Machine Learning

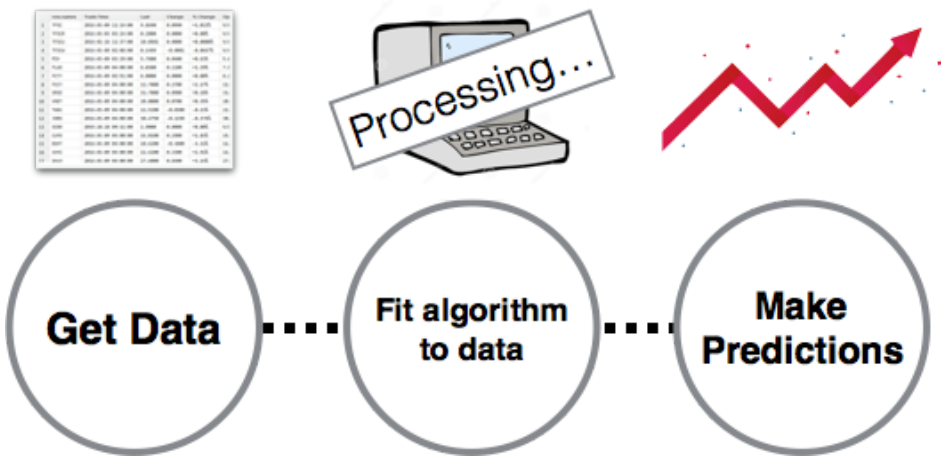
The R Bootcamp
Twitter: @therbootcamp

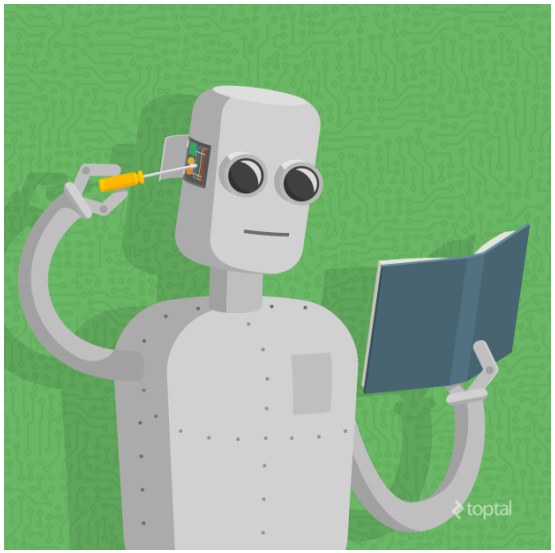
September 2017

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.



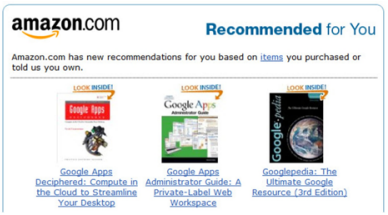


Everyone uses machine learning

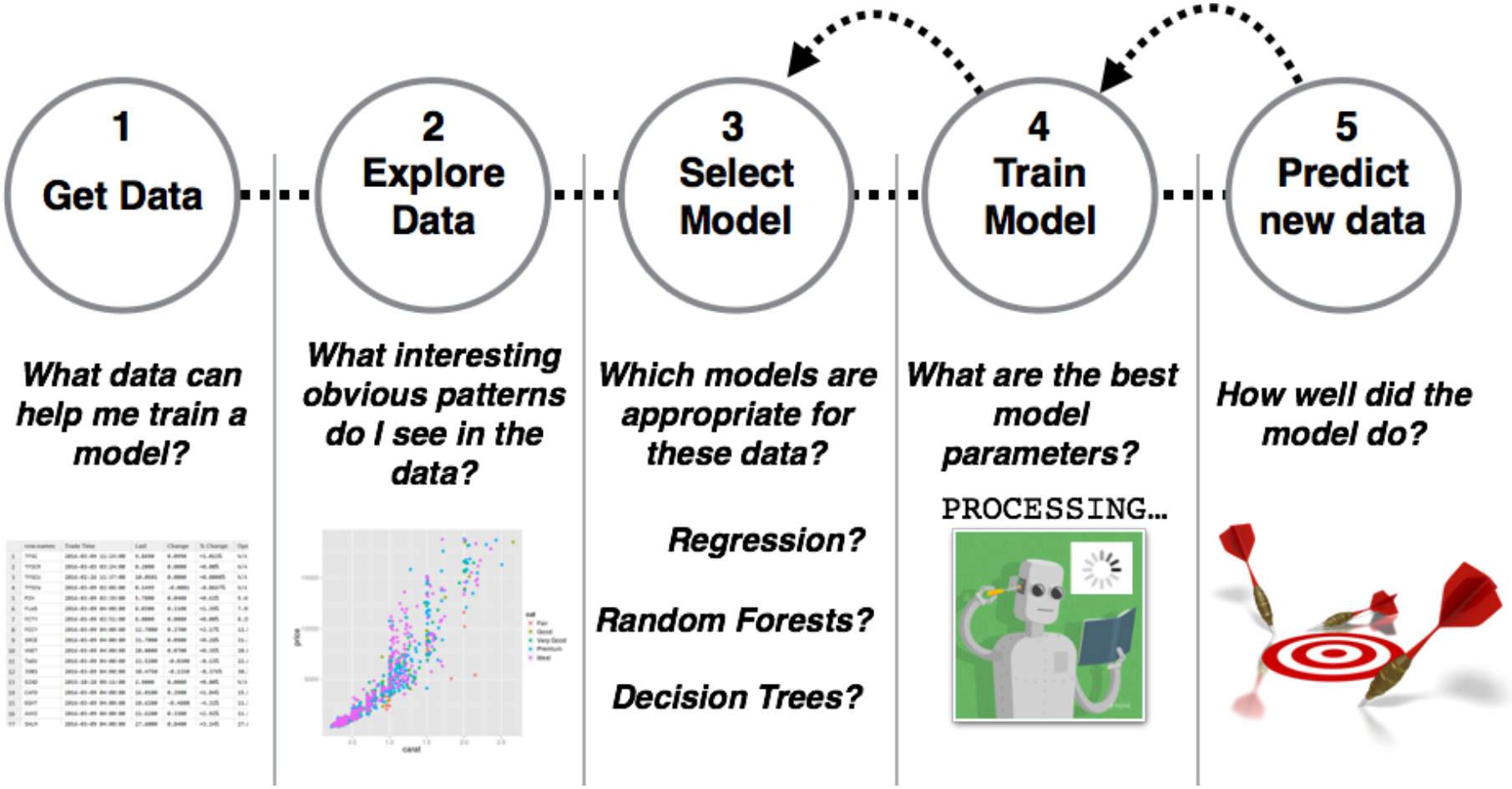
Everyone!

- How does Google know what search results you want?
- How does Amazon know what products to recommend?
- How does Netflix decide what shows you'll want to watch next?
- How do Tesla cars recognize objects and predict accidents?

> 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, Amazon founder



What is the basic machine learning process?

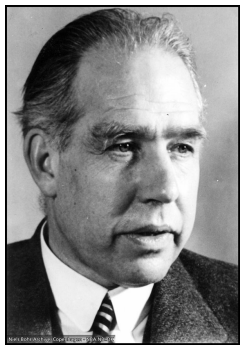


Why do we separate training from prediction?

Just because an algorithm can fit past (training) data well, does *not* necessarily mean that it will *predict* new data well.

Anyone can come up with a model of past stock performance. Predicting future performance is much more difficult.

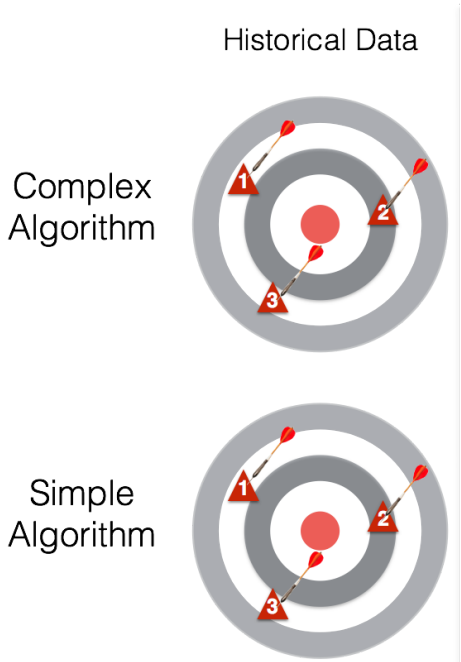
> "Prediction is difficult, especially when it is about the future" ~ Niels Bohr



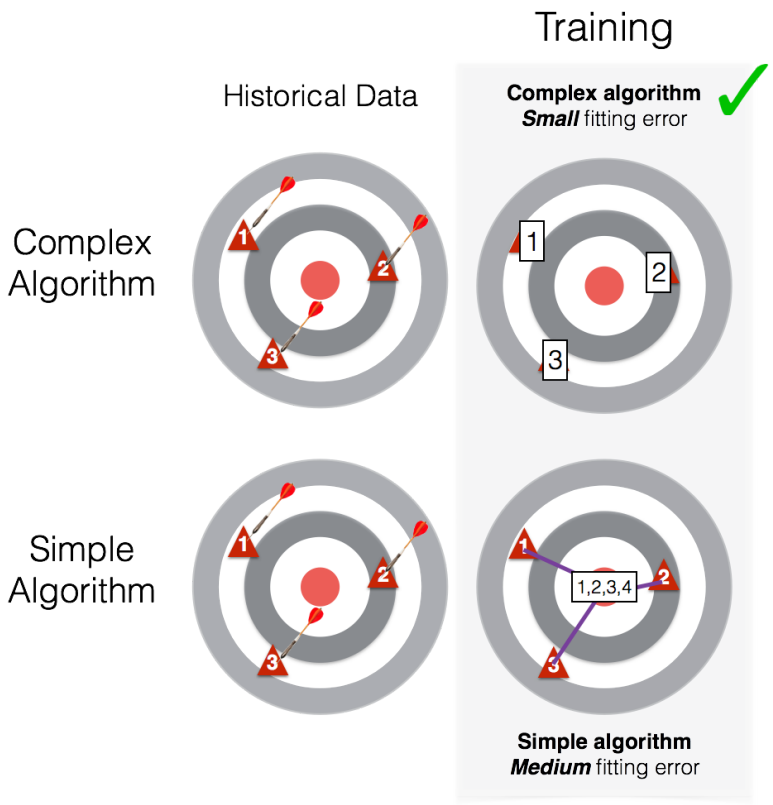
Niels Bohr, Nobel Laureate in Physics

> "An economist is an expert who will know tomorrow why the things he predicted yesterday didn't happen today." ~ Evan Esar

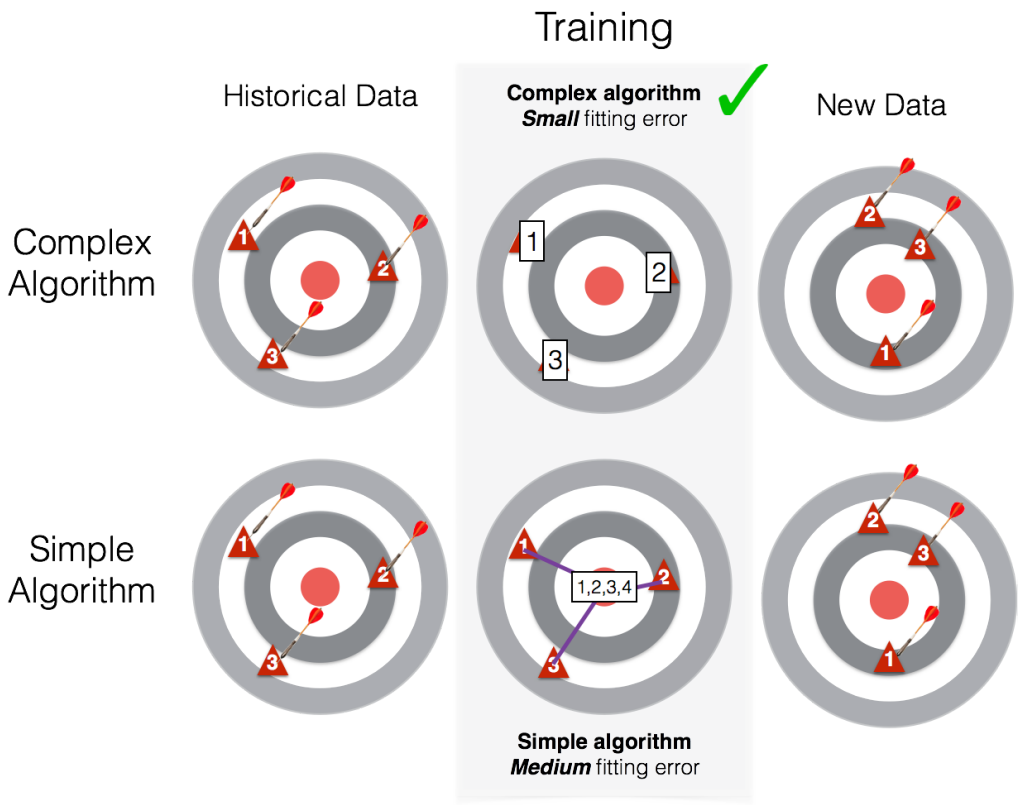
Training (fitting) vs. Testing (prediction)



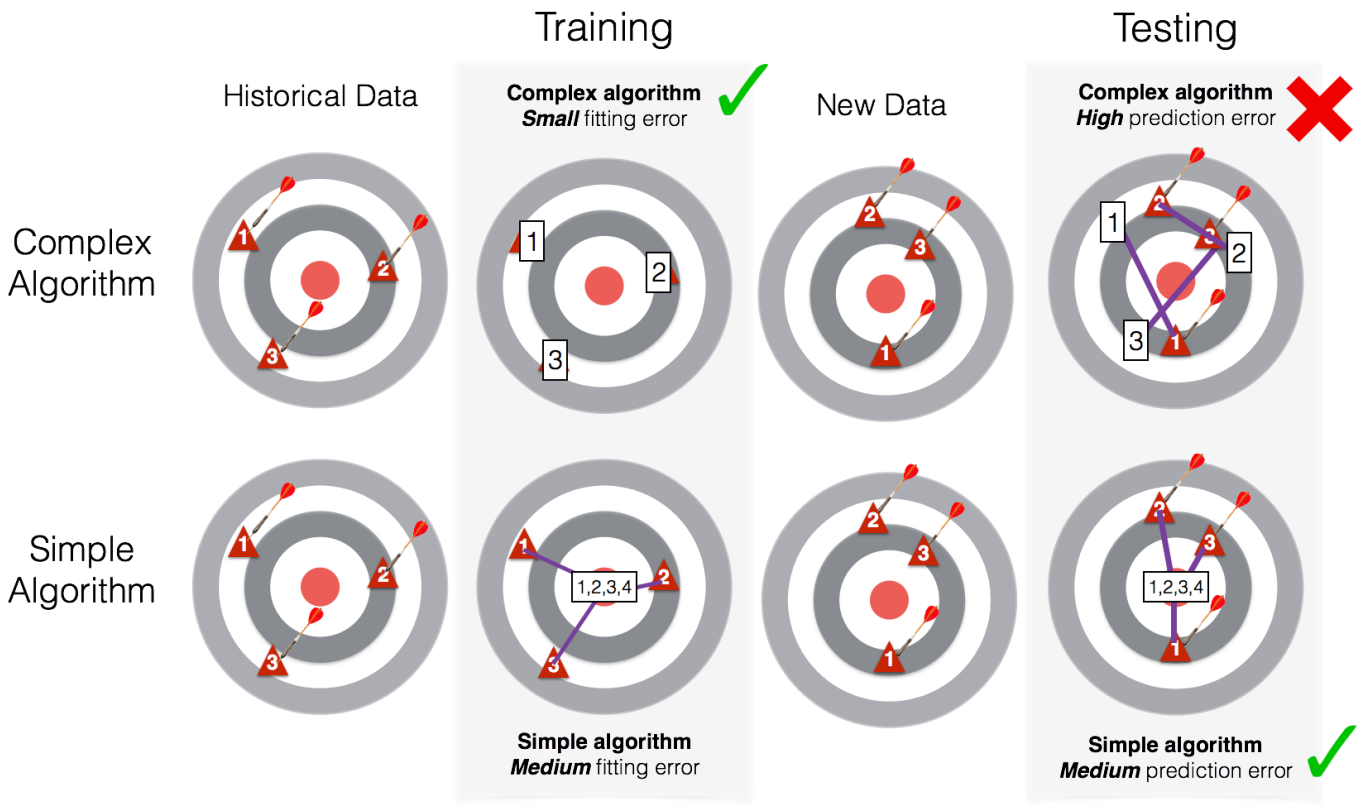
Training (fitting) vs. Testing (prediction)



Training (fitting) vs. Testing (prediction)

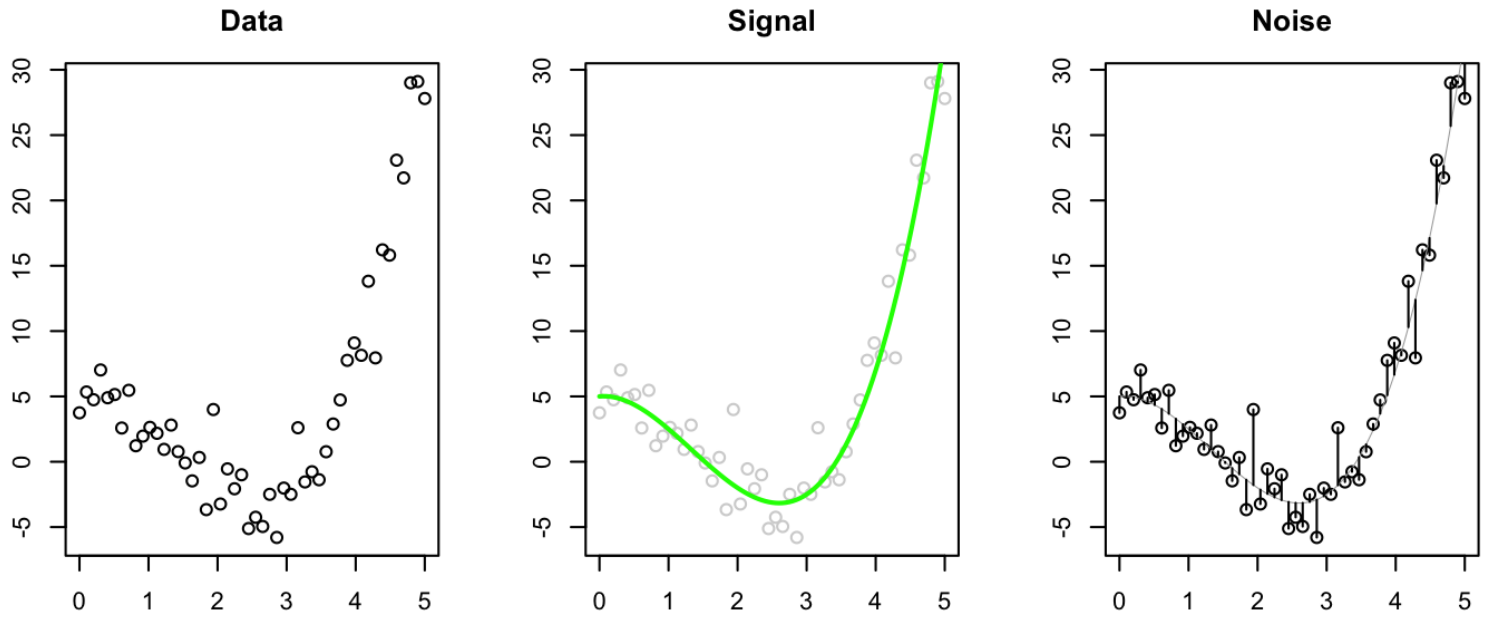


Training (fitting) vs. Testing (prediction)



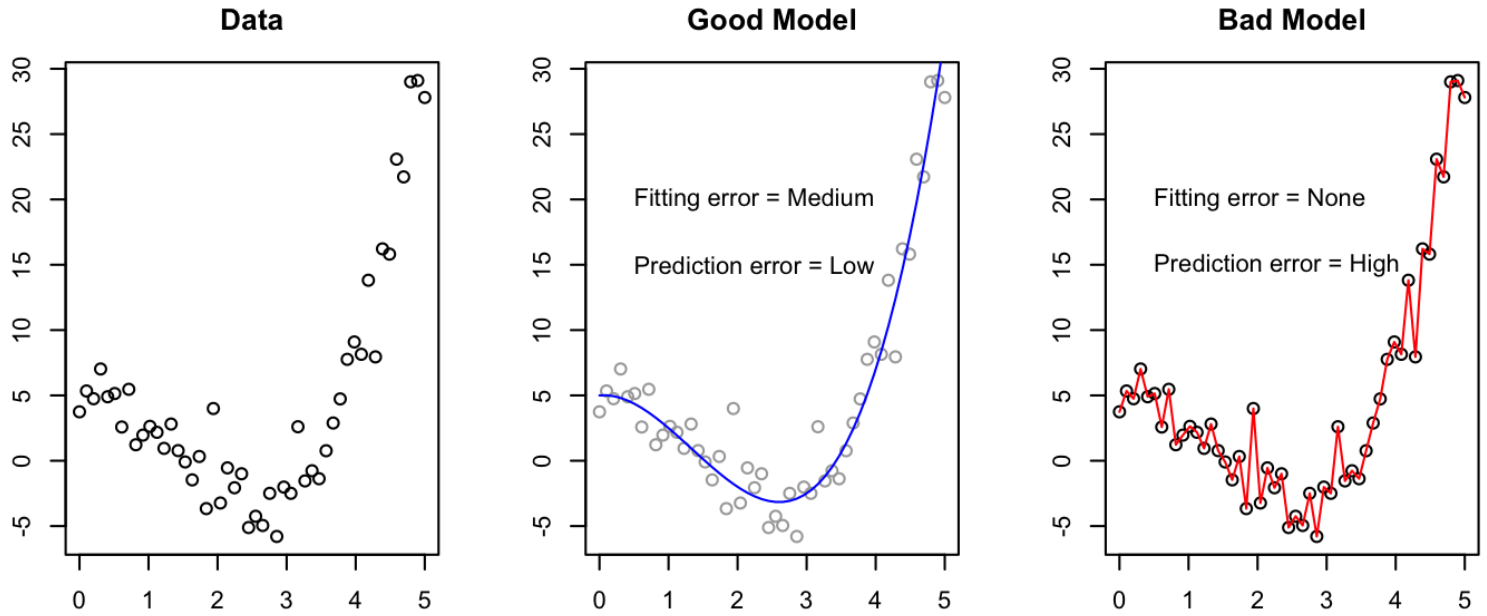
Why do we separate training from prediction?

- Data comes from two processes: *Signal* and *Noise* (aka Error).



Why do we separate training from prediction?

- A good model is one that tries to capture the signal and ignore the noise
- A bad model is one that captures too much unpredictable noise,

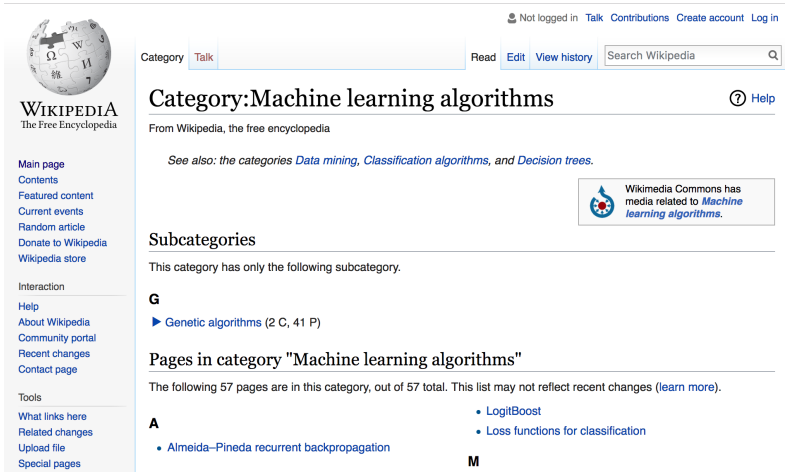


What machine learning algorithms are there?

- There are hundreds if not thousands of machine learning algorithms from many different fields.
  - E.g.; Computer vision, Natural language processing, reinforcement learning, graphical models
- In this section, we will focus on 4:

Algorithm	Complexity?
Regression	Low / Medium
Decision Trees	Low
Random Forests	High
Support Vector Machines	High

Wikipedia lists 57 *Categories* of machine learning algorithms, each with dozens of examples



How do you fit and evaluate models in R?

Fitting a model

```
A_model <- A_fun(formula = y ~.,
                 data = data_train,
                 ...)
```

Argument	Description	Note
formula	Formula indicating variables to use	`y ~ .` is often used as a catch-all
data	The dataset for model training	
...	Optional other arguments	See the function help page for details

Evaluating a model

```
# Common ways to explore / use a model

A_model            # Print generic information

names(A_model)     # Show attributes

summary(A_model)   # Print summary information

predict(A_model,   # Predict test data
        newdata = data_test)

plot(A_model)      # Visualize the model
```

Regression with `glm()`

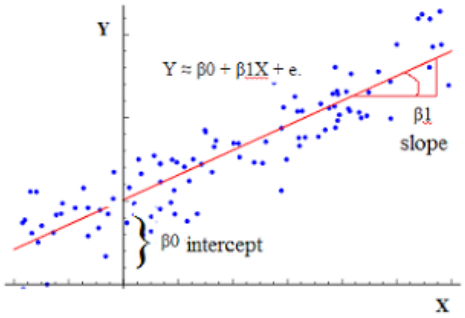
In regression, the criterion is modeled as the weighted sum of predictors times *weights* β_1, β_2

Example: Default on a loan

One could model the risk of defaulting on a loan as:

$$Risk = Age \times \beta_{age} + Income \times \beta_{income} + \ldots$$

Training a model means finding values of β_{Age} and β_{Income} that 'best' match the training data.



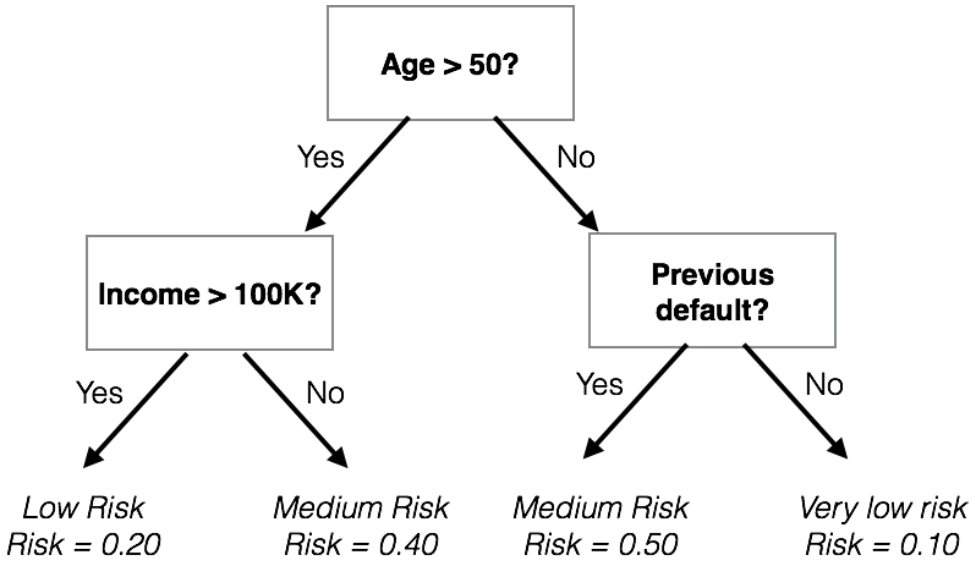
Create regressions using the `glm()` function (part of base-R)

```
# glm() function for regression
glm(formula = y ~.,      # Formula
    data = data_train,   # Training data
    family, ...)         # Optional arguments

# Train glm model
loan_glm_model <- glm(formula = risk ~ .,
                      data = data_train)

# Predict new data with glm model
loan_glm_pred <- predict(loan_glm_model,
                         newdata = data_test)
```

Decision Trees with `rpart::rpart()`

In decision trees, the criterion is modeled as a sequence of logical Yes or No questions.

Example: Default on a loan



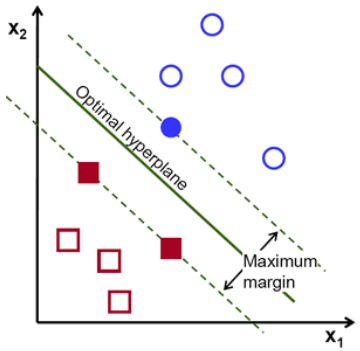
Create decision trees using the `rpart` package

```
# Load the rpart package
library(rpart)

# Calculating a decision tree in R
rpart(formula = y ~., # Formula
      data = data_train, # Training data
      method, parms, cost) # Optional arguments

# Train rpart model
loan_rpart_model <- rpart(formula = risk ~ .,
                          data = loan_data,
                          method = "anova")

# Predict new data with rpart model
loan_rpart_pred <- predict(loan_rpart_model,
                          newdata = data_test)
```

Advanced algorithms

Support Vector Machines

`e1071::svm()`




```
# Creating support vector machine model
library(e1071)

svm_model <- svm(formula = risk ~ .,
                 data = loan_data,
                 ...)
```

Random Forests

`randomForest::randomForest()`

```
# Creating random forest model
library(randomForest)

rf_model <- randomForest(formula = risk ~ .,
                         data = loan_data,
                         ...)
```

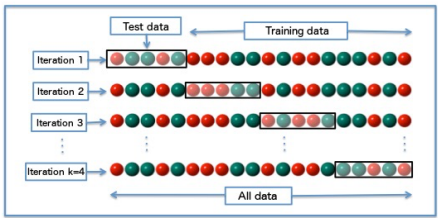
How do I do machine learning in R?

If you're really into machine learning, packages such as `mlr` and `caret` can automate much of the the machine learning process.

`install.packages('mlr')` `install.packages('caret')`

Cross Validation

Multiple training / test data splits



In the practical, we will go through the basic steps "by hand" so you can see the process:

```
# Create training and test data
data_train <- ...
data_test <- ...

# Train models on training data
model_A <- A_fun(formula = y ~ .,
                 data = data_train)

# Model A predictions
pred_A <- predict(model_A,
                  newdata = data_test)

# Calculate Model A error
pred_err_A <- mean(abs(pred_A - data_test$y))

# Compare to Models B, C, D...
```

Questions?

Machine Learning Pratical

Link to Machine Learning practical