MAKİNE ÖĞRENMESİ REGRESYONLARI İLE EV FİYAT TAHMİNİ

2020
BİLGİSAYAR MÜHENDİSLİĞİ
BİTİRME PROJESİ TEZİ

Mustafa SARITEMUR

MAKİNE ÖĞRENMESİ REGRESYONLARI İLE EV FİYAT TAHMİNİ

Mustafa SARITEMUR

Karabük Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümünde
Bitirme Projesi Tezi
Olarak Hazırlanmıştır.

KARABÜK
Haziran 2020

Mustafa SARITEMUR tarafından hazırlanan "MAKİNE ÖĞRENMESİ REGRESYONLARI İLE EV FİYAT TAHMİNİ" başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Oğuz FINDIK .....

Bitirme Projesi Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

...../……/2020

Bilgisayar Mühendisliği bölümü, bu tez ile, Bitirme Projesi Tezini onamıştır

Dr. Öğr. Üyesi Hakan KUTUCU .....

Bölüm Başkanı

"Bu projedeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Mustafa SARITEMUR

ÖZET

Bitime Projesi Tezi

MAKİNE ÖĞRENMESİ REGRESYONLARI
İLE
EV FİYAT TAHMİNİ

Mustafa SARITEMUR

Karabük Üniversitesi
Bilgisayar Mühendisliği
Bilgisayar Mühendisliği Bölümü

Tez Danışmanı:
Doç. Dr. Oğuz FINDIK
Haziran 2020, 41 sayfa

İnsanlar yaşamlarını sürdürmek için birtakım ihtiyaçları gidermek zorundadır. Bu ihtiyaçlar arasındaki en önemli ihtiyaçlardan biri de barınma ihtiyacıdır.[1] Bu ihtiyacımızı gidermek adına birçok ev gezer, kendimiz için en uygun olanı ararız. Eğer bu iş üzerinde uzman değilseniz, belirlediğimiz niteliklerle bulduğumuz evin fiyatını bilemeyebiliriz. Geliştirdiğimiz bu uygulama sayesinde ev sahibi olmak isteyen kişiler, bazı bilgileri girerek sahip olmak istedikleri evin fiyatı hakkında bilgi sahibi olabilecek.

Anahtar Sözcükler : Fiyat tahmini, makine öğrenmesi, regresyon teknikleri.

ABSTRACT

Senior Projcet Thesis

HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES

Mustafa SARITEMUR

Karabük University
Faculty of Engineering
Department of Computer Engineering

Project Supervisor:
Assoc. Prof. Dr. Oğuz FINDIK
June 2020, 41 pages

People have to fulfill a number of needs in order to maintain their lives. One of the most important of these needs is the need for shelter [1]. In order to meet this need, we visit many houses and look for most suitable for ourselves. If you are not expert on this subject, you may not know the price of the house you find in the features you set.

With this application we have developed, people who want to own a house can have some information about the price of the house they want to have by entering some information.

Key Words : Sales price, machine learning, regression techniques.

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde, oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. Oğuz FINDIK'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

KABUL ii
ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ŞEKİLLER DİZİNİ ix

BÖLÜM 1 111
GİRİŞ 111
1.1. ÇALIŞMANIN AMACI 111
1.2. TEZİN ORGANİZASYONU 12

BÖLÜM 2 13
MATERYAL VE YÖNTEM 13
2.1. REGRESYON 15
2.1.1. LINEAR REGRESSION 16
2.1.2. GRADIENT BOOSTING REGRESSION 17
2.1.3. DECISION TREE REGRESSION 18
2.1.4. SUPPORT VECTOR MACHINE REGRESSION 20
2.1.5. RANDOM FOREST REGRESSION 21
2.1.6. LightGBM REGRESSION 22
2.1.6.1. PARAMETRE OPTİMİZASYONU 22

BÖLÜM 3 24
KULLANILACAK VERİLERİN İNCELENMESİ 24
3.1. KORELASYON MATRİSİ 25
3.2. EKSİK VERİ 28

Sayfa

BÖLÜM 4 30
MAKİNE ÖĞRENMESİ MODELLEMELERİ 30
4.1. VERİ SETİ PARÇALAMA VE NORMALİZASYON 30
4.2. REGRESYON DEĞERLENDİRME 31
4.3. LINEAR REGRESSİON 32
4.4. GRADIENT BOOSTING REGRESSION 33
4.5. DECISION TREE REGRESSION 34
4.6. SUPPORT VECTOR MACHINE REGRESSION 35
4.7. RANDOM FOREST REGRESSION 36
4.8. LightGBM REGRESSION 37
4.9. MODEL KARŞILAŞTIRMA 38
4.10. REGRESYON DEĞERLENDİRME 31
4.2. REGRESYON DEĞERLENDİRME 31
4.2. REGRESYON DEĞERLENDİRME 31

BÖLÜM 5 39
SONUÇ VE DEĞERLENDİRME 39
KAYNAKLAR 40
ÖZGEÇMİŞ 41

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Linear Regression. 15
Şekil 2.2. Gradient Boosting Regression. 16
Şekil 2.3. Decision Tree Regression. 119
Şekil 2.4. SVM. ..... 1**Error! Bookmark not defined.**
Şekil 2.5. SVM-Non-Linear, SVM-Linear.. 19
Şekil 2.6. RFR. 20
Şekil 2.7. LightGBM - Büyüme. 22
Şekil 3.1. Korelasyon Matrisi. 24
Şekil 3.2. Parametre Değerlendirme. 25
Şekil 3.3. Parametre Değerlendirme. 25
Şekil 3.4. Parametre Değerlendirme. 25
Şekil 3.5. Parametre Değerlendirme 26
Şekil 3.6. Parametre Değerlendirme. 26
Şekil 3.7. Eksik Veri 27
Şekil 3.8. Korelasyon Matrisi-2. 28
Şekil 4.1. Veri Seti Parçalama 29
Şekil 4.2. Normalizasyon. 29
Şekil 4.3. Veri Seti Normalizasyon. 30
Şekil 4.4. Linear Regresyon. 31
Şekil 4.5. Linear Regression Karşılaştırma. 31
Şekil 4.6. GBR. 32
Şekil 4.7. GBR Karşılaitırma. 32
Şekil 4.8. DTR. 33
Şekil 4.9. DTR karşılaştırma. 33
Şekil 4.10. SVM. 34
Şekil 4.11. SVM Kararlaştırma. 34
Şekil 4.12. RFR. 35
Şekil 4.13. RFR Karşılaştırma. 35

Şekil 4.14. LGBM. 36
Şekil 4.15. LGBM Kararlaştırma. 36
Şekil 4.16. MSE Değer Karşılaştırması. 37
Şekil 4.17. Tüm Modellerin Tahmin 25 Adet Tahmin Değeri. 37

BÖLÜM 1

GİRİŞ

Ev sahibi olmak isteyen kişilerin yaşayabileceği sıkıntılardan biri alınacak evin fiyatının ne kadar olduğunu hakkında fikir yürütememek. Bu durum karşısında ev almak isteyen kişilerin almak istedikleri evin değerinden daha fazla ücrete satın alınmasına sebep olabilir. Böyle bir duruma düşmemek için konunun uzmanı olan bir emlakçı veya ev alım-satımı işleri ile uğraşan bir uzman çağırılarak uzman fikirleri alınabilir. Fakat bu durumda da fikir danışılan uzmana ödeme yapılması gerekmektedir. Fikir alma işlemi, alınmayı düşünülen her ev için yapılacağı düşünülürse gereksiz ödenmiş birçok ücret anlamına gelir.

1.1. ÇALIŞMANIN AMACI

Bu çalışmada halkın büyük bir çoğunluğunu ilgilendiren ev satın alınırken yaşanan finansal problemlerin önüne geçilmesi, değerinden yüksek fiyata ev alınmasının önüne geçilmesi için, fiyat tahminine yardımcı olacak şekilde makine öğrenmesi yöntemleri ile sınıflandırma çalışmaları yapmak amaçlanmıştır.

Proje sayesinde emlakçılara veya alanında uzman kişilere gereksiz ödenen ücretleri ortadan kaldırmak veya ev alacak kişilerin değerinden yüksek fiyata ev almak gibi problemleri ortadan kaldırmayı amaçlanmıştır.

Bu çalışmada makine öğrenmesi regresyon yöntemlerinden olan Linear Regression, Gredient Boosting Regression, Decision Tree Regression, Support Vector Regression, Random Forest Regression, LightGBM Regression kullanılacaktır. Çalışma kapsamında kaggle platformundan bulunan projenin sayfasındaki veriler kullanılacaktır. Çalışma kapsamında hem ev özellikleri ile beraber fiyatı verilmiş ev verileri hem de tahmin edilmesi beklenen sadece ev özellikleri bulunan veriler kullanılacak, verilerin bir kısmı eğitim bir kısmı da test verisi olarak ayrılacak ve her bir yöntemin tahmin etme performansı verilecektir. Böylece fiyat tahmini için

kullanılabilecek makine öğrenmesi yöntemlerinin tercihine yardımcı olmak istenmiştir.

1.2. TEZİN ORGANİZASYONU

Makine öğrenmesi yöntemleri ile ev fiyat tahmini için yazılan bu tez çalışması altı bölümden meydana gelmiştir.

Tez çalışmasının birinci bölümünde ev satın alırken yaşanabilecek durumlar hakkında kısaca bilgi verilerek konuya giriş yapılmış, çalışmanın amacı açıklanarak bu alandaki diğer çalışmalardan özgün yanı vurgulanmıştır.

İkinci bölümde uygulamada kullanılan makine öğrenmesi yöntemlerinin teorik temelleri anlatılmıştır.

Üçüncü bölümde uygulamada kullanılacak verilerin incelenmesi, görselleştirilmesi, eksik olan verilerin belirlenmesi ve eksik verilerin giderilmesi anlatılmıştır.

Dördüncü bölümde kullanılan makine öğrenmesi yöntemlerinin her birine ait modelleme ve sınıflandırma sonuçları verilmiştir.

Beşinci ve son bölümde, çalışmadan elde edilen genel sonuçlar gösterilmiş ve sonraki çalışmalar için öneriler verilmiştir.

BÖLÜM 2

MATERYAL VE YÖNTEM

Makine öğrenmesi esas olarak 1959 yılında bilgisayar biliminin yapay zekada sayısal öğrenme ve model tanıma çalışmalarından geliştirilmiş bir alt dalıdır. Makine öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir. Bu tür algoritmalar statik program talimatlarını harfiyen takip etmek yerine örnek girişlerden veri tabanlı tahminleri ve kararları gerçekleştirebilmek amacıyla bir model inşa ederek çalışırlar. Makine öğrenmesi algoritmaları yeni veriler gönderilirken, performansı iyileştirmek ve zamanla "zekâ" geliştirmek için operasyonları öğrenir ve optimize eder. [2]
Makine öğrenmesine ait başlıca kavramların listesi açıklamalarıyla birlikte aşağıdaki gibidir:

Denetimli Öğrenme: Denetimli öğrenmede, makine örnek olarak öğretilir. Operatör, makine öğrenme algoritmasını, istenen giriş ve çıkışları içeren bilinen bir veri kümesi ile sağlar ve algoritma, bu giriş ve çıkışlara nasıl ulaşacağını belirleyen bir yöntem bulmaktadır. Operatör, sorunun doğru cevaplarını bilmesine rağmen, algoritma verideki kalıpları tanımlar, gözlemleri öğrenir ve tahminlerde bulunur. Algoritma tahminlerde bulunur ve operatör tarafından düzeltilir. Bu işlem algoritma yüksek düzeyde bir doğruluk / performans elde edene kadar devam eder.[3]

Denetimli öğrenme başlığı altında: Sınıflandırma, Regresyon ve Tahmin.

- Sınıflandırma: Sınıflandırma görevinde, makine öğrenme programı gözlemlenen değerlerden bir sonuç çıkarmalı ve yeni gözlemlerin hangi kategoriye ait olduğunu bilmelidir.
- Regresyon görevlerinde i makine öğrenme programı değişkenler arasındaki ilişkileri tahmin etmeli ve anlamalıdır. Regresyon analizi, bir bağımlı değişkene ve bir dizi başka değişkene odaklanır.

- Tahmin: Tahmin, geçmiş ve şimdiki verilere dayanarak gelecekle ilgili tahminler yapma sürecidir ve genellikle eğitimleri analiz etmek için kullanılır.

Yarı Denetimli Öğrenme: Yarı denetimli makine öğrenmesi, denetimli öğrenmeye benzemektedir. Hem etiketli hem de etiketsiz verileri kullanmaktadır. Etiketli veriler esasen anlamlı esasen anlamlı etiketlere sahip bilgilerdir, böylece etiketlenmemiş veriler bu bilgiden yoksundur ve algoritma verileri anlayabilmektedir. Bu kombinasyonu kullanarak, makine öğrenme algoritmaları etiketlenmemiş verileri etiketlemeyi öğrenebilir.

Denetimsiz Makine Öğrenmesi: Makine öğrenmesi algoritması kalıpları tanımlamak için verileri inceler. Talimat verecek bir cevap veya insan operatörü yoktur. Bunun yerine, makine mevcut verileri analiz ederek korelasyonları ve ilişkileri belirlemektedir. Denetimsiz bir öğrenme sürecinde, büyük veri setlerini yorumlamak ve bu verileri buna göre ele almak için makine öğrenme algoritması bırakılmıştır. Algoritma, bu verilerin kümeler halinde gruplandırılması veya daha düzenli görünecek şekilde düzenlenmesi anlamına gelebilmektedir. Algoritma daha fazla veriyi değerlendirdikçe, bu verilerle ilgili kararlar verebilme becerisi giderek artmakta ve daha rafine olmaktadır.

Pekiştirmeye Dayalı Öğrenme: Pekiştirmeye dayalı öğrenme, sonuçlardan öğrenen ve gerçekleştirilecek eylemi kararlaştıran algoritmaları kullanır. Algoritma, her eylemden sonra seçeneği doğru mu, nötr mü yoksa yanlış mı olduğunu belirlemeye yardımcı olan geri bildirim alır. İnsan kılavuzluğu olmadan birçok küçük kararlar alması gereken otomatikleştirilmiş sistemler için kullanılabilecek iyi bir tekniktir.

Yoğun Öğrenme: Yoğun öğrenme yöntemi derin grafiklerde birçok doğrusal ve doğrusal olmayan dönüşümlerden ve çoklu işlem katmanlarından oluşturulmuş

verilerde, üst düzey soyutlamalar kullanılarak elde edilen model girişimlerine dayalı bir dizi algoritmalarla geliştirilmiş makine öğrenmesidir.

İnsanlar bir konu hakkında tahmin yürütebilmesi için o konu hakkında bilgi sahibi olmalıdır. Sahip olunan bilgi miktarı arttıkça yapılan tahminlerin sonuçları daha başarı olmaktadır. İnsanlarda ki bu tecrübe zamanla olmaktadır ve tecrübe edebileceği alanlar sınırlıdır. Makine öğrenmesi ile tecrübeye gerek kalmadan verilerden anlam çıkararak tahminlerde buluna bilinilir.

Aşağıda bu çalışmada kullanılacak makine öğrenmesi yöntemleri verilmiş, teorik temelleri açıklanmıştır.

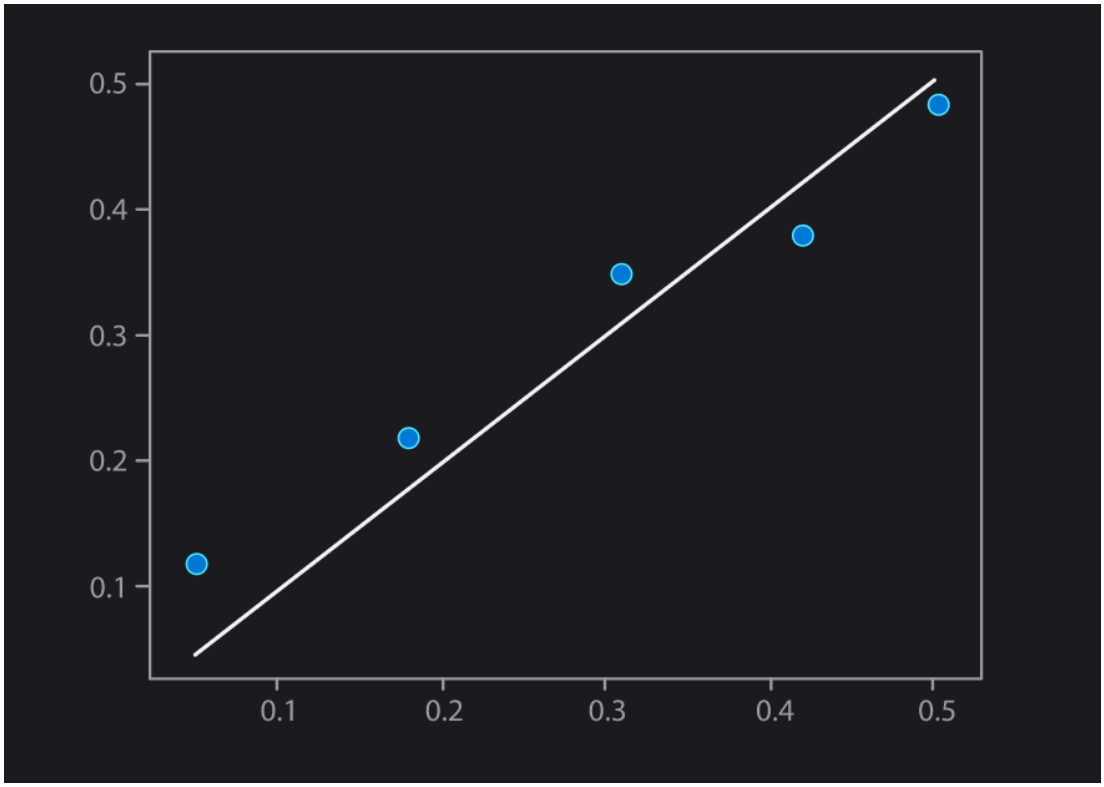
- Linear Regression
- Gredient Boosting Regression
- Decision Tree Regression
- Support Vector Regression
- Random Forest Regression
- LightGBM Regression

2.1. REGRESYON

Regresyon yaklaşımında, iki veya daha fazla sayıdaki değişken arasındaki ilişkiyi istatistiksel veya matematiksel olarak çözerek, verilecek yeni değerler için karşılık gelecek çıktı değerlerini hesaplamak amaçlanmaktadır.

2.1.1 LINEAR REGRESSION

Doğrusal regresyon, bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi açıklayan doğrusal bir yaklaşımdır. Tahmin için kullanılan en temel algoritma türüdür [4]. Basit doğrusal regresyon, veriler arasındaki ilişkiyi özetleyen istatistik bir yöntemdir. X-ekseninde gösterilen birinci değişken tahmin edici, bağımsız değişkendir. Y-ekseninde gösterilen ikinci değişken (tahmin edilen çıktı) ise bağımlı değişkendir. Basit doğrusal regresyon ile bulunan bu ilişki, istatistik bir ilişkidir.



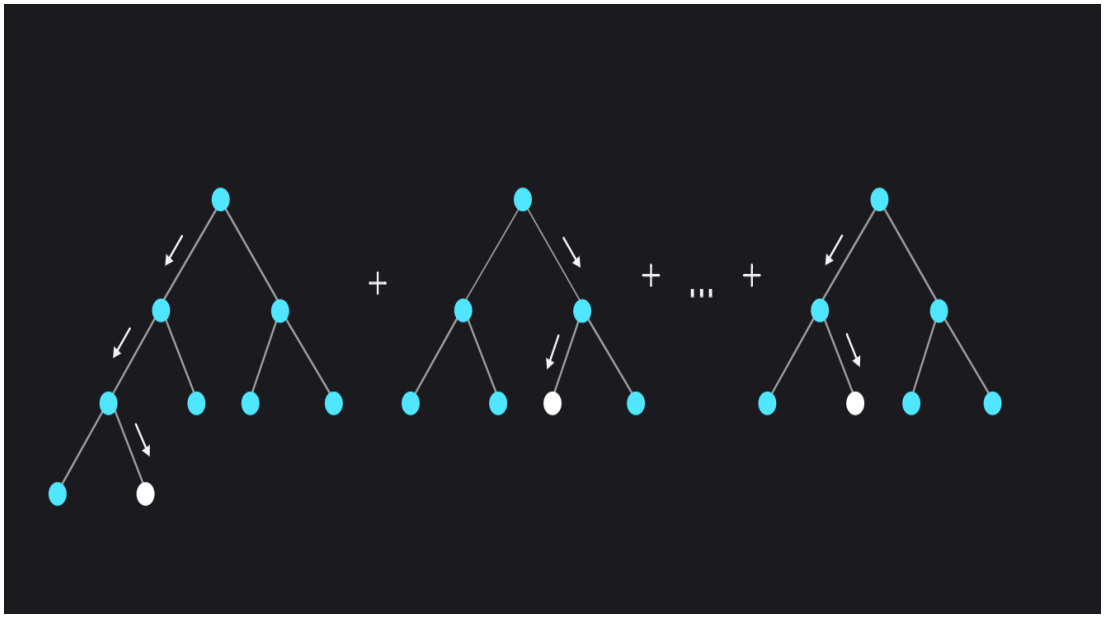
Şekil 2.1. Linear Regression

2.1.2 GRADIENT BOOSTING REGRESSION

Gradyan artırma regresyonu, regresyon ve sınıflandırma problemleri için bir makine öğrenmesi tekniğidir. Bu, zayıf tahmin modellerinin bir araya gelmesiyle tipik olarak karar ağaçlarının oluşturduğu bir model oluşturur. Denetlenen herhangi bir öğrenme algoritmasının amacı, bir kayıp fonksiyonu tanımlamak ve en aza indirmektir.

Gradyan artırma algoritmasının ardındaki sezgi, artıklardaki örüntüleri tekrar tekrar kullanmak ve zayıf tahminlerle bir modeli güçlendirmek ve daha iyi hale getirmektir.

Gradyan artırma algoritmaları, modelin genel performansını geliştiren birleştirme işlemi ile zayıf tahmin modellerini (genellikle karar ağaçlarını) bir araya getiren bir tahmin modeli oluşturur.



Şekil 2.2. Gradient Boosting Regression.

2.1.3 DECISION TREE REGRESSION

Sınıflama modelleri içerisinde yer alan karar ağaçları yöntemleri tahmin edici ve tanımlayıcı özelliklere sahiptir. Karar ağaçları, yorumlanmalarının kolay olması, veri tabanı sistemleri ile kolayca entegre edilebilmeleri, güvenilirliklerinin daha iyi olması nedenlerinden dolayı sınıflama modelleri içerisinde en yaygın kullanıma sahip olan yöntemlerden birisidir. Ayrıca yapılandırması ve anlaşılması daha kolay bir yöntem olması, model şeffaflığını sağlaması ve görsel bir sunuma sahip olması da yaygın kullanımına sebep olarak gösterilebilir.

Karar ağacı karar düğümleri, dallar ve yapraklardan oluşur. Karar düğümleri de üçe ayrılmaktadır.

Kök düğüm: Kendisinden önce bir dal olmayan ve kendisinden bir veya daha fazla dal çıkabilen düğümdür. Kök düğüm sınıflandırmanın hangi değişkene göre yapıldığını gösterir. Kök düğüm bağımlı değişkeni gösterir.

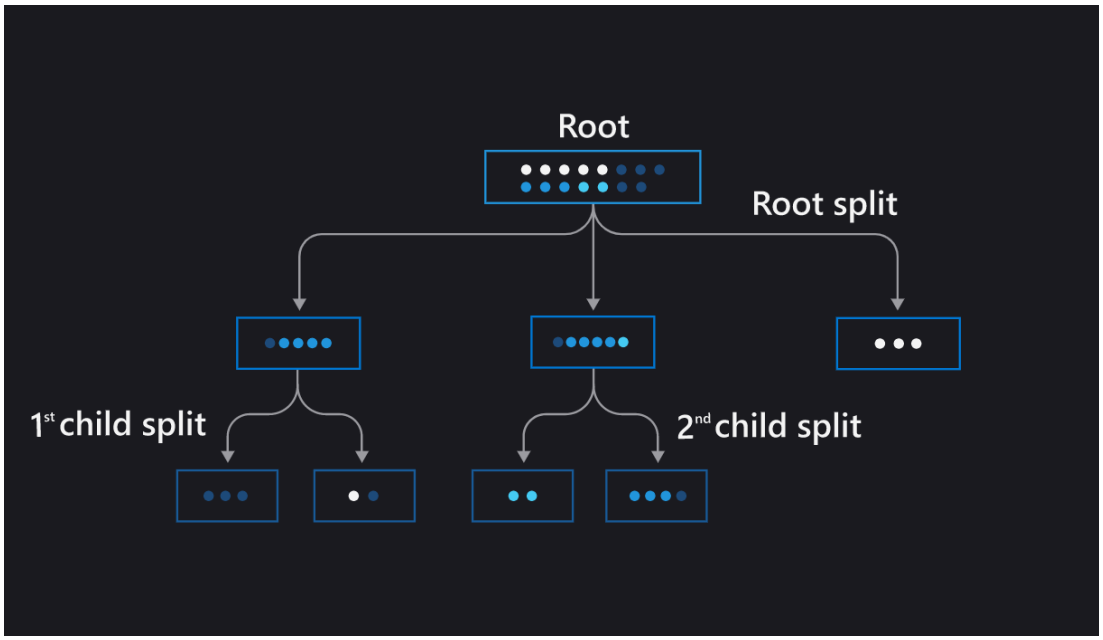
İç düğümler: Kendisinden önce olup kendisine doğru gelen sadece bir dal olan ve kendisinden en az iki veya daha fazla dal çıkan düğümlerdir.

Yaprak veya kutup (terminal) düğümler: Kendisinden önce olup kendisine doğru gelen sadece bir dal olan ve kendisinden hiç dal çıkmayan düğümlerdir.

Düğümler arasındaki testin sonucunu gösteren ve tanımlanacak sınıfın belirlenmesini sağlayan yapı dal olarak adlandırılır. Dalın sonucunda sınıflandırma tamamlanamıyorsa tekrar bir karar düğümü oluşur. Karar ağacında her bir dal sonucunda oluşan düğümlerin bulunduğu yer derinliktir. Derinlik sayısını araştırmacı, karar ağaçlarının veri kümesine uygunluğunun analizini yaparak belirleyebilir. Bir karar ağacında derinlik ile oluşan sınıf sayısı doğru orantılıdır.

Karar ağacı, sorulan sorular ve alınan cevaplar doğrultusunda hareket eder ve sorulan sorulara alınan cevapları birleştirerek kurallar oluşturur. Oluşan ağaç birçok “eğer-ise”(if-then)“ den oluşan kurallar bütünüdür de söylenilebilir. Soru sormaya verideki

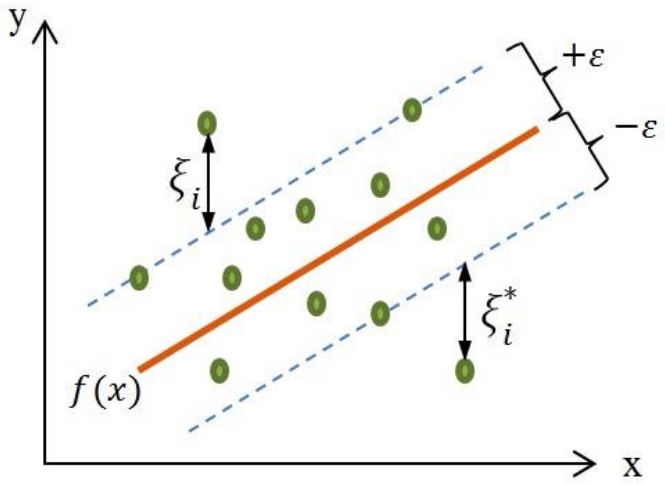
hangi değişkenden başlanacağına karar verildiğinde ilgili değişken ağacın kök düğümünü oluşturmuş olur. Kök düğümü, gerçekleştirilecek testi belirtir. Bu testin sonucu ağacın veri kaybetmeden dallara ayrılmasına neden olur. Her düğümde test ve dallara ayrılma işlemleri ardışık olarak gerçekleşir ve bu ayrılma işlemi üst seviyedeki ayrımlara bağımlıdır. Ağacın her bir dalı sınıflama işlemini tamamlamaya adaydır. Eğer bir dalın ucunda sınıflama işlemi gerçekleşemiyorsa, o daim dalın sonucunda bir karar düğümü oluşur. Ancak daim dalın sonunda belirli bir sınıf oluşuyorsa, o dalın sonunda yaprak vardır. Bu yaprak, veri üzerinde belirlenmek istenen sınıflardan biridir. Karar ağacı işlemi kök düğümünden başlar ve yukarıdan aşağıya doğru yaprağa ulaşana dek ardışık düğümleri takip ederek gerçekleşir. Karar ağacı oluşturulduktan sonra, bir test verisini sınıflandırmak oldukça kolaydır. Kök düğümden başlayarak kayda test koşulu uygulanır ve her sonuç için ona ait uygun dal takip edilir. Buradan ya yeni test koşulunun uygulanacağı başka bir iç düğüme, ya da bir yaprak düğüme ulaşılır. Böylece test verisinin hangi sınıfa ait olduğu hangi yaprakta sonlandığına göre belirlenmiş olur.



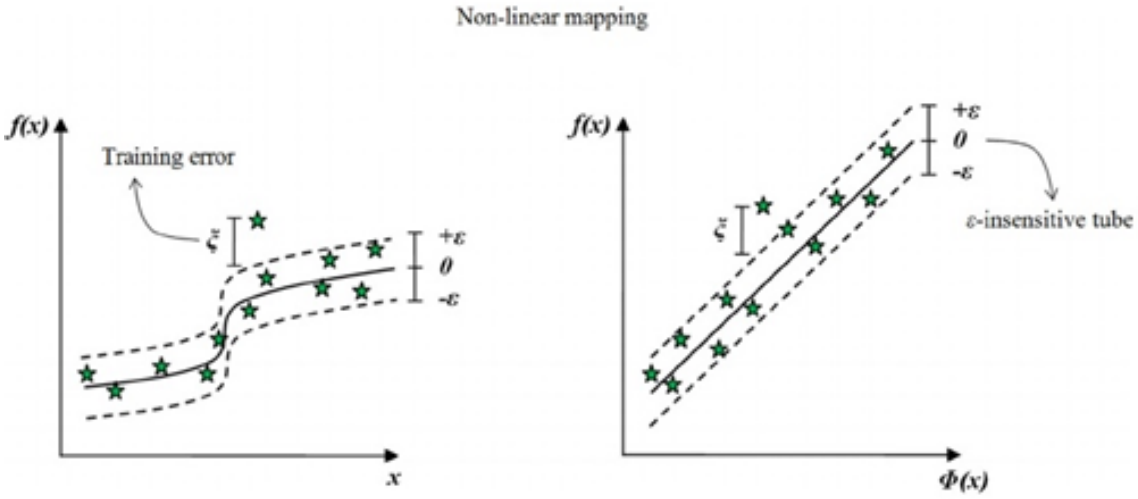
Şekil 2.3. Decision Tree Regression

2.1.4 SUPPORT VECTOR MACHINE REGRESSION

Destek vektör regresyonun temel mantığı doğrusal olarak ayrıştırılabilen veri yapıları için en iyi ayırıcı düzlemin belirlenmesidir. Destek vektör regresyonu sınıflandırıcıları, aralığı maksimum yapan bir en uygun ayırıcı düzlemi oluşturmaya çalışır. Burada bahsedilen aralık kavramı, ayırıcı düzlemden, en yakın veri noktasına olan minimum uzaklığı tanımlamaktadır. Diğer bir deyişle sadece iki sınıfın bulunduğu bir sınıflandırma probleminde Destek vektör regresyonu iki sınıf arasındaki sınırı maksimize eden optimal ayırt etme yüzeyini belirler, yani eğitim kümesi ile ayırt etme yüzeyine en yakın noktaların arasındaki mesafeyi maksimize eder.



Şekil 2.4. SVM

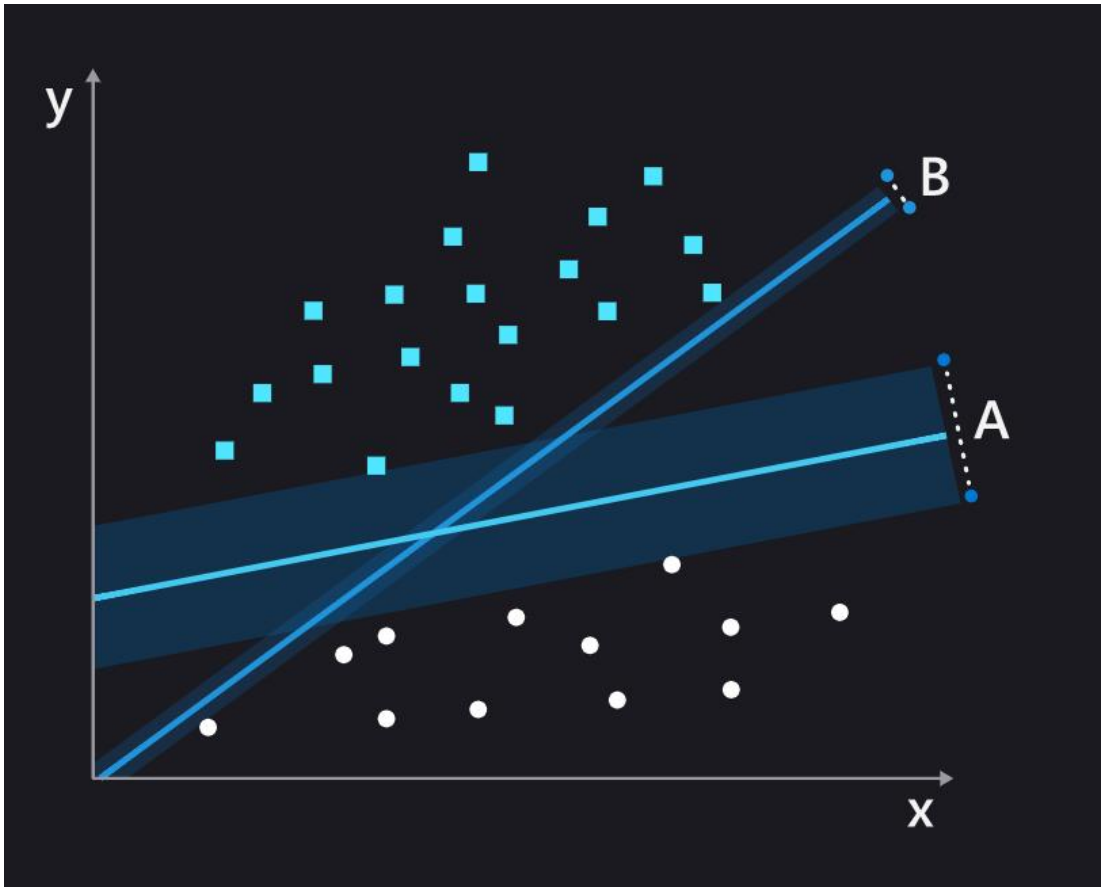


(a) (b)

Şekil 2.5. a) SVM-Non-Linear, b) SVM-Linear

2.1.5 RANDOM FOREST REGRESSION

Rassal orman regresyonu, hiper parametre kestirimi yapılmadan da iyi sonuçlar vermesi hem regresyon hem de sınıflandırma problemlerine uygulanabilir olmasından dolayı popüler makine öğrenmesi modellerinden biridir. Rassal orman regresyonu karar ağaçlarını esas alır. Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenme-veriyi ezberlemedir. Rassal orman modelinde bu problemi çözmek için hem veri setinden hem de öznitelik setinden rassal olarak 10'larca 100'lerce farklı alt-setler seçilir ve bunlar eğitilir. Bu yöntemle 100'lerce karar ağacı oluşturulur ve her bir karar ağacı bireysel olarak tahminde bulunur. Problem regresyon ise karar ağaçlarının tahminlerinin ortalamasını, problem sınıflandırma ise tahminler arasında en çok oy alanı seçilir.



Şekil 2.6. RFR

2.1.6 LightGBM REGRESSION

LightGBM, histogram tabanlı çalışan bir algoritmadır. Sürekli değere sahip olan değişkenleri kesikli hale getirerek hesaplama maliyetini azaltır. Karar ağaçlarının eğitim süresi yapılan hesaplama ve dolayısıyla bölünme sayısı ile doğru orantılıdır. Bu yöntem sayesinde hem eğitim süresi kısalmakta hem de kaynak kullanımı düşmektedir.
Karar ağaçlarında öğreniminde seviye odaklı veya yaprak odaklı olarak iki strateji kullanılabilir. Seviye odaklı stratejide ağaç büyürken ağacın dengesi korunur. Yaprak odaklı stratejide ise kaybı azaltan yapraklardan bölünme işlemi devam eder. LightGBM bu özelliği sayesinde diğer boosting algoritmalarından ayrılmaktadır. Model yaprak odaklı strateji ile daha az hata oranına sahip olur ve daha hızlı öğrenir. Ancak yaprak odaklı büyüme stratejisi veri sayısının az olduğu durumlarda modelin aşırı öğrenmeye yatkın olmasına sebebiyet verir. Bu nedenle algoritma büyük verilerde kullanılmak için daha uygundur. Ayrıca ağaç derinliği, yaprak sayısı gibi parametreler optimize edilip aşırı öğrenmenin önüne geçmeye çalışılabilir.

2.1.6.1 PARAMETRE OPTİMİZASYONU

Num_leaves, ağaçta bulunacak yaprak sayısıdır. Ağacın karmaşıklığını kontrol etmede kullanılan en önemli parametredir. Aşırı öğrenmeden kaçınmak için 2^(max_dept)' den küçük olması gerekmektedir.

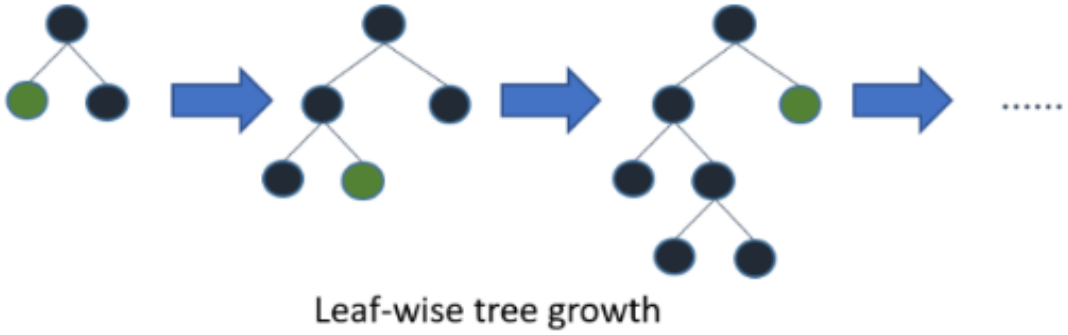
Max_dept, kurulacak ağacın derinliğini limitlemek için kullanılır. Aşırı öğrenmeden kaçınmak için optimum seviyeye getirilmelidir. Çok dallanma aşırı öğrenmeye, az dallanma eksik öğrenmeye sebep olacaktır.

Min_data_in_leaf, aşırı öğrenmeyi engellemek için kullanılacak önemli parametrelerden biridir. Optimum değeri veri büyüklüğüne ve num_leaves'e bağlıdır.

Büyük bir değer olarak ayarlamak ağacın büyümesini engelleyebilir ve eksik öğrenmeye neden olabilir.
Learning_rate, kurulan ağaçları ölçeklendirmek için 0-1 arasında verilen bir değerdir. Bu değerin küçük olması daha iyi tahmin gücüne yardımcı olacaktır. Ancak öğrenim süresini arttıracak ve aşırı öğrenme ihtimalini arttıracaktır.

Feature_fraction, her bir iterasyonda kullanılacak değişken; bagging_fraction, her iterasyonda kullanılacak veri sayısının ayarlanabileceği parametrelerdir.

Num_iteration, öğrenme sürecinde yapılacak iterasyon sayısıdır. feature_fraction, bagging_fraction ve num_iteration sayıları öğrenim süresi ile doğrudan ilgilidir. Bu sayılar ne kadar az olursa öğrenim süresi o kadar az olacaktır. Ancak eksik öğrenmeye dikkat edilmesi çok önemlidir. Birçok deneme yapılarak optimum sayı bulunabilir.



(a)

(b)

Şekil 2.7. a) Yaprak Odaklı Büyüme, b) Seviye Odaklı Büyüme

BÖLÜM 3

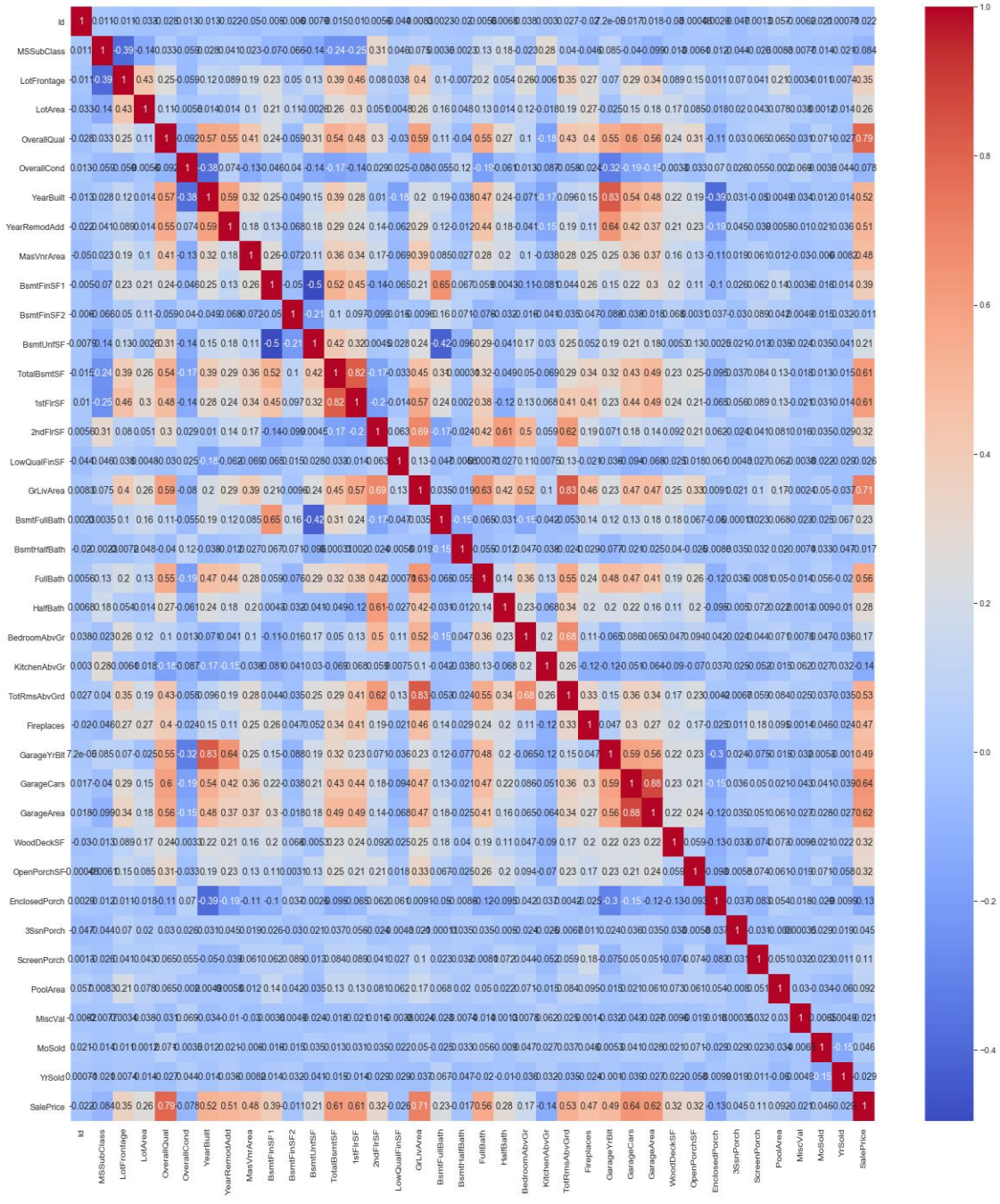
KULLANILACAK VERİLERİN İNCELENMESİ, EKSİK VERİLERİN GİDERİLMESİ VE VERİLERİN HAZIRLIK AŞAMASI

Bu çalışmada makine öğrenmesi modellerinin eğitilmesi için içerisinde 1460 satır ve 81 sütun bulunan eğitim veri setinde ve fiyatı tahmin edilecek olan, içerisinde 1459 satır ve 80 sütun bulunan test veri setlerinden yararlanılacak.

Modelimiz eğitilirken, eğitim veri setinde bulunan 1460 satır ve 80 sütun için her satırda bulunan sütunlardaki verileri inceleyip 81. sütunda bulunan ev fiyatları ile ilişkilendirir, bir kat sayı belirler. Her bir regresyon için model hazırlandıktan sonra ev fiyat tahmini yapılabilecektir.

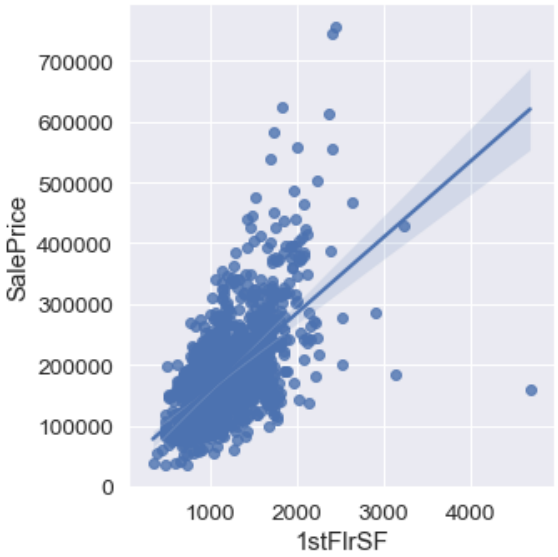
3.1 KORELASYON MATRİSİ

Korelasyon matrisi çoklu değişkenler arasındaki korelasyon katsayılarının tablosudur. Bu tabloda bir değişkenin diğer her değişken ile arasındaki korelasyon görülebilir.[5]

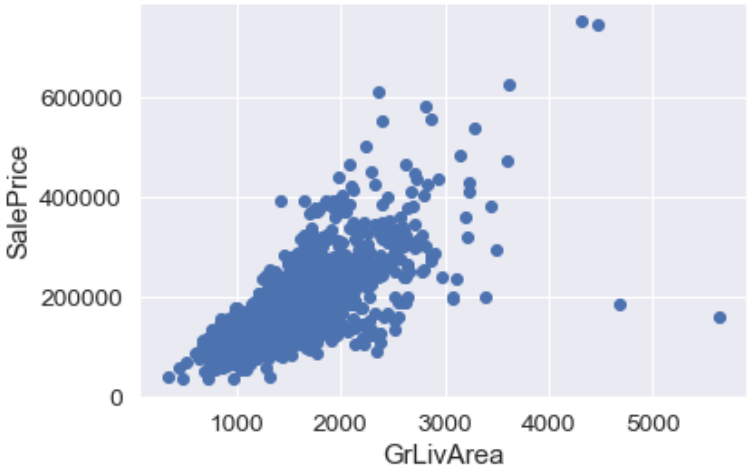


Şekil 3.1 Korelasyon Matrisi

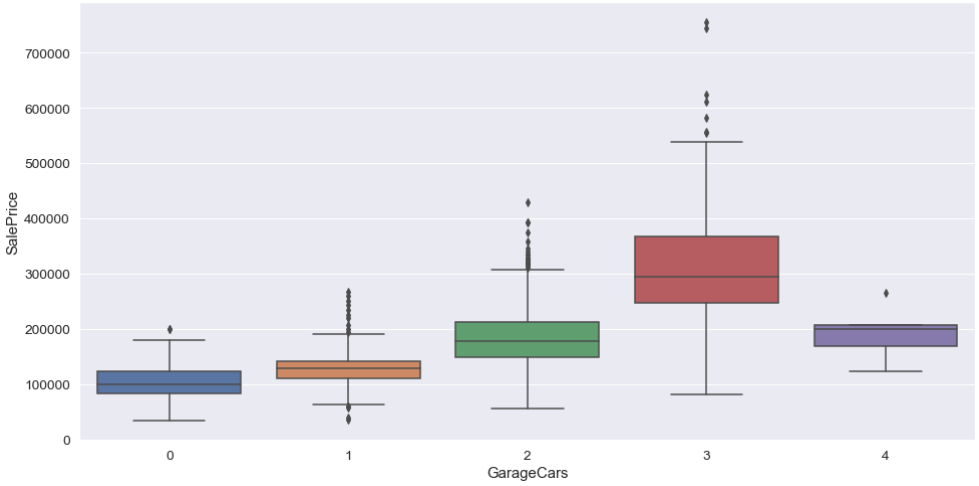
Korelasyon matrisi incelendiğinde '1stFlrSF', 'GrLivArea', 'GarageCars', 'OverallQual', 'GarageArea', 'FullBath' parametreleri ile 'SalePrice' arasındaki ilişkinin güçlü olduğu gözlemlendi.



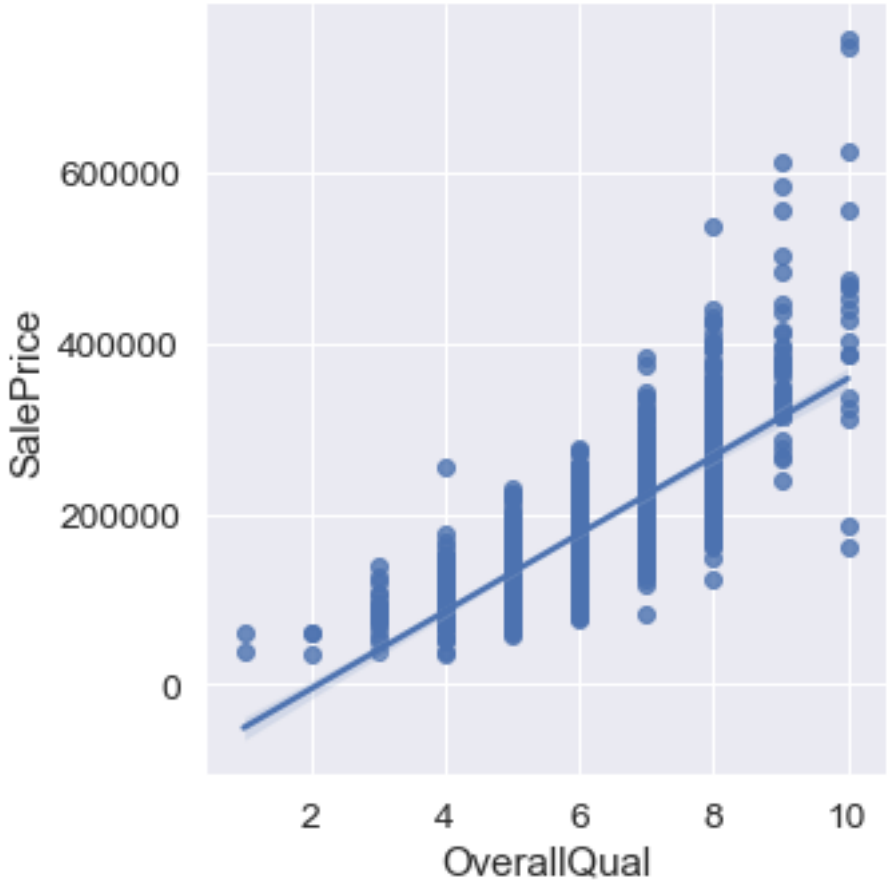
Şekil 3.2 SalePrice-1stFlrSf



Şekil 3.3 SalePrice-GrLivArea



Şekil 3.4 SalePrice-GarageCars



Şekil 3.5 SalePrice-OverallQual

Şekil 3.6 SalePrice-GarageArea

3.2 EKSİK VERİ

	Total	Percent
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageCond	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
BsmtExposure	38	0.026027
BsmtFinType2	38	0.026027
BsmtFinType1	37	0.025342
BsmtCond	37	0.025342
BsmtQual	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685
Utilities	0	0.000000
YearRemodAdd	0	0.000000
MSSubClass	0	0.000000
Foundation	0	0.000000
ExterCond	0	0.000000
ExterQual	0	0.000000

(a)

	Total	Percent
PoolQC	1456	0.997944
MiscFeature	1408	0.965045
Alley	1352	0.926662
Fence	1169	0.801234
FireplaceQu	730	0.500343
LotFrontage	227	0.155586
GarageCond	78	0.053461
GarageQual	78	0.053461
GarageYrBlt	78	0.053461
GarageFinish	78	0.053461
GarageType	76	0.052090
BsmtCond	45	0.030843
BsmtQual	44	0.030158
BsmtExposure	44	0.030158
BsmtFinType1	42	0.028787
BsmtFinType2	42	0.028787
MasVnrType	16	0.010966
MasVnrArea	15	0.010281
MSZoning	4	0.002742
BsmtHalfBath	2	0.001371
Utilities	2	0.001371
Functional	2	0.001371
BsmtFullBath	2	0.001371
BsmtFinSF2	1	0.000685
BsmtFinSF1	1	0.000685

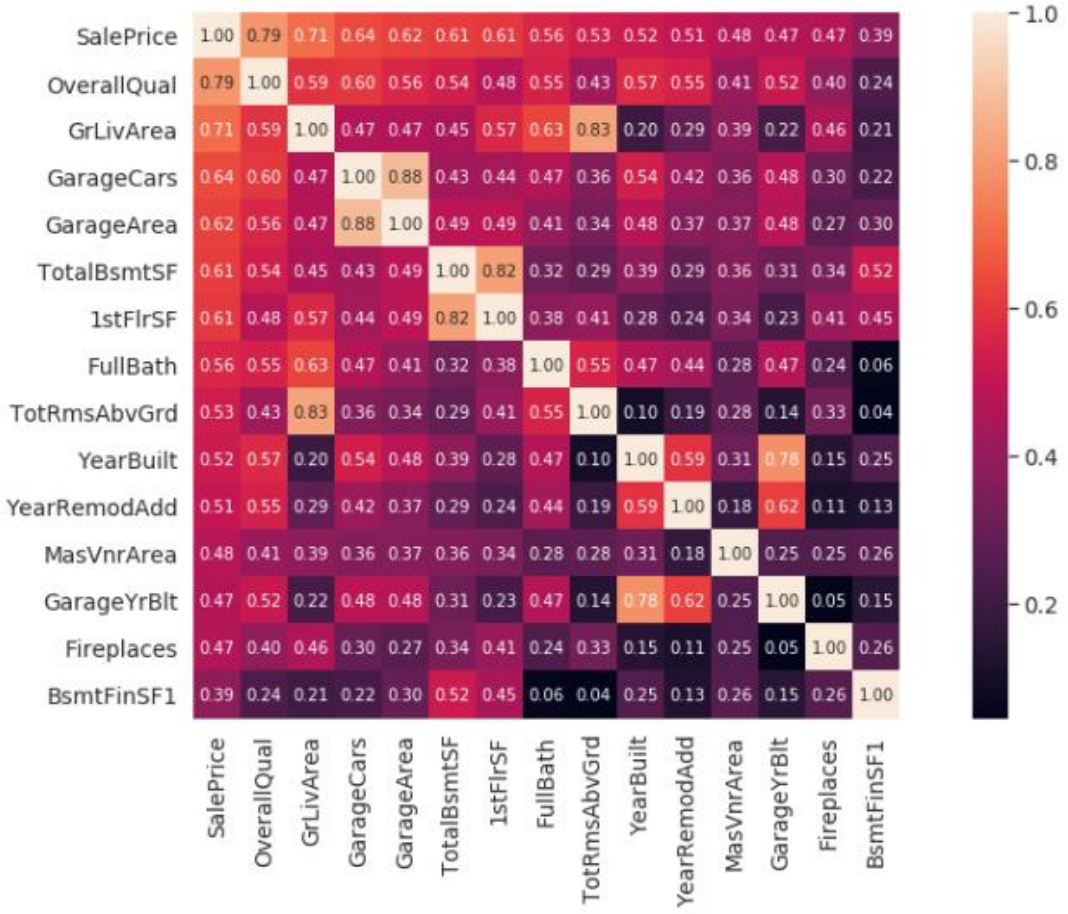
(b)

Şekil 3.7 a) Eğitim Veri Seti Eksik Veriler b)Test Veri Seti Eksik Veriler

Şekil 3.5 incelendiği zaman bazı parametrelerin yüksek oranda eksik. Eğitim veri setinde sınırı 81, test veri setinde sınırı 78 olarak belirleyip, toplamda eksik verileri bu değerlerin üzerinde olan değerleri veri setlerimizden silindi.

Veri setlerimizdeki parametrelerin hala eksik olan değerleri her bir sütunun ortalama değeri ile doldurulduğu.

Veri eğitim veri setimizde bulunan parametreler içinden birbiri ile ilişkisi en kuvvetli 15 parametre seçildi.



Şekil 3.8 Korelasyon Matrisi-2

BÖLÜM 4

MAKİNE ÖĞRENMESİ MODELLEMELERİ

4.1 VERİ SETİ PARÇALAMA VE NORMALİZASYON

Eğitim veri seti x_train, x_test, y_train, y_test olarak bölündü.

X_train: Modele öğrenmesi için verilen parametreler.

X_test: Verilen parametreler karşılığındaki ev değerleri.

y_train: Modelin hiç görmediği parametre değerleri.

y_test: Modelin hiç görmediği parametrelerin ev değerleri.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train.drop('SalePrice', axis=1), train['Sal
ePrice'], test_size=0.3, random_state=101)
```

Şekil 4.1 Veri Seti Parçalama

Normalizasyon, bir popülasyondaki en yüksek ve en düşük değere görece olarak her değerin konumunu hesaplamak için kullanılır[6].

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Şekil 4.2 Normalizasyon

Veri setlerindeki her bir sütunların birbirine baskın gelmemesi için veriler normalize edildi.

```
y_train= y_train.values.reshape(-1,1)
y_test= y_test.values.reshape(-1,1)

from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
y_train = sc_X.fit_transform(y_train)
y_test = sc_y.fit_transform(y_test)
```

,

Şekil 4.3 Veri Seti Normalizasyon

4.2 REGRESYON DEĞERLENDİRME

1- Ortalama Mutlak Hata (MAE), hataların mutlak değerinin ortalamasıdır.
$1n\sum i=1n|yi-\hat{y}\,i|$

2- Ortalama Kare Hatası (MSE) kare hataların ortalamasıdır.
$1n\sum i = 1n\ (yi - \hat{y}\,i)\ 2$

3- Kök Ortalama Kare Hatası (RMSE) kare hataların ortalamasının kare köküdür.[7]
$1n\sum i = 1n\ (yi - \hat{y}\,i)$

Regresyonlarda ki tahmin edilen değer X_test verilerine bakılarak tahmin edilmektedir. X_test eğitim data setinin 4.1 VERİ SETİ PARÇALAMA bölümündeki gibi bölünmesinden sonra elde edilen data settir. Elde edilen tahmin değerleri ile y_test değerleri yukarıda bulunan değerlendirmelere göre değerlendirilir ve en düşük değerli model en iyi eğitilmiş model demektir. Bu model ile test data seti içerisindeki verilere göre bir tahmin yaptırılır.

4.3 LINEAR REGRESSİON

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

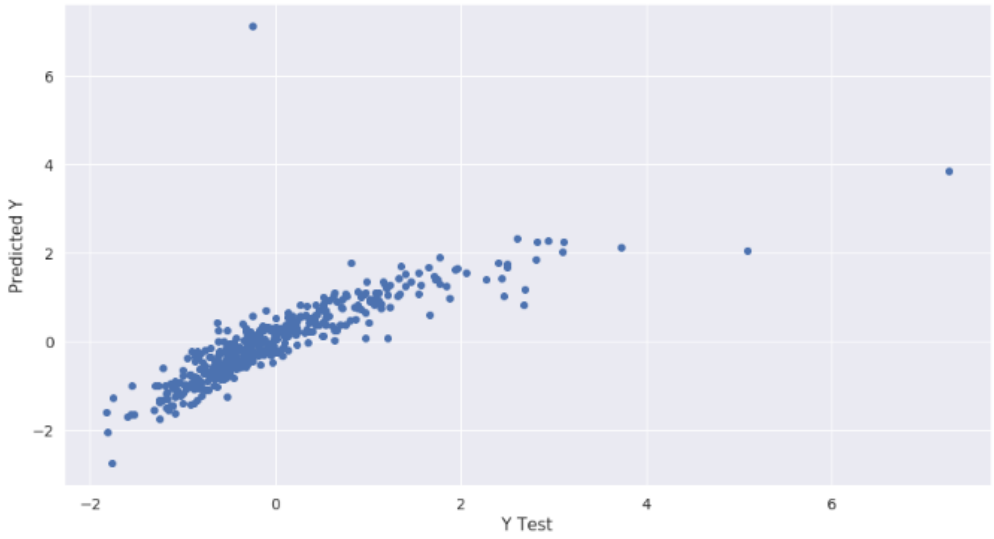
```
lm.fit(X_train,y_train)
```

```
predictions = lm.predict(X_test)
predictions= predictions.reshape(-1,1)
```

Şekil 4.4 Linear Regresyon

Linear Regression Margin of Error:

MAE: 0.29105407971784336
MSE: 0.29995756024517595
RMSE: 0.5476838141164808



Şekil 4.5 Linear Regression Karşılaştırma

4.4 GRADIENT BOOSTING REGRESSION

```
from sklearn import ensemble
from sklearn.utils import shuffle
from sklearn.metrics import mean_squared_error, r2_score
```

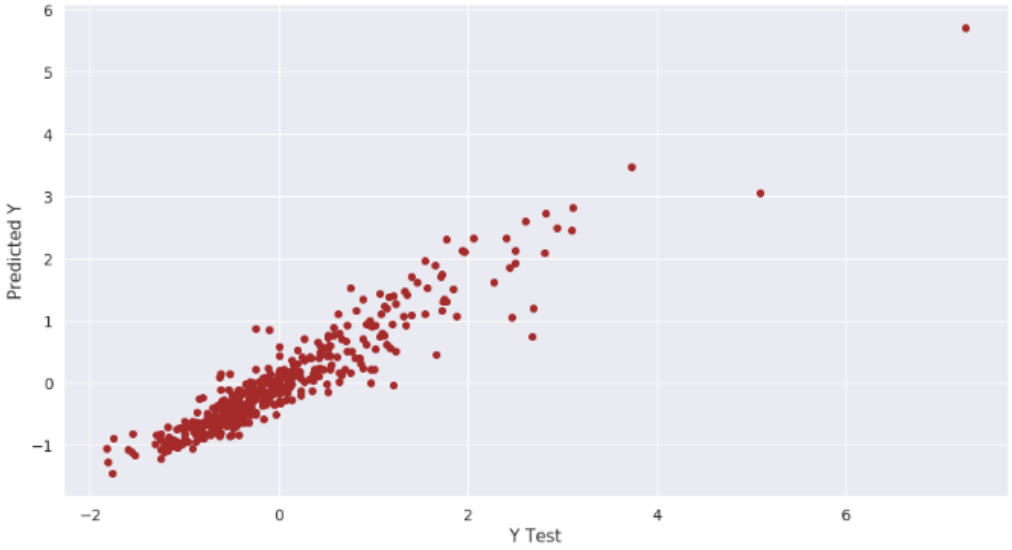
```
params = {'n_estimators': 500, 'max_depth': 4, 'min_samples_split': 2,
          'learning_rate': 0.01, 'loss': 'ls'}
clf = ensemble.GradientBoostingRegressor(**params)

clf.fit(X_train, y_train)
```

```
clf_pred=clf.predict(X_test)
clf_pred= clf_pred.reshape(-1,1)
```

Şekil 4.6 GBR

Gradient Boosting Regression Margin of Error:

MAE: 0.22778255079744467
MSE: 0.11705083087322926
RMSE: 0.34212692216957913



Şekil 4.7 Gradient Boosting Regression Karşılaştırma

4.5 DECISION TREE REGRESSION

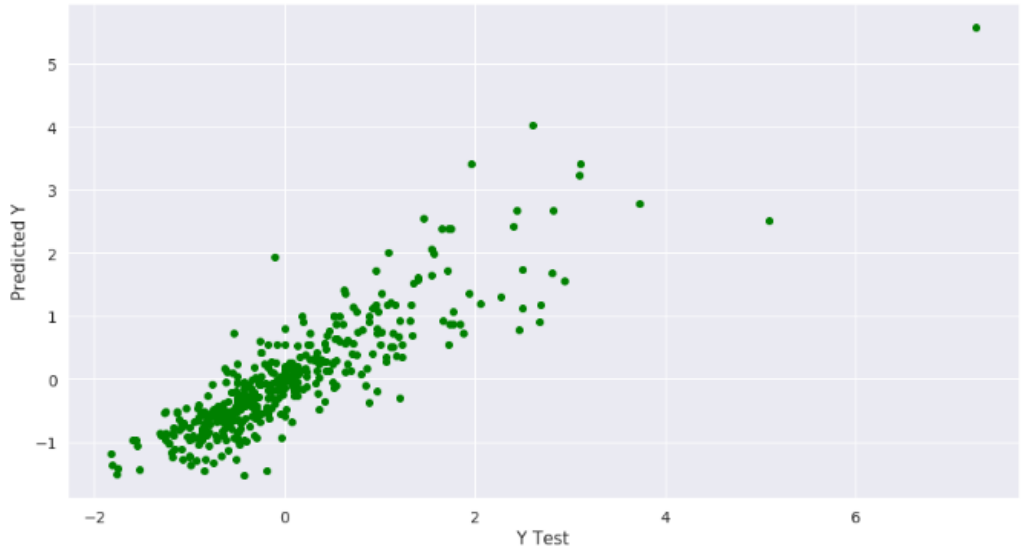
```
from sklearn.tree import DecisionTreeRegressor
dtreg = DecisionTreeRegressor(random_state = 100)
dtreg.fit(X_train, y_train)
```

```
dtr_pred = dtreg.predict(X_test)
dtr_pred= dtr_pred.reshape(-1,1)
```

Şekil 4.8 DTR

Decision Tree Regression Margin of Error:

MAE: 0.3327904077116327
MSE: 0.2297983841710415
RMSE: 0.4793729072142496



Şekil 4.9 Decision Tree Regression Karşılaştırma

4.6 SUPPORT VECTOR MACHINE REGRESSION

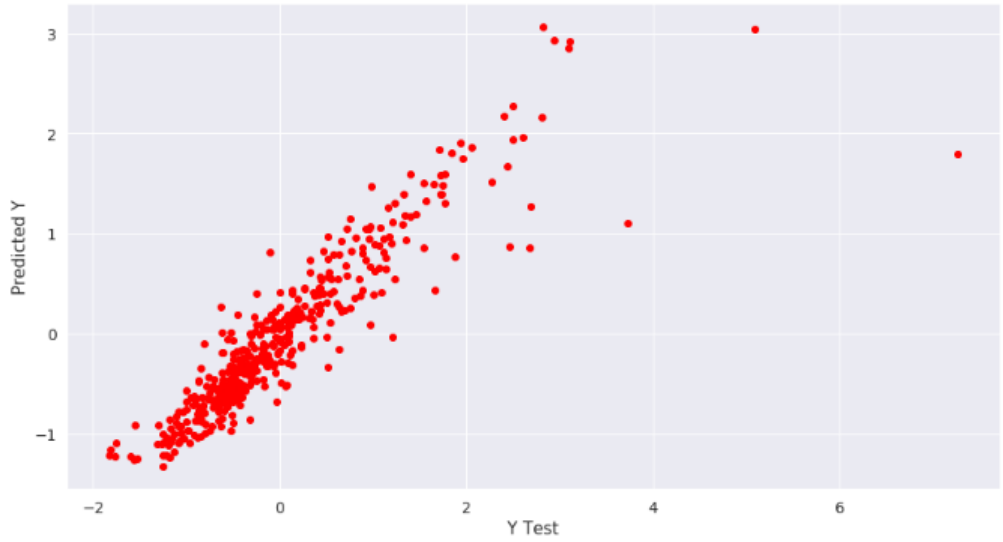
```
from sklearn.svm import SVR
svr = SVR(kernel = 'rbf')
svr.fit(X_train, y_train)
```

```
svr_pred = svr.predict(X_test)
svr_pred= svr_pred.reshape(-1,1)
```

Şekil 4.10 SVM

Support Vector Machine Regression Margin of Error:

MAE: 0.23401679589999025
MSE: 0.1899647870349416
RMSE: 0.43584950044131243



Şekil 4.11 Support Vector Machine Regression Karşılaştırma

4.7 RANDOM FOREST REGRESSION

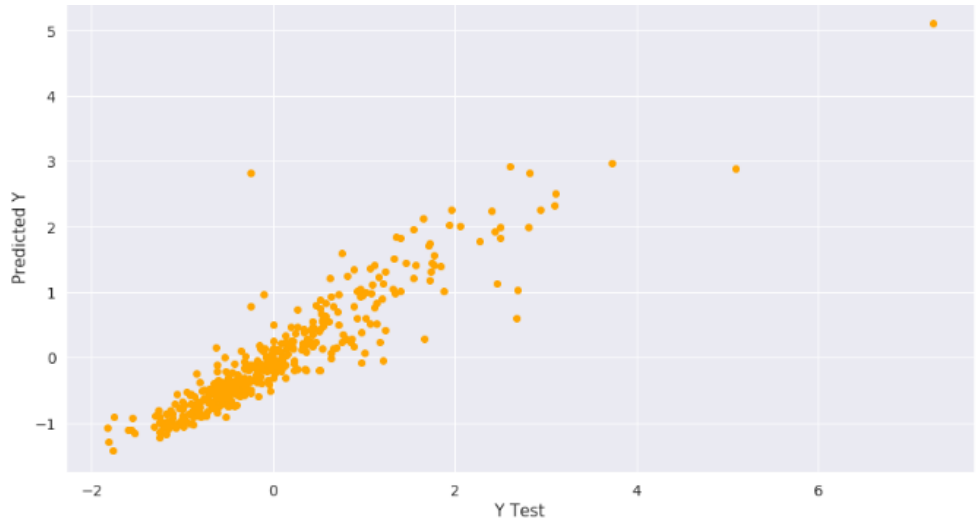
```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(n_estimators = 500, random_state = 0)
rfr.fit(X_train, y_train)
```

```
rfr_pred= rfr.predict(X_test)
rfr_pred = rfr_pred.reshape(-1,1)
```

Şekil 4.12 RFR

Random Forest Regression Margin of Error:

MAE: 0.23723064460015603
MSE: 0.15361242417115376
RMSE: 0.3919342089830304



Şekil 4.13 Random Forest Regression Karşılaştırma

4.8 LightGBM REGRESSION

```
import lightgbm as lgb
```

```
model_lgb = lgb.LGBMRegressor(objective='regression',num_leaves=5,
                              learning_rate=0.01, n_estimators=3000,
                              max_bin = 55, bagging_fraction = 0.8,
                              bagging_freq = 5, feature_fraction = 0.2319,
                              feature_fraction_seed=9, bagging_seed=9,
                              min_data_in_leaf =6, min_sum_hessian_in_leaf = 11)
```

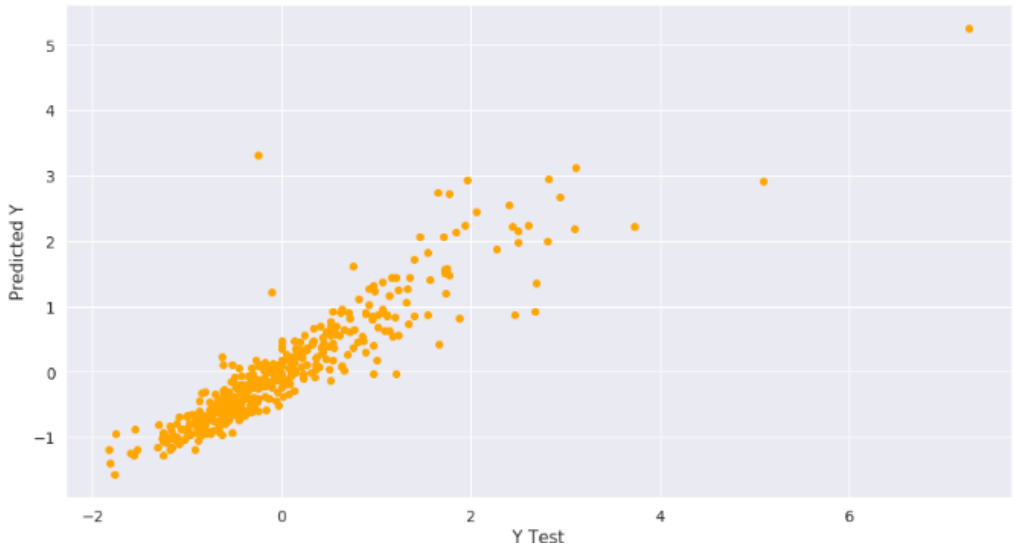
```
model_lgb.fit(X_train,y_train)
```

```
lgb_pred = model_lgb.predict(X_test)
lgb_pred = lgb_pred.reshape(-1,1)
```

Şekil 4.14 LGBM

LightGBM Regression Margin of Error:

MAE: 0.24588472762255686
MSE: 0.15938349328062001
RMSE: 0.39922862282233723



Şekil 4.15 LightGBM Regression Karşılaştırma

4.9 MODEL KARŞILAŞTIRMA

Regresyon modellerinin MSE değerleri şekil 4.9 da ki tabloda görülmektedir.

Regression	Accuracy
Linear Regression	0.299958
Gradient Boosting Regression	0.116288
Decision Tree Regression	0.229798
Support Vector Machine Regression	0.189965
Random Forest Regression	0.153612
LightGBM Regression	0.160357

Şekil 4.16 MSE Değer Karşılaştırması

result - DataFrame

Index	Id	Linear Regression	Gradient Boosting Regression	Decision Tree Regression	Support Vector Machine Regression	Random Forest Regression	LightGBM
0	1461	114106	122628	115302	126699	116287	120406
1	1462	167790	137781	147755	147502	150365	146885
2	1463	183196	177311	170721	185951	167100	182997
3	1464	195795	183699	202425	191327	178957	187576
4	1465	200241	210275	211162	182366	211125	195660
5	1466	175639	172800	178710	174992	176998	176769
6	1467	181580	170618	173717	176337	167979	170636
7	1468	168268	166572	174715	161959	174605	169292
8	1469	207668	182689	170721	197057	185203	188455
9	1470	109432	118555	94831.8	123394	118019	118013
10	1471	218969	192876	182704	214196	197355	201750
11	1472	125238	96000.2	97827.5	90812	94673.4	93270.3
12	1473	123193	101283	103819	101799	104339	100876
13	1474	167194	164133	143761	164076	162641	161406
14	1475	127929	142194	164730	125517	136836	129550
15	1476	312195	363208	287551	361463	370813	389248
16	1477	248679	259324	284645	246878	266860	261503
17	1478	288195	291677	278564	288868	291982	294827
18	1479	270724	257911	144759	292691	244751	271866
19	1480	426211	471792	434491	356290	473258	515463
20	1481	299669	310392	252602	313838	287143	280598
21	1482	221372	215691	200178	209461	211460	216234
22	1483	172686	170118	167716	168287	173886	164011
23	1484	176752	176537	157940	164260	170421	182039
24	1485	182802	171970	186698	172567	171511	175844

Şekil 4.17 Tüm Modellerin Tahmin 25 Adet Tahmin Değeri

BÖLÜM 5

SONUÇ VE DEĞERLENDİRME

Ev alma veya ev satma ciddi ücretler karşılığında yapılan faaliyetlerden biridir. Yanlış alınabilecek kararlardan dolayı alınacak eve gereğinden fazla ödeme yapılabilir veya ev satılırken gereğinden daha düşük bir ücrete satılabilir.

Bu çalışmada Linear Regression, Gredient Boosting Regression, Decision Tree Regression, Support Vector Regression, Random Forest Regression, LightGBM Regression olmak üzere toplam 6 adet regresyon üzerine çalışmalar yapılmıştır.

Yapılan modellemelerin hepsi ile tahmin yapıldı ve incelendi. Elde edilen değerler birbirlerine çok uzak olmadığı gözlemlendi. Model karşılaştırma bölümünde görüldüğü üzere bu veri seti için Gradient Boosting Regression bu veri seti için en iyi sonucu verdi.

Bu çalışma için en iyi değerleri vermiş olan regresyon çeşidi her zaman en iyisidir demek son derece yanlış olacaktır. Kullanılan regresyon çeşidi veri setlerinin yapısına göre veya elde edinilmek istenen sonuca göre değişkenlik gösterebiliyor.

Bu çalışma sonucunda kazanılması hedeflenen veri bilimi ile ilgili son derece yararlı bilgiler kazanıldı. Yapılan bu çalışma bundan sonraki çalışmalara yardımcı olacak, ışık tutacaktır.

KAYNAKLAR

1. İnternet: "ŞEHİRLEŞME, MEKÂN – İNSAN ETKİLEŞİMİNİN BİREY ALGISINA YANSIMASI: BİR VERİ MADENCİLİĞİ ANALİZİ" https://dergipark.org.tr/tr/download/article-file/327425

2. İnternet: "How to start incorporating machine learning in the enterprise arena", https://readwrite.com/2017/06/15/incorporating-machine-learning-enterprises/

3. Xiaojin Zhu, "What is semi-supervised learning? ", Computer Sciences Department, USA (3): 3-5 (2005).

4. D. A. Freedman, Statistical Models. Cambridge University Press, 2009.

5. İnternet: "R ile Korelasyon, Regresyon ve Zaman Serisi Analizleri",http://devveri.com/veri-madenciligi/r-ile-korelasyon-regresyon-ve-zaman-serisi-analizleri

6. İnternet: "İstatistiksel Normalleştirme", http://bilgisayarkavramlari.sadievrenseker.com/2012/01/29/istatistiksel-normallestirme-statistical-normalisation/

7. İnternet: "Metrics and scoring: quantifying the quality of predictions", https://scikit-learn.org/stable/modules/model_evaluation.html

ÖZGEÇMİŞ

Mustafa SARITEMUR 1996 yılında İstanbul'da doğdu; ilk ve orta öğretimini İstanbul'da tamamladı. 2015 yılında Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde öğrenime başlayıp 2020 yılında mezun olmayı planlıyor.

Uzmanlık Alanları Makine Öğrenmesi, Derin Öğrenme ve Görüntü İşleme.

ADRES BİLGİLERİ

Adres : Şemsi Paşa Mah. Şakir Zümre Cad. 39.Sok No12 D5 Küçükköy/Gaziosmanpaşa/İstanbul

Tel : (531) 694 76 18

E-posta : mustafa.saritemur@hotmail.com