Ankara'daki Ev Fiyatlarının Doğrusal ve Doğrusal Olamayan Makine Öğrenmesi Yöntemleri ile Tahmin Edilmesi

House Prices Prediction in Ankara with Linear and Non-linear Machine Learning Methods

Yazarlar: Sercan Yıldırım N21239069, Doç.Dr. Hacer Yalım Keleş

***Özetçe*—Bu çalışmada, bir emlak satış sitesinden web scraping yöntemi ile çekilmiş veriler kullanılarak bir analiz gerçekleştirilmiştir. Temin edilen veri seti, emlak satış bedellerini ve ilgili emlak ilanına ait detaylı bilgileri içermektedir. Toplam 154802 veriden sadece Ankara iline ait olan 11858 adedi analize dahil edilmiştir. Analizde makine öğrenim modellerinden Doğrusal Regresyon, Karar Ağaçları, Destek Vektör Regresyonu, En Yakın Komşu Algoritması, Ridge Regresyonu yöntemleri kullanılmıştır. Doğrusal ve Doğrusal olmayan tekniklerin birlikte kullanılmasına özen gösterilmiştir. Sonuç olarak test verileri üzerinde en başarılı sonucu En Yakın Komşu algoritmasının verdiği anlaşılmıştır.**

Anahtar Kelimeler — Makine Öğrenmesi, Doğrusal Regresyon, Karar Agaçları, SVR, Destek Vektör Regeresyonu, KNN, En Yakın Komşu Algoritması, Piton

***Abstract*— In this study, an analysis was performed using data taken from a real estate sales site by web scraping method. The provided data set includes the real estate sales prices and detailed information about the relevant real estate advertisement Jul. Out of a total of 154802 data, only 11858 belonging to the province of Ankara were included in the analysis. Linear Regression, Decision Trees, Support Vector Regression, Nearest Neighbor Algorithm, Ridge Regression methods from machine learning models were used in the analysis. Care has been taken to use linear and nonlinear techniques together. As a result, it has been understood that the Nearest Neighbor algorithm gives the most successful result on the test data.**

Keywords — Machine Learning, Linear Regression, Decision Tree Regression, Support Vector Regression(SVR), KNeighbors Regression, Ridge(L2) Regression, Python

I. GİRİŞ

Makine Öğrenmesi metotları ile Ankara ilinde yer alan 11858 emlak çeşitli özellikleri ile analiz edilmiştir. Veri seçimi için benzer çalışmalar literatürde taranmış ve github üzerinden veri elde edilmiştir. Makine Öğrenmesi için modeller hazırlanmadan önce keşifsel veri analizi yapılarak veri anlamlandırılmaya çalışılmıştır. Gereksiz satır ve sütunlar kaldırılmıştır. Veri temizleme ve düzenleme işlemleri gerçekleştirilmiştir. 20 sütunun hepsinin girdi olarak kabul edilmesi modeli gereksiz yoracağı değerlendirildiği için girdi sayısı düşürülmüştür. Sonuçlar kıyaslanarak, gerçeğe en yakın tahmini veren yöntemin kullanılması ileriki çalışmalar için önerilmiştir.

II. VERİ DÜZENLEME

Github üzerinden veri temin edildikten sonra, Python kodlama dilinin Pandas kütüphanesi kullanılarak veriler Colab ortamına alınmıştır.[1] Genel bilgilerine göz atılılarak, uç değerlerin tespiti gerçekleştirilmeye çalışılmıştır. Veri sayısının tüm Türkiye'deki emlakları içermesi sebebiyle çok fazla olduğu değerlendirilmiş, örneklem olarak Ankara ilindeki emlak sayısının yeterli olacağına karar verilmiştir. Verinin içerdiği sütun başlıkları ve içerikleri tablo.I 'de verildiği şekildedir.

TABLO I : EMLAK VERİ SETİ SÜTUNLARI

İndeks	Sütun Adı	Dolu Hücre Sayısı
0	Fiyat	154802
1	Adres	154802
2	Oda Sayisi	154802
3	Bulundugu Kat	154802
4	Isitma Tipi	154802
5	Krediye Uygunluk	154802
6	Yapi Durumu	154802
7	Tapu Durumu	154802
8	Esya Durumu	154802
9	Site Içerisinde	154802

İndeks	Sütun Adı	Dolu Hücre Sayısı
10	Türü	154802
11	Tipi	154802
12	Brüt Metrekare	154802
13	Binanin Yasi	154802
14	Binanin Kat Sayisi	154802
15	Kullanim Durumu	154802
16	Yatirima Uygunluk	154802
17	Banyo Sayisi	154802
18	Balkon Sayisi	154802
19	WC Sayisi	154802

Veriye ilk bakıldığında eksik bilginin yer almadığı dikkat çekmektedir. Nicel ve nominal tipteki veri değerlerinin karışık olarak verildiği fark edilmiştir.

İlk olarak verilerin Ankara ili özelinde süzülebilmesi için "Adres" sütunundaki metin, tire ile ayrılarak il – ilçe – mahalle isimlerinde yeni üç adet sütun oluşturulmuştur. Süzme işlemi sonrası daha rahat çalışılabilmesi ve orijinal veriye geri döneme ihtiyacı oluşması ihtimali gözetilerek Ankara iline ait veriler "ankara_df" adında yeni bir dataframe(df) olarak kaydedilmiştir. Veri tipleri ile python'ın otomatik belirlediği veri tipleri arasında uyuşmazlıklar tespit edilmiştir. Örneğin nicel olarak değerlendirilmesi gereken fiyat bilgisini sonundaki "TL" ibaresinden ötürü fiyat sütununun object veri tipinde gösterildiği fark edilmiştir. Veri tiplerinin doğru şekilde tanıtılması için aşağıda maddeler halinde yazılan düzeltme-silme-çıkarma işlemleri icra edilmiştir.

- "Fiyat" sütunundan "TL" ve boşlukların silinmesi

Modeldeki nicel verilerin bilgisayar tarafından integer olarak algılanabilmesi için içerisinde yer alan text ifadelerin kaldırılma işlemleri pandas'ın replace fonksiyonu yardımıyla gerçekleştirilmiştir.

- "Brüt Metrekare" sütununda "m2" biriminin kaldırılması

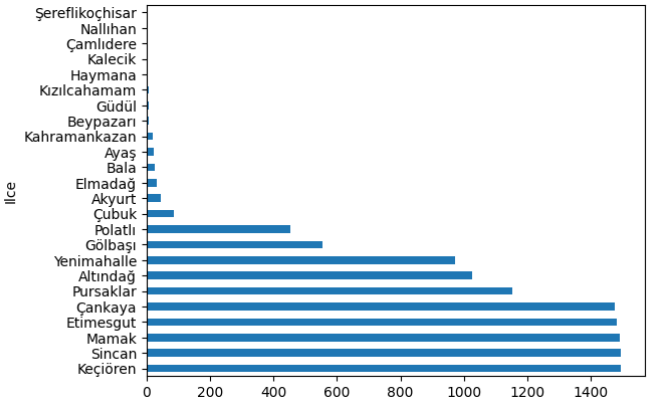
Fiyat sütununda olduğu gibi integer olarak modele tanıtılmak istenen metrekare alanındaki text ifadeler kaldırılmıştır.

- "Oda Sayısı" sütunundaki değerlerden 1+1, 2+1, 3+1 ve 4+1 dışındakilerin "diğer" olarak değiştirilmesi

Analizdeki biricik değer sayılarının analiz süresini uzatması sebebiyle başlıca değişkenler tutulmuş, diğerlerine genelleştirilen değerler verilmiştir.

- "İlçe" sütununda 100'den az olan ilçelerin "diğer" olarak değiştirilmesi

Ortalamayı değiştirecek uç değerlerin kaldırılması için bir histogram oluşturulmuş ve bu şekil I'de gösterilen histogramın üst kısmında kalan değerler analizde dahil edilmemiştir.



ŞEKİL I İLÇELERE GÖRE EMLAK SAYISI

- "Bulunduğu Kat" sütununda 4.Kat, 3.Kat, 2.Kat, 1.Kat, Yüksek Giriş dışındaki değerlerin "diğer katlar" olarak değiştirilmesi

- "Bina Yaşı" sütununda aralık olarak belirtilen değerlerin ortalamalarının atanması

Yapılan düzeltme işlemlerinden sonra, her bir sütun için biricik(unique) değerlere bakılmıştır. Biricik değer olarak tek değişken içeren, analiz için yeterli girdisi olmayan sütunlar analize dahil edilmemesi için atılmıştır. Tablo II'de sütun azaltma ve düzeltme işlemleri sonrası analizde kullanılacak sütun isimleri ve veri tipleri paylaşılmıştır.

TABLO II : ANALİZE DAHİL EDİLEN SÜTUN İSİMLERİ

İndeks	Sütun Adı	Veri Tipi
0	Fiyat	Nicel
1	Oda Sayisi	Nominal
2	Bulundugu Kat	Nominal
3	Isitma Tipi	Nominal
4	Tipi	Nominal
5	Brüt Metrekare	Nicel
6	Binanin Yasi	Nicel
7	Binanin Kat Sayisi	Nominal
8	Banyo Sayisi	Nicel

III. KULLANILAN YÖNTEMLER

Çalışmaya başlanmadan önce bu tahmin problemi için kullanılabilecek makine öğrenmesi yöntemleri hakkında bir literatür taraması gerçekleştirilmiştir. Yapılan araştırmalar neticesi; Doğrusal Regresyon, Karar Ağaçları Regresyonu(KAR), Destek Vektör Regresyonu, En Yakın Komşu Algoritması, Ridge Regresyonunun kullanılması kararlaştırılmıştır.

A. *Doğrusal Regresyon(Lineer Regression)*

Doğrusal Regresyon, makine öğrenmesinin denetimli öğrenme modellerinde, bağımsız değişkenler ile bağlı değişken arasındaki en uyumlu doğruyu çizen bir algoritmadır. Bir veya daha fazla girdi ile çıktı arasındaki istatistiksel ilişkiyi tanımlamada yardımcı olur. Girdi sayısına göre tekli veya çoklu olarak adlandırılmaktadır. Modelin amacı gerçek çıktı ile tahmin arasındaki farkı minimum etmektir. Bu hata minimizasyonu işleminde gradient desent yöntemi kullanılmaktadır.

B. *Karar Ağaçları Regresyonu(Decision Tree Regression)*

KAR yöntemi kısaca, bağımsız değişkenleri bilgi kazançlarına göre aralıklara bölmektedir.[2] Tahminleme kısmında ilgili aralığa denk gelen bir değer sorulduğunda, o aralığın ortalamasını döndürmektedir. Diğer regresyon modellerinden farklı olarak kesikli bir yapıya sahiptir. İlgili aralığa düşen tüm değerler için aynı ortalama değerini verecektir.

C. *Destek Vektör Regresyonu (Support Vector Regression)*

Destek Vektör Makineleri öncelikle sınıflandırma problemleri için kullanılmıştır. Sonrasında regresyon modellerinde de bu yöntemden faydalanılmıştır. Makine öğreniminin denetimli öğrenme modelleri başlığı altında değerlendirilir.[3] Amaç çizilecek doğrunun maksimum sayıda noktayı içermesini sağlamaktır. Bu noktalar destek noktası şeklinde isimlendirilmektedir. SVR modeli uygulanırken Raidal Basis Function(RBF) metodu ile birlikte kullanılırsa, doğrusal olmayan problemlerin çözümünde de kullanılabilmektedir. [4]

D. *En Yakın Komşu Regresyonu(KNeighbors Regression)*

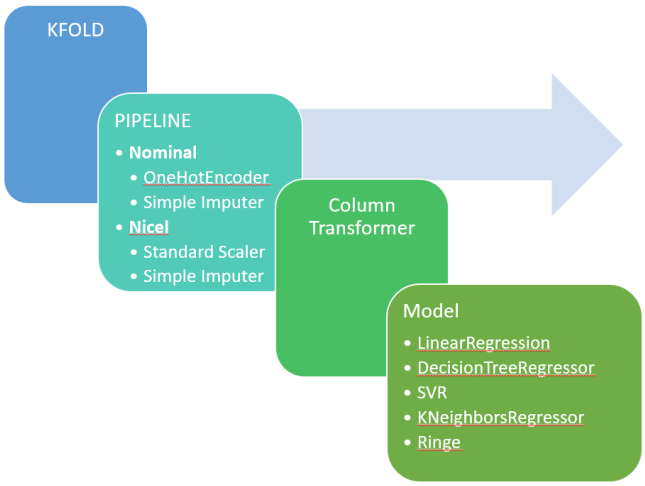
KNN temel olarak problemimizdeki noktaların k adet kümeye ayrılırken küme içindeki noktalar arasındaki mesafeyi minimum, kümeler arasındaki mesafeyide maksimum yapmayı amaçlamaktadır. Regresyonda uygulanmasında ise tahmin edilecek noktanın en yakınındaki küme belirlenir ve küme elemanlarının ortalaması tahmin olarak belirlenmektedir.

E. *Ridge Regresyon*

Ridge regresyon temelde en küçük kareler yöntemine dayanan bir yöntemdir.[5] Genellikle aşırı derecede uyum(overfitting) probleminin olduğu durumlarda kullanılmaktadır. Çoklu bağımsız değişkenlerin olduğu problemlerin regresyon denklemlerindeki katsayıları küçültmek için bir cezalandırma terimi kullanır. Bu ceza terimi katsayıların değerini sıfıra yaklaştırarak modeli daha basit bir hale getirmeyi sağlar. Böylece modelin genelleme yeteneği artırılmış olmaktadır. Bu ceza terimi lambda olarak isimlendirir ve bir hiperparametre ile kontrol edilir.

IV. MAKİNE ÖĞRENMİM MODELLERİNİN KURULMASI

Yapılan veri düzenleme ve formatlama işlemlerinden sonra makine öğrenmesinde sıklıkla kullanılan sütun doğrulaması, verilerin normalize edilmesi, modellerin koşulması işlemleri gerçekleştirilmiştir.



ŞEKİL II ANALİZ İŞ AKIŞI

A. *Sütun Doğrulama*

Analiz edilecek verilerin %80 train, %20 test olarak ikiye ayrılması sırasında belirli alanlarda yatkınlık olması ve rassallıktan uzaklaşılması gibi riskler bulunmaktadır. Bu istenmeyen yönelimlerin önüne geçilebilmesi için KFold yöntemi ile veriler parçalara ayrılıp, harmanlanır. Bu yöntemde kullanıcının belirlediği sayı kadar veri parçalanır ve her bir parça tek tek test datasıymış gibi analiz koşulur. Her bir sonuç en sonunda toplanır ve parça sayısın bölünerek ortalaması alınır. Böylelikle parçalarda var olması istenmeyen yönelimlerin önüne geçilmesi sağlanmaktadır.

B. *Pipeline Oluşturma*

Makine öğrenmesindeki bir diğer husus ise verilerin dinamik olarak sürekli değişiyor, güncellerinin de sete dahil edilebiliyor olmasıdır.[6] Bu durumu yönetebilmek, verilere standart olarak uygulanması istenen yöntemleri bir kurala oturtabilmek amacıyla pipeline'lar kullanılmaktadır. Bu çalışmada yine verilerin olası boş değerlerini doldurmak için "simpleimputer", nominal değerlere bilgisayarın anlayabileceği biricik değerler atayıp sonuçlara dahil etmesi için "onehotencoder" ve nicel değerlerin 1 ile -1 arasında değerler ile scale edilebilmesi için "StandardScaler" yöntemleri pipeline'lar içerisine eklenmiştir. Her bir veri tipine göre uygulanması istenen süreçler farklı olduğu için 2 tip(nicel, nominal) pipeline'lar oluşturulmuştur. Pipeline'lar mühendislerin işlerini kolaylaştıran yalın bir kod yazımına imkan sağlayan kütüphanelerdir.

C. *Sütun Dönüştürme*

Pipeline'lar aracılığı ile tanımlanmış olan akış kurallarının, bizim verimizdeki hangi sütunlara uygulanması gerektiğini tanımladığımız yer sütun dönüştürme kısmıdır. Bu kısımda 2 farklı tip pipeline, hangi sütunlar için geçerli ise onlar arasında eşleştirme yapılmaktadır.

D. *Modelleri Belirleme*

Modelleri içeren kütüphaneler yüklendikten ve veri hazırlıkları tamamlandıktan sonra çalışılacak makine öğrenme yöntemleri bir liste olarak kod bloğumuza dahil edilmiştir. Bu listedeki her eleman tekrarlayan(recursive) bir döngü ile çağrılarak sonuçlar elde edilmiştir. Sırasıyla modellerin gerçek ve tahmin değerleri arasındaki uyum yüzdelik olarak tablo.III'teki gibi hesaplanmıştır.

V. SONUÇ

Tablo III'te görüldüğü üzere gerçek verilere en yakın tahmin değerlerini en yakın komşular metodu (KNN) vermiştir. Bu uyumun daha da iyileştirilebilmesi için boosting algoritmalarının uygulanması değerlendirilebilir.

TABLO III : YÖNTEM UYUMLULUK YÜZDELERİ

Yöntem	**Tahmin ile Test Verisinin Uyum Yüzdesi**
LinearRegression()	-373%
DecisionTreeRegressor()	26%
SVR()	-5%
KNeighborsRegressor()	59%
Ridge()	56%

VI. KAYNAKÇA

[1] B. Köseoğlu, «Github,» 15 05 2021. [Çevrimiçi]. Available: https://github.com/busekoseoglu/WebScraping_HousePricePrediction/blob/main/Datasets/data_analyzed.csv.

[2] B. Köseoğku, «Medium,» 26 Temmuz 2020. [Çevrimiçi]. Available: https://buse-koseoglu13.medium.com/ridge-lasso-ve-elastic-net-b6089bf2f09. [Erişildi: 27 Mayıs 2024].

[3] o. Lahmiri, S. Bekiros ve C. Avdoulas, «A comparative assessment of machine learning methods for predicting housing prices using Bayesian optimization,» Decision Analytics Journal, cilt 6, p. 100166, 2023.

[4] A. K. S. M. R. Gupta, «Forecasting the US real house price index: Structural and non-structural models with and without fundamentals,» Econ.Model., p. 2013–2021, 2021.

[5] A. Özçift, «Forward stage-wise ensemble regression algorithm to improve baseregressors prediction ability: An empirical study,» Expert Syst., p. 31, 2012.

[6] S. G. A. S. V. Vapnik, «Support vector machine for function approx-imation, regression estimation, and signal processing,» Adv. Neural Inf. Process.Syst., pp. 281-287, 1996.