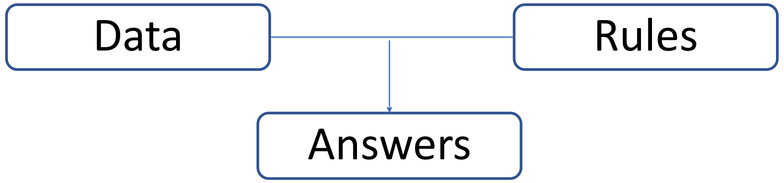
Machine learning glossary

Building blocks

- Machine learning (ML) in essence – approximating complex dependencies

Classical programming

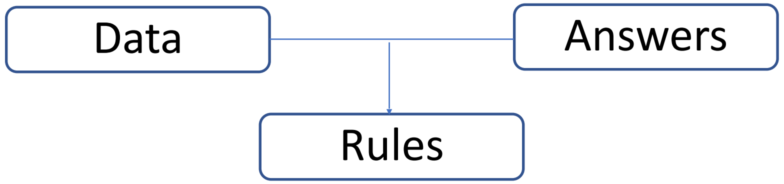


Data

Rules

Answers

ML



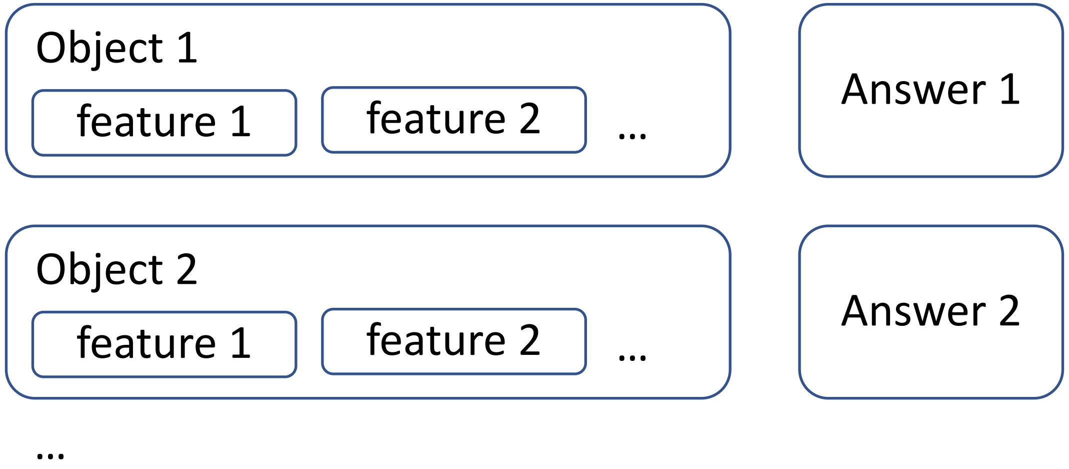
Data

Answers

Rules

- Object – unit of input that gives us a distinct answer
    - Could be simple such as a client deal
    - Could be more complex such as recent 5 days of client activity
- Feature – numeric characteristics of an object
    - For example, how much net dv01 client X bought 3 days before
- Target = answer, this is what we would like to predict
    - Say, $RUB or OFZ yield movement

Zooming in on how data is represented for ML:



Object 1

feature 1 feature 2 …

Answer 1

Object 2

feature 1 feature 2 …

Answer 2

…

Model

Model – representation of rules

- ○ Eg linear equation, or a chain of logical gates

Algorithm – procedure for efficient discovery of rules

- ○ Closely related to model – often used as synonyms

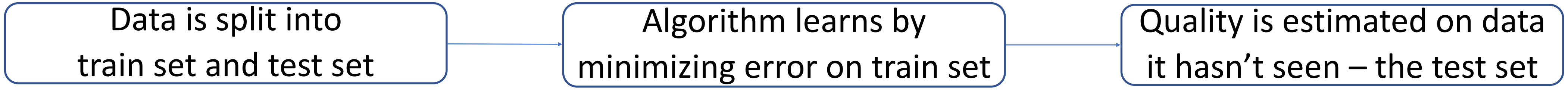
Basic model assumption – nature of dependency:

- Non-linear
  - ○ Most dependencies we deal with, especially flow-related ones
  - ○ For example, if client X buys in combination with client Y, it might be more powerful than simply the sum of two
- Linear
  - ○ Still, in cases where dependency is clearly linear (say, we model where RUB should be trading based on where MXN, ZAR etc are trading), a linear model will have advantage

Model output types:

- Regression – gives exact number
  - ○ Eg size of move over 10 days horizon
- Classification – assigns a label
  - ○ Say, whether first 1% move going forward will be up or down (binary classification)

Learning and evaluation



Data is split into train set and test set

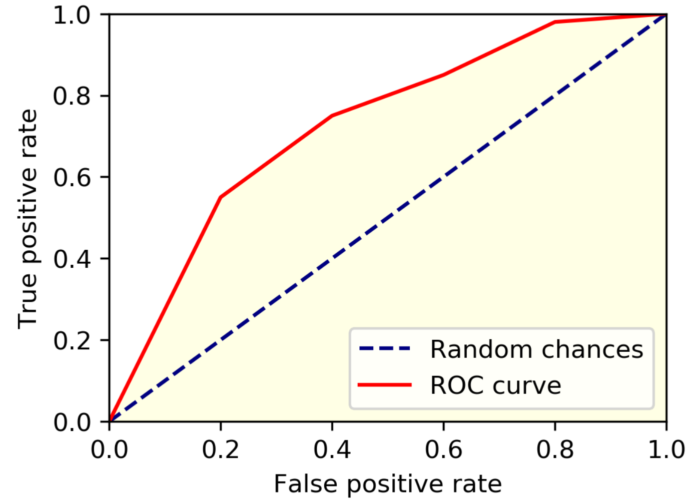
Algorithm learns by minimizing error on train set

Quality is estimated on data it hasn't seen – the test set

Cross-validation technique – several different train/test splits for the same data, so that several estimates of quality are available

Metrics to assess quality

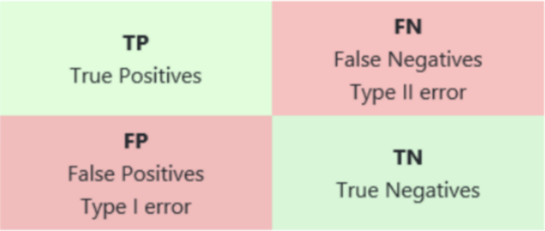
- Regression:
  - ○ R2 – to which extent variance in the target variable is explained by the model
- Classification:
  - ○ Accuracy – percentage of all objects we classified correctly (could be misleading)
  - ○ Precision – out of objects we classified as 'up', how many were 'up' in reality
  - ○ Recall – out of all objects that were in reality 'up', how many we correctly discovered as 'up'
  - ○ ROC AUC – area under curve that shows tradeoff b/w TP and FP rates



All of those can be calculated out of so-called confusion matrix, which plots labels that model put on objects against what they were in reality

⇧ Predicted ⇩

⇧ Actual ⇩



TP True Positives	**FN** False Negatives Type II error
FP False Positives Type I error	**TN** True Negatives

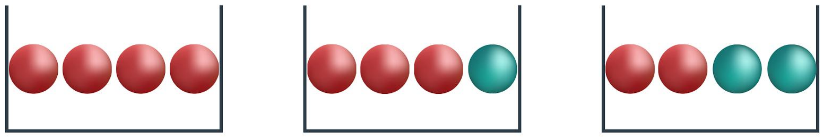
Decision tree ensemble (aka Random Forest)

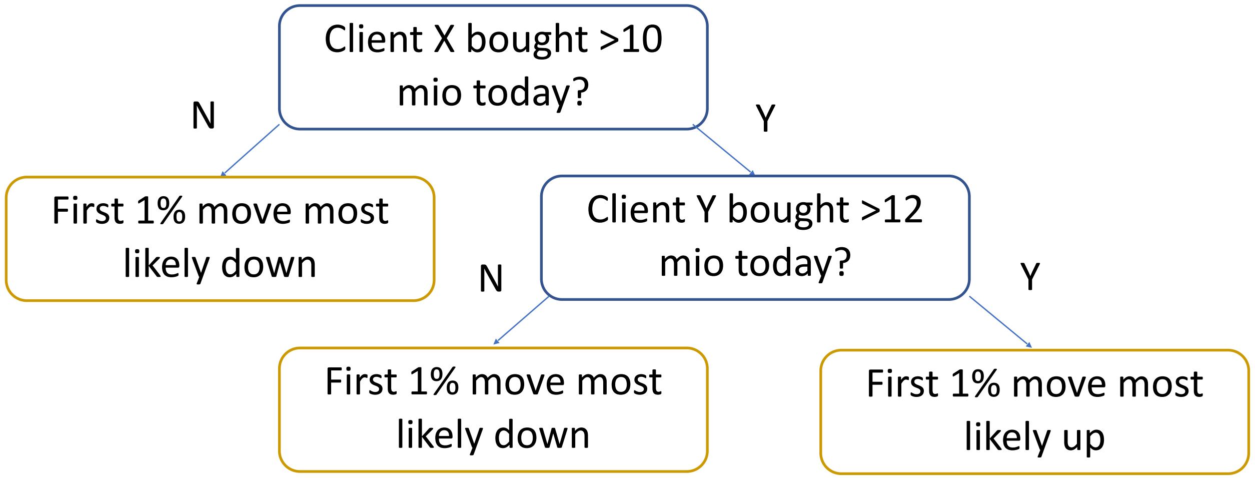
Basic unit of the algorithm – decision tree

- nodes (blue) – logic gates
- leaves (orange) – predictions

Decision tree goes over all features and chooses split that gives lowest entropy

- Entropy – measure of "chaos", "randomness"




Client X bought >10 mio today?

N

Y

First 1% move most likely down

Client Y bought >12 mio today?

N

Y

First 1% move most likely down

First 1% move most likely up

Single tree vulnerability: unstable, change in input dataset leads to a different tree

Solution – ensembling ("forest" of trees):

- Eliminates instability by averaging predictions from many individual trees
- To further improve stability, each tree sees only a random subset of overall training data (method called bootstrap aggregation), so short name for the overall algorithm is Random Forest