

Lecture 3: Predicting Miles per Gallon

Isabel Casas
icasas@deusto.es

Lecture content

We have learnt:

- How to summarise, visualise and clean the *MPG* data
- How to use R to fit a linear regression and regression tree to the MPG data (estimation)

We will learn:

- How to evaluate the estimation errors using some error measures
- How to use R to predict values from a newdataset
- How to evaluate the prediction errors
- Same with a random forest

We will use the following R packages: `rpart`, `rpart.plot`, `ggplot2`, `randomForest`. Install them once if you have not done so.

```
install.packages (c("rpart", "rpart.plot",  
                    "ggplot2", "randomForest"))
```

Load the data

- Upload dataset *mpg_new.csv* that contains the MPG data all cleaned
- In my case, I am working in directory *Lecture03* and the file is in directory *Lecture02/data/*, I upload the table to my file using *read.table()*
- Which are your working directories?

```
mpg.data <- read.table("../Lecture02/data/mpg_new.csv", header = TRUE,  
                        sep = ";")
```

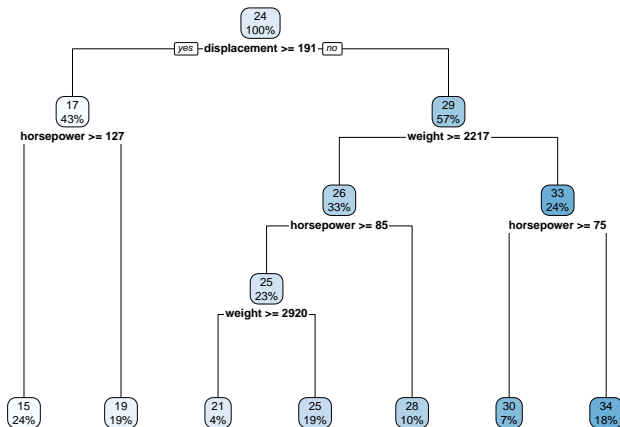
Section 1

Estimating MPG with linear regression and regression tree

Fitting linear regression and tree (Lecture 2)

```
lm.mpg4 <- lm (mpg ~ 1 + horsepower + weight, data = mpg.data)
library(rpart)
rt.mpg <- rpart(mpg ~ ., data = mpg.data[, 1:6])
```

Plotting a regression tree



Pruning a regression tree

The tree algorithm tend to overfit the data (too many nodes). There are different ways of controlling this:

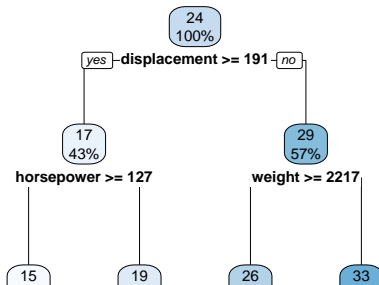
- 1 using arguments *cp*, *minsplit* and *maxdepth* in function *rpart* to make it stop earlier
- 2 using function *prune* which uses an algorithm to prune the tree depending of which value we give it in *cp*

There is no rule of thumb of which is the best pruning, it is up to the knowledge of the research team to choose the best tree.

Pruning a decision tree

By default $cp = 0.01$, we impose $cp = 0.04$ and it will create less nodes (displacement, horsepower and weight)

```
rt.mpg2 <- prune(rt.mpg, cp = 0.04)
rpart.plot(rt.mpg2, roundint = FALSE)
```



Trained model evaluation

First discuss the difference between estimating (or fitting) a model and predicting of values using a model

- ➊ **Estimation** (prediction with the sample used to train the model)
 - We use the same dataset *mpg_new.csv* for training and predicting.
 - Evaluate how well the model fits this particular dataset: R^2 , residual plots and error measures (*mpg - mpg.fitted*), such as the mean absolute error (MAE) and root mean squared error (RMSE)
- ➋ **Prediction** (prediction with a new sample)
 - We use the trained model on the testing dataset, *mpg_predict.txt*, generating predicted values of the response/dependent variable (*mpg*)
 - Evaluate the prediction using the prediction error is *mpg.true - mpg.predict* to calculate their MAE and RMSE.

Trained model evaluation

Which of the two models (regression or tree) fit the *mpg* training dataset best?

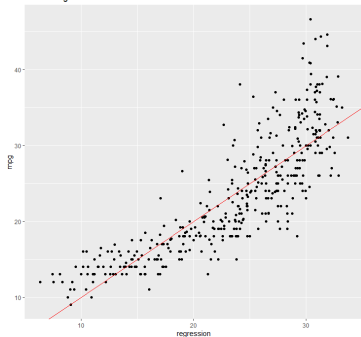
- 1 Get the fitted values and residuals from each model
- 2 Calculate loss functions (measures of fitness):
 - mean absolute error (MAE)
 - root mean squared error (RMSE)
 - there are other measures of fitness (check the book)
- 3 Compare the measures of fitness of each model

Trained model evaluation

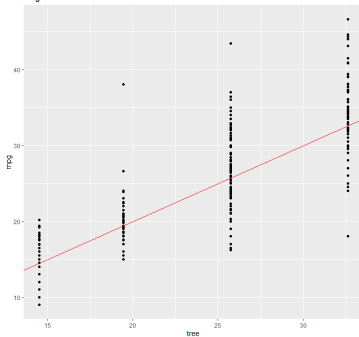
```
library(ggplot2)
dg <- data.frame(regression = fitted(lm.mpg4),
                 tree = predict(rt.mpg2, newdata = mpg.data),
                 mpg = mpg.data$mpg)
ggplot(dg, aes(x=regression, y=mpg)) +
  geom_point() + geom_abline(slope=1, intercept=0, color="red") +
  ggtitle("Linear Regression")
ggplot(dg, aes(x=tree, y=mpg)) +
  geom_point() + geom_abline(slope=1, intercept=0, color="red") +
  ggtitle("Regression Tree")
```

Trained model graphical evaluation

Linear Regression



Regression Tree



Trained model evaluation

The error can be calculated from the residuals.

```
lm.resid <- residuals(lm.mpg4)
rt.resid <- residuals(rt.mpg2)
```

Calculate loss functions

Use residuals to calculate the MAE and RMSE (there are other measures of accuracy, but these are the two most common). The smaller value means the better the fitness.

```
cat("MAE \n")
lm.mae <- mean(abs(lm.resid))
lm.mae
rt.mae <- mean(abs(rt.resid))
rt.mae
cat("RMSE \n")
lm.rmse <- sqrt(mean(lm.resid^2))
lm.rmse
rt.rmse <- sqrt(mean(rt.resid^2))
rt.rmse
```

Calculate loss functions

Use residuals to calculate the MAE and RMSE (there are other measures of accuracy, but these are the two most common). The smaller value means the better the fitness.

```
## MAE
```

```
## [1] 3.254789
```

```
## [1] 2.992863
```

```
## RMSE
```

```
## [1] 4.2489
```

```
## [1] 4.081866
```

Section 2

Prediction with linear regression and regression tree

Prediction power

We are interested in evaluating the model prediction power too. How do we get the prediction error for the test dataset?

- 1 predict values of *mpg* using the predictors from the test sample
 - File *mpg_predict.txt* in ALUD has 50 values with predictors and *mpg*
- 2 calculate the prediction error
- 3 calculate the prediction MAE and RMSE using those prediction errors and compare the two models

Upload test file

The first thing is to check if the test dataset has missing values: - removing missing values, we have to make sure we remove the same line in file algae.sols, or - fill-in the missing values

```
# test mpg
mpg.test <- read.table("../Lecture02/data/mpg_predict.txt",
                        sep=";", header = TRUE)

names(mpg.test)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "model_year"   "origin"
```

```
dim(mpg.test)
```

```
## [1] 50 8
```

```
anyNA(mpg.test)
```

```
## [1] TRUE
```

Prediction of values in test sample using the estimated models

Predicted and true values using *mpg.test* dataset

```
mpg.true <- mpg.test$mpg  
lm.mpg.pred <- predict(lm.mpg4, newdata = mpg.test)  
rt.mpg.pred <- predict(rt.mpg2, newdata = mpg.test)
```

Prediction error

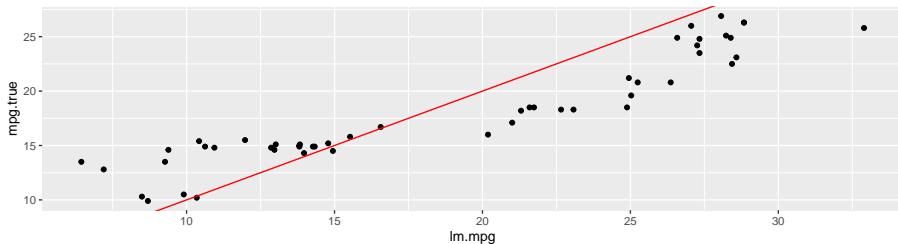
```
lm.pred.error <- mpg.true - lm.mpg.pred  
rt.pred.error <- mpg.true - rt.mpg.pred
```

Prediction graphical evaluation

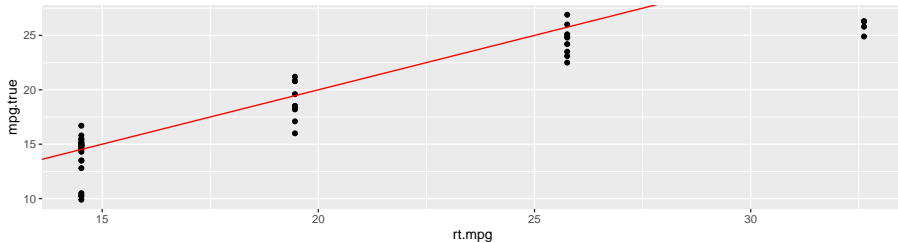
```
library(ggplot2)
par(mfrow = c(1, 2))
dg <- data.frame(lm.mpg = lm.mpg.pred,
                  rt.mpg = rt.mpg.pred,
                  mpg = mpg.true)
ggplot(dg, aes(x=lm.mpg, y=mpg)) +
  geom_point() + geom_abline(slope=1, intercept=0, color="red")
ggtitle("Linear Model")
ggplot(dg, aes(x=rt.mpg, y=mpg)) +
  geom_point() + geom_abline(slope=1, intercept=0, color="red")
ggtitle("Regression Tree")
```

Prediction graphical evaluation

Linear Model



Regression Tree



Calculate the prediction MAE and RMSE

```
cat("MAE of lm prediction: ", mean(abs(lm.pred.error)))
```

```
## MAE of lm prediction: 3.005797
```

```
cat("\nMAE of rf prediction: ", mean(abs(rt.pred.error)))
```

```
##
```

```
## MAE of rf prediction: 1.773308
```

```
cat("\nRMSE of lm prediction: ",sqrt(mean(lm.pred.error^2)))
```

```
##
```

```
## RMSE of lm prediction: 3.587907
```

```
cat("\nRMSE of rf prediction: ",sqrt(mean(rt.pred.error^2)))
```

```
##
```

```
## RMSE of rf prediction: 2.588912
```

Section 3

Estimation and prediction of MPG with randomforest

Randomforest

A very popular machine learning algorithm is called the *randomforest*, which basically run several versions of trees and get the best fit.

It solves the inaccuracy problem of regression/decision trees.

Video: [Randomforest](#) Video: [Randomforest in R](#)

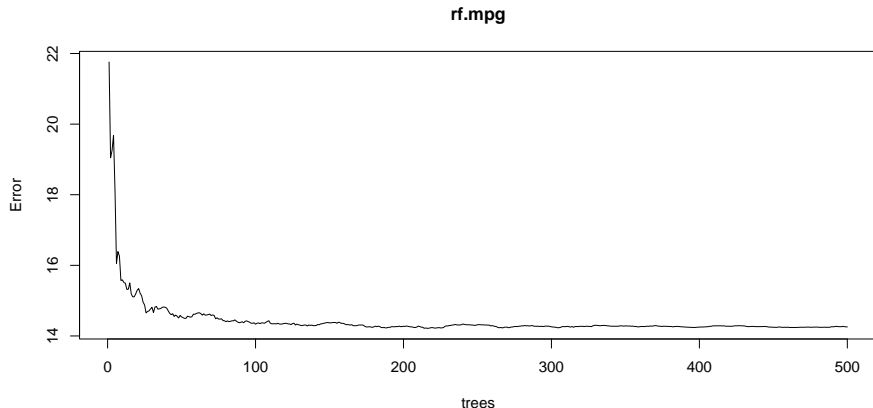
Randomforest estimation

```
library(randomForest)
rf.mpg <- randomForest(mpg ~ ., data = mpg.data[, 1:6],
                       importance = TRUE)
print(rf.mpg)
```

```
##
## Call:
## randomForest(formula = mpg ~ ., data = mpg.data[, 1:6], importance = TR
##               Type of random forest: regression
##               Number of trees: 500
## No. of variables tried at each split: 1
##
##               Mean of squared residuals: 14.25437
##               % Var explained: 76.61
```

Randomforest error/n.trees

```
plot(rf.mpg)
```



- It looks like 70 trees is the best choice
- Let us run the model again with those settings

Randomforest estimation

```
library(randomForest)
rf.mpg2 <- randomForest(mpg ~ ., data = mpg.data[, 1:6],
                        importance = TRUE, ntree = 70)
print(rf.mpg2)
```

```
##
## Call:
## randomForest(formula = mpg ~ ., data = mpg.data[, 1:6], importance = TR
##           Type of random forest: regression
##           Number of trees: 70
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 15.02996
##           % Var explained: 75.33
```

Randomforest prediction

```
rf.mpg.pred <- predict(rf.mpg, newdata = mpg.test)
rf.pred.error <- mpg.true - rf.mpg.pred
cat ("MAE of RF prediction: ", mean(abs(rf.pred.error)), "\n")
cat ("RMSE of RF prediction: ", sqrt(mean(rf.pred.error^2)), "\n")
```

- Which is the best model at predicting this data?
- Are we sure?

Randomforest prediction error

MAE of RF prediction: 1.421827

RMSE of RF prediction: 1.883778

- Which is the best model at predicting this data?
- Are we sure?

Homework

Watch first 6 minutes of [Theory of linear regression](#)