Öztürk, 2023)	8 özniteliğe sahip özel veri seti	Optimize edilmiş YSA	LogMSE: 0.11, R: 0.94	Optimize edilmiş YSA, yüksek doğruluk sağlamıştır.
(Sharma & Vijayvargiya, 2022)	Cocomo81, Cocomonasa_2, Cocomonasa_v1, Desharnais, China	LKMI-tabanlı sinir ağı, nöro-bulanık, genetik tabanlı sinir ağ	Tahmin doğruluğu: %28.51 - %88.23	Önerilen hibrit modeller, geleneksel yöntemlere göre daha iyi performans göstermiştir.
(Azzeh & Nassif, 2016)	Endüstriyel ve öğrenci projelerinden oluşan veri seti	Hibrit model (SVM + Radyal tabanlı sinir ağı)	----------	Hibrit model, kullanım durum noktaları tabanlı modellerden daha iyi sonuç vermiştir.
(Qi vd., 2017)	GitHub'tan toplanan bir veri seti	AdaBoost, CART	----------	Personel faktörü efor tahminini iyileştirmede etkili olmuştur. CART ile çevrimiçi veri artırımı mümkündür.
(Sharma & Vijayvargiya, 2023)	5 farklı endüstriyel veri seti	Nöro-bulanık ağ + Genetik fil güdümü optimizasyonu	----------	Önerilen model, SVR, dalgacık YSA ve karar ağacından daha iyi performans göstermiştir.
(Mukherjee & Malu, 2014)	15 farklı projeden elde edilmiş veri seti	Optimize edilmiş YSA, COCOMO	----------	YSA, COCOMO'dan daha başarılıdır.
(Sharma & Vijayvargiya, 2021)	Çin veri seti (499 örnek)	DR, SVR, YSA, KA, RO, DT	----------	RO ve DT, diğer yöntemlere göre anlamlı üstünlük sağlamıştır.
(Elish, 2009)	NASA	Çoklu eklemeli regresyon ağaçları, RBF ağları, SVR	----------	Çoklu eklemeli regresyon ağaçları diğer modelleri geride bırakmıştır.
(BaniMustafa, 2018)	NASA	Naive Bayes, lojistik regresyon, rastgele orman	Doğruluk, hassasiyet, kesinlik, ROC AUC	Naive Bayes ve rastgele orman en iyi performansı göstermiştir.
(Dragicevic vd., 2017)	40, 50, 70 projeli veri setleri (birleştirilmiş)	Bayes Ağları	Doğruluk: %90 - %99.375, MAE: 0.026 - 0.1065, Bağıl hata: 3.29 - 12.80	Bayes ağları yüksek doğruluk ve düşük hata oranları sağlamıştır.
(Braga vd., 2007)	NASA	Torbalama (Regresyon ağaçları, model ağaçları, MLP, SVR)	----------	Torbalama, model performansını iyileştirmiş ve literatürdeki diğer yöntemleri geride bırakmıştır.
(Mustafa & Osman, 2024)	SEERA	RO	Başarı oranı>%70	Rastgele orman, erken aşama efor tahmini için etkili olmuştur.
(Kassaymeh vd., 2024)	12 farklı veri seti	YSA + Gri kurt optimizasyonu	----------	Hiper parametre optimizasyonu, YSA performansını istatistiksel olarak anlamlı şekilde artırmıştır.
(Hosni, 2024)	Albrecht, COCOMO81, Desharnais, Kemerer, and Miyazaki	SVR tabanlı topluluk yöntemi (farklı çekirdekler)	----------	Topluluk yöntemi ile tek SVR arasında performans farkı bulunmamıştır.
(Özgür vd., 2023)	Geçmiş proje verileri	KA, RO, SVR	MAE: 0.581, MAPE: 0.007, RMSE: 1.005, R: 0.959	Proje başarısını tahmin eden sistem geliştirilmiş ve başarıyla uygulanmıştır.
(Tuncer, 2024)	----------	Yapay zekâ tabanlı risk analizi çerçevesi	----------	Yapay zekâ, proje yöneticilerinin riskleri öngörme ve azaltma becerilerini geliştirmiştir.
(Haris vd., 2023)	USP05-FT veri seti + uzman tahminleri	6 homojen sınıflandırıcı topluluğu	Doğruluk, duyarlılık güvenilirlik, f score	Uzman görüşleri ve topluluk yöntemlerinin birleştirilmesi, tahmin doğruluğunu ve güvenilirliği artırmıştır.

Çalışmanın bulguları, yapay zekâ teknolojilerinin kullanımının proje yöneticilerinin riskleri öngörme ve azaltma becerilerini geliştirebileceğini göstermektedir. Böylece projelerin zamanında, bütçe sınırları içinde ve kalite standartlarına uygun şekilde tamamlanmasının sağlanabileceği ifade edilmiştir (Tuncer, 2024). Yazılım

projelerinin efor tahminini iyileştirmek amacıyla uzman görüşlerini ve topluluk tabanlı bir çerçeveyi birleştiren yenilikçi yöntem sunan çalışmada uzmanlar tarafından sağlanan efor tahminleri, usp05-ft veri setine eklenerek modelin insan deneyimi ve sezgisini yansıtması sağlanmıştır. Altı farklı homojen sınıflandırıcı topluluğunun tahminleri birleştirilerek, daha yüksek doğruluk ve güvenilirlikte sonuçlar elde edilmiştir. Önerilen çerçevenin, yazılım yönetimi süreçlerini güçlendirerek kaynak tahsisini optimize etmesi ve yazılım projelerinin başarı oranını artırması beklenmektedir (Haris vd., 2023). Yapılan literatür taraması sonucunda elde edilen bulgular tablo 1'de özetlenmektedir.

PET için literatürde yapılan çalışmalar genel anlamda değerlendirildiğinde birçok makine öğrenmesi yönteminin kullanıldığı söylenebilmektedir. Bu çalışmalar literatüre farklı katkılar sunarak PET için uygun olan makine öğrenmesi yöntemleri, veri kümeleri ve performans metrikleri hakkında fikirler oluşmasını sağlamıştır. PET için modeller hem tek başına hem de topluluk yaklaşımı olarak önerilmiştir. Parametre optimizasyonu, öznitelik seçimi ve meta sezgisel yaklaşımlar ile modellerin performansını artırma gibi çalışmaların da literatürde yürütüldüğü görülmüştür. Literatürdeki bu durum dikkate alındığında önerdiğimiz çalışmada da farklı makine öğrenmesi modellerinin kullanılması literatür ile uyumluluk sağlamaktadır. Bu durumun yanı sıra çalışmada önerilmiş diğer bir yaklaşım olan açıklamalı yapay zekâ ise literatürde PET için daha önce kullanılmamıştır. Bu durum dikkate alındığında çalışmamızın, literatüre özgün katkılar sunacağı ön görülmektedir.

Çalışmamızı literatürdeki PET çalışmalarından ayıran en önemli yön ise kullanılan SHAP yöntemi olarak değerlendirilmektedir. Literatürde farklı açıklamalı yapay zekâ yöntemlerinin kullanıldığı görülmektedir. Yapılan bir çalışmada SHAP değerlerinin LIME (Local Interpretable Model-agnostic Explanations) yöntemine kıyasla daha tutarlı ve teorik olarak sağlam açıklamalar sunduğu gösterilmiştir. SHAP, oyun teorisindeki Shapley değerlerine dayanarak model çıktılarını küresel ve yerel düzeyde açıklarken, LIME yalnızca lokal yaklaşımlar sunmaktadır. Bu nedenle, SHAP'ın özellikle karmaşık modellerde daha güvenilir sonuçlar verdiği, LIME'nin ise seçilen örnekleme stratejisine bağlı olarak değişkenlik gösterebildiği belirtilmiştir (Lundberg & Lee, 2017). Açıklanabilir güçlendirme makinaları (Explainable Boosting Machine – EBM) ve SHAP yöntemlerinin karşılaştırıldığı çalışmada EBM yönteminin doğası gereği açıklanabilir bir model olması nedeniyle, SHAP gibi yöntemlere kıyasla daha şeffaf ve doğrudan yorumlanabilir olduğu vurgulanmıştır. EBM yönteminin bu avantaja sahip olmasına rağmen, SHAP yönteminin herhangi bir model üzerinde uygulanabilir olması, SHAP yöntemini daha esnek bir araç haline getirmektedir. Çalışmada, EBM yönteminin performansının lineer olmayan ilişkilerde sınırlı kalabildiği, SHAP yönteminin ise daha karmaşık etkileşimleri yakalayabildiği belirtilmiştir (Molnar, 2020). Yapılan başka bir çalışmada Morris Hassasiyet Analizi yönteminin (Morris Sensitivity Analysis - MSA) genel model davranışını anlamada etkili olduğu, ancak SHAP yönteminin aksine bireysel tahminlerin açıklanmasında yetersiz kaldığını ortaya konulmuştur. MSA, özellikle yüksek boyutlu verilerde hesaplama açısından verimli olsa da SHAP yönteminin sunduğu bireysel özellik katkılarını sağlayamamaktadır. Çalışmada, SHAP yönteminin hem global hem de lokal açıklamalarda daha kapsamlı olduğu, MSA yönteminin ise daha çok ön analiz aşamasında kullanılabileceğini öne sürülmüştür (Plumb vd., 2018). Tüm bu çalışmalar dikkate alındığında çalışmamızda SHAP yönteminin kullanılmasının en uygun olacağı kanaatine varılmıştır.

3. Materyal ve Yöntem (Material and Method)

3.1. Veri Setleri (Datasets)

Çalışmada, altı farklı proje efor veri seti kullanılmıştır: china_original, cocomonasa_v1, humans2, nasa93, usp05 ve usp05-ft. Bu veri setlerinin her birinde, yazılım projelerinin efor tahmini için çeşitli öznitelikler ve gerçekleşen süreler yer almaktadır. Veri setleri, yazılım projelerinin efor tahminini doğru bir şekilde yapabilmek için gerekli olan bilgileri içermekte ve her bir projenin farklı özellikleriyle analiz yapılmasını mümkün kılmaktadır. Bu çeşitlilik, modelin genelleme kapasitesini artırmayı hedefleyerek, yazılım mühendisliğinde efor tahmininin doğruluğunu geliştirmek için önemli bir kaynak oluşturmuştur (*Effor Estimation Datasets*, 2024).

3.1.1. china_original

china_original veri seti, Çin'deki yazılım geliştirme projelerinden toplanan toplamda 499 örnekten oluşmaktadır. Veri setinde, projelerin efor tahminlerini yapabilmek için 17 öznitelik yer almakta olup, efor değerleri ise adam/saat cinsinden sınıf değişkeni olarak sunulmaktadır. Bu efor değerleri, yazılım projelerinin tahmin edilmesi gereken önemli bir parametreyi temsil etmektedir. Veri seti, proje efor süresi açısından değerlendirildiğinde, en yüksek efor süresinin 54620 saat, en düşük efor süresinin ise 26 saat olduğu görülmektedir. Ayrıca, efor sürelerinin standart sapma değeri 6474 saat olarak hesaplanmıştır, bu da veri setindeki efor sürelerinin önemli bir dağılım gösterdiğini ve çeşitliliğin yüksek olduğunu işaret etmektedir.

3.1.2. cocomonasa_v1

cocomonasa_v1 veri seti, 1980 ve 1990 yılları arasında gerçekleşen farklı merkezlerden toplanan 60 NASA projesinden oluşmaktadır. Bu veri seti, 16 öznitelik ve adam/saat cinsinden efor değerlerini içeren sınıf değişkenlerini barındırmaktadır. Proje efor süresi açısından değerlendirildiğinde, en yüksek sürenin 3240 saat, en düşük sürenin ise 8.4 saat olduğu gözlemlenmektedir. Ayrıca, süre verileri arasında standart sapma değeri 651 saat olarak hesaplanmıştır. Bu değerler dikkate alındığında, veri setindeki efor sürelerinin geniş bir dağılıma sahip olduğu ve projelerin farklı ölçeklerde yer aldığı görülmektedir. Bu çeşitlilik, modelin genel performansını test etmek için önemli bir zenginlik sunmaktadır.

3.1.3. humans2

Humans2 veri seti, yazılım mühendisliğinin tekrarlanabilir, doğrulanabilir, çürütülebilir ve/veya geliştirilebilir öngörü modellerini teşvik etmek amacıyla kamuya açık olarak sunulan bir PROMISE Yazılım Mühendisliği Deposu veri kümesidir. Bu veri seti, 14 öznitelikten ve katılımcıların tahminlerine yönelik genel göreli hatayı temsil eden bir sınıf değişkeninden oluşmaktadır. Proje efor süresi açısından değerlendirildiğinde, en yüksek sürenin 9.76, en düşük sürenin ise 0.007 olduğu gözlemlenmektedir. Ayrıca, süre verileri arasında standart sapma değeri 2.30 olarak hesaplanmıştır, bu nedenle veri setindeki sürelerin küçük bir aralıkta yoğunlaştığı ve tahminlerin nispeten daha tutarlı olduğunu kanısına varılmıştır. Bu durum, modelin doğruluğunu test etmek ve karşılaştırmak için önemli bir veri noktası sağlamaktadır.

3.1.4. nasa93

Nasa93 veri seti, 1971 ile 1986 yılları arasında yürütülmüş 93 farklı NASA projesinden oluşmaktadır. Bu veri seti, 16 öznitelik ve adam/saat cinsinden belirtilmiş sınıf değişkeni içermektedir. Proje efor süresi açısından değerlendirildiğinde, en yüksek sürenin 8211 saat, en düşük sürenin ise 8.4 saat olduğu gözlemlenmektedir. Ayrıca, süre verileri arasında standart sapma değeri 1129 saat olarak hesaplanmıştır. Elde edilen yüksek standart sapma, veri setindeki efor sürelerinin geniş bir dağılıma sahip olduğunu ve projelerin önemli ölçüde farklı ölçeklerde yer aldığını göstermektedir. Bu çeşitlilik, farklı büyüklükteki projelerin efor tahminlerini değerlendirmek için önemli bir kaynak sağlamaktadır

3.1.5. usp05

Usp05 veri seti, PROMISE Yazılım Mühendisliği Deposu veri kümelerinin bir elemanıdır. Bu veri seti, 15 öznitelik ve 203 örnekten oluşmaktadır. Usp05 veri setinde, adam/ay oranından efor değeri sınıf değişkeni olarak yer almaktadır. Proje efor süresi açısından değerlendirildiğinde, en yüksek sürenin 400 ay, en düşük sürenin ise 0.5 ay olduğu gözlemlenmektedir. Ayrıca, süre verileri arasında standart sapma değeri 34 ay olarak hesaplanmıştır. Bu hesaplamalar, efor sürelerinin geniş bir dağılım gösterdiğini ve projelerin farklı ölçeklerde olduğunu, ancak veri setindeki sürelerin genel olarak bir miktar tutarlılığa sahip olduğunu işaret etmektedir. Bu özellik, efor tahminleri için modelin doğruluğunu test etme açısından önemli bir rol oynamaktadır.

3.1.6. usp05-ft

Usp05-ft veri seti, PROMISE Yazılım Mühendisliği Deposu veri kümelerinin bir elemanıdır. Bu veri seti, 13 öznitelik ve 76 örnekten oluşmaktadır. Usp05-ft veri setinde, adam/ay oranından efor değeri sınıf değişkeni olarak yer almaktadır. Proje efor süresi açısından değerlendirildiğinde, en yüksek sürenin 40 ay, en düşük sürenin ise 0.5 ay olduğu gözlemlenmektedir. Ayrıca, süre verileri arasında standart sapma değeri 8 ay olarak hesaplanmıştır. Elde edilen değerler, efor sürelerinin daha dar bir aralıkta yoğunlaştığını ve proje efor sürelerinin daha tutarlı olduğunu göstermektedir. Bu özellik, efor tahmin modellerinin doğruluğunu değerlendirmek için önemli bir gösterge sunmaktadır.

3.2. Regresyon Modelleri (Regression Models)

Çalışmada PET için süreğen değerlerin tahmin edilmesi amaçlanmaktadır. Literatürde süreğen değerlerin tahmin edilmesinde regresyon modelleri sıklıkla kullanılmaktadır. Regresyon, bağımlı bir değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modellemek ve tahmin etmek için kullanılan bir istatistiksel yöntem olarak tanımlanabilmektedir. Bu çalışmada regresyon modellerini geliştirmek için RO, KA, DR, YSA, GB ve AdaBoost modelleri kullanılmıştır.

RO, birden fazla karar ağacının oluşturduğu bir topluluk yöntemi olup, her ağacın bağımsız bir şekilde eğitilmesiyle sonuçların birleştirilmesini sağlar. Her ağacın farklı bir veri alt kümesi ve rastgele seçilen öznitelikler üzerinde

eğitilmesiyle elde edilen sonuçlar ortalama değer ile birleştirilir. RO aşırı öğrenmeyi azaltmada etkilidir ve regresyon problemlerinde sıklıkla kullanılmaktadır (Breiman, 2001).

KA, veriyi belirli kurallara göre dallara ayırarak bir hedef değişkeni tahmin etmek için kullanılan hiyerarşik bir yapıdır. KA kullanılarak oluşturulan ağaç, her bir düğümde bir karar kuralı oluşturur ve veri kümelerini bölerek yapraklarda tahmin yapar. Kolay anlaşılabilir ve görselleştirilebilir olması nedeniyle avantajlıdır ancak tek başına kullanıldığında aşırı öğrenmeye eğilimlidir (Quinlan, 1986).

DR, bağımlı bir değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi, bu değişkenler arasındaki doğrusal bağıntıyı modelleyerek tahmin etmeyi amaçlayan regresyon yöntemidir. DR, veri kümesindeki en iyi doğrusal çizgiyi belirlemek için en küçük kareler yöntemini kullanmaktadır. Basit kullanımı ve kolay açıklanabilir olması nedeniyle yaygın bir şekilde kullanılan yöntem özellikle doğrusal ilişkilerde etkili çalışmaktadır (Seber & Lee, 2012).

YSA, biyolojik sinir sistemlerinden ilham alan, katmanlı bir yapıya sahip ve veri arasındaki karmaşık ilişkileri öğrenebilen bir makine öğrenimi yöntemidir. YSA, giriş verilerini farklı katmanlardan geçirerek öğrenir ve her katmanda belirli ağırlıklarla hesaplamalar yapar. Doğrusal olmayan ilişkilerde ve büyük veri setlerinde oldukça başarılıdır ancak modelin eğitimi için yüksek hesaplama gücü gerekmektedir (LeCun vd., 2015).

GB, hataları minimize etmek için karar ağaçları gibi zayıf öğreniciler ekleyerek adım adım bir model oluşturur. Her iterasyonda, önceki modelin hata terimlerini düzeltmek için yeni bir ağaç eğitilir. Hata azaltmada oldukça başarılı olan GB, genellikle hiper parametre optimizasyonu ile güçlü bir performans sergilemektedir. Ancak büyük veri setlerinde daha uzun eğitim süreleri gerektirebilir (Friedman, 2001).

AdaBoost, GB yöntemine benzer şekilde karar ağaçları gibi zayıf öğrenicilerin bir topluluğunu ağırlıklı olarak birleştirerek güçlü bir model oluşturmayı amaçlamaktadır. Her bir model, önceki modelin doğru tahmin edemediği örneklere daha fazla ağırlık vererek eğitilmektedir. AdaBoost, aşırı öğrenmeyi genellikle azaltır ve hızlı bir şekilde çalışır, ancak gürültülü verilere karşı duyarlıdır (Freund & Schapire, 1997).

3.3. SHAP (SHAP)

Makine öğrenimi modellerinin kararlarını açıklamak için kullanılan SHAP, her özelliğin modelin sonucuna olan katkısını adil bir şekilde tahmin etmeyi amaçlamaktadır. SHAP, kooperatif oyun teorisine dayanan Shapley değerlerini kullanmaktadır ve bir özelliğin diğer özelliklerle birlikte modeli nasıl etkilediğini hesaplamaktadır. Shapley değerleri, her özelliğin katkısını hesaplamak için tüm olası alt küme kombinasyonlarını değerlendirerek, adil ve tutarlı bir açıklama sağlamaktadır. SHAP, hem global hem de lokal açıklamalar sunarak, modelin genel davranışını ve bireysel tahminleri anlamayı mümkün kılmaktadır. Özelleştirilebilir ve görselleştirilebilir açıklama araçları sunan SHAP, model-agnostik bir yapıda olmasından dolayı herhangi bir makine öğrenimi algoritmasına uygulanabilmektedir (Lundberg & Lee, 2017).

3.4. Performans Değerlendirme

Çalışmada önerilen modellerin farklı koşullardaki performansını daha sağlıklı bir şekilde ölçmek için tekrarlayan sınama (Repeated Holdout) yöntemi kullanılmıştır. Tekrarlayan sınama, bir veri kümesini eğitim ve test olarak rastgele bölerek model performansını değerlendiren bir çapraz doğrulama yöntemidir. Bu işlem birçok kez tekrarlanır ve her bir tekrarın sonuçlarının ortalaması alınarak modelin genelleme performansı tahmin edilir. Her bir tekrarda farklı bir eğitim-test bölünmesi yapılması, modelin veri dağılımından kaynaklanabilecek varyanslara karşı daha dayanıklı olmasını sağlar. Tekrarlayan sınama, model performansını hem rastgele veri seçimine hem de farklı test senaryolarına göre değerlendirdiği için, sonuçların güvenilirliğini artırmaktadır (Kim, 2009).

Her bir sınamada eğitim ve test olarak ikiye bölünen veri setinde modeller, eğitim veri seti üzerinde eğitilmiş ve model performans sonuçları test veri seti üzerinde hesaplanmıştır. Performans hesabı için ortalama mutlak hata (Mean Absolute Error - MAE), ortalama logaritmik kare hatası (Mean Squared Logarithmic Error - MSLE), belirleme katsayısı (R^2 Score - R) ve ortalama göreli büyüklük hatası (Mean Magnitude Relative Error - MMRE) metrikleri kullanılmıştır. MAE, tahmin edilen değerlerle gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasını ölçmektedir. Duyarlılığı yüksektir ve tüm hatalara eşit ağırlık verir (Willmott & Matsuura, 2005). MSLE, logaritmik dönüştürülmüş tahmin ve gerçek değerler arasındaki farkın karesinin ortalamasını ölçer. Daha küçük değerleri ön plana çıkarır ve orantısal hataları cezalandırır (Amruthnath & Gupta, 2018). R, modelin toplam değişkenliğin ne kadarını açıkladığını ifade eden bir ölçüttür. 1'e yakın değerler daha iyi bir uyum gösterir; 0, modelin tahmin gücünün olmadığını belirtir (Draper & Smith, 1998). MMRE, tahmin edilen değer ile gerçek değer arasındaki farkın, gerçek değere oranının mutlak ortalamasını ölçer. Özellikle oranlara dayalı analizlerde

kullanışlıdır (Kitchenham & Mendes, 2004).

4. Deney Sonuçları (Experiment Results)

Çalışma kapsamında yapılan deneylerin ilk aşamasında önerilen regresyon modelleri Python dilinde var olan sklearn kütüphanesi kullanılarak geliştirilmiştir (*scikit-learn: machine learning in Python*, 2024). Geliştirilen modellerin tamamımın hiper-parametreleri sklearn kütüphanesinin 1.5.0 versiyonundaki varsayılan ayarda bırakılmıştır. Bu değerler geniş veri kümeleri ve farklı problem türleri için genelleştirilmiş ve kabul edilebilir bir performans sağlaması hiper-parametreleri varsayılan değerde bırakma sebebimizdir. Varsayılan ayarlar, hızlı model geliştirme ve test etme aşamalarında verimli bir başlangıç noktası sunarak, daha detaylı ayarlamalara gerek kalmadan güvenilir sonuçlar elde edilmesine olanak tanımaktadır. Model geliştirmeleri tamamlandıktan sonra veri setinden rastgele %10 örnek seçilerek test, kalan örneklerle ise eğitim veri setleri oluşturulmuştur. Model performans ölçümlerinin daha anlamlı olması için farklı veri kümeleri kullanılarak ölçümler yapılması gerekmektedir. Bu bağlamda eğitim ve test veri seti oluşturma süreçleri 50 kez tekrar edilerek 50 farklı eğitim ve test veri setleri oluşturulmuştur. Önerilen modeller her bir eğitim veri seti ile ayrı ayrı eğitilmiş daha sonra eğitim veri setine karşılık gelen test verisi üzerindeki başarı oranları hesaplanmıştır. Tablo 2-7 arasında her bir veri seti için hesaplanan ortalama performans değerleri ve sınamalar arasında performans metrik skorları kullanılarak hesaplanan standart sapma (STD) değerleri gösterilmektedir. Bu tablolarda kalın olarak gösterilen değerler her bir metrik için en yüksek skoru elde eden modeli temsil etmektedir.

Tablo 2. china_original veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the china_original dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	205.17	8.50	0.213	0.154	**0.992**	0.004	0.178	0,178
RO	**156.72**	18.55	0.019	0.023	0.991	**0.001**	0.088	0,088
AB	587.12	42.13	0.493	0.062	0.965	0.006	0.903	0,903
GB	181.46	**3.72**	0.026	0.020	0.990	0.002	0.104	0,104
KA	229.92	22.62	0.028	0.033	0.973	0.012	0.092	0,092
YSA	168.51	15.71	**0.009**	**0.001**	0.991	0.004	**0.079**	**0,079**

Tablo 2'de yer alan sonuçlar incelendiğinde en başarılı modellerin MAE metriğine göre RO, MSLE metriğine göre YSA, R metriğine göre DR ve MMRE metriğine göre YSA olduğu görülmektedir. STD açısından inceleme yapıldığında ise YSA modelinin en başarılı sonuçları aldığı metriklerde, yine en başarılı STD değerlerini elde ettiği görülmektedir. MAE ve R metriklerinde ise en başarılı modellerin sırasıyla RO ve DR olmasına rağmen, MAE_STD ve R_STD değerlerine göre en başarılı modeller sırasıyla GB ve RO olmuştur. Özellikle MAE metriğine göre AB diğer modellere göre oldukça kötü bir performans sergilemiştir. Tüm bu sonuçlar dikkate alındığında ise china_original veri seti için en başarılı modelin YSA olduğu değerlendirmesi yapılmaktadır.

Tablo 3. cocomonasa_v1 veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the cocomonasa_v1 dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	183.03	23.09	1.728	0.882	0.473	0.339	3.741	3.741
RO	**80.07**	24.86	**0.449**	**0.242**	0.654	0.494	0.858	0.858
AB	119.06	19.06	1.591	0.670	0.608	0.747	2.944	2.944
GB	95.62	25.12	0.779	0.408	0.836	**0.151**	1.313	1.313
KA	90.42	25.63	0.788	0.405	**0.986**	0.226	**0.761**	**0.761**
YSA	221.61	**12.24**	2.717	1.400	0.478	0.427	6.012	6.012

Tablo 3'te yer alan sonuçlar incelendiğinde MAE, MSLE, R ve MMRE metriklerine göre en başarılı modellerin sırasıyla RO, RO, KA ve KA olduğu gözlemlenmektedir. STD açısından sonuçlar irdelendiğinde ise MAE, MSLE, R ve MMRE metriklerinde elde edilen sonuçlara göre en başarılı modellerin sırasıyla YSA, RO, GB ve KA olduğu gözlemlenmektedir. Tabloda yer alan değerlere göre cocomonasa_v1 veri seti için özellikle YSA modelinin tüm metriklerde çok kötü sonuçlar elde ettiği kanaatine varılmaktadır. YSA için elde edilen MAE metrik skorunun yüksek olması ve STD değerinin düşük olmasından ötürü, bu modelin tüm veri kümelerinde MAE açısından düşük performans sergilediği çıkarımı yapılmaktadır. YSA modeline benzer bir durum ise DR modelinde bulunmaktadır.

DR modeli de R metriğinde en kötü sonucu elde ederken, diğer üç metrikte ise en kötü ikinci model olmuştur. YSA ve DR modelleri için R metrik skorunun 0'a yakın olması ise bu modellerin cocomonasa_v1 veri setinde oldukça başarısız performans sergilediği yorumunu yapmamıza neden olmuştur. Tüm bu durumlar dikkate alındığında en başarılı iki modelin ise RO ve KA olduğu yorumu yapılmaktadır.

Tablo 4. humans2 veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the humans2 dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	1.22	0.10	0.438	0.089	0.243	0.660	14.673	14.673
RO	1.30	0.14	0.495	0.141	0.572	0.644	14.982	14.982
AB	1.39	**0.07**	0.476	0.108	0.146	0.601	14.752	14.752
GB	**1.02**	0.13	**0.300**	**0.072**	**0.817**	0.201	**5.5870**	**5.5870**
KA	1.05	0.29	0.405	0.152	0.238	**0.058**	18.445	18.445
YSA	1.34	0.14	0.447	0.093	0.549	0.640	8.3301	8.3301

Tablo 4'te yer alan sonuçlar incelendiğinde humans2 veri seti için MAE, MSLE, R ve MMRE metriklerinin tamamında GB modelinin en iyi sonuçları elde ettiği görülmektedir. Bu durumun yanı sıra GB modeli STD_MSLE ve STD_MMRE değerlerine göre en düşük sonucu elde etmiştir. STD_MAE ve STD_R değerlerine göre ise en düşük sonuçlar sırasıyla AB ve KA modelleri ile elde edilmiştir. GB modeli STD_MAE ve STD_R değerlerine göre ilk sırada yer almasa da düşük sonuçlar elde etmiştir. Bu sonuçlar doğrultusunda GB modelinin hem yüksek performans skorları hem de düşük STD değerleri elde ettiği görülmektedir. Bu durum dikkate alındığında GB modelinin humans2 veri seti için en başarılı model olduğu yorumu yapılmaktadır. Tablo 4'te yer alanlar sonuçlardan çıkarılabilecek diğer bir sonuç ise MAE metriğine göre AB, MSLE metriğine göre RO, R metriğine göre AB ve MMRE metriğine göre KA modelinin en kötü sonuçları elde etmiş olmasıdır. Özellikle R metrik sonucunda AB modelinin 0'a yakın bir değer elde etmesi, AB modelinin humans2 veri seti için uygun olmadığı yorumunun yapılmasına neden olmaktadır.

Tablo 5. nasa93 veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the nasa93 dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	181.72	42.15	1.943	1.008	0.388	0.770	2.172	2.172
RO	124.41	30.76	0.417	0.325	0.629	0.162	0.892	0.892
AB	203.42	20.26	1.515	0.807	0.331	0.201	3.124	3.124
GB	**94.12**	28.23	**0.217**	**0.128**	0.654	**0.152**	0.426	0.426
KA	122.13	38.50	0.390	0.170	**0.725**	0.183	**0.416**	**0.416**
YSA	142.17	**16.67**	0.776	0.512	0.571	0.637	1.619	1.619

Tablo 5'de yer alan sonuçlar incelendiğinde MAE, MSLE, R ve MMRE metriklerinde en başarılı modellerin sırasıyla GB, GB, KA ve MMRE olduğu görülmektedir. Bu durumun yanı sıra GB modeli, MMRE metriğine göre en başarılı model olan KA modeline yakın bir sonuç elde ederek en iyi ikinci model olmuştur. R skoruna göre ise GB modeli en iyi üçüncü model olmuştur. STD değerlerine göre sonuçlar incelendiğinde ise STD_MSLE ve STD_R skorlarına göre de en iyi model GB olmuştur.

Tablo 6. usp05 veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the usp05 dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	6.90	0.99	1.346	0.216	0.299	0.673	3.960	3.960
RO	3.07	0.38	0.306	**0.109**	0.415	0.337	0.952	0.952
AB	5.67	**0.27**	1.123	0.167	**0.848**	**0.307**	3.522	3.522
GB	2.89	0.59	0.362	0.134	0.348	0.348	0.989	0.989
KA	**2.27**	0.47	**0.244**	0.112	0.516	0.584	**0.457**	**0.457**
YSA	5.82	0.43	0.899	0.173	0.500	0.680	2.724	2.724

Tüm bu durumlar dikkate alındığında nasa93 veri seti için en uygun modelin GB olduğu yorumu yapılmaktadır. Tablo 5'de yer alan sonuçlardan çıkarılan diğer bir sonuç ise AB ve DR modellerinin tüm metriklerde en kötü modeller olmasıdır. Bu modeller hem STD hem de metrik skorları açısından düşük performans sergilemiş, bu nedenle bu modellerin nasa93 veri seti için uygun olmadığı kanaatine varılmıştır.

Tablo 6'da yer alan sonuçlar incelendiğinde R dışında tüm metriklerde KA modelinin en iyi sonuçları elde ettiği görülmektedir. STD açısından incelendiğinde ise KA modelinin STD_MMRE değerinde en düşük skoru elde ettiği gözlemlenmiştir. R metriğine göre ise en başarılı sonuçlar AB modeli ile elde edilmiştir. STD_MAE, STD_MSLE ve STD_R sonuçlarına göre ise en iyi modeller sırasıyla AB, RO ve AB olmuştur. KA modeli üç metrikte hesaplanan STD değerlerine göre en başarılı model olmasa da özellikle STD_MAE ve STD_MSLE değerlerine göre düşük sonuçlar elde etmiştir. Bu durum dikkate alındığında KA modelinin usp05 veri seti için en başarılı model olduğu ve farklı veri kümelerinde benzer sonuçlar elde ettiği yorumu yapılmaktadır. DR modeli ise tüm metrik değerlerinde en kötü sonucu elde etmiştir. Özellikle R skoruna göre 0'a yakın bir skor elde etmesi DR modelinin zayıf bir yönü olarak değerlendirilmektedir. DR modeli için STD değerlerinin de yüksek olması, modelin farklı veri kümelerinde farklı sonuçlar elde ettiğinin bir göstergesidir. Bu sonuçlar dikkate alındığında DR modeli, usp05 veri seti için hem kötü sonuçları elde etmiştir hem de stabil olmayan bir modeldir. Bu nedenle DR modeli, usp05 veri seti için en uygun olmayan model olarak değerlendirilmektedir.

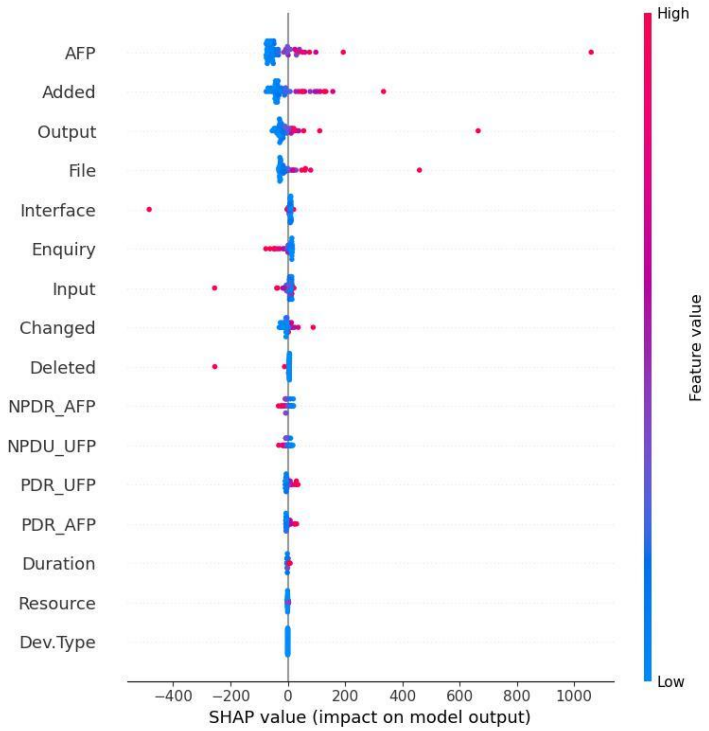
Tablo 7. usp05-ft veri setinde 50 tekrarlayan sınama yöntemi sonucunda elde edilen performans skorları (Performance scores obtained from 50 repeated holdouts on the usp05-ft dataset)

Model Adı	MAE	STD_MAE	MSLE	STD_MSLE	R	STD_R	MMRE	STD_MMRE
DR	1.53	0.42	0.270	0.136	0.722	5.661	1.064	1.06
RO	0.76	0.21	0.099	0.062	0.463	5.594	0.398	0.39
AB	1.44	0.21	0.253	0.068	0.306	5.404	1.165	1.16
GB	0.66	0.24	**0.049**	**0.025**	0.310	**0.259**	0.280	0.28
KA	**0.39**	**0.14**	0.054	0.043	0.716	3.211	**0.184**	**0.18**
YSA	1.45	0.36	0.201	0.137	**0.972**	1.715	0.692	0.69

Tablo 7'de yer alan sonuçlar incelendiğinde, KA modelinin MAE, STD_MAE, MMRE ve STD_MMRE skorlarında, GB modelinin MSLE, STD_MSLE ve STD_R skorlarında, YSA modelinin ise R skorunda en iyi model olduğu gözlemlenmektedir. KA modeli, MSLE ve STD_MSLE skorlarında en iyi ikinci, R ve STD_R skorlarında ise en iyi üçüncü model olmuştur. Bu bilgiler dikkate alındığında usp05-ft veri seti için en iyi modelin KA olduğu yorumu yapılmaktadır. Tablo 7'de yer alan dört metrik sonucunun değerlerine bakıldığında yapılabilecek diğer bir yorum ise DR, YSA ve AB modellerinin diğer modellere göre düşük sonuçlar elde etmesidir.

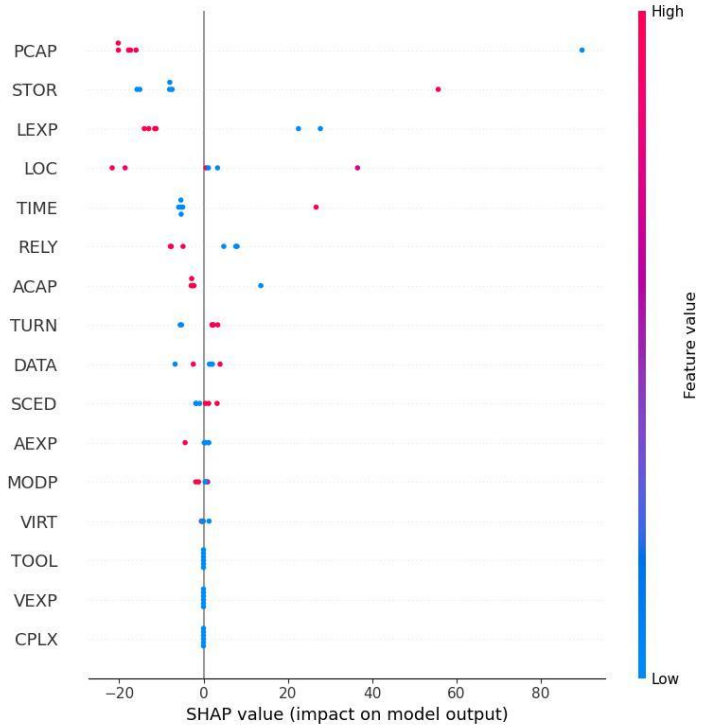
Çalışmada kullanılan regresyon modellerinden bazılarının belirli veri setleri üzerinde düşük performans göstermesinin temel nedenin, bu veri setlerindeki özellik dağılımlarının model varsayımlarıyla uyumsuzluğundan kaynaklandığı değerlendirilmektedir. Özellikle, yüksek boyutlu ve seyrek veri yapısına sahip olan veri setlerinde, DR modelinin lineer varsayımlarının aşırı basitleştirici etki yaparak öğrenememe durumuna yol açmış olabileceği ön görülmektedir. Benzer şekilde ağaç tabanlı modellerin, aşırı gürültülü veya aykırı değerler içeren veri setlerinde aşırı öğrenme eğilimi göstermiş olabileceği düşünülmektedir. YSA modelinin düşük performansının ise, küçük ölçekli veri setlerinde yetersiz eğitim örnekleri nedeniyle parametre optimizasyonunun yetersiz kalmasıyla ilişkilendirilmektedir.

Makine öğrenmesi modellerinin iç işleyişinin insanlar tarafından rahatlıkla anlaşılamaması bu modellerin kara kutu olarak adlandırılmasına neden olmaktadır (Gilpin vd., 2019; Kök, 2024; Lipton, 2017). DR ve KA gibi modeller diğerlerine göre daha anlaşılabilir olsa da bu modellerin de yorumlanabilir olması, makine öğrenmesi tarafından elde edilen sonuçların kullanılabilirliği açısından önemlidir. Bu nedenle son zamanlarda açıklanabilir yapay zekâ yöntemleri sıklıkla kullanılmıştır. Bu çalışmada eğitilen modelleri açıklamak için SHAP yöntemi kullanılmıştır. SHAP yöntemi Python dilinde var olan kütüphane kullanılarak geliştirilmiştir (*SHAP*, 2024). Her bir veri seti için altı farklı modelle 50 tekrarlı sınama ile analizler yapıldığı için 300 farklı model açıklaması yapılabilmesi mümkündür. Fazla sayıda açıklama karmaşıklığa neden olacağı için her bir veri setinde en başarılı sonucu elde eden modelin en başarılı sınama veri seti kullanılarak açıklaması yapılmıştır. Şekil 1-6'da her bir veri seti için yapılmış olan açıklamalara ait çizilen grafikler gösterilmektedir.



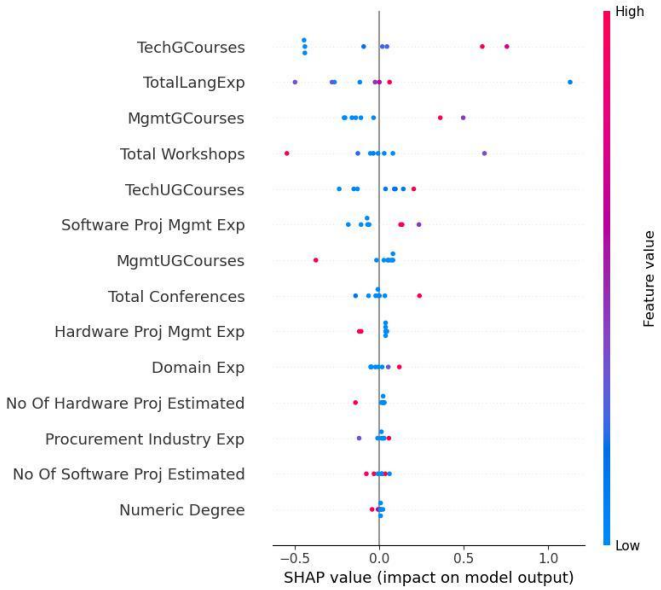
Şekil 1. china_original veri seti için YSA modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the artificial neural network model for the china_original dataset)

Şekil 1'de yer alan sonuçlar china_original veri seti için YSA yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 139.45, 0.0104, 0.994 ve 0.0855 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "AFP", "Added", "Output", "File" ve "Interface" olarak değerlendirilmektedir. "Dev.Type", "Resource", "Duration", "PDR_AFP" ve "PDR_UFP" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.



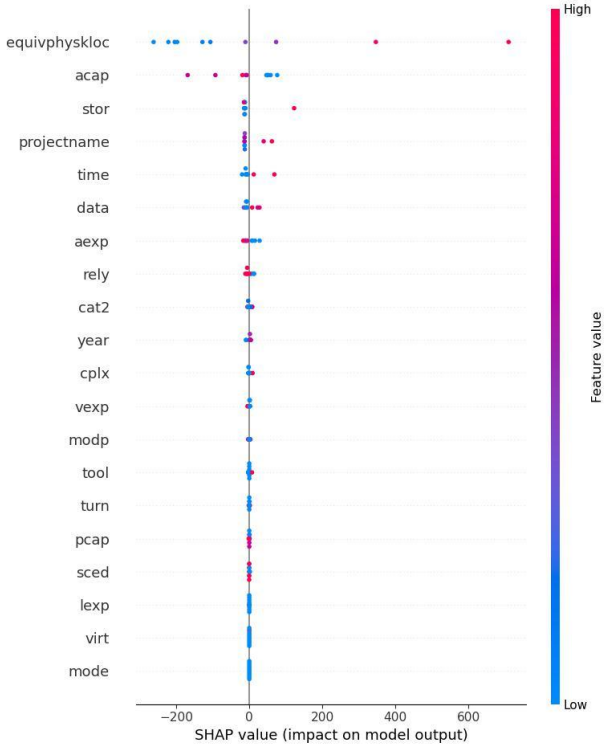
Şekil 2. cocomonasa_v1 veri seti için RO modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the random forest model for the cocomonasa_v1 dataset)

Şekil 2'de yer alan sonuçlar cocomonasa_v1 veri seti için RO yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 35.26, 0.3607, 0.8306 ve 0.801 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "PCAP", "STOR", "LEXP", "LOC" ve "TIME" olarak değerlendirilmektedir. "CPLX", "VEXP", "TOOL", "VIRT" ve "MODP" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.



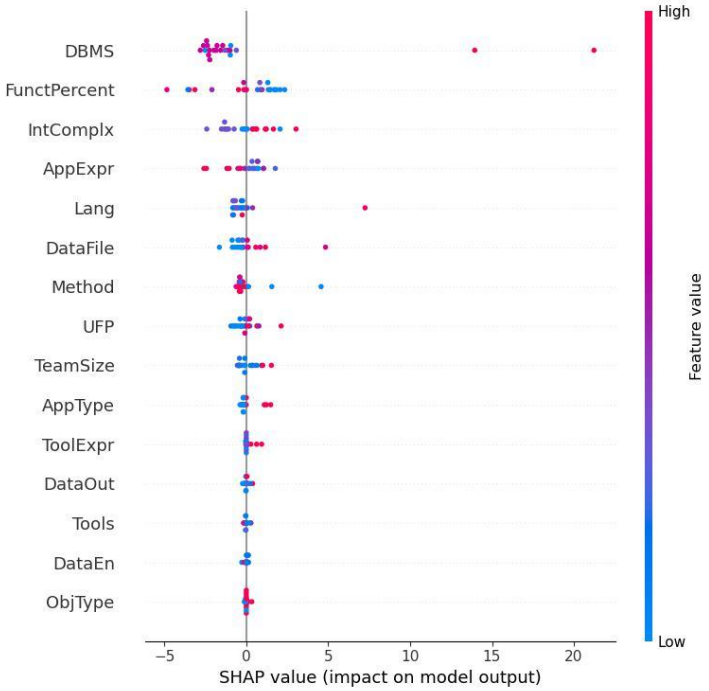
Şekil 3. humans2 veri seti için GB modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the gradient boosting model for the humans2 dataset)

Şekil 3'te yer alan sonuçlar humans2 veri seti için GB yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 0.763, 0.262, 0.875 ve 4.60 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "TechGCourses", "TotalLangExp", "MgmtGCourses", "Total Workshops" ve "TechUGCourses" olarak değerlendirilmektedir. "Numeric Degree", "No Of Software Proj Estimated", "Procurement Industry Exp", "No Of Hardware Proj Estimated" ve "Domain Exp" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.



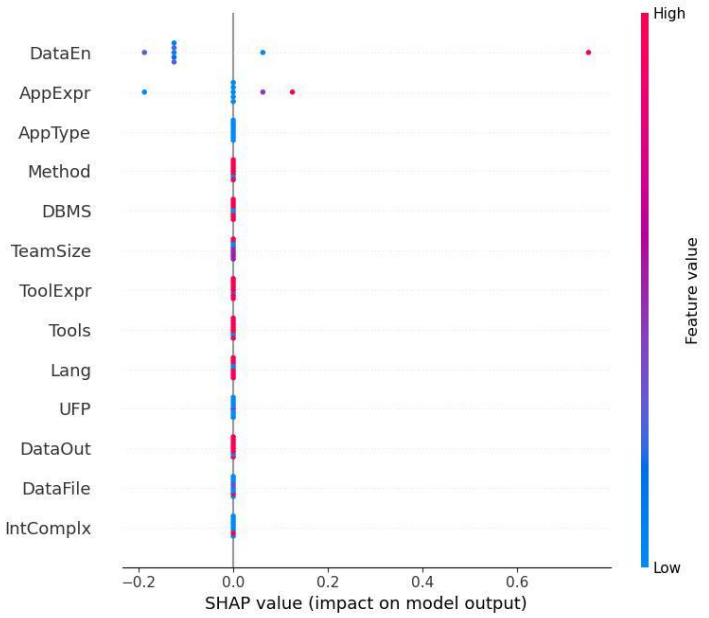
Şekil 4. nasa93 veri seti için GB modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the gradient bossting model for the nasa93 dataset)

Şekil 4'te yer alan sonuçlar nasa93 veri seti için GB yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 51.75, 0.217, 0.693 ve 0.364 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "equivphyskloc", "acap", "stor", "projectname" ve "time" olarak değerlendirilmektedir. "mode", "virt", "lexp", "sced" ve "pcap" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.



Şekil 5. usp05 veri seti için KA modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the decision tree model for the usp05 dataset)

Şekil 5'te yer alan sonuçlar usp05 veri seti için KA yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 1.56, 0.091, 0.528 ve 0.275 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "DBMS", "FunctPercent", "IntComplx", "AppExpr" ve "Lang" olarak değerlendirilmektedir. "ObjType", "DataEn", "Tools", "DataOut" ve "ToolExpr" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.



Şekil 6. usp05-ft veri seti için KA modeli kullanılarak elde edilen model açıklama grafiği
(Model description graph obtained using the decision tree model for the usp05-ft dataset)

Şekil 6'da yer alan sonuçlar usp05-ft veri seti için KA yönteminin en başarılı sınama veri setiyle eğitilen modeli kullanılarak elde edilmiştir. Bu model, MAE, MSLE, R ve MMRE metriklerinde sırasıyla 0.125, 0.0205, 0.811 ve 0.0625 değerlerini elde etmiştir. Şekilde yer alan açıklama sonuçlarına göre model karar aşamasında en etkili olan ilk beş öznitelik "DataEn", "AppExpr", "AppType", "Method" ve "DBMS" olarak değerlendirilmektedir. "IntComplx", "DataFile", "DataOut", "UFP" ve "Lang" öznitelikleri ise model karar alma aşamasında en etkisiz olan beş öznitelik olarak değerlendirilmektedir.

5. Sonuç ve Tartışma (Result and Discussion)

Proje efor tahmini amacıyla, bu çalışmada altı farklı regresyon temelli makine öğrenmesi modeli geliştirilmiştir. Bu modeller, sırasıyla RO, KA, DR, YSA, GB ve AB olarak adlandırılmaktadır. Geliştirilen bu modeller, farklı veri setleri üzerinde test edilmiştir; bu veri setleri china_original, cocomonasa_v1, humans2, nasa93, usp05 ve usp05-ft olarak belirlenmiştir. Modellerin farklı koşullardaki performanslarını karşılaştırmak ve daha doğru bir değerlendirme yapmak amacıyla, tekrarlayan sınama yöntemi kullanılmıştır. Bu yöntem kapsamında, her bir veri setinden rastgele örnekler seçilerek 50 farklı eğitim ve test veri kümesi oluşturulmuş, böylece her bir modelin performansı birden fazla kez ölçülmüştür. Model performanslarının değerlendirilmesinde, MAE, MSLE, R ve MMRE gibi yaygın metrikler kullanılmıştır. Ayrıca, yapılan sınamalar arasındaki değişkenliği daha iyi anlayabilmek için her bir metrik skoruna ait standart sapma değerleri hesaplanmıştır. Yapılan analizler sonucunda, YSA, RO, GB ve KA modelleri, farklı veri setlerinde en başarılı performansı gösteren modeller olarak belirlenmiştir. Özellikle cocomonasa_v1 veri setinde hem RO hem de KA modelleri en başarılı modeller olarak öne çıkarken, diğer veri setlerinde KA modeli iki farklı veri setinde, GB modeli ise iki farklı veri setinde başarı göstermiştir. YSA modeli ise yalnızca bir veri setinde en başarılı model olmuştur. Bu bulgular, KA modelinin proje efor tahmininde genel olarak daha etkili olduğunu ve bu çalışma kapsamında incelenen modeller arasında en başarılı model olarak değerlendirilebileceğini göstermektedir. Bu sonuçlar, KA modelinin özellikle proje efor tahmini alanında güçlü bir performans sergilediğini ve farklı veri setleri üzerinde tutarlı bir başarı sağladığını ortaya koymaktadır.

Piyasada kullanılan sistemlerde bir modelin anlaşılabilir olması büyük bir öneme sahiptir, çünkü anlaşılabilir olma durumu modelin doğruluğu ve güvenilirliği hakkında kullanıcıların bilgi sahibi olmasını sağlamaktadır. Bu bağlamda, çalışmada önerilen modellerin açıklanabilirliğini sağlamak amacıyla SHAP yöntemi kullanılmıştır. SHAP yöntemi, model kararlarının arkasındaki mantığı anlamaya yönelik bir araç olarak, her bir özelliğin modelin çıktısına olan katkısını net bir şekilde ortaya koymaktadır. Veri setlerinde farklı modellerin ve tekrarlayan sınama yaklaşımının kullanılması, bir veri seti için çok sayıda açıklama elde edilmesine olanak sağlamaktadır. Bu çalışmada her bir veri seti için yalnızca bir açıklamaya yer verilmiş ve bu açıklama, en başarılı modelin, en başarılı sınama veri setine dayalı olarak seçilmiştir. Yapılan model açıklamaları sayesinde, her bir veri setindeki hangi özniteliklerin model karar aşamasında daha etkili olduğu belirlenmiştir. Bu analizler sonucunda, bazı özniteliklerin, modelin karar verme sürecinde diğer özniteliklere kıyasla belirgin bir üstünlük sağladığı gözlemlenmiştir. Örneğin, china_original veri setinde "AFP", "Added", "Output" ve "File" öznitelikleri; cocomonasa_v1 veri setinde "PCAP", "STOR" ve "LEXP" öznitelikleri; humans2 veri setinde "TechGcourses", "TotalLangExp", "MgmtGcourses" ve "Total Workshops" öznitelikleri; nasa93 veri setinde "equivphyskloc" ve "acap" öznitelikleri; usp05 veri setinde "DBMS", "FunctPercent" ve "AppExpr" öznitelikleri; ve usp05-ft veri setinde "DataEn" ve "AppExpr" öznitelikleri, diğer özniteliklere kıyasla model karar aşamasında daha büyük bir etkiye sahip olmuştur.

SHAP analizlerinden elde edilen bulgular, yazılım projelerinin planlanması ve kaynak tahsisi süreçlerinde kritik bir rol oynamaktadır. Model kararlarında etkili olan özniteliklerin belirlenmesi, proje yöneticilerine, hangi faktörlere öncelik vermeleri gerektiği konusunda değerli bilgiler sunmaktadır. Örneğin, "AFP", "PCAP" veya "TechGcourses" gibi özniteliklerin yüksek etkiye sahip olduğu projelerde, bu alanlara daha fazla kaynak ayırmanın proje başarısını artırabileceği çıkarımı yapılabilmektedir. Ayrıca, SHAP analizleri, proje risklerini değerlendirme ve potansiyel sorunları önceden belirleme konusunda da yardımcı olabilecektir. Çalışmada sonucunda elde edilen SHAP bulgularının, projelerin daha verimli ve başarılı bir şekilde yönetilmesine imkân sağlayacağı düşünülmektedir. Makine öğrenmesi modellerinin sonuçları incelendiğinde YSA modelinin sadece china_original veri setinde en iyi sonucu; cocomonasa_v1 veri setinde en kötü sonucu; humans2, usp05 ve usp05-ft veri setlerinde en kötü ikinci sonucu; nasa93 veri setinde ise en kötü üçüncü sonucu elde ettiği görülmektedir. Veri setlerinden china_original veri setinin en fazla, cocomonasa_v1 veri setinin ise en az örnek sayısına sahip olması, YSA modelinin eğitiminde örnek sayısının artmasının modelin performansını olumlu etkilediği yorumunu yapmamıza neden olmaktadır. KA modeli ise, china_original hariç tüm veri setlerinde ya en iyi ya da en iyi ikinci model olmuştur. Bu sonuç doğrultusunda KA modelinin örnek ve öznitelik sayısının düşük olduğu veri setlerinde iyi sonuçlar verdiği çıkarımı yapılmıştır. Sonuçlar genel anlamda incelendiğinde ise, yüksek performans sergileyen modellerde standart sapma değerinin de düşük olduğu görülmektedir. Bu durum dikkate alındığında model performansı artıkça modelin daha tutarlı olduğu çıkarımı da yapılmaktadır.

Çalışmada kapsamında yapılmış olan analizler sayesinde, tasarlanacak bir proje efor tahmin sisteminde hangi

regresyon modelinin kullanımının daha uygun olduğu konusunda yorum yapabilmek mümkündür. Bu durumun yanı sıra yapılan açıklamalar sayesinde hangi özelliklerin proje eforunu sağlıklı bir şekilde tahmin etmede daha etkin olduğu da çalışma kapsamında yapılan analizler sonucunda anlaşılabilmektedir. Gelecek çalışmalarda veri seti ve model sayısının artırılarak analizlerin tekrar edilmesi, daha kapsamlı bir çalışmanın yapılmasına katkı sunacaktır. Gelecekte çalışmayla ilgili yapılabilecek diğer bir analiz ise kullanılan modellerde hiperparametre optimizasyonların yapılması olarak düşünülmektedir. Gelecek çalışmalarda LIME, Morris Sensitivity Analysis ve Explainable Boosting Machine gibi farklı açıklamalı yapay zekâ yöntemlerini kullanılarak hem yerel hem de global açıklamaların yapılması, yapılan açıklamaların ise uygun metriklerle karşılaştırılarak daha detaylı bir model analizinin yapılması mümkündür.

Teşekkür (Acknowledgement)

Bu araştırmada yer alan tüm/kısmi nümerik hesaplamalar TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nde (TRUBA kaynaklarında) gerçekleştirilmiştir. Bu çalışma Dr. Öğretim Üyesi Yasin GÖRMEZ danışmanlığında yürütülen Esma Nur KAYA'ya ait "Proje Efor Tahmini İçin Makine Öğrenmesi Modellerinin Geliştirilmesi ve SHAP Yöntemi Kullanılarak Açıklanması" başlıklı tez çalışması kapsamında yapılan çalışma sonuçları kullanılarak üretilmiştir. Çalışma sonuçları ilgili tezde kullanılacaktır. Çalışmanın sonuçlarının bir kısmı ise (SHAP açıklamaları hariç) "Ege 12th International Conference on Applied Sciences" başlıklı konferansta özet metin olarak kabul edilmiştir.

Çıkar Çatışması (Conflict of Interest)

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir. No conflict of interest was declared by the authors.

Kaynaklar (References)

Amruthnath, N., & Gupta, T. (2018). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. 2018 5th International Conference on Industrial Engineering and Applications (ICIEA), 355-361. https://doi.org/10.1109/IEA.2018.8387124

Azzeh, M., & Nassif, A. B. (2016). A hybrid model for estimating software project effort from Use Case Points. Applied Soft Computing, 49, 981-989. https://doi.org/10.1016/j.asoc.2016.05.008

BaniMustafa, A. (2018). Predicting Software Effort Estimation Using Machine Learning Techniques. 2018 8th International Conference on Computer Science and Information Technology (CSIT), 249-256. https://doi.org/10.1109/CSIT.2018.8486222

Braga, P. L., Oliveira, A. L. I., Ribeiro, G. H. T., & Meira, S. R. L. (2007). Bagging Predictors for Estimation of Software Project Effort. 2007 International Joint Conference on Neural Networks, 1595-1600. https://doi.org/10.1109/IJCNN.2007.4371196

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. Journal of Systems and Software, 127, 109-119. https://doi.org/10.1016/j.jss.2017.01.027

Draper, N. R., & Smith, H. (1998). Applied Regression Analysis. John Wiley & Sons.

Effor Estimation Datasets. (2024). GitHub. https://github.com/danrodgar/DASE/tree/master/datasets/effortEstimation

Elish, M. O. (2009). Improved estimation of software project effort using multiple additive regression trees. Expert Systems with Applications, 36(7), 10774-10778. https://doi.org/10.1016/j.eswa.2009.02.013

Erasmus, I. P., & Daneva, M. (2013). ERP Effort Estimation Based on Expert Judgments. 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, 104-109. https://doi.org/10.1109/IWSM-Mensura.2013.25

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189-1232.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining Explanations: An Overview of Interpretability of Machine Learning (arXiv:1806.00069). arXiv. https://doi.org/10.48550/arXiv.1806.00069

Hameed, S., Elsheikh, Y., & Azzeh, M. (2023). An optimized case-based software project effort estimation using genetic algorithm. Information and Software Technology, 153, 107088. https://doi.org/10.1016/j.infsof.2022.107088

Haris, M., Chua, F.-F., & Lim, A. H.-L. (2023). An Ensemble-Based Framework to Estimate Software Project Effort. 2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS), 47-52. https://doi.org/10.1109/ICSECS58457.2023.10256337

Hosni, M. (2024). Comparative Analysis of Single and Ensemble Support Vector Regression Methods for Software Development Effort Estimation: Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 509-516. https://doi.org/10.5220/0013072300003838

Jorgensen, M. (2005). Practical guidelines for expert-judgment-based software effort estimation. IEEE Software, 22(3), 57-63. IEEE Software. https://doi.org/10.1109/MS.2005.73

Kassaymeh, S., Alweshah, M., Al-Betar, M. A., Hammouri, A. I., & Al-Ma'aitah, M. A. (2024). Software effort estimation modeling and fully connected artificial neural network optimization using soft computing techniques. Cluster Computing, 27(1), 737-760. https://doi.org/10.1007/s10586-023-03979-y

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis, 53(11), 3735-3745. https://doi.org/10.1016/j.csda.2009.04.009

Kitchenham, B., & Mendes, E. (2004). Software productivity measurement using multiple size measures. IEEE Transactions on Software Engineering, 30(12), 1023-1035. IEEE Transactions on Software Engineering. https://doi.org/10.1109/TSE.2004.104

Kök, İ. (2024). Açıklanabilir Yapay Zekaya Dayalı Müşteri Kaybı Analizi ve Elde Tutma Önerisi. Mühendislik Bilimleri ve Araştırmaları Dergisi, 6(1), Article 1. https://doi.org/10.46387/bjesr.1344414

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

Lipton, Z. C. (2017). The Mythos of Model Interpretability (arXiv:1606.03490). arXiv. https://doi.org/10.48550/arXiv.1606.03490

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Molnar, C. (2020). Interpretable Machine Learning. Lulu.com.

Mukherjee, S., & Malu, R. K. (2014). Optimization of project effort estimate using neural network. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 406-410. https://doi.org/10.1109/ICACCCT.2014.7019474

Mustafa, E. I., & Osman, R. (2024). A random forest model for early-stage software effort estimation for the SEERA dataset. Information and Software Technology, 169, 107413. https://doi.org/10.1016/j.infsof.2024.107413

Özgür, A. S., Tarhan, Ç., Komesli, M., & Tecim, V. (2023). Yapay Zeka Teknikleri Kullanılarak Proje Üretim Sistemlerinin Tasarımı ve Geliştirilmesi. Journal of Information Systems and Management Research, 5(1), Article 1. https://doi.org/10.59940/jismar.1214440

Plumb, G., Molitor, D., & Talwalkar, A. S. (2018). Model Agnostic Supervised Local Explanations. Advances in Neural Information Processing Systems, 31. https://proceedings.neurips.cc/paper_files/paper/2018/hash/b495ce63ede0f4efc9eec62cb947c162-Abstract.html

Pospieszny, P., Czarnacka-Chrobot, B., & Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. Journal of Systems and Software, 137, 184-196. https://doi.org/10.1016/j.jss.2017.11.066

Qi, F., Jing, X.-Y., Zhu, X., Xie, X., Xu, B., & Ying, S. (2017). Software effort estimation based on open source projects: Case study of Github. Information and Software Technology, 92, 145-157. https://doi.org/10.1016/j.infsof.2017.07.015

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106. https://doi.org/10.1007/BF00116251

Ritu, & Bhambri, P. (2023). Software Effort Estimation with Machine Learning – A Systematic Literature Review. Içinde Agile Software Development (ss. 291-308). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119896838.ch15

scikit-learn: Machine learning in Python. (2024). https://scikit-learn.org/stable/

Seber, G. A. F., & Lee, A. J. (2012). Linear Regression Analysis. John Wiley & Sons.

SHAP. (2024). https://shap.readthedocs.io/en/latest/

Sharma, S., & Vijayvargiya, S. (2021). Applying Soft Computing Techniques for Software Project Effort Estimation Modelling. Içinde V. Nath & J. K. Mandal (Ed.), Nanoelectronics, Circuits and Communication Systems (ss. 211-227). Springer. https://doi.org/10.1007/978-981-15-7486-3_21

Sharma, S., & Vijayvargiya, S. (2022). Modeling of software project effort estimation: A comparative performance evaluation of optimized soft computing-based methods. International Journal of Information Technology, 14(5), 2487-2496. https://doi.org/10.1007/s41870-022-00962-5

Sharma, S., & Vijayvargiya, S. (2023). An Optimized Neuro-Fuzzy Network for Software Project Effort Estimation. IETE Journal of Research, 69(10), 6855-6866. https://doi.org/10.1080/03772063.2022.2027282

Shepperd, M., Schofield, C., & Kitchenham, B. (1996). Effort estimation using analogy. Proceedings of IEEE 18th International Conference on Software Engineering, 170-178. https://doi.org/10.1109/ICSE.1996.493413

Şengüneş, B., & Öztürk, N. (2023). An Artificial Neural Network Model for Project Effort Estimation. Systems, 11(2), Article 2. https://doi.org/10.3390/systems11020091

Tsunoda, M., Monden, A., Keung, J., & Matsumoto, K. (2012). Incorporating Expert Judgment into Regression Models of Software Effort Estimation. 2012 19th Asia-Pacific Software Engineering Conference, 1, 374-379. https://doi.org/10.1109/APSEC.2012.58

Tuncer, Y. (2024). Artificial Intelligence Based Risk Analsis in Project Management [M.Eng.]. https://www.proquest.com/docview/3143984193/abstract/4ABE168365614041PQ/1

Walkerden, F., & Jeffery, R. (1999). An Empirical Study of Analogy-based Software Effort Estimation. Empirical Software Engineering, 4(2), 135-158. https://doi.org/10.1023/A:1009872202035

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research, 30(1), 79-82. https://doi.org/10.3354/cr030079