ENM421 – VERİ BİLİMİNE GİRİŞ

EMLAK PİYASASI ANALİZİ VE MAKİNE ÖĞRENMESİ İLE KONUT FİYATI TAHMİNLEME

Güz 2023

Ezgi ALTINTOP

Hatice EFLATUN

Ümmü SAVRAN

Yağmur IŞIK

Dr. Öğr. Üyesi Zeliha ERGÜL AYDIN

Eskişehir Teknik Üniversitesi

Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü

Eskişehir

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

1. GİRİŞ

Günümüzde, emlak sektörü büyük bir dinamizme sahiptir ve sürekli değişen ekonomik, demografik ve sosyal faktörlere bağlı olarak şekillenmektedir. Emlak piyasası, konut alım-satım işlemleri, kira fiyatları ve konut talepleri gibi birçok faktörden etkilenir. Bu karmaşık yapı içinde, konut fiyatlarını anlamak ve tahmin etmek, hem bireysel alıcılar hem de satıcılar, yatırımcılar ve emlak profesyonelleri için kritik öneme sahiptir.

Bu proje, emlak piyasasındaki karmaşık etmenleri analiz etmek ve bu etmenleri kullanarak konut fiyatlarını tahmin etmek amacını taşımaktadır. Bu noktada, makine öğrenimi teknikleri, veri analizi ve istatistiksel yöntemlerin kullanımı, konut piyasasındaki dinamikleri anlamak ve gelecekteki konut fiyat hareketlerini öngörmek için güçlü araçlar sunmaktadır.

1.1. Projenin Amacı Ve Kapsamı

Bu çalışmanın temel odak noktası, emlak piyasasında konut fiyatları üzerine analiz ve makine öğrenimi tekniklerinin uygulanmasıdır. Bu amaç doğrultusunda, projenin ilk aşamasında kullanılan veri seti detaylı bir inceleme ve analiz sürecinden geçirilmiştir. Veri setinde yer alan çeşitli faktörlerin dağılımı, istatistiksel özellikleri ve bu faktörler arasındaki korelasyonlar detaylı bir şekilde değerlendirilmiştir.

Veri setinin analizi sonrasında, makine öğrenimi modelinin daha etkili çalışabilmesi için gerekli önişleme adımları uygulanmıştır. Aykırı değer temizleme, eksik veri değerlerinin doldurulması ve özellik mühendisliği gibi önişleme teknikleri, veri setinin hazırlanmasında kullanılmıştır.

Daha sonra, konut fiyatı tahminleme aşamasında çeşitli makine öğrenimi algoritmaları denenmiştir. Linear Regression, DecisionTreeRegressor, RandomForestRegressor ve KNeighborsRegressor komşu algoritmaları, doğruluk oranları üzerinden değerlendirilmiştir. En yüksek doğruluk oranına sahip olan algoritmanın belirlenmesiyle, bu algoritma yeni bir veri seti üzerinde konut fiyatı tahminleme işlemi için seçilmiştir.

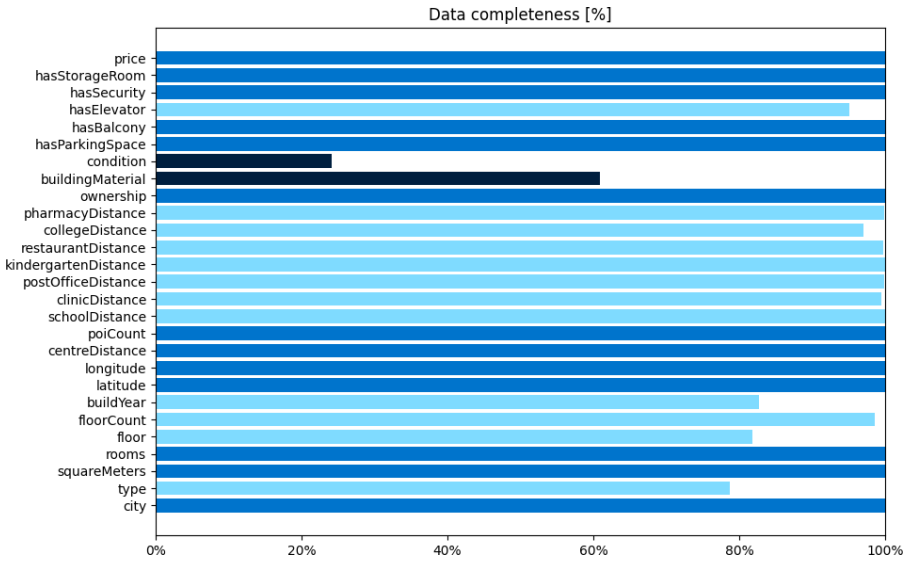
Son aşamada, belirlenen en iyi algoritma kullanılarak yeni veri seti üzerinde konut fiyatı tahminlemesi gerçekleştirilmiştir. Bu adım, modelin genelleme yeteneğini ve gerçek dünya verileriyle başa çıkma kabiliyetini değerlendirmek adına önemlidir.

2. YÖNTEMLER

2.1. Keşifsel Veri Analizi

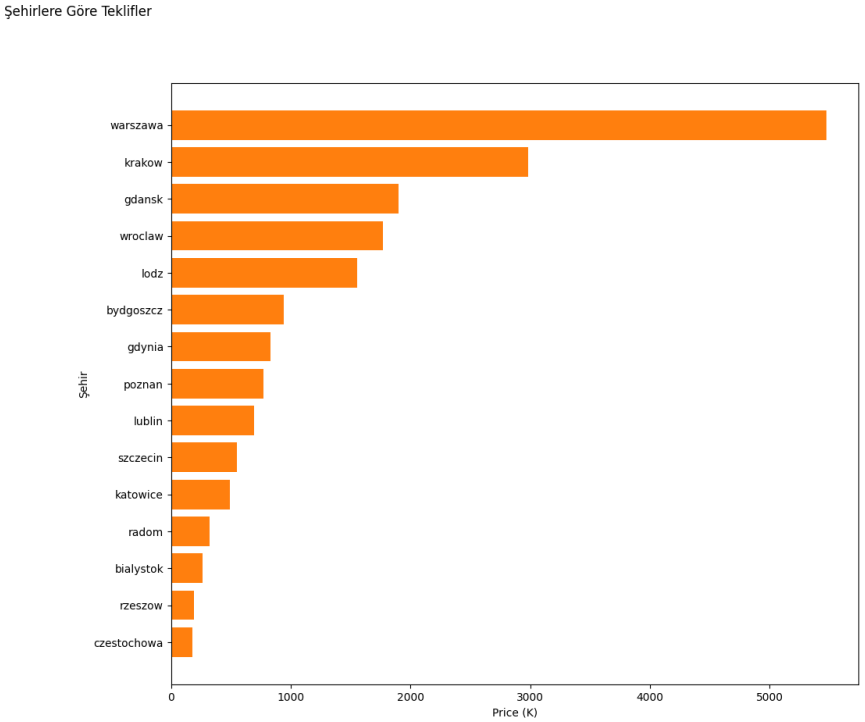
Kullanılan veri seti Polonya'da konut fiyatlarını ve özelliklerini içermektedir. Veri seti konutların metrekare, oda sayısı, kat bilgisi, önemli konumlara uzaklığı, fiyat vb. özelliklerini içermektedir. Veri seti en başta 18905 satır ve 28 sütundan oluşmaktadır Keşifsel veri analizi ile veri setinin genel istatistikleri incelenmiştir. Veri seti kullanılarak ortalama, medyan, standart sapma gibi istatistiksel değerlerle veriyi tanımlanmıştır. Scatter plotlar, histogram, korelasyon matrisi ve heatmap ile görselleştirmeler kullanılarak özellikler arasındaki ilişkiler incelenmiştir. Keşifsel veri analizi ile fiyatı etkileyen önemli faktörler belirlenmiştir. Yapılan analizler aşağıda adım adım verilmiştir.

1. Veri seti yüklendi ve veri setindeki öznitelikler incelendi.
2. Indeks 1'den başlatıldı.
3. Veri setindeki boş değerler ve bunların veri seti içinde kapladığı yüzdece alan saptandı:



Şekil 0.1 *Veri bütünlüğü*

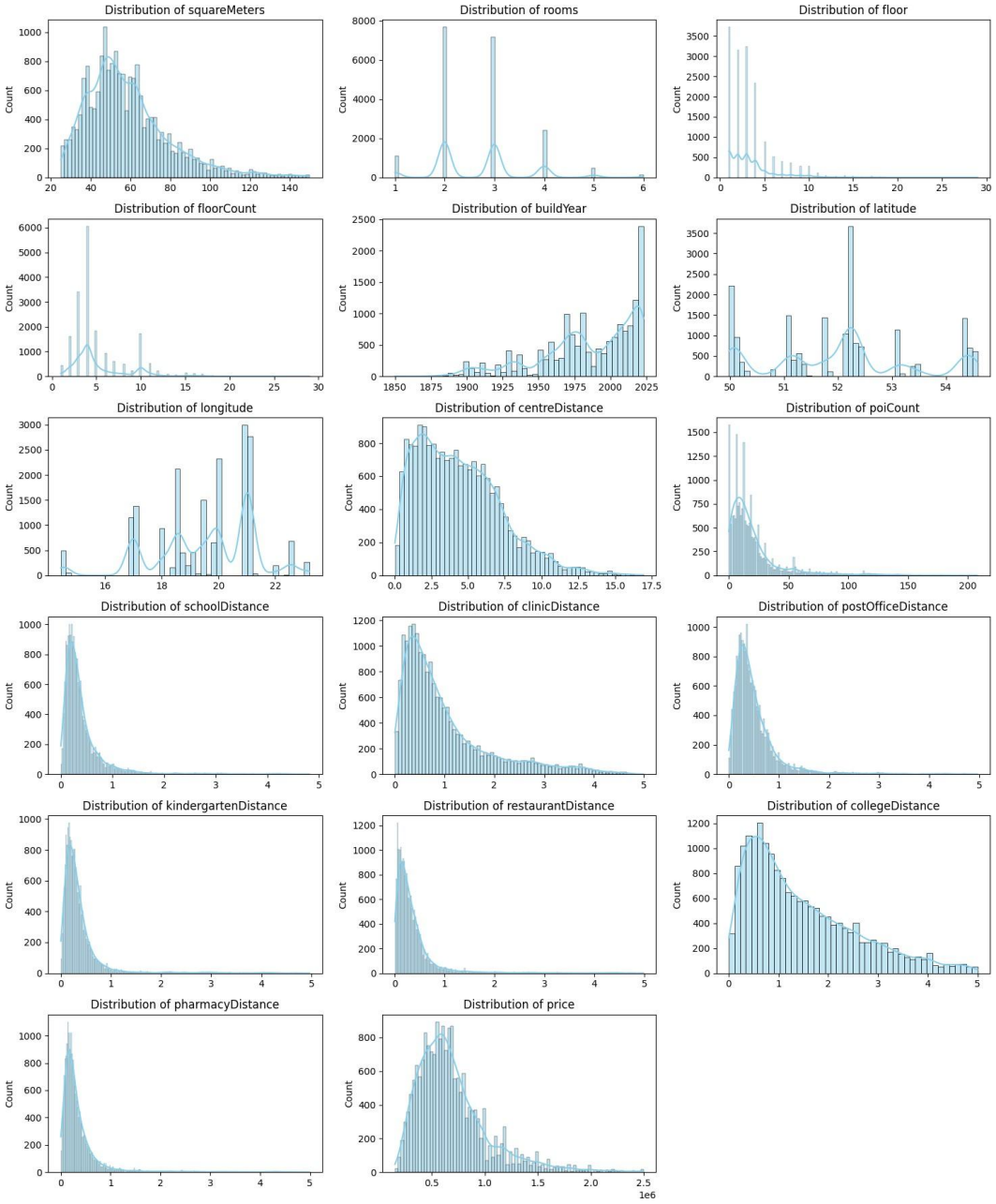
4. Şekil 2.1'de görüldüğü gibi "condition" ve "buildingMaterial" özniteliklerindeki boş değerlerin yüzdelikleri yüksektir. "condition"daki boş değerler %75, "buildinMaterial"daki boş değerler ise %39 oranında yer kaplamaktadır; genel bir varsayım olarak sütundaki boş değerler %75'ten büyük ise silinebilmektedir, bu bağlada "condition" sütunu veri ön işleme adımında silinmiştir.
5. Veri setindeki şehirlere göre fiyat teklifinin gösterimi için histogram grafiği oluşturuldu:



Şekil 0.2 *Şehirlere göre fiyat teklifleri*

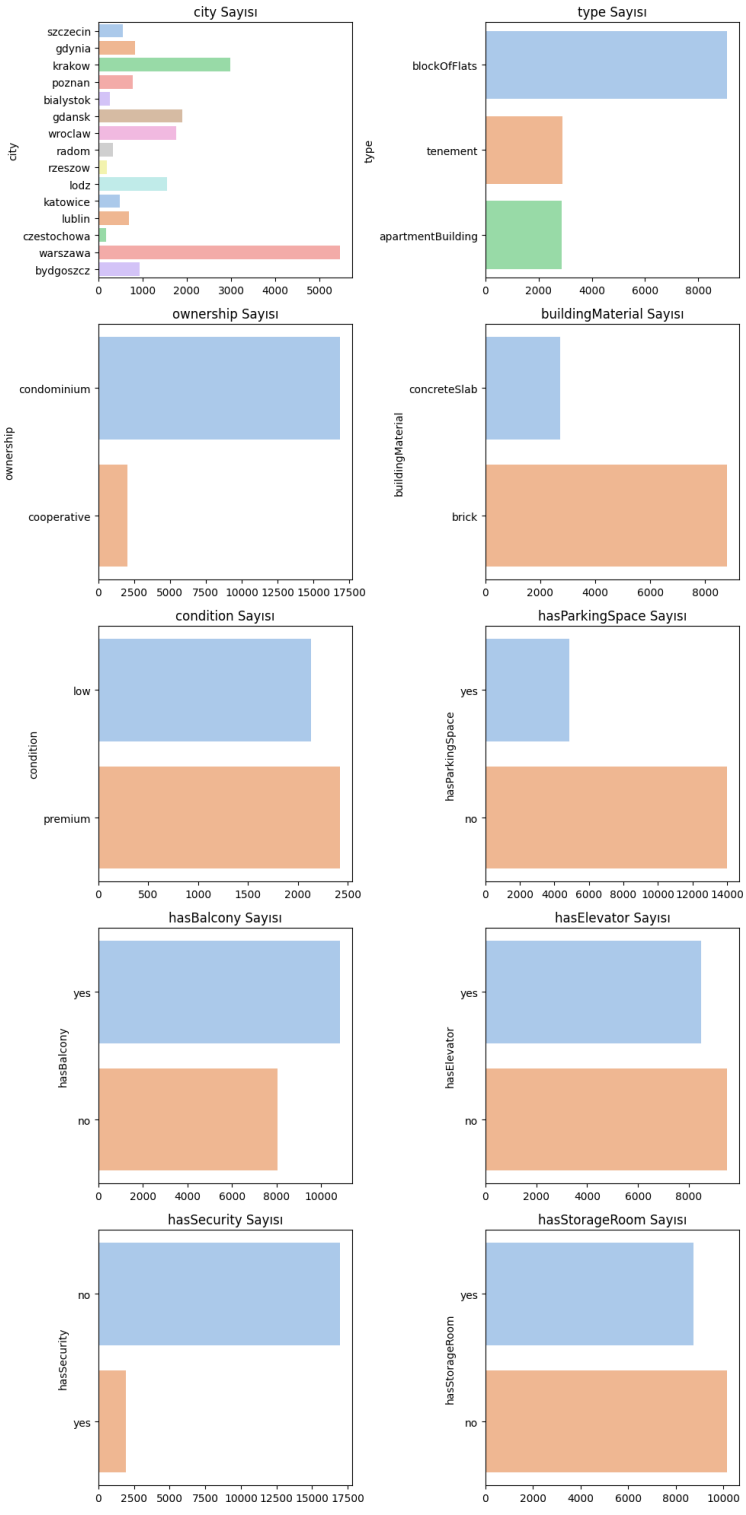
Şekil 2.2'de görüldüğü gibi "Warszawa" en yüksek fiyat tekliflerini alırken "Czestochowa" en düşük fiyat tekliflerini almıştır.

6. Nümerik değerler için dağılım grafikleri çizilmiştir;



Şekil 0.3 *Nümerik değerler için dağılım grafikleri*

7. Kategorik değerler için sayım grafikleri çizilmiştir;



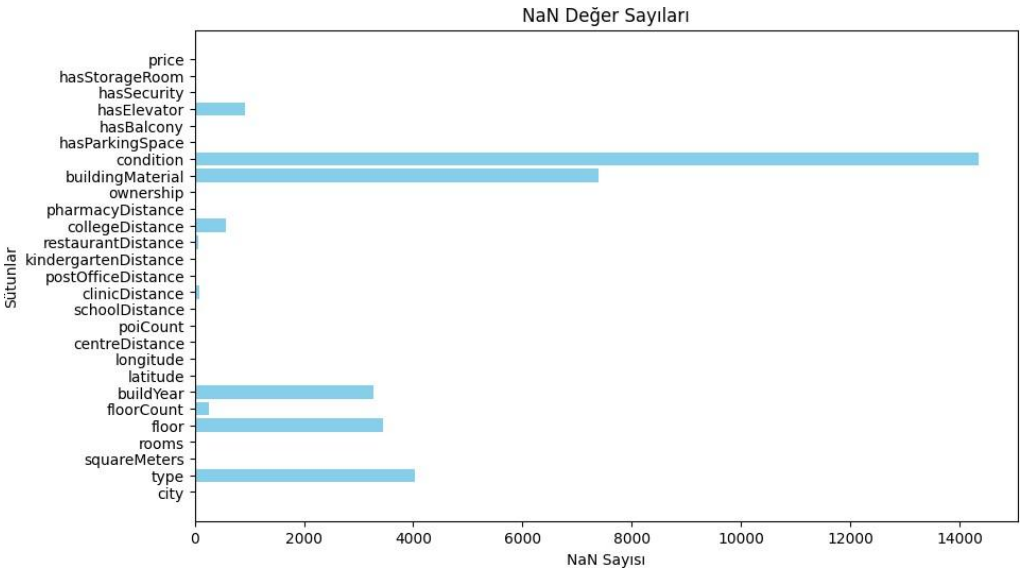
Şekil 0.4 *Kategorik değerler için sayım grafikleri*

2.2. Veri Temizleme ve Ön İşlem

Proje kapsamında veri temizleme adımında eksik değerler uygun yöntemlerle doldurulmuş, aykırı değerler düzeltilmiş ve kategorik veriler nümerik verilere dönüştürülmüştür. Veri seti modeli eğitmek için eğitim ve test

verisi olarak ayrılmıştır. Veri temizleme ve ön işleme adımı aşağıda adım adım açıklanmıştır.

1. Öncelikle veri setinde özniteliklere ait “nan” yani boş değer ve bunların sayısı saptanmış, ilgili grafik aşağıad verilmiştir.

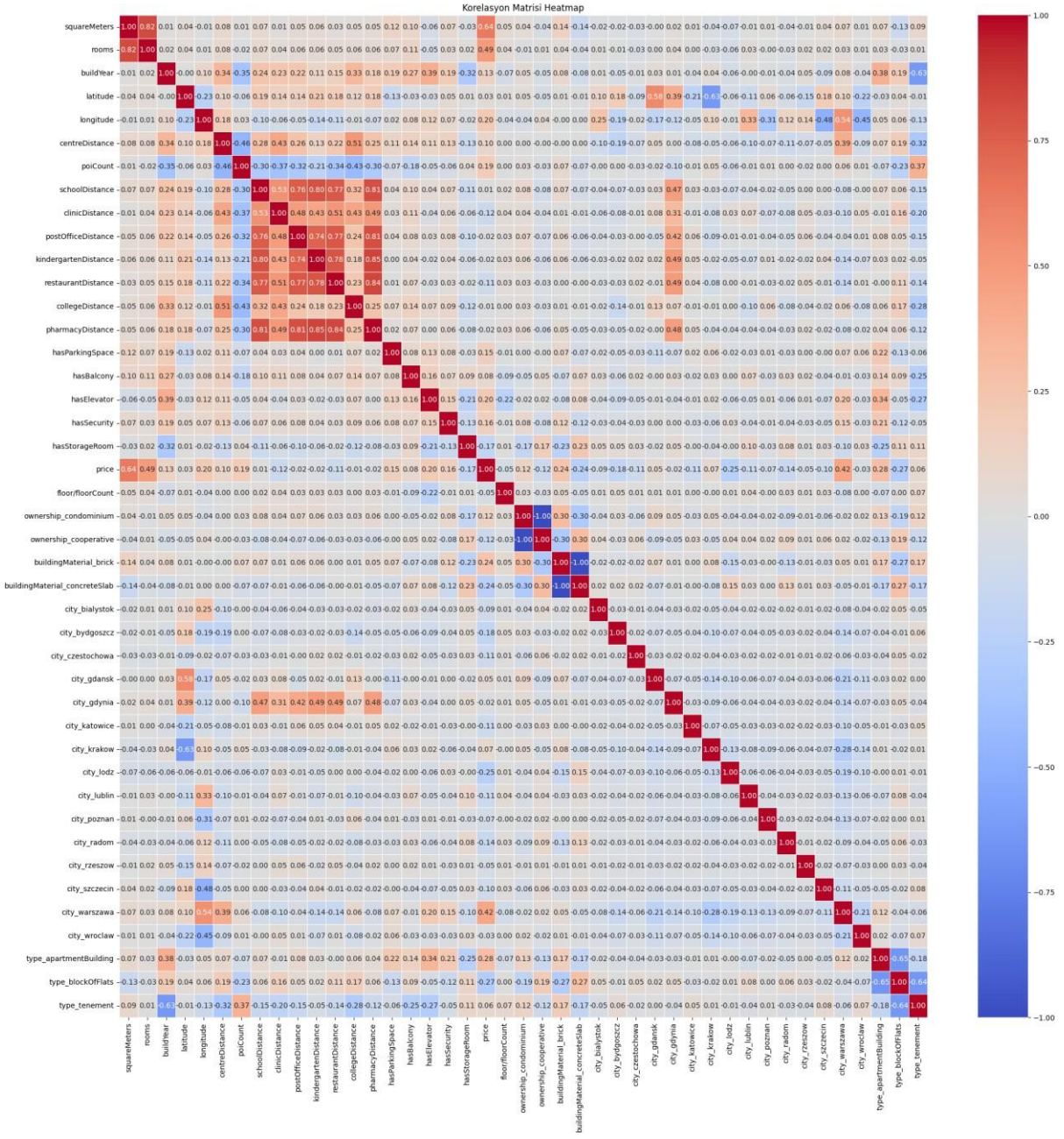


Şekil 0.5 *NaN değerlerin herbir öznitelikteki toplam sayısı*

2. “condition” sütunundaki boş değerler %76 olarak bulunduğu için sütun silindi.
3. “buildigMetarial” için değişim katsayısı hesaplandı, yüksek olduğu için boş değerler medyan değeri ile dolduruldu.
4. “type” ve “hasElevator” sütunundaki boş değerler en sık kullanılan değerler ile dolduruldu.
5. “floor ” ve “floorCount” birbiriyle bağımlı değişkenler , öncelikle ikisi için de değişim katsayısı hesaplandı. “floor” için değişim katsayısı 76 , “floorCount” için ise 63 bulundu , her iki öznitelik için de boş değerler medyan değerleri ile dolduruldu. Floor yani katsayısı, floorCount(toplam katsayısı)’ndan büyük olamayacağı için bu koşulu sağlamayan satırlar silindi.
6. Kalan diğer nümerik öznitelikler 'buildYear', 'schoolDistance', 'clinicDistance', 'postOfficeDistance', 'kindergartenDistance', 'restaurantDistance', 'collegeDistance', 'pharmacyDistance' için tekrar değişim katsayısı hesabı yapıldı. Sadece BuildYear

özniteliğinin değişi katsayısı 1.68 olarak bulundu ve buradaki boş değerler ortalama ile dolduruldu. Diğerleri ise medyan ile dolduruldu.

7. “floor” ve “floorcount ” birbirleriyle bağımlı olduğu için ikisinin oranı alındı ve “floor/floorcount” adında yeni bir öznitelik eklendikten sonra floor ve floorcount sütunları silindi.
8. Boş veriler temizlendikten ve doldurulduktan sonra özniteliklerin sayısallaştırılması adımına geçildi.”hasBalcony”, hasElevator“,”hasSecurity”,”hasStorageRoom”,”hasParkşngSpac e” öznitelikleri evet ve hayırlardan oluşmaktadır. Bu yüzden bu öznitelikler 1 ve 0’lara dönüştürüldü; “ownership”,”buildingMaterial”,”city” ve “type” sütunlarıdaki veriler nominal olduğu için “one-hot-encoding” ile sayısallaştırıldı. Bu adımların sonucunda boş olan tüm değerler doldurulmuş ve sayısallaştırılmış oldu.
9. Bağımlı ve bağımsız değişkenler arasındaki ilişkiyi incelemek için kolerasyon matrisi kullanılmıştır. Korelasyon matrisi heatmap'i, veri setindeki değişkenler arasındaki ilişkileri renk tonları ile görsel olarak ifade etmek için kullanılır. Heatmap, yüksek pozitif korelasyonların belirgin bir renk tonuyla, yüksek negatif korelasyonların ise farklı bir renk tonuyla vurgulanmasına olanak tanır. Bu görselleştirme aracı, değişkenler arasındaki güçlü ilişkileri, zayıf veya yok denebilecek korelasyonları, ve multikolinerite işaret eden durumları hızlıca değerlendirilmesine yardımcı olur.



Şekil 0.6 *Kolerasyon matrisi- heatmap*

Oluşturulan kolerasyon matrisine göre değişkenlerin “price” ile kolerasyonları azalan sırada aşağıdaki gibidir;

Tablo 0.1 *Değişkenlerin "price" ile olan korelasyonları*

Değişken	**Korelasyon**
price	1
squareMeters	0.636267813
rooms	0.489866755
city_warszawa	0.422718741
type_apartmentBuilding	0.280074315
type_blockOfFlats	0.269468365
city_lodz	0.245782066
buildingMaterial_brick	0.241767019
buildingMaterial_concreteSlab	0.241767019
longitude	0.201258453
hasElevator	0.199940905
poiCount	0.186528561
city_bydgoszcz	0.18063525
hasStorageRoom	0.167579844
hasSecurity	0.156300046
hasParkingSpace	0.150463074
city_radom	0.140037102
buildYear	0.127836674
clinicDistance	0.121671612
ownership_cooperative	0.121425415
ownership_condominium	0.121425415
city_lublin	0.113635827
city_katowice	0.112839084
city_czestochowa	0.106851999
restaurantDistance	0.106803784
city_szczecin	0.097257753
centreDistance	0.097110748
city_bialystok	0.086055888
hasBalcony	0.077693994
city_poznan	0.072083411
city_krakow	0.068854424
type_tenement	0.064141379
city_rzeszow	0.053054456
floor/floorCount	0.051280313
city_gdansk	0.049899171
city_wroclaw	0.026598112
latitude	0.026307227
city_gdynia	0.021643096
kindergartenDistance	0.019506049
postOfficeDistance	0.019412478
pharmacyDistance	0.018917705
collegeDistance	0.008374568
schoolDistance	0.006133922

Tablo 2.1’de görüldüğü gibi 'kindergartenDistance', 'postOfficeDistance', 'pharmacyDistance', 'collegeDistance', 'schoolDistance' özniteliklerinin ‘price’ ile korelasyonu düşüktür. Model eğitimi aşamsında bu öznitelikler çıkarılacaktır.

Modeller oluşturulmadan önce öznitelikler standartlaştırılmıştır. ‘squareMeters', 'rooms','buildYear', 'latitude', 'longitude', 'centreDistance', 'poiCount', 'schoolDistance', 'clinicDistance', 'postOfficeDistance',

'kindergartenDistance', 'restaurantDistance', 'collegeDistance', 'pharmacyDistance' öznitelikler için bu işlem gerçekleştirilmiş , diğer öznitelikler ise 0 ile 1 arasında ölçeklendirilmiş oldukları için standartlaştırma işlemi uygulanmamıştır.

Standartlaştırma, bir veri setindeki özellikleri aynı ölçekte ifade etmek için kullanılan bir ön işleme yöntemidir. Bu işlem, farklı birimlere veya ölçeklere sahip olan özellikleri standart normal dağılıma getirir, yani ortalama değeri 0, standart sapması ise 1 olacak şekilde dönüştürür. Bu sayede, modelin eğitilmesi ve performansının değerlendirilmesi daha tutarlı hale gelir. Standartlaştırma, farklı büyüklükteki özellikler arasındaki etkileşimi düzeltir, modelin aykırı değerlere karşı direncini artırır ve genelleme yeteneğini iyileştirir. Ayrıca, gradient tabanlı algoritmaların daha etkili çalışmasına olanak sağlar. Bu nedenlerle, standartlaştırma, bir veri setini modellemeye hazırlamak ve daha güvenilir sonuçlar elde etmek için yaygın olarak kullanılan bir uygulamadır.

Tüm veri ön işleme adımlarından sonra kullanılan verinin boyutu 18306 satır ve 26 sütun olarak güncellenmiştir.

2.3. Kullanılan Makine Öğrenmesi Algoritmaları

Proje kapsamında konut fiyatlarını tahmin etmek için 4 farklı algoritma kullanılmış ve sonuçları birbiri ile karşılaştırılmıştır. Bu algoritmalar: Linear Regression, Decision Tree Regressor, Random Forest Regressor ve KNeighbors Regressor algoritmalarıdır.

2.3.1. Linear Regression

Regresyon analizi, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi modellemek için kullanılır. Linear Regression, bu ilişkiyi doğrusal bir denklemle ifade eder.

Linear Regression, en küçük kareler yöntemini kullanarak veri setine en iyi uyan doğrusal bir çizgi oluşturmaya çalışır. Bu çizgi, bağımsız değişkenlerin bağımlı değişken üzerindeki etkilerini temsil eder [1].

2.3.2. Decision Tree Regressor

Decision Tree Regressor, sınıflandırma ve regresyon problemlerini çözmek için kullanılır. Decision Tree Regressor, veriyi özelliklerine göre sınıflandıran veya değer tahmini yapan bir ağaç yapısıdır.

Decision Tree Regressor, veri setini bölerek homojen alt gruplar oluşturan bir dizi karar kuralı içerir. Bu alt gruplar, sonuçları tahmin etmek veya sınıflandırmak için kullanılır. Ağacın en üstünde bulunan kural, veri setini en iyi şekilde bölen birinci kuralı temsil eder [2].

2.3.3. Random Forest Regressor

Random Forest Regressor, sınıflandırma ve regresyon problemleri için kullanılan bir ensemble (bir araya getirme) algoritmasıdır. Birden çok karar ağacının bir araya getirilmesiyle oluşur.

Random Forest Regressor, birçok karar ağacının bağımsız olarak eğitildiği ve her bir ağacın tahminlerinin bir araya getirildiği bir ensemble algoritmasıdır. Bu, tek bir ağacın aşırı uydurma eğilimini azaltır ve daha genel geçer modeller elde etmeye yardımcı olur [3].

2.3.4. K Neighbors Regressor

K Neighbors Regressor algoritması, sınıflandırma ve regresyon problemleri için kullanılan basit bir algoritmadır. Tahminlerini, çevresindeki en yakın komşuların etkisiyle yapar.

K Neighbors Regressor, veri setinde bir noktayı sınıflandırmak veya tahmin yapmak için, bu noktanın çevresindeki k en yakın komşusunu belirler. Daha sonra, bu komşuların sınıfları veya değerleri kullanılarak tahmin yapılır [4].

Bu algoritmalar, farklı kullanım durumlarına ve veri setlerine uygun farklı güçlü yanlara sahiptir. Projede bu dört algoritmayı kullanarak modeller eğitilmiş ve performansları karşılaştırılmıştır.

3. YÖNTEMLERİN UYGULANMASI VE SONUÇ

3.1. Makine Öğrenmesi Modellerinin Uygulanması Ve Performanslarının Karşılaştırılması

Önişlemlerin ardınan K Neighbors Regressor, Decision Tree Regressor, Random Forest Regressor ve Linear Regression makine öğrenmesi algortimaları ile modeller eğitilip test edildi. Sklearn kütüphanesinden gerekli metotlar import edildikten sonra veri setinin %80'i train (eğitim) ve %20'si test veri seti olmak üzere ikiye ayrıldı.

Dört model de öncelikle default hiperparametrelerle eğitildi ve ardından randomizedseach ile belirlenen uzayda hiperparametre araması yapıldı. Randomizedsearch için cross-validation (çapraz doğrulama) değeri tüm modeller için 5, n_iter (iterasyon sayısı) ise 10 olarak belirlendi. Aşağıdaki tabloda randomizedsearch ile hiperparametre araması sonucunda bulunan hiperparametreler verilmiştir.

Tablo 3.1 *Hiperparametre araması sonucu bulunan hiperparametre kombinasyonları*

Modeller	RandomizedSearch ile bulunan hiperparametreler
LinearRegression	{'positive': False, 'fit_intercept': False}
RandomForestRegressor	{'n_estimators': 200, 'min_samples_split': 5, 'min_samples_leaf': 4, 'max_features': 'auto', 'max_depth': None}
DecisionTreeRegressor	{'splitter': 'random', 'min_samples_split': 2, 'min_samples_leaf': 4, 'max_depth': 30, 'criterion': 'friedman_mse'}
KNeighborsRegressor	{'weights': 'distance', 'p': 1, 'n_neighbors': 12}

Hiperparametre aramasıyla bulunan değerlerle modeller eğitildi ve öncesi ve sonrası durum için R2 değerleri hem eğitim hem de test veri seti için aşağıdaki tabloda verilmiştir.

Tablo 3.2 *Modellerin R-kare metriğine göre kıyaslanması*

	LinearRegression	RandomForestRegressor	DecisionTreeRegressor	KNeighborsRegressor
Hiperparametre tuning öncesi R-kare skoru(test veri seti için)	0.75325	0.87935	0.75674	0.81342
Hiperparametre tuning öncesi R-kare skoru(train veri seti için)	0.74738	0.98333	0.99998	0.87106
Hiperparametre tuning sonrası R-kare skoru(test veri seti için)	0.75325	0.87490	0.79079	0.83379
Hiperparametre tuning sonrası R-kare skoru(train veri seti için)	0.74738	0.95881	0.96367	0.99998

Modellerin performansını değerlendirmek için ortalama kare hata, kök ortalama kare hata, ortalama kare hata ve R2 skoru gibi metrikler kullanılmaktadır.

R2 skoru genellikle regresyon modellerinin performansını değerlendirmek için kullanılır ve modelin ne kadar iyi açıklayıcı olduğunu gösterir. R2 skoru, toplam varyansın model tarafından açıklanan varyansa

oranını ifade eder. Değer 1'e ne kadar yakınsa, modelin veriyi o kadar iyi açıkladığı kabul edilir.

Linear regresyon için bakıldığında hiperparametre aramasının hiçbir değişikliğe yol açamadığı (aramanın sonucunda en iyi parametreler olarak default parametreler çıktı olarak alındığı için); RandomForestRegressor için hiperparametre araması sonucunda overfitting (aşırı öğrenme) durumunun biraz düştüğü; DesicionTreeRegressor modelinde hiperparametre araması sonucunda bulunan hiperparametreler ile model oluşturulduğunda overfitting düştüğü ve test veri setindeki skordaki artış; KneighborsRegressor için ise test veri setindeki skorun artışı gözlemlenmiştir.

Tablodaki test veri seti için R2 skorlarına bakıldığında en başarılı sonucu RandomForestRegressor vermiştir.

3.2. Seçilen Modelin Gerçek Dünya Veri Seti Üzerindeki Performansının Değerlendirilmesi

RandomForestRegressor modeli pickle formatıyla kaydedildi. Model eğitimleri 8. ay için yapılmış, bu aşamada 9. ay veri seti üzerinden modelin reel performansı değerlendirilmiştir. Öncelikle tüm ön işlem adımları 9. ay veri seti için de yapılmıştır ve kaydedilen model load() metodu ile yüklenip tahminleme yapılmıştır.Modelin tahminleme performansı R^2 metriği cinsinden 0.8787 olarak bulunmuştur. Gerçek değerler ve tahmin edilen değerler (ilk 10 satır için) aşağıda verilmiştir;

Table 0.3 *Gerçek değerlerin ve tahmin değerlerinin kıyaslanması*

	Gerçek Değerler	**Tahmin Edilen Değerler**
1	489000	637734.6506
2	846000	833839.3677
3	950000	698225.2463
4	1653563	1234699.716
5	339000	376652.0735
6	718539	605468.1354
7	330000	364665.976
8	340505	312865.6306
9	245000	273427.9796
10	307000	330627.7361

KAYNAKÇA

Veri Seti: https://www.kaggle.com/datasets/camnugent/california-housing-prices?select=housing.csv

[1] https://en.wikipedia.org/wiki/Linear_regression

[2] https://en.wikipedia.org/wiki/Decision_tree_learning

[3] Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 200.1

[4] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm