# DSM5007-Denetimli İstatistiksel Öğrenme-Final

Melih TAŞKIN - 2023900106

2024-01-11

# REGRESYON

2052 isimli(numaralı) satırda `Height` değeri 1.3 girilmiş, yanlışlık olduğunu düşündüğüm için 0.13 olarak değiştirdim. 1258 ve 3997 numaralı satırlardaki `Height` değişkeninin 0 olamayacağını düşündüğüm için bu satırları çıkarmayı tercih ediyorum. `Sex` değişkeni için `I` olmayanların yani `F` ve `M` dağılımlarının `Rings` değişkenine göre dağılımlarının benzer olması ve doğrusal regresyon modelini kurarken `I` anlamlı çıkarken diğerlerinin anlamsız çıkması üzerine `Sex` değişkeninin içeriğini `I` ve `NI` olarak düzenledim. Bu düzenlemeyi yaparken modellerin performansları üzerine incelemeler gerçekleştirdim ve doğrusal regresyon modelimde ciddi bir artış gözlemlerken diğer modellerimde değişim görmedim diyebilirim fakat gözümden kaçmış olabilecek $10^{-3}$. terimde gerçekleşmiş olabilir(R-squared için). Bu işlemleri hiyerarşik olarak yapmadığım için karşılaştırma durumları raporda ve kod dosyaları arasında bulunmamakta.

```
abalone <- read.csv("C:/Users/melih/Documents/DSM5007-Final/abalone_veriseti.data")
abalone$Sex <- ifelse(abalone$Sex == "I", abalone$Sex, "NI")
abalone$Sex <- factor(abalone$Sex, levels = c("NI","I"))
abalone$Rings <- as.integer(abalone$Rings)
abalone$Height[2052] = 0.130
abalone <- abalone[-c(1258,3997), ]
```

## 1-Doğrusal regresyon (LM)

- $H_0$ : Katsayı sıfırdır.
- $H_1$ : Katsayı sıfırdan farklıdır.

```
lm_model <- lm(Rings~.+I(Length^2), data = train_abalone)
summary(lm_model)
```

```
##
## Call:
## lm(formula = Rings ~ . + I(Length^2), data = train_abalone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5654 -1.3064 -0.3286  0.8950 11.8817
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.2171     0.5570  -0.390  0.69681
## SexI           -0.7341     0.1063  -6.906 6.10e-12 ***
## Length         21.8256     3.5383   6.168 7.85e-10 ***
## Diameter        7.8269     2.6449   2.959  0.00311 **
## Height         13.9186     2.6101   5.333 1.04e-07 ***
## WholeWeight     8.7009     0.8614  10.101  < 2e-16 ***
```

```
## ShuckedWeight -17.2408     0.9744 -17.694  < 2e-16 ***
## VisceraWeight  -6.8713     1.5436  -4.451 8.85e-06 ***
## ShellWeight    10.5604     1.3603   7.763 1.14e-14 ***
## I(Length^2)   -29.3996     3.4422  -8.541  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.16 on 2912 degrees of freedom
## Multiple R-squared:  0.5462, Adjusted R-squared:  0.5448
## F-statistic: 389.4 on 9 and 2912 DF,  p-value: < 2.2e-16
```

p-değeri, önem düzeyi 0.05'ten küçük olan değişkenler için, $H_0$ hipotezi reddedilir, 0.05'ten büyük olan değişkenler için $H_0$ hipotezini reddedemeyiz.

Oluşturduğum modelde anlamsız katsayılara sahip değişkenler bulunmamaktadır.

**Varsayım 1: Lineerlik**

Korelasyon katsayısının 0 olduğu durumlar olmadığı için bu varsayım sağlanmaktadır. `Rmd` dosyasında uygun kod satırı mevcuttur.

**Varsayım 2: Hataların Normal Dağılımı**

- $H_0$ : Hatalar normal dağılmaktadır.
- $H_1$ : Hatalar normal dağılmamaktadır.

```
##
##  Shapiro-Wilk normality test
##
## data:  lm_model$residuals
## W = 0.93077, p-value < 2.2e-16
```

Shapiro-Wilk testi sonucunda p-değeri 0.05'ten küçük çıktığı için $H_0$ hipotezimizi reddederiz. Dolayısıyla `lm_model` modelinin hataları normal dağılmamaktadır.

**Varsayım 3: Homojen Varyans**

- $H_0$ : Hatalar homojen varyansa sahiptir.
- $H_1$ : Hatalar homojen varyansa sahip değildir.

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_model
## BP = 295.86, df = 9, p-value < 2.2e-16
```

Breusch-Pagan testi sonucunda p-değeri, $a = 0.05$ anlamlılık düzeyinden küçük olduğu için, $H_0$ hipotezi reddedilir. Dolayısıyla `lm_model` modeli için homojen varyans varsayımı ihlal edilmiştir.

Doğrusal modelimiz için gerekli varsayımlar sağlanmamaktadır. Bu varsayımları sağlanması için gerekli dönüşümler ve uygulamalar yapılabilir fakat, hataların normal dağılımı testi ve varyans testi çok küçük değer çıkmış olduğu için bu varsayımları sağlarken veri kaybı yaşayacağımı düşünüyorum. Bu sebepten dolayı tahmin performansının düşük olmasını göze alarak, diğer modelleri kurup tahmin performanslarını karşılaştırma sonucunda aradaki farkın az olması durumunda bu geliştirmeleri yapmayı düşünüyordum, hataların normal dağıldığı ve homojen varyanslılık varsayımlarını kanıtlayamadığım için, değişkene anlamlılık kazandırması amacıyla `Length` değişkeninin karesini aldım ve bunu yaparken oluşturduğum diğer doğrusal modellerin performanslarıyla karşılaştırdım. Şu haliyle en iyi performansı göstermektedir.

```
##    Dataset     RMSE      MAE R_squared
## 1:   Train 2.156008 1.562266  0.546176
```

```
## 2:    Test 2.160891 1.545440  0.565309
```

Şu haliyle doğrusal modelimizin eğitim verisine aşırı uyumu gözlenmemektedir.

## 2-Regresyon Ağacı (RT)

Regresyon Ağacı oluştururken `rpart` kütüphanesinden yardım alma sebebim, özet çıktısını tek bir fonksiyon yardımıyla vermesi ve oluşturulan ağaç hakkında daha detaylı bilgileri bize sunabiliyor olmasından dolayı. Bu işlem sonucunda herhangi bir budama veya iyileştirme yapılmadığı haliyle `tree` fonksiyonuyla aynı sonucu vermektedir.

Bu özet çıktıda CP değerleri, değişkenlerin önemi ve düğümler hakkında bilgiler mevcuttur. Rapor içerisinde özet bilgiye yer vermedim, yorumlamamı grafik üzerinden yapmayı uygun buldum.

```
rt_model <- rpart(Rings ~ ., data = train_abalone)
```

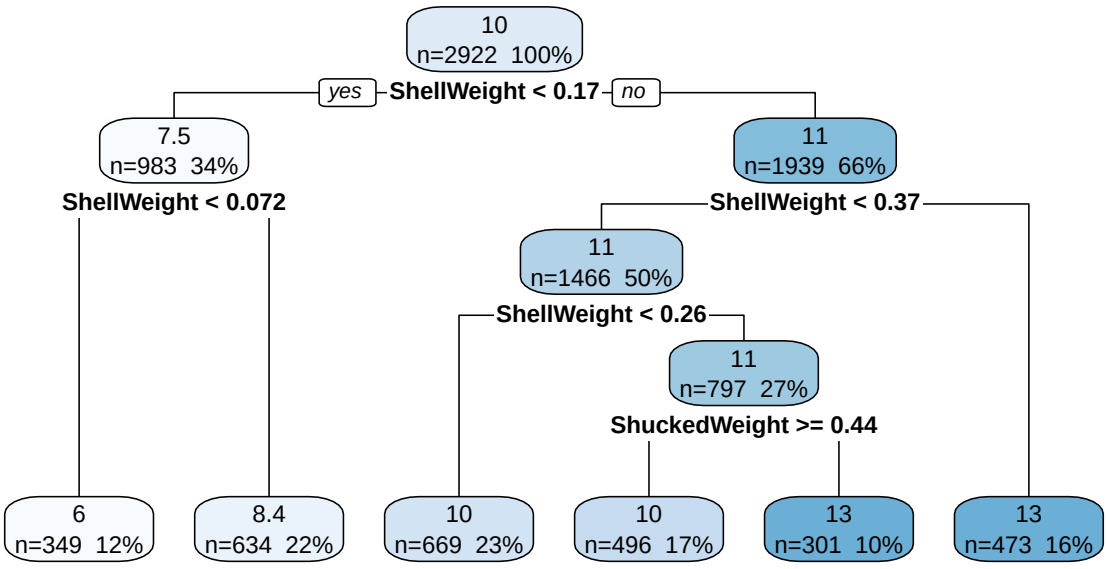
Bir diğer önemli mevzu olan budama işlemlerini yaptım ve test verisi üzerinde tahminleme performansında düşüş yani `R-squared` değerinde düşüş ve `MSE`, `MAE`, `RMSE` değerlerinde artış gözlemledim. Aşağıdaki tablo içerisindeki metrikler budanmamış regresyon ağacı üzerinden hesaplanmıştır.

```
##    Dataset     RMSE      MAE R_squared
## 1:   Train 2.321383 1.699307 0.4738857
## 2:    Test 2.451416 1.776283 0.4405655
```

Eğitim veri seti ve test veri seti üzerinden modelin metriklerine baktığımda, R-Squared değerleri bağımlı değişkenin varyansının yarısını dahi açıklayamamış olması, çok iç açıcı bir durum değil fakat amacımız diğer modellerle karşılaştırmak olduğu için burada sadece aşırı uyma durumu gözükmemekte diyebilirim.

Regresyon ağacını açıklamayı uzatmamak ve kolaylık olması için, budama işlemi yapıyorum ve `rt_model` özetindeki cp değerlerini inceledikten sonra budamak için cp değerini 0.02'den seçtim.

```
pruned_rt_model <- prune(rt_model, cp = 0.02)
rpart.plot(pruned_rt_model, type = 2, extra = 101)
```




- Kurutulmuş deniz kabuğu ağırlığı 0.17 gramdan düşükken halka sayısının 7.5 olması beklenir. 0.17 gramdan büyükse halka sayısının 11 olması beklenir.

- Kurutulmuş kabuk ağırlığı 0.072 gramdan küçükken halka sayısının 6 olması beklenirken, 0.072 ile 0.17 gram arasındaysa halka sayısının 8.4 adet olması beklenir.
- 0.17 ile 0.26 gram arasında kabuk ağırlığına sahip olunması durumunda halka sayısı 10 adet beklenir.
- Yine kabuk ağırlığı 0.26 ile 0.37 aralığında olması durumunda halka sayısı 11 beklenirken, eğer bu aralıktaki örneklerin;
  - Et ağırlığı 0.44 gramdan büyük ya da eşitse halka sayısı 10 adet beklenir.
  - Et ağırlığı 0.44 gramdan küçükse halka sayısının 13 olması beklenir.
- Kurutulmuş kabuk ağırlığının 0.37 gramdan büyük olması durumundaysa halka sayısının 13 olması beklenir.

## 3-Bagging ile regresyon ağacı (BRT)

```
set.seed(106)
brt_model <- randomForest(formula = Rings ~ .,
                          data = train_abalone,
                          ntree = 50,
                          mtry = 8, importance = TRUE)
```

Bagging modeli oluştururken, random foresttan ayıran en önemli özellik, `mtry` değeri bağımsız değişken sayısının tamamıdır. `ntree` değeriyle ağaç sayısını 50 aldım.

```
brt_model
```

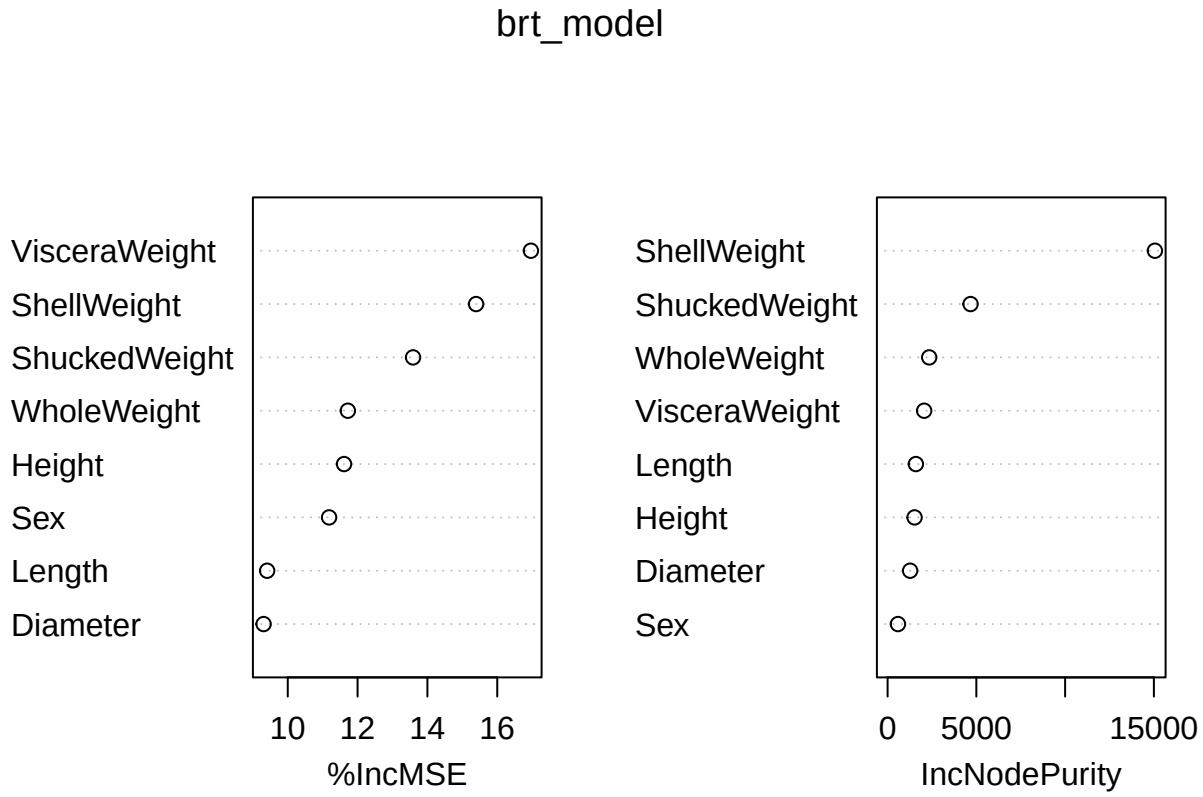
```
##
## Call:
##  randomForest(formula = Rings ~ ., data = train_abalone, ntree = 50,      mtry = 8, importance = TRU
##                Type of random forest: regression
##                      Number of trees: 50
## No. of variables tried at each split: 8
##
##           Mean of squared residuals: 4.946333
##                     % Var explained: 51.71
```

Oluşturduğum `brt_model` için, ağaç sayısı 50, ve mtry sayısı 8'dir. Hataların karelerinin ortalama değeri 4.695049 alırken bu modelin bağımlı değişkenin varyansının %51.71'i model tarafından açıklandığını gösterir.

```
##    Dataset      RMSE       MAE R_squared
## 1:   Train 0.9577586 0.6592763 0.9104432
## 2:    Test 2.2316026 1.5540399 0.5363943
```

Aşırı uyma durumunu kontrol etmek isterken ağaç sayısına göre gözlemledim, fakat açıklanabilen varyansta ve test r-squaredde karşılaşılan değişimlerden dolayı ağaç sayısını 50 olarak tercih ettim. Şu haliyle, aşırı uyum varlığından söz edilebilir. Amacımız hangi modelin daha başarılı olduğunu bulmak olduğu için, bu durumu şimdilik gözardı ediyorum.

```
varImpPlot(brt_model)
```




`ShellWeight` değişkeni, her iki metrikte de yüksek skorları elde etmiş. Bu, `ShellWeight` değişkenin modelde önemli olduğunu ve tahminlerin çoğunu etkilediğini gösteriyor.

`ShuckedWeight` değişkeni, özellikle IncNodePurity(Düğüm Saflığı Artışı) metriğinde `ShellWeight` değişkeninin hemen ardından geliyor. Bu,`ShuckedWeight` değişkenin modelin ağaçlarında düğümleri daha etkili bir şekilde böldüğünü ve sınıfları iyi ayırdığını gösteriyor.

Diğer değişkenler de modelde belirli bir öneme sahiptir, ancak `ShellWeight` ve `ShuckedWeight` bu ölçümlerde daha belirgin bir etkiye sahiptir.

En fazla etkili olan 4 değer ikisi için de aynıdır ve bunlar; `ShuckedWeight`, `ShellWeight`, `VisceraWeight`, `WholeWeight`.

## 4-Rassal Ormanlar Regresyonu (RFR)

```
set.seed(106)
rfr_model <- randomForest(formula = Rings ~ .,
                          data = train_abalone,
                          mtry = 3, #p/3
                          importance = TRUE)
```

```
rfr_model
```

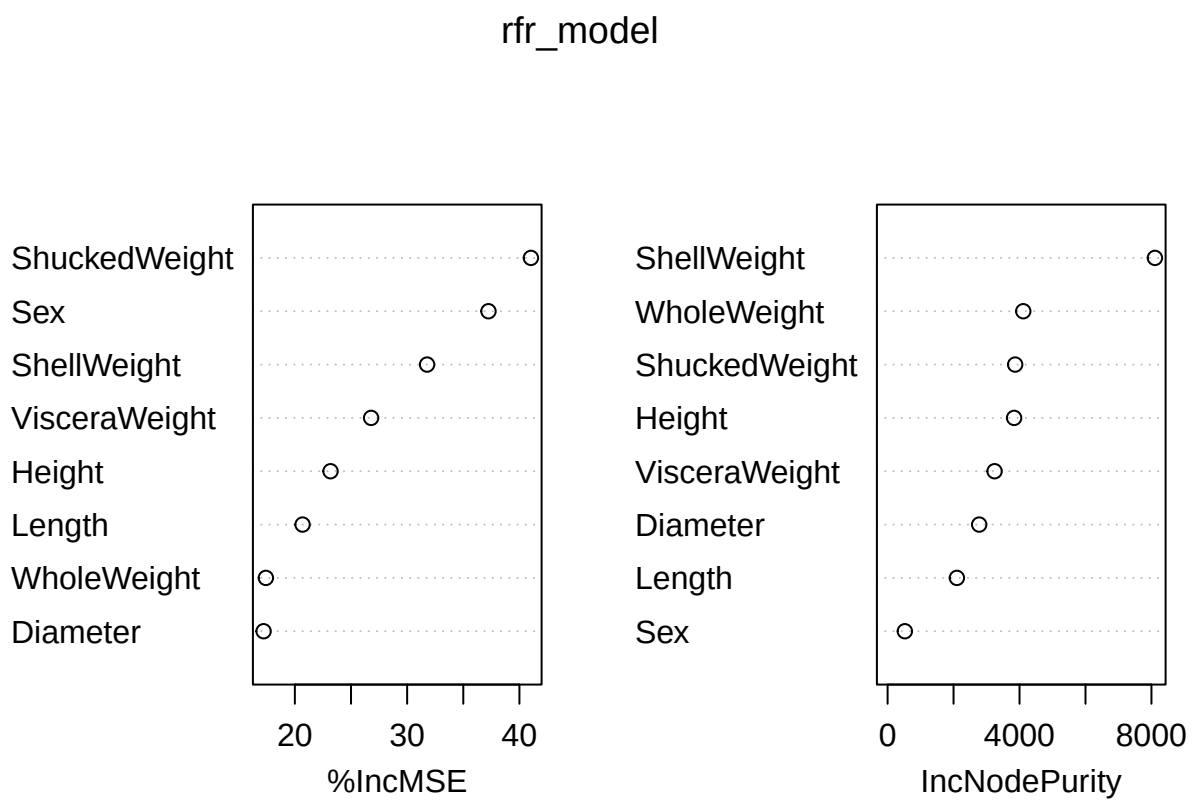
```
##
## Call:
##  randomForest(formula = Rings ~ ., data = train_abalone, mtry = 3,      importance = TRUE)
##                Type of random forest: regression
```

```
##                       Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 4.554569
##                     % Var explained: 55.53
```

Değişken sayısı/3 işlemi sonucunda elde ettiğim en yakın tam sayı değeri 3 olduğu için, mtry değerini 3'e eşitledim.

Hataların karelerinin ortalama değeri 4.572395 alırken bu modellin bağımlı değişkenin varyansının %55.53'ü model tarafından açıklandığını gösterir.

```
varImpPlot(rfr_model)
```




`ShellWeight` değişkeni, her iki metrikte de yüksek skorlara sahip olduğu için modelde en önemlilerden olduğu görülmekte ve tahminlerin çoğunu etkilemektedir.

`ShuckedWeight` ve `ShellWeight` değişkenleri, diğer değişkenlere göre modelde ve tahminlerde daha önemli bir yere sahiptir.

`Sex` değişkeni düğümde kullanılma durumu en düşük olmasına rağmen, tahminlerde önemli bir etkiye sahiptir.

```
##    Dataset      RMSE       MAE R_squared
## 1:   Train 0.9711509 0.6779606 0.9079211
## 2:    Test 2.1731176 1.5154498 0.5603759
```

Bu sonuca göre modelin eğitim setine aşırı uyma durumu olduğunu söyleyebileceğimi düşünüyorum.

## 5-Karşılaştırma

```
set.seed(106)
# Test seti üzerinde tahminler yap
lm_tahminler <- predict(lm_model, newdata = test_abalone)
rt_tahminler <- predict(rt_model, newdata = test_abalone)
brt_tahminler <- predict(brt_model, newdata = test_abalone)
rfr_tahminler <- predict(rfr_model, newdata = test_abalone)
```

```
performans_tablosu <- data.frame(Method = c("Doğrusal Regresyon", "Regresyon Ağacı",
                                            "Bagging ile Regresyon Ağacı", "Rassal Ormanlar"),
                                 R_squared = c(lm_rsq, rt_rsq, brt_rsq, rfr_rsq),
                                 MSE = c(lm_mse, rt_mse, brt_mse, rfr_mse),
                                 MAE = c(lm_mae, rt_mae, brt_mae, rfr_mae),
                                 RMSE = c(lm_rmse, rt_rmse, brt_rmse, rfr_rmse))
sira_indeks <- order(performans_tablosu$R_squared, decreasing = TRUE)
performans_tablosu <- performans_tablosu[sira_indeks, ]
print(performans_tablosu)
```

```
##                         Method R_squared      MSE      MAE     RMSE
## 1           Doğrusal Regresyon 0.5653090 4.669448 1.545440 2.160891
## 4              Rassal Ormanlar 0.5603759 4.722440 1.515450 2.173118
## 3 Bagging ile Regresyon Ağacı 0.5363943 4.980050 1.554040 2.231603
## 2              Regresyon Ağacı 0.4405655 6.009442 1.776283 2.451416
```

Tabloyu açıklamadan önce veri üzerinde ve modeller üzerinde bir çok deneme yaptığımı ve en basit ve uygun gördüğüm haliyle modelleri oluşturduğumu belirtmek isterim. Ayrıca her model için aşırı uyma durumu kontrollerini modelleri oluşturduktan sonra inceledim. Test seti üzerinden R-squared değeri eğitime göre yüksek olan sadece doğrusal regresyon modeli var. Yazdığım yorumlar ve karşılaştırmalar sadece bu veri seti özelindedir.

R-squared değerlerine baktığımızda doğrusal regresyon modelimizin varsayımları sağlamıyor olmasına rağmen test verisi üzerinde başarılı tahminlerde bulunması ve bunu eğitim setine göre daha başarılı yapmış olması, doğrusal regresyon modelini diğerlerine göre bir adım ön plana çıkarıyor. Rassal ormanlar modeli eğitim esnasında iyi bir öğrenme değerine sahip fakat test verisi üzerindeki tahminlemelerinde doğrusal regresyon modelimizden bir tık geride kalmış.

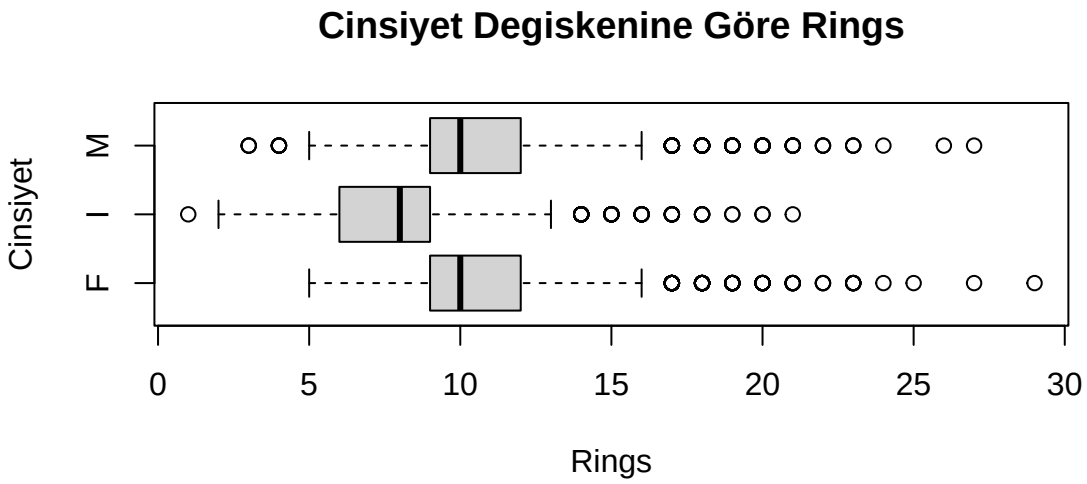
Regresyon ağacı modeliyse budama yapılmamasına rağmen eğitim seti üzerinden dahi `Rings`'in varyansını açıklamakta başarılı olduğunu söyleyemiyorum, test seti üzerindeki tahminlemeleri de bariz bir şekilde diğerlerine göre en zayıf performansı gösteren model.

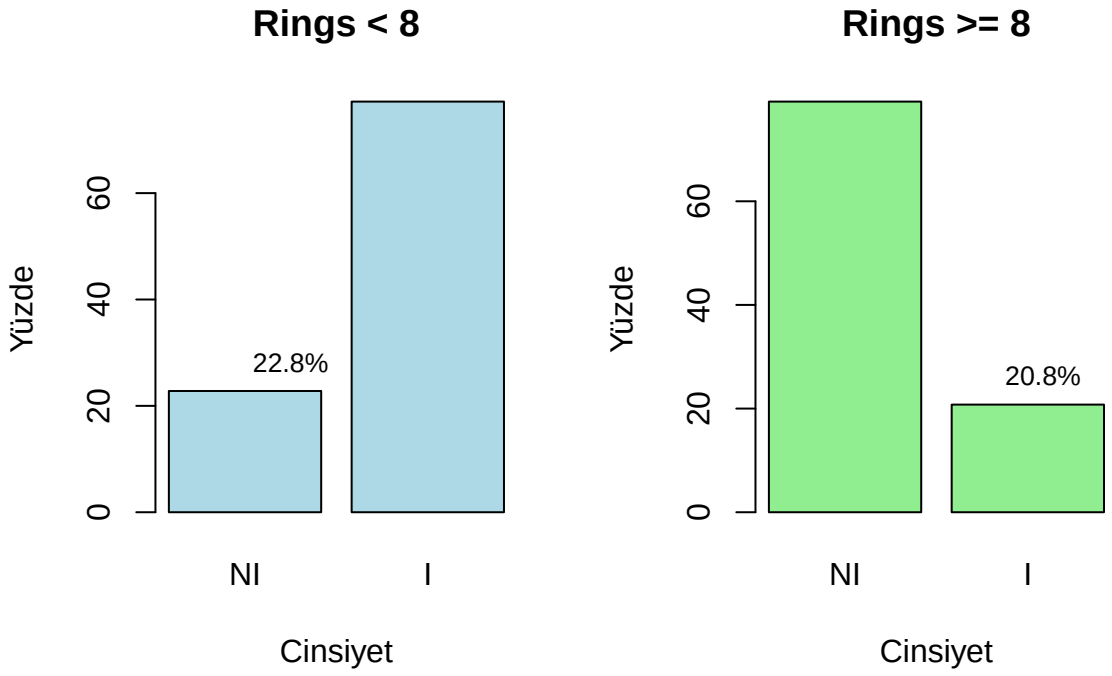
Doğrusal regresyon, bagging ve rassal ormanlar modellerinde iyileştirmeler ve geliştirmelerin mümkün olduğunu fakat denediğim bir çok yöntemde rassal ormanlar modelinin test verisi üzerinden tahminleri sonucu R-squared değerinde 0.54 ile 0.565 arasında değiştiğini, doğrusal regresyon modelininse 0.52'lerden şuanki seviyesine getirebildiğimi söylemek isterim.

Yazdığım durumlarla birlikte bir model üzerinden geliştirmek ve düzenlemeye devam etmek istesem doğrusal regresyon modeli üzerinden geliştirmelerime devam ederim çünkü, eğitim verisinde yüksek öğrenme değerine sahip olmamasına rağmen test setinde yani gerçek hayattaki verileri tahminlemede daha başarılı olduğunu göstermiştir.

# SINIFLANDIRMA

```
abaloneClass <- read.csv("C:/Users/melih/Documents/DSM5007-Final/abalone_veriseti.data")
abaloneClass$Rings <- as.integer(abaloneClass$Rings)
abaloneClass$Height[2052] = 0.130
abaloneClass <- abaloneClass[-c(1258,3997), ]
abaloneClass <- na.omit(abaloneClass)
```




Cinsiyet değişkenine göre halka sayılarının dağılımına kutu grafiğine baktığımda; `Infant` özelliği baz alındığında en uygun eşikdeğerinin `Infant` olmayanların 1. çeyrekliğine ve `Infant` olanlarınsa 3. çeyrekliğine denk gelen 9 değerini uygun gördüm. Bu değere göre sınıflandırma yaptım ve etiketlendiği sınıfa ait olmayan-

ların yüzdelik dilimlerine ve sayılarına kutu ve sütun grafiğiyle baktım. Sütun grafiklerine ve elde ettiğim yüzdelik dilimlerine göre eşik değerimi en uygun gördüğüm 8 olarak belirledim.

Yani şu haliyle sınıflandırmamı; eğer halka sayısı 8'den azsa deniz salyangozumuz henüz bebekken, 8 adet veya daha fazla halkası bulunan deniz salyangozumuz bebek değildir.

Ayrıca sınıflandırmayı bebek ya da bebek değil olarak yapacağımız için veri setimden `Sex` değişkenini çıkardım.

```
set.seed(106)
INI <- ifelse(abaloneClass$Rings<8,"Infant","Not Infant")
abaloneClass <- data.frame(abaloneClass,INI)
abaloneClass$INI <- as.factor(abaloneClass$INI)
train_index <- sample(1:nrow(abaloneClass), 0.7 * nrow(abaloneClass))
abaloneClass <- abaloneClass[,-1]
train_abaloneClass <- abaloneClass[train_index, ]
test_abaloneClass <- abaloneClass[-train_index, ]
str(train_abaloneClass)
```

```
## 'data.frame':    2922 obs. of  9 variables:
##  $ Length       : num  0.595 0.46 0.59 0.515 0.55 0.645 0.645 0.495 0.565 0.395 ...
##  $ Diameter     : num  0.465 0.36 0.49 0.39 0.43 0.5 0.51 0.37 0.44 0.305 ...
##  $ Height       : num  0.145 0.125 0.135 0.12 0.15 0.175 0.165 0.125 0.135 0.105 ...
##  $ WholeWeight  : num  1.107 0.547 1.008 0.613 0.84 ...
##  $ ShuckedWeight: num  0.402 0.216 0.422 0.302 0.395 ...
##  $ VisceraWeight: num  0.241 0.11 0.224 0.137 0.195 ...
##  $ ShellWeight  : num  0.31 0.19 0.285 0.141 0.223 ...
##  $ Rings        : int  12 8 11 8 8 11 11 18 9 8 ...
##  $ INI          : Factor w/ 2 levels "Infant","Not Infant": 2 2 2 2 2 2 2 2 2 2 ...
```

## 6-Sınıflandırma Ağacı(CT)

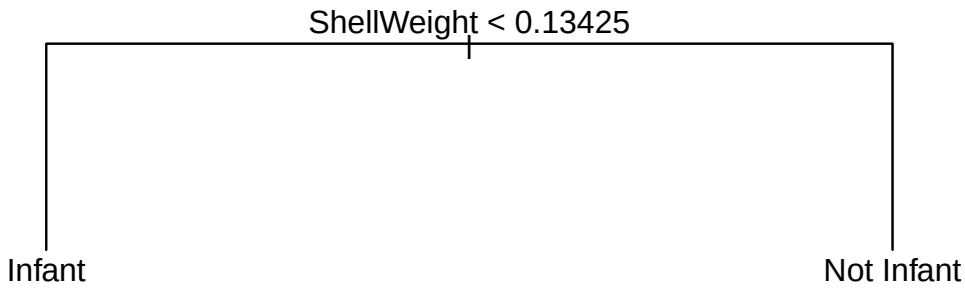
Budama sonrası `accuracy` değerinde yaklaşık 0.006 artış gördüm.

```
set.seed(106)
tree.train_abaloneClass <- tree(INI~.-Rings , train_abaloneClass)
ct_model <- prune.misclass(tree.train_abaloneClass,best=2)
summary(ct_model)
```

```
##
## Classification tree:
## snip.tree(tree = tree.train_abaloneClass, nodes = 3:2)
## Variables actually used in tree construction:
## [1] "ShellWeight"
## Number of terminal nodes:  2
## Residual mean deviance:  0.5813 = 1697 / 2920
## Misclassification error rate: 0.1164 = 340 / 2922
```

`ct_model` yani pruned classification tree modelim 2 terminal node ile `ShellWeight` değişkeni üzerinden verinin %58'ini açıklama başarısını yaklaşık %11 hata oranıyla elde ettiğini göstermektedir.

```
plot(ct_model)
text(ct_model ,pretty =0)
```




Ağacın yapısına baktığımızda açıkça kabuk ağırlığı 0.13425 gramdan küçükse bebektir, 0.13425 grama eşit ya da büyükse bebek değildir olarak sınıflandırmıştır.

```
set.seed(106)
tree.pred <- predict(ct_model ,train_abaloneClass , type="class")
ct_conf_matrix <- table(tree.pred ,train_abaloneClass$INI)
ct_conf_matrix
```

```
##
## tree.pred    Infant Not Infant
##   Infant        491        250
##   Not Infant     90       2091
```

Eğitim verileri üzerinden tahminlerine baktığımızda `Infant` sınıfından olması gereken bireylerin 90 tanesini, `Not Infant` sınıfında olması gereken bireylerdense 250 tanesini yanlış tahmin ettiği gözükmektedir.

```
sum(diag(ct_conf_matrix))/sum(ct_conf_matrix)
```

```
## [1] 0.8836413
```

Doğruluk değerine baktığımızda, `ct_model`inin eğitim veri setindeki gözlemlerin yaklaşık %88'inin sınıfını doğru tahmin ettiğini görüyoruz.

## 7-Bagging ile Sınıflandırma Ağacı (BCT)

```
set.seed(106)

bct_model <- randomForest(INI~.-Rings,
                    data=train_abaloneClass,
                    mtry=7,
                    importance=TRUE)
```

```
yhat.bag <- predict(bct_model,newdata=train_abaloneClass)
bct_conf_matrix <- table(yhat.bag ,train_abaloneClass$INI)
bct_conf_matrix
```

```
##
```

```
## yhat.bag     Infant Not Infant
##   Infant        581          0
##   Not Infant      0       2341
```

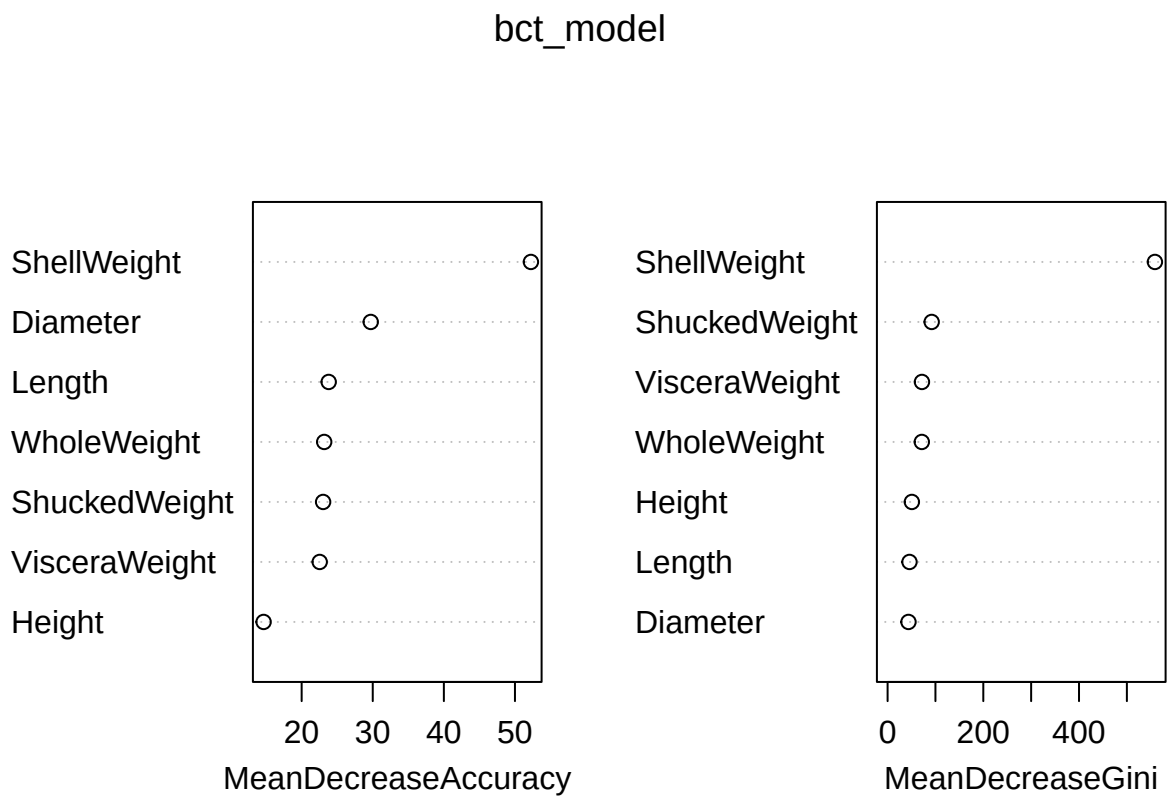
bct_modelimin eğtim verisi üzerinde hatasız tahminleme yaptığı gözükmektedir.

```
sum(diag(bct_conf_matrix))/sum(bct_conf_matrix)
```

```
## [1] 1
```

bct_modelinin eğitim seti üzerinden doğruluk oranı %100'dür.

```
varImpPlot(bct_model)
```




`MeanDecreaseAccuracy` değerlerine göre `ShellWeight` değişkeni sınıflandırmanın doğruluğuna çok büyük katkı yaparken `Diameter` değişkeni onun hemen ardından diğer değişkenlere göre daha fazla fakat çok büyük diyemeyeceğimiz katkısının olduğunu söyleyebiliriz.

`MeanDecreaseGini` yani ağacın saflığına olan katkılara baktığımızda `ShellWeight` değişkeni çok büyük bir katkıda bulunmuş ve diğer değişkenlere göre açık bir fark oluşturduğu gözükmekte.

## 8-Rassal Ormanlarla Sınıflandırma(RFC)

```
set.seed(106)
rfc_model <- randomForest(INI~.-Rings,
                          data=train_abaloneClass,
                          mtry=2,
                          importance=TRUE)
```

```
yhat.rfc <- predict(rfc_model,newdata=train_abaloneClass)
rfc_conf_matrix <- table(yhat.rfc ,train_abaloneClass$INI)
rfc_conf_matrix
```

```
##
## yhat.rfc     Infant Not Infant
##   Infant        581          0
##   Not Infant      0       2341
```

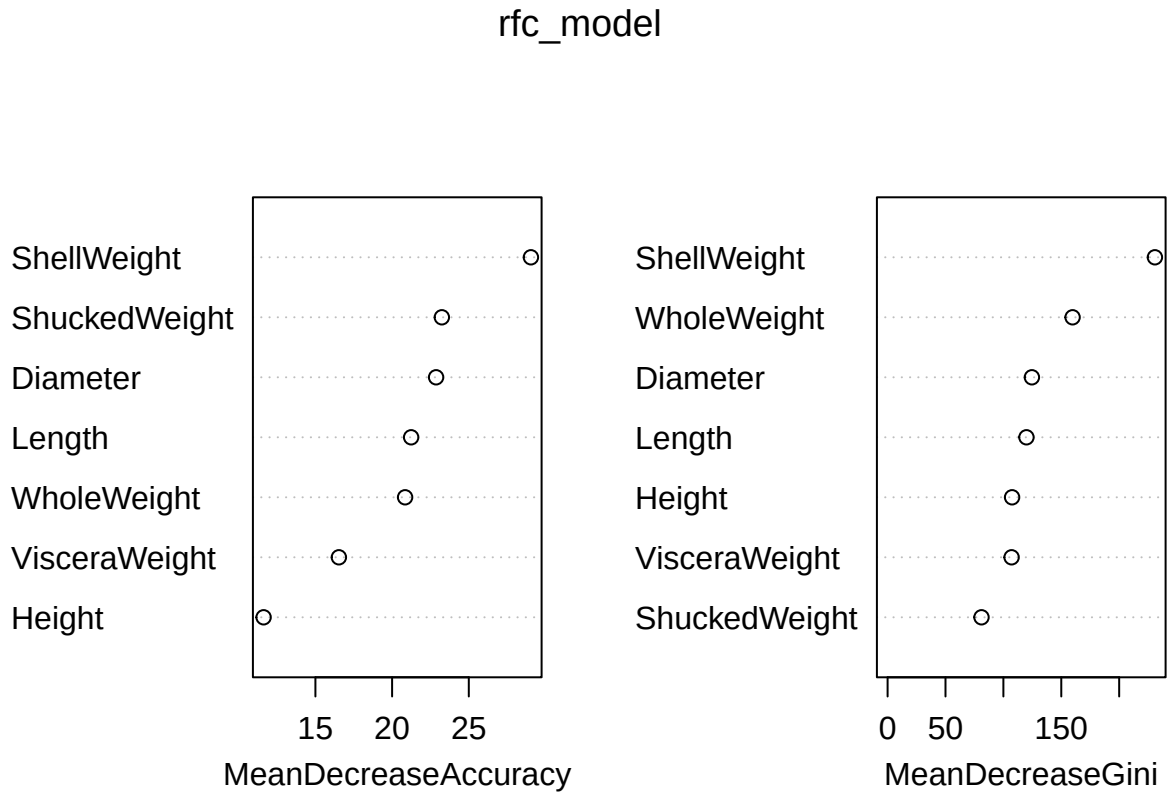
Rassal ormanlarla sınıflandırma modelim eğitim verisi üzerinde tahminlemeleri hatasızdır.

```
sum(diag(rfc_conf_matrix))/sum(rfc_conf_matrix)
```

```
## [1] 1
```

`rfc_model`inin eğitim verisi üzerinde doğruluk oranı %100'dür.

```
varImpPlot(rfc_model)
```




Ağacın saflığına en çok etki eden iki değişkenimiz `ShellWeight` ve `WholeWeight` değişkenleriyken, sınıflandırma doğruluğuna en çok katkıda bulunan üç değişkenimiz `ShellWeight`, `ShuckedWeight` ve `Diameter` değişkenlerimizdir.

## 9-Lojistik Regresyon(LR)

```
set.seed(106)
lr_model <- glm(INI~.-WholeWeight-VisceraWeight-Rings-Length,
```

```
               data=train_abaloneClass,
               family = binomial)
summary(lr_model)
```

```
##
## Call:
## glm(formula = INI ~ . - WholeWeight - VisceraWeight - Rings -
##     Length, family = binomial, data = train_abaloneClass)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.4875     0.7239  -8.962  < 2e-16 ***
## Diameter        9.2686     3.1505   2.942  0.00326 **
## Height         22.3231     5.4794   4.074 4.62e-05 ***
## ShuckedWeight -10.0366     1.2781  -7.853 4.07e-15 ***
## ShellWeight    27.8456     3.4136   8.157 3.43e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2914.9  on 2921  degrees of freedom
## Residual deviance: 1323.5  on 2917  degrees of freedom
## AIC: 1333.5
##
## Number of Fisher Scoring iterations: 8
```

- Model, şu haliyle anlamsız katsayı bulundurmamaktadır.

```
set.seed(106)
vif(lr_model)
```

```
##      Diameter        Height ShuckedWeight   ShellWeight
##      8.677155      3.341249      5.963234      9.023293
```

- Çoklu doğrusal bağlantı yoktur.

```
set.seed(106)
durbinWatsonTest(lr_model)
```

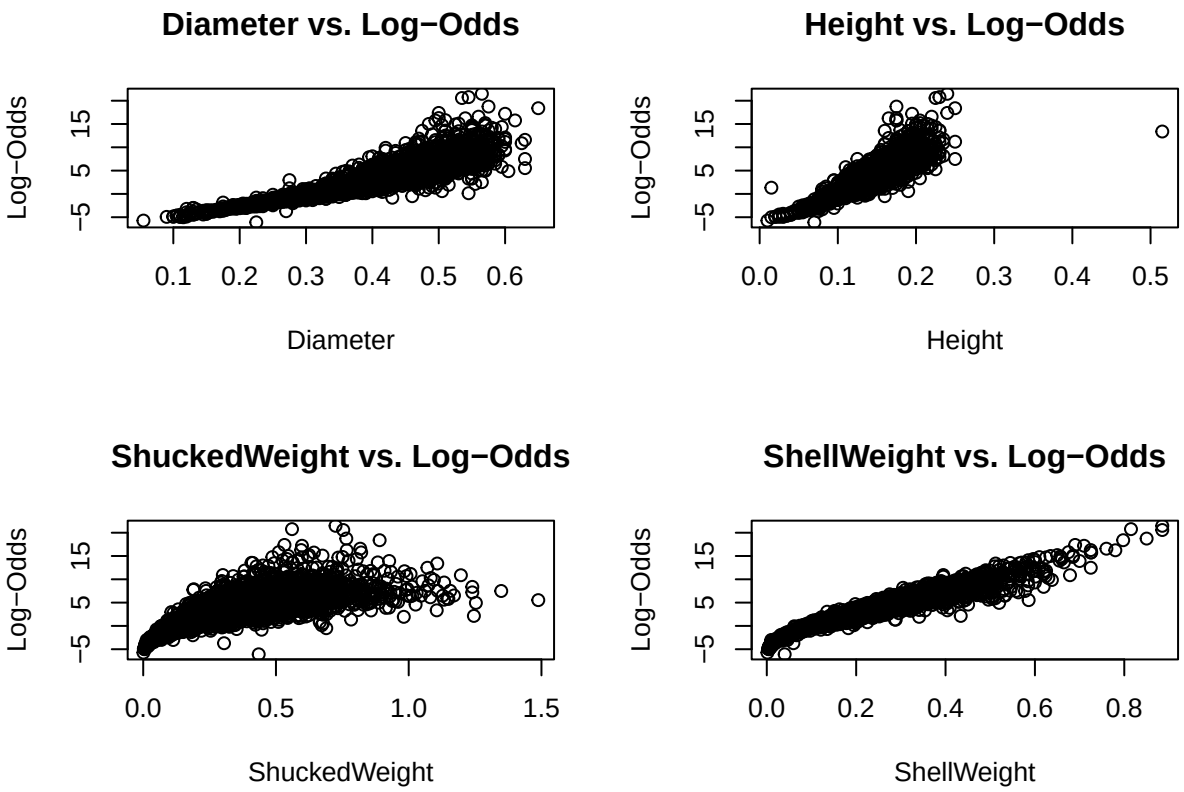
```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02802168      2.056038   0.162
##  Alternative hypothesis: rho != 0
```

- $H_0$ hipotezi otokorelasyon(gözlemler arasında lineer ilişki) yoktur, hipotezimiz Durbin Watson testi sonucu p-value>0.05 olduğu için reddedilemez. Dolayısıyla gözlemler birbirinden bağımsızdır.

```
nrow(train_abaloneClass)
```

```
## [1] 2922
```

- 4 bağımsız değişken üzerinden kurduğum lojistik regresyon modeli için örneklem genişliğimiz yeterince fazladır, dolayısıyla bu varsayım da sağlanmış olur.




- Bağımsız değişkenler ile Log-Odds değerleri arasındaki oluşturduğum grafikleri incelediğimizde lineerlik olduğu söylenebilir. Bu durumu test eden istatistik veya kod bulamadığım, bulduklarımdan yorum yapamadığım için grafiklere göre yorumladım. `ShuckedWeight` değişkeni ile Log-Odds değerleri arasındaki ilişkiye bu varsayımı sağlatmama ihtimali olduğunu söyleyebilirim.

```
set.seed(106)
lrPredicts <- predict(lr_model, newdata = train_abaloneClass)
predicted_classes <- ifelse(lrPredicts > 0.5, "Not Infant", "Infant")
lr_conf_matrix <- table(predicted_classes, train_abaloneClass$INI)
lr_conf_matrix
```

```
##
## predicted_classes Infant Not Infant
##        Infant        463        170
##        Not Infant    118       2171
```

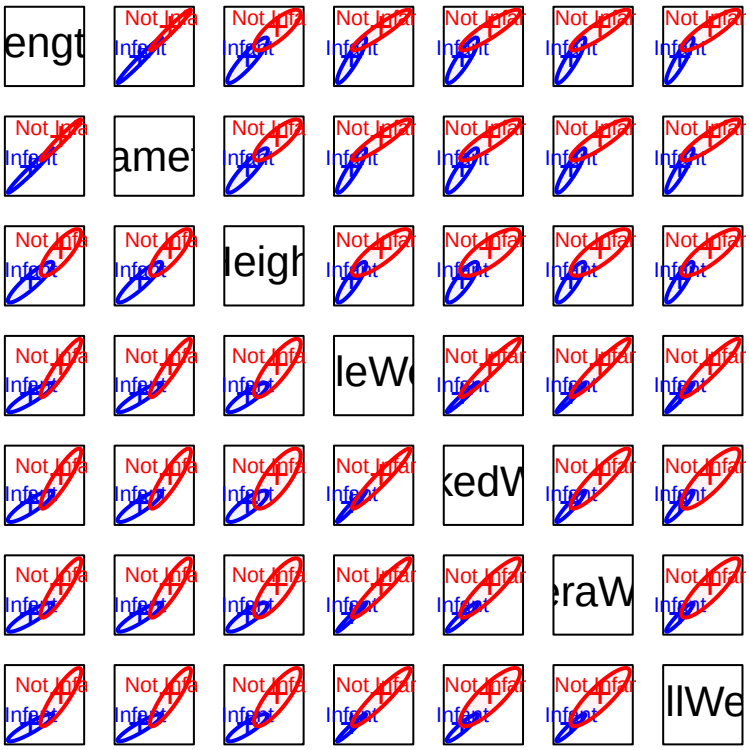
Eğitim seti üzerinden karışıklık matrisine baktığımda `lr_modelim`, `Infant` sınıfından olan 118 gözlemi `Not Infant` olarak etiketlerken, 170 gözlemi `Not Infant` sınıfından olmasına rağmen `Infant` sınıfında tahmin etmiş.

```
set.seed(106)
sum(diag(lr_conf_matrix)) / sum(lr_conf_matrix)
```

```
## [1] 0.9014374
```

Doğruluk değerine baktığımda %90 civarında bir başarısı olduğu söyleyebilirim.

## 10-Doğrusal Ayırma Analizi(LDA)



Grafiklere baktığımızda tüm değişkenler için neredeyse aynı durum gözüküyor, yani tüm değişkenlerde sınıflandırma için doğrusal ayırma yapılabilir.

```
set.seed(106)
lda_model <- lda(INI ~.-Rings-Length-VisceraWeight-Diameter,
                 data = train_abaloneClass)
lda_model
```

```
## Call:
## lda(INI ~ . - Rings - Length - VisceraWeight - Diameter, data = train_abaloneClass)
##
## Prior probabilities of groups:
##     Infant Not Infant
##  0.1988364  0.8011636
##
## Group means:
##                 Height WholeWeight ShuckedWeight ShellWeight
## Infant     0.09126506   0.2878468     0.1312513  0.08158262
## Not Infant 0.15244126   0.9745579     0.4209761  0.28103738
##
## Coefficients of linear discriminants:
##                         LD1
## Height         28.91935033
## WholeWeight     0.01541341
## ShuckedWeight -0.69325154
## ShellWeight     2.04235918
```

- **Prior probabilities of groups:**

  - "Infant" sınıfını içeren gözlemleri eğitim veri setindeki toplam gözlem sayısına göre %19.88 oranında tahmin ediyor.
  - "Not Infant" sınıfını içeren gözlemleri eğitim veri setindeki toplam gözlem sayısına göre %80.12 oranında tahmin ediyor.

- **Group means:** Sınıflarımıza ait grupların ortalamalarını göstermektedir.
- **Coefficients of linear discriminants:** Doğrusal ayrıştırma yönteminde sadece 1 tane ayırma fonksiyonu üretilmiş ve bu doğrusal fonksiyon için katsayıları yazdırılmıştır.

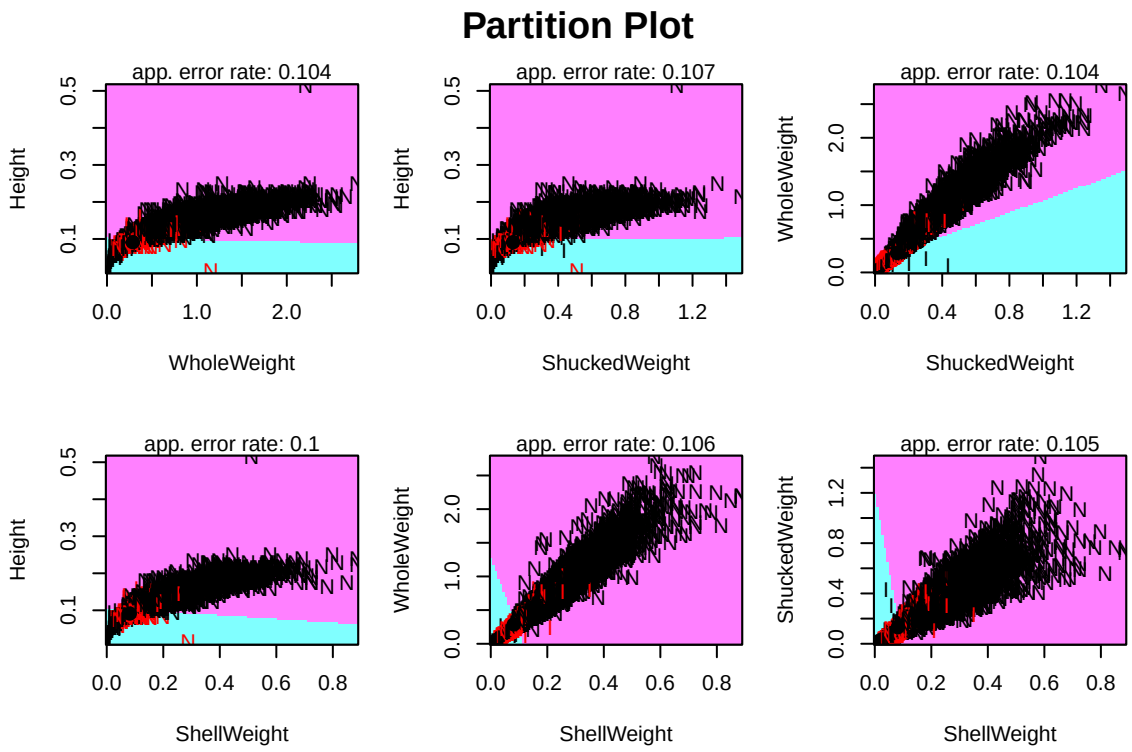
```
boxm <- boxM(train_abaloneClass[, 1:7], train_abaloneClass$INI)
boxm
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  train_abaloneClass[, 1:7]
## Chi-Sq (approx.) = 3288.3, df = 28, p-value < 2.2e-16
```

- $H_0$ Sonuç değişkeninin kovaryans matrisleri tüm gruplarda eşittir.
- $H_1$ Sonuç değişkeninin kovaryans matrisleri en az bir grup için farklıdır.

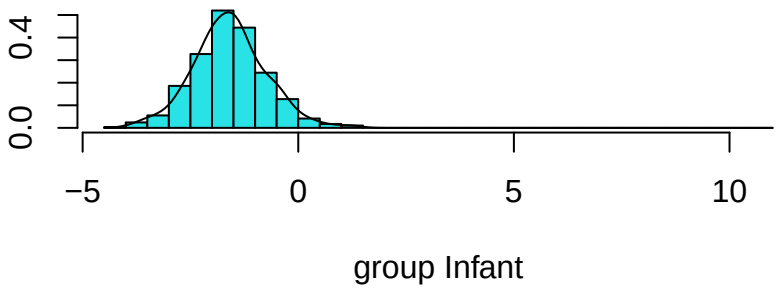
p-değeri$< 0.05$ olduğu için null hipotezi reddedilir. Yani, bu durumda grupların kovaryans matrisleri en az bir grup için farklıdır.

```
partimat(INI ~.-Rings-Length-VisceraWeight-Diameter,
         data=train_abaloneClass,method="lda")
```

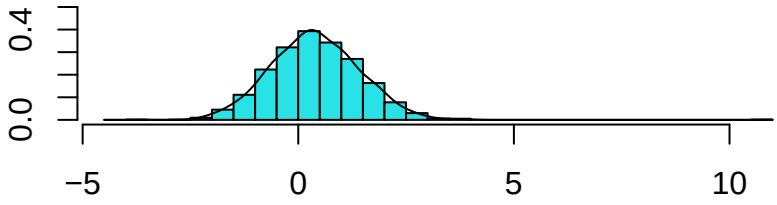



Doğrusal ayırma analizi ile partition matrisine baktığımda hata oranların birbirine çok yakın ve %10 civarında olduğu görülmekte.

```r
tahmin_1<-predict(lda_model,train_abaloneClass)
hist_lda1<-ldahist(data=tahmin_1$x[,1],g=train_abaloneClass$INI, type = "both")
```




group Infant




group Not Infant

Bu histogram grafiklerine baktığımda ayrışmanın net olmadığı, tahminlerimde yanılabileceğim durumlar olacağı görülüyor.

```r
set.seed(106)
lda_predictions <- predict(lda_model, newdata = train_abaloneClass)
lda_conf_matrix <- table(lda_predictions$class, train_abaloneClass$INI)
lda_conf_matrix
```

```
##
##              Infant Not Infant
##   Infant        402        114
##   Not Infant    179       2227
```

`lda_model`im 179 gözlem için `Infant` sınıfında olması gereken, 114 gözlem için de `Not Infant` sınıfından olması gerekirken yanlış tahminlemiş.

```r
sum(diag(lda_conf_matrix))/sum(lda_conf_matrix)
```

```
## [1] 0.8997262
```

`lda_model`imin eğitim verileri üzerinden doğruluk oranı %90 civarında.

# 11-Eğrisel Ayırma Analizi(QDA)

Doğrusal ayırma analizininin ilk başında verdiğim grafiklere baktığımızda tüm değişkenler için neredeyse aynı durum gözüküyor, yani tüm değişkenlerde sınıflandırma için eğrisel ayırma yapılabilir.

```
set.seed(106)
qda_model <- qda(INI~.-Rings-WholeWeight-Diameter-ShuckedWeight,
                 data=train_abaloneClass)
qda_model
```

```
## Call:
## qda(INI ~ . - Rings - WholeWeight - Diameter - ShuckedWeight,
##     data = train_abaloneClass)
##
## Prior probabilities of groups:
##     Infant Not Infant
##  0.1988364  0.8011636
##
## Group means:
##                Length     Height VisceraWeight ShellWeight
## Infant     0.3663081 0.09126506    0.06205938  0.08158262
## Not Infant 0.5652606 0.15244126    0.21272661  0.28103738
```

- **Prior probabilities of groups:**
  - "Infant" sınıfını içeren gözlemleri eğitim veri setindeki toplam gözlem sayısına göre %19.88 oranında tahmin ediyor.
  - "Not Infant" sınıfını içeren gözlemleri eğitim veri setindeki toplam gözlem sayısına göre %80.11 oranında tahmin ediyor.
- **Group means:** Sınıflarımıza ait grupların ortalamalarını göstermektedir.

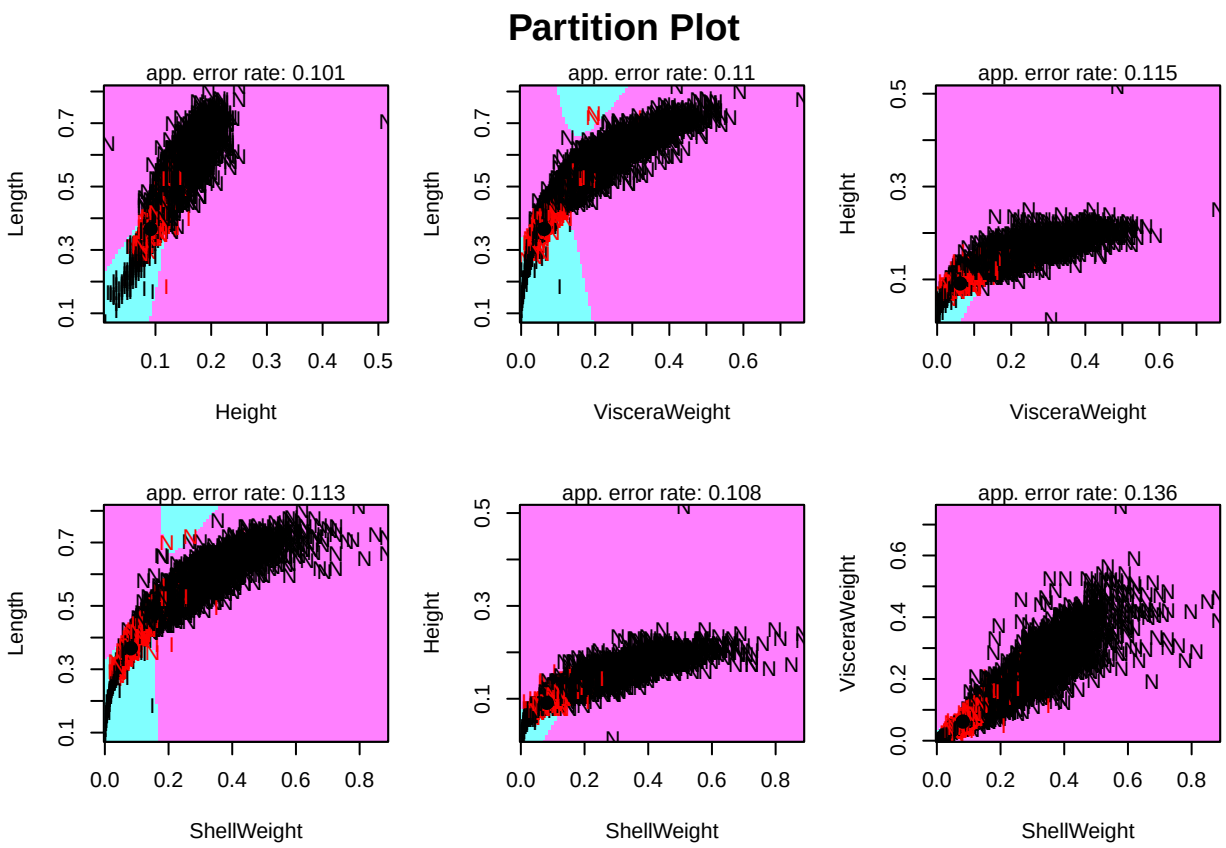
```
boxm <- boxM(train_abaloneClass[, c(1,3,6,7)], train_abaloneClass$INI)
boxm
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  train_abaloneClass[, c(1, 3, 6, 7)]
## Chi-Sq (approx.) = 2307.6, df = 10, p-value < 2.2e-16
```

- $H_0$ Sonuç değişkeninin kovaryans matrisleri tüm gruplarda eşittir.
- $H_1$ Sonuç değişkeninin kovaryans matrisleri en az bir grup için farklıdır.

p-değeri$< 0.05$ olduğu için null hipotezi reddedilir. Yani, bu durumda grupların kovaryans matrisleri en az bir grup için farklıdır.

```
partimat(INI~.-Rings-WholeWeight-Diameter-ShuckedWeight,
                 data=train_abaloneClass,method="qda")
```




Eğrisel ayırma analiziyle partition matrisine bakıp yüksek olduğu durumlara etki eden değişkenleri modelimden çıkararak `qda_model`imi tekrar kurdum. Farklı sonuçlarla karşılaşmadım ve şu haliyle bıraktım. Buna göre ayırmaya göre hata oranı en fazla %13.6 ile `VisceraWeight` ve `ShellWeight` arasında gerçekleşen ayırmadadır.

```
set.seed(106)
qda_predictions <- predict(qda_model, newdata = train_abaloneClass)
qda_conf_matrix <- table(qda_predictions$class, train_abaloneClass$INI)
qda_conf_matrix
```

```
##
##              Infant Not Infant
##   Infant        489        296
##   Not Infant     92       2045
```

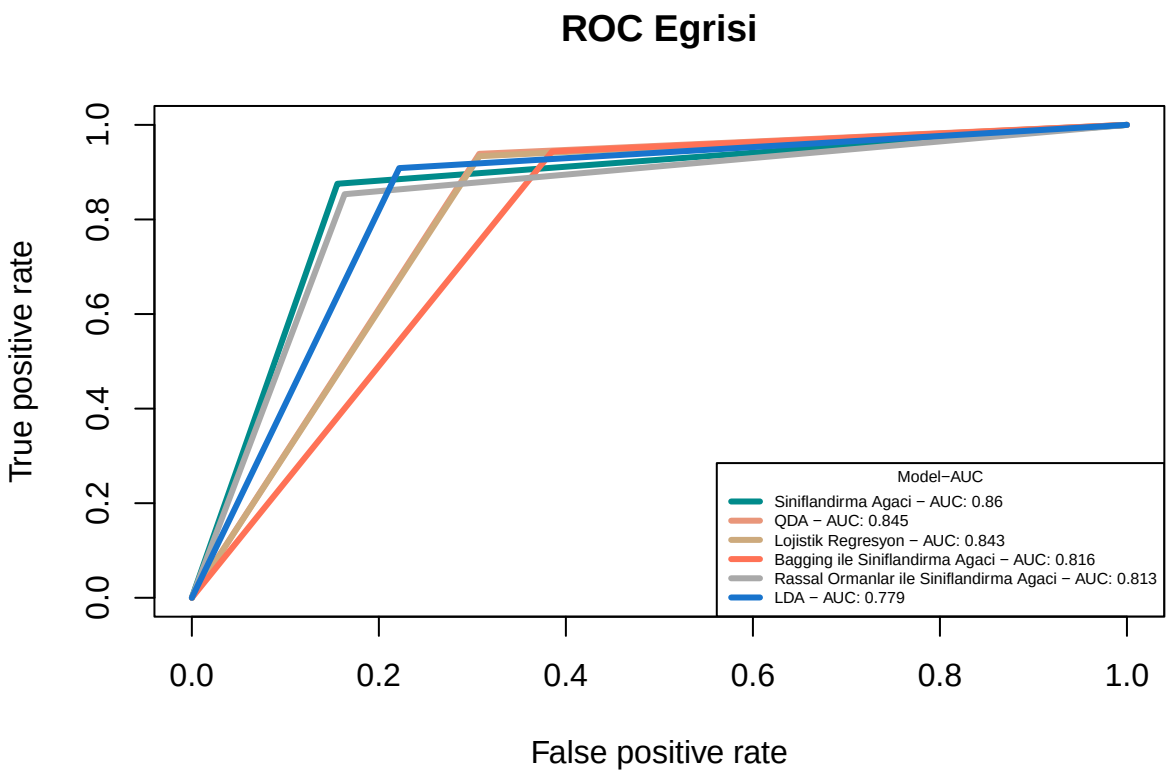
Karışıklık matrisine baktığımızda 92 adet `Infant` sınıfında olması gereken, 296 adet de `Not Infant` sınıfında olması gereken gözlem yanlış tahmin edilmiş.

```
sum(diag(qda_conf_matrix))/sum(qda_conf_matrix)
```

```
## [1] 0.8672142
```

Eğrisel ayrıştırma modelim eğitim verileri üzerinde neredeyse %87 doğru tahminler üretmiş.

## 12-ROC ve AUC




ROC (Receiver Operating Characteristic) eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir grafiksel araçtır. Bu eğri, modelin hassasiyet (sensitivity) ve özgüllük (specificity) performansını görsel olarak gösterir.

Eğri altında kalan alan (AUC - Area Under the Curve), modelin sınıflandırma yeteneğini ölçen bir değerdir. AUC değeri 1'e ne kadar yakınsa, modelin performansı o kadar iyidir.

Her model için karışıklık matrisi ve doruluk değerlerine modele ait bölümlerde incelediğim için, eğrilerimi test seti üzerinden yaptırdığım tahminlere göre çizdirdim. Çizimi yaptırırken kullandığım kodlar `.rmd` dosyasında mevcuttur. Buna göre test setinde en iyi tahmin yapan modelim sınıflandırma ağacı modelim yani `ct_model` modeli. Legend içerisinde `AUC` değerlerine göre en yüksekten en düşüğe sıralı şekilde verilmiştir.

## 13-Karşılaştırma

```
test_matrixes <- list(ct_conf_matrix_test,
                      bct_conf_matrix_test,
                      rfc_conf_matrix_test,
                      lr_conf_matrix_test,
                      lda_conf_matrix_test,
                      qda_conf_matrix_test)

accuracies <- sapply(test_matrixes,
                     function(matris)
                       sum(diag(matris))/sum(matris))
```

```r
model_names <- c("Sınıflandırma Ağacı", "Bagging ile Sınıflandırma Ağacı",
                 "Rassal Ormanlar ile Sınıflandırma Ağacı",
                 "Lojistik Regresyon",
                 "LDA", "QDA")

accuracy_table <- data.frame(Model = model_names,
                             Accuracy = unlist(accuracies))

order_indices <- order(accuracy_table$Accuracy, decreasing = TRUE)

accuracy_table <- accuracy_table[order_indices, ]

accuracy_table
```

```
##                                    Model  Accuracy
## 2         Bagging ile Sınıflandırma Ağacı 0.8882682
## 3 Rassal Ormanlar ile Sınıflandırma Ağacı 0.8842777
## 4                      Lojistik Regresyon 0.8818835
## 5                                     LDA 0.8762969
## 1                     Sınıflandırma Ağacı 0.8691141
## 6                                     QDA 0.8499601
```

Test verileri üzerinden yapılan tahminlemelerin doğruluk değerlerine bakıldığında, tahmin performansı en iyi olan %88.8 ile Bagging ile Sınıflandırma Ağacı modeli ve onun hemen ardından %88.4 ile Rassal Ormanlar geliyor. Fakat bu modellerin her ikisi de eğitim setindeki doğruluk değerleri %100'ken test setinde başarısında ciddi bir düşüş yaşamış.

Lojistik Regresyon modelimiz eğitim seti üzerinde %90 başarılı tahminler yaparken test setindeki tahminleri %88.2 civarındadır. BCT ve RFC modellerine göre performans düşüşü daha az gerçekleşmiş.

Sınıflandırma Ağacı modelim eğitim verileri üzerinde %88 civarında doğruluk değerini yakalarken, test verilerinden yaklaşık %87'sini doğru tahmin etmiş.

LDA ve QDA için varsayımların sağlanmasında emin olmadığım durumlar var. LDA ve QDA modellerimin performansları karşılaştırdığımda, QDA'nın daha başarısız olmasının sebebi modelimi farklı değişkenlerle kurmuş olmamdan kaynaklandığını söyleyebilirim.

Eğitim ve test verileri üzerinden modellerin performanslarına baktığımda aşırı öğrenme durumunun bulunmasını istemediğim için, Lojistik Regresyon modelim yani `lr_model` modelimi diğerlerine göre daha başarılı buldum.