



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Enerji Verimli Binaların Isıtma ve Soğutma Yüklerinin Düzenli Modeller ile Tahmin Edilmesi

 Burak DİLBER^{a,*},  A. Fırat ÖZDEMİR^a

^a İstatistik Bölümü, Fen Bilimleri Enstitüsü, Dokuz Eylül Üniversitesi, İzmir, TÜRKİYE

* Sorumlu yazarın e-posta adresi: burakdilber91@gmail.com

DOI: 10.29130/dubited.939933

ÖZ

Fosil yakıtların azalması ile birlikte enerji verimliliği kavramı gün geçtikçe önem kazanmaktadır. Enerjinin büyük bir bölümü binaların yapım ve kullanımında tüketildiğinden binalardaki enerji tasarrufunu sağlamak için ısıtma ve soğutma yüklerinin hesaplanması gerekir. Bu yükler enerji simülasyon yazılımları kullanılarak hesaplanmaktadır ancak bu süreç çok zaman alabilir ve kullanıcı deneyimi gerektirir. Makine öğrenmesi algoritmaları ısıtma ve soğutma yüklerinin tahminlenmesinde hızlı ve güvenilir bir alternatif haline gelmiştir. Bu çalışmada, binaların ısıtma ve soğutma yükleri Doğrusal Regresyon, K – En Yakın Komşuluk, Destek Vektör Makineleri, Karar Ağaçları, Rastgele Orman, Uç Gradyan Artırma ve Yapay Sinir Ağları kullanılarak tahminlenmiştir. Analizler R İstatistiksel Programlama Dili'nin güncel çalışma alanlarından olan düzenli modeller ("tidymodels") ile gerçekleştirilmiştir. Düzenli modeller "tidyverse" ilkeleri ile makine öğrenmesi uygulamaları yapmak için geliştirilmiş, hızlı ve pratik çözümler sunan paketler koleksiyonudur. Yapılan çalışma ile elde edilen sonuçlara göre tahmin doğruluğu açısından en başarılı algoritma Uç Gradyan Artırma Algoritması olmuştur. Binaların çeşitli özelliklerine ait değerler girilerek ısıtma ve soğutma yüklerini yüksek hassasiyet ile tahmin etmek ve hızlı sonuç almak için düzenli modelleme yaklaşımı ile Uç Gradyan Artırma Algoritması'nın kullanılması önerilmektedir.

Anahtar Kelimeler: Enerji verimliliği, R, Düzenli modeller, Makine öğrenmesi, Uç gradyan artırma

Estimating Heating and Cooling Loads of Energy Efficient Buildings with Tidy Models

ABSTRACT

With the decrease in fossil fuels, the concept of energy efficiency gains importance day by day. Since most of the energy is consumed in the construction and use of buildings, heating and cooling loads must be calculated in order to save energy in buildings. These loads are calculated using energy simulation software, but this process can take a lot of time and requires user experience. Machine learning algorithms have become a fast and reliable alternative for predicting heating and cooling loads. In this study, heating and cooling loads of buildings are estimated using Linear Regression, K - Nearest Neighbors, Support Vector Machines, Decision Trees, Random Forest, Extreme Gradient Boosting and Artificial Neural Networks. Analyzes are carried out with tidy models ("tidymodels"), which is one of the current study areas of R Statistical Programming Language. Tidy models are a collection of packages that are developed for machine learning applications with the principles of "tidyverse", offering fast and practical solutions. According to the results obtained from the study, the most successful algorithm in terms of prediction success has been the Extreme Gradient Boosting Algorithm. With this model, the use of a fast-running method that can predict heating and cooling loads with high precision by entering values for various properties of buildings is proposed.

Keywords: Energy efficiency, R, Tidy models, Machine learning, Extreme gradient boosting

I. GİRİŞ

Küresel iklim değişikliği ve enerji güvenliği ile ilgili endişelerin artmasıyla birlikte “enerji verimliliği” kavramı gün geçtikçe önemini artırmaktadır. Enerji kaynaklarının önemli bir bölümünü oluşturan fosil yakıtların azalması ve bu yakıtların dağıtım sırasında ortaya çıkan çevresel sorunlar nedeniyle enerji verimliliğine daha fazla ihtiyaç duyulmaktadır. Fosil enerji kaynaklarının tükenmesi ile birlikte çevreye daha az zarar veren alternatif enerji kaynaklarının kullanılması ve enerji tüketimini en aza indirmek açısından enerji verimliliği uygulamaları bir çok ülkede teşvik edilmektedir.

Binaların yapım ve kullanımında çok enerji tüketilmektedir. Bu yüzden, tüm dünyada inşaat sektöründeki enerji tasarrufu potansiyeli çok yüksektir [1]. Günümüzde binalar için enerji verimliliği ısıtma ve soğutma yüklerinin azaltılmasıyla sağlanmaktadır. Enerji verimli bina tasarımı söz konusu olduğunda, konforlu iç hava koşullarını korumak için ısıtma ve soğutma yüklerinin hesaplanması gerekir [2]. Gerekli ısıtma ve soğutma yüklerini hesaplamak için mimarlar ve bina tasarımcıları, binanın ve çevrenin özellikleri, iklim ve kullanım amacı hakkında bilgilere ihtiyaç duymaktadırlar.

Bina enerji simülasyon araçları enerji verimli binaların tasarımını ve işletimini kolaylaştırmak ve bina enerji tüketimini tahmin etmek için yaygın olarak kullanılmaktadır. Bu araçlardan bulunan simülasyon sonuçlarının gerçek ölçümleri çoğu zaman doğru bir şekilde yansıtılabileceği gösterilmiştir. Ancak bu süreç çok zaman alabilir ve kullanıcı deneyimi gerektirir. Ayrıca yapılan tahminlerin doğruluğu farklı yazılımlar arasında değişiklik gösterebilmektedir [3]. Bu nedenle bir çok araştırmacı enerji verimli binalar için ısıtma ve soğutma yüklerini tahminlerken makine öğrenmesi araçlarından faydalanmaktadır. Çünkü, binalar için gerekli veri setleri mevcut olduğunda bu araçları kullanmak daha kolay ve hızlı çözümler sunmaktadır. Bu veri setleri yeterince eğitildikten sonra binaların bazı özellikleri değiştirilerek cevaplar elde etmek son derece hızlıdır.

Literatürde, binaların ısıtma ve soğutma yükünü tahmin etmek için bir çok çalışma mevcuttur. Tsanas ve Xifara, Ecotect simülasyon yazılımı ile elde edilen veri setini kullanarak Yeniden Ağırlıklandırılmış En Küçük Kareler ve Rastgele Orman Algoritmaları'nın performanslarını incelemiştir [2]. Chou ve Bui binaların enerji performansı için Destek Vektör Regresyonu, Yapay Sinir Ağları, Ki – Kare Otomatik Etkileşim Dedektörü, Genel Doğrusal Regresyon ve Topluluk Çıkarım Modeli dahil olmak üzere çeşitli veri madenciliği tekniklerini kullanarak ısıtma ve soğutma yüklerini tahmin etmişlerdir. Çalışmalarında SPSS Clementine programını kullanmış, ısıtma yükü için Destek Vektör Regresyonu, soğutma yükü için Destek Vektör Regresyon ve Yapay Sinir Ağlarının birlikte kullanıldığı Topluluk Çıkarım Modelini önermişlerdir [4]. Peker vd. regresyon tabanlı makine öğrenmesi algoritmalarını kullanarak yapıların ısıtma ve soğutma yüklerini tahminlemişlerdir. Çalışmalarında tahmin başarısı açısından en yakın sonucu Rastgele Orman Regresyonu Algoritması ile elde etmişler ve yaptıkları analiz için Orange veri madenciliği programını kullanmışlardır [5]. Gümüşçü vd. tarafından yapılan çalışmada binaların ısıtma ve soğutma yükü sınıfları K – En Yakın Komşuluk, Destek Vektör Makineleri ve Karar Ağaçları yöntemleri kullanılarak tahminlenmiştir [6]. Seyedzadeh vd. EnergyPlus ve Ecotect simülasyon yazılımları ile elde edilen iki farklı simüle edilmiş veri seti üzerinde çalışmışlardır. Python programlama dili ile yaptıkları çalışmada yapıların ısıtma ve soğutma yüklerinin tahmininde popüler makine öğrenimi modellerinin doğruluğunu araştırmış ve sonuçları karşılaştırmışlardır [7]. Roy vd. Derin Sinir Ağları, Gradyan Destekli Makine, Gauss Süreci Regresyonu ve Minimax Olasılık Makine Regresyonu Algoritmalarını kullanarak binaların ısıtma ve soğutma yüklerini tahminlemek için modeller önermişlerdir. Yaptıkları çalışmada R programlama dili ve MATLAB programını kullanmışlardır [8]. Moradzadeh vd. Çok Katmanlı Yapay Sinir Ağları ve Destek Vektör Regresyon Algoritmalarını kullanarak binaların ısıtma ve soğutma yüklerini tahminlemişlerdir [9].

Literatürde yer alan uygulamalara bakıldığında makine öğrenmesi algoritmaları kullanılarak çeşitli çalışmalar yapıldığı görülmektedir. Bu çalışmada düzenli modeller ile enerji verimli binaların ısıtma ve soğutma yüklerini tahminlemek için makine öğrenmesi uygulaması yapılmıştır. Düzenli modeller R programlama dilinin güncel çalışma alanlarından biridir. Çalışmanın 2. bölümünde düzenli modeller

tanıtılmıştır. 3. Bölümde, yapılan makine öğrenmesi uygulaması R kodları ile açıklanmış ve son bölümde de elde edilen sonuçlar yorumlanarak öneriler yapılmıştır.

II. DÜZENLİ MODELLER

Veri manipülasyonu, hesaplama, grafik gösterimi gibi çeşitli uygulamalar için kullanılan R programlama dili yeni geliştirilen veri çözümlene yöntemlerine ait uygulamaların da hızlı bir biçimde yapılmasını sağlar [10]. Ayrıca R’de hazır paketler kullanılarak karmaşık yöntemlere ait işlemler tek satırlık bir kod ile çalıştırılabilir.

R’de yer alan bazı standart paketler ile veri bilimi alanında çalışmalar yapmak mümkündür. Ancak, bu paketlerde yer alan fonksiyonlardaki söz dizimi farklılıkları kod yazmayı zahmetli ve zaman alan bir süreç haline getirmiştir. Kodların okunabilirliğini arttırmak ve daha anlaşılır söz dizimi oluşturmak için Bache ve Wickham tarafından pipe (%>%) operatörü önerilmiştir. Bu operatör fonksiyonların birleşimi için kullanılan bir araçtır ve büyük problemleri küçük parçalara ayırarak çözmeyi kolaylaştırır [11].

Hadley Wickham vd. 2019 yılında yaptıkları çalışmada sözdizimi farklılıklarından kaynaklanan zorlukların aşılması için tidyverse paketler koleksiyonunu önermiştir [12]. Veri bilimi alanında çeşitli uygulamaları daha hızlı ve düzenli bir şekilde yapabilmek için bu paketler koleksiyonu kullanılabilir. Tidyverse istatistiksel modelleme için gerekli fonksiyonları içermez ancak veri bilimi için önemlidir. Tidyverse paketi ile birlikte R istatistiksel programlama diline bir “düzenlilik” kavramı getirilmiştir.

Tidyverse paketler koleksiyonundaki istatistiksel modelleme ve makine öğrenimi uygulamaları eksiğini gidermek için Max Kuhn ve Hadley Wickham tarafından tidymodels paketler koleksiyonu önerilmiştir. Tidymodels paketler koleksiyonunu kullanarak tidyverse ilkeleri ile modelleme ve makine öğrenimi uygulaması yapmak mümkündür [13]. Böylece, R’de düzenli modeller oluşturabilmek için kullanışlı ve daha pratik çözümler sunulmuştur. Tidymodels paketler koleksiyonunda yer alan paketlerdeki fonksiyonlar birbirleriyle etkileşimli bir şekilde çalışarak modeller oluşturmamıza olanak tanır.

Tidymodels paketler koleksiyonu bir veri bilimi projesi için gerekli olan veri ön işleme, yeniden örnekleme, modelleme ve performans ölçümleri gibi çeşitli aşamaları içeren paketlerden oluşmaktadır. Bu paketler sırasıyla aşağıda belirtilmiştir:

recipes: Veri ön işleme, veri manipülasyonu ve analizi gibi adımlardan oluşan pakettir. Adımlar için istatistiksel parametreler bir başlangıç veri setinden tahmin edilebilir ve daha sonra diğer veri setlerine uygulanabilir [14]. Veri dönüştürme, veri normalleştirme ve kayıp gözlem giderme gibi çeşitli veri ön işleme aşamaları bu pakette yer alan fonksiyonlar kullanılarak yapılabilir.

rsample: Farklı türlerdeki yeniden örnekleme nesnelere oluşturmak ve özetlemek için fonksiyonlar içeren pakettir [15]. Veri setini eğitim verisi ve test verisi olarak ayırmak ve çapraz doğrulama setleri oluşturmak için çeşitli yeniden örnekleme yöntemleri kullanılabilir.

broom: İstatistiksel nesnelere hakkında önemli bilgileri içerir. Bu paket, sonuçları raporlamayı, grafikler oluşturmayı ve aynı anda çok sayıda modelle tutarlı bir şekilde çalışmayı kolaylaştırır [16]. Bu pakette temel olarak kullanılan 3 fonksiyon bulunmaktadır. *tidy()*, regresyon katsayıları gibi model bileşenleri hakkında bilgileri özetler. *glance()*, Akaike Bilgi Kriteri ve Bayes Bilgi Kriteri gibi uyum iyiliği ölçümleri ile model hakkında bilgi verir. *augment()*, bir veri setinin tahminlenen değerleri ve etkin gözlemleri gibi her bir gözlem için bilgi verir.

parsnip: Farklı fonksiyonlar ve hesaplama motorları boyunca farklı argüman adlarını hatırlamak zorunda kalmadan modellerin belirlenmesine olanak sağlamak için bir arayüz sağlanmıştır [17]. Popüler makine öğrenmesi algoritmalarının kullanımı için fonksiyonlar içerir.

workflows: *recipes* paketi ile oluşturulan veri ön işleme aşamaları ve *parsnip* ile oluşturulan model birleştirilerek modelleme sürecini kolaylaştırır [18].

tune: Modellerdeki ön işleme yöntemlerinde ve işlem sonrası adımlarda hiperparametrelerin makul değerlerini bulmak için kullanılacak fonksiyonları içerir [19].

dials: Bir çok model için verilerden doğrudan tahmin edilemeyen parametreleri içerir. Bu pakette yer alan fonksiyonlar bu tür parametrelerin değerlerini oluşturmak, simüle etmek veya doğrulamak için kullanılabilir [20].

yardstick: Karmaşıklık matrisleri, sınıf olasılık eğrileri ve regresyon metrikleri gibi modelin bir veri setine ne kadar iyi uyduğunu ölçmek için düzenli fonksiyonlar sağlar [21].

Düzenli modeller ile makine öğrenme algoritmaları uygulanırken çeşitli motorlar kullanılmaktadır. Örneğin Doğrusal Regresyon için *lm()* fonksiyonu kullanılır. Düzenli modellerde bu fonksiyon kullanılacaksa motor olarak belirtilmesi gerekir. Doğrusal regresyon için düzenli modeller kullanılarak model oluşturulurken yazılacak kodlar

```
linear_reg() %>%  
  set_engine("lm")  
## Linear Regression Model Specification (regression)  
##  
## Computational engine: lm
```

şeklinde olabilir. Ayrıca; farklı paketleri kullanmak için *set_engine()* fonksiyonunda bu paketler belirtilmelidir. Doğrusal regresyon için kullanılacak motorlar; *glmnet*, *stan*, *spark* ve *keras* olarak söylenebilir.

Bazı makine öğrenmesi algoritmaları hem regresyon hem de sınıflandırma tabanlı olarak kullanılabilir. Bu algoritmalar kullanılarak modelleme yapılırken regresyon ya da sınıflandırma olarak belirtilmesi gerekir. Bunun için *set_mode()* fonksiyonu kullanılmaktadır. Örneğin; K – En Yakın Komşu Algoritması için *kknn* paketi kullanılarak modeller oluştururken yazılacak kodlar

```
nearest_neighbor() %>%  
  set_engine("kknn") %>%  
  set_mode("regression")  
## K-Nearest Neighbor Model Specification (regression)  
##  
## Computational engine: kknn
```

şeklinde olabilir. Sınıflandırma tabanlı çalışmalar için *set_mode()* fonksiyonunda *regression* yerine *classification* yazılmalıdır.

Diğer algoritmalar için düzenli modellerde kullanılacak çeşitli motorlar vardır. Artırılmış Ağaçlar için *xgboost*, *C5.0* ve *spark* motorları kullanılabilir. Uç Gradyan Artırma için bu motor *xgboost* olarak belirlenmelidir. Karar Ağaçları Algoritması için kullanılacak motorlar; *rpart*, *C5.0* ve *spark* olarak söylenebilir. Lojistik Regresyon için *glm*, *glmnet*, *stan*, *spark* ve *keras* motorları, Rastgele Orman Algoritması için *ranger*, *randomForest* ve *spark* motorları, Destek Vektör Makineleri için *kernlab* ve *liquidSVM* motorları ve Yapay Sinir Ağları için *nnet* ve *keras* motorları kullanılmaktadır.

Yapılan çalışmada düzenli modeller ile sekiz bağımsız değişken ve iki bağımlı değişkenden oluşan bir veri seti üzerinde binaların ısıtma ve soğutma yüklerini tahmin edebilen modeller oluşturulmuştur. Kullanılan yöntemler; Doğrusal Regresyon (DR), K – En Yakın Komşuluk (KEYK), Destek Vektör Makineleri (DVM), Karar Ağaçları (KA), Rastgele Orman (RO), Uç Gradyan Artırma (UGA) ve Yapay Sinir Ağları (YSA)'dır.

III. UYGULAMA

Bu uygulamada Athanasios Tsanas ve Angeliki Xifara isimli arařtırmacıların sundukları veri seti kullanılmıřtır [22]. 768 satır veri içeren bu veri seti 8 bağımsız deęiřken (X_1, \dots, X_8) ve 2 bağımlı deęiřkenden (Y_1, Y_2) oluřmaktadır. Her bir gözlem ayrı bir bina içindir. Gerçek deęerli iki bağımlı deęiřkenin tahmin edilmesi amaçlanmaktadır. Veri setindeki deęiřkenler sırasıyla ařađıda listelenmiřtir.

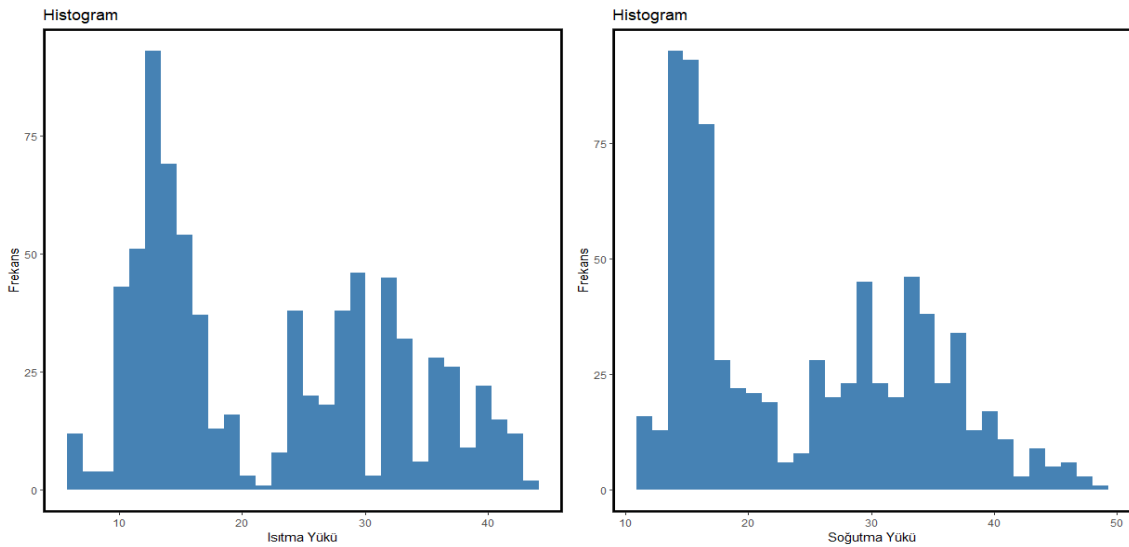
1. Nispi Yoęunluk (X_1)
2. Yüzey Alanı (X_2)
3. Duvar Alanı (X_3)
4. Çatı Alanı (X_4)
5. Toplam Yükseklik (X_5)
6. Yönlendirme (X_6)
7. Cam Alanı (X_7)
8. Cam Alanı Dağılımı (X_8)
9. Isıtma Yüğü (Y_1)
10. Soęutma Yüğü (Y_2)

Veri setinde yer alan deęiřkenlere ait tanımlayıcı istatistikler Tablo 1’de gösterilmektedir.

Tablo 1. Deęiřkenler için tanımlayıcı istatistikler

Deęiřkenler	Minimum	1. Çeyreklik	Median	Mean	3. Çeyreklik	Maksimum
X1	0.62	0.6825	0.75	0.7642	0.83	0.98
X2	514.5	606.4	673.8	671.7	741.1	808.5
X3	245	294	318.5	318.5	343	416.5
X4	110.2	140.9	183.8	176.6	220.5	220.5
X5	3.5	3.5	5.25	5.25	7	7
X6	2	2.75	3.5	3.5	4.25	5
X7	0	0.1	0.25	0.2344	0.4	0.4
X8	0	1.75	3	2.812	4	5
Y1	6.01	12.99	18.95	22.31	31.67	43.10
Y2	10.90	15.62	22.08	24.59	33.13	48.03

Bağımlı deęiřkenler için oluřturulan histogramlar řekil 1’de gösterilmektedir.



řekil 1. Bağımlı deęiřkenlere ait histogramlar

İlk olarak tidymodels paketler koleksiyonu yüklenmektedir. R programlama dilinde *install.packages()* fonksiyonu kullanılarak gerekli paket indirilmektedir. *library()* ile indirilen pakete ait fonksiyonlarla çalışabilmek için bir kütüphane oluşturulur. Böylece, tidymodels paketler koleksiyonuna ait tüm paketler ve fonksiyonları R programlama dilinde kullanılabilir.

```
##install.packages("tidymodels")  
library(tidymodels)
```

Veri setini R programlama diline aktarmak için *rio* paketinin içerisindeki *import* fonksiyonu kullanılmıştır. Bu fonksiyonun içerisinde veri setinin yolu belirtilmiştir.

```
##install.packages("rio")  
library(rio)  
ENB2012_data <- import(file = 'https://archive.ics.uci.edu/ml/machine-learning-  
databases/00242/ENB2012_data.xlsx')
```

Bir modelleme projesine başlamak için veri seti, eğitim seti ve test seti olmak üzere 2 bölüme ayrılmalıdır. Eğitim seti, parametreleri tahmin etmek, modelleri karşılaştırmak, modeller için hiperparametre ayarlaması yapmak gibi özellikler için kullanılır. Test seti ise projenin sonuna kadar yedekte tutulur ve model performansını ölçmek için kullanılır.

Verilerin bu bölümlerini oluşturmanın farklı yolları vardır. En yaygın yaklaşım, rastgele bir örnek kullanmaktır. Bu çalışmada verilerin %25'i test ve %75'i eğitim seti için ayrılmıştır.

Rassal örnekleme, rassal veriler ürettiğinden herkesin aynı sonucu bulması için *set.seed()* fonksiyonu kullanılmaktadır.

```
set.seed(123)
```

Veri setini yukarıda bahsedilen bölümlere ayırmak için *rsample* paketinde yer alan *initial_split()* fonksiyonu kullanılmaktadır. Bu fonksiyon verileri alır ve bölümlerin nasıl yapılacağına dair bilgileri kaydeder. *initial_split()* fonksiyonunun *prop* argümanı kullanılarak modelleme ve analiz için saklanacak verilerin oranı belirtilir. Bu uygulama da oran "0.75" olarak belirlenmiştir.

```
enb_split <- initial_split(ENB2012_data, prop = 0.75)  
enb_split  
## <Analysis/Assess/Total>  
## <576/192/768>
```

enb_split isminde tanımladığımız nesnenin çıktısına bakıldığında 768 örneklemin 576'sı analiz, 192'si ise değerlendirme yapacağımız set için ayrıldığı görülmektedir.

Elde edilen verileri çıkarmak için *rsample* paketinin içerisindeki *training()* ve *testing()* fonksiyonları kullanılmaktadır.

```
enb_train <- training(enb_split)  
enb_test <- testing(enb_split)
```

Eğitim seti ve test seti, *enb_train* ve *enb_test* isimli nesnelere tutulmaktadır.

Ayrıca modellemede hiperparametre ayarlaması yapmak için çapraz doğrulama setleri kullanılmıştır. Tidymodels paketinde çapraz doğrulama yapabilmek için *rsample* paketinin içerisinde yer alan *vfold_cv()* fonksiyonu kullanılmaktadır. Bu fonksiyonda yer alan *v* argümanı veri setinin bölümlerinin sayısını ifade eder ve bu uygulamada 10 olarak belirlenmiştir.

```
enb_cv <- vfold_cv(enb_train, v = 10)
```

Tidymodels paketler koleksiyonunda veri ön işleme aşamaları için *recipes* paketi kullanılmaktadır. Bu paket içerisinde yer alan *recipe()* fonksiyonunun ilk argümanı tilde (~) işareti içeren bir formüldür. Tilde (~) işaretinin solundaki değişken bağımlı (ısıtma yükü) ve sağındaki değişkenler ise bağımsız değişkenleri belirtmektedir. *data* argümanı kullanılarak veri seti bu fonksiyon ile ilişkilendirilir. Bu çalışmada bağımsız değişkenlerin değişim aralığını yaklaştırmak amacıyla karekök dönüşümü uygulanmıştır. Bu işlem, *step_sqrt()* ve *all_predictors()* fonksiyonları ile gerçekleştirilmiştir.

```
enb_recipe_hl <-  
  recipe(Y1 ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8, data = ENB2012_data) %>%  
  step_sqrt(all_predictors())  
enb_recipe_hl  
## Data Recipe  
##  
## Inputs:  
##  
##   role #variables  
## outcome      1  
## predictor     8  
##  
## Operations:  
##  
## Square root transformation on all_predictors()
```

Bu uygulamada 6 farklı regresyon tabanlı makine öğrenme algoritması kullanılmıştır. Bu algoritmalar; Doğrusal Regresyon, K – En Yakın Komşuluk, Destek Vektör Makineleri, Karar Ağaçları, Rastgele Orman, Uç Gradyan Artırma ve Yapay Sinir Ağları'dır.

Burada, binaların ısıtma yükü için Uç Gradyan Artırma Algoritması kullanılarak modellemenin nasıl yapıldığı anlatılacaktır.

Uç Gradyan Artırma Algoritması için *parsnip* paketinin içerisinde yer alan *boost_tree()* fonksiyonu kullanılmaktadır. Hesaplama motoru olarak *xgboost* kullanılmıştır ve bu işlem *xgboost* paketi çalıştırılarak gerçekleştirilmektedir. Ayrıca regresyon tabanlı uygulama olduğu için *set_mode()* fonksiyonu *regression* olarak belirtilmiştir.

Uç Gradyan Artırma Algoritması'nda *tree_depth*, *trees*, *mtry*, *min_n* ve *sample_size* parametreleri için hiperparametre ayarlaması yapılmıştır. Bu işlem *boost_tree()* fonksiyonunda parametrelerin *tune()* fonksiyonuna eşitlenmesi ile gerçekleştirilmiştir.

```
boost_model <-  
  boost_tree( tree_depth = tune(),  
             trees = tune(),  
             mtry = tune(),  
             min_n = tune(),  
             sample_size = tune()  
  ) %>%  
  set_engine("xgboost") %>%  
  set_mode("regression")
```

Oluşturulan model ve veri ön işleme aşamaları *workflows* paketinin *workflow()* fonksiyonu ile birleştirilir. Bu işlem *add_model()* ve *add_recipe()* fonksiyonları ile gerçekleştirilir. *workflow()* kullanılarak modeli eğitmek ve test etmek daha kolay hale gelir.

```

set.seed(123)
boost_wf <-
  workflow() %>%
  add_model(boost_model) %>%
  add_recipe(enb_recipe_hl)
boost_wf
## == Workflow =====
## Preprocessor: Recipe
## Model: boost_tree()
##
## -- Preprocessor -----
## 1 Recipe Step
##
## * step_sqrt()
##
## -- Model -----
## Boosted Tree Model Specification (regression)
##
## Main Arguments:
## mtry = tune()
## trees = tune()
## min_n = tune()
## tree_depth = tune()
## sample_size = tune()
##
## Computational engine: xgboost

```

Bir *workflow* oluşturulduktan sonra çapraz doğrulama setlerinin her biri için *tune* paketinin *tune_grid()* fonksiyonu ile hiperparametre ayarlaması yapılmıştır. *metric_set* fonksiyonu ile karşılaştırma yapılacak kriterler belirlenmiştir. Bu çalışmada kök ortalama kare hatası (RMSE), ortalama mutlak hata (MAE) ve R - Kare (R2) kriterleri kullanılmıştır. Bu kriterlere ait formüller sırasıyla Eş. 1, Eş. 2 ve Eş. 3'de verilmiştir;

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (1)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2)$$

$$R2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (3)$$

Burada, N veri setindeki örneklem sayısını göstermektedir. \hat{y}_i tahmin edilen değerleri, y_i veri setindeki bağımlı değişkenlerin gerçek değerlerini ve \bar{y}_i ise veri setindeki bağımlı değişkenlerin gerçek değerlerinin ortalamasını belirtmektedir. RMSE, bir makine öğrenmesi modelinin tahmin ettiği değerler ile gerçek değerleri arasındaki uzaklığın bulunmasında sıklıkla kullanılan, hatanın büyüklüğünü ölçen bir kriterdir. Hatalar, regresyon modelinin veri setindeki gerçek değerlerden ne kadar uzakta olduğunun bir ölçüsüdür. RMSE ise bu hataların ne kadar yayıldığına bir ölçüsüdür. RMSE, 0'dan sonsuza kadar değer alabilir. Bir makine öğrenmesi modelinin iyi performans göstermesi için RMSE değerinin 0'a yakın olması gerekir. MAE, gerçek değer ile tahmin edilen değer arasındaki ortalama mutlak mesafedir. MAE değeri kolay yorumlanabilir olduğu için regresyon ve zaman serileri problemlerinde sıklıkla kullanılmaktadır. MAE, 0'dan sonsuza kadar değer alabilir. Bir makine öğrenmesi modelinin iyi performans göstermesi için MAE değerinin 0'a yakın olması gerekir. R2,

hataların kareler toplamı ve veri setindeki bağımlı değişkende yer alan değerlerin ortalamaya uzaklığının kareler toplamından faydalanılarak hesaplanır. R2 denkleminde de yer alan bu değerlerin oranı ne kadar küçük ise R2 o kadar yüksek olacaktır. Bir makine öğrenmesi modelinin iyi performans göstermesi için R2 değerinin 1'e yakın olması gerekir.

```
boost_results <-
  boost_wf %>%
  tune_grid(resamples = enb_cv,
            metrics = metric_set(rmse, rsq, mae)
  )
```

Hiperparametre ayarlaması için çapraz doğrulama setlerindeki sonuçlar *tune* paketinin içerisinde yer alan *collect_metrics()* fonksiyonu kullanılarak listelenmektedir.

```
boost_results %>%
  collect_metrics()
## # A tibble: 30 x 11
##   mtry trees min_n tree_depth sample_size .metric .estimator mean   n
##   <int> <int> <int>   <int>     <dbl> <chr>   <chr>   <dbl> <int>
## 1     1     1  904     4       12  0.993 mae     standard  0.277  10
## 2     1     1  904     4       12  0.993 rmse    standard  0.417  10
## 3     1     1  904     4       12  0.993 rsq     standard  0.998  10
## 4     2     2 1570    31        5  0.533 mae     standard  0.837  10
## 5     2     2 1570    31        5  0.533 rmse    standard  1.39   10
## 6     2     2 1570    31        5  0.533 rsq     standard  0.981  10
## 7     3     3 1866    18        5  0.584 mae     standard  0.313  10
## 8     3     3 1866    18        5  0.584 rmse    standard  0.431  10
## 9     3     3 1866    18        5  0.584 rsq     standard  0.998  10
## 10    3    1350   36       13  0.254 mae     standard  1.04   10
## # ... with 20 more rows, and 2 more variables: std_err <dbl>, .config <chr>
```

Bu sonuçlar arasından en iyisini elde etmek için *select_best()* fonksiyonu kullanılmaktadır. *metric* argümanı ile kullanmak istediğimiz metrik belirlenir. Bu uygulama da *rmse* olarak tercih edilmiştir.

```
param_final <- boost_results %>%
  select_best(metric = "rmse")
param_final
## # A tibble: 1 x 6
##   mtry trees min_n tree_depth sample_size .config
##   <int> <int> <int>   <int>     <dbl> <chr>
## 1     1     1  904     4       12  0.993 Preprocessor1_Model01
```

Çıktıya göre 1. modele ait hiperparametre değerleri listelenmektedir. Uç Gradyan Artırma Algoritması için bu değerler kullanılarak tekrar *workflow()* oluşturulur. Bu işlem için *finalize_workflow()* fonksiyonu kullanılmıştır.

```
boost_wf <- boost_wf %>%
  finalize_workflow(param_final)
boost_wf
## == Workflow =====
## Preprocessor: Recipe
## Model: boost_tree()
##
## -- Preprocessor -----
```

```
## 1 Recipe Step
##
## * step_sqrt()
##
## -- Model -----
## Boosted Tree Model Specification (regression)
##
## Main Arguments:
## mtry = 1
## trees = 904
## min_n = 4
## tree_depth = 12
## sample_size = 0.993323943288997
##
## Computational engine: xgboost
```

Çıktıya bakıldığında önışleme aşaması için karekök dönüşümü uygulandığı ve Uç Gradyan Artırma Algoritması için hiperparametre değerlerinin listelendiği görülmektedir.

Bu değerler kullanılarak test verisi modellenmiştir. Bunun için *tune* paketinin içerisinde yer alan *last_fit* fonksiyonu kullanılmıştır.

```
boost_fit <- boost_wf %>%
  last_fit(enb_split)
```

Test verisi tahminlendikten sonra sonuçları görmek için *collect_predictions()* fonksiyonu kullanılmıştır.

```
test_performance <- boost_fit %>% collect_predictions()
test_performance
## # A tibble: 192 x 5
##   id      .pred .row  Y1 .config
##   <chr>   <dbl> <int> <dbl> <chr>
## 1 train/test split 21.0    5 20.8 Preprocessor1_Model1
## 2 train/test split 18.0   13 17.0 Preprocessor1_Model1
## 3 train/test split 18.1   14 17.4 Preprocessor1_Model1
## 4 train/test split 27.3   20 28.8 Preprocessor1_Model1
## 5 train/test split 24.6   23 24.8 Preprocessor1_Model1
## 6 train/test split  6.07  26  6.05 Preprocessor1_Model1
## 7 train/test split  6.71  34  6.79 Preprocessor1_Model1
## 8 train/test split  7.34  39  7.1  Preprocessor1_Model1
## 9 train/test split  7.53  40  7.1  Preprocessor1_Model1
## 10 train/test split 10.7   41 10.8 Preprocessor1_Model1
## # ... with 182 more rows
```

Çıktıya göre *.pred* sütunu ısıtma yüklerine ait tahminlenen değerleri ve *Y1* sütunu ise ısıtma yükleri için gerçek değerleri göstermektedir. *.row* sütunu ise bu değerlerin veri setindeki sırasını ifade etmektedir. *yardstick* paketinin *metric_set()* fonksiyonu ile metrikler belirlenmiştir. Burada kullanılan metrikler; kök ortalama kare hatası (RMSE), R - Kare (R2) ve ortalama mutlak hata (MAE) metrikleridir.

```
enb_metrics <- metric_set(rmse, rsq, mae)
```

enb_metrics isminde tanımladığımız nesne kullanılarak metriklerin değerleri listelenmektedir. *truth* argümanı ile gerçek değerler, *estimate* argümanı ile tahminlenen değerler belirtilmiştir.

```
enb_metrics(data = test_performance, truth = Y1, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>         <dbl>
## 1 rmse    standard      0.314
## 2 rsq     standard      0.999
## 3 mae     standard      0.202
```

Bu çıktıda *.estimate* sütunu kullanılan metriklere ait sonuçları listelemektedir. Eğitim seti Uç Gradyan Artırma Algoritması kullanılarak modellendiğinde, test seti için RMSE 0.314, R2 0.999 ve MAE 0.202 olarak elde edilmiştir. Analizlerin tümü soğutma yükü için de uygulanmış ve sonuçlar elde edilmiştir. Ayrıca, diğer yöntemler için de ısıtma ve soğutma yükleri tahminlenmiştir. Son bölümde bu tahminlere ait RMSE, R2 ve MAE değerleri tablolar halinde gösterilmiş ve sonuçlar yorumlanmıştır.

IV. SONUÇLAR VE TARTIŞMA

Tablo 2’de ısıtma yüküne ait elde edilen sonuçlar gösterilmektedir. En küçük RMSE ve MAE ve en büyük R2 değerlerine sahip Uç Gradyan Artırma Algoritması diğer yöntemlerden daha başarılıdır. K – En Yakın Komşuluk, Karar Ağaçları, Rastgele Orman ve Yapay Sinir Ağları yöntemleri de iyi sonuç vermekle beraber en kötü sonuçlar Doğrusal Regresyon ve Destek Vektör Makineleri Algoritmaları ile elde edilmiştir. Tablo 3’te soğutma yüküne ait elde edilen sonuçlar gösterilmektedir. Soğutma yükü için yöntemlerin tahminlemedeki başarılarının azaldığı görülmektedir. Ancak yine Uç Gradyan Artırma Algoritması diğer yöntemlerden daha güvenilir sonuçlar vermiştir. Uç Gradyan Artırma Algoritmasına en yakın başarı oranları K – En Yakın Komşuluk, Karar Ağaçları, Rastgele Orman ve Yapay Sinir Ağları yöntemleri ile elde edilmiştir. En kötü sonuçlar Doğrusal Regresyon ve Destek Vektör Makinesi Algoritmaları ile alınmıştır.

Tablo 2. Isıtma yükü tahmini için elde edilen sonuçlar

Yöntemler	RMSE	R2	MAE
Doğrusal Regresyon (DR)	2.640	0.924	1.980
K – En Yakın Komşuluk (KEYK)	0.728	0.994	0.569
Destek Vektör Makineleri (DVM)	2.490	0.934	1.740
Karar Ağaçları (KA)	0.548	0.997	0.379
Rastgele Orman (RO)	0.466	0.998	0.331
Uç Gradyan Artırma (UGA)	0.314	0.999	0.202
Yapay Sinir Ağları (YSA)	0.567	0.996	0.442

Tablo 3. Soğutma yükü tahmini için elde edilen sonuçlar

Yöntemler	RMSE	R2	MAE
Doğrusal Regresyon (DR)	3.100	0.884	2.230
K – En Yakın Komşuluk (KEYK)	1.480	0.973	0.972
Destek Vektör Makineleri (DVM)	3.040	0.888	2.110
Karar Ağaçları (KA)	1.490	0.973	0.968
Rastgele Orman (RO)	1.570	0.970	1.030
Uç Gradyan Artırma (UGA)	0.557	0.996	0.410
Yapay Sinir Ağları (YSA)	1.420	0.976	0.924

Test veri setinin ilk 10 gözlemi için ısıtma ve soğutma yüklerinin gerçek değerleri ve yöntemlere ait tahminlenen değerler Tablo 4 ve Tablo 5’te gösterilmektedir. Beklendiği gibi ısıtma ve soğutma yükleri için en iyi tahminler Uç Gradyan Artırma ile elde edilmiştir. Uç Gradyan Artırmaya en yakın sonuçları veren yöntemler ise Karar Ağaçları, K-En Yakın Komşuluk, Rastgele Orman ve Yapay Sinir Ağları’dır.

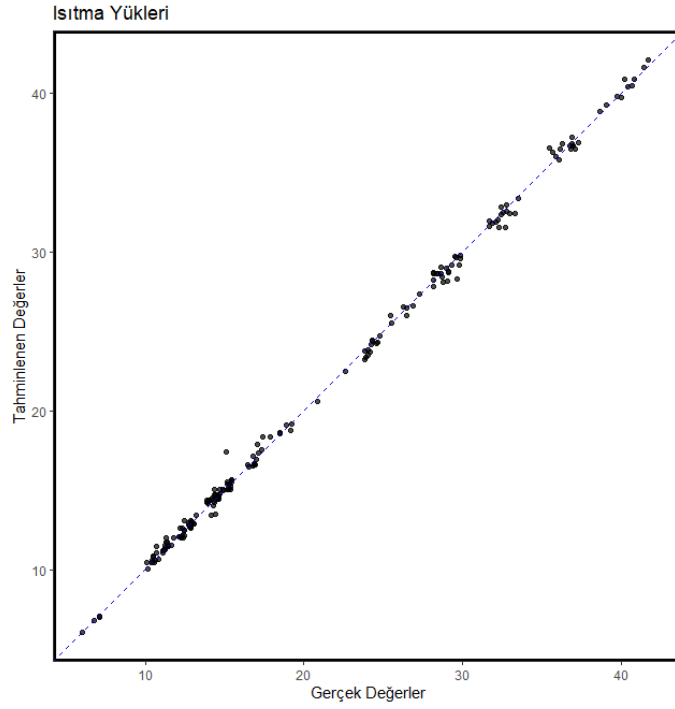
Tablo 4. Isıtma yükü için test veri setinin ilk 10 gözleminin gerçek değerleri ve tahminlenen değerler

Gerçek Değerler	DR	KEYK	DVM	KA	RO	UGA	YSA
20.8	22.5	21.4	19.9	18.0	20.1	21.0	20.9
17.0	23.5	19.3	19.2	18.0	17.6	18.0	16.9
17.4	23.5	19.4	18.9	18.0	17.6	18.1	16.7
28.8	24.7	27.9	19.6	26.8	27.8	27.3	29.5
24.8	28.4	25.0	23.6	26.8	25.8	24.6	23.9
6.05	2.71	6.17	7.69	6.53	6.14	6.07	5.95
6.79	4.98	6.79	7.98	6.53	6.88	6.71	7.07
7.10	6.37	7.94	8.67	6.53	7.32	7.34	7.77
7.10	6.36	7.93	9.20	6.53	7.32	7.53	7.82
10.8	7.34	9.84	9.94	9.42	10.1	10.7	11.0

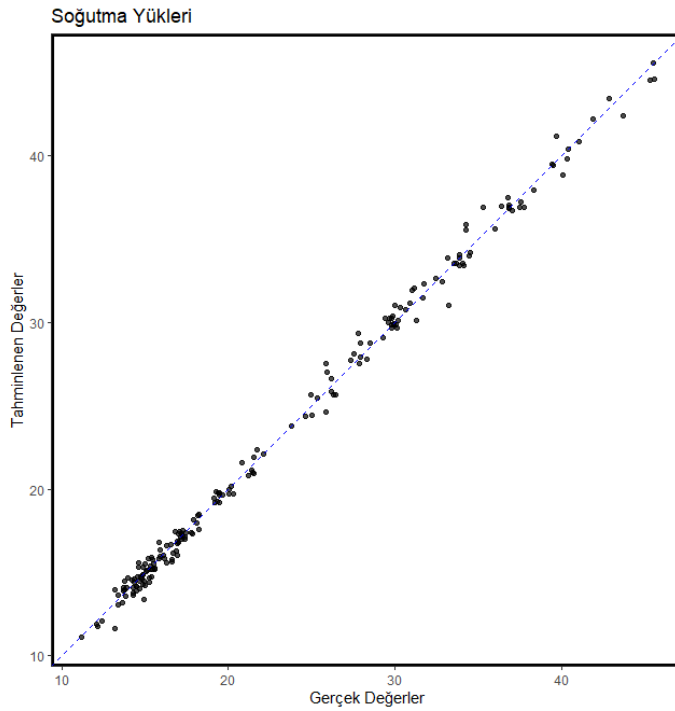
Tablo 5. Soğutma yükü için test veri setinin ilk 10 gözleminin gerçek değerleri ve tahminlenen değerler

Gerçek Değerler	DR	KEYK	DVM	KA	RO	UGA	YSA
28.3	26.8	27.2	24.9	24.0	26.6	28.2	28.0
23.8	28.0	26.3	24.4	24.0	24.8	24.4	23.8
21.5	28.1	24.2	24.5	24.0	24.4	21.7	20.4
39.4	29.3	32.8	25.3	37.8	35.9	37.5	34.4
29.8	31.7	30.0	28.2	33.3	31.3	30.8	28.6
11.2	8.48	11.2	12.4	11.6	11.4	11.0	12.0
12.0	10.3	11.8	13.1	11.6	11.9	12.0	12.1
12.4	11.5	13.1	13.7	11.6	12.6	12.1	12.1
12.1	11.6	13.1	14.2	11.6	12.6	12.4	12.1
16.8	12.0	15.4	14.5	18.8	17.0	16.9	18.0

Uç Gradyan Artırma Algoritması ile elde edilen sonuçlara göre test veri seti için ısıtma ve soğutma yüklerinin gerçek değerleri ve tahminlenen değerler arasındaki ilişki sırasıyla Şekil 2 ve Şekil 3'te gösterilmektedir. Bu şekillerde x eksenini gerçek değerleri ve y eksenini ise tahminlenen değerleri göstermektedir.



Şekil 2. Isıtma yükleri için gerçek değerler ve tahminlenen değerler arasındaki ilişki



Şekil 3. Soğutma yükleri için gerçek değerler ve tahminlenen değerler arasındaki ilişki

Bu çalışmada model değerlendirme amacıyla eğitim verilerine 10 katlı çapraz doğrulama uygulanmış ve test verileri tahmin edilerek yöntemler arasında performans karşılaştırması yapılmıştır. Buna ek olarak, tüm veri seti üzerinde 10 katlı çapraz doğrulama uygulanarak ikinci bir model değerlendirme çalışması daha gerçekleştirilmiştir. Yapılan bu ek değerlendirmeye ait sonuçlar Tablo 6 ve Tablo 7’de gösterilmektedir. Bu sonuçlara göre de ısıtma ve soğutma yükleri için en iyi tahminler Uç Gradyan Artırma ile elde edilmiştir.

Tablo 6. İkinci model değerlendirme çalışmasında ısıtma yükü tahmini için elde edilen sonuçlar

Yöntemler	RMSE	R2	MAE
Doğrusal Regresyon (DR)	2.8279580	0.9199922	2.0844323
K – En Yakın Komşuluk (KEYK)	0.7579723	0.9943716	0.5443130
Destek Vektör Makineleri (DVM)	1.2402958	0.9847578	1.0156323
Karar Ağaçları (KA)	0.5824454	0.9964269	0.4027332
Rastgele Orman (RO)	0.5296280	0.9970863	0.3650237
Uç Gradyan Artırma (UGA)	0.3579582	0.9987080	0.2517534
Yapay Sinir Ağları (YSA)	0.5710616	0.9967602	0.4239747

Tablo 7. İkinci model değerlendirme çalışmasında soğutma yükü tahmini için elde edilen sonuçlar

Yöntemler	RMSE	R2	MAE
Doğrusal Regresyon (DR)	3.1958456	0.8863866	2.2969929
K – En Yakın Komşuluk (KEYK)	1.5746363	0.9725457	1.0309807
Destek Vektör Makineleri (DVM)	1.7081706	0.9685843	1.2424118
Karar Ağaçları (KA)	1.8523717	0.9617301	1.2550302
Rastgele Orman (RO)	1.7138051	0.9674601	1.1501710
Uç Gradyan Artırma (UGA)	0.6111269	0.9959043	0.4313350
Yapay Sinir Ağları (YSA)	1.3889763	0.9785829	0.9643851

Yapılan bu çalışma ile enerji verimli binaların ısıtma ve soğutma yükleri R İstatistiksel Programlama Dili'nin güncel çalışma alanlarından biri olan düzenli modeller ile tahminlenmiştir. R'de makine öğrenmesi için farklı paketler kullanılmaktadır. Bu paketlerdeki sözdizimleri de farklılık gösterdiği için kodları yazmak zahmetli ve zaman alan bir süreç olmaktadır. Düzenli modeller ile birlikte tüm makine öğrenmesi algoritmaları tek bir pakette birleştirilmiş ve büyük bir kullanım kolaylığı sağlanmıştır. R'de bu uygulamalar tidymodels paketi yüklenerek gerçekleştirilir. Bu çalışmada düzenli modeller tanıtılmış, popüler makine öğrenmesi algoritmaları ve yapay sinir ağları kullanılarak binaların ısıtma ve soğutma yükleri düzenli modeller ile tahminlenmiştir. RMSE, R2 ve MAE kriterleri kullanılarak alınan sonuçlara göre en başarılı algoritma Uç Gradyan Artırma olmuştur. Bu algoritmaya en yakın sonuçları veren yöntemler ise Karar Ağaçları, K-En Yakın Komşuluk, Rastgele Orman ve Yapay Sinir Ağları olarak görülmektedir. Doğrusal Regresyon ve Destek Vektör Makinesi Algoritmaları'nın ise ısıtma ve soğutma yüklerinin tahminlenmesinde başarısız olduğu gözlemlenmiştir. Uygulama bölümünün başında verilen açıklayıcı değişken değerleri bilindiğinde Uç Gradyan Artırma Algoritması ile herhangi bir binayı ısıtmak ya da soğutmak için gerekli yükler yüksek hassasiyet ile tahmin edilebilir. Kullanılan değişken listesine yenilerinin eklenmesi veya veri setinin değişmesi durumunda açıklanan adımlar tekrar edilip yeni duruma ilişkin tahminler kısa sürede elde edilebilir. Binalarda ısıtma ve soğutma yüklerinin tahmini için Uç Gradyan Artırma Algoritması'nın kullanılması önerilmektedir. Uygulamaya ait R kodları github üzerinden paylaşılmıştır (<https://github.com/burakdilber/Tidymodels-Article>).

V. KAYNAKLAR

- [1] Ö. Aydın, "Binalarda enerji verimliliği kapsamında yapılan projelerin değerlendirilmesi: Türkiye örneği," *Mimarlık ve Yaşam Dergisi*, c. 4, s. 1, ss. 55–68, 2019.
- [2] A. Tsanas, and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and Buildings*, vol. 49, pp. 560–567, 2012.
- [3] A. Yeziro, B. Dong, and F. Leite, "An applied artificial intelligence approach towards assessing building performance simulation tools," *Energy and Buildings*, vol. 40, pp. 612–620, 2008.

- [4] J. S. Chou, and D. K. Bui, “Modeling heating and cooling loads by artificial intelligence for energy-efficient building design,” *Energy and Buildings*, vol. 82, pp. 437–446, 2014.
- [5] M. Peker, O. Özkaraca ve B. Kesimal, “Enerji tasarruflu bina tasarımı için ısıtma ve soğutma yüklerini regresyon tabanlı makine öğrenmesi algoritmaları ile modelleme,” *Bilişim Teknolojileri Dergisi*, c. 10, s. 4, ss. 443–449, 2017.
- [6] A. Gümüşçü, M. E. Tenekeci, N. Beşli, M. A. İlkan, E. Aslan ve S. Abamor, “Yapılarda ısıtma ve soğutma yükü sınıfların makine öğrenme yöntemleri ile belirlenmesi,” *Harran Üniversitesi Mühendislik Dergisi*, c. 3, s. 2, ss. 60–66, 2018.
- [7] S. Seyedzadeh, F. P. Rahimian, P. Rastogi, and I. Glesk, “Tuning machine learning models for prediction of building energy loads,” *Sustainable Cities and Society*, vol. 47, pp. 1–18, 2019.
- [8] S. S. Roy, P. Samui, I. Nagtode, H. Jain, and V. Shivaramakrishnan, “Forecasting heating and cooling loads of buildings: A comparative performance analysis,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1253–1264, 2020.
- [9] A. Moradzadeh, A. M. Saatloo, B. M. Ivatloo, and A. A. Moghaddam, “Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings,” *Applied Sciences*, vol. 10, no. 11, pp. 1–12, 2020.
- [10] A. F. Özdemir, E. Yıldıztepe ve M. Binar, “İstatistiksel yazılım geliştirme ortamı: R,” *Akademik Bilişim’10-XII. Akademik Bilişim Konferansı Bildirileri*, Muğla, Türkiye, 2010, ss. 293–297.
- [11] S. M. Bache, and H. Wickham. (2022, March 30). *magrittr: A forward-pipe operator for R* [Online]. Available: <https://cran.r-project.org/web/packages/magrittr/index.html>
- [12] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani, “Welcome to the tidyverse,” *The Journal of Open Source Software*, vol. 4, no. 43, pp. 1–6, 2019.
- [13] M. Kuhn, and H. Wickham. (2022, March 19). *tidymodels: Easily install and load the ‘tidymodels’ packages* [Online]. Available: <https://cran.r-project.org/web/packages/tidymodels/index.html>
- [14] M. Kuhn, and H. Wickham. (2022, February 18). *recipes: Preprocessing tools to create design matrices* [Online]. Available: <https://cran.r-project.org/web/packages/recipes/index.html>
- [15] J. Silge, F. Chow, M. Kuhn, and H. Wickham. (2021, November 8). *rsample: General resampling infrastructure* [Online]. Available: <https://cran.r-project.org/web/packages/rsample/index.html>
- [16] D. Robinson, A. Hayes, and S. Couch. (2022, January 28). *broom: Convert statistical objects into tidy tibbles* [Online]. Available: <https://cran.r-project.org/web/packages/broom/index.html>
- [17] M. Kuhn, and D. Vaughan. (2022, March 17). *parsnip: A common API to modeling and analysis functions* [Online]. Available: <https://cran.r-project.org/web/packages/parsnip/index.html>
- [18] D. Vaughan. (2022, March 18). *workflows: Modeling workflows* [Online]. Available: <https://cran.r-project.org/web/packages/workflows/index.html>
- [19] M. Kuhn. (2022, March 19). *tune: Tidy tuning tools* [Online]. Available: <https://cran.r-project.org/web/packages/tune/index.html>

- [20] M. Kuhn. (2022, April 6). *dials: Tools for creating tuning parameter values* [Online]. Available: <https://cran.r-project.org/web/packages/dials/index.html>
- [21] M. Kuhn, and D. Vaughan. (2021, November 22). *yardstick: Tidy characterizations of model performance* [Online]. Available: <https://cran.r-project.org/web/packages/yardstick/index.html>
- [22] A. Tsanas, and A. Xifara. (2012, November 30). *Energy efficiency data set* [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Energy+efficiency>