Machine Learning

Basel R Bootcamp

www.therbootcamp.com

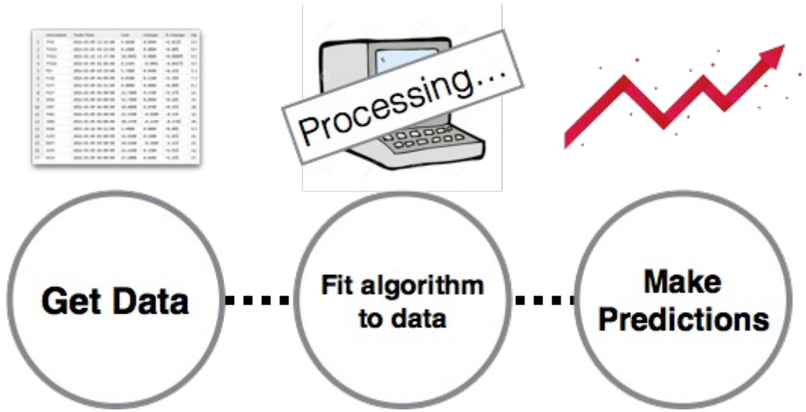
@therbootcamp

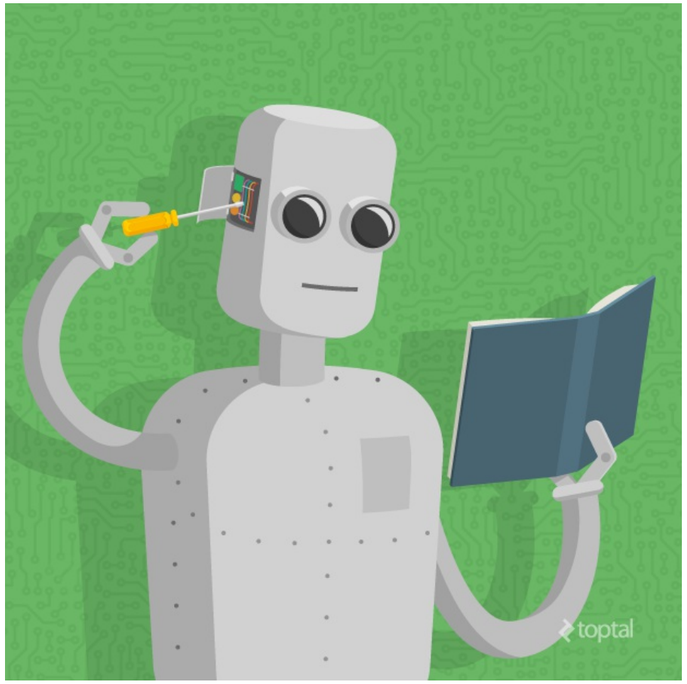
July 2018

What is machine learning?

Algorithms autonomously learning from data.

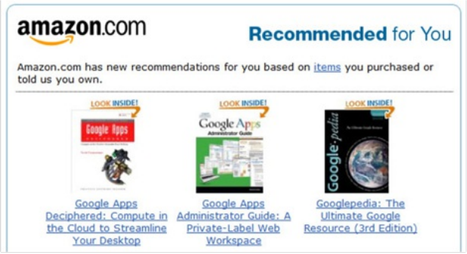
Given data, an algorithm tunes its **parameters** to match the data, understand how it works, and make predictions for what will occur in the future.



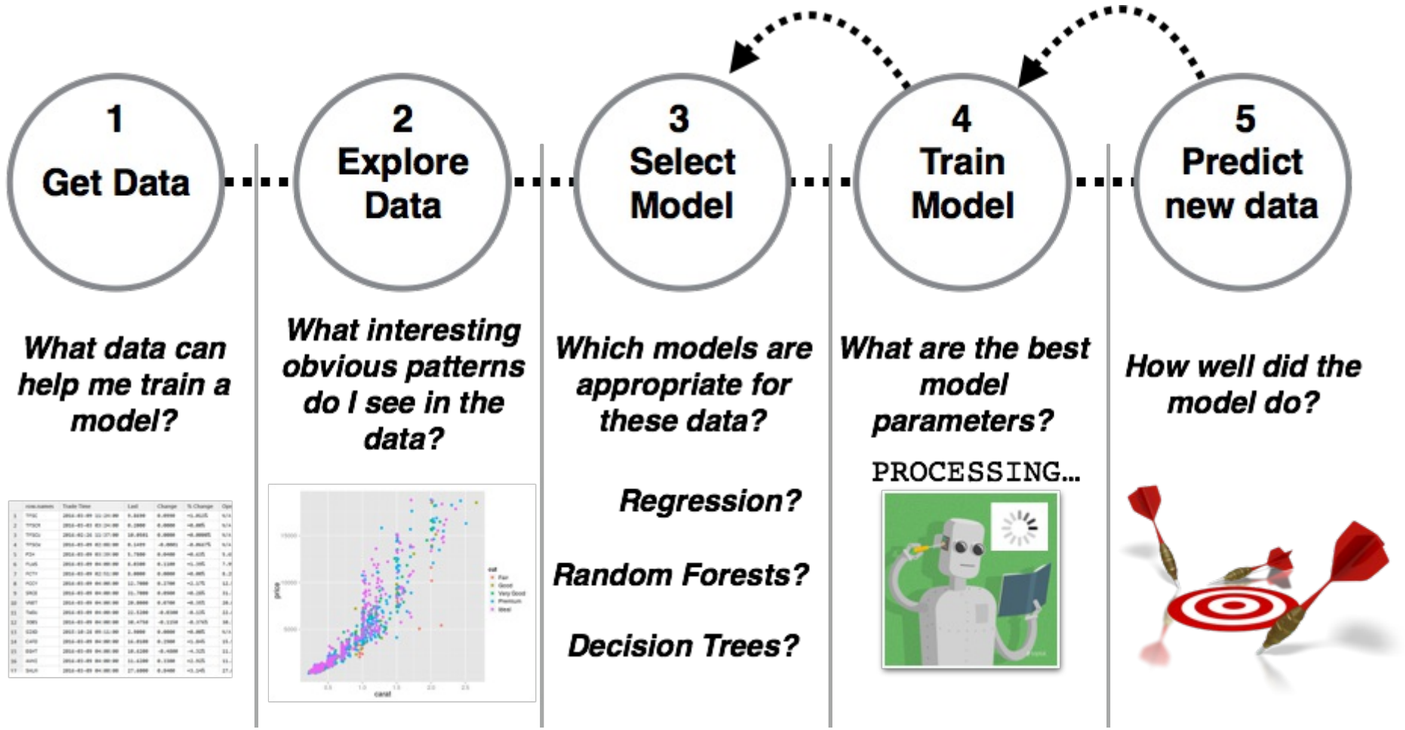


Everyone uses machine learning

> "Machine learning drives our algorithms for demand forecasting, product search ranking, product and deals recommendations, merchandising placements, fraud detection, translations, and much more."
>
> Jeff Bezos, founder of Amazon



What is the basic machine learning process?



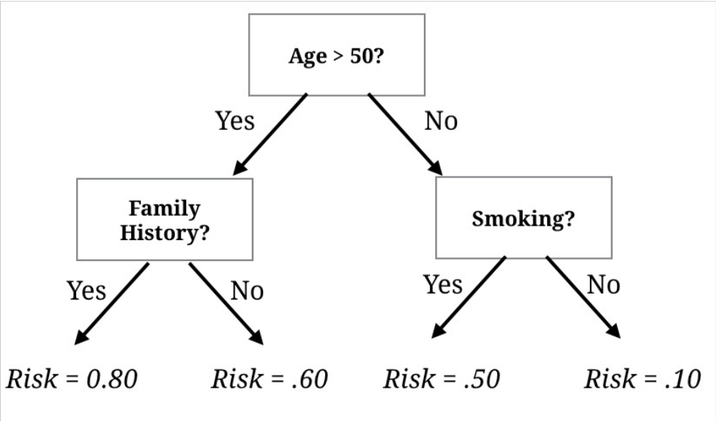
What is a model?

A model is a **formal** (mathematical) procedure describing the relationships between variables.

Most data have one main **criterion** or variable of interest, and several **features**.

id	sex	age	fam_history	smoking	disease
1	m	45	No	FALSE	0
2	m	43	Yes	FALSE	1
3	f	40	Yes	FALSE	1
4	m	51	Yes	FALSE	1
5	m	44	No	TRUE	0

Decision Tree



Weighted Additive (Regression)

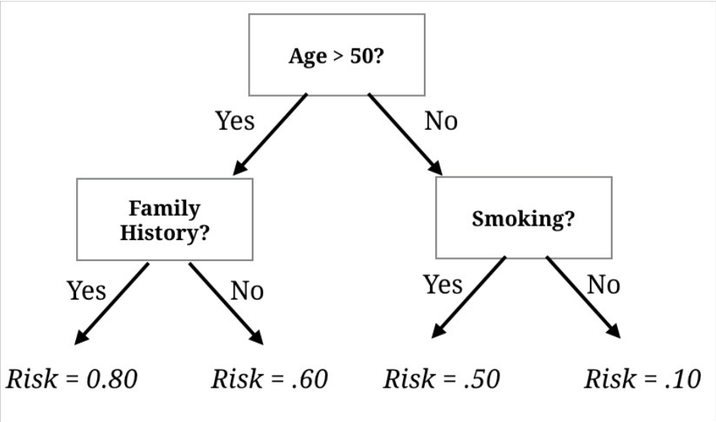
What is model training?

Model **training** (aka, fitting) is the process of matching a model's **parameters** to a specific dataset.

Q: What are the parameters in the two models on the right?

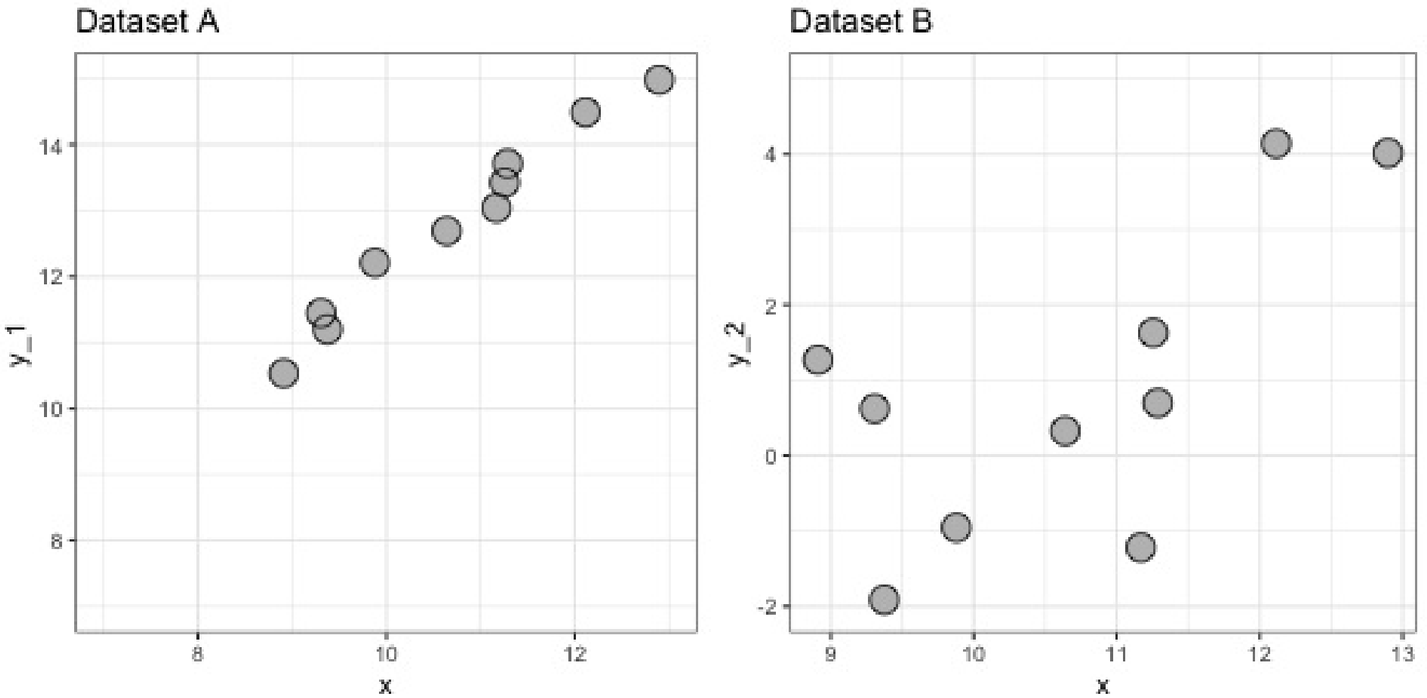
id	sex	age	fam_history	smoking	disease
1	m	45	No	FALSE	0
2	m	43	Yes	FALSE	1
3	f	40	Yes	FALSE	1
4	m	51	Yes	FALSE	1
5	m	44	No	TRUE	0

Decision Tree

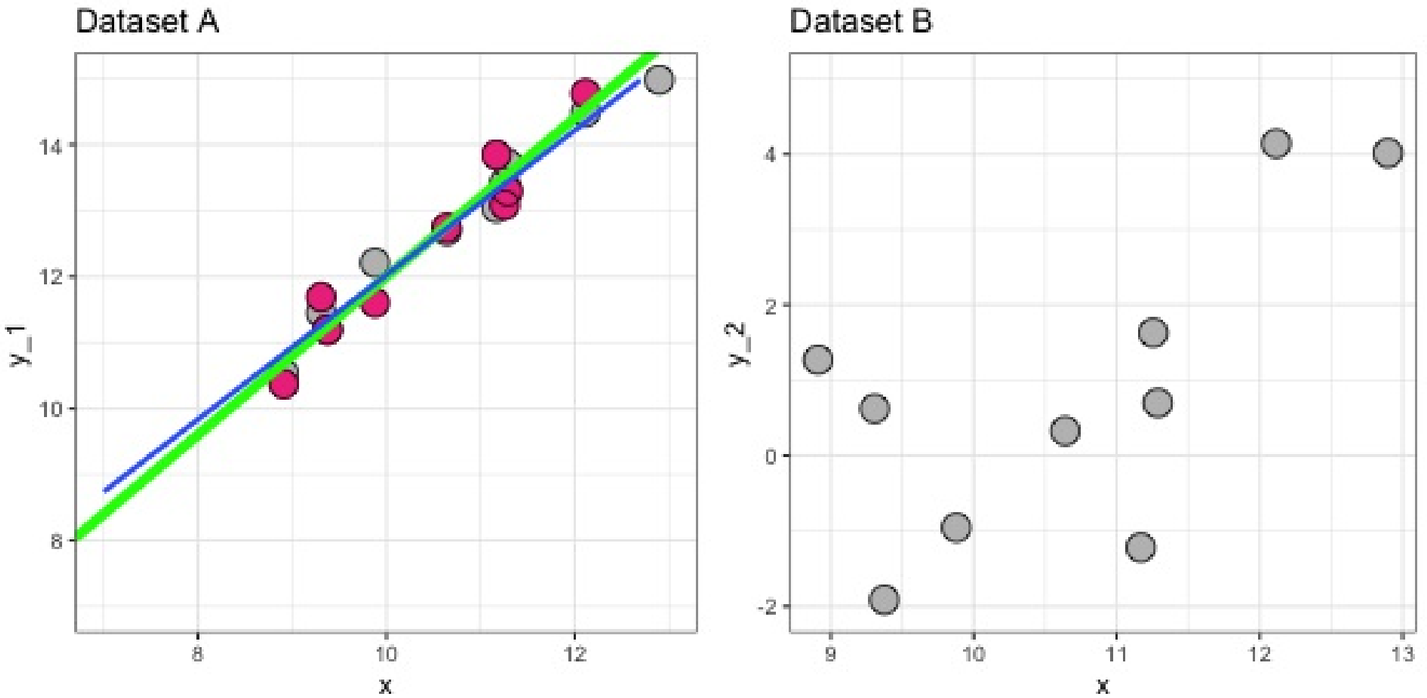


Weighted Additive (Regression)

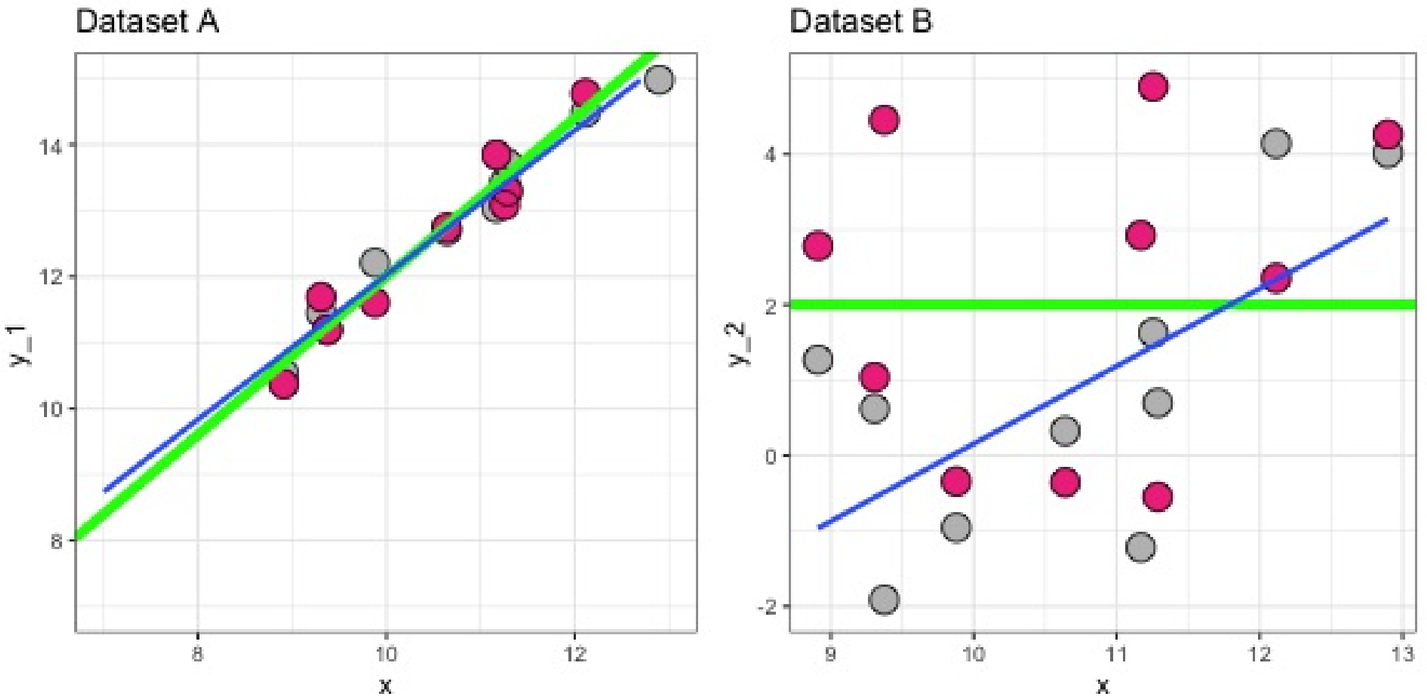
Fit your own linear model!



Fit your own linear model!



Fit your own linear model!

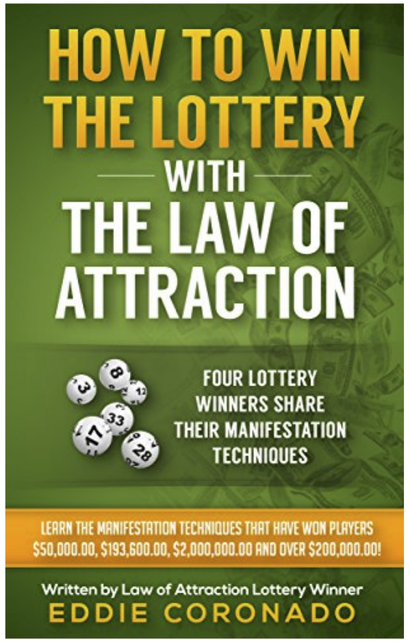


Why do we separate training from prediction?

Just because a model can **fit past data well**, does necessarily mean that it will **predict new data well**.

Anyone can come up with a model of past data (e.g.; stock performance, lottery winnings).

Predicting what you can't see in the future is much more difficult.



"Can you come up with a model that will perfectly match past data but is worthless in predicting future data?"

Past **Training** Data

id	sex	age	fam_history	smoking	disease
1	m	45	No	FALSE	0
2	m	43	Yes	FALSE	1
3	f	40	Yes	FALSE	1
4	m	51	Yes	FALSE	1
5	m	44	No	TRUE	0

Future **Test** Data

id	sex	age	fam_history	smoking	disease
91	m	51	Yes	TRUE	?
92	f	47	No	TRUE	?
93	m	39	No	TRUE	?
94	f	51	Yes	TRUE	?
95	f	50	Yes	FALSE	?

Two types of prediction tasks

Classification Task

x1	x2	x3	y
345	0	0.05	A
654	0	0.23	A
745	1	0.86	C
325	0	0.43	?

← Trying to predict a discrete **Category**

Is that a pedestrian?

What kind of consumer is this?

Heart attack?

Regression Task

x1	x2	x3	y
345	0	0.05	23K
654	0	0.23	15K
745	1	0.86	10K
325	0	0.43	?

← Trying to predict a continuous **Number**

What will the price of bitcoin be in a year?

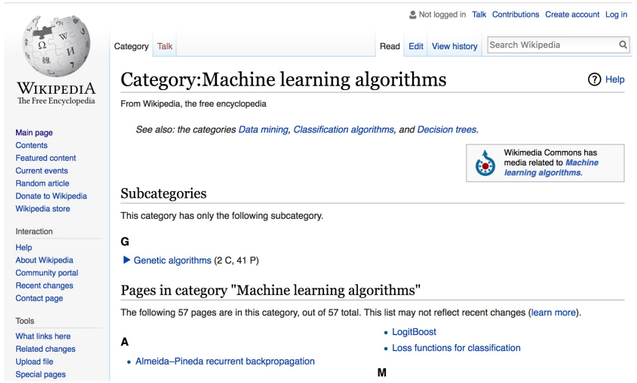
What will the orange yield be?

How expensive will this project be?

What machine learning algorithms are there?

There are thousands of machine learning algorithms from many different fields.

Wikipedia lists 57 categories of machine learning algorithms:



Algorithims we focus on

We will focus on 3 algorithms that apply to most ML tasks:

Algorithm	Complexity
Decision Trees	Low
Regression	Low / Medium
Random Forests	High

How do you fit and evaluate ML models in R?

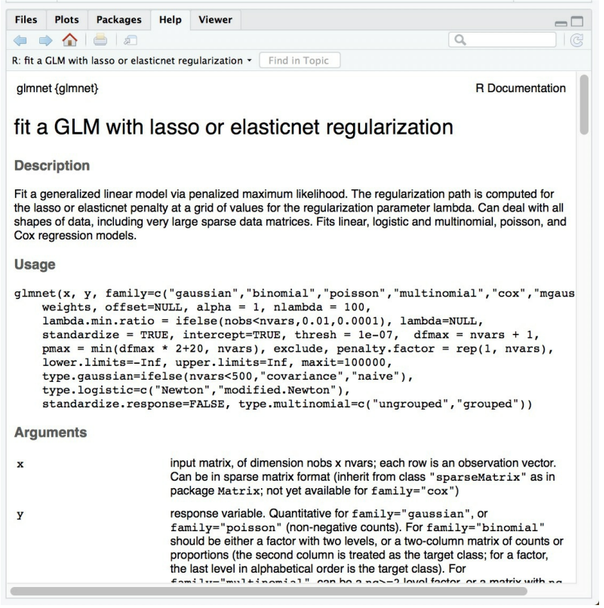
ML models work the same way you fit standard statistical models. Install the package, load, and find the main fitting functions.

```
# Install the glmnet package
install.packages("glmnet")

# Load glmnet
library(glmnet)

# Look at help menu
?glmnet
```

Note: Some functions will use the standard `FUN(formula, data)` arguments, but others (like `glmnet()`) require other arguments, like `x, y` (numeric matrices).

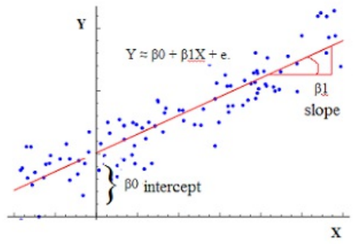


Regression

In regression, the criterion is modeled as the **sum of predictors times weights , .**

Loan example
For instance, one could model the risk of defaulting on a loan as:

Training a model means finding values of and that 'best' match the training data.



Regression with glm()

The `glm()` function in the base stats package performs standard regression

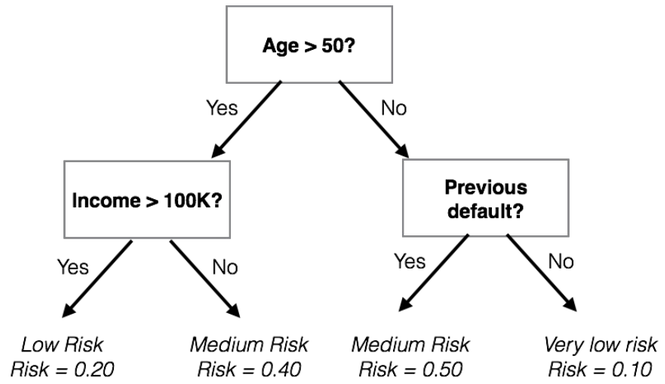
```
# Standard linear regression
glm_mod <- glm(formula = happiness ~ .,
               data = baselers)

# Logisitic regression with family = 'binomial'
glm_mod <- glm(formula = sex ~ .,
               data = baselers.
               family = "binomial")
```

Decision Trees

In decision trees, the criterion is modeled as a **sequence of logical YES or NO questions**.

Loan example



Decision trees with rpart

This codes runs decision trees with functions from the `rpart`-package.

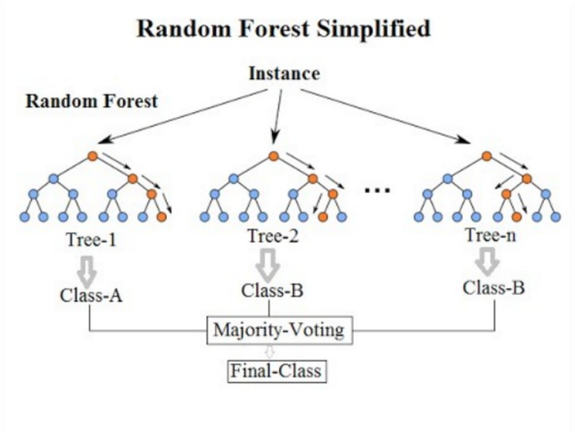
```
install.packages("rpart")
library(rpart)

# Train rpart model
loan_rpart_mod <- rpart(formula, data,
                        method = "class",
                        rpart.control)
```

Random Forest

In **Random Forest**, the criterion is models as the **aggregate prediction of a large number of decision trees** each based on different features.

Loan example



Source

Random Forests with `randomforest`

```
install.packages("randomForest")
library(randomForest)

# Create a randomForest model
randomForest(formula = y ~.,    # Formula
             data = data_train, # Training data
             ntree, mtry)  # Tuning parameters
```

Tuning parameters

Parameter	Description
`ntree`	Number of trees in forest
`mtry`	Number of variables randomly selected at splits

Exploring ML objects

Just like objects from statistical functions, objects from machine learning functions are **lists** that you can explore using **generic functions**:

Function	Description
`summary()`	Overview of the most important information
`names()`	See all named elements you can access with $
`plot()`	Visualise the object (sometimes)
`predict()`	Predict new data based on the ML model

```
# Create a regression object
baselers_glm <- glm(income ~ age + height + children,
                    data = baselers)

# Look at summary results
summary(baselers_glm)
# [...]
```

```
# Look at all named outputs
names(baselers_glm)
```

```
##  [1] "coefficients"      "residuals"         "fitted.values"     "e
##  [6] "rank"              "qr"                "family"            "l
## [11] "aic"               "null.deviance"     "iter"              "w
## [16] "df.residual"       "df.null"           "y"                 "c
## [21] "model"             "na.action"         "call"              "f
## [26] "data"              "offset"            "control"           "m
## [31] "xlevels"
```

```
# Access specific outputs
baselers_glm$coefficients
```

```
## (Intercept)         age      height    children
##     574.740     149.302       1.720       7.727
```

Predict new data with predict()

All machine learning objects will allow you to **predict the criterion of new data** using `predict()`.

Compare the predicted values to the true criterion values of `newdata` to see how well your model did.

argument	description
object	A machine learning / statistical object created from `glm()`, `randomforest()`, ...
newdata	A dataframe of new data

Predict values from `zurichers` data frame:

id	age	children	height	income
1	65	0	1.66	7500
2	75	3	1.96	5400
3	35	1	1.76	8400
4	54	0	1.73	9500
5	65	2	1.59	3700

```
# produce vector of new predictions
predict(object = baselers_glm,  # ML object
        newdata = zurichers)    # DF of new data
```

```
##     1     2     3     4     5
## 10282 11799  5811  8640 10298
```

Practical

Link to practical