



Ağaç Tabanlı Yapay Öğrenme Algoritmalarına Giriş

R Uygulaması İle

Arş. Gör. Ozancan Özdemir
İstatistik Bölümü, Orta Doğu Teknik Üniversitesi

Ankara, 2022


Ozancan Özdemir



- **Lisans:** ODTÜ İstatistik Bölümü, 2017
- **Yüksek Lisans:** ODTÜ İstatistik Bölümü, 2020
Tez Başlığı: Performance Comparison of Machine Learning Methods and Traditional Time Series Methods for Forecasting.
(Danışman: Prof.Dr.Ceylan Yozgatlıgil)
- **Doktora:** ODTÜ İstatistik Bölümü, Devam Ediyor
- **Araştırma Alanları:** Yapay ve Derin Öğrenme, Zaman Serileri, Ardışık Veri, Veri Görselleştirme

-
- **Araştırma Görevlisi,** ODTÜ İstatistik Bölümü, 2017- ..
 - **Kurucu Ortak,** Veripie, 2020-... (<http://www.veripie.com.tr/>)



-
- **WhyR Turkey** (<https://why.pl/2022/turkey/>)
 -  Açık Veri, Veri Bazlı Politika, COVID-19, R Programlama



ozancan@metu.edu.tr



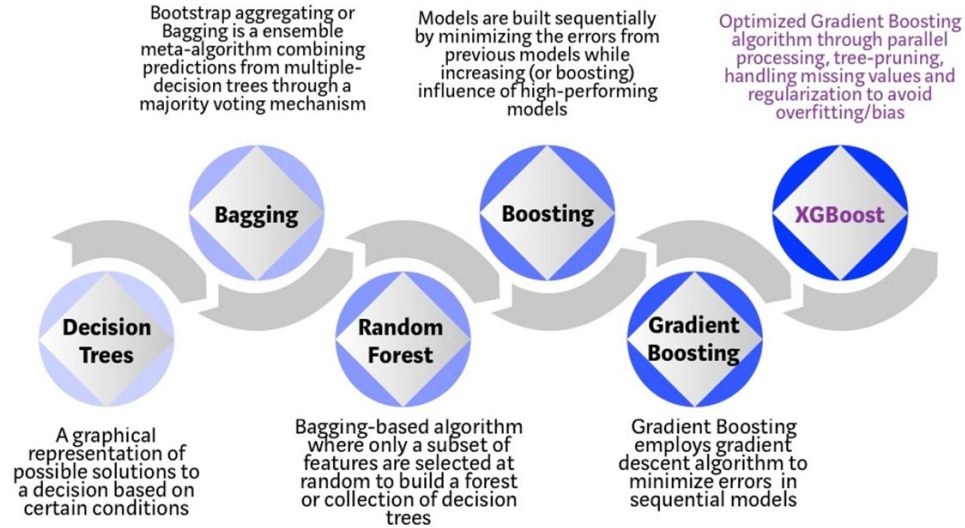
<http://users.metu.edu.tr/ozancan>



<https://twitter.com/OzancanOzdemir>



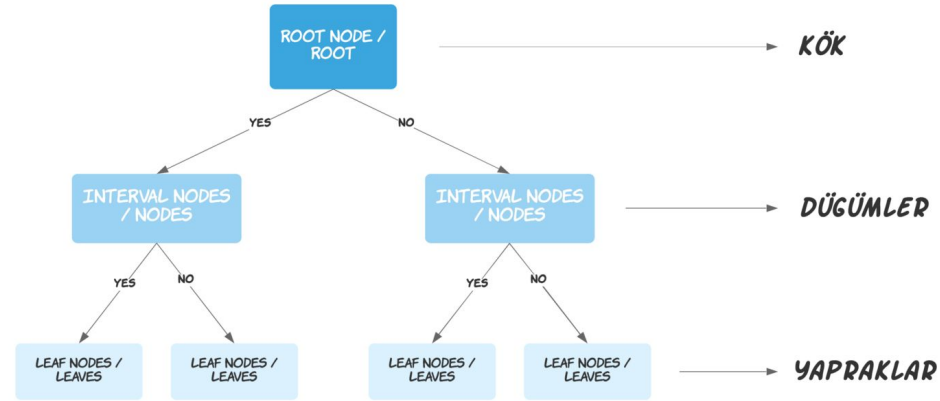
<https://github.com/ozancanozdemir>



Karar Ağaçlarından Ekstrem Gradyan Artırım Modeline...

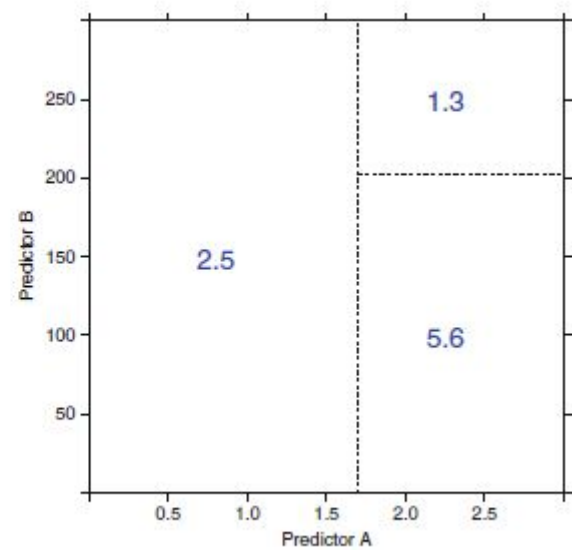
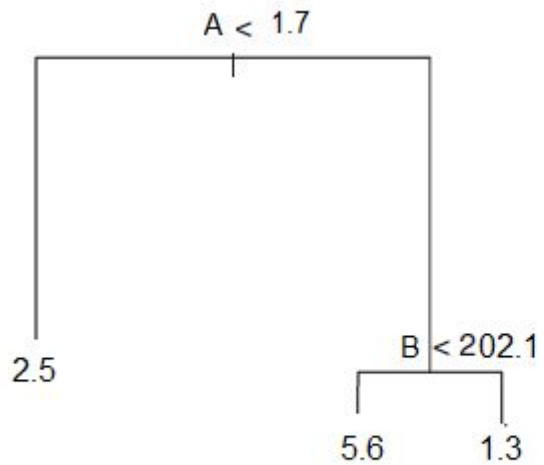
<https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>

Karar Ağacı



<https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5>

Karar Ağacı





Neden Karar Ağacı?

- Gözetimli öğrenme teknikleridir
- Hem bağlantım (regression) hem sınıflandırma (classification) problemlerine uygulanabilir.
- İnsan karar alma mekanizmasına oldukça benzerdir.
- Uygulaması ve sonuçlarının yorumlanması kolaydır.
- Diğer gözetimli öğrenme tekniklerinin aksine, örneğin yapay sinir ağları, veri ön işlemesine ihtiyaç duymazlar.
- Modellerin uygulanması için ön gereklilik ihtiyaçları yoktur.
- Kayıp veriler ile kendi içlerinde baş edebilirler.
- Sadece tahmin yapmak için değil, öznelilik seçimi (feature selection) amacı ile de kullanılabilir.
- Veri madenciliğinde de sıkça kullanılır.

Bazı Karar Ağacı Türleri | Kullanım Örnekleri

- ID3
 - CART
 - Rastgele Ormanlar
 - Torbalama
 - Takviyeleme
 - C5.0
 - C4.5
 - CHAID
- Sahtekarlık Tespiti
 - Kalp Hastalıkları Tespiti
 - Kredi Skor Hesaplaması
 - BP'nin açık deniz platformlarında gaz ve petrolü ayırmaya yönelik GasOIL sistemi, (C.4.5) uzmanların tahmininden daha iyi performans gösterdi ve BP milyonlarını kurtardı. (1986)

Bağlanım ve Sınıflandırma Ağaçları (CART)

Bağlanım Ağaçları

- Breiman, Freedman, Olshen, Stone, 1984.
- Kısaca çalışma adımları
 - Öznitelik uzayı J tane ayrık ve çakışmayan bölgelere ayrılıyor (R_1, \dots, R_J)
 - R_j bölgesindeki eğitim verisinin çıktı değerlerinin (y_i) ortalaması tahmin olarak kullanılıyor.
- **Amaç:** Artık Kareler Toplamının (RSS) en küçük olduğu bölgelerin bulunması

$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

\hat{y}_{R_j} : R_j bölgesindeki çıktıların ortalaması



Bağlanım ve Sınıflandırma Ağaçları (CART)

Bağlanım Ağaçları

Özyinelemeli İkili Ayırma (Recursive Binary Splitting)

İlk olarak

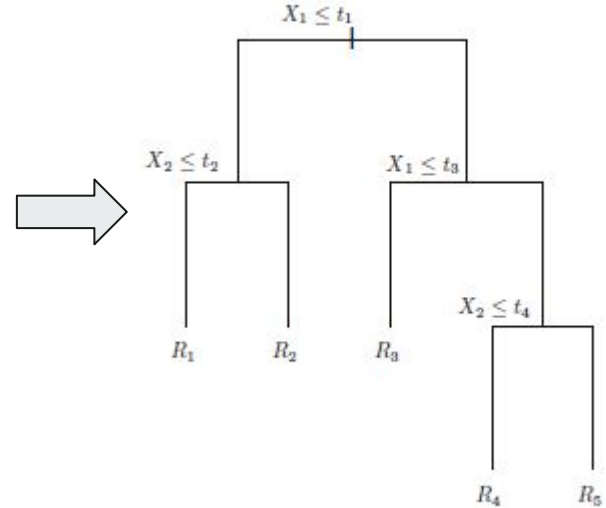
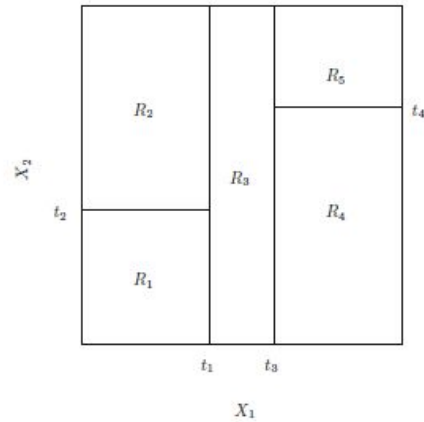
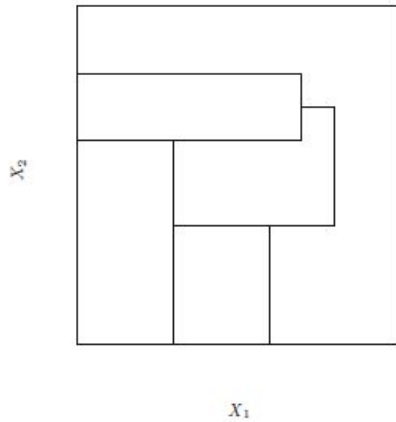
- Tüm değişkenler (X_1, \dots, X_p) ve değerler sıralanır.
- Maliyet fonksiyonunda (RSS) en fazla azalmanın sağlandığı değişken ve ayırım noktası bulunur.

Daha sonra

- Tüm değişkenler yerine bir önceki adımda bölünmüş olan bölgelerden biri kullanılır.
- Terminal düğümlerde çok az sayıda, örneğin 5, gözlem kalınca durur.

Bağlanım ve Sınıflandırma Ağaçları (CART)

Bağlanım Ağaçları



Ağaç Budama (Tree Pruning)

- **Amaç:** Kompleks ağaçlar sonucu meydana gelebilecek aşırı öğrenme problemini engellemek.

***Aşırı öğrenme:** Yüksek eğitim verisi performansı, düşük test verisi performansı.

- Yüksek eşik değeri (treshhold value) kullanarak daha sık ağaçlar elde etmek
- Kompleks bir ağaç oluşturup onu budamak. (prune)



Ağaç Budama (Tree Pruning)

- Maliyet Karmaşıklık Budaması (Cost complexity pruning)

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

$T \subset T_0$: alt ağaç (subtree)

$|T|$: T ağacındaki terminal düğüm sayısı

R_m : m. terminal düğüme karşılık gelen bölge

α : sabit parametre



Çapraz geçerlilik (cross-validation)
yardımı ile elde edilir.

- Sabit parametre arttıkça ağaç boyutu küçülüyor.

Bağlanım ve Sınıflandırma Ağaçları (CART)

Sınıflandırma Ağaçları

- **Amaç:** Saflık derecesinin en yüksek olduğu bölgelerin bulunması. (Hata oranının düşük olduğu bölgelerin bulunması)

Sınıflandırma Hata Oranı (Classification Error Rate): $E = 1 - \max_k(\hat{p}_{mk}).$

Entropi (Entropy): $D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$

Gini: $G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk}).$

\hat{p}_{mk}

:m. bölgedeki eğitim verisindeki k. sınıftan olanların oranı

- **Tahmin:** Rj bölgesine düşen gözlemler arasında sıklığı yüksek olan sınıf kullanılıyor.

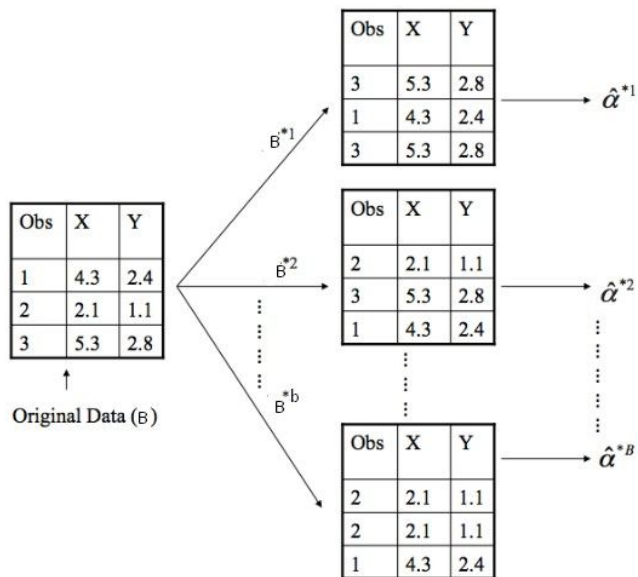


Bağlanım ve Sınıflandırma Ağaçları (CART)

- Her zaman gösterilen örneklerin aksine iyi bir performansa sahip değildir.
- Veri setindeki değişikliklere karşı sağlam değildir.
- Yüksek varyans sorunu yaşayabilir.

Torbalama (Bootstrap Aggregating ya da Bagging)

Zorlama (Bootstrap)



Torbalama (Bagging)

- Leo Breiman, 1996.
- Topluluk yöntemidir.
- Sadece karar ağaçları değil, diğer yöntemlere de uygulanabilir.
- **Amaç:** Yüksek varyansı düşürmek birden fazla ağaç oluşturarak onlardan elde edilecek tahminlerin ortalamasını (bağlanım problemleri) ya da sıklığı fazla olanı (sınıflandırma) hesaplamak ve daha stabil tahminler elde etmek.

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$$



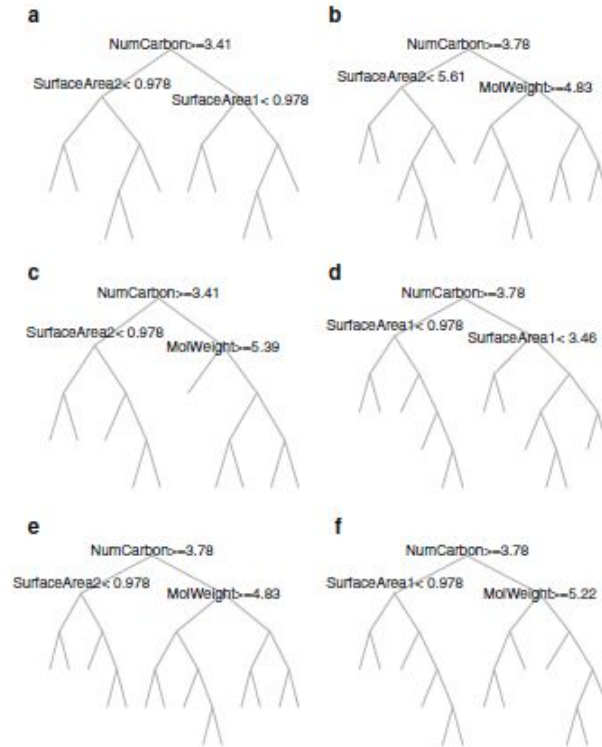
Torbalama (Bagging)

Torba Dışı Hata (Out of Bag Error-OOB)

- Her ağaç için gözlemlerin %63.2'si kullanılır.
- Tüm işlem sonunda gözlemlerin tamamı en az bir kere olsun kullanılmış olur.
- Modellerin performansını ölçmekte kullanılır.

Torbalama (Bagging)

Applied Predictive Modeling, M. Kuhn,
K. Johnson., Springer, 2013



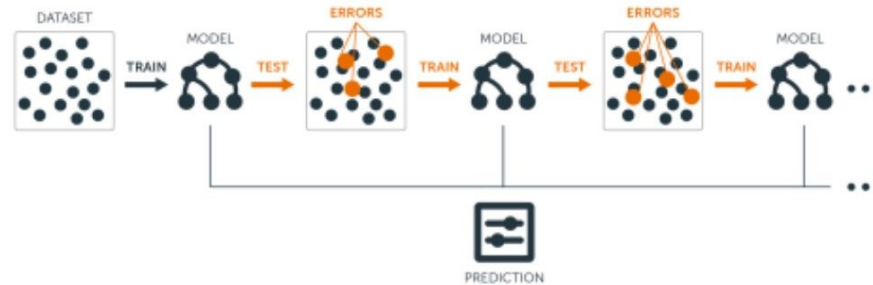


Rastgele Ormanlar (Rassal Ormanlar)

- Torbalama sonucu ortaya çıkan ilişkili ağaçları bağımsız bir hale dönüştürmek için geliştirilmiştir.
- Torbalama ile aynı çalışma disiplinine sahiptir; topluluk yöntemi.
- Her bir ağaç için rassal sayıda seçilen değişkenler kullanılır. ($m \approx \sqrt{p}$)
- Birbirleriyle ilişkili özniteliklerin olduğu veri setlerinde kullanışlıdır.

Takviye (Boosting)

- Schapire, 1990.
- Sadece karar ağaçları değil, diğer yöntemlere de uygulanabilir.
- **Amaç:** Birden fazla sıg ve zayıf olmayan ağacı birleştirerek güçlü ve tek bir ağaç oluşturmak.
- Ağaçlar ardışık bir şekilde oluşturulur.



Takviye (Boosting)

- Sığ bir ağaçtan tahmin elde et.
- Artık değer hesapla ve onu kullanarak sığ bir ağaç oluştur.
- Elde ettiğin değeri ilk tahmine ekle ve yeniden artık değer hesapla.
- Yeni artık değerler üzerine tekrar sığ bir ağaç oluştur.
- Elde edilen yeni tahmini bir önceki tahmine ekle.

$$F_1(x) = y,$$

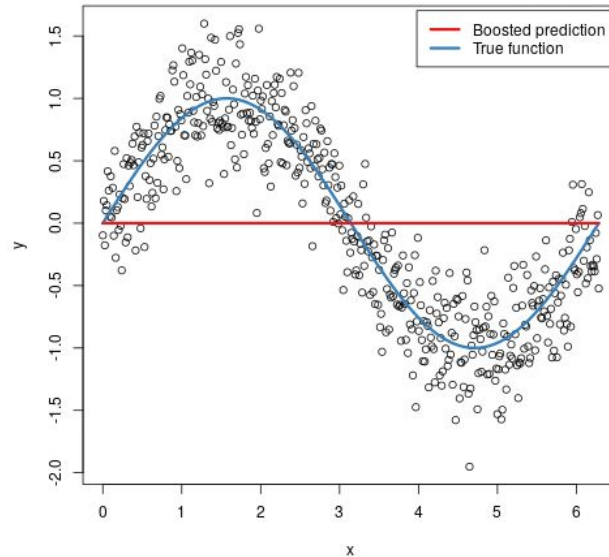
$$h_1(x) = y - F_1(x)$$

$$F_2(x) = F_1(x) + h_1(x)$$

$$F_2(x): h_2(x) = y - F_2(x)$$

$$F_3(x) = F_2(x) + h_2(x)$$

Takviye (Boosting)





Gradyan Takviye ve Ekstrem Gradyan Takviye (Gradient Boosting & XGBoost)

Gradyan Takviye

- Friedman, 1999.
- Ağaçlar artık değerler kullanarak değil, artık değerlerin gradyanları kullanılarak oluşturuluyor.

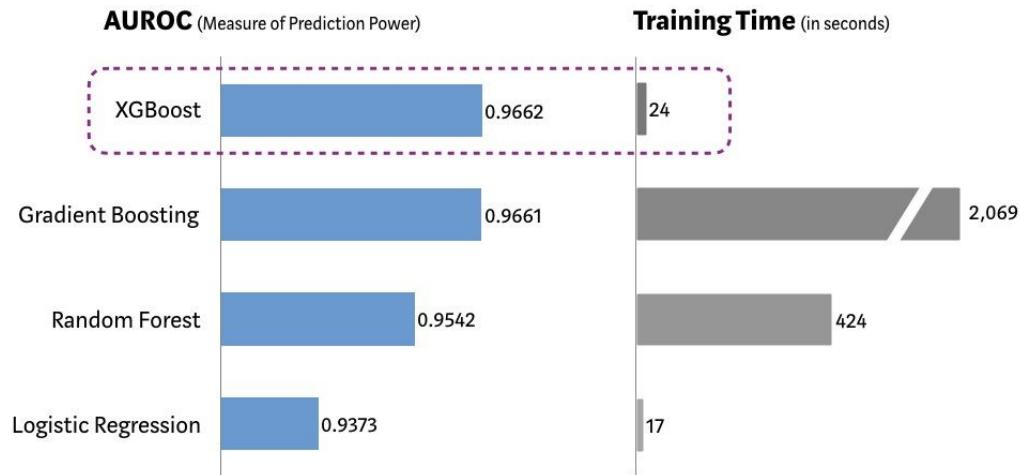
Ekstrem Gradyan Takviye

- Chen ve Guestrin, 2016.
- Regülerizasyon, Erken Durma, Paralel İşleme, Maliyet Fonksiyonu, Yeniden Kullanılabilirlik, Farklı basit tahmin modelleri.

Gradyan Takviye ve Ekstrem Gradyan Takviye

Performance Comparison using SKLearn's 'Make_Classification' Dataset



(5 Fold Cross Validation, 1MM randomly generated data sample, 20 features)



R Uygulaması (Bağlanım Problemi)

```
> library(caret) #model ve veri seti kütüphanesi
> library(ggplot2) #veri görselleştirme
> library(GGally) #veri görselleştirme
> library(tidyverse) #veri manipülasyonu
> data(Sacramento) #veri setini yükle
> head(Sacramento) #ilk 6 satırı görüntüle
```

	city	zip	beds	baths	sqft	type	price	latitude	longitude
1	SACRAMENTO	z95838	2	1	836	Residential	59222	38.63191	-121.4349
2	SACRAMENTO	z95823	3	1	1167	Residential	68212	38.47890	-121.4310
3	SACRAMENTO	z95815	2	1	796	Residential	68880	38.61830	-121.4438
4	SACRAMENTO	z95815	2	1	852	Residential	69307	38.61684	-121.4391
5	SACRAMENTO	z95824	2	1	797	Residential	81900	38.51947	-121.4358
6	SACRAMENTO	z95841	3	1	1122	Condo	89921	38.66260	-121.3278



```
> str(Sacramento) #değişkenlerin yapılarını gösteriyor
'data.frame': 932 obs. of 9 variables:
 $ city      : Factor w/ 37 levels "ANTELOPE","AUBURN",...: 34 34 34 34 34 34 34 34 29 31 ...
 $ zip       : Factor w/ 68 levels "z95603","z95608",...: 64 52 44 44 53 65 66 49 24 25 ...
 $ beds      : int  2 3 2 2 2 3 3 3 2 3 ...
 $ baths     : num  1 1 1 1 1 1 2 1 2 2 ...
 $ sqft      : int  836 1167 796 852 797 1122 1104 1177 941 1146 ...
 $ type      : Factor w/ 3 levels "Condo","Multi_Family",...: 3 3 3 3 3 1 3 3 1 3 ...
 $ price     : int  59222 68212 68880 69307 81900 89921 90895 91002 94905 98937 ...
 $ latitude  : num  38.6 38.5 38.6 38.6 38.5 ...
 $ longitude : num  -121 -121 -121 -121 -121 ...
```

```
> sacramento_ev <- Sacramento%>%subset(city == "SACRAMENTO")
> head(sacramento_ev)
```

	city	zip	beds	baths	sqft	type	price	latitude	longitude
1	SACRAMENTO	z95838	2	1	836	Residential	59222	38.63191	-121.4349
2	SACRAMENTO	z95823	3	1	1167	Residential	68212	38.47890	-121.4310
3	SACRAMENTO	z95815	2	1	796	Residential	68880	38.61830	-121.4438
4	SACRAMENTO	z95815	2	1	852	Residential	69307	38.61684	-121.4391
5	SACRAMENTO	z95824	2	1	797	Residential	81900	38.51947	-121.4358
6	SACRAMENTO	z95841	3	1	1122	Condo	89921	38.66260	-121.3278

```
> str(sacramento_ev)
'data.frame': 438 obs. of 9 variables:
 $ city      : Factor w/ 37 levels "ANTELOPE","AUBURN",...: 34 34 34 34 34 34 34 34 34 34 ...
 $ zip       : Factor w/ 68 levels "z95603","z95608",...: 64 52 44 44 53 65 66 49 64 52 ...
 $ beds      : int  2 3 2 2 2 3 3 3 3 3 ...
 $ baths     : num  1 1 1 1 1 1 2 1 2 2 ...
 $ sqft      : int  836 1167 796 852 797 1122 1104 1177 909 1289 ...
 $ type      : Factor w/ 3 levels "Condo","Multi_Family",...: 3 3 3 3 3 1 3 3 3 3 ...
 $ price     : int  59222 68212 68880 69307 81900 89921 90895 91002 100309 106250 ...
 $ latitude  : num  38.6 38.5 38.6 38.6 38.5 ...
 $ longitude : num  -121 -121 -121 -121 -121 ...
```

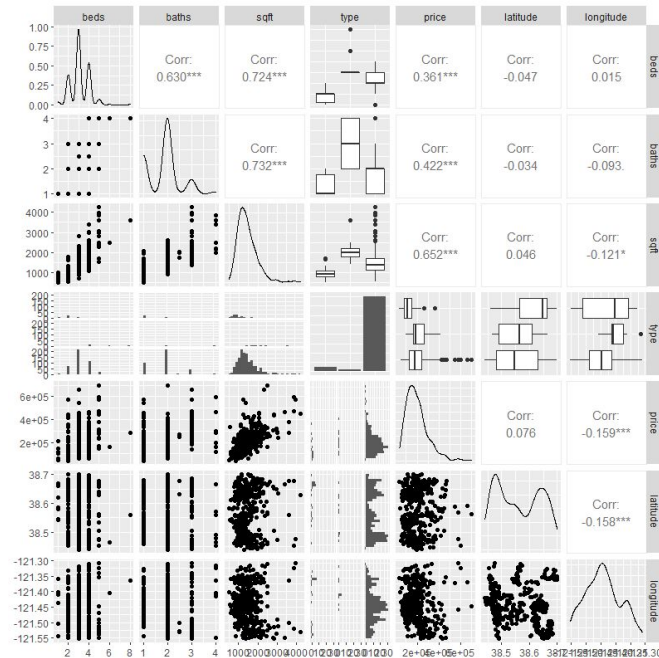
> summary(sacramento_ev) #tanımlayıcı istatistikler ve sıklık değerleri


city	zip	beds	baths	sqft
SACRAMENTO :438	z95823 : 61	Min. :1.00	Min. :1.000	Min. : 484
ANTELOPE : 0	z95828 : 45	1st Qu.:3.00	1st Qu.:1.000	1st Qu.:1100
AUBURN : 0	z95835 : 37	Median :3.00	Median :2.000	Median :1355
CAMERON_PARK : 0	z95838 : 37	Mean :3.13	Mean :1.848	Mean :1453
CARMICHAEL : 0	z95822 : 24	3rd Qu.:4.00	3rd Qu.:2.000	3rd Qu.:1682
CITRUS_HEIGHTS: 0	z95820 : 23	Max. :8.00	Max. :4.000	Max. :4246
(Other) : 0	(Other):211			

type	price	latitude	longitude
Condo : 26	Min. : 40000	Min. :38.44	Min. : -121.6
Multi_Family: 10	1st Qu.:124325	1st Qu.:38.48	1st Qu.: -121.5
Residential :402	Median :178240	Median :38.56	Median : -121.4
	Mean :197674	Mean :38.56	Mean : -121.4
	3rd Qu.:243488	3rd Qu.:38.64	3rd Qu.: -121.4
	Max. :699000	Max. :38.70	Max. : -121.3



```
> sacramento_ev <- sacramento_ev%>%select(-c(city,zip))
> head(sacramento_ev)
  beds baths sqft      type price latitude longitude
1    2     1  836 Residential 59222 38.63191 -121.4349
2    3     1 1167 Residential 68212 38.47890 -121.4310
3    2     1  796 Residential 68880 38.61830 -121.4438
4    2     1  852 Residential 69307 38.61684 -121.4391
5    2     1  797 Residential 81900 38.51947 -121.4358
6    3     1 1122      Condo 89921 38.66260 -121.3278
```






```
> ## Veri Düzenleme
> ## One Hot Encoding
> dummy<-dummyVars(" ~ .", data=sacramento_ev)
> sacramento_ev<-predict(dummy, newdata =sacramento_ev)
> head(sacramento_ev)
```

	beds	baths	sqft	type.Condo	type.Multi_Family	type.Residential	price	latitude	longitude
1	2	1	836	0	0	1	59222	38.63191	-121.4349
2	3	1	1167	0	0	1	68212	38.47890	-121.4310
3	2	1	796	0	0	1	68880	38.61830	-121.4438
4	2	1	852	0	0	1	69307	38.61684	-121.4391
5	2	1	797	0	0	1	81900	38.51947	-121.4358
6	3	1	1122	1	0	0	89921	38.66260	-121.3278

```

> set.seed(1104)
> egitim_index <- createDataPartition(sacramento_ev$price, p = .8,list=FALSE)
> egitim <- sacramento_ev[egitim_index,]
> test <- sacramento_ev[-egitim_index,]
> head(names(getModelInfo())) #model isimleri
[1] "ada"      "AdaBag"    "AdaBoost.M1" "adaboost"
[5] "amdai"    "ANFIS"
> ## Modelleme
> ### CART
> ## Ayarlanabilir parameterler
> modelLookup("rpart")
  model parameter          label   forReg forClass probModel
1 rpart      cp Complexity Parameter  TRUE   TRUE     TRUE

```

```
> set.seed(1104)
> cartFit <- train(price~.,data = egitim, method = "rpart",trControl = fitcontrol,metric="RMSE")
> cartFit #eğitim sürecini ve parametreleri görüntüle
CART
```

```
352 samples
 8 predictor
```

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 316, 316, 318, 317, 316, 317, ...

Resampling results across tuning parameters:

cp	RMSE	Rsquared	MAE
0.003352317	80899.09	0.4249302	58103.94
0.010404115	82328.38	0.3971679	60120.38
0.017726268	82764.89	0.3843382	60954.15

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was cp = 0.003352317.

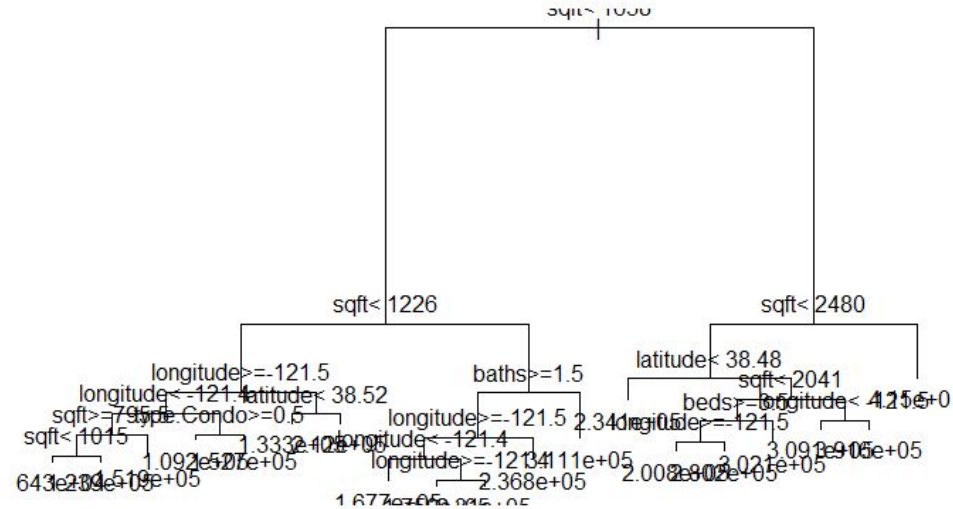



```
> cartFit$finalModel #en iyi sonucu veren model detaylarını görüntüle
n= 352
```

```
node), split, n, deviance, yval
  * denotes terminal node
```

```
1) root 352 3.777928e+12 199489.9
2) sqft< 1657.5 257 1.538379e+12 164074.2
4) sqft< 1225.5 135 4.592364e+11 133100.8
8) longitude>=-121.4808 109 2.707172e+11 122223.0
16) longitude< -121.4279 66 1.561158e+11 110251.2
32) sqft>=795.5 58 9.184129e+10 104501.0
64) sqft< 1015 30 2.699509e+10 86427.9 *
65) sqft>=1015 28 4.454805e+10 123865.0 *
33) sqft< 795.5 8 4.845285e+10 151940.4 *
17) longitude>=-121.4279 43 9.062325e+10 140598.2
34) type.Condo>=0.5 12 2.530097e+10 109249.6 *
35) type.Condo< 0.5 31 4.896453e+10 152733.1 *
9) longitude< -121.4808 26 1.215506e+11 178704.0
18) latitude< 38.52207 11 1.407352e+10 133300.2 *
19) latitude>=38.52207 15 6.817112e+10 212000.1 *
....
```

```
> plot(cartFit$finalModel) #modeli grafikleştir
> text(cartFit$finalModel)
```





```
> ## Bagging
> modelLookup("treebag")
  model parameter  label forReg forClass probModel
1 treebag parameter parameter TRUE  TRUE  TRUE
> baggingFit<- train(price~.,data = egitim, method = "treebag",trControl = fitcontrol,metric="RMSE")
> baggingFit #eğitim sürecini ve parametreleri görüntüle
Bagged CART
```

352 samples
8 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 316, 316, 318, 317, 316, 317, ...
Resampling results:

RMSE	Rsquared	MAE
77385.59	0.4539737	56087.51

```
> baggingFit$finalModel #en iyi sonucu veren model detaylarını görüntüle
```

Bagging regression trees with 25 bootstrap replications

```

> ## Random Forest
> modelLookup("rf")
  model parameter          label forReg forClass probModel
1   rf      mtry #Randomly Selected Predictors  TRUE    TRUE    TRUE
> set.seed(1104)
> rfFit<- train(price~.,data = egitim, method = "rf",trControl = fitcontrol,metric="RMSE")
> rfFit #eğitim sürecini ve parametreleri görüntüle
Random Forest
352 samples
 8 predictor
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 316, 316, 318, 317, 316, 317, ...
Resampling results across tuning parameters:
  mtry RMSE      Rsquared  MAE
  2   74756.23 0.4915653 53132.30
  5   75103.18 0.4852230 53249.98
  6   75650.31 0.4797569 53752.09

```

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 2.



```
> rffit$finalModel #en iyi sonucu veren model detaylarını görüntüle
```

Call:

```
randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)))
```

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 2

Mean of squared residuals: 5786647335

% Var explained: 46.08

```

> ## Gradient Boosting
> modelLookup("gbm")
  model      parameter      label forReg forClass probModel
1  gbm      n.trees  # Boosting Iterations  TRUE   TRUE   TRUE
2  gbm interaction.depth      Max Tree Depth  TRUE   TRUE   TRUE
3  gbm      shrinkage      Shrinkage  TRUE   TRUE   TRUE
4  gbm  n.minobsinnode Min. Terminal Node Size  TRUE   TRUE   TRUE
> set.seed(1104)
> gbmFit<- train(price~.,data = egitim, method = "gbm",trControl = fitcontrol,metric="RMSE")
> gbmFit #eğitim sürecini ve parametreleri görüntüle
Stochastic Gradient Boosting
352 samples
 8 predictor
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 316, 316, 318, 317, 316, 317, ...

```



Resampling results across tuning parameters:

shrinkage	interaction.depth	n.minobsinnode	n.trees	RMSE	Rsquared	MAE
0.1230857	10	9	3696	83526.38	0.4214183	59989.04
0.2513721	5	19	1533	87331.69	0.3801141	63408.19
0.3603457	5	12	1846	91098.68	0.3616729	66140.37

RMSE was used to select the optimal model using the smallest value.

The final values used for the model were n.trees = 3696, interaction.depth = 10, shrinkage = 0.1230857 and n.minobsinnode = 9.

```
> gbmFit$finalModel #en iyi sonucu veren model detaylarını görüntüle
```

A gradient boosted model with gaussian loss function.

3696 iterations were performed.

There were 8 predictors of which 7 had non-zero influence.



```
> ##XGBoost
```

```
> modelLookup("xgbTree")
```

	model	parameter	label	forReg	forClass	probModel
1	xgbTree	nrounds	# Boosting Iterations	TRUE	TRUE	TRUE
2	xgbTree	max_depth	Max Tree Depth	TRUE	TRUE	TRUE
3	xgbTree	eta	Shrinkage	TRUE	TRUE	TRUE
4	xgbTree	gamma	Minimum Loss Reduction		TRUE	TRUE TRUE
5	xgbTree	colsample_bytree	Subsample Ratio of Columns	TRUE	TRUE	TRUE
6	xgbTree	min_child_weight	Minimum Sum of Instance Weight	TRUE	TRUE	TRUE
7	xgbTree	subsample	Subsample Percentage	TRUE	TRUE	TRUE

```
> set.seed(1104)
```

```
> xgbFit<- train(price~.,data = egitim, method = "xgbTree",trControl = fitcontrol,metric="RMSE")
```



```
> xgbFit #eğitim sürecini ve parametreleri görüntüle
eXtreme Gradient Boosting
```

```
352 samples
8 predictor
```

```
No pre-processing
```

```
Resampling: Cross-Validated (10 fold, repeated 5 times)
```

```
Summary of sample sizes: 316, 316, 318, 317, 316, 317, ...
```

```
Resampling results across tuning parameters:
```


eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	RMSE	Rsquared	MAE
0.1230857	10	4.370809	0.5397388	7	0.8436265	322	80400.73	0.4416156	57834.20
0.2513721	5	8.696266	0.4239902	17	0.9431193	630	84642.10	0.4090298	62020.62
0.3603457	5	4.711066	0.5085853	16	0.7009257	645	87791.98	0.3811346	64208.84

```
RMSE was used to select the optimal model using the smallest value.
```

```
The final values used for the model were nrounds = 322, max_depth = 10, eta = 0.1230857, gamma = 4.370809,
```

```
colsample_bytree
```

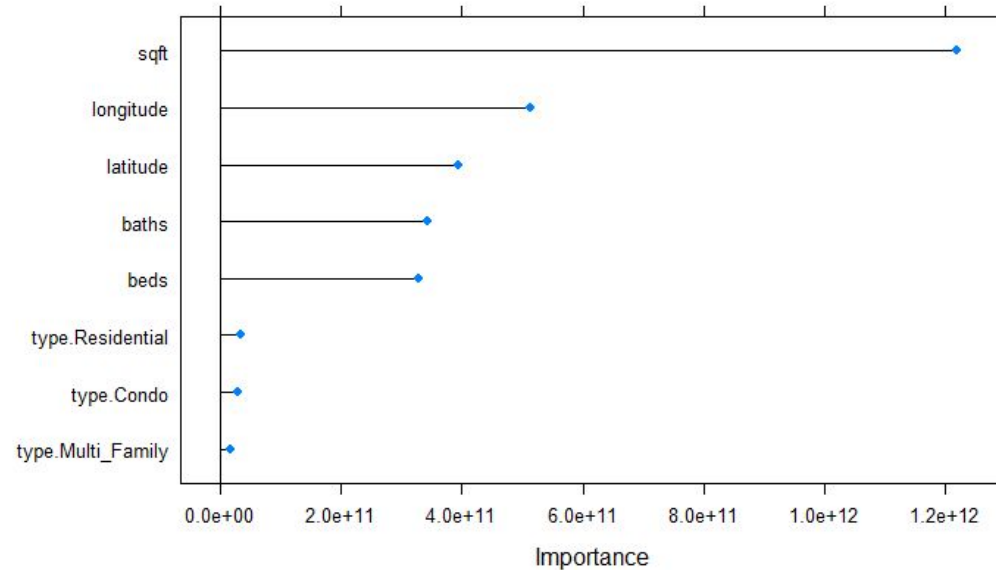
```
= 0.5397388, min_child_weight = 7 and subsample = 0.8436265.
```



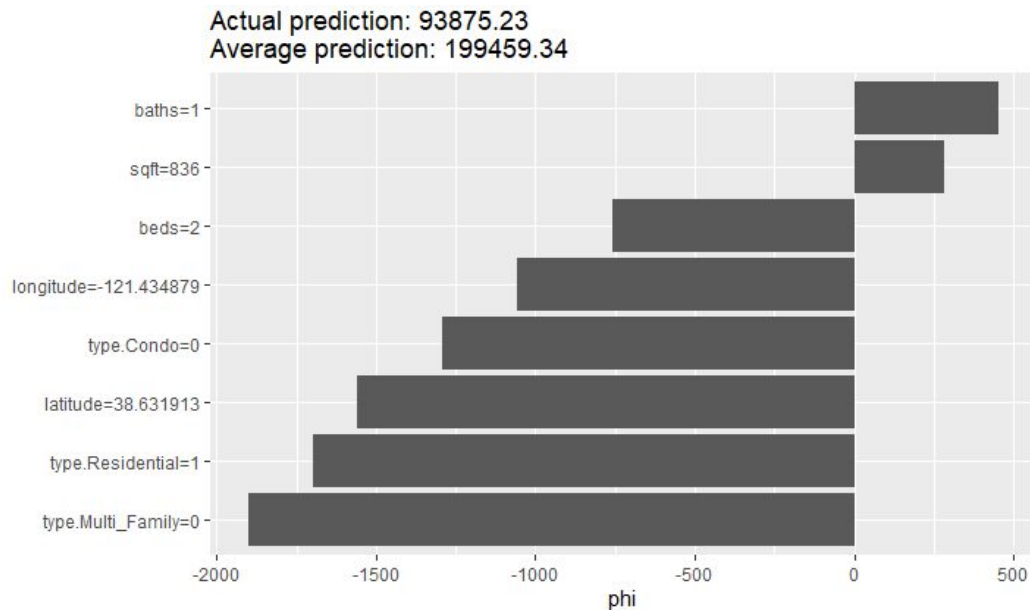
```
> ##Performans Karşılaştırması
> cart_tahmin <- cartFit %>% predict(test)
> bag_tahmin <- baggingFit %>% predict(test)
> rf_tahmin <- rffit %>% predict(test)
> gbm_tahmin <- gbmFit %>% predict(test)
> xgb_tahmin <- xgbFit %>% predict(test)
```

```
> # RMSE Hesaplama
> RMSE(cart_tahmin, test$price)
[1] 87279.74
> RMSE(bag_tahmin, test$price)
[1] 77399.99
> RMSE(rf_tahmin, test$price) # en iyi sonuç
[1] 68477.05
> RMSE(gbm_tahmin, test$price)
[1] 86359.93
> RMSE(xgb_tahmin, test$price)
[1] 82142.13
```

> ## Değişken Önem Grafiği
> plot(varImp(rfFit, scale = FALSE))



```
> library(iml)
> predictor = Predictor$new(rfFit, data = egitim[, -7], y = egitim$price)
> shapley = Shapley$new(predictor, x.interest = egitim[, -7])
> plot(shapley)
```





Referanslar

- Applied Predictive Modeling, M. Kuhn, K. Johnson., Springer, 2013
- An Introduction to Statistical Learning with Applications in R, G. James,D. Witten,T.Hastie,R.Tibshirani, Springer, 2013.
- İlker Birbil, Karar Ağaçları.
- Ceylan Yozgatlıgil, Ders Notları
- Ozancan Ozdemir, Ders Notları



TEŞEKKÜRLER!