Programmazione Base in R: ITS Machine Learning 2020-2022

HEART DISEASE
DESCRIPTIVE ANALYSIS

INTRODUCTION

I will analyze all the data of the heart.csv database, what attributes are, the management of the outliers, the elimination of unnecessary columns, the code used and the management of the NAs, data plots, graphs used, packets used, linear regression and the ML algorithms used.

INTRODUCTION TO CODE

I package utilizzati sono: tidyverse, ggplot2 e caret

```
heart <- read.csv("heart.csv", header = TRUE, stringsAsFactors = FALSE)
str(heart)
summary(heart)
View(heart)
```

I imported heart.csv and checked data and attributes.

- age : age in year
- sex : (1 = male; 0 = female)
- cp : chest pain type
- trestbps : resting blood pressure (in mm Hg on admission to the hospital)
- chol : serum cholestoral in mg/dl
- fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- restecg : resting electrocardiographic results
- thalach : maximum heart rate achieved
- exang : exercise induced angina (1 = yes; 0 = no)
- oldpeak : ST depression induced by exercise relative to rest

- slope : the slope of the peak exercise ST segment
- ca : number of major vessels (0–3) colored by flourosopy
- thal : 1 = normal; 2 = fixed defect; 3 = reversable defect
- target : 1 = disease; 0 = no disease

interval attributes are: trestbps, chol, thalach
ordinal attributes are: age
nominal attributes are: sex, cp, fbs, restecg, exang, slop, thal, ca, target

```
names(heart)[names(heart) == "trestbps"] <- "restbps"
names(heart)[names(heart) == "thalach"] <- "maxhrh"
```

i have renamed these column

```
heart <- subset(heart, select = -x)
as_tibble(heart)
options(max.print=999999)
heart %>% distinct()
```

I deleted x because it was irrelevant and then a deleted all the duplicates

```
heart$sex[heart$sex == "unspecified"] <- NA
heart$sex <- as.factor(heart$sex)
levels(heart$sex)
```

transformed “unspecified” in NA then sex as factor

```
heart$chol[heart$chol == "undefined"] <- NA
heart$chol <- as.integer(heart$chol)
```

i did the same with undefined and chol to integer

```
heart$cp <- as.factor(heart$cp)
heart$fbs <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exang <- as.factor(heart$exang)
heart$slope <- as.factor(heart$slope)
heart$ca <- as.factor(heart$ca)
heart$thal <- as.factor(heart$thal)
heart$target <- as.factor(heart$target)
```

the remaining nominal attributes to factor

```
heart$thal[heart$thal == 0] <- NA
heart$thal <- droplevels(heart$thal)
```

thal should be 1:3 values, i transformed 0 to NA

```
heart$age <- replace(heart$age, heart$age < 0, NA)
heart$age[is.na(heart$age)] <- median(heart$age, na.rm = TRUE)
```

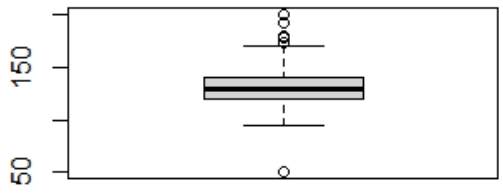
in age, values < 0 to NA then transformed to median

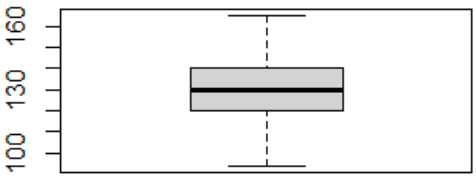
```
heart$ca <- replace(heart$ca, heart$ca == 4, NA)
heart <- na.omit(heart)
```

ca should be 0:3 so i transformed 4 to NA then i deleted all the NA in the database heart

outliers

```
boxplot(heart$restbps)
q3 <- quantile(heart$restbps, .75)
q1 <- quantile(heart$restbps, .25)
iqr <- (q3 - q1)
min <- (q1 - (1.5 * iqr))
max <- (q3 + (1.5 * iqr))
sum(heart$restbps < max | heart$restbps > min)
heart <- heart[heart$restbps > min & heart$restbps < max, ]
boxplot(heart$restbps)
```

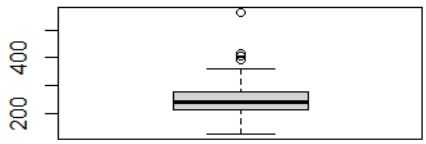


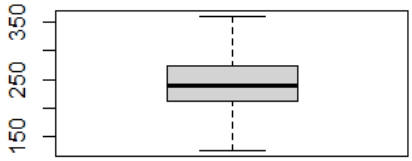


i made the boxplot before and after i remove the ouliers

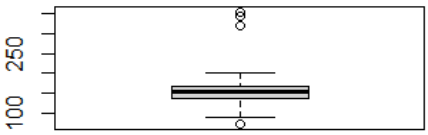
i did the same with other attributes

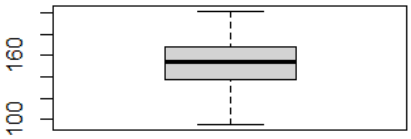
```
boxplot(heart$chol)
q3 <- quantile(heart$chol, .75)
q1 <- quantile(heart$chol, .25)
iqr <- (q3 - q1)
min <- (q1 - (1.5 * iqr))
max <- (q3 + (1.5 * iqr))
sum(heart$chol < max | heart$chol > min)
heart <- heart[heart$chol > min & heart$chol < max, ]
boxplot(heart$chol)
```



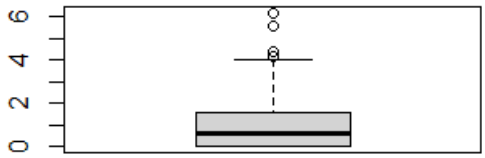


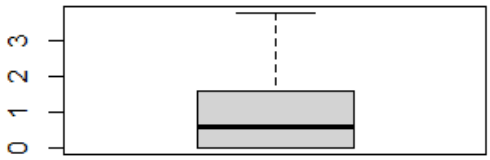

```
boxplot(heart$maxhrh)
q3 <- quantile(heart$maxhrh, .75)
q1 <- quantile(heart$maxhrh, .25)
iqr <- (q3 - q1)
min <- (q1 - (1.5 * iqr))
max <- (q3 + (1.5 * iqr))
sum(heart$maxhrh < max | heart$maxhrh > min)
heart <- heart[heart$maxhrh > min & heart$maxhrh < max, ]
boxplot(heart$maxhrh)
```






```r
boxplot(heart$oldpeak)
q3 <- quantile(heart$oldpeak, .75)
q1 <- quantile(heart$oldpeak, .25)
iqr <- (q3 - q1)
min <- (q1 - (1.5 * iqr))
max <- (q3 + (1.5 * iqr))
sum(heart$oldpeak < max | heart$oldpeak > min)
heart <- heart[heart$oldpeak > min & heart$oldpeak < max, ]
boxplot(heart$oldpeak)
```

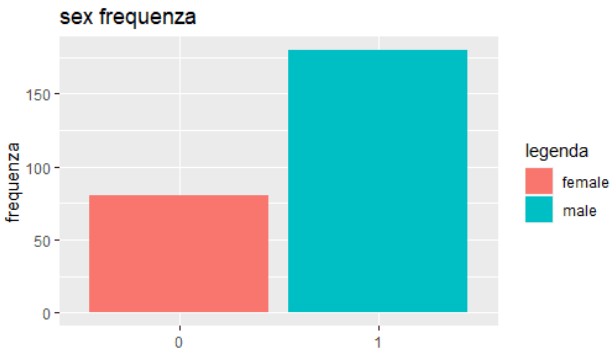


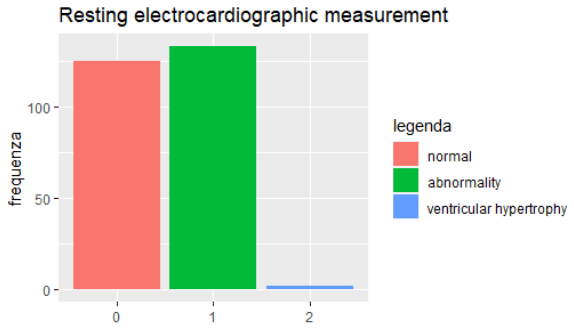


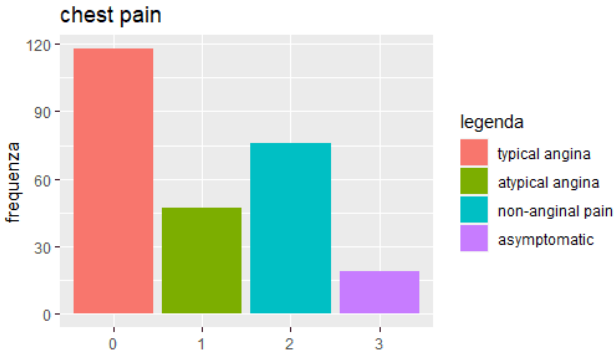
```
barplotfunc <- function(dato, titolo, x) {
  ggbar <- ggplot(heart, aes(x = dato, fill = dato, )) +
    scale_fill_discrete(name = "legenda", labels = x) +
    geom_bar() +
    labs(
      title = titolo,
      x = "", fill = "legenda",
      y = "frequenza"
    )
  plot(ggbar)
}
```

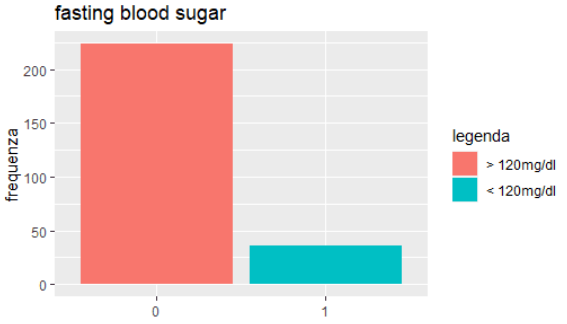
i made a function to see the frequencies

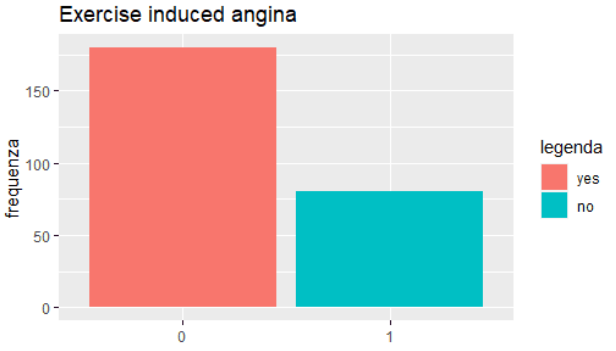
```
barplotfunc(heart$sex, "sex frequenza", c("female", "male"))
barplotfunc(heart$restecg, "Resting electrocardiographic measurement",
c("normal", "abnormality", "ventricular hypertrophy"))
barplotfunc(heart$cp, "chest pain", c("typical angina", "atypical angina",
"non-anginal pain", "asymptomatic"))
barplotfunc(heart$fbs, "fasting blood sugar", c("> 120mg/dl", "<
120mg/dl"))
barplotfunc(heart$exang, "Exercise induced angina", c("yes", "no"))
barplotfunc(heart$slope, "slope", c("upsloping", "flat", "downsloping"))
barplotfunc(heart$ca, "The number of major vessels", c("0", "1", "2", "3"))
barplotfunc(heart$target, "heart disease", c("yes", "no"))
barplotfunc(heart$thal, "thalassemia", c("normal", "fixed defect",
"reversable defect"))
```

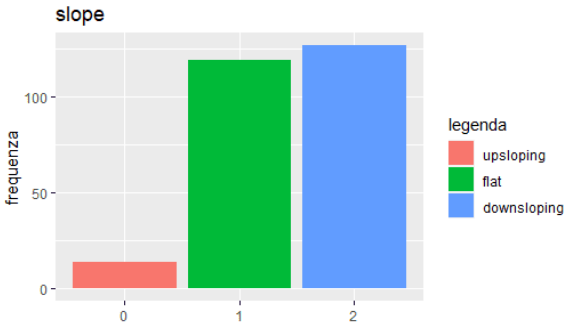

sex frequenza
frequenza
150
100
50
0
0
1
legenda
female
male

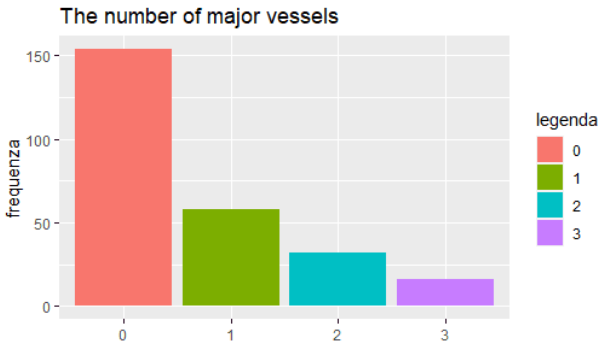

Resting electrocardiographic measurement
frequenza
100
50
0
0
1
2
legenda
normal
abnormality
ventricular hypertrophy

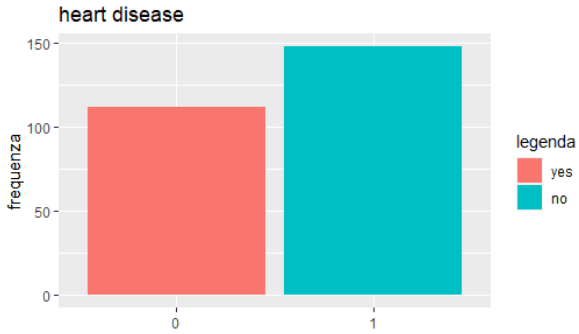

chest pain
frequenza
120
90
60
30
0
0
1
2
3
legenda
typical angina
atypical angina
non-anginal pain
asymptomatic

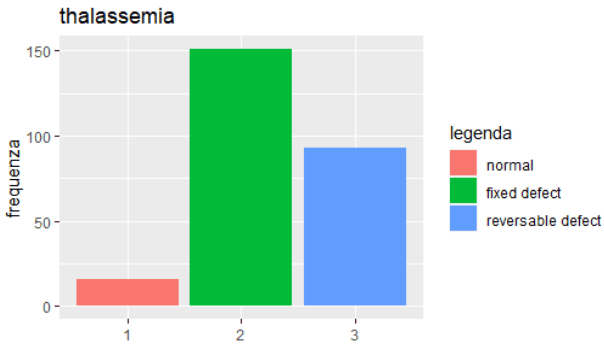

fasting blood sugar
frequenza
200
150
100
50
0
0
1
legenda
> 120mg/dl
< 120mg/dl


Exercise induced angina
frequenza
150
100
50
0
0
1
legenda
yes
no


slope
frequenza
100
50
0
0
1
2
legenda
upsloping
flat
downsloping



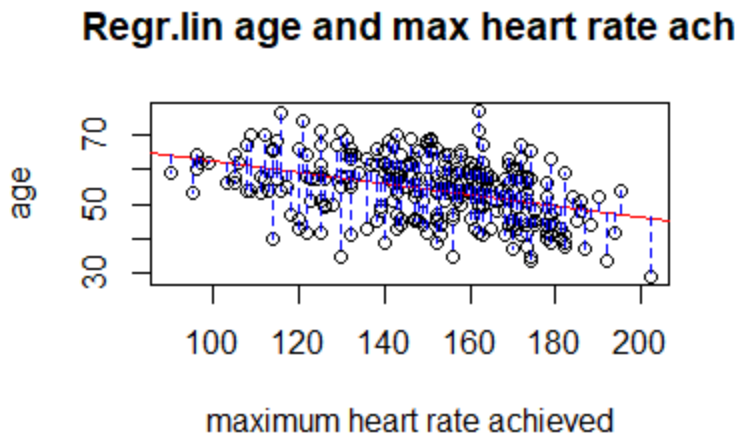




LINEAR REGRESSION

i made linear regression with age and maxhrh

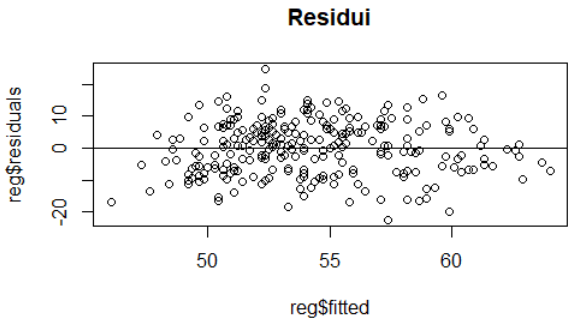
```
plot(heart$maxhrh, heart$age, xlab = "maximum heart rate achieved", ylab =
"age")
reg <- lm(age ~ maxhrh, data = heart)
abline(reg, col = "red")
segments(heart$maxhrh, fitted(reg), heart$maxhrh, heart$age, col = "blue",
lty = 2)
title(main = "Regr.lin age and max heart rate ach")
```

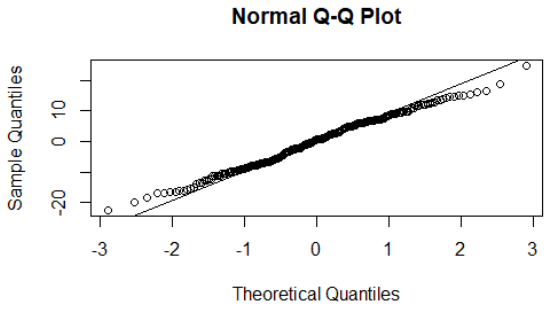



```
reg_2 <- lm(heart$age ~ I(heart$maxhrh - mean(heart$maxhrh)))
summary(reg_2)
confint(reg)
summary(reg)
r <- cov(heart$age, heart$maxhrh) / (sd(heart$age) * sd(heart$maxhrh))
r^2
```

the Intercept is 77.90437 and R-squared: 0.1589. That means that there is almost no correlation between age and maxhrh.

```
plot(reg$fitted, reg$residuals, main = "Residui")
abline(0, 0)
qqnorm(reg$residuals)
qqline(reg$residuals)
```





This is the residual distribution compared to normal distribution.

then i tried to predict 10 values that are not on the database.

```
reg1 <- lm(heart$age ~ heart$maxhrh)
pred <- data.frame("maxhrh" = c(90, 137, 200, 210, 150, 97, 267, 145, 100,
165))
predict(reg, pred, interval = "confidence")
```

```
       fit      lwr      upr
1  63.70142 60.83374 66.56909
2  56.28431 55.10688 57.46175
3  46.34224 43.90229 48.78220
4  44.76414 41.91373 47.61454
5  54.23278 53.21258 55.25297
6  62.59674 60.01789 65.17559
7  35.76893 30.47089 41.06697
8  55.02183 53.97567 56.06799
9  62.12331 59.66654 64.58007
10 51.86562 50.65142 53.07982
```

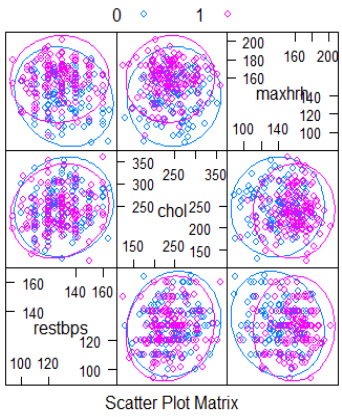
MACHINE LEARNING

I create the dataset: maxhrh, chol, restbps and target

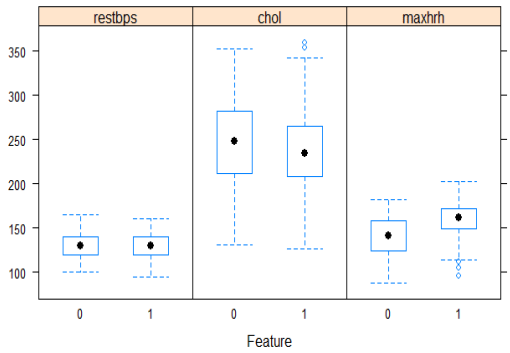
```
dataset <- heart[c(-1:-3,-6,-7,-9:-13)]
dim(dataset)
sapply(dataset, class)
head(dataset)
levels(dataset$target)
percentage <- prop.table(table(dataset$target)) * 100
cbind(freq = table(dataset$target), percentage = percentage)
summary(dataset)
x <- dataset[, 1:3]
dim(x)
y <- dataset[, 4]
dim(t(y))
```

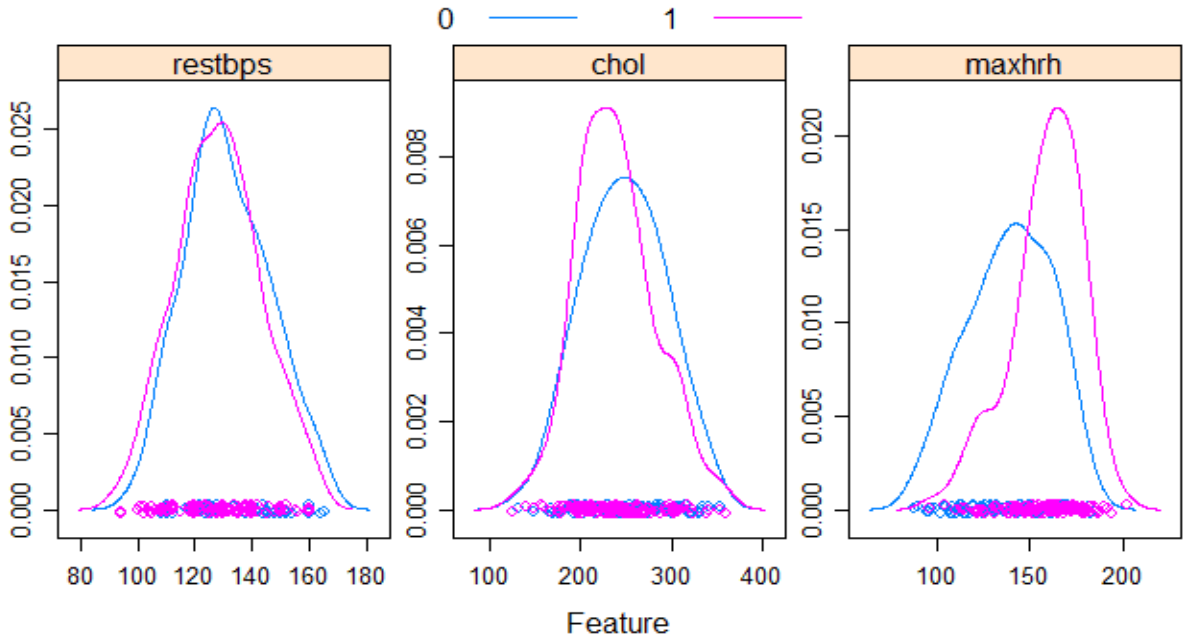
then i created some plot to see better

```
par(mfrow = c(1, 3))
for (i in 1:3) {
  boxplot(x[, i], main = names(dataset)[i])
}
plot(y)
featurePlot(x = x, y = y, plot = "ellipse", auto.key = list(columns = 2))
featurePlot(x = x, y = y, plot = "box")
scales <- list(x = list(relation = "free"), y = list(relation = "free"))
featurePlot(x = x, y = y, plot = "density", scales = scales, auto.key =
list(columns = 2))
```



Scatter Plot Matrix





Create a Test set. We will split the loaded dataset into two:

- 80% of which we will use to *train* our models;

- 20% that we will hold back as a *test* set.

Then set a random seed to ensure that the evaluation of each algorithm is performed using exactly the same data splits.

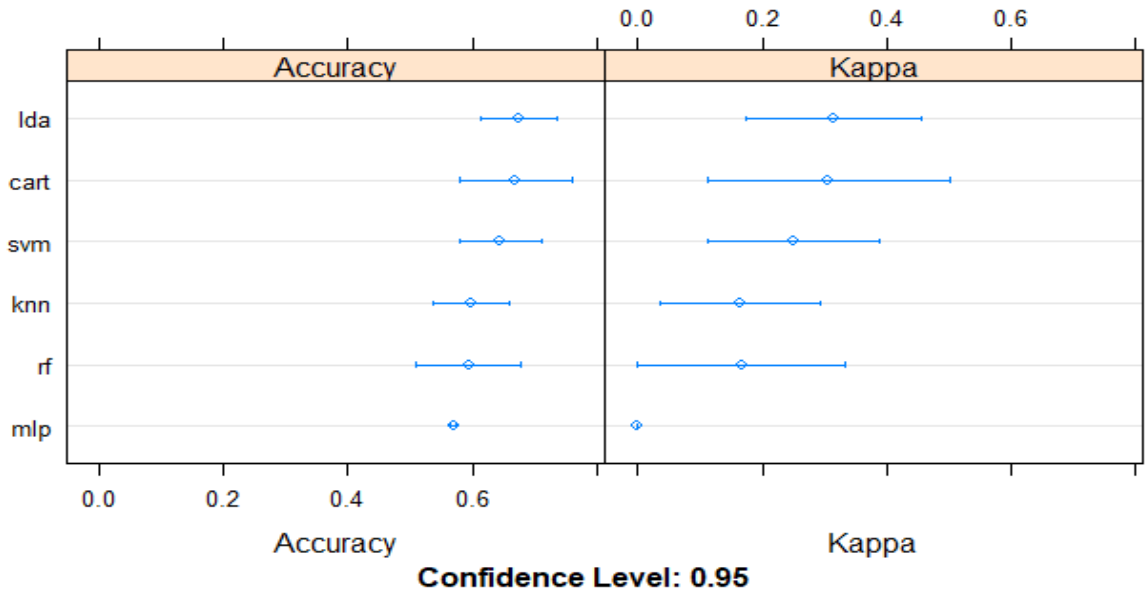
Then run algorithms using 10-fold cross validation.

```
set.seed(1500)
training_index <- createDataPartition(dataset$target, p = .80, list =
FALSE)
training_set <- dataset[training_index, ]
nrow(training_set)
test_set <- dataset[-training_index, ]
nrow(test_set)
summary(test_set)
seed = set.seed(1500)
control <- trainControl(method = "cv", number = 10, seed = seed)
metric <- "Accuracy"
```

i run all these algorithms

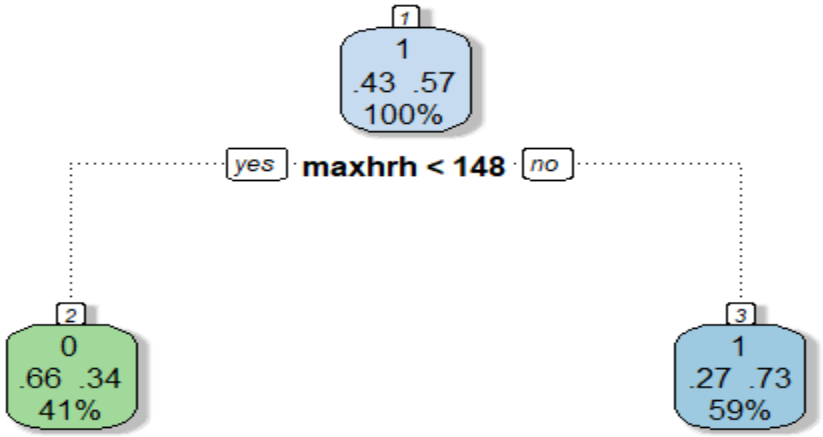
```
# linear algorithms
fit_lda <- train(target ~ ., data = training_set, metric = metric,
trControl = control, method = "lda")
## CART
fit_cart <- train(target ~ ., data = training_set, metric = metric,
trControl = control, method = "rpart")
## kNN
fit_knn <- train(target ~ ., data = training_set, metric = metric,
trControl = control, method = "knn")
## MLP
fit_mlp <- train(target ~ ., data = training_set, metric = metric,
trControl = control, method = "mlp")
## Random Forest
fit_rf <- train(target ~ ., data = training_set, metric = metric, trControl
= control, method = "rf")
## SVM
fit_svm <- train(target ~ ., data = training_set, metric = metric,
trControl = control, method = "svmRadial")

results <- resamples(list(lda = fit_lda, cart = fit_cart, knn = fit_knn,
mlp = fit_mlp, rf = fit_rf, svm = fit_svm))
summary(results)
dotplot(results)
```



After checking the dotplot it seems like lda is the most accurate.

```
fit_lda$results
summary(dataset)
predictions <- predict(fit_lda, test_set)
confusionMatrix(predictions, test_set$target)
library(rattle)
library(rpart.plot)
library(RColorBrewer)
fancyRpartPlot(fit_cart$finalModel)
```



I checked the summary: Accuracy : 0.8235 we can se that we are not in the 98.3% accuracy +- 3% interval so we can say we don't have a reliable and accurate model.

i used the lda algorithms:

The **linear Discriminant analysis** estimates the probability that a new set of inputs belongs to every class. ... **LDA** uses Bayes' Theorem to estimate the probabilities. If the output class is (k) and the input is (x), here is how Bayes' theorem **works** to estimate the probability that the data belongs to each class.