KİRA TAHMİN SİSTEMİ

Ad Soyad: Giray Doruk Yurtseven
Üniversite / Bölüm: Haliç Üniversitesi – Yazılım Mühendisliği
Proje Adı: Kira Tahminleme Sistemi
Seçilen Görev: Görev 4 – Kira Tahminleme Sistemi
Proje Kod Dosyası: https://github.com/giraydorukyurt7/Rent-Estimation-System

İçindekiler:

1-Giriş

Bu çalışmanın amacı, kamuya açık bir konut veri seti kullanılarak Türkiye'deki evlerin kira fiyatlarını makine öğrenmesi yöntemleriyle tahmin etmektir. Gerçek dünya verileri üzerinden modelleme yapılarak hem öznitelik mühendisliği hem de model karşılaştırma yeteneklerinin değerlendirilmesi hedeflenmiştir.

2-Veri Seti

[1]Kullanılan veri seti kaggle üzerinde yayınlanan Turkish House Rent Prediction adlı kamuya açık veri setidir. Boyut ve yapı olarak 2 ayrı parçaya bölünmüş şekilde verilmiştir. Eğitim verisi 6166 gözlem ve test verisi 2642 veriden ve onaltışar adet öznitelikden oluşmaktadır. Çok fazla ön işlemi gerektiği için bu 2 veri seti birleştirilip daha sonra analiz bittikten sonra tekrar split edilerek modellere verilmiştir. Öznitelik olarak veri setinde; Şehir, İlçe, Mahalle, Apartman türü, ev türü, ev yaşı, ev boyutu(m2), Oda sayısı(oda+salon), kat bilgisi, eşya bilgisi, banyo sayısı, kaçıncı el olduğu, ısıtma türü, ısıtma yakıtı ve fiyatı bulunmaktadır. Ekstra olarak gereksiz şekilde bulunan fazladan index sütunu çıkartılmıştır.

3-Veri Analizi (Öznitelik işleme ve görselleştirme)

Veri analizine başlamadan önce test ve train verileri birleştirilip (8808, 16)'lık bir matris elde edilmiştir. Verilerin bir çoğu sayısal bilgi taşımasına rağmen başka formatlarda bulunmaktadır. Bunlar için çok kapsamlı bir veri analizi uygulanmıştır.

3.1 Öznitelik işleme

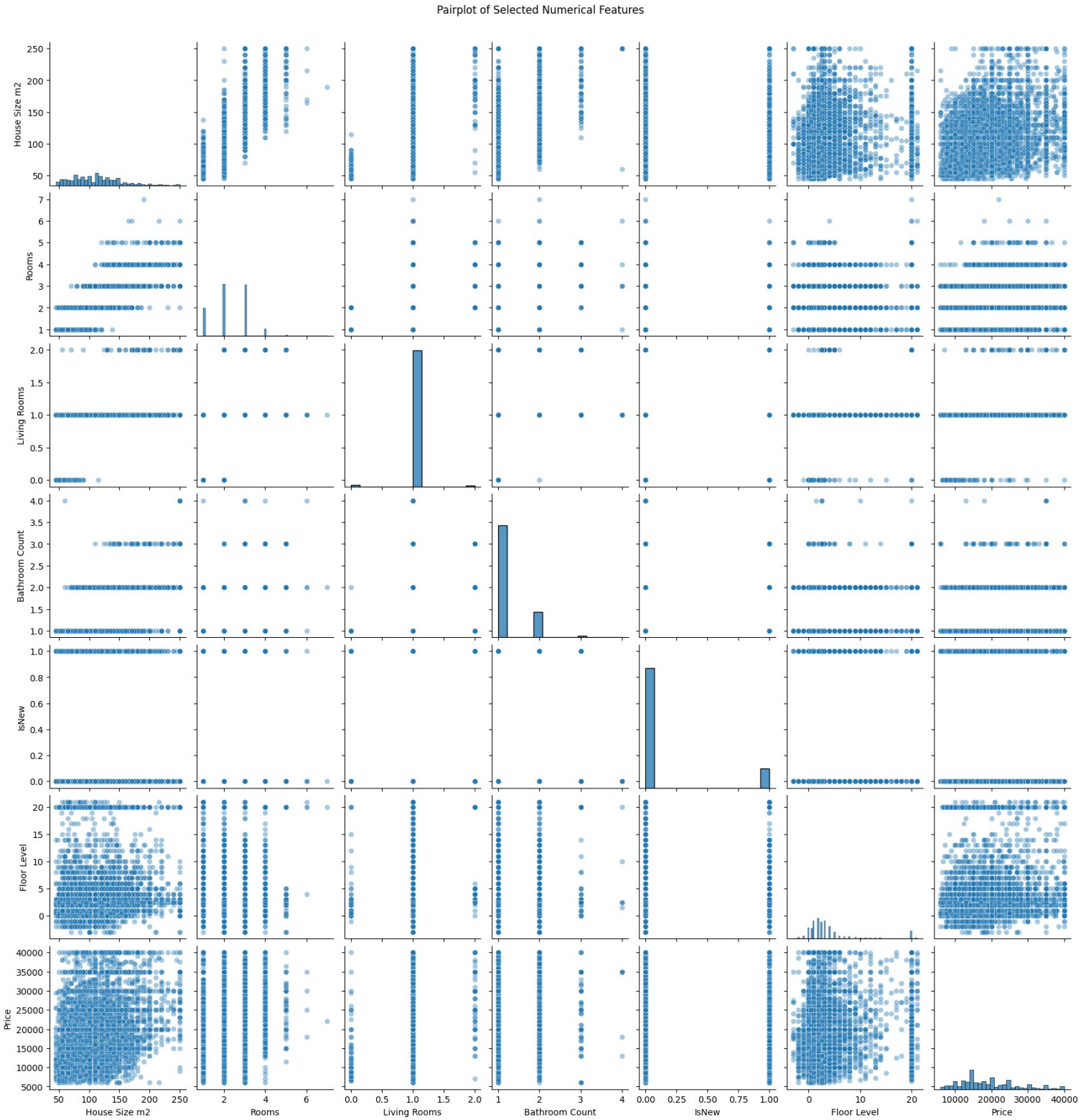
-Oda Sayısı: 3+1, 2+1 gibi bulunan dairelerin önce veri setinde bulunma adetleri incelenmiştir. 41+1 gibi çok nadir bulunan daireler direkt veri setinden atılmıştır. Daha sonra bu sütun "Oda sayısı" ve "salon sayısı" olmak üzere 2 ayrı sütuna bölünmüştür.
-Ev Yaşı: "Sıfır" integer'a dönüştürülmüştür ve 10 taneden az aynı yaşta bulunan veriler(270 yıllık ev gibi.) veri setinden atılmıştır.
-Ev m2: "120 m2" gibi bulunan bu veriler integer'a dönüştürülmüştür. Ve veri setindeki veri miktarlara göre mantıklı olması açısından 45<=m2<=250 olan veriler tutulmuştur.
-Ev Fiyat: Yine veri adetleri incelenip buna göre 6000<=fiyat<=40000 olan veriler tutulmuştur.
-Kat Sayısı: Yine string formatındadır, ayrıca ekstra olarak "ara kat", "yüksek giriş", "yarı bodrum" gibi farklı kat çeşitleri de yer almaktadır. Özel isimli olanlara mantıklı float değerleri verilmiştir. 21 üzerinde olan katlar ise 21 olarak değer almıştır.
-Ev Mahalle, Ev İlçe, Ev Şehir: bu kategorik değişkenlere direkt olarak one-hot-encoding uygulamamız mümkün değildir. Sebebi ise farklı farklı illerde "Merkez", "Atatürk", "Cumhuriyet" gibi aynı isme sahip yerlerin bulunabilmesi sorunudur. Bunun için adresler daha çok özelleştirilip tekrar kaydedilmiştir. Mahalle için "şehir – ilçe – mahalle", ilçe için ise "şehir – ilçe" şeklinde birleştirmeler yapılmıştır. Ayrıca bu bölgeler için mahalle ve ilçelerde 20, şehirde 10 olacak şekilde alt eşik değer konulup, bu değerin altında olanlar "other" olarak etiketlenmiştir.

-Apartman tipi ve ev tipi: Bu sütunların birbirlerine benzerliği dikkat çekmekteydi. Bu sebebten ötürü “crosstab” fonksiyonu uygulanarak farklılık olmadığı gözlemlenmiştir. Bu yüzden bir sütun veri setinden atılmıştır. Ev tipi verisinde ise sonrasında “Dağ evi”, “Yalı Dairesi” gibi çok az veri bulunduran özellikler atılmıştır.
-Isıtma yakıtı: Bu kategoride kömür, odun gibi düşük veri bulunduran veriler veri setinden atılmıştır ve sadece Doğalgaz ile Elektrik bulunduran evler ile devam edilmiştir.
-Isıtma türü: 15 farklı ısıtma türünden en çok veri bulundurmuş olan "Kombi", "Klima", "Merkezi", "Merkezi (Pay Ölçer)", "Yerden Isıtma","Isıtma Yok" özellikleri seçilmiştir.
-Kaçıncı el olduğu bilgisi: Bilinmeyenler(NaN), ikinci el olarak atanmıştır. Betonarme özelliği atılmıştır. Özellik olarak sadece ikinci el ve sıfır daireler seçilmiştir. Sıfır olmak değer katacağı için ve model eğitimine daha uygun olduğu için sütun adı “Yeni mi” diye değiştirilip “Sıfır”:1 ve “İkinci El”:0 şeklinde atanmıştır.
-Eşya bilgisi: NaN olanlar False olarak atanmıştır.
-Banyo sayısı: adet<=4 olacak şekilde filtrelenmiştir.

3.2 Korelasyon Analizi:

Sayısal değişkenlerin Fiyata ile ilişkisi	Price(Fiyat)
Price (Fiyat)	1.000000
Bathroom Count (Banyo Sayısı)	0.332318
House Size m2 (Ev boyutu m2)	0.316582
Rooms (Oda sayısı)	0.283462
Living Rooms (Salon Sayısı)	0.121039
Floor Level (Kat)	0.086650
House Age (Evin yaşı)	0.039236
IsNew (Yeni mi?)	0.015240

3.3 Pairplot ile Dağılım Grafiklerinin Görselleştirilmesi



3.4-Son Veri Ön İşlemeler

Modele verilebilmeleri için kategorik değişkenler için one-hot encoding uygulanmıştır. Bu sayede son veri matrisi boyutu yapılan diğer tüm veri ön işleme çalışmalarıyla birlikte (6624, 178) olmuştur. Sonrasında modelde kullanılmak üzere train_test_split fonksiyonu kullanılarak %80'e %20 olacak şekilde tekrardan bölünmüştür ve train setinde 5299, test setinde 1325 gözlem olacak şekilde eğitime başlanmıştır.

4-Kullanılan Algoritmalar ve Ensemble Yöntemler

4.1-Algoritmalar:

Bu çalışmada farklı yaklaşım ve özelliklere sahip yedi adet regresyon algoritması test edilmiştir: KNN, SVR, DecisionTree, RandomForests, GBM, XGBoost, LightGBM. Tüm algoritmalarda hiperparametre seçimi için GridSearchCV ile detaylı bir tarama yapılmış ve her model, eğitim sonrası RMSE ve MAPE metriklerine göre değerlendirilmiştir.

4.2-Ensemble Yöntemler:

Tekil modellerin ötesine geçmek amacıyla iki farklı ensemble yöntemi uygulanmıştır: **Averaging** ve **Stacking**.

1- Averaging (Model Ortalaması)

Bu yöntemde birden fazla modelin tahminleri alınır ve ortalamaları hesaplanarak final sonuç elde edilir. Bu yöntem basit ancak genellikle yüksek doğrulukta tahmin sağlar, özellikle modellerin hataları birbirinden farklı ise.

2-Stacking (Üst Modelleme)

Stacking yöntemi, alt modellerin tahminlerini alır ve bunları bir meta-model ile tekrar öğrenir. Bu projede alt modellerin çıktıları LinearRegression ile ikinci bir model olarak yeniden tahminlenmiştir. Yani:

1. Alt modeller X_train'de eğitilir ve X_test için tahmin yapar.
2. Bu tahminler yeni bir veri seti oluşturur.
3. Meta-model bu tahminleri alıp gerçek y_test ile eğitilir ve final tahmini üretir.

4.3 Kombinasyonlar ve Deneme Sayısı

Bu projede yalnızca sabit bir ensemble yapı değil, 3 ve daha fazla modellerin **olası tüm kombinasyonları** değerlendirilmiştir:

7 model ve 2 çeşit ensemble yöntemi için $2*[C(7,3)+C(7,4)+C(7,5)+C(7,6)+C(7,7)]=198$ ensemble model.

5-Sonuçların Analizi

Model	RMSE	MAPE	Best Params	Eğitim süresi (s)
XGBoost	4744.075053	0.203971	{'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 200, 'reg_alpha': 0.1, 'subsample': 0.8}	560.906501
GBM	4827.084521	0.209387	{'learning_rate': 0.1, 'loss': 'squared_error', 'max_depth': 7, 'max_features': 'sqrt', 'n_estimators': 200, 'subsample': 0.8}	191.147500
RandomForest	4786.827367	0.210086	{'bootstrap': False, 'criterion': 'squared_error', 'max_depth': None, 'max_features': 'sqrt', 'min_samples_split': 5, 'n_estimators': 200}	322.959500
LightGBM	4953.731574	0.218108	{'feature_fraction': 0.8, 'learning_rate': 0.2, 'max_depth': 4, 'n_estimators': 200, 'reg_alpha': 0.1, 'reg_lambda': 0, 'subsample': 0.8}	61.733998
SVR	5398.246758	0.228871	{'C': 100, 'epsilon': 0.1, 'gamma': 'scale', 'kernel': 'linear'}	287.153000
DecisionTree	5899.696648	0.260703	{'criterion': 'squared_error', 'max_depth': 10, 'max_features': None, 'min_samples_leaf': 4, 'min_samples_split': 10}	16.535501
KNN	6419.426173	0.292046	{'n_neighbors': 11, 'p': 1, 'weights': 'distance'}	35.341001

Tekil model eğitimleri sonucunda en iyi model 0.2039 ile XGBoost çıkmıştır.
Ensemble modellerden en iyi ilk 10 ve sıralamadaki ilk averaging ensemble yapan model en iyiden başlayarak:

Ensemble Type	**RMSE**	**MAPE**	**Models used**
Stacking	4657.24	0.1994	DecisionTree + RandomForest + XGBoost
Stacking	4661.00	0.1999	RandomForest + GBM + XGBoost
Stacking	4665.10	0.1999	KNN + DecisionTree + RandomForest + XGBoost
Stacking	4663.06	0.2000	KNN + RandomForest + XGBoost
Stacking	4667.92	0.2005	KNN + DecisionTree + SVR + RandomForest + XGBoost
Stacking	4660.99	0.2005	KNN + SVR + RandomForest + GBM + XGBoost
Stacking	4653.16	0.2005	KNN + RandomForest + GBM + XGBoost
Stacking	4673.64	0.2006	DecisionTree + SVR + RandomForest + XGBoost
Stacking	4667.95	0.2007	SVR + RandomForest + GBM + XGBoost
Stacking	4669.41	0.2007	KNN + DecisionTree + SVR + RandomForest + GBM + XGBoost
Averaging	4660.31	0.2031	RandomForest + GBM + XGBoost

Bu sonuçlar incelendiğinde görülecektir ki her 2 ensemble yönteminde de hata oranı düşürülmüştür. Özellikle Stacking yöntemiyle birinci model olan XGBoost ve DecisionTree, RandomForests algoritmaları birlikte kullanıldığı zaman problemi çözen en iyi sonuca ulaşılmıştır. 2 ensemble yöntemini karşılaştırılmasında ise Stacking methodunun(Arada tabloya alınmamış daha çok ensemble model var.) Averaging methodundan daha iyi sonuç verdiği görülmektedir.

6-Kaynakça

[1]**Atakan, A.** (2024). *Turkish House Rent Prediction Dataset* [Veri seti]. Kaggle. https://www.kaggle.com/datasets/atakanak/turkish-house-rent-prediction-dataset/data