BIL 470 Final Raporu

Berkay Yıldız  
Bilgisayar Mühendisliği Bölümü  
TOBB Ekonomi ve Teknoloji Üniversitesiberkay.yildiz@etu.edu.tr

*Özet*—Bu projede amacımız, Amerika Birleşik Devletleri'nde gayrimenkul piyasasını analiz etmek için geniş kapsamlı bir veri kümesini incelemektir. Veri seti, çeşitli özellikler ve parametrelerle ilgili bilgiler içermektedir. Bu verileri kullanarak, gayrimenkul piyasasının eğilimleri, fiyat değişiklikleri ve diğer önemli faktörler hakkında anlayış kazanmayı amaçlıyoruz.

Anahtar Kelimeler—makine öğrenmesi, gayrimenkul, Amerika Birleşik Devletler emlak, veri analizi, fiyat trendleri

# Giriş

Bu projede amacımız, Amerika Birleşik Devletleri gayrimenkul piyasasını derinlemesine incelemek ve anlamak için geniş bir veri kümesini kullanmaktır.

Veri seti, gayrimenkul özellikleri, fiyatları ve diğer ilgili bilgileri içeren kapsamlı bir veri kaynağı sunmaktadır. Bu verileri analiz ederek, Amerika Birleşik Devletleri'ndeki gayrimenkul piyasasının eğilimleri, bölgesel farklılıklar ve fiyat değişiklikleri gibi önemli konularda bilgi edinmeyi amaçlıyoruz. Gayrimenkul yatırımları, finansal kararlar ve piyasa tahminleri konularında önem taşımaktadır. Bu veri seti, yatırımcılar, emlak profesyonelleri ve araştırmacılar için faydalı bilgiler sunma potansiyeline sahiptir.

Projenin amacı, bu veri kümesini kullanarak değerli içgörüler elde etmektir. Veri kümesinin ayrıntılarına bu bağlantıdan erişebilirsiniz. Bu proje, gayrimenkul verilerinin analizi ile ilgilenen birçok kişi için faydalı olabilir. Veri setinin sağladığı bilgileri kullanarak gelecekteki gayrimenkul eğilimleri hakkında önemli bilgiler elde edebiliriz.

Günümüzde Türkiye Emlak piyasasındaki aşırı artış beni emlak piyasasındaki gidişat için meraklandırdı. Ev alırken nelere bakarak ev alınıyor, evin fiyatını belirleyen etmenler neler gibi sorulara cevaplar bulmak beni motive etti. Proje linki:

<https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset?datasetId=3202774&sortBy=voteCount>

Problemin çıktı sonucu evin fiyatıdır.

# Veri Seti, Veri Özellikleri, Veri Öznitelikleri

Veri kaynağı:

<https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset?datasetId=3202774&sortBy=voteCount>

## Veri Setindeki Öznitelikler ve Tanımları

* status (Durum): Bu öznitelik, mülkün satışa hazır olup olmadığını veya inşa edilmeye hazır olup olmadığını belirtir. Bu durum, piyasadaki mevcut mülklerin durumunu ve inşa projelerinin ilerlemesini yansıtabilir.
* bed (Yatak Sayısı): Bu öznitelik, mülkte bulunan yatak odası sayısını gösterir. Bu bilgi, bir mülkün boyutu ve kullanım alanını anlamamız için önemlidir.
* bath (Banyo Sayısı): Bu öznitelik, mülkte bulunan banyo sayısını ifade eder. Bu özellik, bir mülkün rahatlığını ve işlevselliğini etkileyebilir.
* acre\_lot (Arazi Alanı): Bu öznitelik, mülkün arazi veya arsa alanını ifade eder. Büyük bir arazi alanı genellikle daha fazla özgürlük ve genişleme olanağı sunar.
* city (Şehir): Bu öznitelik, mülkün bulunduğu şehir adını gösterir. Şehir, mülkün konumu ve piyasa değeri üzerinde büyük bir etkiye sahip olabilir.
* state (Eyalet): Bu öznitelik, mülkün bulunduğu eyalet adını belirtir. Eyalet bazen vergi oranları ve diğer yasal düzenlemeler nedeniyle mülk değerini etkileyebilir.
* zip\_code (Posta Kodu): Bu öznitelik, mülkün bulunduğu posta kodunu gösterir. Posta kodları, coğrafi bölgeleri tanımlamak için kullanılabilir.
* house\_size (Ev Alanı): Bu öznitelik, mülkte bulunan ev alanını kare fit cinsinden ifade eder. Bu bilgi, bir mülkün büyüklüğünü ve yaşam alanını yansıtabilir.
* prev\_sold\_date (Önceki Satış Tarihi): Bu öznitelik, mülkün daha önce satıldığı tarihi belirtir. Bu bilgi, mülkün geçmiş performansını ve satış geçmişini anlamamıza yardımcı olabilir.
* price (Fiyat): Bu öznitelik, mülkün fiyatını ifade eder. Eğer mülk hali hazırda satılıyorsa mevcut liste fiyatını, yakın zamanda satıldıysa son satış fiyatını yansıtabilir.

# Onişleme Aşamaları

Kullandığımız veri setini makine öğrenmesi için direk bu şekilde kullanamayız. Çünkü veri setinde makine öğrenmesi için kullanılamayacak değerler var. Bunu makine öğrenmesi için uygun hale getirmeliyiz.

## Null Değerleri Doldurma

Bunun için 2 method kullandım.

#### Silme: Bazı özniteliklerde boş değerler verilerin yarısından fazlasını oluşturuyordu. Ayrıca dolu olan kısımlarda da anlamsız veriler bulunuyordu. Hem anlamsız veriler varken hem de boş değerlerin çok fazla olması sebebiyle özniteliği kullanmama kararı aldım.

#### Uygun Değer Yerleştirme: Bazı verilerimde ise null değerler için mod, medyan gibi matematik operatörleri kullandım. Bu yöntemlerden hangisini seçmem gerektiğini datanın dağılımına göre seçmeye çalıştım.

## Aykırı Değer Temizleme

Aykırı veriler (outliers), genellikle diğer verilere göre büyük ölçüde farklı değerlere sahip olan verilerdir. Bu aykırı değerler, istatistiksel analizleri veya makine öğrenimi modellerini olumsuz yönde etkileyebilir. Bu aykırı değeleri temizledim.

## Numerik Dataya Çevirme

Veri setindeki bazı özniteliklerin verileri numerik olmayan verilere sahipti. Bu şekli ile makine öğrenmesi için kullanılamayacak fakat uygun formata çevrilebilecek olan verileri 2 farklı yaklaşım ile numerik veriye çevirdim.

#### One-Hot Encoding: Bu yaklaşım kategorik değişkenlerin binary olarak temsili olarak kullanılmasıdır. Bunun için ilk başta verilerin bir numerik tamsayıya eşitlenmesi gerekir ve daha sonrasında binary vektör şeklinde gösterilir.

#### Label Encoding: Veriyi direk olarak sayısallaştırmada kullanılır. Her kategorik veri için farklı bir sayısal değer kullanılır.

## Ölçeklendirme

Bu aşamada, veriyi modele uygun hale getirme, ölçeklendirme ve normalize etme gibi işlemler gerçekleştirilir. Veri setindeki öznitelikler farklı aralıklarda olabilir. Bazı algoritmalar, öznitelikler arasındaki büyüklük farklarından etkilenebilir. Ayrıca bazı modeler için ölçeklendirilmiş veriler daha uygundur ve çok daha hızlı çalışır. Bu bağlamda, iki yaygın kullanılan yöntem "MinMax Scaler" ve "StandardScaler" metodlarını kullandım.

#### MinMax Scaler: Bu yöntem, her özniteliği belirli bir aralığa yeniden ölçeklendirir. Varsayılan olarak, bu aralık 0 ile 1 arasındadır. Her bir örnek için, minimum değer 0'a, maksimum değer ise 1'e denk getirilir. Bu yöntem, veriyi bir öznitelik aralığında sıkıştırırken, orijinal dağılımın genel yapısını korumasını sağlar.

#### StandardScaler: Bu yöntem ise, her bir özniteliği ortalama değerini çıkararak ve standart sapmasına bölererek standartlaştırır. Böylece, her öznitelik ortalaması sıfır ve standart sapması bir olacak şekilde dönüştürülür. Bu sayede veri, yaklaşık olarak normal dağılıma benzer bir dağılım alır.

# Kullanılan Modeller

Makine öğrenmesi için birçok sayıda model mevcut. Bunlardan bazıları bu proje için kullanmaya uygun gördüm. Bu modelleri ve neden kullandığımı madde madde açıklayacağım. Ayrıca test\_size değerini 0.1 olarak belirledim.

## Linear Regression

Bu regresyon yöntemi, veri setindeki özniteliklerin bağımlı değişkenle olan ilişkisini analiz etmek için kullanılır. Öznitelikler ve bağımlı değişken arasındaki doğrusal ilişkiyi yakalamaya çalışır. Bu sayede, örneğin öznitelik değerleri arttıkça bağımlı değişkenin nasıl değişeceğini anlamamıza yardımcı olur. Özellikle tek bir bağımlı değişkeni tahminlemek veya ilişkileri anlamak için tercih edilir.

## Random Forest Regression

Veri setini birden fazla karar ağacı kullanarak analiz eden bir regresyon algoritmasıdır. Bu ağaçlar, rastgele öznitelikler ve rastgele veri altkümeleri kullanılarak oluşturulur. Her ağaç bağımsız olarak sonuç tahminleri yapar, ardından bu tahminler birleştirilerek daha doğru bir sonuç elde edilir. Bu model, karmaşık ilişkileri yakalamak ve aşırı öğrenmeyi azaltmak için kullanılır. Karar ağacını kullandığım için bu modeli de çalıştırmak ve sonuçlarını görmek istedim.

## Decision Tree Regression

Karar Ağacı Regresyonu, veri setini ağaç yapıları kullanarak analiz eden bir regresyon yaklaşımıdır. Veriyi özniteliklerin değerlerine göre böler ve her bir ağaç sonuç tahmininde bulunur. Bu yöntem, basit ve anlaşılır sonuçlar elde etmek istendiğinde kullanılır. Ancak tek bir ağaç kullanıldığında aşırı öğrenme riski taşıyabilir. Karar ağaçlarını sevdiğim için bu modeli kullanmayı tercih ettim.

## K-Nearest Neighbor(KNN)

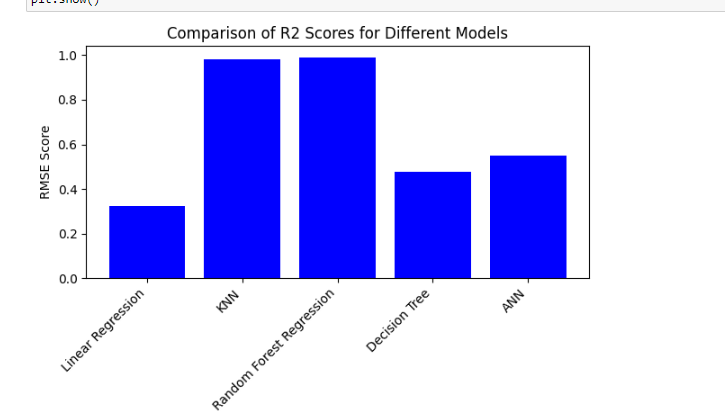
KNN, hem sınıflandırma hem de regresyon problemlerinde kullanılan bir yöntemdir. Bir veri noktasının sonucunu tahminlemek için en yakınındaki K veri noktasına bakar. K'ya bağlı olarak, bu yakın komşuların sonuçlarını birleştirerek tahminde bulunur. Bu yöntem, verinin örüntülerini ve benzerliklerini temel alır. Genellikle basit modelleri değerlendirmek ve hızlı sonuçlar elde etmek için kullanılır. Diğer modellere karşı, etiketsiz çalışan bu modelin durumunu incelemek istediğim için bu modeli tercih ettim.

## Artificial Neural Network (ANN)

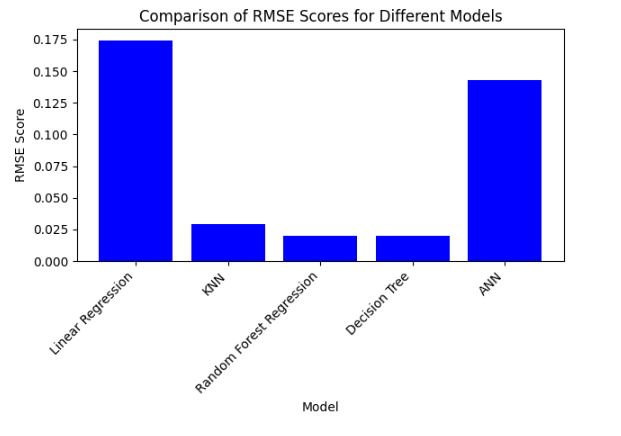
Yapay Sinir Ağları (YSA), karmaşık yapılardaki ilişkileri anlamak için kullanılan bir derin öğrenme modelidir. Birbirine bağlı katmanlar ve nöronlardan oluşur. Giriş katmanı, gizli katmanlar ve çıkış katmanından oluşur. İleri ve geri yayılım gibi algoritmalar kullanarak ağı eğitir. YSA, büyük ve karmaşık verileri analiz etmek için kullanılır ve öğrenilen örüntüleri yakalayabilir.

# Test Sonuçları

Modellerin Eğitimi sonucunda çıkan test sonuçları büyük farklılık göstermemektedir. Açıklanabilirlik anlamında büyük farklar bulunmaktadır. Bu da isteğe göre değişebilir. Çıkan Rmse ve R2 tablolarının karşılaştırmalarını aşağıdaki görsellerde inceleyebilirsiniz.



Görüntü-1 R2Scores



Görüntü-2 RMSE Scores

# Sonuçlar

Bu çalışmada evin sahip olduğum bilgilerinden fiyatlarını tahminlemeye çalıştım. Tahminlemek için kullandığım modelleri araştırmalarım sonucundaki ilgimi çekme düzeyi ve problemimle uyum sağlayacak şekilde seçtim.

Makine öğrenmesi modellerimi eğitmek için öncelikle veri setimi uygun hale getirmeliydim. Feature selection ve Feature Extraction yöntemlerini kullandım. Daha sonra veri setimdeki null valueları tespit edip bunları uygun yöntemler ile doldurdum. Mod, medyan bu yöntemler arasındadır.

Outlierları çıkarttığımda daha iyi sonuçlar elde ettim ve önemini anlamış oldum.

Makine öğrenmesinin öğrenmeye uygun olabilmesi için veri setindeki özniteliklerin datatype’ı numerik olmalıdır. Benim veri setimde numerik harici verilerde olduğundan bunları uygun yöntemlerle numerik şekle getirdim. Label Encoding ve One Hot encoding yöntemlerini kullandım.

Yapay öğrenme için 4 tane makine öğrenmesi bir tane de derin öğrenme kullandım. Yapay öğrenme metodlarını çalıştırıp r2\_score ‘lara göre birbirleriyle kıyasladım.

Tüm bunları yaparken teorikte öğrenmiş olduğum bilgileri Pratik olarak da görmüş oldum. Derin öğrenme modelinin en iyi sonucu vermemesi beni şaşırttı. Bazı yaptığım hataları hızlıca farkedemem sonucu zaman kaybettim. Gelecek çalışmalarımda bu çalışmada yaparak öğrendiğim yanlışları tekrar yapmamak benim için zaman kazandırıcı olacaktır. Ayrıca bu gibi çalışmaların gerçek hayatta ne kadar uygulanabilir olduğumu tekrar farketmem sonucu bu alana ilgim olduğunu farketmemde yardımcı oldu.

Bu modeli geliştirmek için daha fazla veriye sahip olmak derin öğrenmeye de yardımcı olacağı gibi diğer modellerin gelişimine de katkı sağlar. Modellerin az kullandığı öznitelikler yerine daha uygun veriler elde edilip bu öznitelikler ile değiştirilmesi modelin geliştilmesi için iyi olabiliceğini düşünüyorum.

##### References

1. <https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d>
2. <https://u-next.com/blogs/machine-learning/popular-regression-algorithms-ml/>
3. <https://www.ibm.com/topics/linear-regression#:~:text=Linear%20regression%20analysis%20is%20used,is%20called%20the%20independent%20variable>
4. <https://towardsdatascience.com/random-forest-regression-5f605132d19d>
5. <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/>
6. <https://www.geeksforgeeks.org/k-nearest-neighbours/>
7. <https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/>