Veri Madenciliği Dersi Ödevi

DENİZ SERBES,ELİF ZÜLAL ÇAVDAROĞLU,DİDEM TAŞPINAR,TUĞÇE YÜNCÜOĞLU

27 11 2021

Amaç: winequality-red verileri ordinal kategorik veriye dönüştürülerek karar ağacı algoritmalarından sınıflandırma algoritması ile bağımlı değişkenin ne kadarlık bir doğrulukla tahmin edileceğini ortaya koymak amaçlanmıştır. Burada, şarap kalitesi (quality) kategorik veriye dönüştürülerek hedef değişken olarak belirlenmiştir.

Gerekli kütüphaneleri indirelim.

```
library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(ggpubr)

str(winequality_red)

## spec_tbl_df [1,599 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8
7.5 ...
##  $ volatile acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65
0.58 0.5 ...
##  $ citric acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36
...
##  $ residual sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2
6.1 ...
##  $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075
0.069 0.065 0.073 0.071 ...
##  $ free sulfur dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
##  $ total sulfur dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3
3.39 3.36 3.35 ...
##  $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46
```

```
0.47 0.57 0.8 ...
##  $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5
10.5 ...
##  $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `fixed acidity` = col_double(),
##   ..   `volatile acidity` = col_double(),
##   ..   `citric acid` = col_double(),
##   ..   `residual sugar` = col_double(),
##   ..   chlorides = col_double(),
##   ..   `free sulfur dioxide` = col_double(),
##   ..   `total sulfur dioxide` = col_double(),
##   ..   density = col_double(),
##   ..   pH = col_double(),
##   ..   sulphates = col_double(),
##   ..   alcohol = col_double(),
##   ..   quality = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

verimiz 1599 gözlem 12 değişkenden oluşmaktadır.

Missing Data (NA) olan değerler veri setinden çıkarılmıştır.

NA değerleri çıkarıldıktan sonraki gözlem değerleri gözden geçirilmiştir.

```
df<-winequality_red %>% drop_na()
str(df)
```

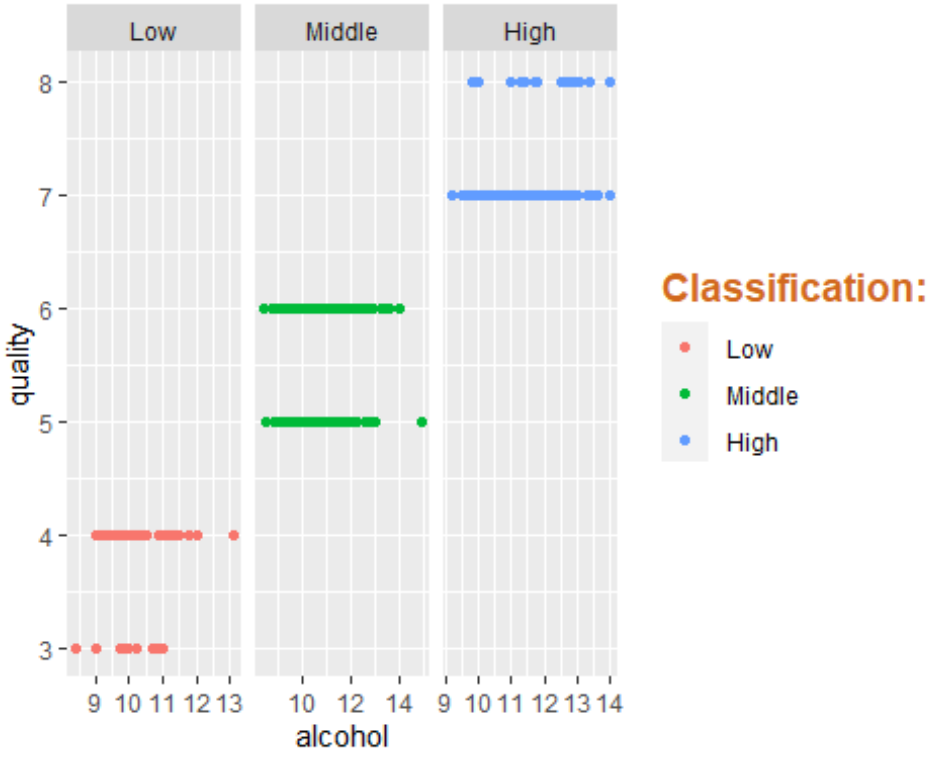
```
## tibble [1,599 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8
7.5 ...
##  $ volatile acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65
0.58 0.5 ...
##  $ citric acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36
...
##  $ residual sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2
6.1 ...
##  $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075
0.069 0.065 0.073 0.071 ...
##  $ free sulfur dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
##  $ total sulfur dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3
3.39 3.36 3.35 ...
##  $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46
```

```
0.47 0.57 0.8 ...
##  $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5
10.5 ...
##  $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...

library(tibble)#esnek ve gelişmiş tablo oluşturmak için
df_20<-cut(winequality_red$quality, breaks = 3, labels = c("Low", "Middle",
"High"))#kalite verisinin kategorize edilerek ordinal veri türüne
dönüştürülmesi için

library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

describe(df_20)#Kategorize edilen veriye ait tanımlayıcı istatistikleri
vermek için

## df_20
##        n  missing distinct
##     1599        0        3
##
## Value         Low Middle   High
## Frequency      63   1319    217
## Proportion  0.039  0.825  0.136

t<-tibble(residualsugar=df$`residual sugar`, alcohol=df$alcohol,
quality=df$quality, Classification=df_20)#Gelişmiş ve esnek tablo formatı
içine analiz kapsamında değerlendirilecek değişkenler ve bunlara ait veri
seti alınmıştır.

head(t,20)# Veri setinin ilk 20 satırını vermek için

## # A tibble: 20 x 4
##    residualsugar alcohol quality Classification
##            <dbl>   <dbl>   <dbl> <fct>
##  1           1.9     9.4       5 Middle
##  2           2.6     9.8       5 Middle
##  3           2.3     9.8       5 Middle
```

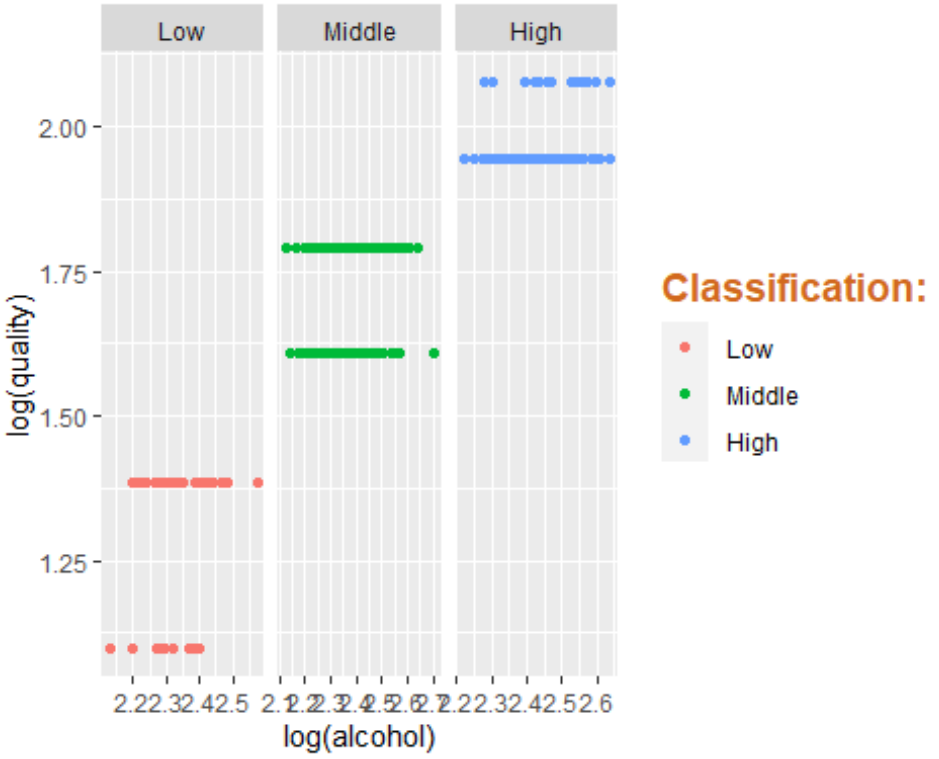
```
##  4           1.9     9.8       6 Middle
##  5           1.9     9.4       5 Middle
##  6           1.8     9.4       5 Middle
##  7           1.6     9.4       5 Middle
##  8           1.2    10         7 High
##  9           2       9.5       7 High
## 10           6.1    10.5       5 Middle
## 11           1.8     9.2       5 Middle
## 12           6.1    10.5       5 Middle
## 13           1.6     9.9       5 Middle
## 14           1.6     9.1       5 Middle
## 15           3.8     9.2       5 Middle
## 16           3.9     9.2       5 Middle
## 17           1.8    10.5       7 High
## 18           1.7     9.3       5 Middle
## 19           4.4     9         4 Low
## 20           1.8     9.2       6 Middle
```

Değişkenler arasındaki ilişkiler temelinde log dönüşümü de uygulanarak karşılaştırmalı olarak çoklu saçılım grafikleriyle verilmiştir.

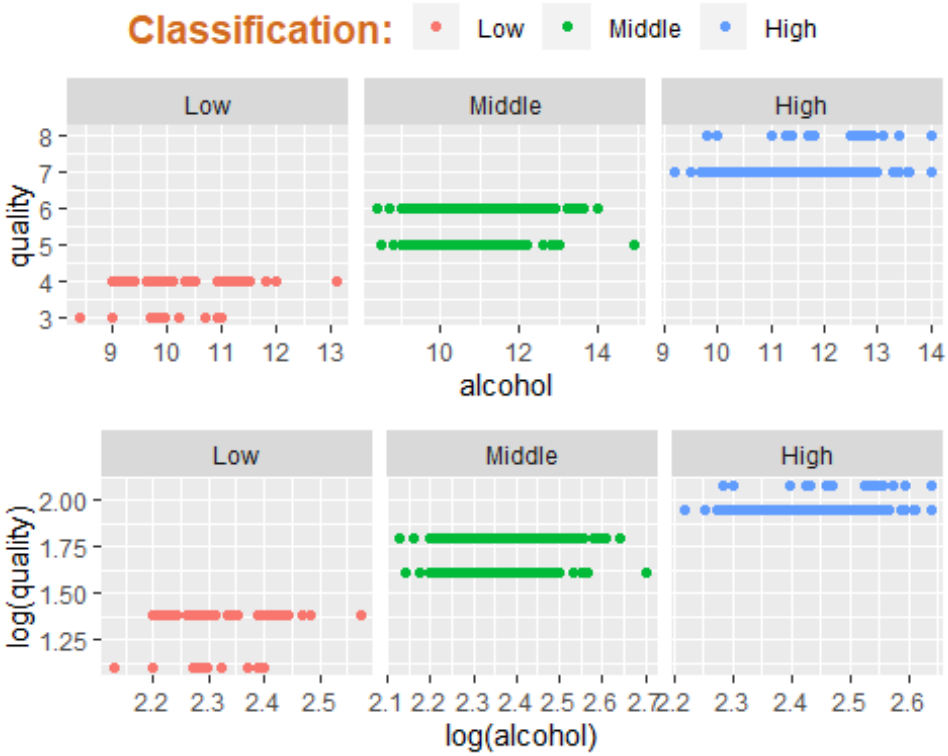
```
#Değişkenler arasındaki ilişkiler işsizlik oranı sınıflandırılması temelinde
çoklu saçılım grafiklerle verilmiştir.
v1<-t%>%ggplot(aes(x=alcohol, y=quality, color =
    factor(Classification)))+
    geom_point()+
    facet_grid(. ~ Classification, scales = "free")+
    labs(color = "Classification:") +
    theme(legend.title = element_text(color = "chocolate",
                                      size = 14, face = "bold"))
v1
```




```
v2<-t%>%ggplot(aes(x=log(alcohol), y=log(quality), color =
    factor(Classification)))+
    geom_point()+
    facet_grid(. ~ Classification, scales = "free")+
    labs(color = "Classification:") +
    theme(legend.title = element_text(color = "chocolate",
                                      size = 14, face = "bold"))
v2
```




```
ggarrange(v1, v2, ncol = 1, nrow = 2, common.legend = TRUE)
```



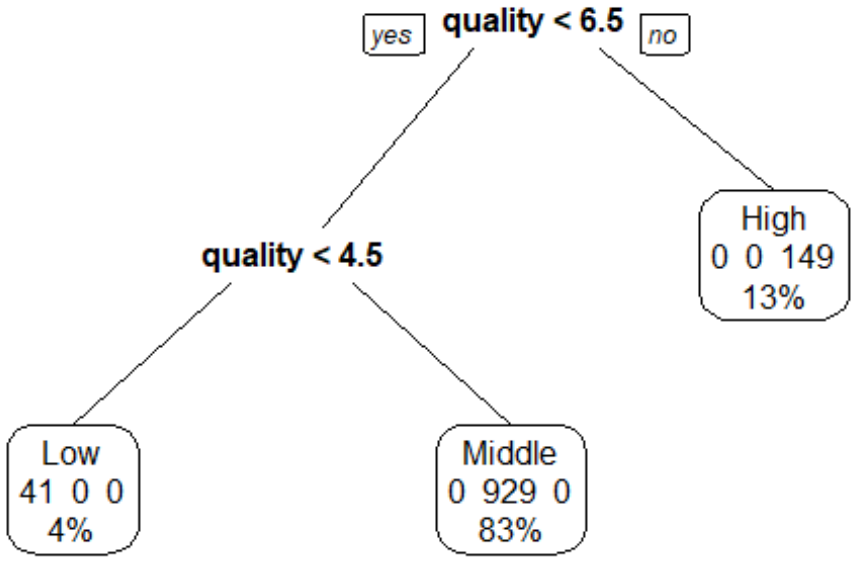
kod bloklarının çalıştırılmasıyla elde edilen saçılım grafiklerden değişken veri setlerine logaritmik (log) dönüşüm uygulandığında değişkenler arasında pozitif artış olduğu görülmektedir.verinin

temizlenmesi ve değişkenler arasındaki ilişkilerin gösterimi adı altında veri madenciliği işlemleri yapılmıştır. Burada veri madenciliği konu alanı altında pek çok diğer işlem de yapılabilir. Bu durum verinin türüne, kaynağına, analizin ve araştırmanın amacına göre farklılık göstermektedir. Bu aşamadan sonra ise analize uygun hale getirilen veri yine veri madenciliği algoritmalarından olan karar ağacı algoritması kullanılarak analiz edilecektir.

Karar Ağacı Oluşturma-1

Veri seti basit tesadüfi örneklem çekilerek eğitilecek ve test edilecek veri setine ayrılır. Daha sonra veri setindeki "Classification" değişkeni hedef değişken belirlenerek modele sokulmuştur. Elde edilen sonuçlar karar ağacında görselleştirildikten sonra hata ve doğruluk oranları verilmiştir.

```
library(DMwR2)#Karar ağacı kütüphanesini yüklemek için

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

set.seed(1000)
n=NROW(t)
n

## [1] 1599

SRS <- sample(sample(1:n, size = round(0.7*n), replace=FALSE))#veri setinden
basit tesadüfi örneklem çekilerek eğitilecek veri seti ve test setine ayırmak
için

training <- t[SRS, ]#eğitilecek veri seti
test <- t[-SRS, ]#test edilecek veri seti
model <- rpartXse(Classification ~ ., training, se=0.5)

pred1 <- predict(model, test, type="class")
head(pred1)

##      1      2      3      4      5      6
## Middle Middle Middle Middle Middle   High
## Levels: Low Middle High

library(rpart.plot)#karar ağacının görselleştirilmesi için

## Loading required package: rpart

prp(model, type=0, extra=101)

## Warning: Cannot retrieve the data used to build the model (so cannot
determine roundint and is.binary for the variables).
## To silence this warning:
##     Call prp with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
(cm <- table(pred1, test$Classification))
```

```
##
## pred1    Low Middle High
##   Low     22      0    0
##   Middle   0    390    0
##   High     0      0   68
```

```
error_rate<-100*(1-sum(diag(cm))/sum(cm))
error_rate
```

```
## [1] 0
```

```
print(paste('Error Rate is %', round(error_rate))) # the error rate (hata
oranı)
```

```
## [1] "Error Rate is % 0"
```

```
accuracy_test <- sum(diag(cm)) / sum(cm)
print(paste('Accuracy for test is %', 100*accuracy_test))#Accuracy
rate(doğruluk oranı)
```

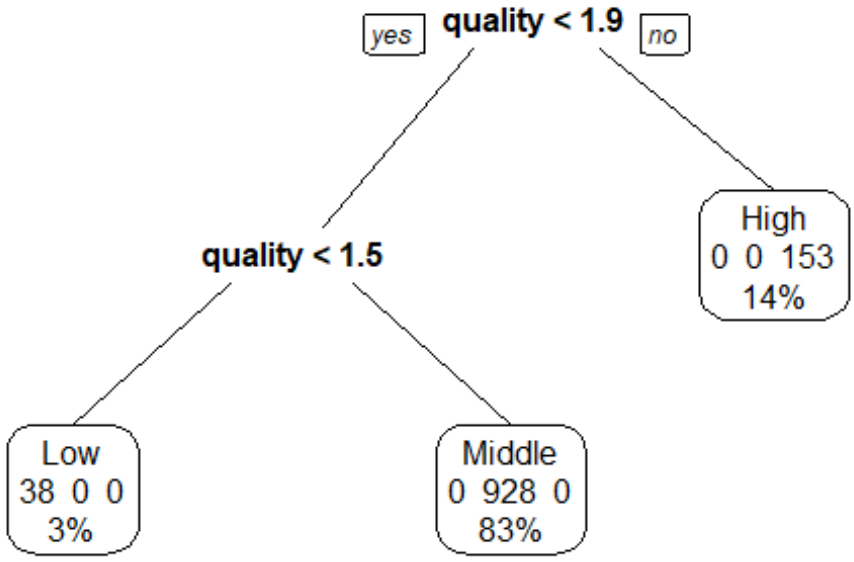
```
## [1] "Accuracy for test is % 100"
```

Modele ilişkin hata oranı (error rate) ve doğruluk oranı (accuracy rate) aşağıda verilmiştir. Kurulan model % 0 hata oranı (error rate) ve % 100 doğruluk oranı (accuracy rate) ise hedef değişkeni tahmin etmiştir.

Karar Ağacı Oluşturma–2

Veri seti basit tesadüfi örneklem çekilerek eğitilecek ve test edilecek veri setine ayrılır. Daha sonra veri setindeki "Classification" değişkeni hedef değişken belirlenerek numerik değişkenlere logaritmik (log) dönüşüm uygulanarak modele sokulmuştur. Elde edilen sonuçlar karar ağacında görselleştirildikten sonra hata ve doğruluk oranları verilmiştir.

```
rd<-log(t$residualsugar)
a<-log(t$alcohol)
q<-log(t$quality)
t11<-tibble(residualsugar=rd, alcohol=a,quality=q,
Classification=t$Classification)

set.seed(1230)
n=NROW(t11)#veri setindeki satır sayısı
n

## [1] 1599

SRS <- sample(sample(1:n, size = round(0.7*n), replace=FALSE))#veri setinden
basit tesadüfi örneklem çekilerek eğitilecek veri seti ve test setine ayırmak
için
training <- t11[SRS, ]#eğitilecek veri seti
test <- t11[-SRS, ]#test edilecek veri seti
model <- rpartXse(Classification ~ ., training, se=0.5)

pred1 <- predict(model, test, type="class")
head(pred1)

##      1      2      3      4      5      6
## Middle   High   High Middle Middle Middle
## Levels: Low Middle High

library(rpart.plot)#karar ağacının görselleştirilmesi için
prp(model, type=0, extra=101)

## Warning: Cannot retrieve the data used to build the model (so cannot
determine roundint and is.binary for the variables).
## To silence this warning:
##     Call prp with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



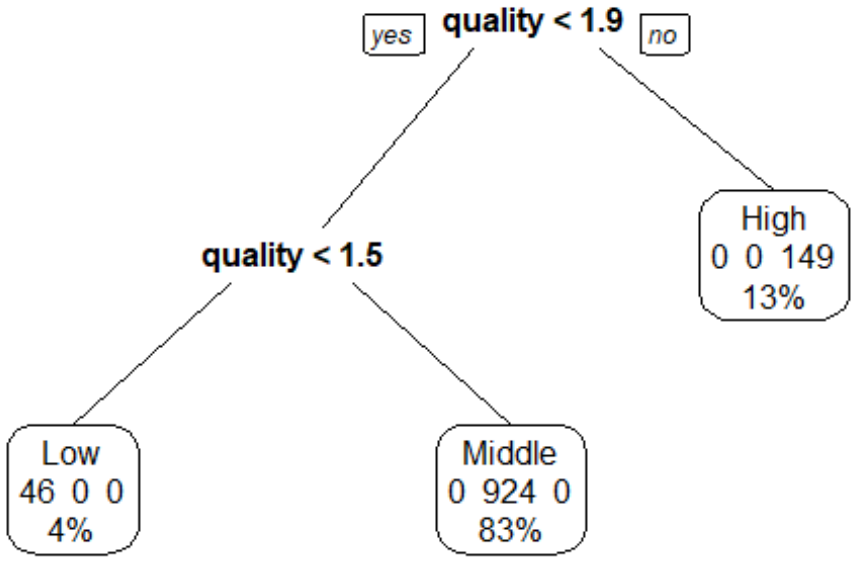

```
(cm <- table(pred1, test$Classification))

##
## pred1    Low Middle High
##   Low     25      0    0
##   Middle   0    391    0
##   High     0      0   64

error_rate<-100*(1-sum(diag(cm))/sum(cm))

print(paste('Error Rate is %', round(error_rate))) # the error rate (hata
oranı)

## [1] "Error Rate is % 0"

accuracy_test <- sum(diag(cm)) / sum(cm)
print(paste('Accuracy for test is %', 100*accuracy_test))#Accuracy
rate(doğruluk oranı)

## [1] "Accuracy for test is % 100"
```

Log dönüşümü uygulandıktan sonra modele ilişkin hata oranı (error rate) ve doğruluk oranı (accuracy rate) aşağıda verilmiştir. Kurulan model % 0 hata oranı (error rate) ve % 100 doğruluk oranı (accuracy rate) hedef değişkeni tahmin etmiştir. Elde edilen sonuçlar log dönüşümü uygulanmadan önceki elde edilen sonuçlarla aynıdır.

Karar Ağacı Oluşturma–3

Veri seti basit tesadüfi örneklem çekilerek eğitilecek ve test edilecek veri setine ayrılır. Daha sonra veri setindeki "Classification" değişkeni hedef değişken belirlenerek numerik değişkenler normalize edilerek modele sokulmuştur. Elde edilen sonuçlar karar ağacında görselleştirildikten sonra hata ve doğruluk oranları verilmiştir.

```
lf<-scale(t$residualsugar)
e<-scale(t$alcohol)
u<-scale(t$quality)
t12<-tibble(residualsugar=rd, alcohol=a,quality=q,
Classification=t11$Classification)

set.seed(1500)
n=NROW(t12)

SRS <- sample(sample(1:n, size = round(0.7*n), replace=FALSE))#veri setinden
basit tesadüfi örneklem çekilerek eğitilecek veri seti ve test setine ayırmak
için
training <- t12[SRS, ]#eğitilecek veri seti
test <- t12[-SRS, ]#test edilecek veri seti
model <- rpartXse(Classification ~ ., training, se=0.5)

pred1 <- predict(model, test, type="class")
head(pred1)

##      1      2      3      4      5      6
## Middle   High Middle Middle   High Middle
## Levels: Low Middle High

library(rpart.plot)#karar ağacının görselleştirilmesi için
prp(model, type=0, extra=101)

## Warning: Cannot retrieve the data used to build the model (so cannot
determine roundint and is.binary for the variables).
## To silence this warning:
##     Call prp with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



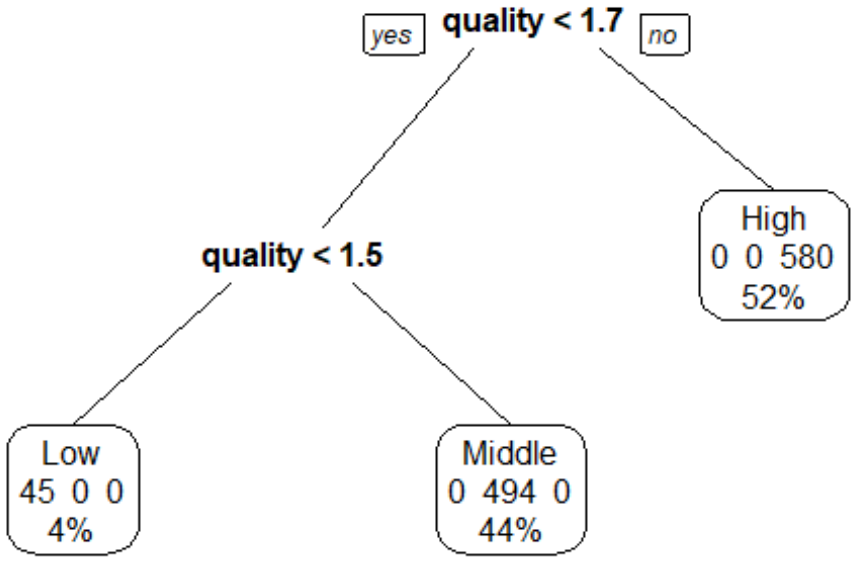

```
(cm <- table(pred1, test$Classification))

##
## pred1    Low Middle High
##   Low     17      0    0
##   Middle   0    395    0
##   High     0      0   68

error_rate<-100*(1-sum(diag(cm))/sum(cm))

print(paste('Error Rate is %', round(error_rate))) # the error rate (hata
oranı)

## [1] "Error Rate is % 0"

accuracy_test <- sum(diag(cm)) / sum(cm)
print(paste('Accuracy for test is %', 100*accuracy_test))#Accuracy
rate(doğruluk oranı)

## [1] "Accuracy for test is % 100"
```

Veri seti normalize edildikten sonra modele ilişkin hata oranı (error rate) ve doğruluk oranı (accuracy rate) aşağıda verilmiştir. Kurulan model % 0 hata oranı (error rate) ve % 100 doğruluk oranı (accuracy rate) ile hedef değişkeni tahmin etmiştir. Elde edilen sonuçlar önceki sonuçlarla aynıdır.

Karar Ağacı Oluşturma-4 İlk olarak “Classification” değişkenine temel oluşturan quality(kalite) oranı değişkeni ve bağımlı değişkenlerin logaritmik dönüşümü yapılmıştır.

Daha sonra veri seti basit tesadüfi örneklem çekilerek eğitilecek ve test edilecek veri setine ayrılmıştır. Elde edilen sonuçlar karar ağacında görselleştirildikten sonra hata ve doğruluk oranları verilmiştir.

```
df_20_log<-cut(log(winequality_red$quality), breaks = 3, labels = c("Low",
"Middle", "High"))

t_log<-tibble(residual_sugar=log(winequality_red$`residual sugar`),
alcohol=log(winequality_red$alcohol), quality=log(winequality_red$quality),
Classification=df_20_log)

set.seed(1535)
n=NROW(t_log)
SRS <- sample(sample(1:n, size = round(0.7*n), replace=FALSE))
training <- t_log[SRS, ]
test <- t_log[-SRS, ]
model <- rpartXse(Classification ~ ., training, se=0.5)

pred1 <- predict(model, test, type="class")
head(pred1)

##      1      2      3      4      5      6
## Middle Middle   High Middle   High Middle
## Levels: Low Middle High

prp(model, type=0, extra=101)

## Warning: Cannot retrieve the data used to build the model (so cannot
determine roundint and is.binary for the variables).
## To silence this warning:
##     Call prp with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
(cm <- table(pred1, test$Classification))

##
## pred1    Low Middle High
##   Low     18      0    0
##   Middle   0    187    0
##   High     0      0  275

error_rate<-100*(1-sum(diag(cm))/sum(cm))

print(paste('Error Rate is %', round(error_rate))) # the error rate (hata
oranı)

## [1] "Error Rate is % 0"

accuracy_test <- sum(diag(cm)) / sum(cm)
print(paste('Accuracy for test is %', 100*accuracy_test))#Accuracy
rate(doğruluk oranı)

## [1] "Accuracy for test is % 100"
```

Veri setinin tamanına log dönüşümü uygulandıktan sonra modele ilişkin hata oranı (error rate) ve doğruluk oranı (accuracy rate) aşağıda verilmiştir. Kurulan model % 0 hata oranı (error rate) ve % 100 doğruluk oranı (accuracy rate) ile hedef değişkeni tahmin etmiştir.