

KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ

MÜHENDİSLİK MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ



Ocak 2022

GÜZ DÖNEMİ

BİL 413

ÖRÜNTÜ TANIMA DERSİ

FİNAL PROJESİ

KÜRESEL ISINMADA BİNALARIN ENERJİ VERİMLİLİĞİNİN ARTIRILMASI

AMAÇLI MAKİNE ÖĞRENMESİ ÇALIŞMASI

Hazırlayan

HÜSEYİN TAŞ

18110131021

Özet_ Küresel ısınma konusu günümüzde çok bilinir hale geldi. Çünkü küresel ısınmanın etkileri günümüzde daha çok hissetmeye başladık. Küresel ısınma, başlıca atmosfere salınan gazların neden olduğu düşünülen sera etkisinin sonucunda, dünya üzerinde yıl boyunca kara, deniz ve havada ölçülen ortalama sıcaklıklarda görülen artışa verilen isimdir. Küresel ısınmanın birçok alanda olumsuz etkisi görülmektedir. Küresel ısınmanın en büyük sebebi insan faaliyetlerinin sonucunda etrafa yayılan sera gazlarıdır. Bu faaliyetler arasında en önemlisi elektrik üretimi, ısınma ve taşımacılık için yakılan fosil yakıtlardır. Bu yüzden sürdürülebilir temiz enerji kaynaklarına yönelmeli ve enerjinin verimli kullanılması önemlidir. Bu çalışmada enerjinin verimli kullanılması gerekliliğini ortaya çıkaran bir makine öğrenmesi yaklaşımı sunmaktadır. Çalışma ile binaların enerji verimliliği ısıtma ve soğutma talebinin sistematik olarak azaltılmasını amaçlamaktadır.

Anahtar Kelimeler_ Makine öğrenmesi, Küresel ısınma, enerji verimliliği

1.GİRİŞ

Küresel ısınma, karbondioksit gibi ısıyı tutan gazların atmosferde artması oluşan ve atmosfere salınan sera gazlarının neden olduğu düşünülen sera etkisinin sonucunda, sıcaklıkların artmasıdır.[1]

Küresel iklim değişikliği kuraklık, fosil yakıt tüketimi ve şiddetli yağışlar ve seller vb. gibi birçok olumsuz etkiye sebep olmaktadır. Küresel ısınma 21. yüzyılda üzerinde tartışılan en önemli çevre sorunudur. Küresel ısınma sadece gelişmiş ülkeleri değil tüm dünyayı etkilemektedir.

Sanayi devriminden önce insanlar sadece doğadan elde ettiklerini tüketiyorlardı. Sanayi devriminden sonra ise ticari etkinler sonrası fosil yakıtların kullanımı nedeniyle dünyamızın tahrip olmasına ve küresel ısınma etkilerini daha çok hissetmeye başlanılan günlere geldik. Yaşamımızı daha güzel ve konforlu sürdürebilmek için enerjiyi tasarruf yapmamızı ve sürdürebilir enerji kaynaklarına yönlendirmemiz gerekiyor.

Daha bilinçli davranarak yaptığımız işlerde kullandığımız enerjinin yarısıyla halledebiliriz. Uzmanlar küresel ısınma sebebiyle birkaç yıl içerisinde dünyamızda kimseye yetecek kaynak kalmayacağını söylüyor.[2]

Evlerimizde enerji tasarrufu yapılabilir, mantolama ve yalıtım yaptırarak enerji kullanımımızı azaltabiliriz ya da az enerji kullanan ürünler kullanabiliriz.

Bu çalışmada evlerdeki enerji verimliliğini artırmayı hedeflemekteyiz. Binaların çevreye verilen zararı minimumda tutmaya çalışıyoruz. Binanın en uygun mimarı ile tasarlanması hedeflenmektedir.

Proje hangi sosyal sorumluluğu desteklemektedir?

Proje küresel ısınmanın etkisini azaltabilmek için ve çevreye verilen zararı minimuma indirmeyi hedeflemektedir. Binanın ısıtma ve soğutma yükünü azaltmayı amaçlamaktadır. Binaın en iyi mimarı yapısını ve konumunu sağlayacak, en optimum enerji ihtiyacını bulacaktır.

Geliştireceğiniz örüntü tanıma modeli kimlere, hangi konuda faydalı olacaktır?

Bu proje tüm herkese faydalı olacaktır. Çünkü enerji verimliliği küresel ısınmayı etkileyen önemli bir konudur. Ve enerji kullanımı azaldığı için herkesten daha az para gidecektik dolayısıyla herkese bir fayda sağlayacaktır. Devlete de bir faydası vardır enerjiyi dışarıdan ithal bir ülke olarak daha az maliyeti olmuş olacaktır.

Geliştireceğiniz örüntü tanıma modeli için ne tip verilere ihtiyacınız vardır ve bu verileri ne şekilde temin ettiniz?

Geliştirecek model için gerekli olan bina özellikleri lazımdır. Veriyi UCI sitesinden Energy efficiency dataset[3] üzerinden veriyi elde ettim.

2.MATERYAL METHOD

Bu çalışmada 2.40 GHz Intel Core i5-9300H işlemci, 16GB RAM, 1TB SSD bellek ve windows10 işletim sistemine sahip bir sistem ve Anaconda dağıtımı üzerinden jupyter lab kullanılarak Python programlama dili kullanılmıştır.

1) Veriseti

Bu çalışmada UCI sitesinden alınmış, simüle edilen 12 farklı bina şeklini kullanarak elde edilen veri setini kullanılmıştır. Veriseti 768 satır ve 10 sütundan oluşmaktadır. Veri setindeki her bir satır, binaya ait özellikleri içermektedir.

	X1	X2	X3	X4	X5	X6	X7	X8	Y1	Y2
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28

Resim1- Ham veri seti örneği

Öznitelikler

- X1 (Relative Compactness : Bağlı kompaktlık)
- X2 (Surface Area : Yüzey Alanı)
- X3 (Wall Area : Duvar Alanı)
- X4 (Roof Area : Çatı Alanı)
- X5 (Overral Height : Toplam Yükseklik)
- X6 (Orientation - 2:North, 3: East, 4:South, 5:West)
- X7 (Glazing Area : Cam Alanı)
- X8 (Glazing Are Distribution:Cam alanının dağılımı-1:Uniform, 2:North, 3:East, 4:South, 5:West)
- Y1 (Heating Load : Isıtma Yüğü)
- Y10 (Cooling Load : Soğutma Yüğü)

2)Veri Önişleme

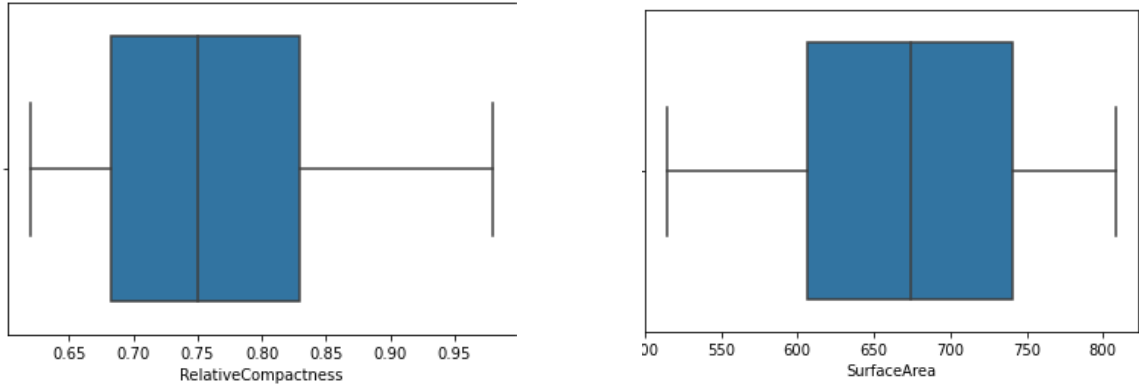
Veri seti ile ilgili gerekli dönüşümler, eksik veri analizi, aykırı değerler gibi işlemlere bakılmıştır.

- **Eksik Veri Analizi :** Veri setinde eksik veri yoktur.

```
RelativeCompactness    0
SurfaceArea             0
WallArea                0
RoofArea                0
OverallHeight           0
Orientation              0
GlazingArea             0
GlazingAreaDist         0
HeatingLoad             0
CoolingLoad             0
dtype: int64
```

Resim2-Eksik veri analizi sonucu

- **Aykırı değer Analizi:** Aykırı değer yoktur. Aykırı değer analizini boxplot kullanılarak yapılmıştır. Ve her grafiğin sonucu Resim3'teki gibi hiç aykırı değer olmadığı görülür.



Resim3-Aykırı veri analizi

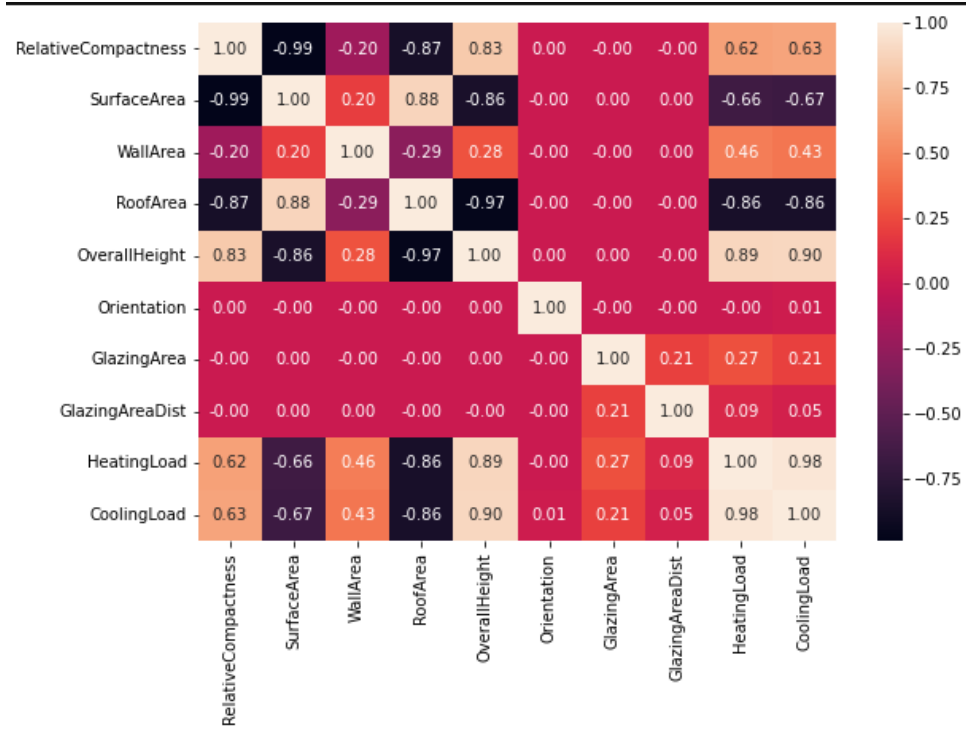
- **Veri sütunlarını yeniden adlandırma:** Veri setindeki sütun isimleri anlamsız olduğu için sütun isimlerini daha anlaşılır şekilde güncellenmiştir. Ve kullanılacak yeni veri seti Resim4'teki gibidir.

	RelativeCompactness	SurfaceArea	WallArea	RoofArea	OverallHeight	Orientation	GlazingArea	GlazingAreaDist	HeatingLoad	CoolingLoad
0	0.98	514.5	294.0	110.25	7.0	2	0.0	0	15.55	21.33
1	0.98	514.5	294.0	110.25	7.0	3	0.0	0	15.55	21.33
2	0.98	514.5	294.0	110.25	7.0	4	0.0	0	15.55	21.33
3	0.98	514.5	294.0	110.25	7.0	5	0.0	0	15.55	21.33
4	0.90	563.5	318.5	122.50	7.0	2	0.0	0	20.84	28.28
5	0.90	563.5	318.5	122.50	7.0	3	0.0	0	21.46	25.38
6	0.90	563.5	318.5	122.50	7.0	4	0.0	0	20.71	25.16
7	0.90	563.5	318.5	122.50	7.0	5	0.0	0	19.68	29.60
8	0.86	588.0	294.0	147.00	7.0	2	0.0	0	19.50	27.30
9	0.86	588.0	294.0	147.00	7.0	3	0.0	0	19.95	21.97

Resim4- Modelde kullanılacak veri seti

3)Veri Görselleştirme

Veri setindeki sütunlar arasındaki ilişkinin daha iyi anlaşılması için veri görselleştirilmiştir.

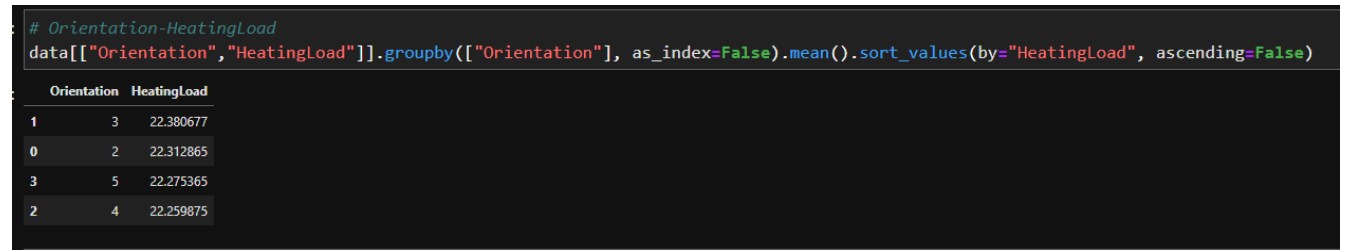


Resim5-Veriler arasındaki ilişkinin görselleştirilmesi

4) Veri Analizi

Veri seti ile ilgili bazı bilgiler elde etmek için ve veri setini daha iyi tanımak için birkaç tane basit düzeyde veri analizi yapılmıştır.

- Binanın hangi yöne baktığının ısıtma yüküne etkisi nedir?



Resim6-Orientation HeatinLoad Etkisi

- Binanın hangi yöne baktığının soğutma yüküne etkisi nedir?

```
# Orientation-CoolingLoad
data[["Orientation", "CoolingLoad"]].groupby(["Orientation"], as_index=False).mean().sort_values(by="CoolingLoad", ascending=False)
```

	Orientation	CoolingLoad
3	5	24.953646
0	2	24.604531
2	4	24.480313
1	3	24.312552

Resim7-Orientation CoolingLoad Etkisi

- Cam alanının ısıtma yüküne etkisi nedir?

```
# GlazingArea - HeatingLoad
data[["GlazingArea", "HeatingLoad"]].groupby(["GlazingArea"], as_index=False).mean().sort_values(by="HeatingLoad", ascending=False)
```

	GlazingArea	HeatingLoad
3	0.40	25.411667
2	0.25	22.756917
1	0.10	20.357167
0	0.00	14.286375

Resim8-GlazingArea HeatingLoad Etkisi

- Cam alanının soğutma yüküne etkisi nedir?

```
# GlazingArea - CoolingLoad
data[["GlazingArea", "CoolingLoad"]].groupby(["GlazingArea"], as_index=False).mean().sort_values(by="CoolingLoad", ascending=False)
```

	GlazingArea	CoolingLoad
3	0.40	26.911417
2	0.25	24.892625
1	0.10	22.935542
0	0.00	19.706250

Resim9-GlazingArea CoolingLoad Etkisi

5) Feature Selection

Bu aşamada veri seti makine öğrenmesi için %80 ve %20 oranlarında eğitim ve test verisi olarak ayrılmıştır. Eğitim verisinde 614 satır, test verisinde ise 154 satır veri bulunmaktadır.

Artık veri setimiz makine öğrenmesi için hazır haldedir. Bundan sonraki işlem modeli oluşturmamız ve eğitime işlemidir.

6) Eğitim

a- Model oluşturma

Eğitim aşamasına başlamadan önce verimiz hangi öğrenme türüne uygun olduğunu ve kullanılacak algoritmalar belirlenecektir. Veri setimiz etiketli veri olduğu için bu bir denetimli öğrenmedir. Bizde bu çalışmada bir binanın ısıtma ve soğutma yükü tahmin edilecektir. Biz projemizde Linear regresyon(Linear regression), Rastgele orman

regressor(Random forest regressor), karar ağacı regressor(Decision Tree regressor) algoritmaları kullanılmıştır. Tahmini yapılacak veri sürekli bir veri olduğu için regresyon (tahminleme) yapılmıştır. Veriler algoritmalar ile eğitilir ve veriler arasındaki ilişkiler tespit edilir.

b- Modeli test verisine uygulama

Belirlediğimiz her bir algoritmayla model oluşturduktan sonra bu modellerin performanslarını öğrenmek için model test verisine uygulanır. Ve her bir modelin performans değerini elde ederiz. Ve Resim11’deki ilkel test başarılarını elde etmiş oluyoruz.

Model	İlkel Skor
Linear Regression	0.90
Random Forest	0.97
Decision Tree	0.96

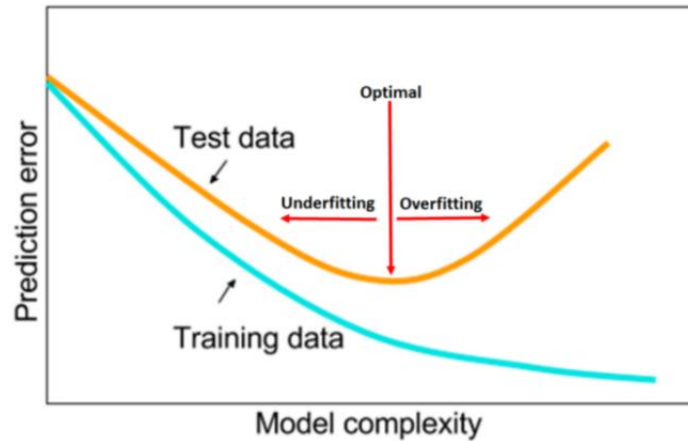
Resim11-İlkel model skorlarımız

c-Modellerin en iyi parametre kombinasyonunu bulma

En iyi hiperparametrenin bulunması aslında bir hiperparametre optimizasyonu işlemidir.

Hiperparametre optimizasyonu, bir makine öğrenmesi algoritması için belirlenen başarı metriğine göre en iyi hiperparametre kombinasyonunun bulunması işlemidir.

- Hiperparametre optimizasyonu ile model karmaşıklığı dengelenerek overfitting (aşırı öğrenme) ve underfitting(az öğrenme) dengesi sağlanabilir.[4]



Resim12- Hiperparametre optimizasyonu

GridSearchCV programlamacı tarafından model hiperparametreler arasındaki belirlenen metriğe göre en başarılı hiperparametre setini belirler.

d-Modellerin en iyi parametre kombinasyonu ile tekrardan eğitilmesi

Linear regresyon algoritmasında oynanacak hiperparametre olmadığı için bu modelde bir değişiklik yapılmamıştır. Random Forest(Rastgele Orman) ve Decision Tree(Karar Ağaçları) ile oluşturulan modeller GridSearchCV ile en iyi hiperparametre seti bulunduğundan sonra tekrar eğitim işlemi yapılmıştır ve yeni modellerimiz elde edilmiştir.

Hiperparametre setini programlamacı her algoritmanın parametre setine göre belirler ve GridSearchCV ile verilen model seti içerisinde en iyisi kombinasyonu bulur.

```
param_grid = {"criterion": ["mse", "mae"],
              "min_samples_split": [10, 20, 40],
              "max_depth": [2, 6, 8],
              "min_samples_leaf": [20, 40, 100],
              "max_leaf_nodes": [5, 20, 100],
              }
dtree_cv_model = GridSearchCV(dtree_reg_model, param_grid, cv=10)
dtree_cv_model.fit(X_train, y_train)
```

Resim13- Decision Tree için Hiperparametre optimizasyonu

```
print("En iyi parametre kombinasyonu:", dtree_cv_model.best_params_)
print("En iyi Skor:", dtree_cv_model.best_score_)

En iyi parametre kombinasyonu: {'criterion': 'mse', 'max_depth': 6, 'max_leaf_nodes': 100, 'min_samples_leaf': 20, 'min_samples_split': 10}
En iyi Skor: 0.956096789898498
```

Resim14- Decision Tree için en iyi parametre seti ve en iyi skor

```
# en iyi hiperparametre kombinasyonunu bulma
rf_params = {'max_depth' : list(range(1,10)),
             'min_samples_split': [3,5,7,10],
             'min_samples_leaf' : [4,5,6],
             'n_estimators': [50,100,500,1000]}
rf_cv_model = GridSearchCV(rf_model, rf_params, cv=10, n_jobs=-1)
rf_cv_model.fit(X_train,y_train)
```

Resim15- Random Forest için Hiperparametre optimizasyonu

```
print("En iyi parametre kombinasyonu:", rf_cv_model.best_params_)
print("En iyi Skor:", rf_cv_model.best_score_)

En iyi parametre kombinasyonu: {'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 1000}
En iyi Skor: 0.9789236520346103
```

Resim16-Random Forest için en iyi parametre seti ve en iyi skor

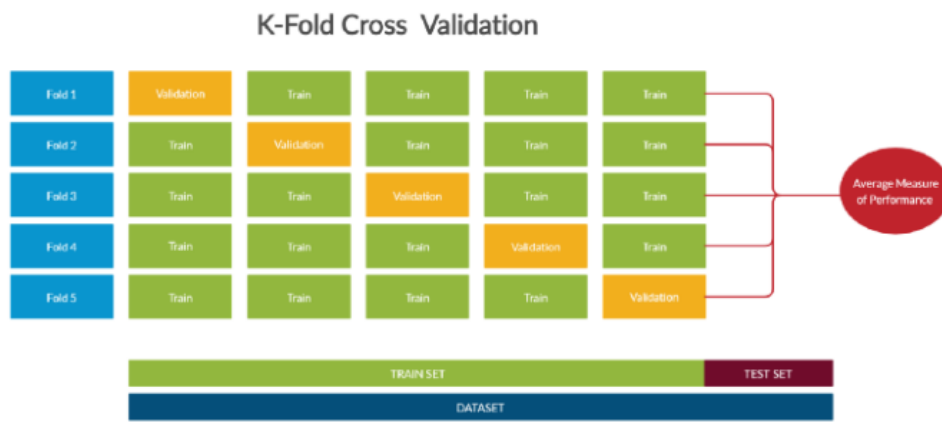
e-Model başarısı hesaplama

Model başarımını hesaplamak için modellerin skorları ölçülmüş ve bu skorların doğruluğunu kanıtlamak cross validation kullanılmıştır.

Cross validation, makine öğrenmesi modelinin görmediği veriler üzerindeki performansının mümkün olduğunca objektif ve doğru bir şekilde değerlendirmek için kullanılan istatistiksel bir yöntemdir.[5]

Örneğin modelimizin başarısını %90 olarak ölçtük ama Kaggle sitesinde bize modelin başarı değerinin %80 olduğunu söyledi. Bunun sebebi elbette ki overfitting(aşırı öğrenme) ve yapacak bir şey yok.

Fakat biz modelimizin başarısını tek bir train-test split işlemiyle yapmak yerine k katlı cross-validation uygularsak bize modelimiz için k farklı doğruluk değerleri elde eder ve ortalama olarak bize modelimizin doğruluk değerini söyler. K katlı cross validation bize modelimizin doğrulanmış, valide edilmiş skor değerini verir. K katlı cross validation ile modelin başarısı sınanır.



Resim17-5 katlı cross validation

```
# Model doğrulama -> Doğrulanmış başarı metriklerimiz
# Doğrulanmış Train başarı metrikleri
kf = KFold(n_splits=10)
train_score = cross_val_score(rf_tuned_model, X_train, y_train, cv=kf, scoring="r2")
train_score_mean = train_score.mean()
train_rmse = np.sqrt(-cross_val_score(rf_tuned_model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error")).mean()
train_mse = - cross_val_score(rf_tuned_model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error").mean()
print("Train score degerleri: ", train_score)
print("Train Mean Score:", train_score_mean)
print("Train Std : " + str(train_score.std()))
print("Train RMSE:", train_rmse)
print("Train MSE:", train_mse)

# Doğrulanmış Test başarı metrikleri
test_score = cross_val_score(rf_tuned_model, X_test, y_test, cv=kf, scoring="r2")
test_score_mean = test_score.mean()
test_rmse = np.sqrt(-cross_val_score(rf_tuned_model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error")).mean()
test_mse = - cross_val_score(rf_tuned_model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error").mean()
print("Test score degerleri: ", test_score)
print("Test Mean Score:", test_score_mean)
print("Test Std : " + str(test_score.std()))
print("Test RMSE:", test_rmse)
print("Test MSE:", test_mse)
```

Resim18-Random Forest K katlı cross validation uygulama

```

# Model doğrulama -> Doğrulanmış başarı metriklerimiz
# Doğrulanmış Train başarı metrikleri
kf = KFold(n_splits=10)
train_score = cross_val_score(model, X_train, y_train, cv=kf, scoring="r2")
train_score_mean = train_score.mean()
train_rmse = np.sqrt(-cross_val_score(model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error")).mean()
train_mse = - cross_val_score(model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error").mean()
print("Train score degerleri: ", train_score)
print("Train Mean Score:", train_score_mean)
print("Train Std : " + str(train_score.std()))
print("Train RMSE:", train_rmse)
print("Train MSE:", train_mse)

# Doğrulanmış Test başarı metrikleri
test_score = cross_val_score(model, X_test, y_test, cv=kf, scoring="r2")
test_score_mean = test_score.mean()
test_rmse = np.sqrt(-cross_val_score(model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error")).mean()
test_mse = - cross_val_score(model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error").mean()
print("Test score degerleri: ", test_score)
print("Test Mean Score:", test_score_mean)
print("Test Std : " + str(test_score.std()))
print("Test RMSE:", test_rmse)
print("Test MSE:", test_mse)

```

Resim19- Linear Regression K Katlı cross validation uygulama

```

# Model doğrulama -> Doğrulanmış başarı metriklerimiz
# Doğrulanmış Train başarı metrikleri
kf = KFold(n_splits=10)
train_score = cross_val_score(dtrees_tuned_model, X_train, y_train, cv=kf, scoring="r2")
train_score_mean = train_score.mean()
train_rmse = np.sqrt(-cross_val_score(dtrees_tuned_model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error")).mean()
train_mse = - cross_val_score(dtrees_tuned_model, X_train, y_train, cv=kf, scoring="neg_mean_squared_error").mean()
print("Train score degerleri: ", train_score)
print("Train Mean Score:", train_score_mean)
print("Train Std : " + str(train_score.std()))
print("Train RMSE:", train_rmse)
print("Train MSE:", train_mse)

# Doğrulanmış Test başarı metrikleri
test_score = cross_val_score(dtrees_tuned_model, X_test, y_test, cv=kf, scoring="r2")
test_score_mean = test_score.mean()
test_rmse = np.sqrt(-cross_val_score(dtrees_tuned_model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error")).mean()
test_mse = - cross_val_score(dtrees_tuned_model, X_test, y_test, cv=kf, scoring="neg_mean_squared_error").mean()
print("Test score degerleri: ", test_score)
print("Test Mean Score:", test_score_mean)
print("Test Std : " + str(test_score.std()))
print("Test RMSE:", test_rmse)
print("Test MSE:", test_mse)

```

Resim20- Decision Tree K katlı cross validation uygulama

Her modelin başarısı k katlı cross validation ile sınılandıktan sonra doğrulanmış başarı skorumuzu elde etmiş oluyoruz.

7) BULGULAR ve TARTIŞMA

Modelin başarısı da sınanıp doğrulanmış modeli elde ettikten sonra tablo1 ve tablo2 deki sonuçlar elde edilmiştir.

Tablo1 Model Skor Tablosu

Model	Model Skoru
Linear Regresyon	0.88
Random Forest	0.93
Decision Tress	0.94

Tablo1'e bakıldığında model skor oranlarına bakıldığında en iyi sonucu veren %94 ile Decision Tree algoritmasına aittir.

Tablo2 Model MSE Değerleri

Model	MSE Değerleri
Linear Regresyon	10.33
Random Forest	5.61
Decision Tree	4.87

- Tablo2'deki MSE değeri ortalama kare hatasıdır. Tahmin değerimiz ile gerçek değerin arasındaki farkların karesinin ortalamasıdır.[6]

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

Resim21- MSE formülü.

Çözümde hangi örüntü tanıma yöntemi neden kullanılmıştır?

Random Forest, Decision Tree ve Linear Regression algoritmaları kullanılmıştır. Denetimli öğrenme ile örüntü tanınması yapılacağı, ürekli veri tahmin edileceği için ve karşılaştırma yapılacağı için bu yöntemler kullanılmıştır.

Önerilen modelin testi nasıl yapılmıştır, hangi ölçekler ile ne sonuçlar alınmıştır?

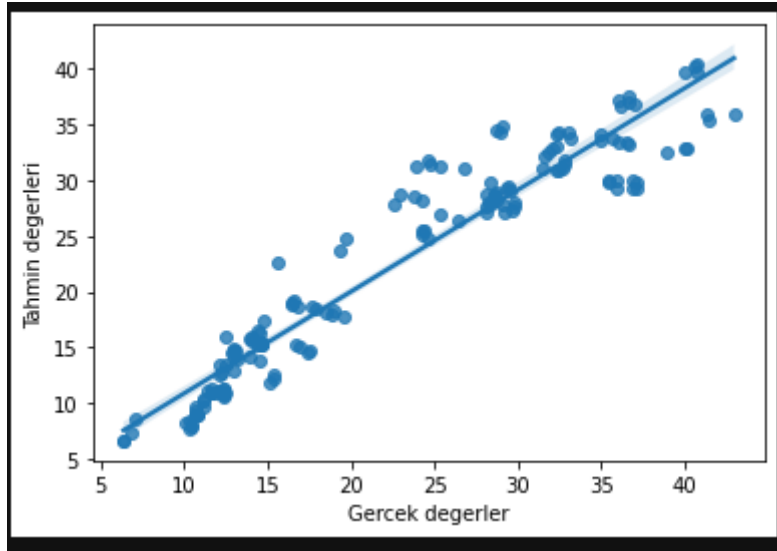
Veri seti %80 ve %20 olarak ayrılmıştır. Model başarıları test verisi ile test edilmiştir. Başarı metrikleri olarak r2_score değeri ve mean_squared_error değerleri ölçek alınmıştır.

Çözüm ve önerilen model başarılı mıdır? Yorumlayınız.

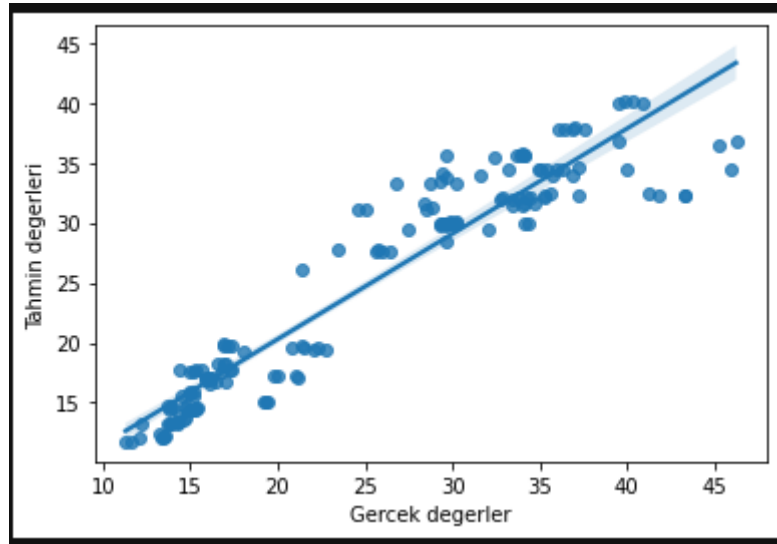
Karşılaştırma yapılarak 3 model denenmiş ve hepsi %85 in üzerindedir ancak en iyi performans Decision Tree algoritması ile yakalanmıştır. Yeni verilerle kullanılabilir.

Çözüme yönelik önerilen modeli destekleyici görseller üretilebilir mi? Ekleyiniz.

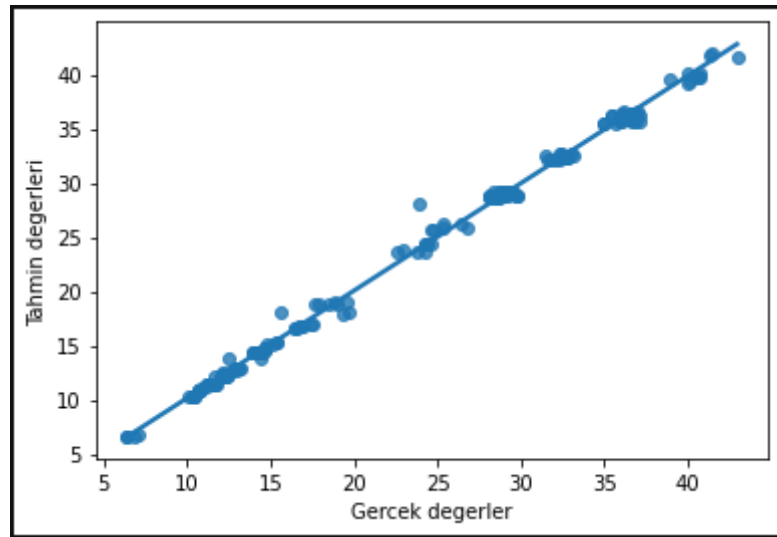
Üretilebilir. Regresyon grafikleri ile görselleştirilmiştir.



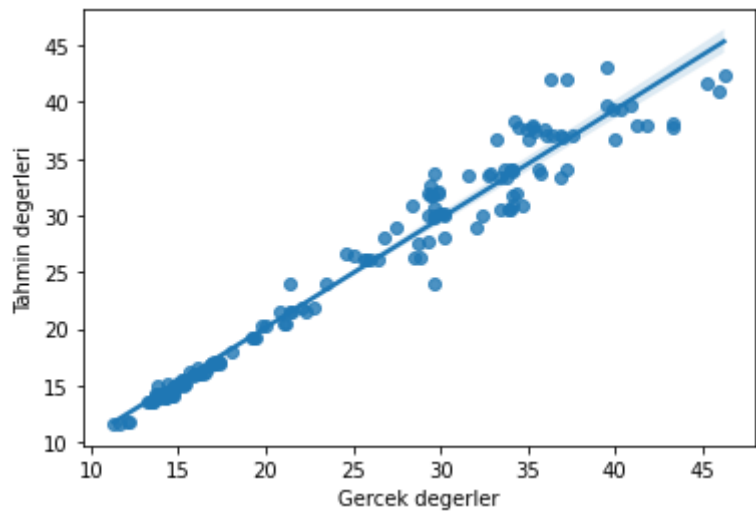
Resim22-Linear Regresion ile ısıtma tüketiminin tahminlenmesi grafiği



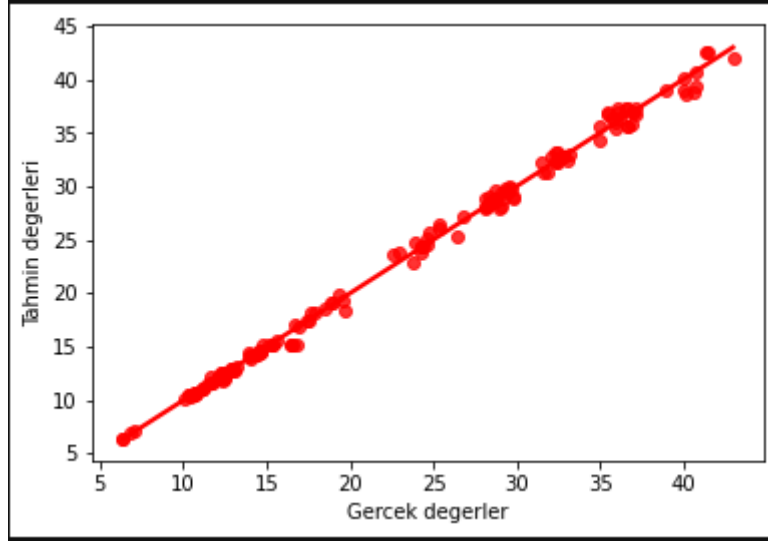
Resim23-Linear Regresion ile soğutma tüketiminin tahminlenmesi grafiği



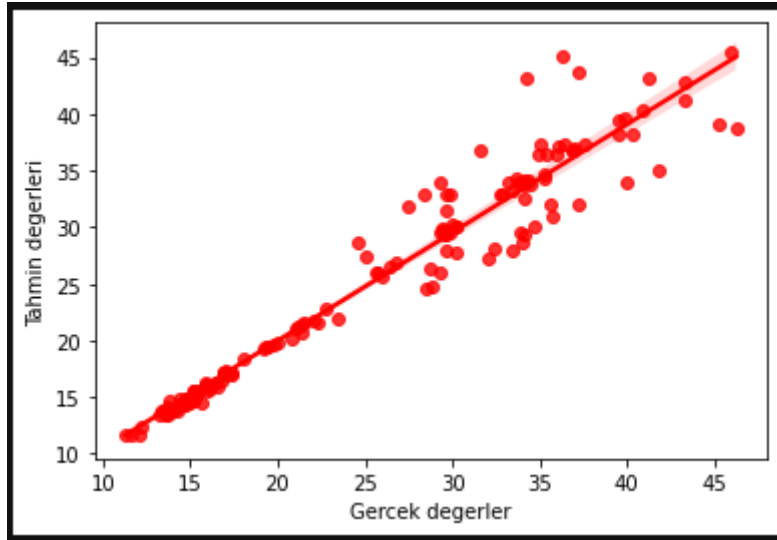
Resim24-Random Forest ile ısıtma tüketiminin tahminlenmesi grafiği



Resim25-Random Forest ile soğutma tüketiminin tahminlenmesi grafiği



Resim25-Decision Tree ile ısıtma tüketiminin tahminlenmesi grafiği



Resim26-Decision Tree ile soğutma tüketiminin tahminlenmesi grafiği

8.SONUÇ

Bu çalışmada 12 farklı bina tipine göre elde edilen verilere bakarak binanın ısıtma ve soğutma yükünü tahmin edilmektedir. Çalışmanın genel amacı binanın mimarı yapısı ve konumuna göre özelliklere bakılarak binanın enerji tüketimini tahmin etmek ve çevreye verdiği zararı minimuma en aza indirmeyi amaçlamaktadır. Amaç binanın en iyi mimarı ve konum değerlerini belirleyerek enerji tüketim yükünü azaltmak amaçlanmaktadır.

Bu projeye göre temel hedefler;

- Binanın güneşe göre konumlandırılması ve şekillenmesi, açıklıkların buna göre düzenlenmesi ve doğal ışığın etkin kullanımı
- Enerjinin ve kaynakların kullanımında verimlilik
- Binaya bütünleştirilmiş pasif ve aktif ısıtma / soğutma sistemleri
- Alan içerisinde enerjinin toplanması ve üretilmesi

Bu hedefler doğrultusunda bina içerisine enerjiyi verimli kullanmak amaçlanmıştır.

KAYNAKLAR

1. https://tr.wikipedia.org/wiki/%C4%B0klim_de%C4%9Fi%C5%9Fikli%C4%9Fi
2. <https://www.filliboya.com/eko-rehber-detay/yasanabilir-bir-dunya-icin-enerjinizi-verimli-kullanin.html>
3. <https://archive.ics.uci.edu/ml/datasets/energy+efficiency>
4. <https://medium.com/bili%C5%9Fim-hareketi/hiperparametre-optimizasyonu-9ba0e7f32e6f>
5. <https://medium.com/bili%C5%9Fim-hareketi/cross-validation-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-4ec4736e5142>
6. <https://www.veribilimiokulu.com/bir-bakista-k-fold-cross-validation/>