Homework 3

Vedant Limbhare

2023-05-02

Q1.Fit an logistic regression model to classify forged banknote from genuine banknotes.

```
bank_auth <- read.csv("data_banknote_authentication.csv")
str(bank_auth)
```

```
## 'data.frame':    1372 obs. of  5 variables:
##  $ Variance: num  3.622 4.546 3.866 3.457 0.329 ...
##  $ skewness: num  8.67 8.17 -2.64 9.52 -4.46 ...
##  $ curtosis: num  -2.81 -2.46 1.92 -4.01 4.57 ...
##  $ entropy : num  -0.447 -1.462 0.106 -3.594 -0.989 ...
##  $ class   : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
bank_auth$class <- factor(bank_auth$class, levels = c(0,1), labels=c("genuine", "forged"))
```

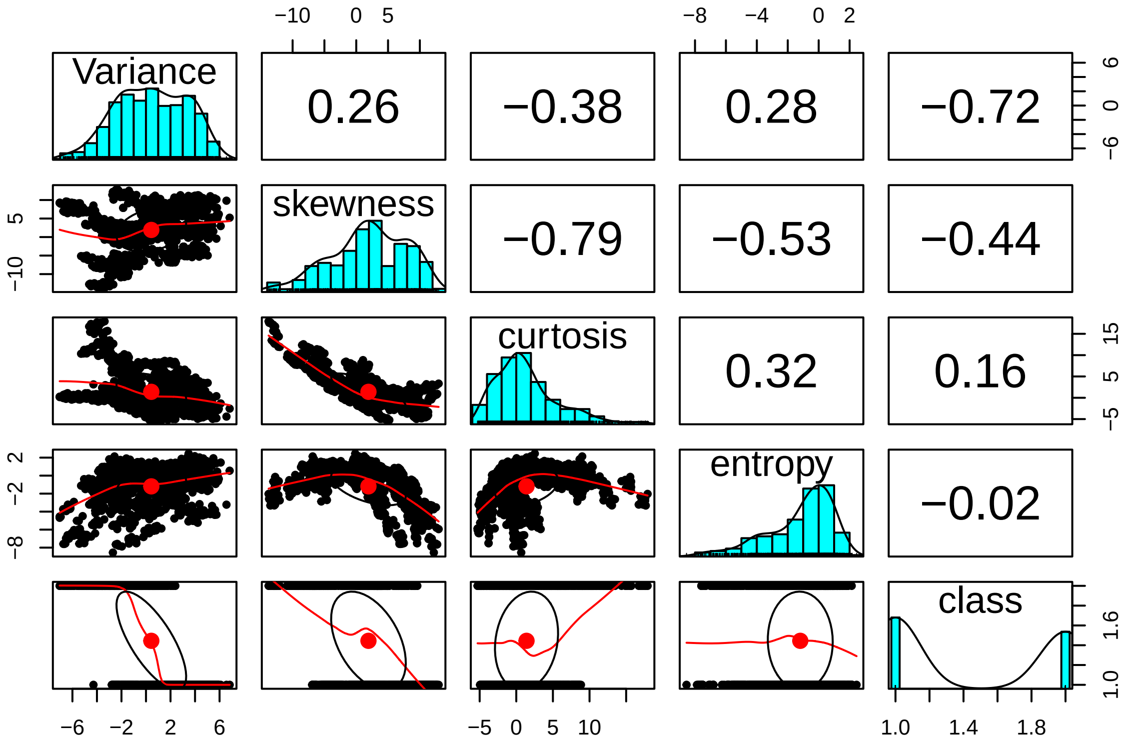
```
attach(bank_auth)
```

```
summary(bank_auth)
```

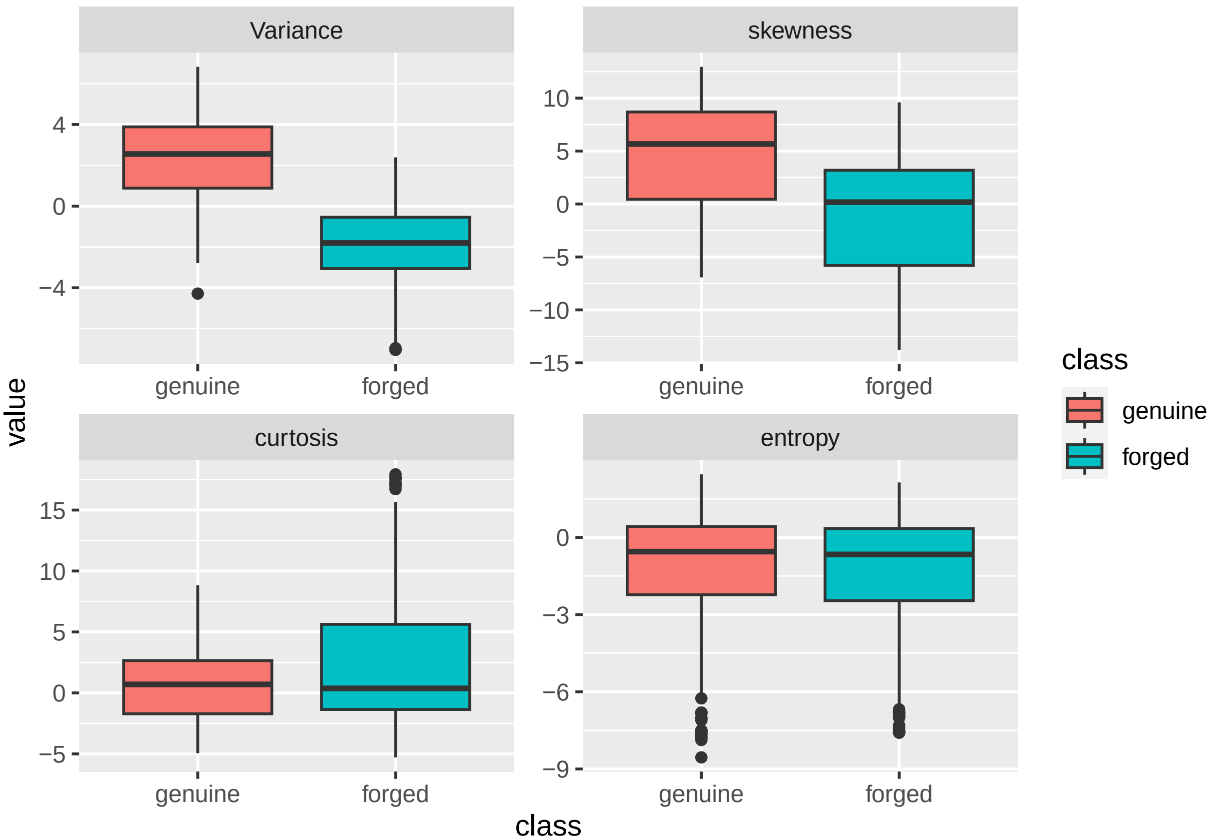
```
##     Variance          skewness          curtosis          entropy
##  Min.   :-7.0421   Min.   :-13.773   Min.   :-5.2861   Min.   :-8.5482
##  1st Qu.:-1.7730   1st Qu.: -1.708   1st Qu.:-1.5750   1st Qu.:-2.4135
##  Median : 0.4962   Median :  2.320   Median : 0.6166   Median :-0.5867
##  Mean   : 0.4337   Mean   :  1.922   Mean   : 1.3976   Mean   :-1.1917
##  3rd Qu.: 2.8215   3rd Qu.:  6.815   3rd Qu.: 3.1793   3rd Qu.: 0.3948
##  Max.   : 6.8248   Max.   : 12.952   Max.   :17.9274   Max.   : 2.4495
##      class
##  genuine:762
##  forged :610
##
##
##
##
```

Q1.1 Produce some numerical and graphical summaries of the data set. Explain the relationships.

```
pairs.panels(bank_auth)
```




```r
df_box <- melt(bank_auth, id.var = "class")


ggplot(data = df_box, aes(x=class, y=value)) +
  geom_boxplot(aes(fill=class)) + facet_wrap(~variable,  scales = "free")
```



From the above Box plots, we can see that Variance has correlation with the class variable. We can easily differentiate between forged and genuine class values by using Variance.

Q1.2 Is this a balanced data set?.

```
table(bank_auth$class)
```

```
##
## genuine  forged
##     762     610
```

From the above values, we can observe that the genuine and forged values are almost equal hence there is no biases in the dataset and we can conclude that the dataset is Balanced.

Q1.3 Use the full data set to perform a logistic regression with Class as the response variable. Do any of the predictors appear to be statistically significant? If so, which ones?

```
glm.class=glm(class~.,data=bank_auth,family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm.class)
```

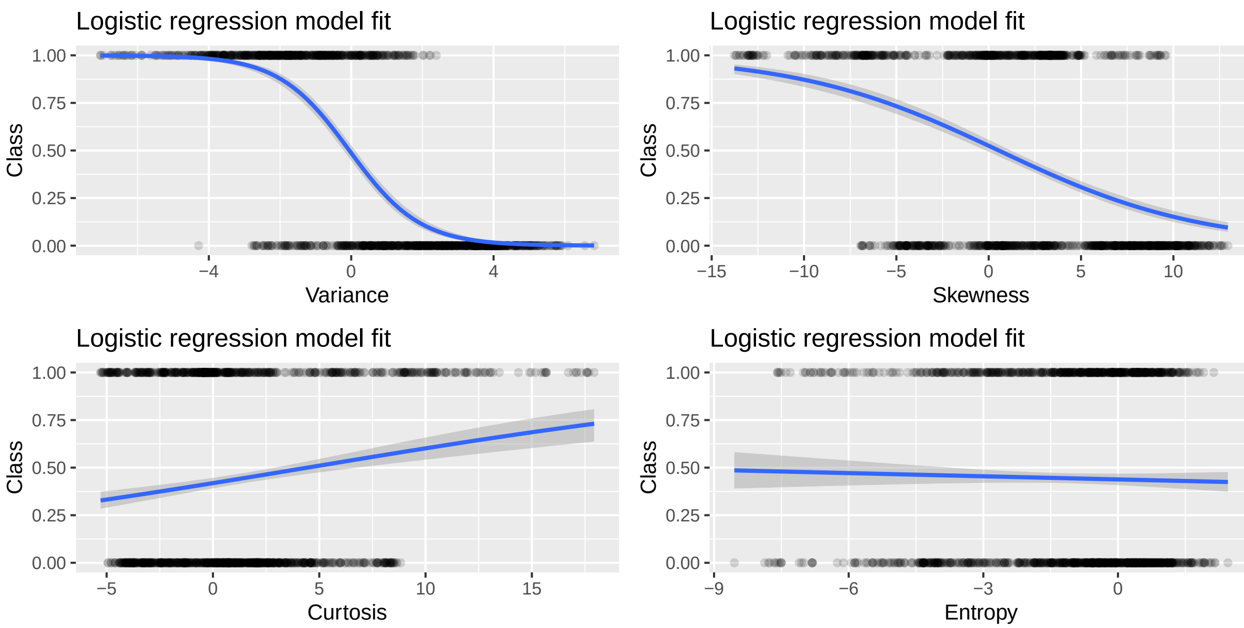
```
##
## Call:
## glm(formula = class ~ ., family = "binomial", data = bank_auth)
```

```
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70001   0.00000   0.00000   0.00029   2.24614
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   7.3218     1.5589   4.697 2.64e-06 ***
## Variance     -7.8593     1.7383  -4.521 6.15e-06 ***
## skewness     -4.1910     0.9041  -4.635 3.56e-06 ***
## curtosis     -5.2874     1.1612  -4.553 5.28e-06 ***
## entropy      -0.6053     0.3307  -1.830   0.0672 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1885.122  on 1371  degrees of freedom
## Residual deviance:   49.891  on 1367  degrees of freedom
## AIC: 59.891
##
## Number of Fisher Scoring iterations: 12
```

```r
p1 <- bank_auth %>%
    mutate(prob = ifelse(class == "forged", 1, 0)) %>%
    ggplot(aes(Variance, prob)) +
    geom_point(alpha = 0.15) +
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    ggtitle("Logistic regression model fit") +
    xlab("Variance") +
    ylab("Class")

p2 <- bank_auth %>%
    mutate(prob = ifelse(class == "forged", 1, 0)) %>%
    ggplot(aes(skewness, prob)) +
    geom_point(alpha = 0.15) +
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    ggtitle("Logistic regression model fit") +
    xlab("Skewness") +
    ylab("Class")

p3 <- bank_auth %>%
    mutate(prob = ifelse(class == "forged", 1, 0)) %>%
    ggplot(aes(curtosis, prob)) +
    geom_point(alpha = 0.15) +
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    ggtitle("Logistic regression model fit") +
    xlab("Curtosis") +
    ylab("Class")

p4 <- bank_auth %>%
    mutate(prob = ifelse(class == "forged", 1, 0)) %>%
    ggplot(aes(entropy, prob)) +
    geom_point(alpha = 0.15) +
```

```
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    ggtitle("Logistic regression model fit") +
    xlab("Entropy") +
    ylab("Class")


grid.arrange(p1, p2, p3, p4, nrow=2, ncol= 2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
exp(coef(glm.class))
```

```
##   (Intercept)     Variance     skewness     curtosis      entropy
## 1.512932e+03 3.861323e-04 1.513170e-02 5.054731e-03 5.459003e-01
```

```
glm.var=glm(class~Variance,data=bank_auth,family="binomial")
```

```
confint(glm.var)
```

```
## Waiting for profiling to be done...

##                  2.5 %     97.5 %
## (Intercept) -0.214843  0.1105295
## Variance    -1.119059 -0.9115042
```

From the above observed graphs, Variance has relation between class variable. The skewness does shows a little relation but other variables are insignificant to determine class values.

Q1.4 Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
glm.probs=predict(glm.class,type="response")
glm.probs[1:10]
```

```
##            1            2            3            4            5            6
## 2.220446e-16 2.220446e-16 2.185822e-10 2.220446e-16 4.579103e-01 2.220446e-16
##            7            8            9           10
## 2.220446e-16 1.435064e-11 2.220446e-16 2.220446e-16
```

```
glm.pred=rep("genuine",nrow(bank_auth))
glm.pred[glm.probs>.5]="forged"
```

```
confusionMatrix(as.factor(glm.pred),class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction genuine forged
##    genuine     757      6
##    forged        5    604
##
##                Accuracy : 0.992
##                  95% CI : (0.9857, 0.996)
##     No Information Rate : 0.5554
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9838
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9934
##             Specificity : 0.9902
##          Pos Pred Value : 0.9921
##          Neg Pred Value : 0.9918
##              Prevalence : 0.5554
##          Detection Rate : 0.5517
##    Detection Prevalence : 0.5561
##       Balanced Accuracy : 0.9918
##
##        'Positive' Class : genuine
##
```

We can distinguish two different sorts of errors made by the logistic regression model based on the confusion matrix:

False negatives: The model wrongly classified 6 legitimate papers as forgeries . Due to the model's failure to recognize these papers as authentic, they are known as false negatives.

False positives: The model predicted 5 fabricated documents as genuine when it should have been forgery. These are referred to as false positives because the model predicted wrong forged entries.

In total, the logistic regression model only miscalculated 11 times out of 1362 documents. The model show's high predictability of 0.992.

Q1.5 Create a training set with 80% of the observations, and a testing set containing the remaining 20%.Compute the confusion matrix and the overall fraction of correct prediction for the testing data set.

```
set.seed(12)

index <- createDataPartition(bank_auth$class, p = 0.8, list = FALSE)
train <- bank_auth[index, ]
test <- bank_auth[-index, ]

glm.class2 <- glm(class ~ ., data = train, family = binomial)

glm.pred2 <- predict(glm.class2, newdata = test, type = "response")
glm.pred2 <- ifelse(glm.pred2 >= 0.5, "forged", "genuine")

confusionMatrix(as.factor(glm.pred2), as.factor(test$class))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction genuine forged
##    genuine     152      2
##    forged        0    120
##
##                Accuracy : 0.9927
##                  95% CI : (0.9739, 0.9991)
##     No Information Rate : 0.5547
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9852
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.9836
##          Pos Pred Value : 0.9870
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5547
##          Detection Rate : 0.5547
##    Detection Prevalence : 0.5620
##       Balanced Accuracy : 0.9918
##
##        'Positive' Class : genuine
##
```

Overall fraction of correct predictions for testing dataset given by confusion matrix are:

The overall fraction of correct prediction, or accuracy, is 0.9927. This indicates that the model correctly predicted 99.27% of the papers as forged and genuine in the testing data-set.

True negatives: The model had 122 forged documents in the test data-set, and the model predicted 120 of them as genuine and 2 documents as false negatives.

True positives: The model had 152 genuine documents in the test data-set, and the model predicted all of them as genuine.

In total, the logistic regression model only miscalculated 2 times out of 272 documents. The model show's high predictability of 99.27%.

Q2. Fit an regression tree model to predict quality of wine.

```
wine_ds2 <- read.csv2("winequality.csv")
```

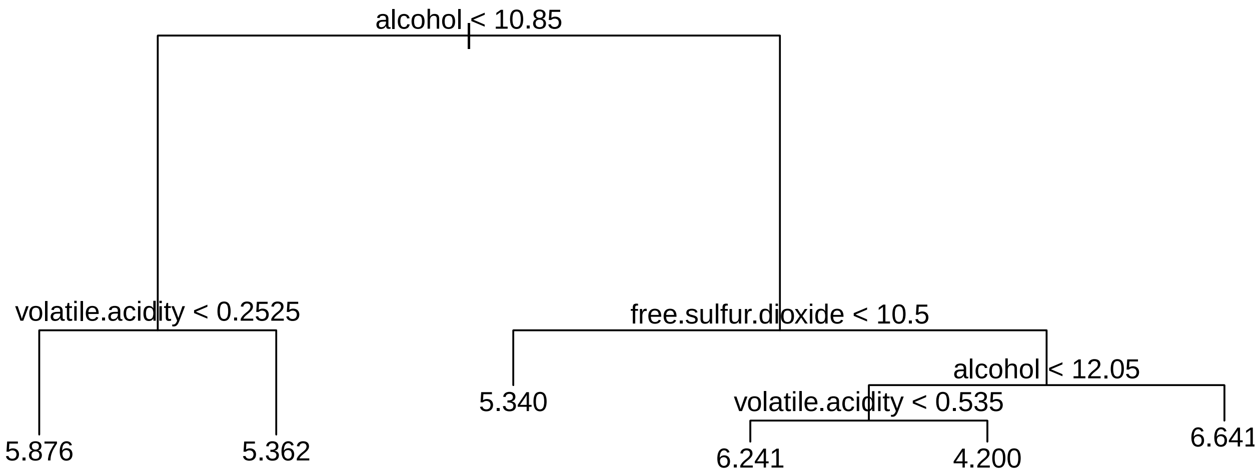
```
set.seed(1)
train = sample(1:nrow(wine_ds2), nrow(wine_ds2)/2)
```

Q2.1 Produce some numerical and graphical summaries of the data set. Explain the relationships.

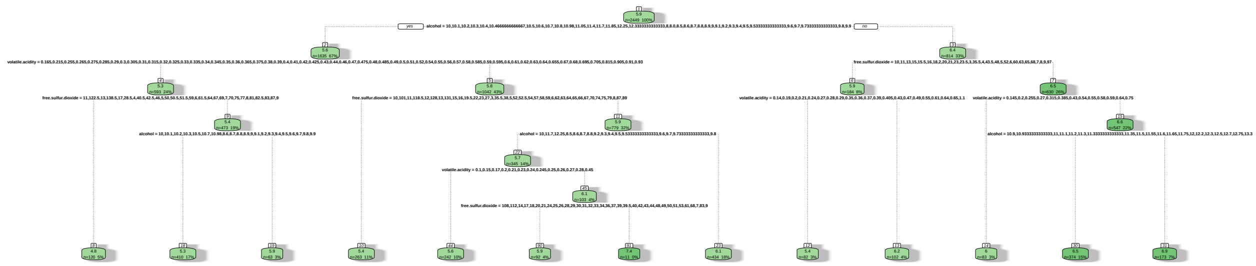
```
tree.wine_ds2=tree(quality~.,wine_ds2,subset=train)
summary(tree.wine_ds2)
```

```
##
## Regression tree:
## tree(formula = quality ~ ., data = wine_ds2, subset = train)
## Variables actually used in tree construction:
## [1] "alcohol"             "volatile.acidity"    "free.sulfur.dioxide"
## Number of terminal nodes:  6
## Residual mean deviance:  0.5862 = 1432 / 2443
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.8760 -0.3618 -0.2409  0.0000  0.6382  2.6600
```

```
plot(tree.wine_ds2)
text(tree.wine_ds2,pretty=0)
```




```
tree.wine_ds2_1 = rpart(quality~ alcohol+volatile.acidity+free.sulfur.dioxide,wine_ds2,subset = train)

fancyRpartPlot(tree.wine_ds2_1)
```

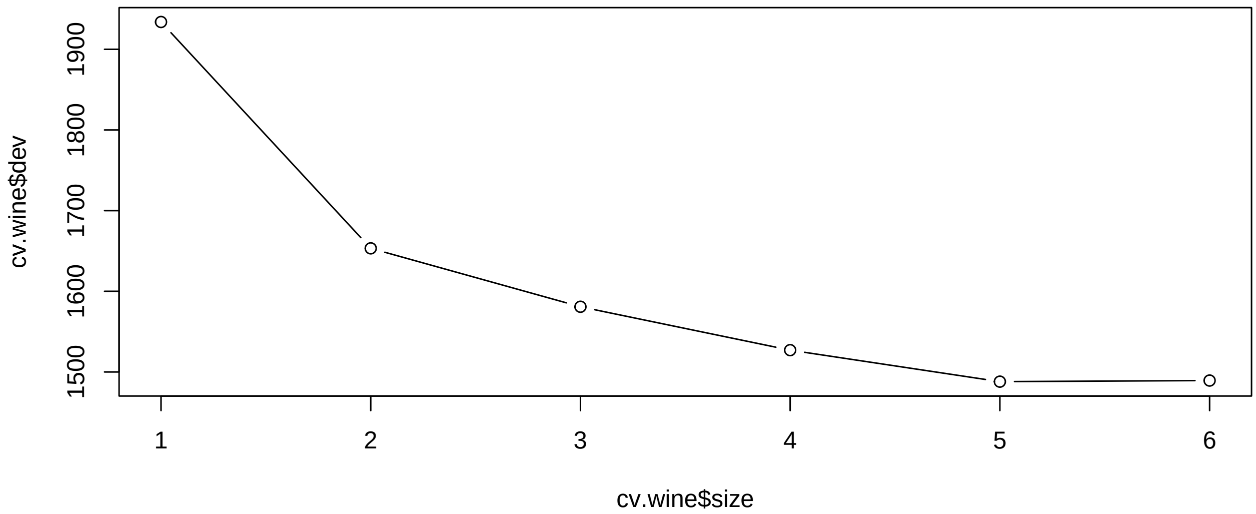


Rattle 2023−May−02 06:16:31 vedant

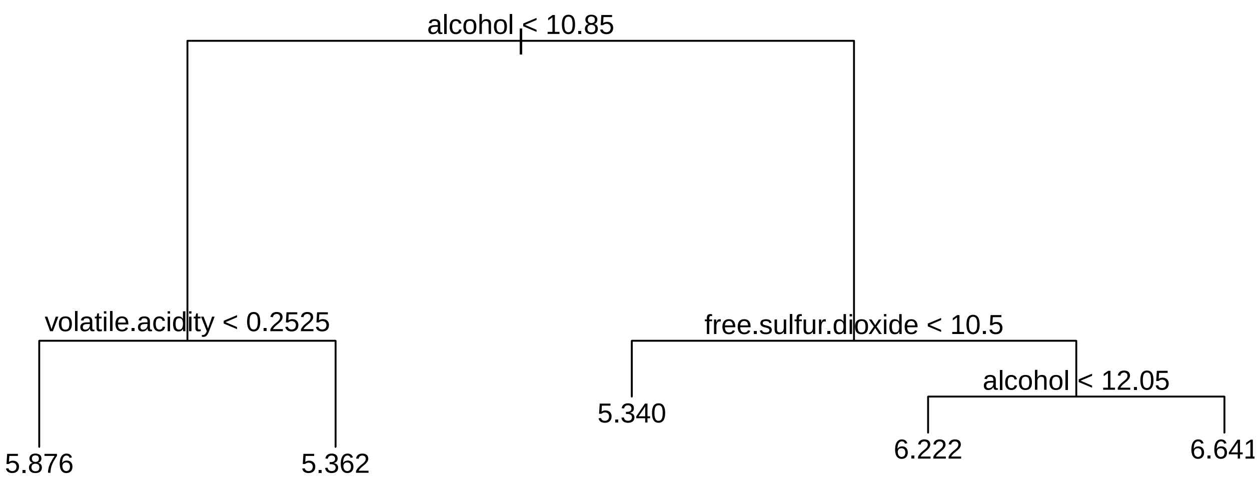
```
visTree(tree.wine_ds2_1)
```

Export as png

```r
cv.wine=cv.tree(tree.wine_ds2)
plot(cv.wine$size ,cv.wine$dev,type='b')
```




```
prune.wine=prune.tree(tree.wine_ds2,best=5)
plot(prune.wine)
text(prune.wine,pretty=0)
```



As seen from above observations, we can conclude that the variable "Quantity" is mostly dependent on 3 variables from wine dataset mainly: 'volatile.acidity', 'free.sulphur.dioxide' and 'alcohol'. Further we have tried to plot the decision tree using only this 3 variables. After plotting, we can see that the least 'dev' value we have got is using 5 decision leafs and hence we have used the best=5 method in our prune model for cross-validation.

Q2.2 Create a training set with 80% of the observations, and a testing set containing the remaining 20%.

```
set.seed(11)
train = sample(1:nrow(wine_ds2), 0.8*nrow(wine_ds2))
```

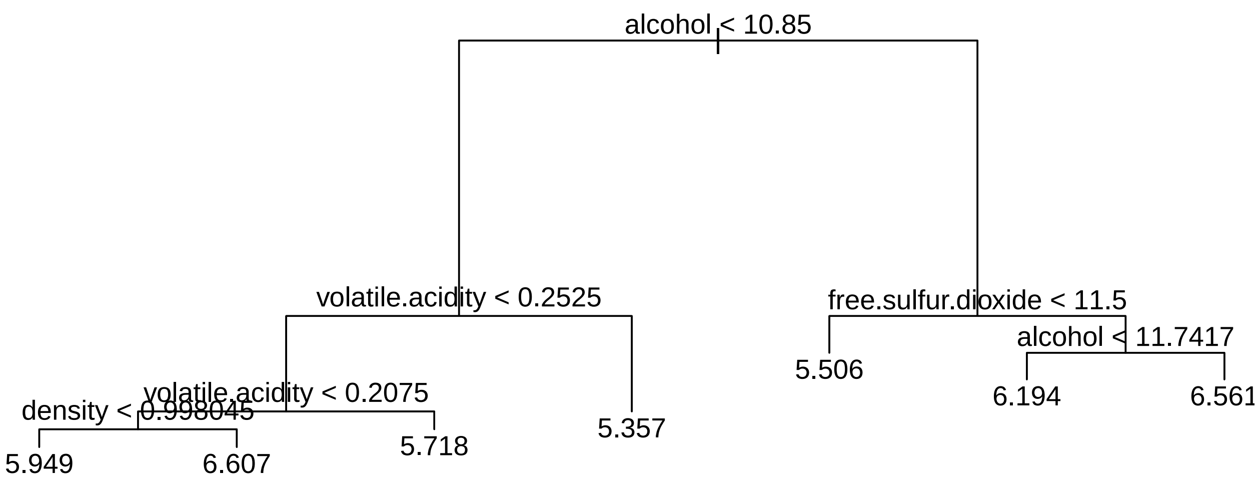
Q2.3 Fit a regression tree with quality as the response variable using the training set. Plot the tree and interpret the results. What test MSE do you obtain?

```
tree.wine_ds2=tree(quality~.,wine_ds2,subset=train)
summary(tree.wine_ds2)
```

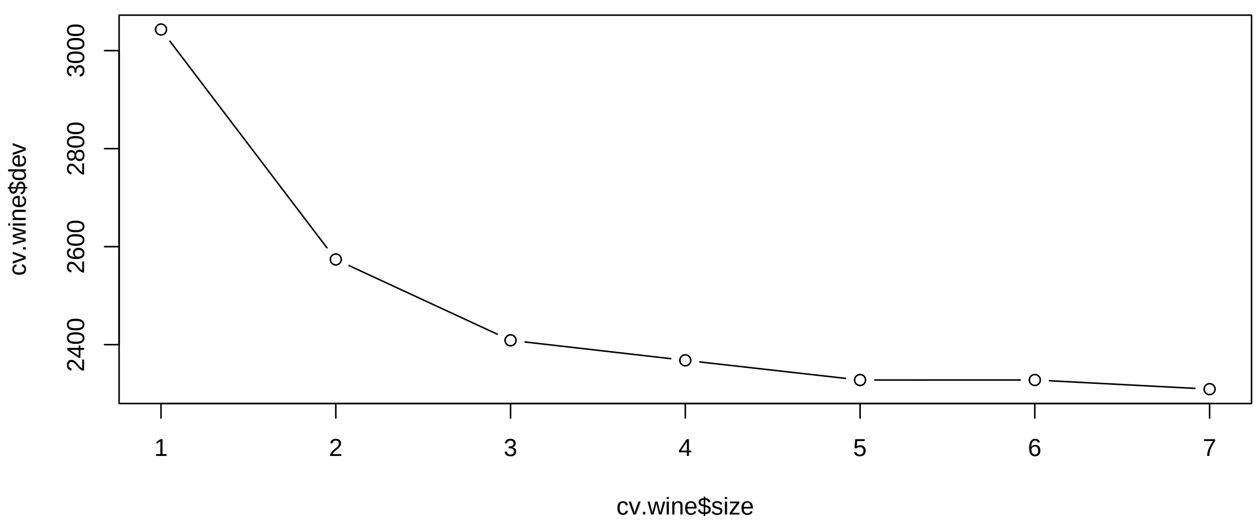
```
##
```

```
## Regression tree:
## tree(formula = quality ~ ., data = wine_ds2, subset = train)
## Variables actually used in tree construction:
## [1] "alcohol"              "volatile.acidity"    "density"
## [4] "free.sulfur.dioxide"
## Number of terminal nodes:  7
## Residual mean deviance:  0.568 = 2222 / 3911
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.5610 -0.3570 -0.1941  0.0000  0.4944  3.6430
```

```r
plot(tree.wine_ds2)
text(tree.wine_ds2,pretty=0)
```




```r
cv.wine=cv.tree(tree.wine_ds2)
plot(cv.wine$size ,cv.wine$dev,type='b')
```




```r
yhat=predict(tree.wine_ds2,newdata=wine_ds2[-train,])
wine.test=wine_ds2[-train,"quality"]

mean((yhat-wine.test)^2)
```

```
## [1] 0.5623576
```

```r
sqrt(mean((yhat-wine.test)^2))
```

```
## [1] 0.749905
```

After fitting 80% observations in training set, we observe that the variables used for constructing decision tree is 4 variables mainly: "alcohol", "volatile.acidity", "density" and "free.sulfur.dioxide".

Q2.4 Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

```r
yhat=predict(prune.wine,newdata=wine_ds2[-train,])
wine.test=wine_ds2[-train,"quality"]

mean((yhat-wine.test)^2)
```

```
## [1] 0.5774985
```

```r
sqrt(mean((yhat-wine.test)^2))
```

```
## [1] 0.7599333
```

After pruning, we can see a slight increase in the MSE value which proves that using best 5 variables actually increase the MSE value. Hence, we will consider the method without pruning.

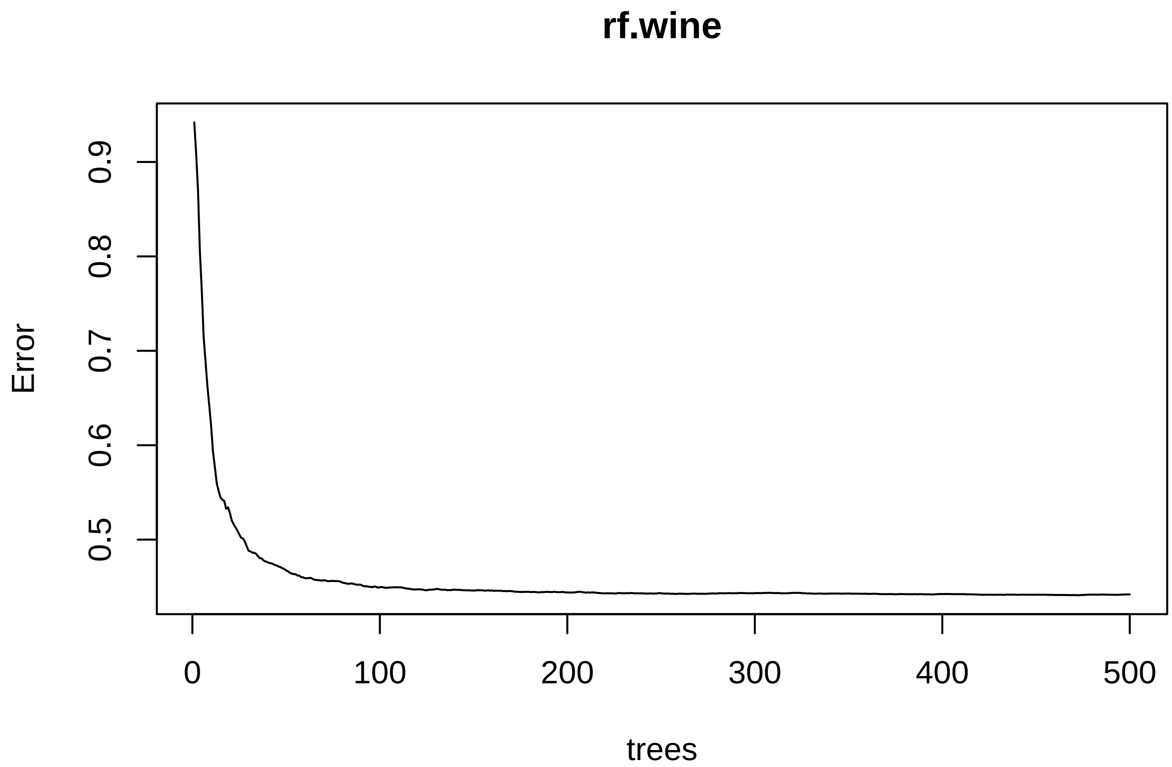
Q2.5 Use random forests to analyze this data. What test MSE do you obtain?

```r
set.seed(12345)
rf.wine=randomForest(quality~.,data=wine_ds2,subset=train,mtry=3,importance=TRUE)
yhat.rf = predict(rf.wine,newdata=wine_ds2[-train,])
mean((yhat.rf - wine.test)^2)
```

```
## [1] 0.4767698
```

After using Random forest method on the same dataset, we are getting the lowest MSE value when using predictor variables as 3 for each tree by using mtry=3. The MSE value that we get after using Random forest is 0.4760 which is lower than the MSE value 0.5702 which we obtained by using Decision Tree method.

```r
plot(rf.wine)
```

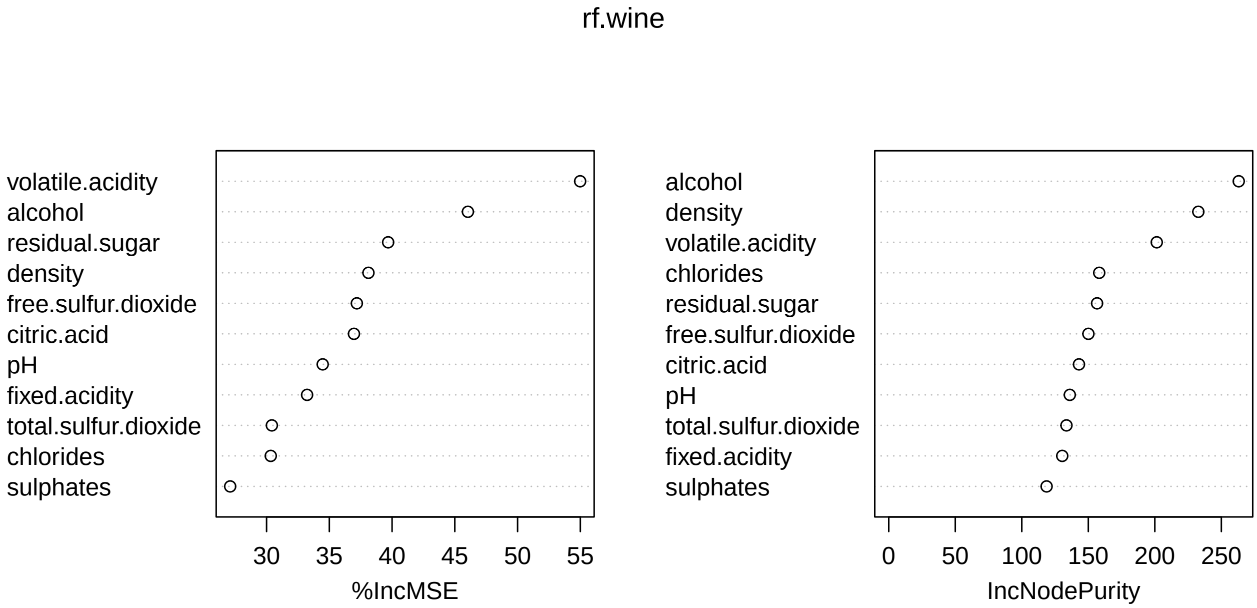


Q2.6 Use the importance() function to determine which variables are most important.

```
importance(rf.wine)
```

```
##                       %IncMSE IncNodePurity
## fixed.acidity        33.22879      130.4068
## volatile.acidity     54.98630      201.4475
## citric.acid          36.96238      142.9901
## residual.sugar       39.68919      156.5936
## chlorides            30.33338      158.2045
## free.sulfur.dioxide  37.19351      150.0679
## total.sulfur.dioxide 30.42046      133.5794
## density              38.11791      232.6974
## pH                   34.47169      136.0848
## sulphates            27.10016      118.6530
## alcohol              46.04182      263.0629
```

```
pred_Imp <- varImpPlot(rf.wine)
```



The most important variables that we get are density, alcohol, volatile.acidity, residual sugar.