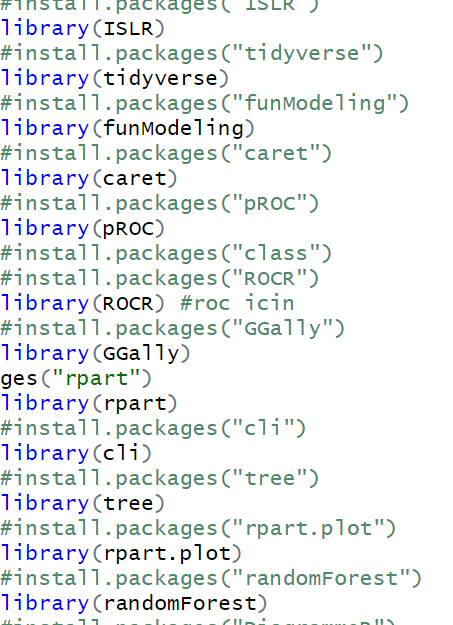
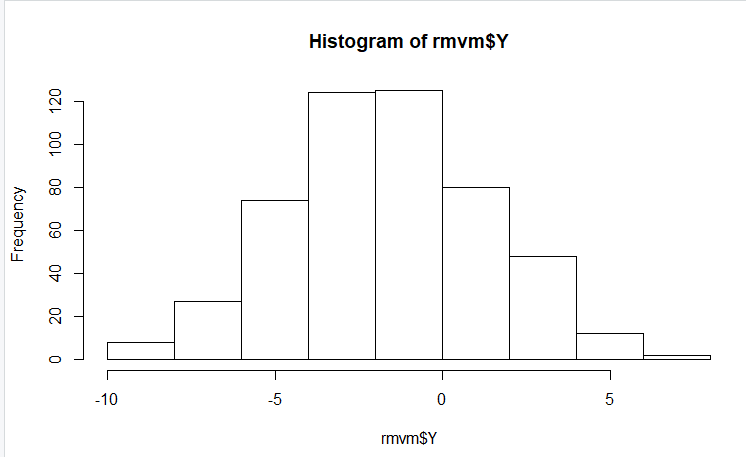
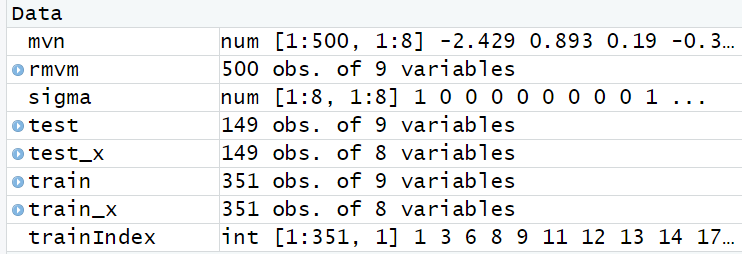
SİMAY UĞUR

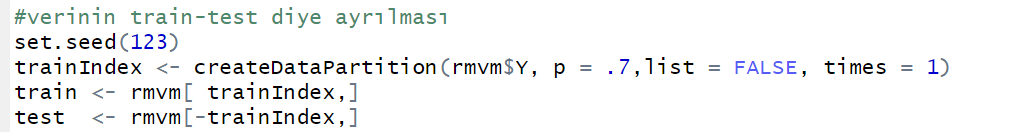
KULLANILAN KUTUPHANELER:

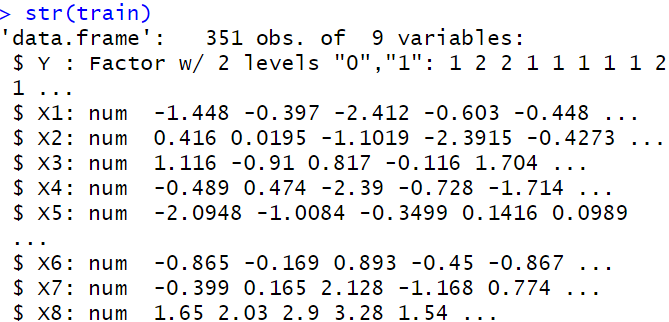


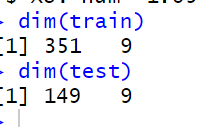
Y nin factor olmadan önceki histogram Dağılım grafiği.



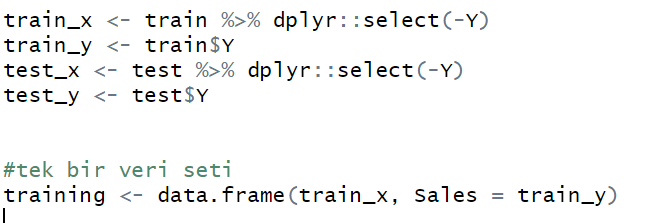








Train gözlem sayım 351,test setinde ki gözlem sayım 149 .Gerçekten veriler %70 train %30 test diye ayrılmıştır.

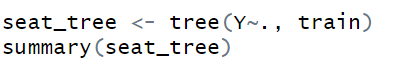


**KARAR AĞAÇLARI:**

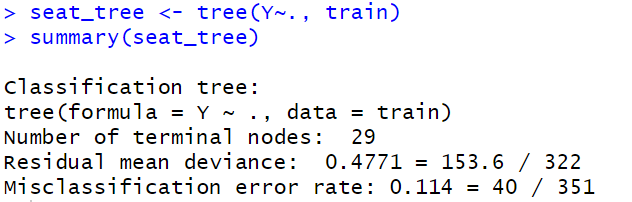
Amaç veri setinde ki karmaşık yapıları basit karar yapılara dönüştürmektir.

Veri seti belirlenmiş hedef değişkene (Y) göre alt gruplara ayrılır.

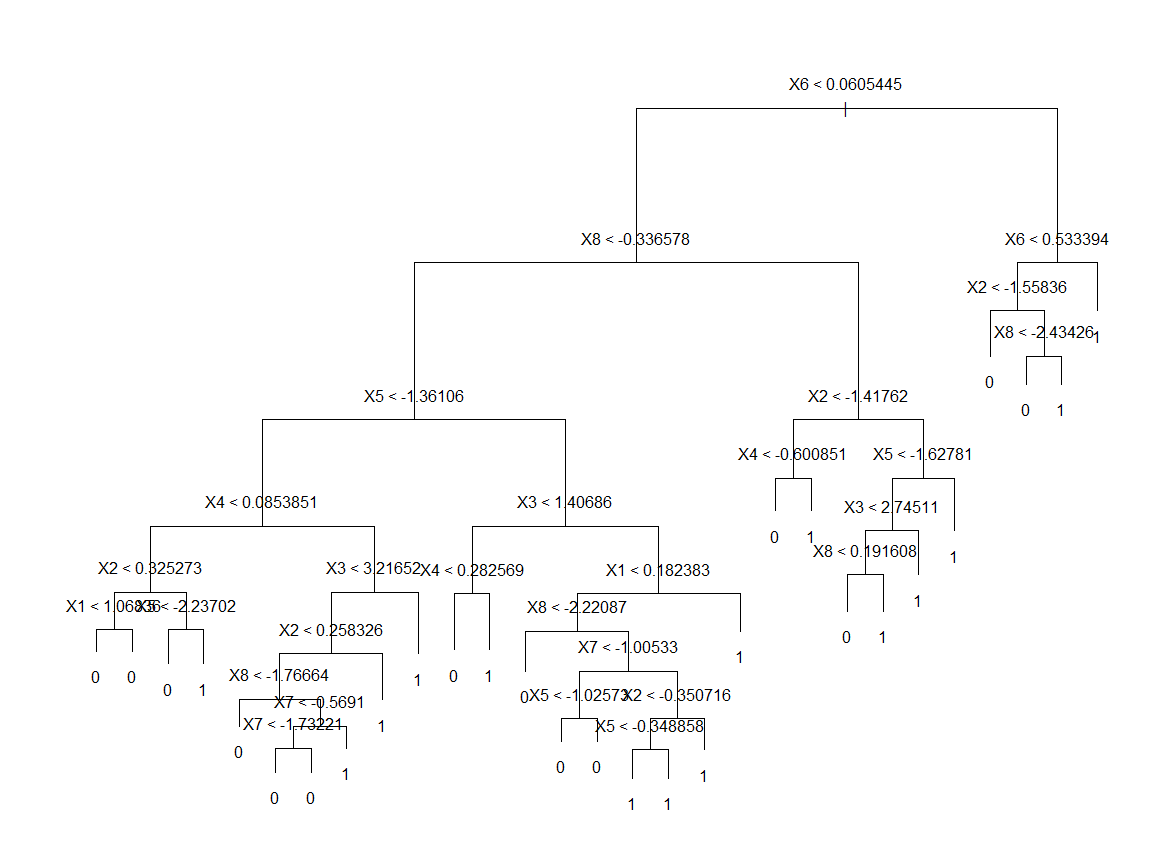
**Model**:



Bu kod ile Prune edilmeden direk ağaç çıktısı elde ederiz.



Ağacta 29 node var.Yanlış sınıflandırma oranı %11.4 yani Hatalı sınıflandırılan örneklerin toplama oranı 0,114 tür.



Karmaşıklık durumunu ortaya çıkardığı 2 durum var.

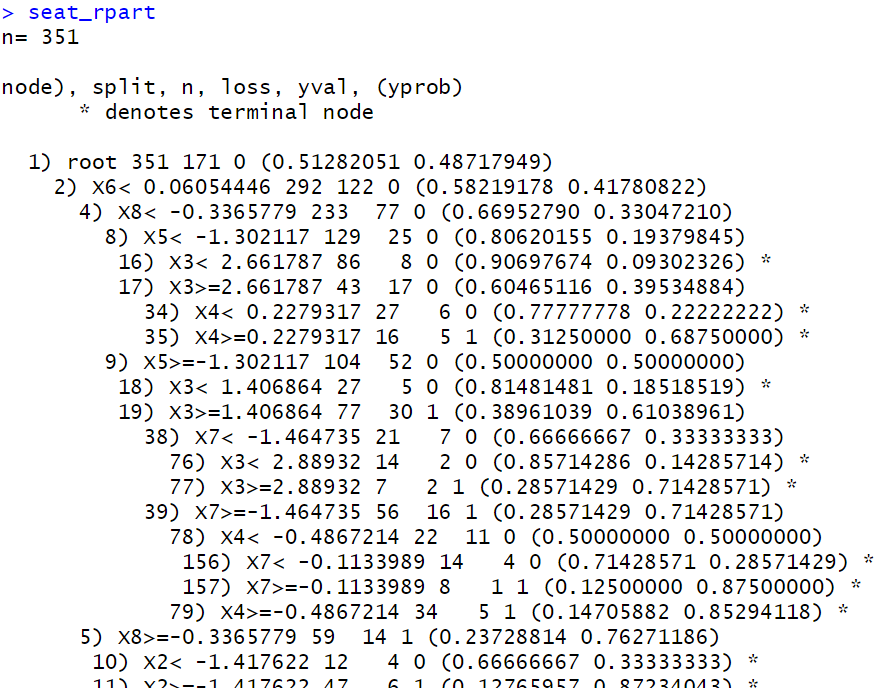
1. Görsel anlamda kötü.Karmaşık görsel yorumlama zordur.
2. Train setinin başarılı olması bakımından iyidir.Fakat overfitting e sebeb olabilir yani Bu yapıyı detaylı bir şekilde öğrenip hiç görmediği bir veri setinde uygulandığında test başarısı düşer.Prune edilmesi gerek yani Budama işlemi yapılır.

Karmaşıklık parametresi ekleyerek Overfitting önüne geçilir.

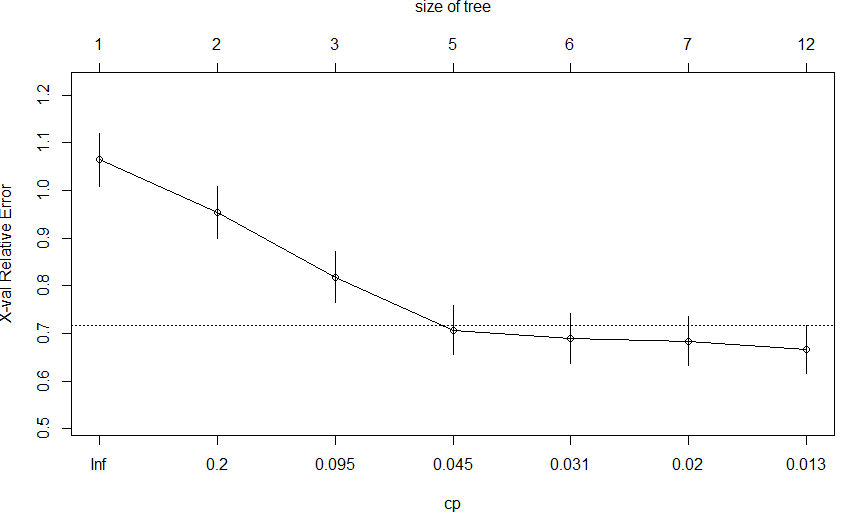
**Rpart Fonksiyonu ile modelleme:**

Rpart fonksiyonu görselleştirme açısından daha zengindir.

Rpart Hatalara daha dayanıklıdır.

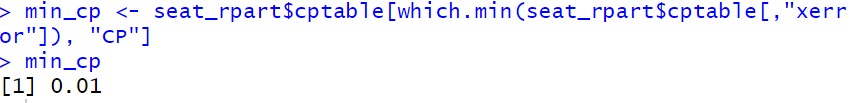


Karar yapılarını gözlemlemiş oluruz.



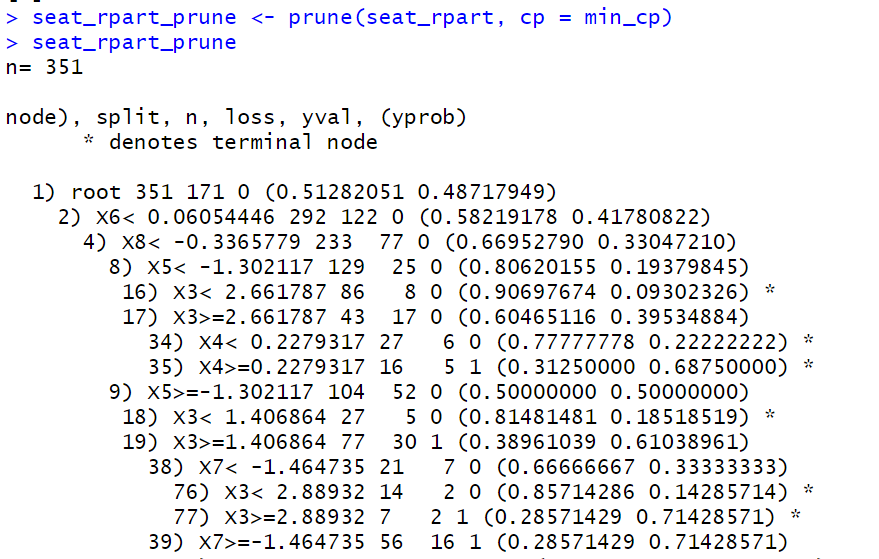
Karmaşıklık parametresine karşılık oluşan değerler verilmiştir.Karmaşıklık parametresi Overfittingin önüne geçmek için kullandığımız ve ağaca dur dediğimiz budama işlemini yaptığımız parametredir.

En iyi karmaşıklık parametresine erişmek istediğimizi varsayalım.



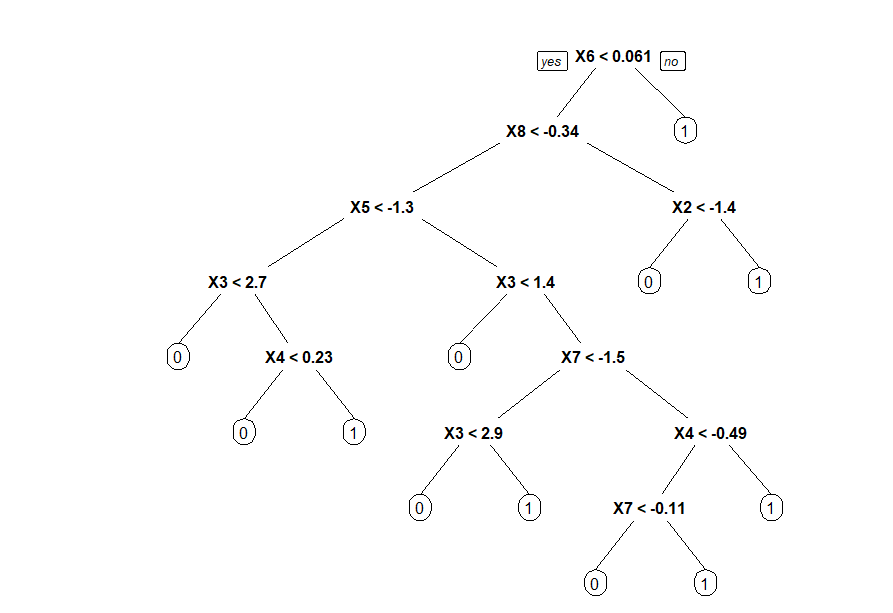
En iyi cp değeri 0.01.

Ağacın budanmış yapısı:

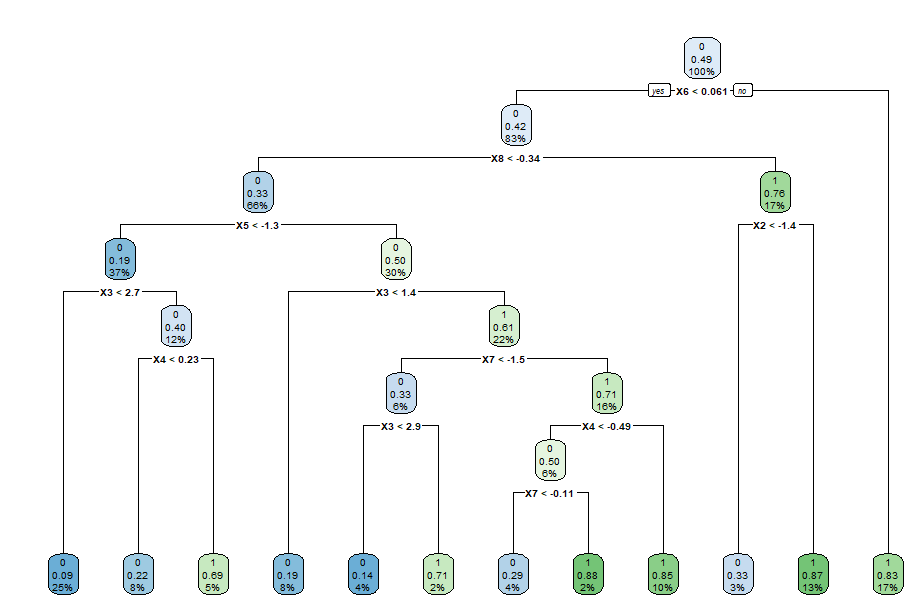


> prp(seat\_rpart\_prune)

Budanmış ağacın görseli:



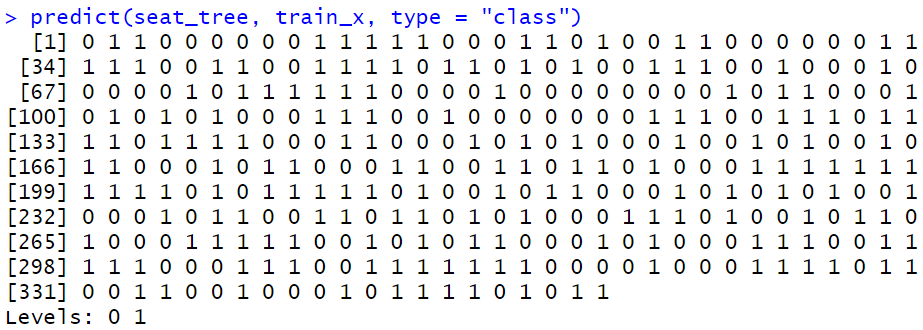
Gelişmiş rpart\_prune gösterimi:



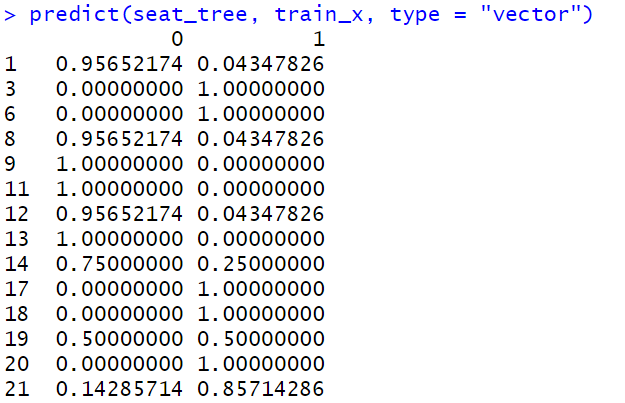
En önemli değişken X6 1. Önemli değişkenimdir.2.önemli değişkenimX8 dir.

12 node düşmüş oldu.

**TAHMİN:**

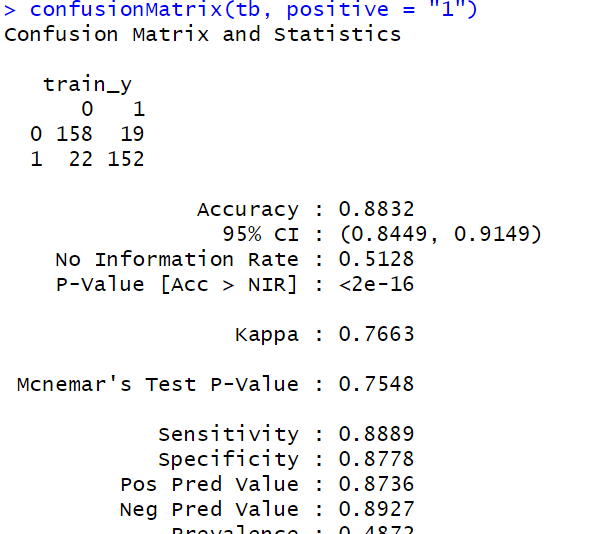


Sınıf tahmin değerleri



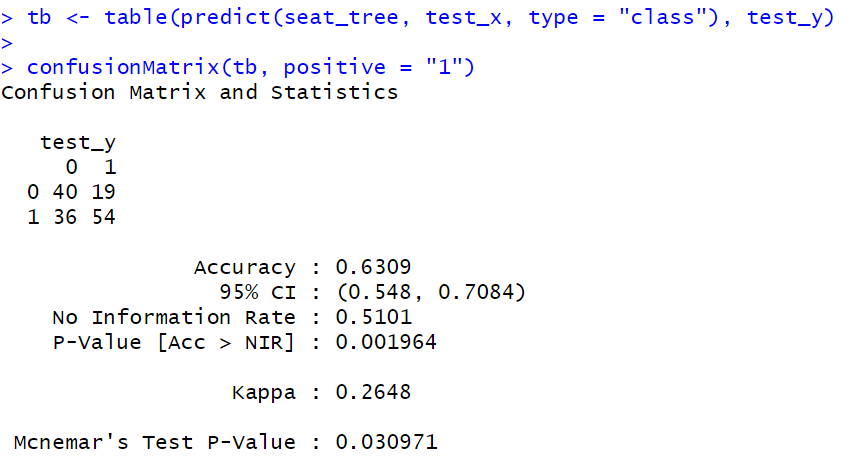
Olasılıksal tahmin değerleri.

Eğitim seti için karmaşıklık matrisi oluşturalım.



Doğru sınıflandırma oranımız:%88

Test için confusion matrix oluşturalım:

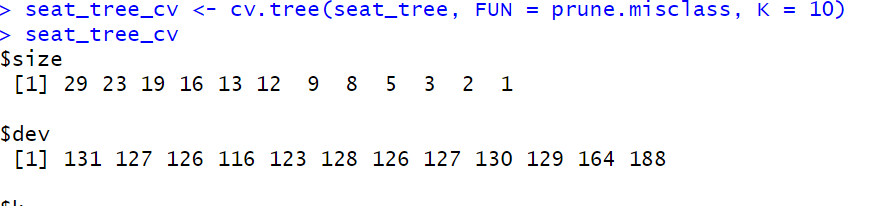


Görüldüğü üzere daha düşük Accuracy değeri elde ettik.Eğitim setinde eğitim başarısında doğru sınıflandırma oranımız daha yüksekti.Overfitting söz konusu.

Model tuning de overfitting oranının düşürmeye çalışacağım.

**Model Tuning.:**

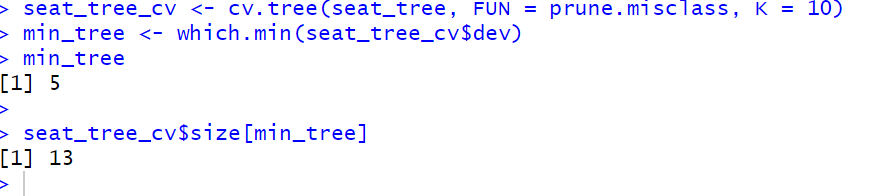
CV ile budama yaparak model tuning islemleri:



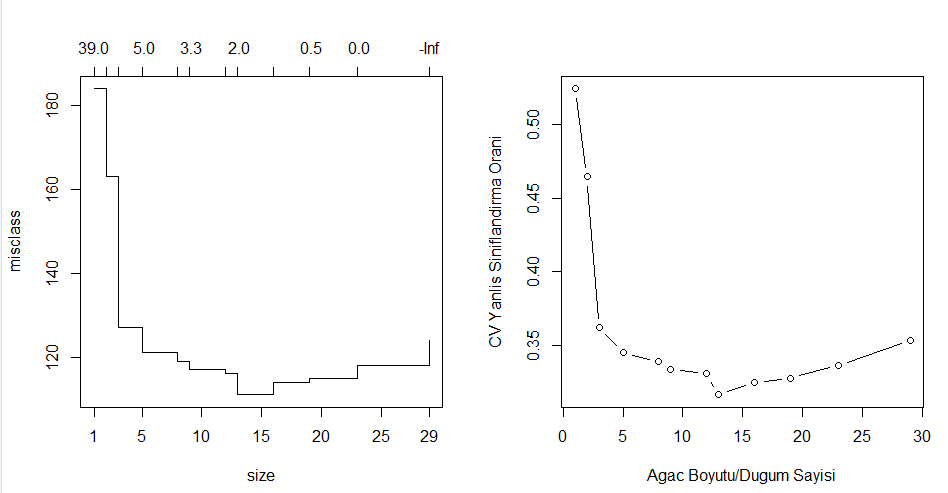
Fun ile budama işlemi gerçekleştirilir.

En küçük hataya sahip ağacın indexi:

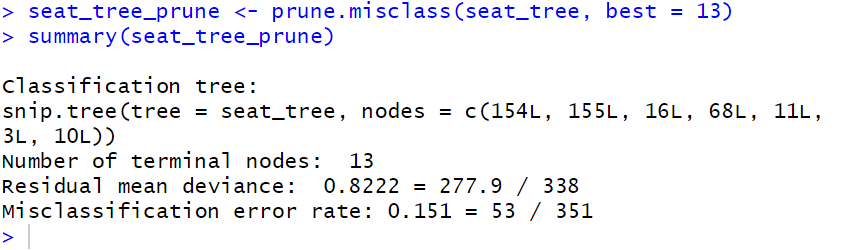
Ağaç kaç node dan oluşmuş?



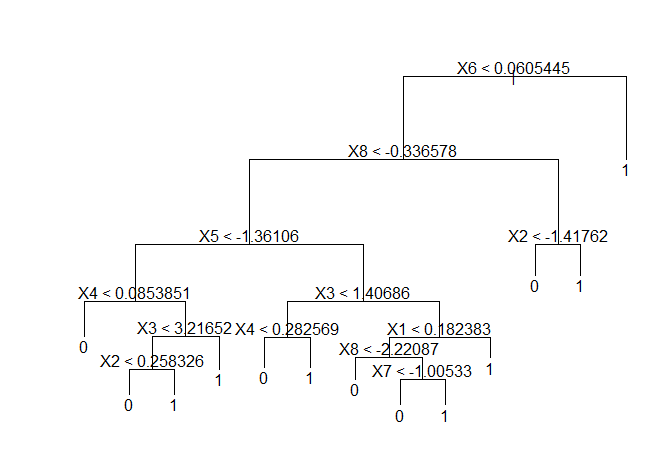
En küçük hataya sahip ağacın indexi 5 . Ağaç 13 node dan oluşmuş.



Ağaç boyutuna karşı yanlış sınıflandırma oranının nasıl gerçekleştiğini gözleriz .Yani düğüm sayısı yaklaşık 13 olunca en düşük hataya sahip oluruz denebilir.

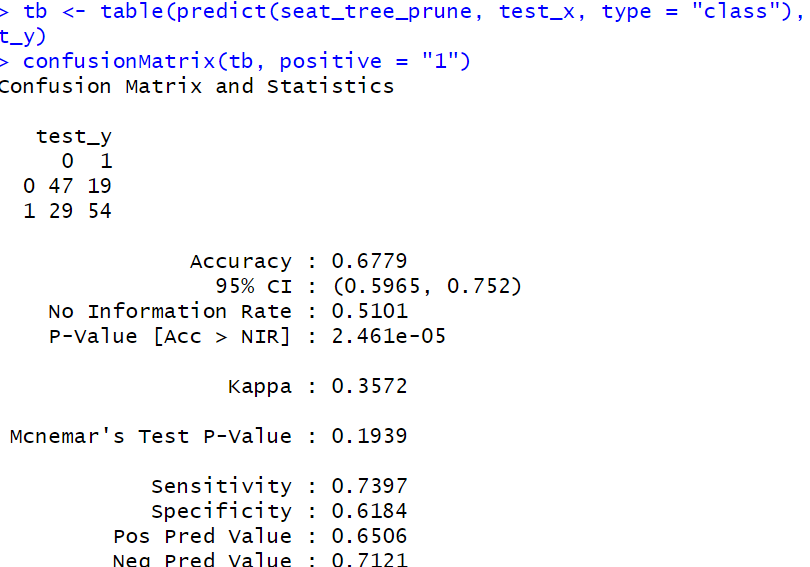


Node sayısı 13 olduğunda .



Karmaşıklık daha da azalmış.

**SONUÇLARIN KARŞILAŞTIRILIMASI:**

****

Ağacın overfit durumunu azalttığımızdan dolayı accuracy değerim artmış oldu.

Bir önceki acuracy değerim %63 tü .Prune edilen accuracy doğruluk oranım:%67

**BAGGING**

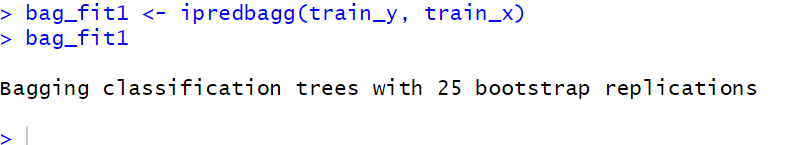
Birden fazla ağaç fit edilmekte ve bu birden fazla ağaç fit edildikten sonra herbirinin ürettiğisonuçlar toplanarak nihai sınıfın tahmininde bulunulur.

Yerine koymalı örneklemleri birleştirmedir Bootsrap Aggregation.Overfitting önüne geçildi.

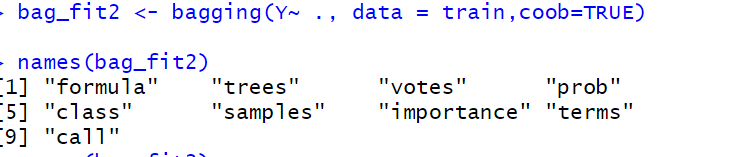
Ağaçların daha önce oluşturulan ağaçları ile bağımlılıkları yoktur.Örneklerin her bir ağacın oluşturulmasında kullanılması şansı eşittir.

3 yöntem var.

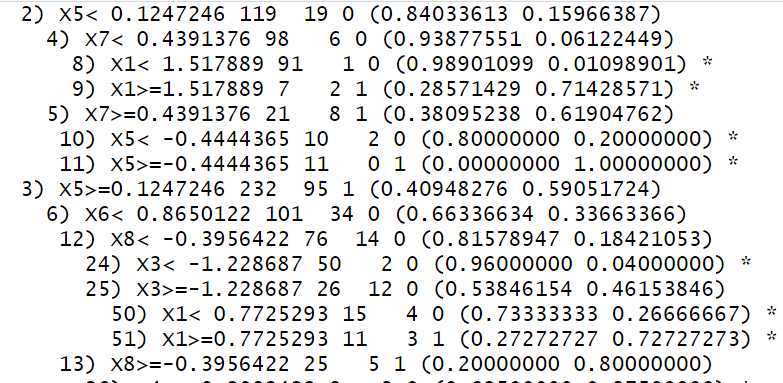
1.yöntem Ipred dir.1. argüman bağımlı değişken ,2.argüman bağımsız değişkendir.

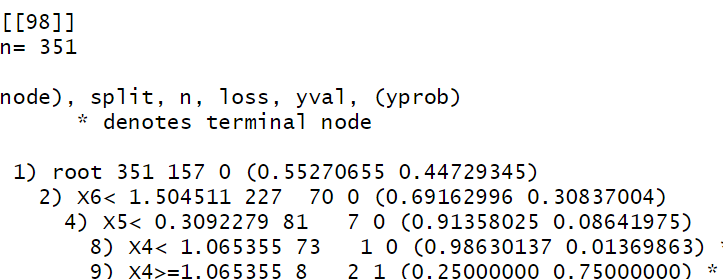


2.yöntem bagging fonksiyonudur.



Kulanılabilecek argümanları gösterir.

bagfit2$trees ile çekilen rartgele örneklemlerden oluşturulan ağaçları göürürz

****

**3.yöntem RANDOM FOREST methodu kullanarak BAGGING.:**

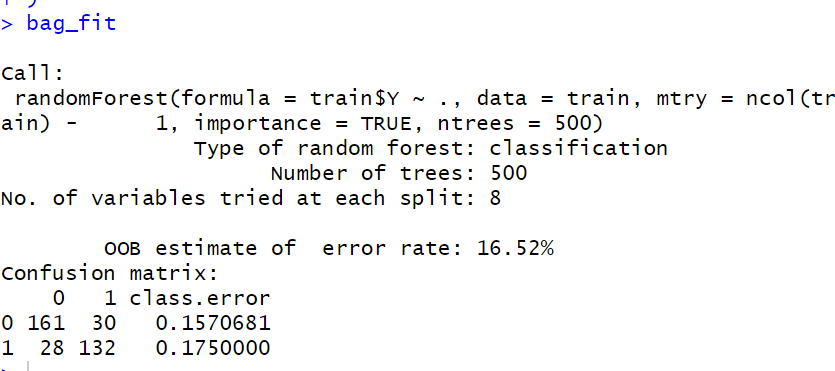
Random Forest ta hem gözlemler hem değişkenler rastgele seçilmektedir.

Baggingde gözlemleri restgele seçilerek birbirinden farklı ağaçlar fit edilmektedir.

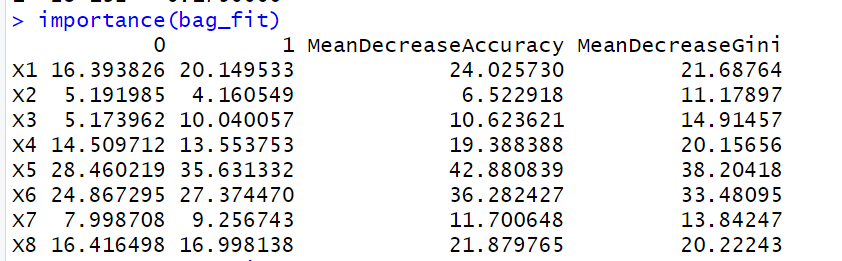
Bir random fonksiyonu içerisinde Seçilecek olan değişken sayısını mevcut değişken sayısına eşitlersek bu durumda random forest bagginge indirgemiş oluruz.

mtry=8 bağımsız değişkenim sayısı kadar.

mtry rastgele değişken seçimi yapma sayısıdır.Rastgele değişken seçimi yapma sayısını veri setindeki değişken sayısına eşitlersem bu durumda geriye sadece gözlemlerden rastgele seçim yapma işlemi kalacağından dolayı random forest algoritmasını bagginge indirgemiş olurum.



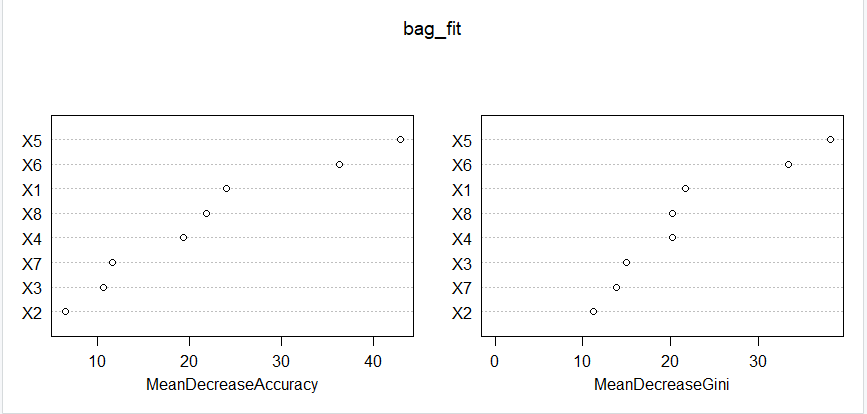
Her bir splitte kullanılan değişken sayım 8 dir. 0 için sınıflandırma hatası %15 1 için sınıflandırma hatası %17 dir.Out of bagging(OOB) yani ağaç kendi içinde veriyi ayırıp test eder onun hata oranı %16.52 dir.



Değişken önem düzeyleri yukarıdaki gibidir.

Gini önemi belirli bir değişkeni bölerek ortalama saflık kazancını ölçer.

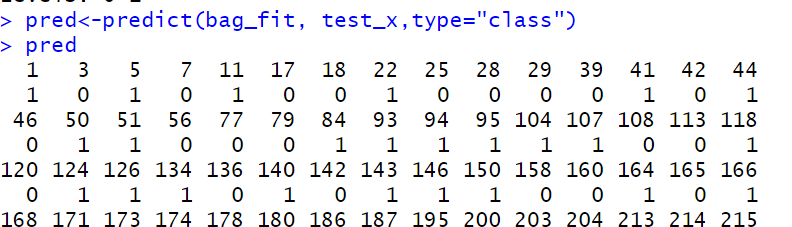
Değişken faydalıysa, karmaşık düğümleri saf tek sınıf düğümlerine bölme eğilimindedir.

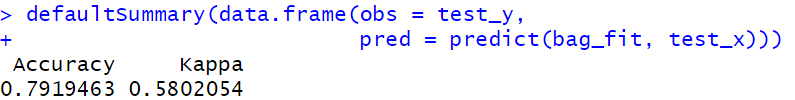


Değişkenlerin önem düzeylerini görselleştirmiş olduk.Ortalama azalan accuracy ve Ortalama azalan gini .

Sonuçlar, bagging uygulanan tüm ağacın X5 ve X6 değişkenlerinin en önemli 2 değişken olduğunu göstermektedir

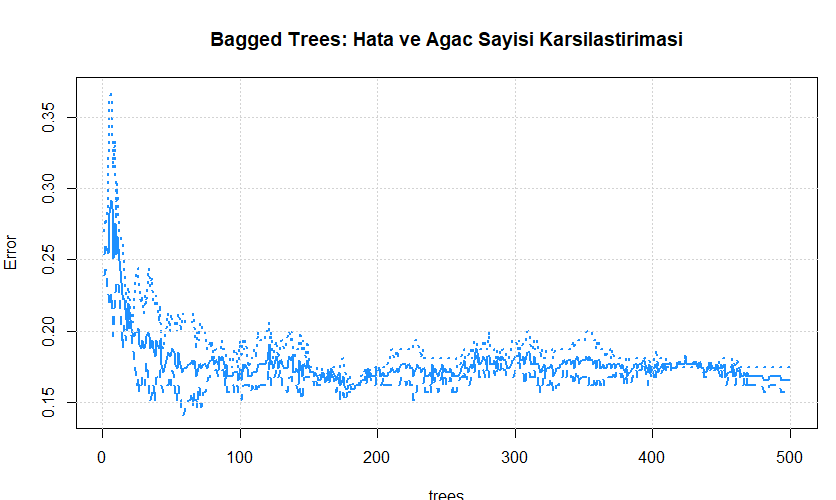
**TAHMİN:**





Birden fazla ağaç oluşturma işlemini gerçekten test hatasına etkisini ölçebilirmiyiz?

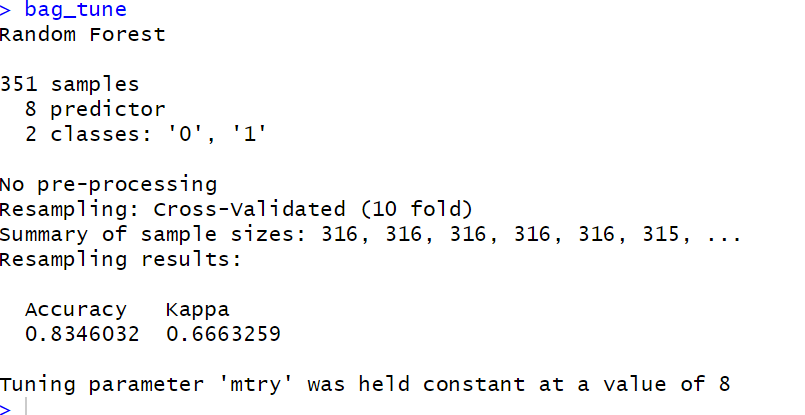
Yani tek bir ağaç oluşturduğumuzda elde ettiğimiz hata ile Bagging yöntemi ile elde ettiğimiz ağaçların ortaya çıkardığı hatalar ağaç sayısı arttıkça nasıl değişir.Bunun için plot ile grafiği çizdirelim.



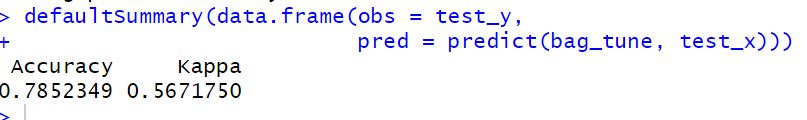
Ağaç sayım arrtıkça hata kareler ortalamam düşmüş.

Bu yöntem rasallığı sağlayarak overfitting yani ezberlemenin önüne geçmektedir .

MODEL TUNING:



Bag\_tune ile modeli valide etmiş olduk.



model doğruluğu % 78 dir.

Doğruluk değeri için düzeltilmiş olan kappa % 56 dır.

RANDOM FOREST:

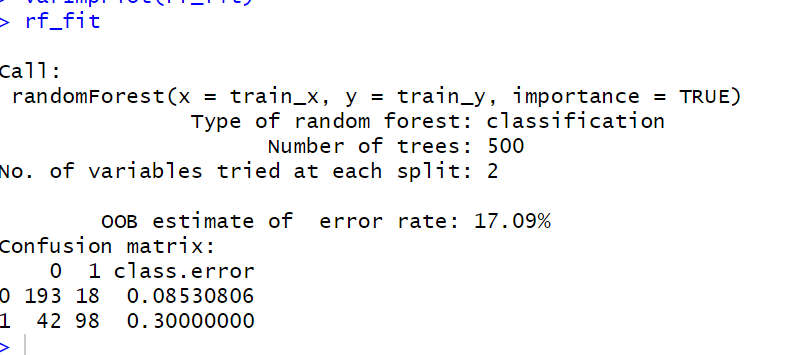
Temeli birden çok karar ağacın ürettiği tahminlerin biraraya getirilerek değerlendirilmesine dayanır.

Ağaçlar için gözlemler bootstrap rastgele örnek seçim yöntemiyle değişkenler random space seçilir.

Karar ağacının herbir düğümünde en iyi dallara ayırıcı (bilgi kazancı) değişken ,tüm değişkenler arasından rastgele seçilen daha az sayıda ki değişken arasından seçilir.

Son tahmin değeri için ağaçların herbirinden tahmin değeri talep edilir her bir ağacın daha önce hesaplanan hata oranları göz önüne alınarak ağaçlara ağırlık verilir.

**MODELİ KURMA:**

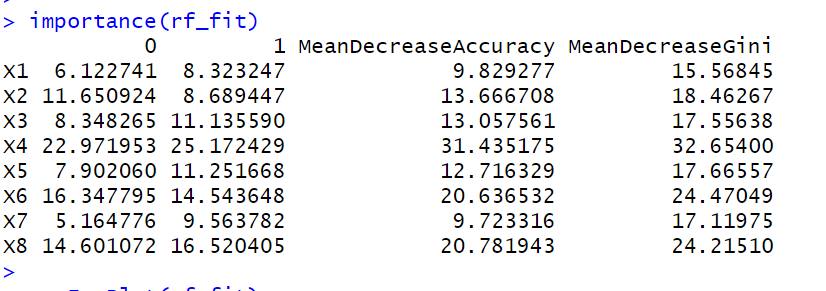


Sınıflandırma hataları için bilgileri görüyoruz.0 için yanlış sınıflandırma hatası%8, 1 için yanlış sınıflandırma hatası %30 .

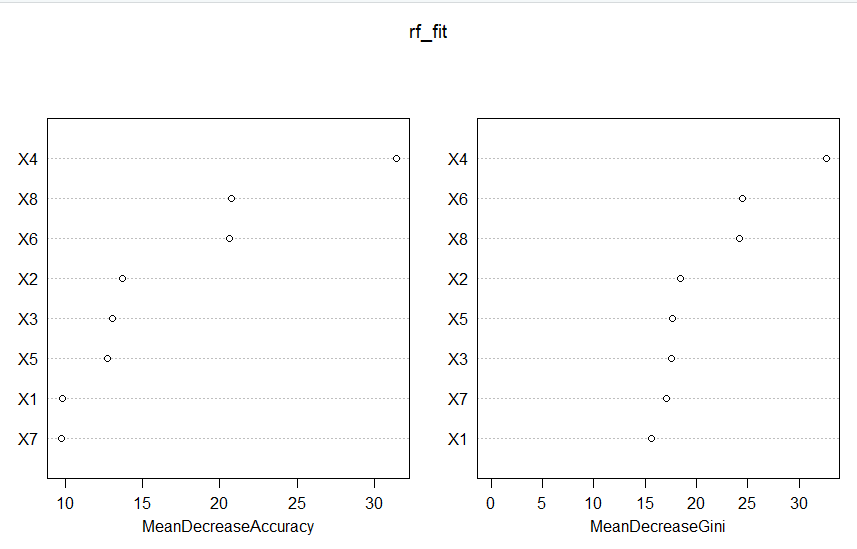
Model kurma dışında bırakılıp Test için kullanılan gözlemler için hata oranı %17dir.

Her bir bölünmede kullanılacak değişken sayısı :2

Fonksiyonda Kullanılan ağaç sayım:500



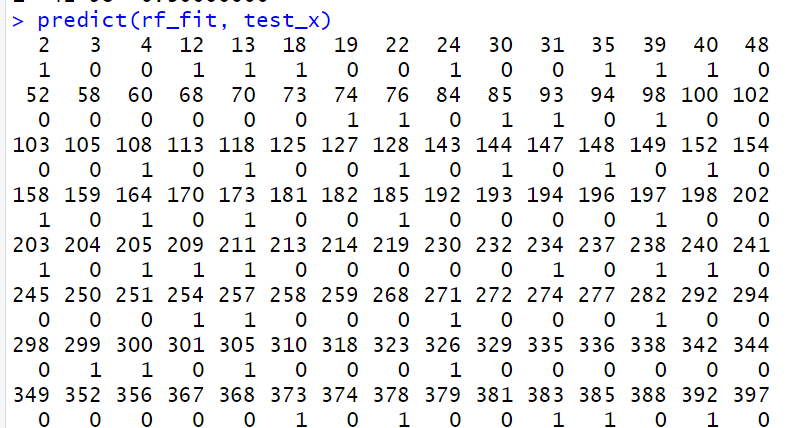
Değişken önem düzeylerini gösterir.



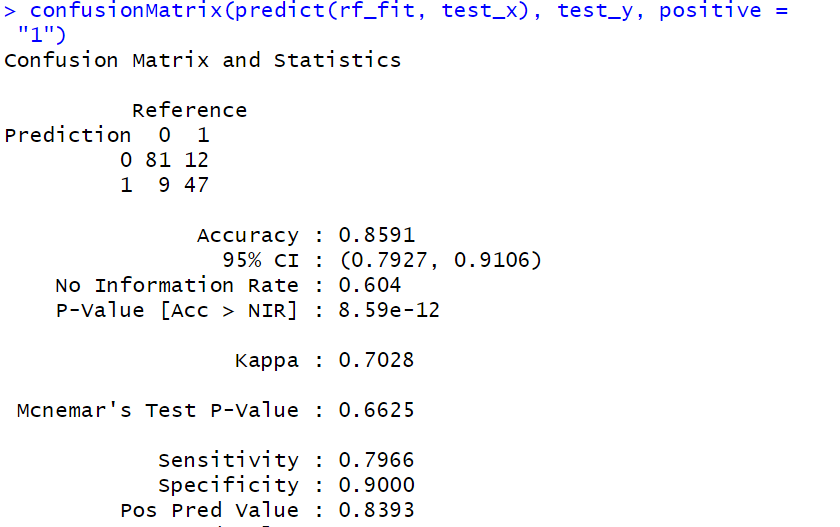
Sonuçlar, Bagging uygulanan tüm ağaçlarda X4 ,X8 ve X6 değişkenlerinin en önemli 3 değişken olduğunu göstermektedir.

TAHMINLEME:

Tahmin edilen değerler aşağıdaki gibidir.



Kurulan modelin Test hatası hesaplama işlemi aşağıdadır.

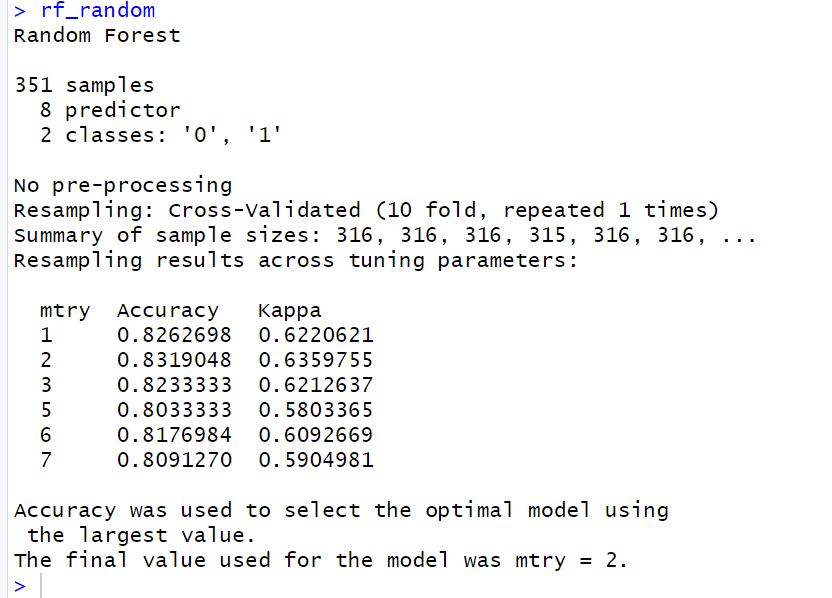


Test setimin doğruluk değeri %85 tir.Kappa değeri %70 dir.Kurulan modelin duyarlılığı %79 dur.

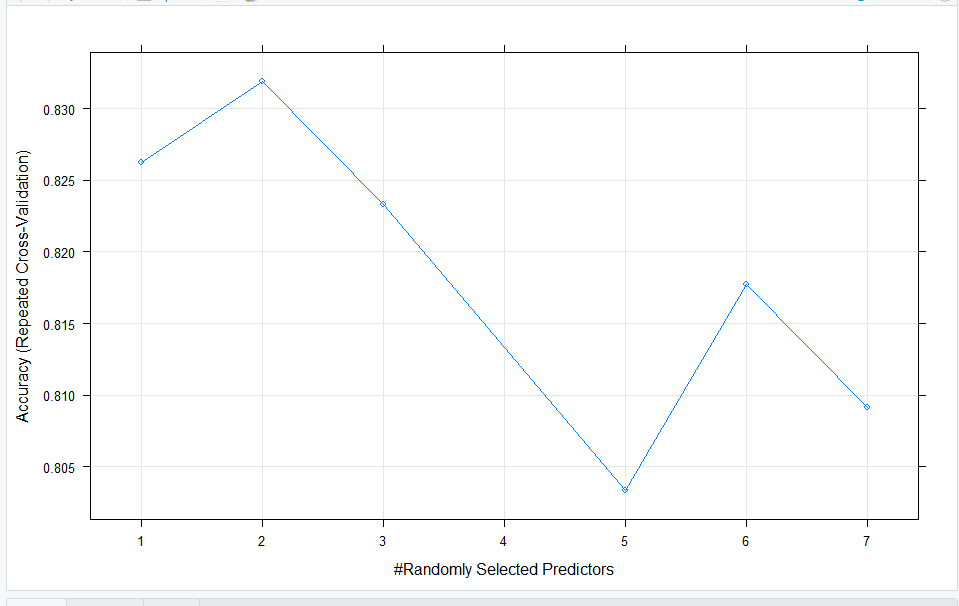
Random Forestte optimize edilmesi gereken iki parametre var.

1. Ağaç sayısı
2. Her bölünme işleminde rastegele seçilecek değişken sayısı .GRİD SEARCH ve RANDOM SEARCH methodu kullanılır.

RANDOM SEARCH ile optimizasyon:

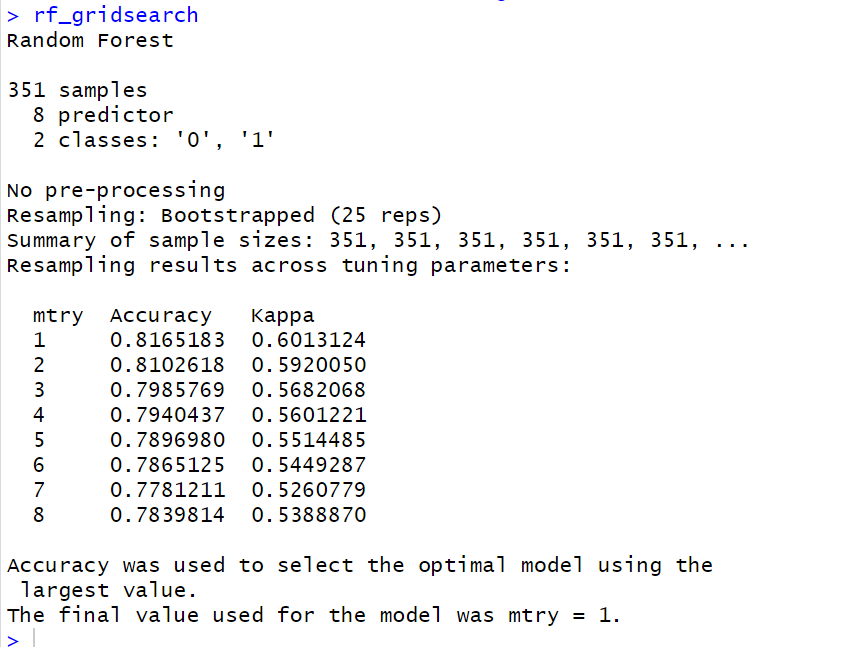


Her bölünmede rastgele değişken seçimimden sonra 2 tane kullanılması durumu en optimal durumdur.

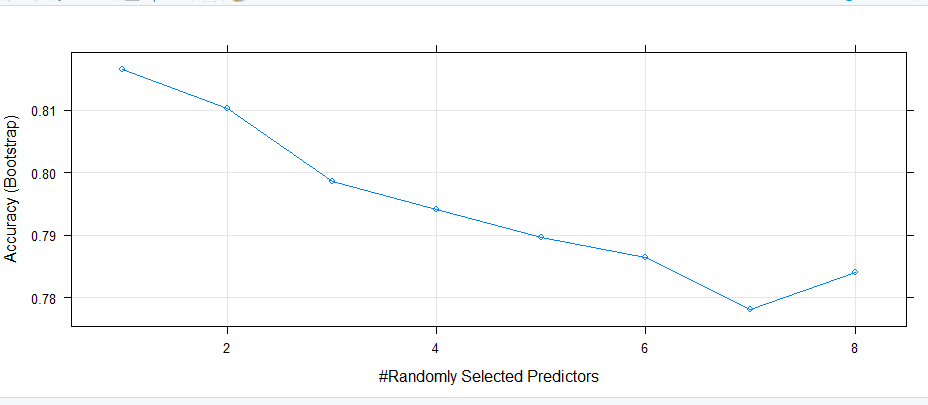


Rastgele seçilen tahminciler (bağımsız değişken) karşılık doğruluk değerinin nasıl değiştiği görülür. 2 değerim optimal değişkenimdir(X2).

GRID SEARCH OPTİMİZASYONU:



X1 değişkenim en optimal değişken çıktı.

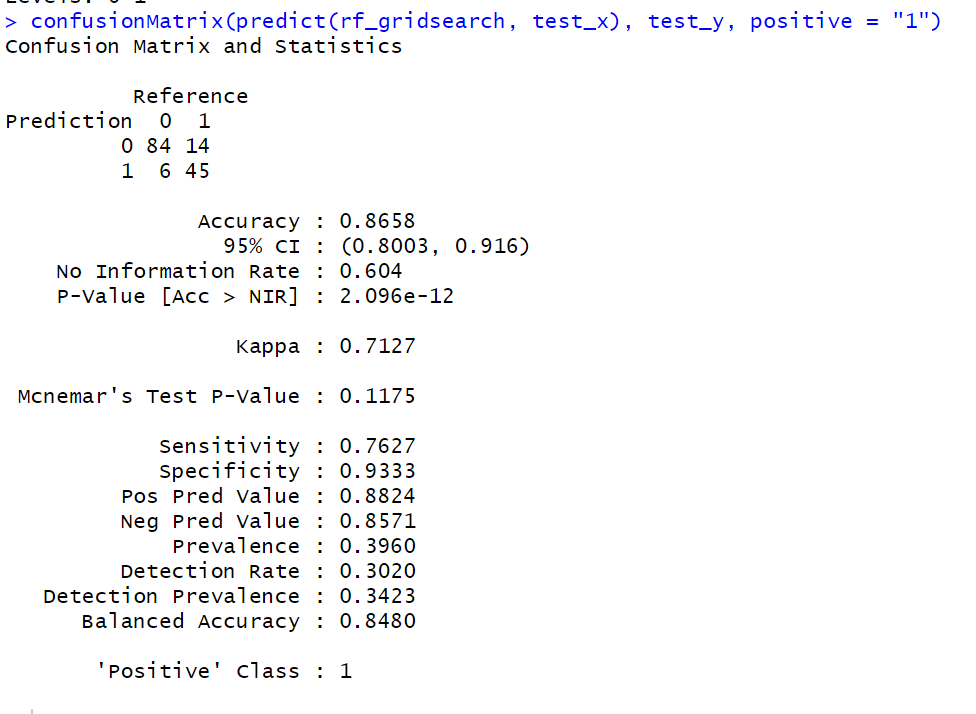


Rastgele seçilen tahminleyici de doğruluk oranını en yüksek veren x1 dir.

**TUNE EDİLEN MODELİN TESTİ:**

predict(rf\_gridsearch, test\_x)

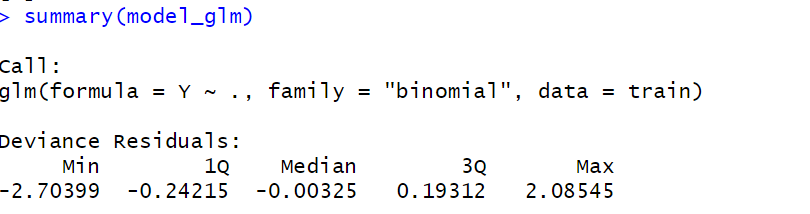
Kurulan model ile rf\_gridsearch test setinde ki bağımsız değişkenlerle tahmin işlemi yapılır. Ve bu tahmin değerlerini test\_y ile karşılaştırılır.

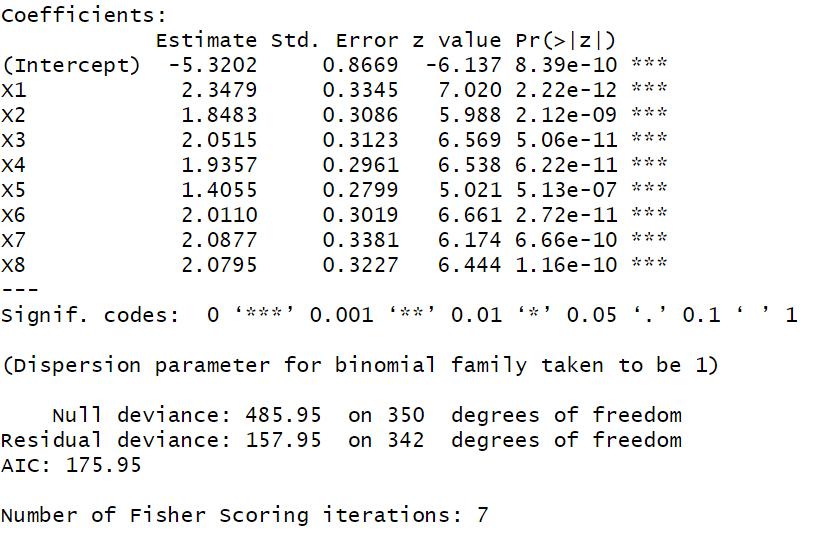


Gridsearche göre tahmin yapıp test hatası gözlemlendi.

Tune edilen modelin Doğruluk değeri %86dır ve kappa değeri %71 dir.Model optimizayonu yapılmadan önceki Accuracy değerim %85 ti ve Kappa değeri %70 ti.Kurulan modelin duyarlılığı %79 du tüne edilen modelin duyarlılığı %76 dır .Duyarlılık değerim azalmış.Doğruluk değerim %1 artmıştır.Özgüllük değeri model tune edilmeden önce %90 model\_tune edildikten sonra %93 yani %3 fark ile artmış.

**LOJİSTİC REGRESYON**

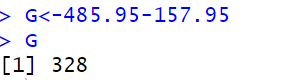
****

.

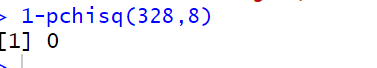
#null deviance: 485.95

#Residual deviance: residual devince:157.95

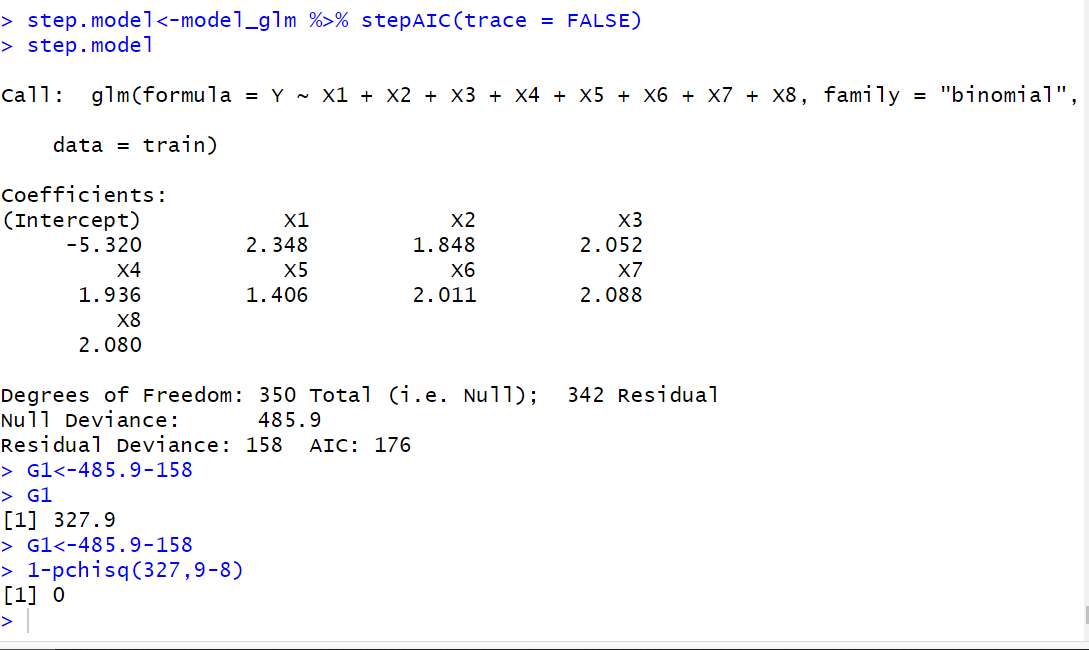
* p-value değerleri<0.05 ten küçük olduğundan bütün bağımsız değişkenler **%95 güvenle modelde anlamlıdır**.

0< alfa =0.05 ten küçük olduğundan H0 reddedilir. Değişken eklenmiş modelim anlamlıdır**.**

* *Bağımsız değişkenlerden en az birisinin Y bağımlı değişkenini istatistiksel olarak anlamlı bir şekilde etkilediği söylenebilir.*

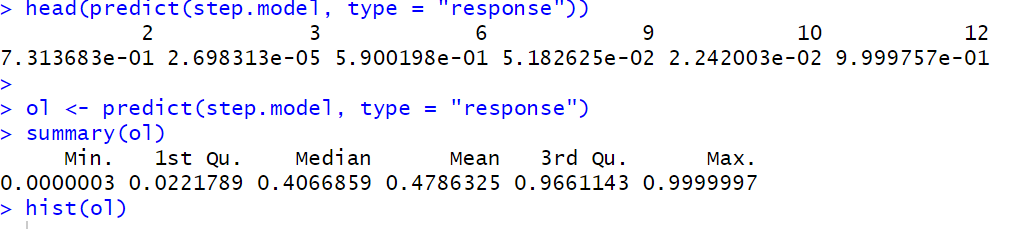


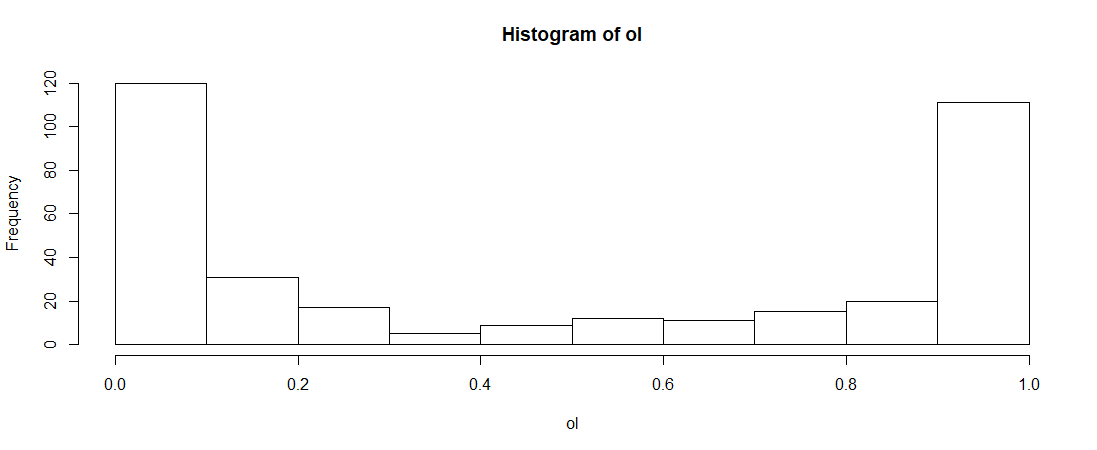
**Model TUNE**

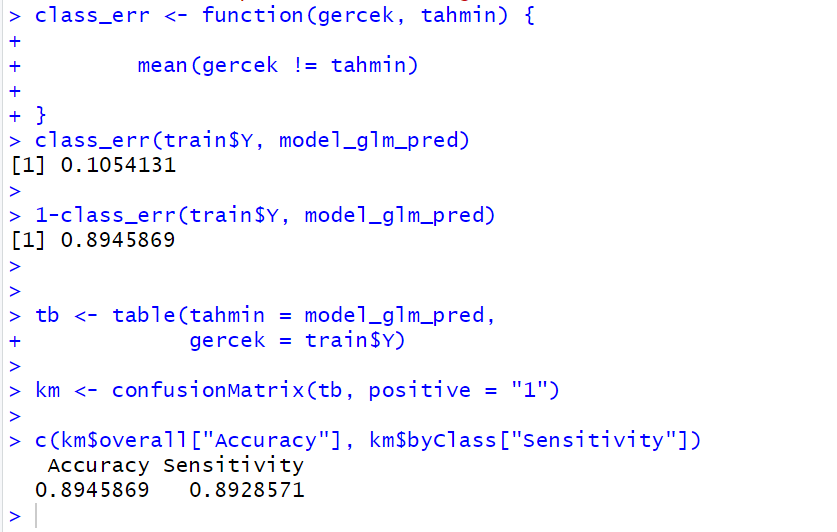


0<H0 olduğundan model anlamlıdır. Kurulan yeni modelim de bağımsız değişkenlerimden en az biri modelde anlamlıdır.

**TAHMIN DEĞERLERİM**





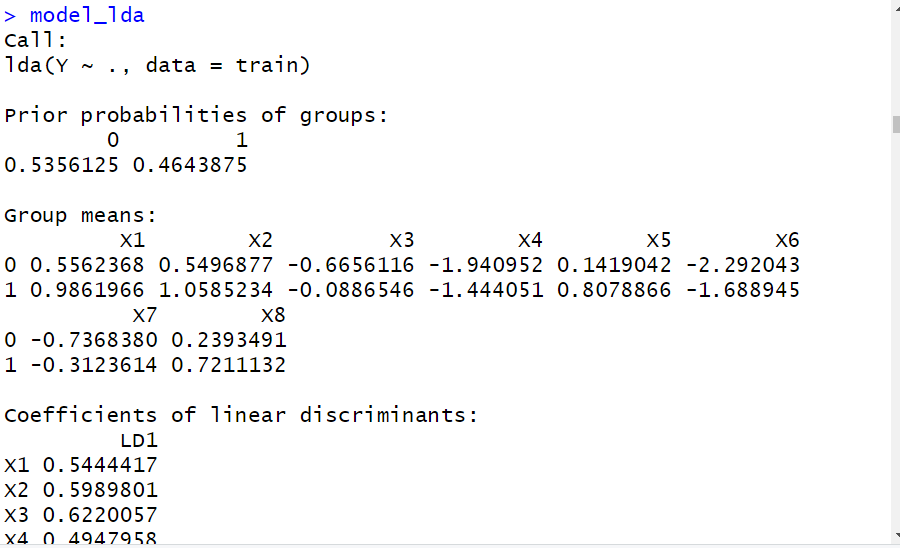


Modelin Sınıflandırma hatası %10 dur.Sonucu 1 olan bireylerin %89 u pozitif olarak sınıflandırılmıştır(sensivity)duyarlılık.

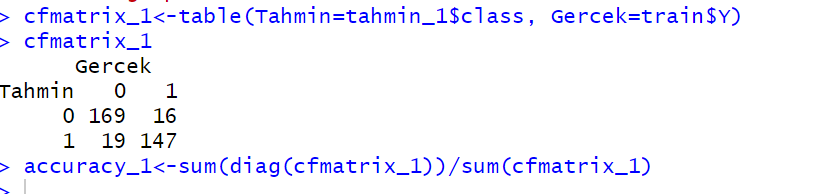
Modelin gözlemlerin doğru sınıflama oranı %89 durr.(accuracy)

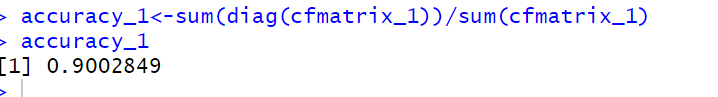
***LDA\_QDA***

**LINEAR DISCRIMINANT ANALYSIS**



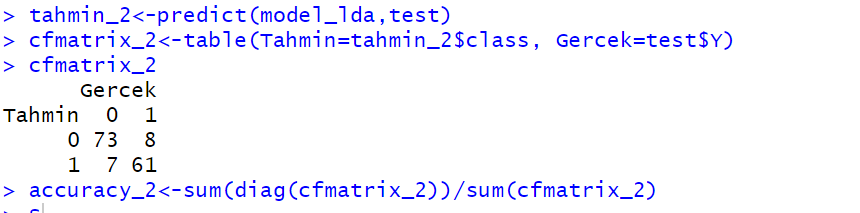
Trinin doğru sınıflandırma matrix:

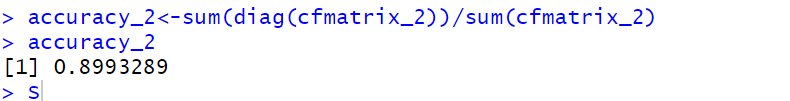


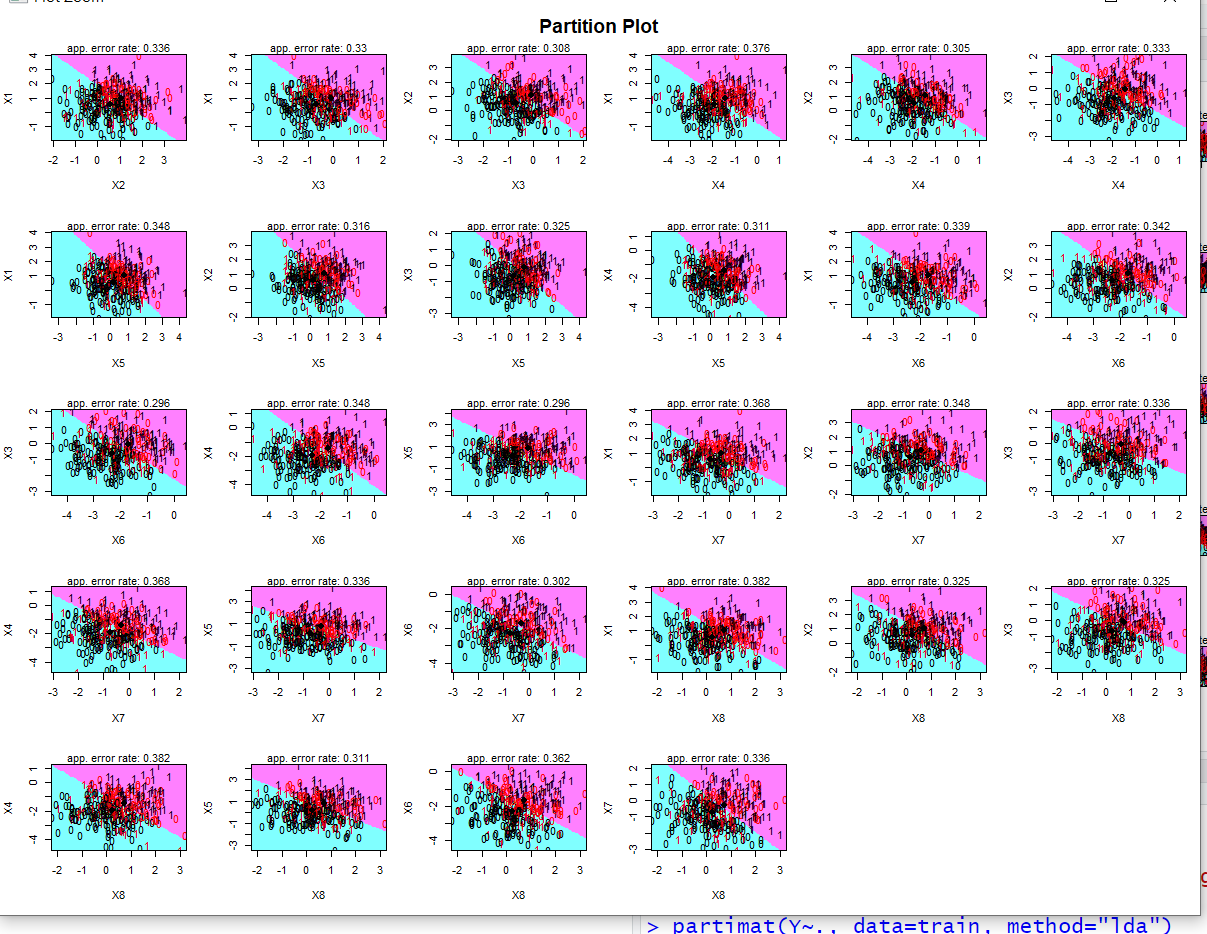


Train setinin doğruluk değeri %90 dır.

Test setinin model uygulandığında doğruluk değeri %89 dur .train ve test setleriin accuracy değerleri birbirne çok yakındır Kurulan LDA model başarılıdır.

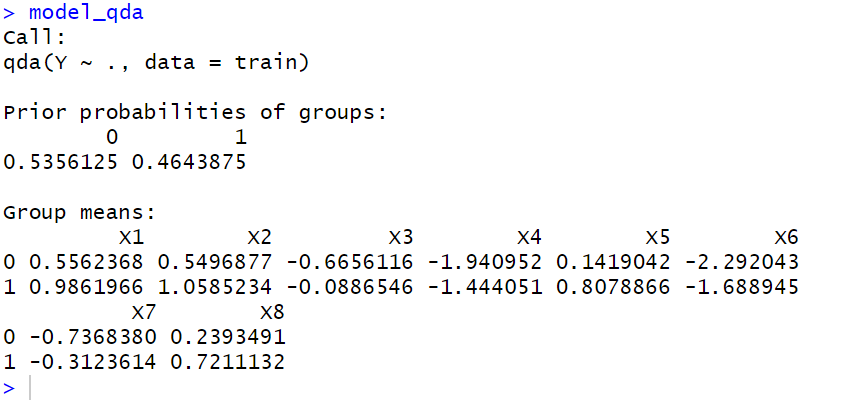




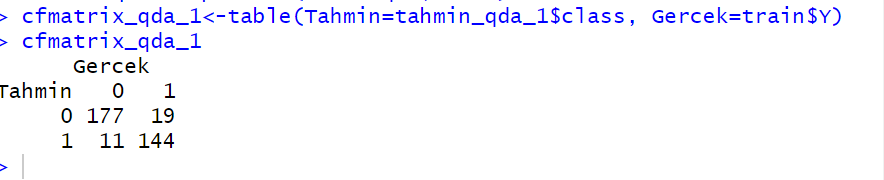


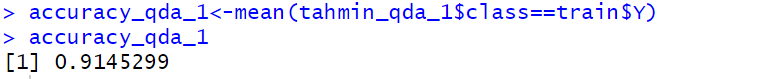
Her bir bağımsız değişkenin modelde diğer bağımsız değişkenle olan sınıflandırmasını sırasıyla gösterir.Yanlış sınıflandırma oranları bilgiside verilmektedir. Yanlış sınıflandırılan değişkenin gerçek değeri kırmız ile gösterilmiştir.

**QDA:**



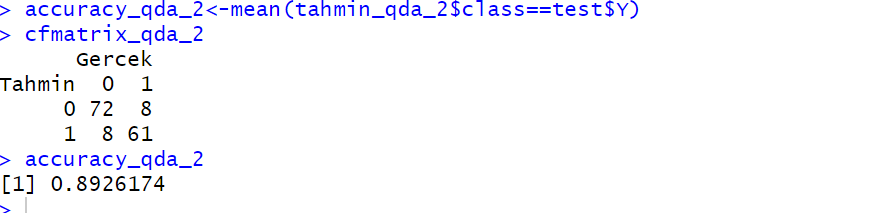
Eğitim verisinin doğruluk değeri ,sensivitiy,spesifit değerini aşağıdaki tablodan uluşabiliriz.



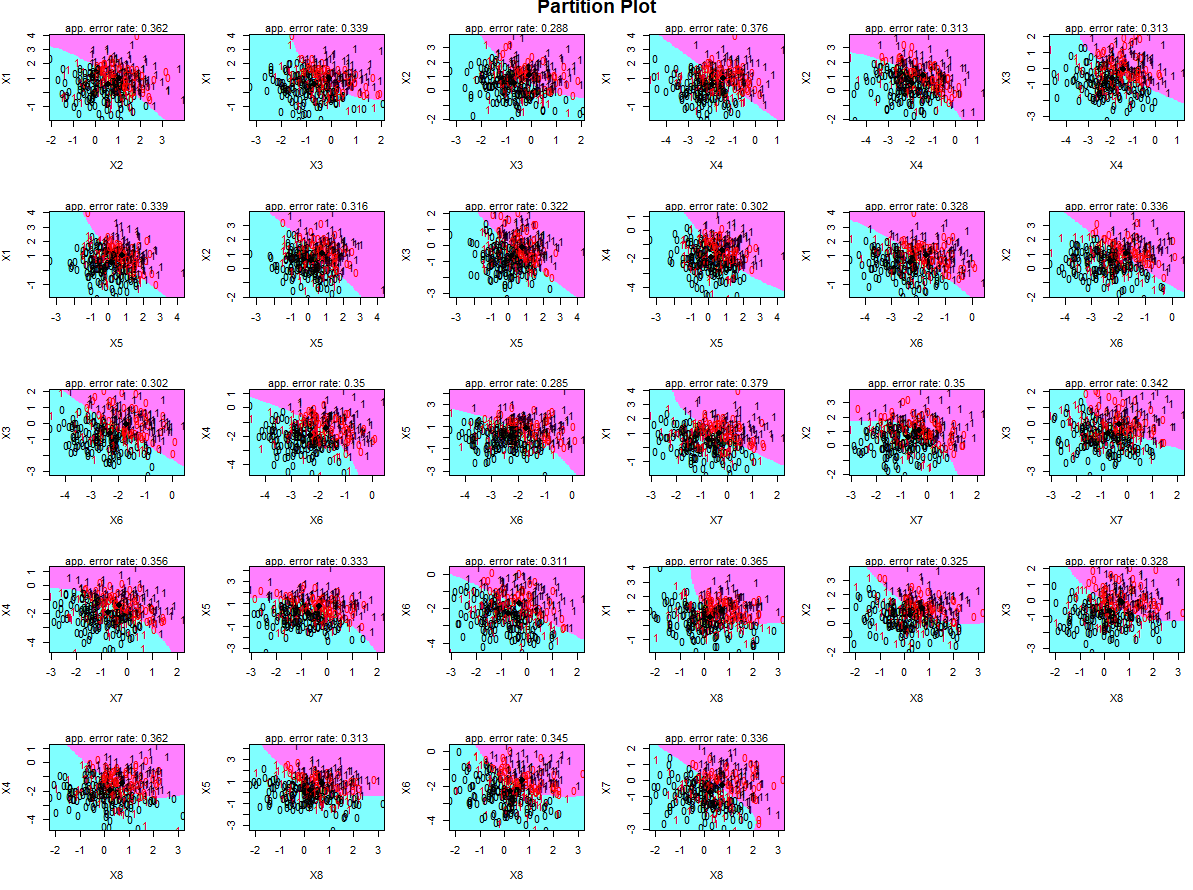


Train veri setitin modelde ki dopruluk değeri %91 dir.

Test verisi üzerinde model uygulandığında oluşacak karmaşıklık matrisi ve doğruluk değeri:

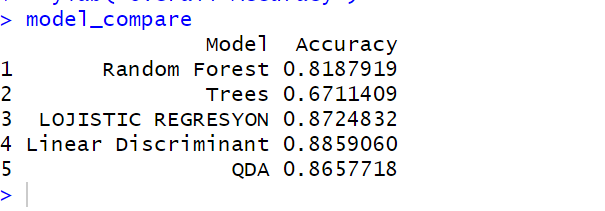


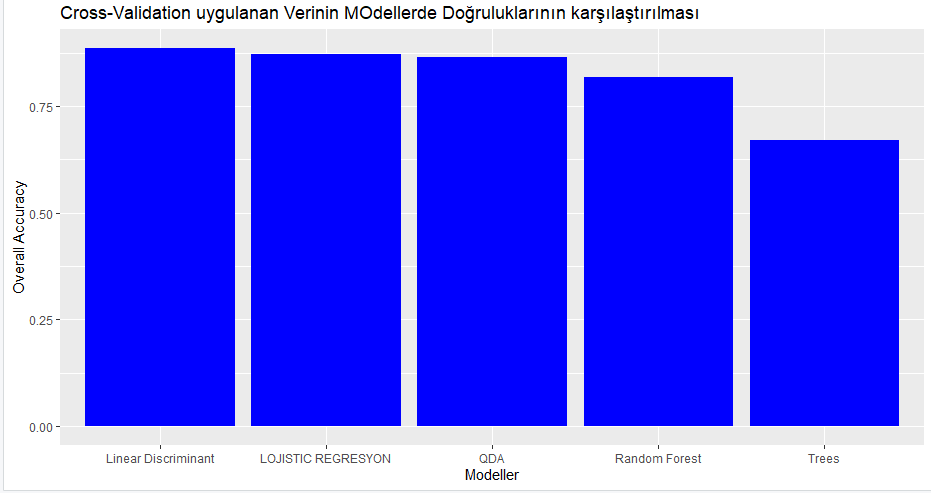
Kurulan modelin test verisindeki doğruluk değeri:%89(0.892617)dur.



Her bir bağımsız değişkenin modelde diğer bağımsız değişkenle olan sınıflandırmasını sırasıyla gösterir.Yanlış sınıflandırma oranlarıvbilgiside verilmektedir.Yanlış sınıflandırılan değişkenin gerçek değeri kırmız ile gösterilmiştir.

SORU2)



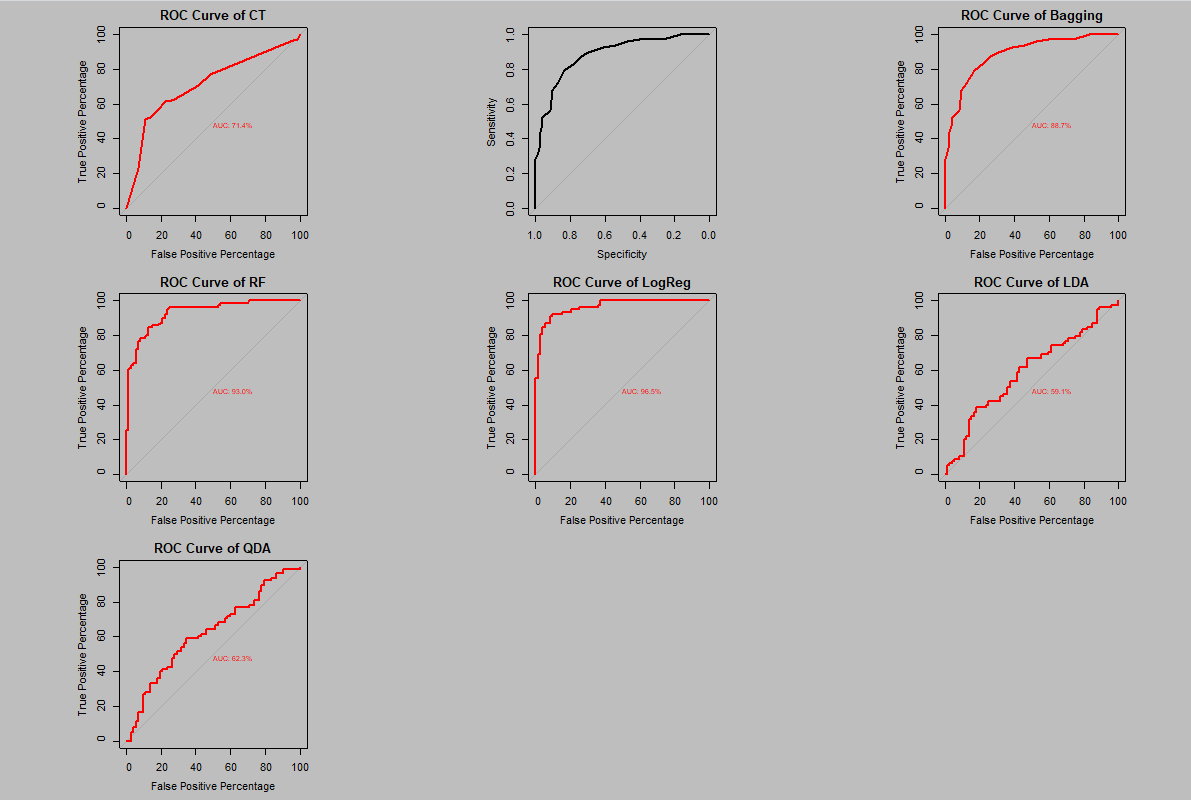


Modelin başarısı en yüksek Linerar DİScriminant %88 ikinci en yüksek Lojistic regresyon %87 dir.

Sınıflandırma yönetemlerinde Accuracy değerleri en yüksek ten en küçüğe sırasıyla:

* Linear Discreminant analysis
* Lojistic Regresyon
* QDA
* RANDOM Forest,
* Bagging(model doğruluğu % 78 dir.)
* Trees.

3)



ROC eğrisi, gerçek pozitif oranlar ve çeşitli eşiklerde hatalı pozitif oran arasındaki kontrastın grafiksel bir gösterimidir. Roc curve Duyarlılığa ne kadar yakınsa ve altında kalan alan ne kadar büyükse modelim o kadar güçlüdür.Kurduğum model de ROC curve AUC değerlerine göre en iyi sonucu **Lojistic regresyon** vermekte.En uygun model LOJİSTİC REGRESYONDUR. AUC değeri %94.3 tür.

