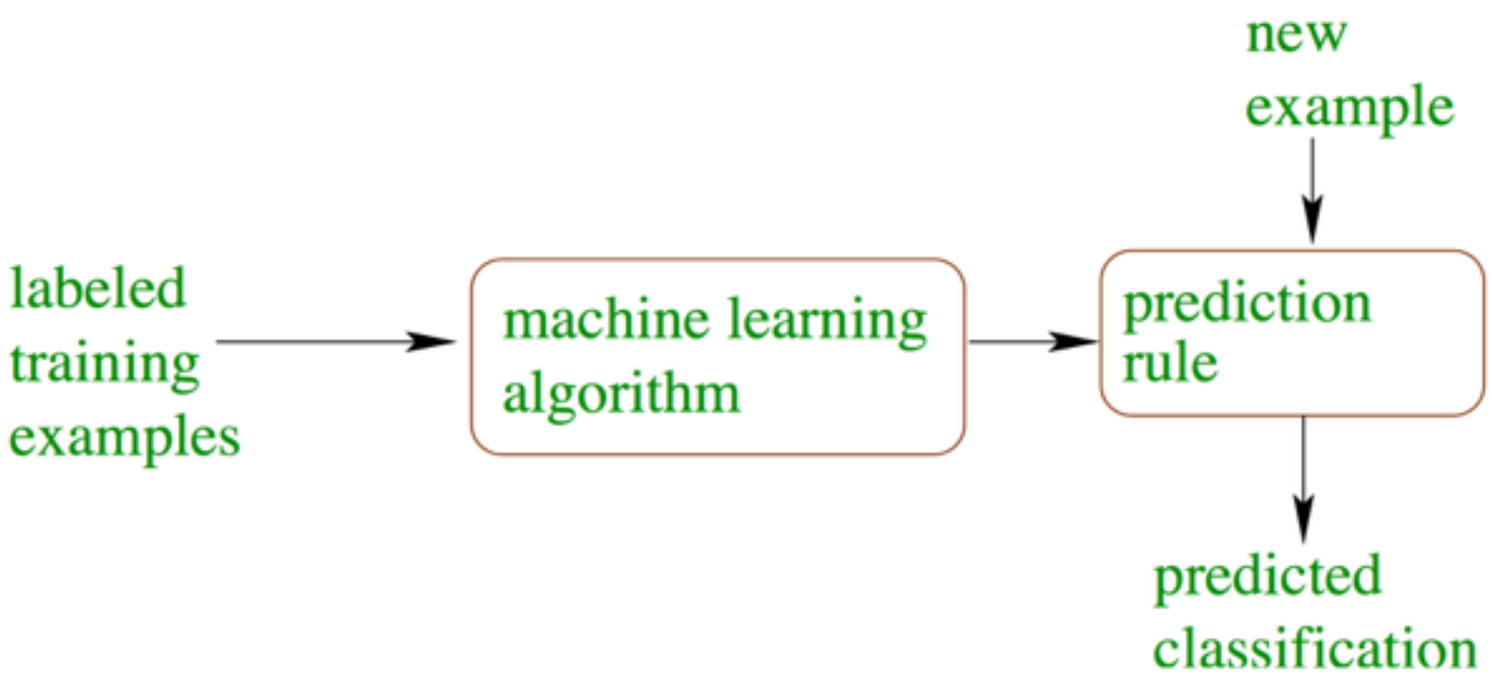


Figure 1: Diagram of a typical learning problem.

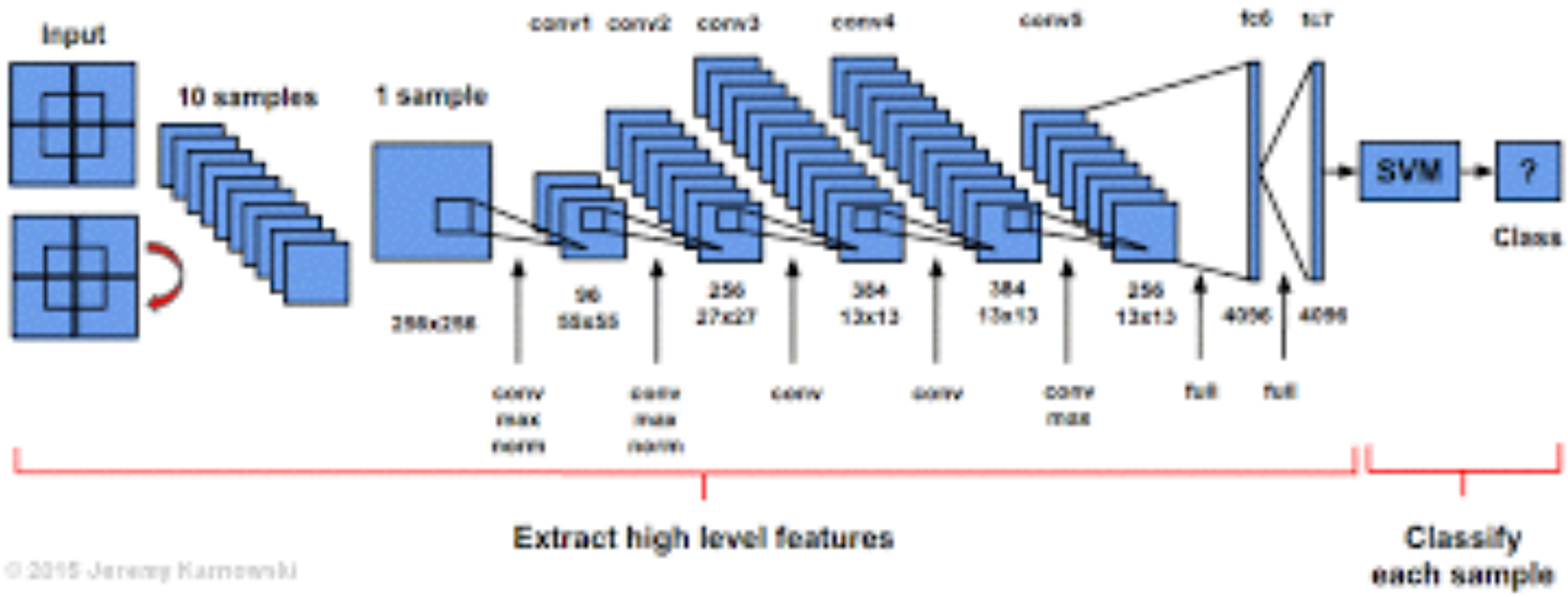
The machine learning “pipeline”

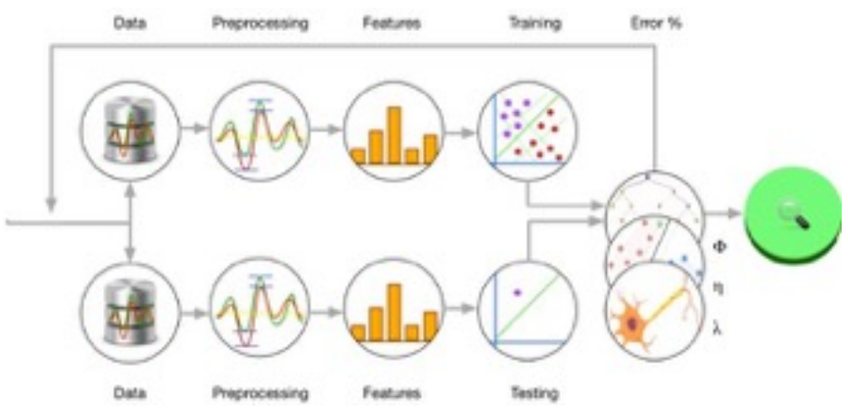
Two-day workshop

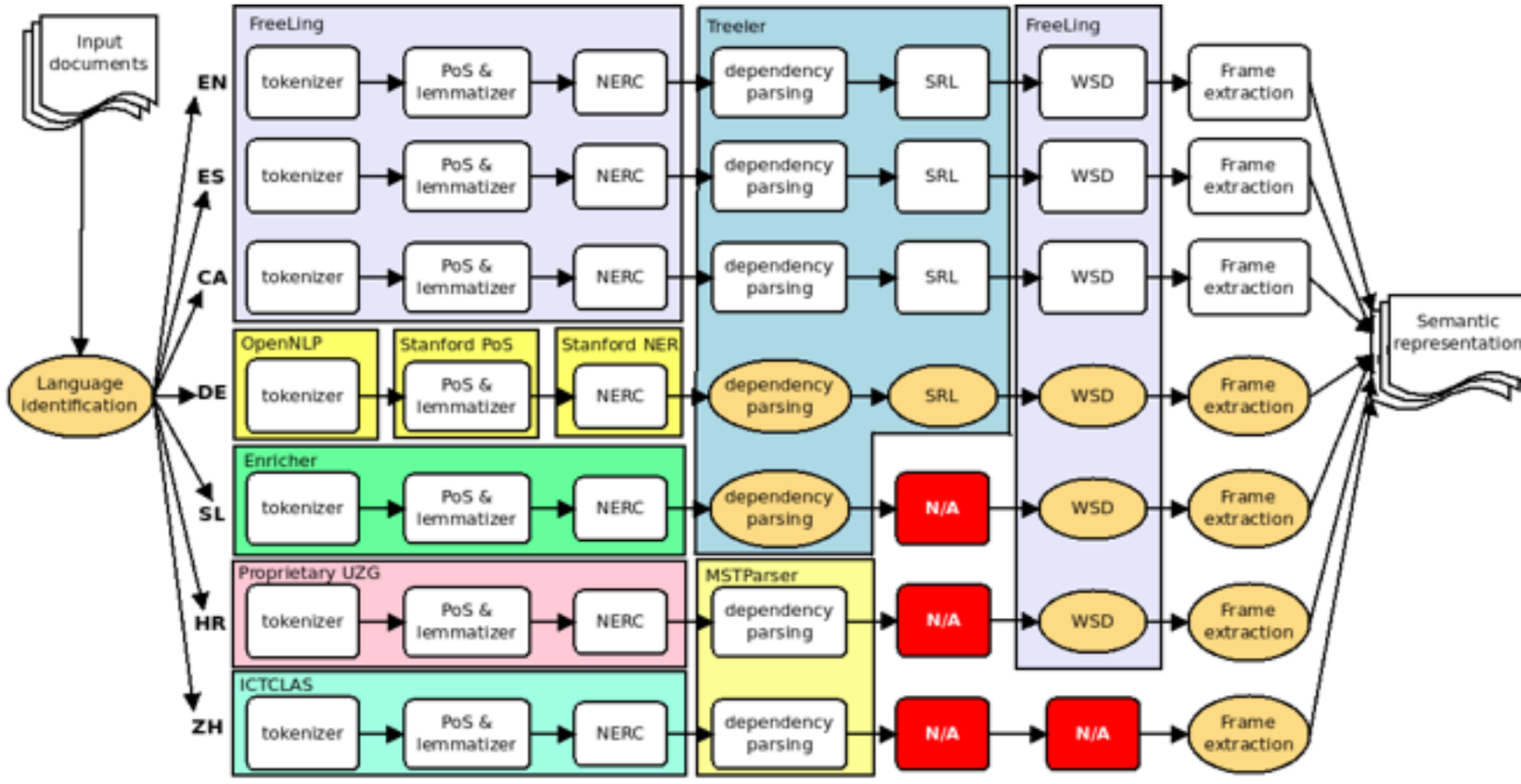
Duke University

Adam Chekroud

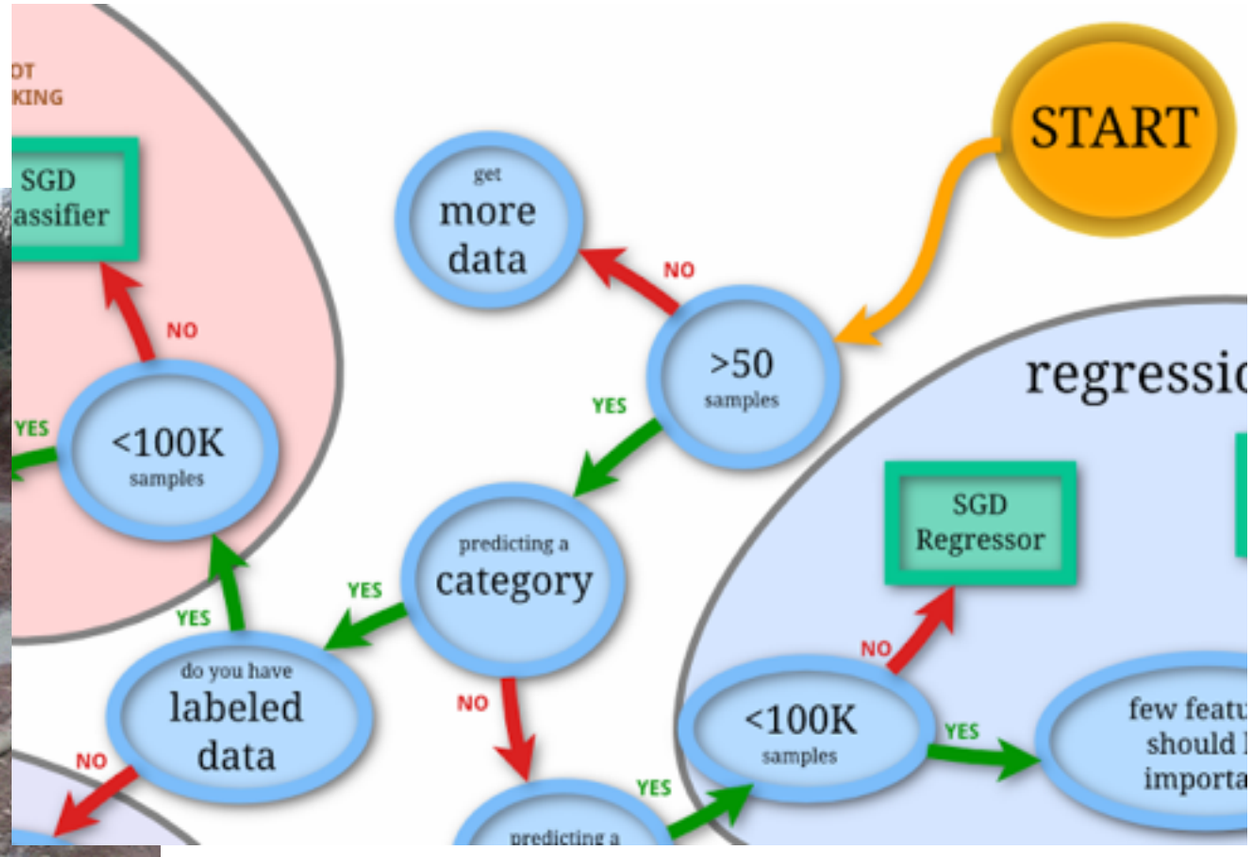
Objectives


Input
10 samples
1 sample
conv1
conv2
conv3
conv4
conv5
fc6
fc7
SVM
?
Class
256x256
96
55x55
256
27x27
384
13x13
384
13x13
256
13x13
4096
4096
conv
max
norm
conv
max
norm
conv
conv
conv
max
full
full
Extract high level features
Classify
each sample
© 2015 Jeremy Karnowski


Data
Preprocessing
Features
Training
Error %
Data
Preprocessing
Features
Testing


Input documents
Language identification
EN
ES
CA
DE
SL
HR
ZH
FreeLing
tokenizer
PoS & lemmatizer
NERC
OpenNLP
Stanford PoS
Stanford NER
Enricher
Proprietary UZG
ICTCLAS
Treeler
dependency parsing
SRL
MSTParser
N/A
FreeLing
WSD
Frame extraction
Semantic representation

Groundwork….



- Biggest barrier to machine learning is the **quantity** and **quality** of data

Groundwork….

Getting data

- ML is not an alternative to collecting a lot of data
    - algorithms developed for problems with hundreds of thousands of data points

- ML is not a silver bullet for noisy data
    - imprecise data (esp. clinical report)
    - limited signal (genetics)
    - consistency in data (MRI scans vary within a person within a day, let alone across scanners/sites)

Groundwork….

Data processing

- Almost every data set is, in practice, flawed
    - Manually entered data is extremely suspect
    - Even automatically generated/scripted data can have (systematic) error
- Always good to process data in pairs
- **Everything** should be scripted and documented

Getting started

Data inspection/exploration

- Organise data
    - **Default***: rows = observations, columns = predictors
- Have a look around!
    - You don’t have to rely on the algorithm to figure everything out
        - Create new variables based on domain-expertise!
    - Missing data?
        - Most ML methods cannot use NAs
    - Unsupervised methods can be helpful here

*Wickham (2014), *J. Statistical Software*

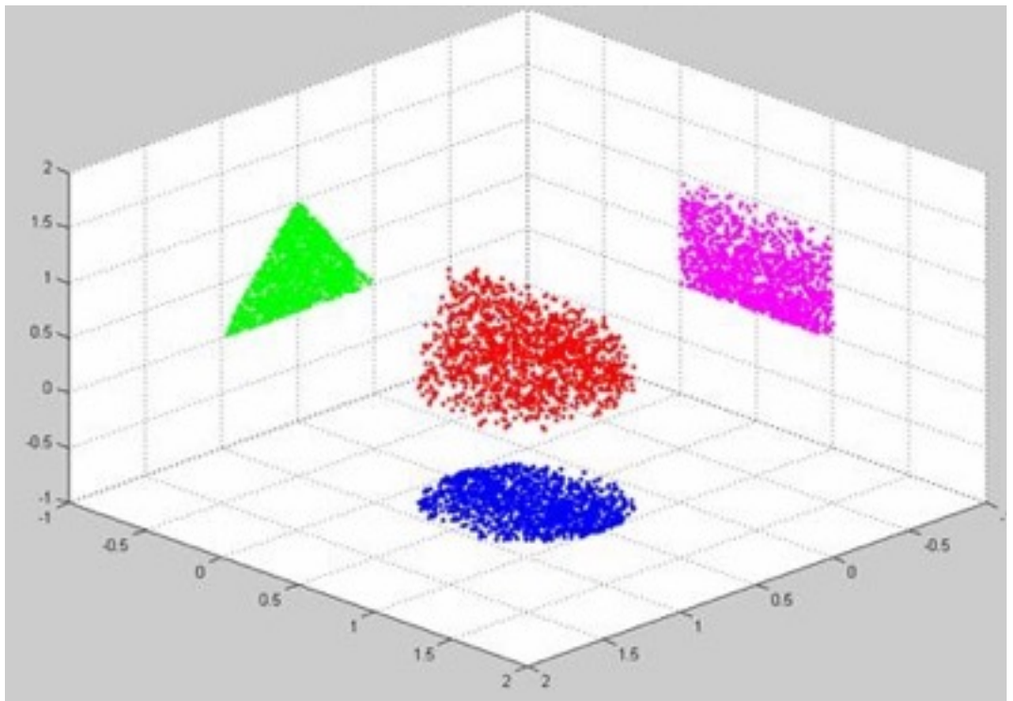
Getting started

Identify problem

- "Target" or dependent variable
    - Continuous outcome? *Regression* problem
        - Binary outcome? *Classification* problem
            - Can be extended to multi-class classification
- It is okay to have a few outcome measures that you are interested in modeling
    - (Bear in mind consequences: multiple comparisons, FPR..)

*Wickham (2014), *J. Statistical Software*

Feature Selection



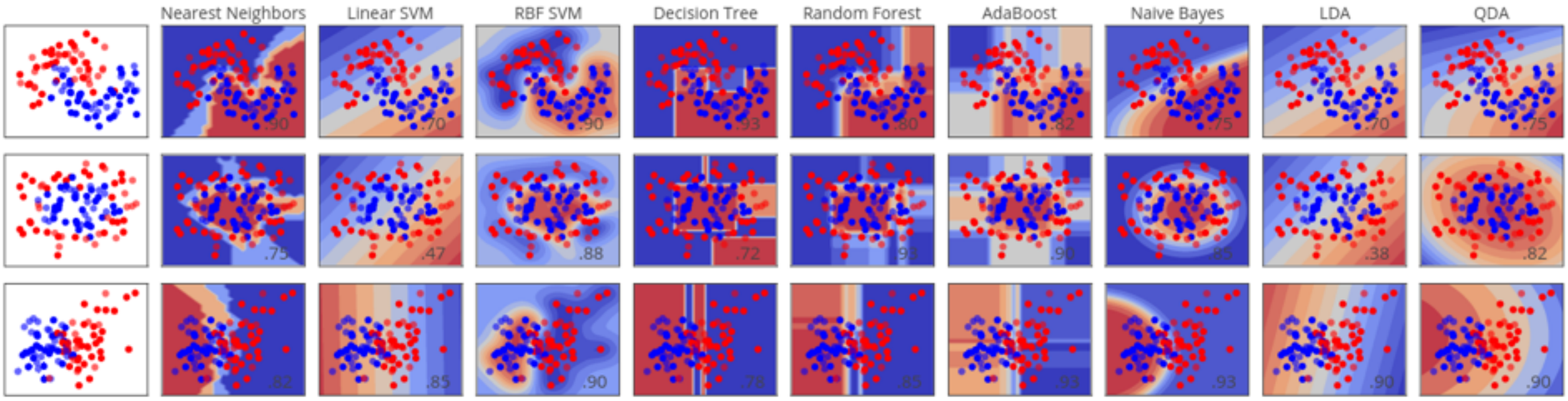
Tailoring data to our problem

- Models don't like too much information
- We can *create* features!

- What information is predictive?
- How much do we need?
- Do we need to reduce or transform the data?
- Can we use our own knowledge to make better (custom) predictors

- Cut through noise in data, but try and keep signal

Algorithm Selection



- Use predictors to “train” predictive algorithm
- Algorithm “learns” optimal mapping between your data and the outcome
    - Form and scope of mapping varies across algorithms
- Use this mapping to guess what happens for unseen data

- Many extremely powerful algorithms available “off the shelf”

Testing, testing, testing!

Model Performance?

- How do we quantify performance?
- Choose a key performance metric
    - Accuracy, PPV; R^2, Mean absolute error?

Model Validation?

- Is this model a good model of reality?
- How will our model do on future data?
    - test-train split; leave-one-out; split half, k-fold

Classification 101

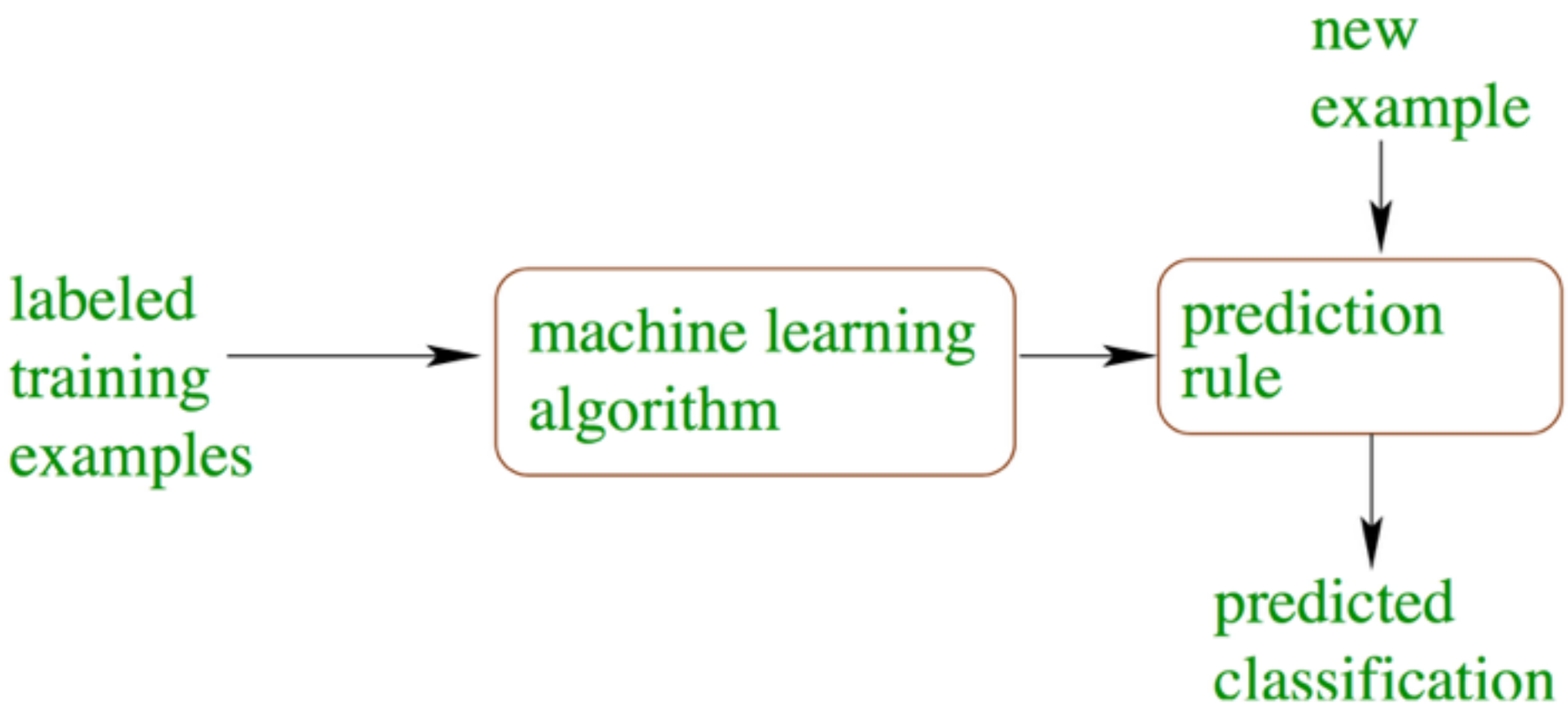


Figure 1: Diagram of a typical learning problem.