BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ

YBS461 - Veri Madenciliği

Doç. Dr. Nur Kuban TORUN

HAZIRLAYAN

Talha İŞLİYEN

KALP RAHATSIZLIĞI TAHMİNİ

2023-2024

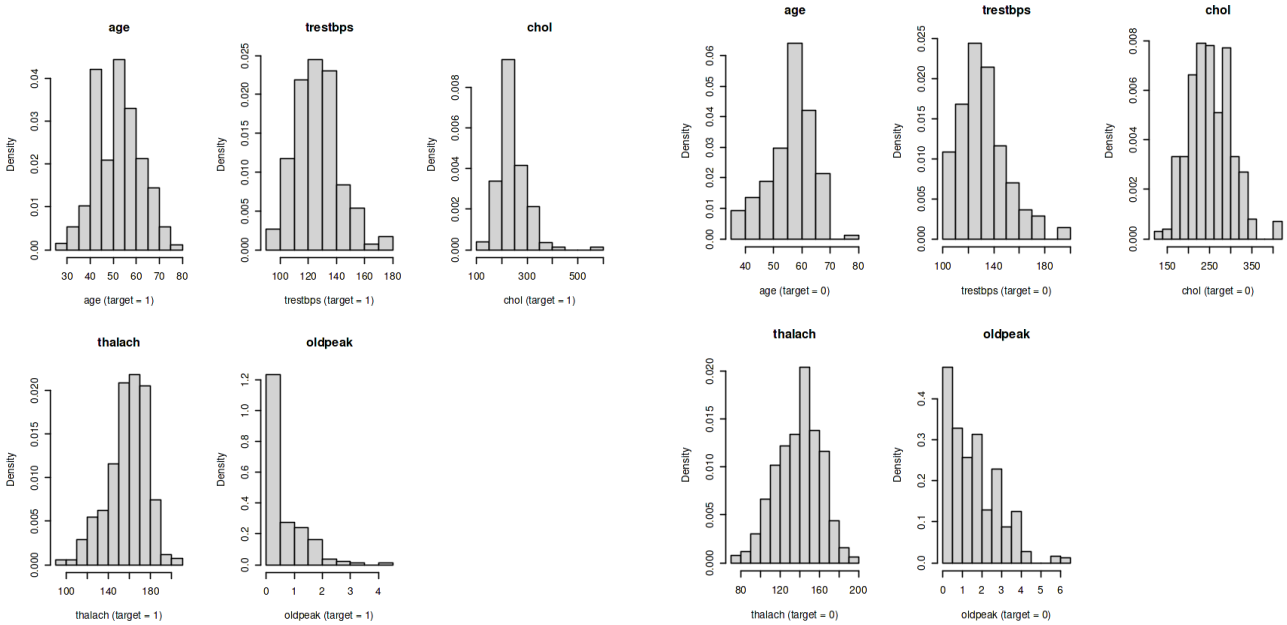
GÜZ DÖNEMİ

ÖZET

Kaggle'dan aldığım hazır bir kalp rahatsızlığı veri setini R dilinde analiz ettim. Veri setini kullanarak çeşitli tablolar oluşturarak, kalp rahatsızlıklarını tahmin etmeye yönelik bir çalışma gerçekleştirdim.

YÖNTEMLER

```
target0 = subset(heart, subset = (target == 0))
target1 = subset(heart, subset = (target == 1))

library(ggplot2)

par(mfrow=c(2,3))

hist(target0$age, freq=F, main = "age",
     xlab = "age (target = 0)", ylab = "Density")
hist(target0$trestbps, freq=F, main = "trestbps",
     xlab = "trestbps (target = 0)", ylab = "Density")
hist(target0$chol, freq=F, main = "chol",
     xlab = "chol (target = 0)", ylab = "Density")
hist(target0$thalach, freq=F, main = "thalach",
     xlab = "thalach (target = 0)", ylab = "Density")
hist(target0$oldpeak, freq=F, main = "oldpeak",
     xlab = "oldpeak (target = 0)", ylab = "Density")


par(mfrow=c(2,3)) # numeric var histogram

hist(target1$age, freq=F, main = "age",
     xlab = "age (target = 1)", ylab = "Density")
hist(target1$trestbps, freq=F, main = "trestbps",
     xlab = "trestbps (target = 1)", ylab = "Density")
hist(target1$chol, freq=F, main = "chol",
     xlab = "chol (target = 1)", ylab = "Density")
hist(target1$thalach, freq=F, main = "thalach",
     xlab = "thalach (target = 1)", ylab = "Density")
hist(target1$oldpeak, freq=F, main = "oldpeak",
     xlab = "oldpeak (target = 1)", ylab = "Density")
```

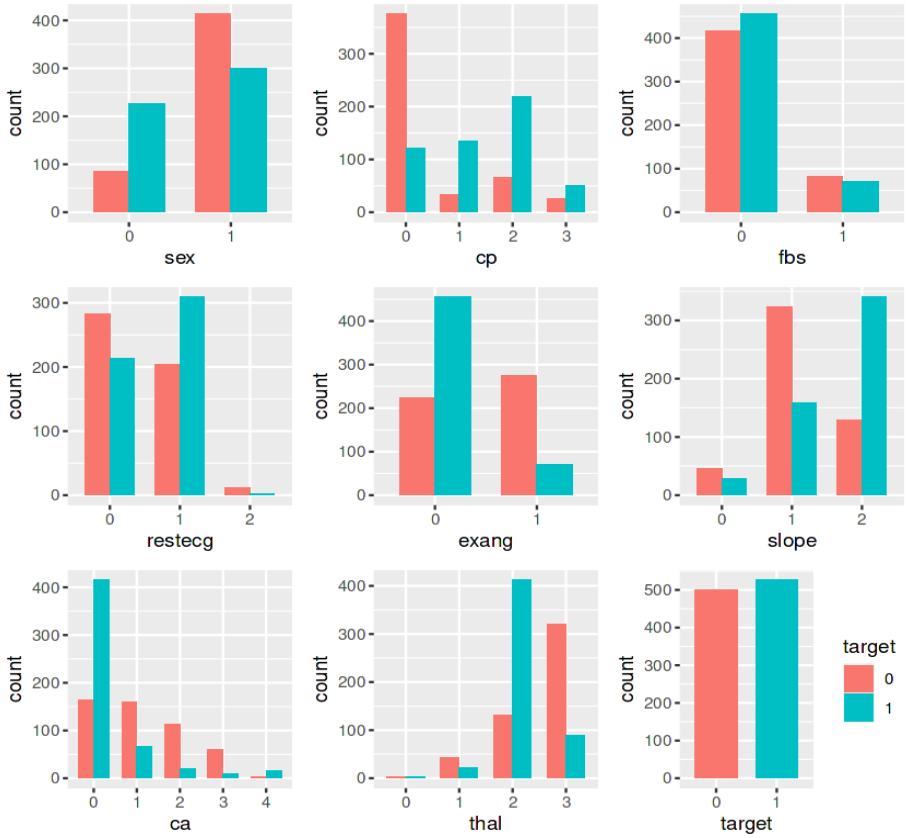


"Ggplot2" R programlama dilinde bulunan bir grafik oluşturma paketidir. Bu paketle oluşturduğum tablolar yukarıdadır;

“gridExtra”, R programlama dilindeki grid paketinin bir parçasıdır. Bu paket, grid sistemi üzerinde çoklu grafikleri düzenlemek ve bir araya getirmek için kullanılır. “gridExtra” paketi, “grid” paketinin bazı özelliklerini geliştirerek kullanımı daha kolay hale getirir. Bu paketi kullanarak da bir tablo oluşturdum.

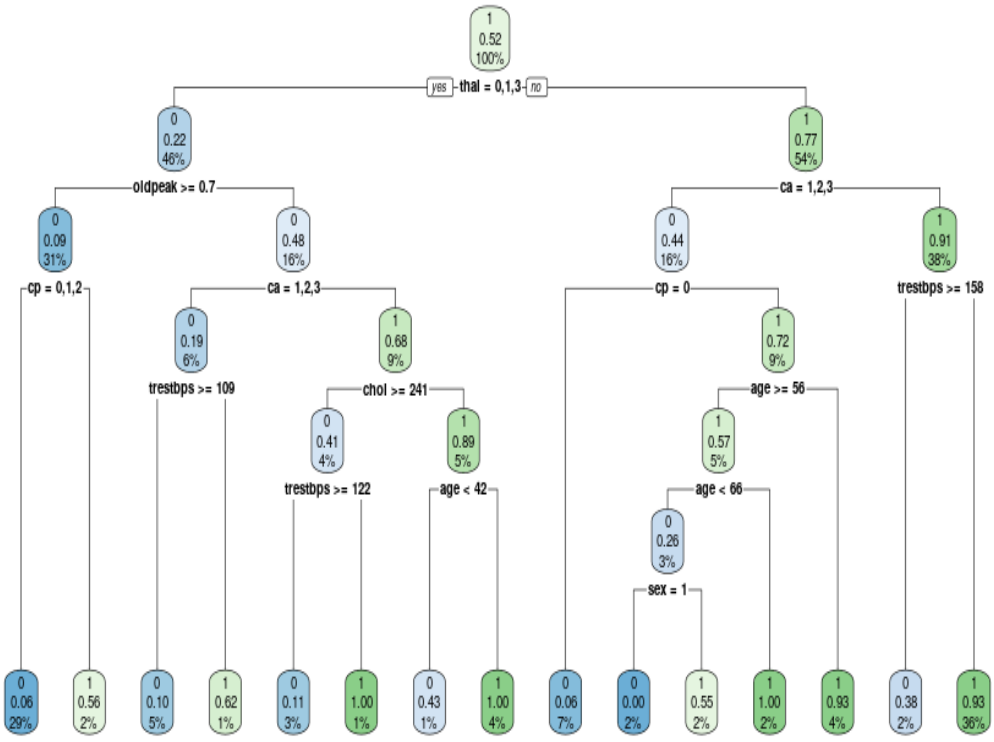
```
library(gridExtra)

# sex, cp, fbs, restecg, exang, slope, ca, thal, target
a = ggplot(heart, aes(x=factor(sex), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="sex", y = "count") +
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

b = ggplot(heart, aes(x=factor(cp), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="cp", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

c = ggplot(heart, aes(x=factor(fbs), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="fbs", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

d = ggplot(heart, aes(x=factor(restecg), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="restecg", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

e = ggplot(heart, aes(x=factor(exang), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="exang", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

f = ggplot(heart, aes(x=factor(slope), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="slope", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

g = ggplot(heart, aes(x=factor(ca), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="ca", y = "count")+
  theme(legend.title = element_blank()) +
```

```
  theme(legend.position = 'none')

h = ggplot(heart, aes(x=factor(thal), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="thal", y = "count")+
  theme(legend.title = element_blank()) +
  theme(legend.position = 'none')

i = ggplot(heart, aes(x=factor(target), fill=target))+
  geom_bar(stat="count", width=0.7, position=position_dodge()) +
  labs(x="target", y = "count")

grid.arrange(a,b,c,d,e,f,g,h, i, nrow=3, ncol=3)
```

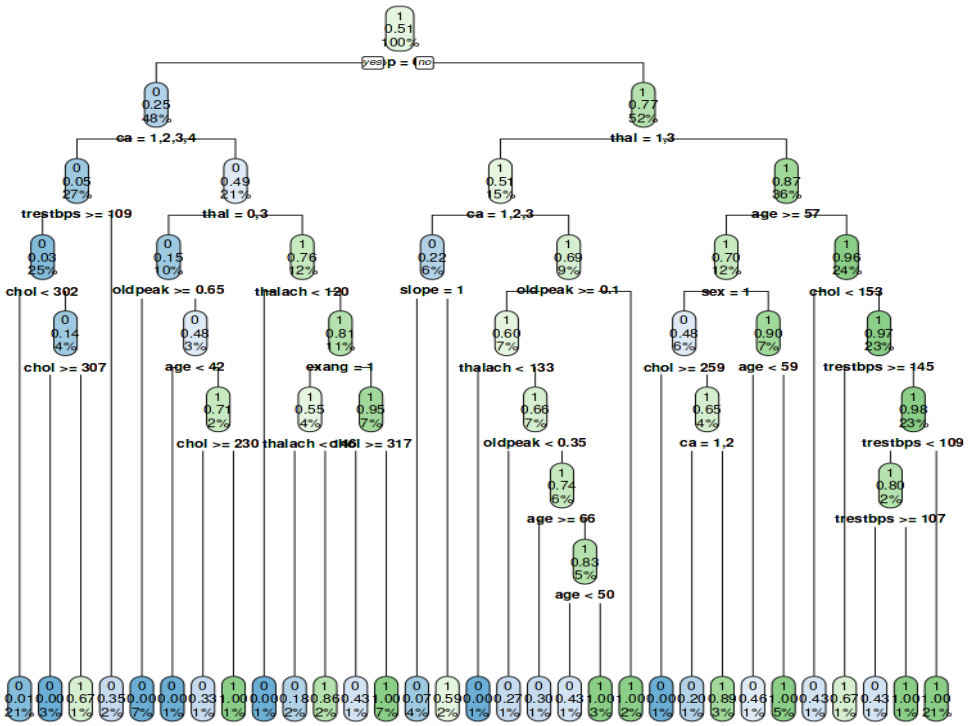


“ROCR”, R programlama dilinde ROC eğrisi analizi ve performans değerlendirmesi için kullanılan bir pakettir. ROC (Receiver Operating Characteristic) eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir grafiksel yöntemdir. Bu paketi kullanarak analizimi daha detaylandırdım.

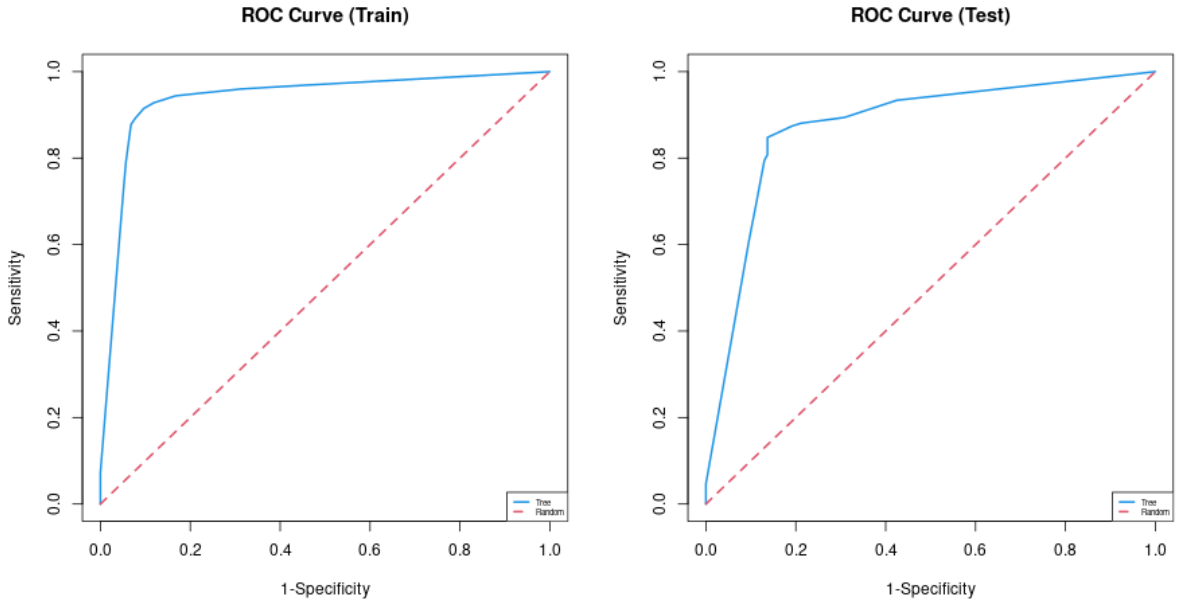
```
library(ROCR) #install.packages("ROCR")

pred2 = predict(fit.pruned, newdata=heart, type="prob") #prediction
pred = prediction(pred2[,2], heart$target)
perf = performance(pred, "tpr","fpr")

plot(perf, col = 4, lwd = 2, xlab = "1-Specificity", ylab = "Sensitivit
y", main = "ROC Curve") #ROC
lines(x = c(0,1), y = c(0,1), col = 2, lty = 2, lwd = 2)
legend("bottomright", legend = c("Tree","Random"), col = c(4,2), lty =
c(1,2), lwd = 2)

performance(pred, "auc")
```



Bu durumda, 499 gözlem yanlış sınıflandırılmış ve hata oranı 0.48683 olarak hesaplanmış.



0.927804878048781 Doğruluk oranı, toplam doğru sınıflandırılan gözlemlerin toplam gözlem sayısına oranıdır. 0.0721951219512195 Hata oranı, toplam yanlış sınıflandırılan gözlemlerin toplam gözlem sayısına oranıdır. 1 sınıfı için hassasiyet 0.9258 iken, 0 sınıfı için hassasiyet 0.9299'dur.



```
eart0 <- heart_dummies

heart0x <- select(heart0, -target)
heart0y <- select(heart0, target)

# minmax Scaling
max1 = apply(heart0x, 2, max)
min1 = apply(heart0x, 2, min)

gdat = scale(heart0x, center = min1, scale = max1 - min1)
gdat = as.data.frame(cbind(heart0x, heart0y))

heart0 = gdat

X <- select(heart0, -target)
# y <- select(heart0, target)
y <- heart0[ ,31]
```

In [6]:

linkcode

```
## Data partitioning
set.seed(123)
V = 2
n =  NROW(heart0)

id = sample(1:V, n, prob = c(0.7,0.3), replace = T) # Partitioning 7:3
ii = which(id==1)
X.train = X[ii,]; X.test  = X[-ii,]
y.train = y[ii];  y.test  = y[-ii]

# Modeling
knn.5 <- knn(train=X.train, test=X.test, cl=y.train, k=5)
knn.10 <- knn(train=X.train, test=X.test, cl=y.train, k=10)

ctable.5 = table(knn.5, y.test)
ctable.10 = table(knn.10, y.test)

confusionMatrix(ctable.5)
confusionMatrix(ctable.10)
```

Bu çıktılar, iki farklı K-NN (K-Nearest Neighbors) modelinin performansını değerlendirmek için kullanıla n karmaşıklık matrislerini ve çeşitli sınıflandırma istatistiklerini içerir. İki farklı K-NN modeli için ayrı ayr ı değerlendirme sonuçları vardır, birincisi K=5 için yapılan değerlendirme, ikincisi ise K=10 için yapılan değerlendirmedir.

K=5 İçin Değerlendirme:

y.test

knn.5 0 1

0 109 68

1 34 90

İstatistikler:

- Accuracy (Doğruluk): 0.6611
- Kappa: 0.3282
- Sensitivity (Hassasiyet): 0.7622
- Specificity (Özgüllük): 0.5696
- Pozitif Tahmin Değer Oranı: 0.6158
- Negatif Tahmin Değer Oranı: 0.7258

K=10 İçin Değerlendirme:

y.test

knn.10 0 1

0 110 42

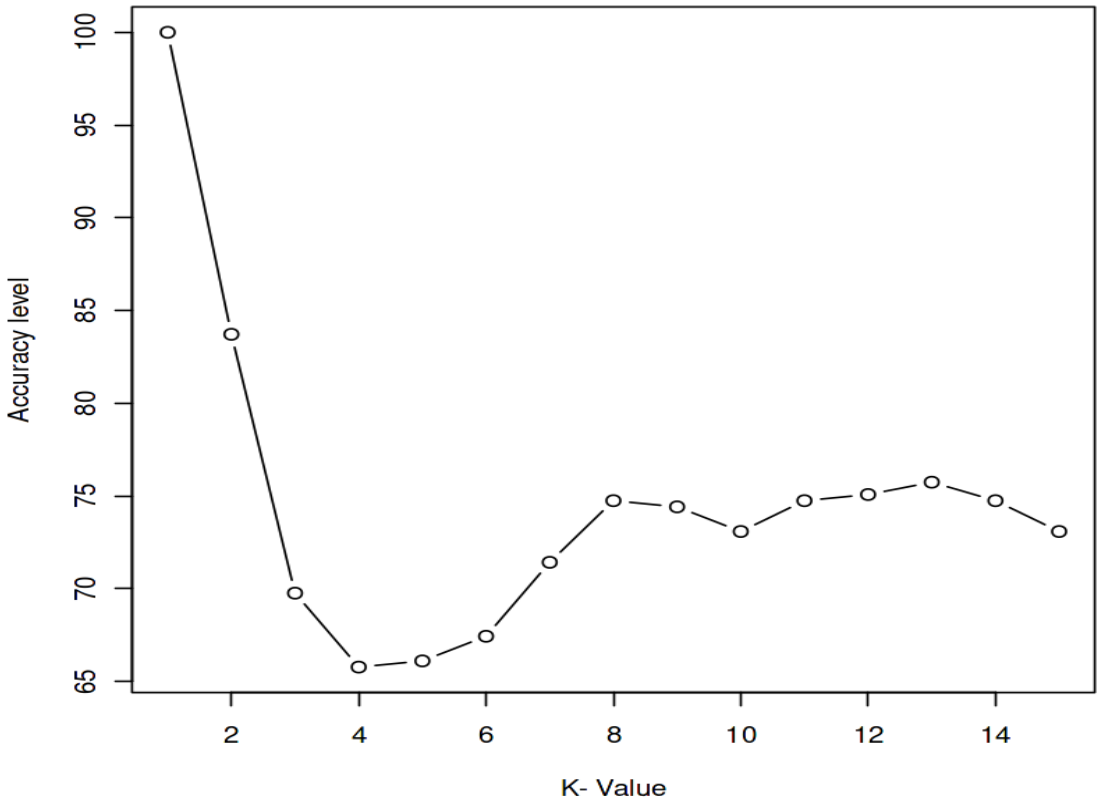
1 33 116

İstatistikler:

- Accuracy (Doğruluk): 0.7508
- Kappa: 0.5019
- Sensitivity (Hassasiyet): 0.7692
- Specificity (Özgüllük): 0.7342
- Pozitif Tahmin Değer Oranı: 0.7237
- Negatif Tahmin Değer Oranı: 0.7785

Her iki durumda da, modelin performansını değerlendirmek için kullanılan çeşitli istatistikler bulunmaktadır. "0" ve "1" sınıfları arasındaki performans, modelin doğruluğu, hassasiyeti, özgüllüğü gibi ölçütlerle değerlendirilmiştir.

```
i=1
k.optm=1
for (i in 1:15){
  knn.mod <- knn(train=X.train, test=X.test, cl=y.train, k=i)
  k.optm[i] <- 100 * sum(y.test == knn.mod)/NROW(y.test)
  k=i
  cat(k,'=',k.optm[i],'
')
}
plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")
```



```r
set.seed(123)
V = 2
n =  NROW(heart0)

id = sample(1:V, n, prob = c(0.7,0.3), replace = T) # Partitioning 7:3
ii = which(id==1)
X.train = X[ii,]; X.test  = X[-ii,]
y.train = y[ii];  y.test  = y[-ii]

#X.train = X.train[, c("cp_0", "sex_0")]; X.test  = X.test[, c("cp_0", "s
ex_0")]
#y.train = y[ii];  y.test  = y[-ii]

vars = c("cp_0", "sex_0", "thal_2", "slope_1", "ca_2")
X.train = X.train[, vars]; X.test  = X.test[, vars]
y.train = y[ii];  y.test  = y[-ii]

## Optimization
i=1
k.optm=1
for (i in 1:15){
  knn.mod <- knn(train=X.train, test=X.test, cl=y.train, k=i)
  k.optm[i] <- 100 * sum(y.test == knn.mod)/NROW(y.test)
  k=i
  cat(k,'=',k.optm[i],'
')
}

#Accuracy plot
plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")
```