Practical session - Machine Learning Regression models

Mathilde Mougeot, HMVC

April 2025

Goal of the practical session

- Machine Learning Regression models. CART, Bagging, Random Forest, Extra Trees.

Warnings and Advices

- The goal of this practical session is not “just to program with R” but more specifically to understand the framework of Modeling, to learn how to developp appropriate models for answering to a given operationnal question and a given data set. This course belongs to the **Data Science courses** and is a preliminary step before using more advanced methods introduced in other ‘machine learning’ courses. → For each practical session, you should **first understand** the mathematical and statistical backgrounds, **then write your own program with ‘R’** to practically answer to the question.

Remarks

- In order to start any run with a clean environment, the two following lines should be always put in the beginning of your code: `rm(list=ls()); graphics.off()`.

I. The data

The ozone data set.

a) Introduction. The goal of this application is to model the ozone rate (Y variable) given few explanatory variables (temperature, radiation, wind).

Execute the following instruction to upload the data into your R environment and get some information on your dataset.

```r
rm(list=ls()); graphics.off();
tab=read.table('data/ozone.txt', header=T);
head(tab)
print(names(tab)); #p=4 variables
print(dim(tab));
tab[1:5,];
```

b) Visualization of the data set.

```r
plot(tab,main="ozone data",pch='+');
```

c) Relation between variables. Compute and visualize the correlation between your variables with the help of the following instructions. What conclusions can you draw?

```r
library(corrplot);
corrplot(cor(tab))
```

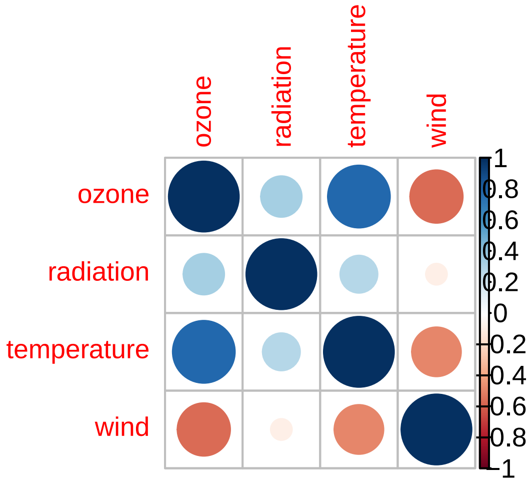


Figure 1: Correlation

c) Study the help of the R functions to personalize your plots

```r
help("cor");
help("corrplot");
```

II. Classification and Regression Trees

First Regression Tree model (on target variable vs on explanatory variable)

a) Install the package Tree in your computer

```
install.packages("tree");
```

Study the help of the package and the help of the package functions before making the following exercices.

```
library(tree);
help("tree");
```

b) `Training a decision tree model`. Execute the following instructions to download the "Tree" library into the R environment, to train the tree model on the ozone dataset. The last instruction let you to "visualize" the decision tree model and the decision nodes.

```
library("tree"); #load the Tree library
mytree=tree('ozone~temperature',data=tab); #Train/compute the Tree regression model
par(cex=0.6);plot(mytree); text(mytree); print(mytree); #model visualization
```

With the help of the function, analyse the information return by function tree

```
attributes(mytree)
```

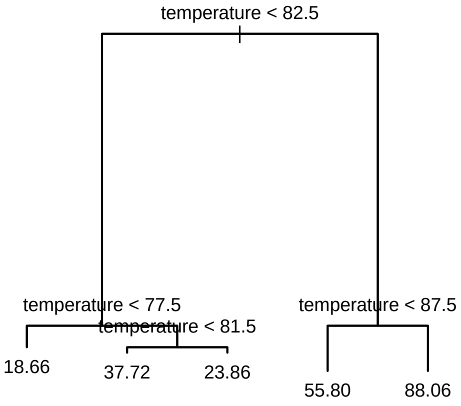


Figure 2: Decision Tree model

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 106 79470.0 37.97
##    2) temperature < 82.5 75 13720.0 23.72
##      4) temperature < 77.5 50  4093.0 18.66 *
##      5) temperature > 77.5 25  5781.0 33.84
##       10) temperature < 81.5 18  4506.0 37.72 *
##       11) temperature > 81.5 7   306.9 23.86 *
##    3) temperature > 82.5 31 13670.0 72.45
##      6) temperature < 87.5 15  3042.0 55.80 *
##      7) temperature > 87.5 16  2569.0 88.06 *
```

b) `Prediction computations on new input data given the pre-trained decision Tree model.`

```
#Visualisation of the prediction model
newx=seq(min(tab$temperature),max(tab$temperature),0.5); #new data
py=predict(mytree,list(temperature=newx)); #prediction on new input data
plot(tab$temperature,tab$ozone,pch=16); #visualisation on the new prediction function(in red)
lines(newx,py,col='red',lwd=2);
points(newx,py,col='gray',pch='x');
grid()
```

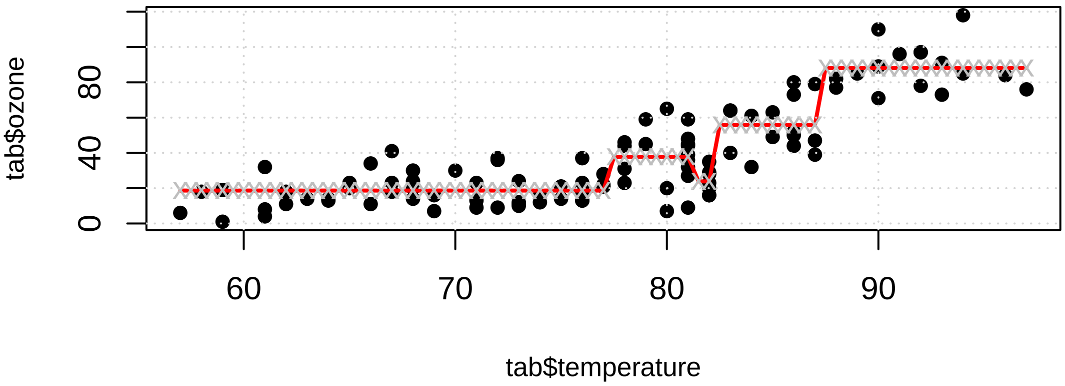


Figure 3: Decision Tree model. Prediction model

b) Evaluation on the pre-trained decision Tree model given Test (input,output) data .

```
ptrain=0.75; #Proportion of data to train the model
indtrain=sample(nrow(tab),trunc(ptrain*nrow(tab)),replace=F); #train indices
indtest=-indapp; #test indices data

#Machine learning model
mytree=tree('ozone~temperature',data=tab[indtrain,]); #trained model
ytest=predict(mytree,newdata=tab[indtest,]); #Prediction on new data

ytrain=predict(mytree,newdata=tab[indtrain,]); #Prediction on train data
EtrainTree=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainTree) #Train error
EtestTree=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(EtestTree) #Test Error
```

c) Visualization of the train and test predictions for the decision tree model

```
par(mfrow=c(2,1),mar=c(4,4,2,2))
plot(tab$temperature,tab$ozone,pch='+',cex=2,col="gray",main="Decision Tree model: ozone vs temperature",xlab
points(tab[indtrain,"temperature"],ytrain,pch="x",col="red",cex=2);
points(tab[indtest,"temperature"],ytest,pch="*",col="blue",cex=2.5);
grid();

plot(tab[indapp,"ozone"],ytrain,pch="x",col="red",cex=2,xlab="y",ylab="ypred",main="Pairwise plot");
points(tab[indtest,"ozone"],ytest,pch="*",col="blue",cex=2.5);
abline(a=0,b=1,col="gray",lwd=2)
grid();
```

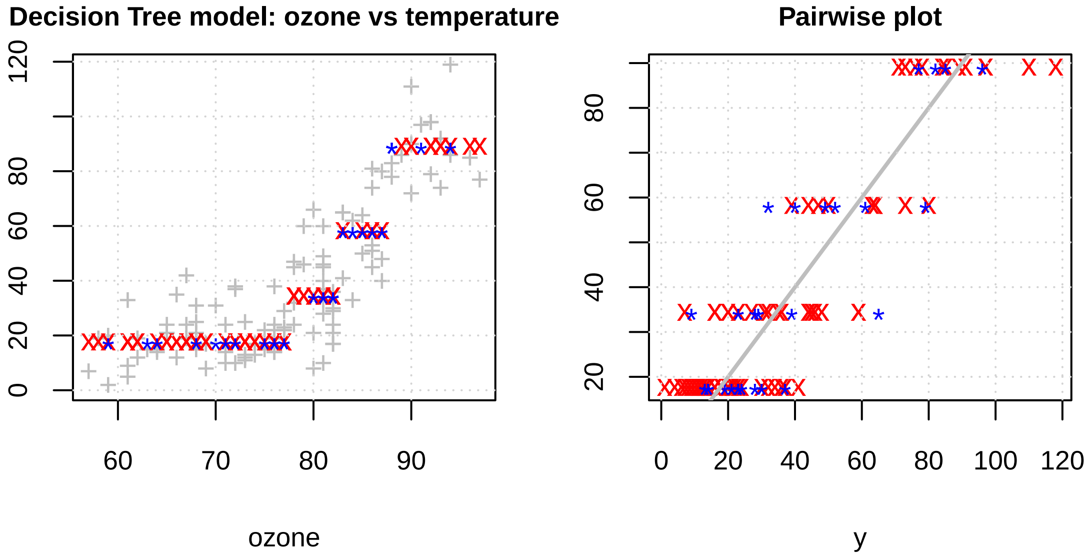


Figure 4: Decision Tree model. Prediction on test and train data

Regression Tree model (on target variable vs several explanatory variables)

```
mytree=tree('ozone~.',data=tab); #model
par(cex=0.8);plot(mytree); text(mytree); print(mytree); #tree model visalization
```

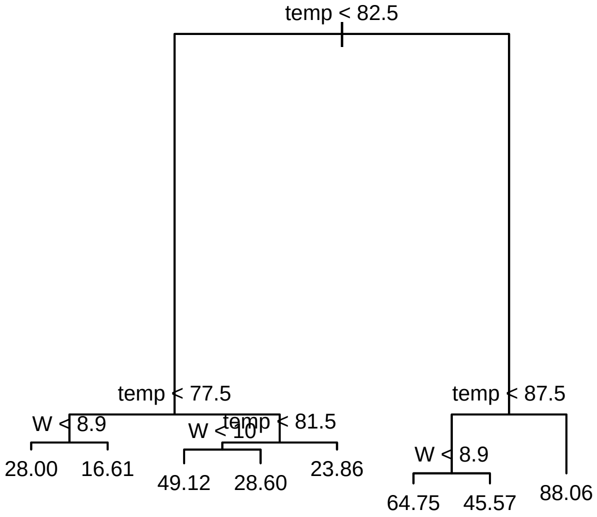


Figure 5: Decision Tree model

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 106 79470.0 37.97
##    2) temp < 82.5 75 13720.0 23.72
##      4) temp < 77.5 50  4093.0 18.66
##        8) W < 8.9 9   612.0 28.00 *
##        9) W > 8.9 41  2524.0 16.61 *
##      5) temp > 77.5 25  5781.0 33.84
##       10) temp < 81.5 18  4506.0 37.72
##         20) W < 10 8   870.9 49.12 *
##         21) W > 10 10  1762.0 28.60 *
##       11) temp > 81.5 7   306.9 23.86 *
##    3) temp > 82.5 31 13670.0 72.45
##      6) temp < 87.5 15  3042.0 55.80
##       12) W < 8.9 8  1052.0 64.75 *
##       13) W > 8.9 7   617.7 45.57 *
##      7) temp > 87.5 16  2569.0 88.06 *
```

#Decision Tree model: train and test evaluation

```
ptrain=0.75; #Proportion of data to train the model
indtrain=sample(nrow(tab),trunc(ptrain*nrow(tab)),replace=F); #train indices
indtest=-indtrain; #test indices data

mytrain=tree('ozone~.',data=tab[indtrain,]);
ytrain=predict(mytree,newdata=tab[indtrain,]);
EtrainTree=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainTree)
ytest=predict(mytree,newdata=tab[indtest,]);
EtestTree=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(EtestTree)
```

III. Bagging and Random Forest models

a) Install the useful packages in your computer

```
install.packages("ipred");
install.packages("randomForest");
```

Study the help of the package and the help of the package functions before making the following exercices.

```
library(ipred);
help("bagging");
library(randomForest);
help("randomForest");
```

b) Bagging model: train and test evaluation

```
#Bagging
library(ipred);
mybag = bagging(ozone~.,data = tab[indtrain,],coob = TRUE); #apprentissage
ytrain = predict(mybag, newdata = tab[indtrain,]); #prediction sur le feuillet test
Etrainbag=sqrt(mean((tab[indapp,"ozone"]-ytrain)^2)); print(Etrainbag)
ytest = predict(mybag, newdata = tab[indtest,]); #prediction sur le feuillet test
Etestbag=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(Etestbag)
```

c) RandomForest model: train and test evaluation

```
#RandomForest
library(randomForest);
myRF = randomForest(ozone~.,data = tab[indtrain,],coob = TRUE); #apprentissage
ytrain = predict(myRF, newdata = tab[indtrain,]); #prediction sur le feuillet test
EtrainRF=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainRF)
ytest = predict(myRF, newdata = tab[indtest,]); #prediction sur le feuillet test
EtestRF=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(EtestRF)
```

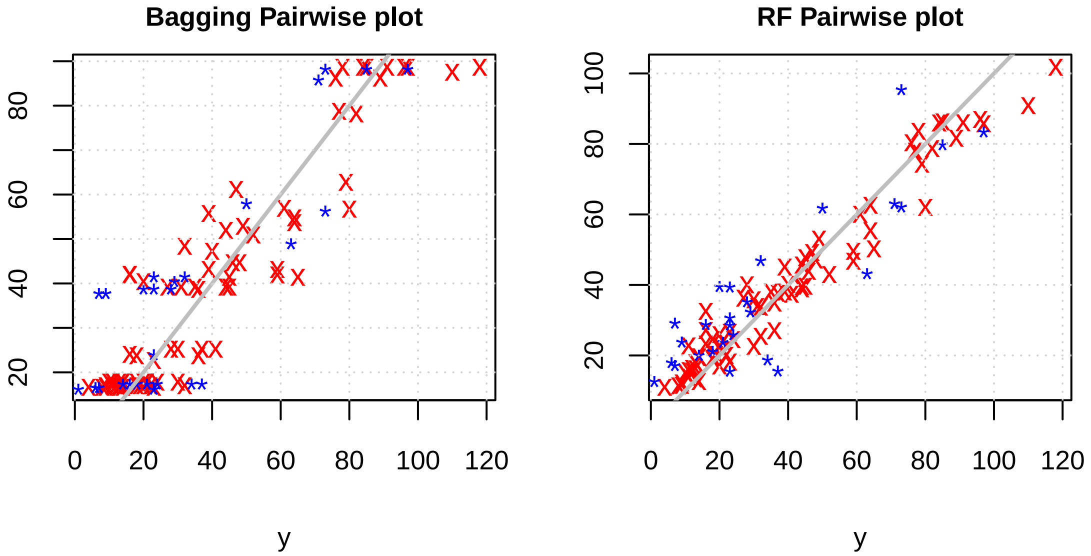


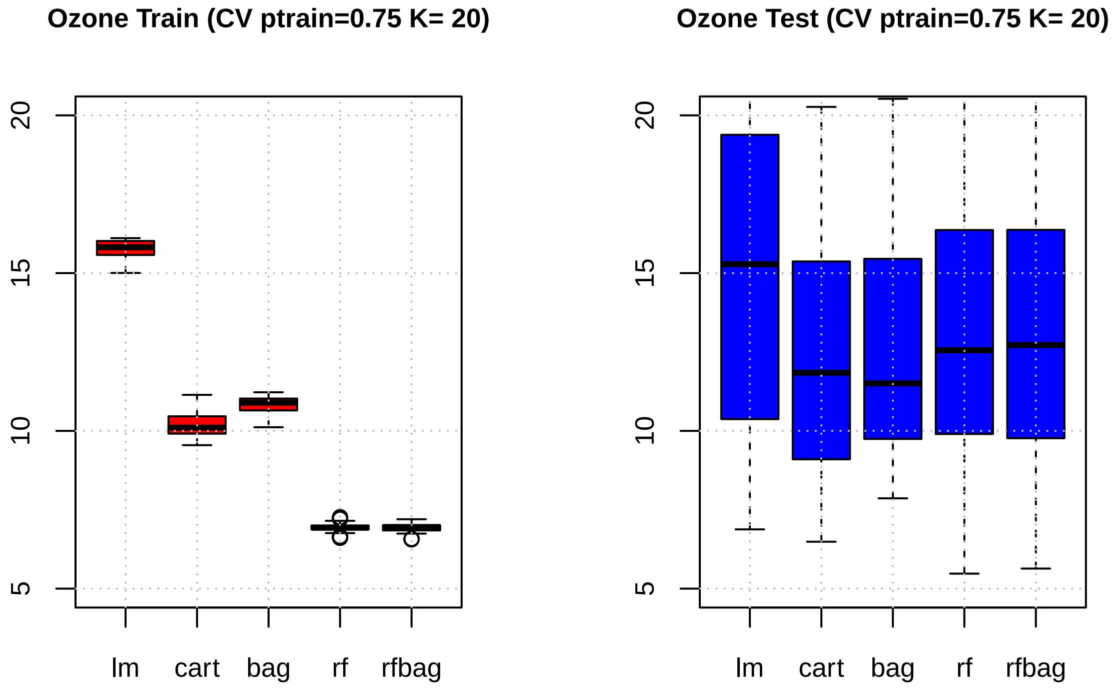
Figure 6: Bagging and Random Forest models. Prediction on test and train data

```
par(mfrow=c(1,1))
varImpPlot(myRF)
```

III. Model comparison with K fold cross validation

```
tab=read.table('data/ozone.txt', header=T);
K=50; #for example
nK=trunc(nrow(tab)/K)
tabrestest=data.frame(matrix(nrow=K,ncol=5));
names(tabrestest)=c('lm','cart','bag','rf','rfbag')
tabrestrain=tabrestest;

indall=nK*K; tab=tab[sample(1:indall),]
for (k in 1:K)
{
indtest=(1+(k-1)*nK):(k*nK);
indtrain=-indtest;

#Decision Tree model
mytree=tree('ozone~.',data=tab[indtrain,]);
ytrain=predict(mytree,newdata=tab[indtrain,]);
EtrainTree=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainTree)
ylm=predict(mytree,newdata=tab[indtest,]);
EtestTree=sqrt(mean((tab[indtest,"ozone"]-ylm)^2));print(EtestTree)

#Linear model
mylm=lm(ozone~.,data=tab[indtrain,]);  summary(mylm)
ytrain=predict(mylm,newdata=tab[indtrain,]);
EtrainLm=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainLm)
ylm=predict(mylm,newdata=tab[indtest,]);
EtestLm=sqrt(mean((tab[indtest,"ozone"]-ylm)^2));print(EtestLm)

#Bagging
mybag = bagging(ozone~.,data = tab[indtrain,],coob = TRUE); #apprentissage
ytrain = predict(mybag, newdata = tab[indtrain,]); #prediction sur le feuillet test
Etrainbag=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(Etrainbag)
ytest = predict(mybag, newdata = tab[indtest,]); #prediction sur le feuillet test
Etestbag=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(Etestbag)

#RAndomForest
myRF = randomForest(ozone~.,data = tab[indtrain,],coob = TRUE); #apprentissage
ytrain = predict(myRF, newdata = tab[indtrain,]); #prediction sur le feuillet test
EtrainRF=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainRF)
ytest = predict(myRF, newdata = tab[indtest,]); #prediction sur le feuillet test
EtestRF=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(EtestRF)

#RandomForest
myRF = randomForest(ozone~.,data = tab[indtrain,],coob = TRUE,mytry=ncol(tab)-1); #apprentissage
ytrain = predict(myRF, newdata = tab[indtrain,]); #prediction sur le feuillet test
EtrainRFbag=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(EtrainRF)
ytest = predict(myRF, newdata = tab[indtest,]); #prediction sur le feuillet test
EtestRFbag=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(Etestbag)

restrain=c(EtrainLm,EtrainTree,Etrainbag,EtrainRF,EtrainRFbag)
restest=c(EtestLm,EtestTree,Etestbag,EtestRF,EtestRFbag)
tabrestest[k,]=restest;
tabrestrain[k,]=restrain;
}
par(mfrow=c(2,1))
boxplot(tabrestrain,cex.axis=2,main=sprintf('Ozone Train (CV ptrain=0.75 K=%3.0f)',K),
        cex.main=2,ylim=c(5,20),col="red");  grid(col='gray');
```

```
boxplot(tabrestest,cex.axis=2,main=sprintf('Ozone Test (CV ptrain=0.75 K=%3.0f)',K),
        cex.main=2,ylim=c(5,20),col="blue"); grid(col='gray');
```



IV. Investigation hyperparameter model. The caret package

```r
#RandomForest
library(gbm)
ptrain=0.75; #Proportion of data to train the model
indtrain=sample(nrow(tab),trunc(ptrain*nrow(tab)),replace=F); #train indices
indtest=-indtrain; #test indices data

mygbm = gbm(ozone~.,data = tab[indtrain,],interaction.depth=2);
ytrain = predict(mygbm, newdata = tab[indtrain,]);
Etraingbm=sqrt(mean((tab[indtrain,"ozone"]-ytrain)^2)); print(Etraingbm)
ytest = predict(mygbm, newdata = tab[indtest,]);
Etestgbm=sqrt(mean((tab[indtest,"ozone"]-ytest)^2));print(Etestgbm)
```

“‘