Intro to Machine Learning

and understanding model error

What is Machine Learning?

The ability to make accurate predictions based on what was experienced in the past ”Programming by example”

Basic tasks:

- Regression: predict the value of a function (e.g., stock price prediction)
- Classification: categorize objects into fixed categories (e.g., detecting fraudulent transactions)

Genetic Interactions - Type

Our case can be translated into two options:

- Classification: Is this pair a genetic interaction? (true/false)
- Regression: What is the cell growth of this pair? (0..1)

Example - Spam detection

Input Attributes

Target Attribute

Number of new Recipients	Email Length (K)	Country (IP)	Customer Type	Email Type
0	2	Germany	Gold	Ham
1	4	Germany	Silver	Ham
5	2	Nigeria	Bronze	Spam
2	4	Russia	Bronze	Spam
3	4	Germany	Bronze	Ham
0	1	USA	Silver	Ham
4	2	USA	Silver	Spam

Instances

Numeric

Nominal

Ordinal

Supervised Learning

Functions $\mathcal{F}$

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Training data

$$\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$$

LEARNING

$$\text{find } \hat{f} \in \mathcal{F} \quad \text{s.t. } y_i \approx \hat{f}(x_i)$$

Learning machine

New data

$$x$$

PREDICTION

$$y = \hat{f}(x)$$

Genetic Interactions - Data

X is the vector of features created from the Ontology for each gene pair, each GO term has possible values of 0,1,2

Y is the cell growth for each gene pair - a value between 0 and 1

Model Training

Choose the simplest model/algorithm that fits the problem/data

- Unbalanced data (95% from a single class)
- Feature amount
- Feature type - continuous or categorical
- Missing values
- Prior work with the model

Model example - Decision Trees

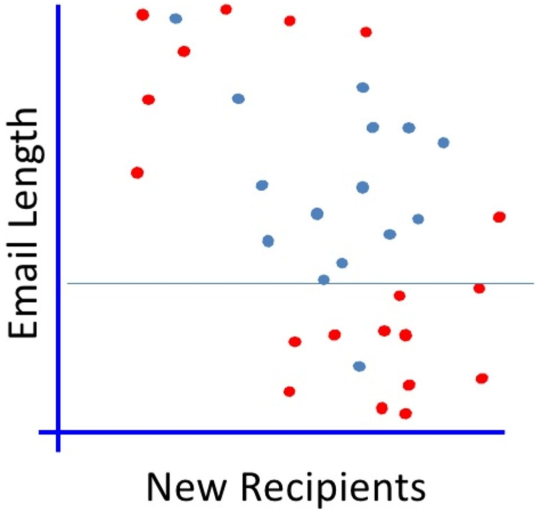
Flow-chart like tree structure

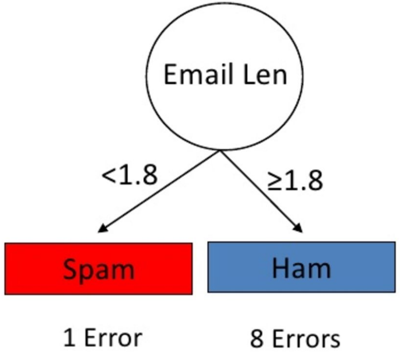
Node - a test on a feature

Branch - a result of the test

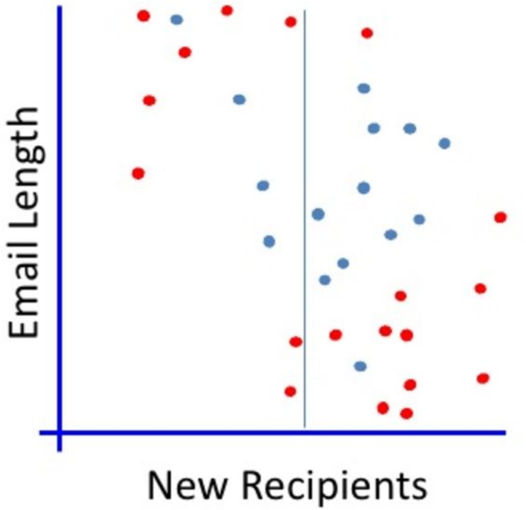
Leaf - label

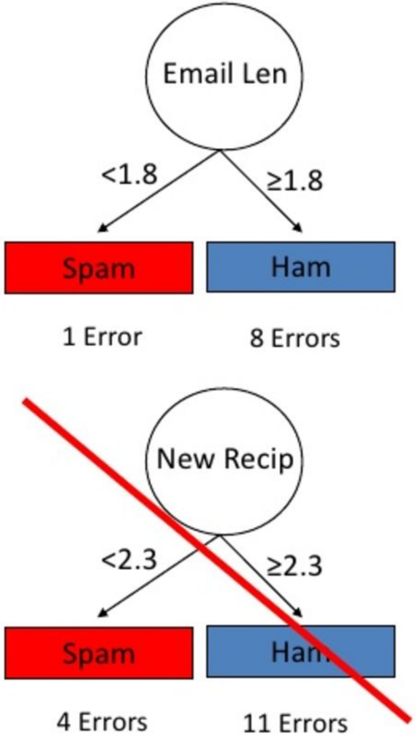
Model example - Decision Trees



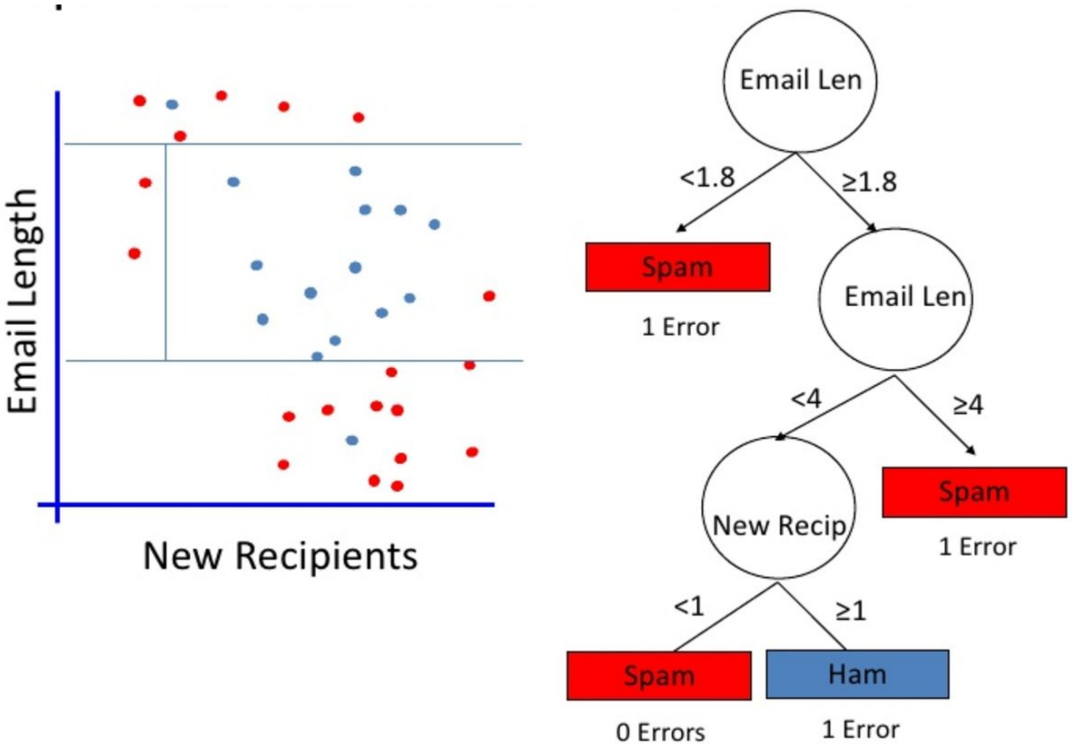


Model example - Decision Trees

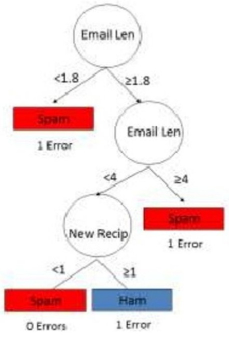


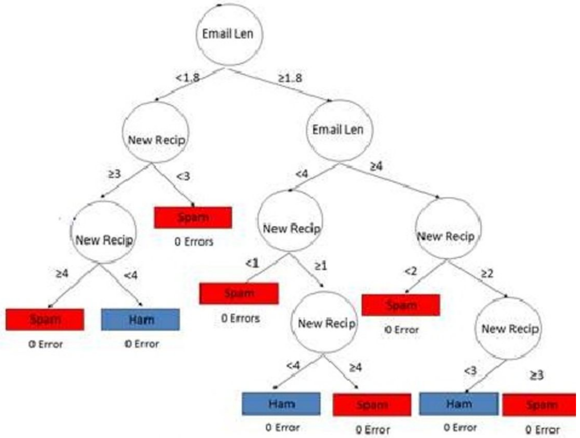


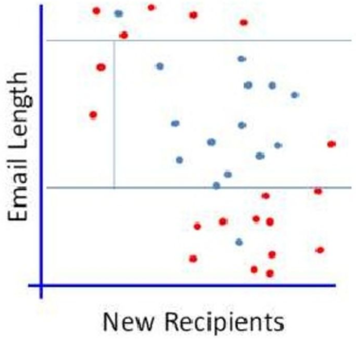
Model example - Decision Trees

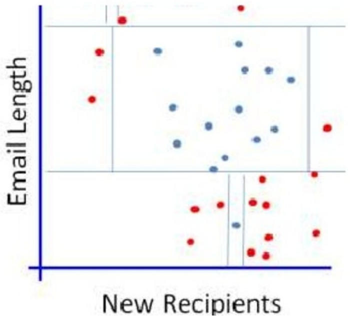


Model example - Which one?









Generalization

Do well on test data that is not known during learning

Minimizing the loss function (error) on the training data is not necessarily the best policy

We want the learning machine to model the true regularities in the data and to ignore the noise in the data

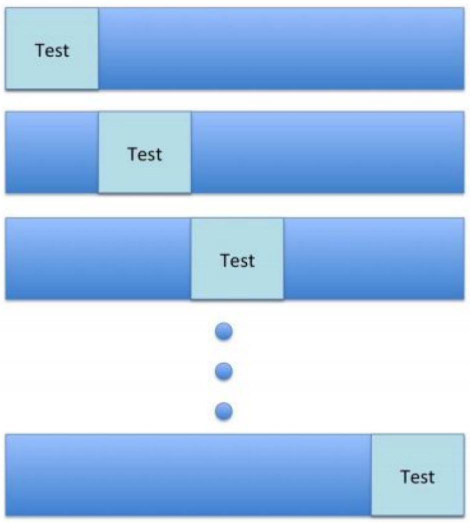
http://www.robots.ox.ac.uk/~az/lectures/ml/lect1.pdf

Train and test sets

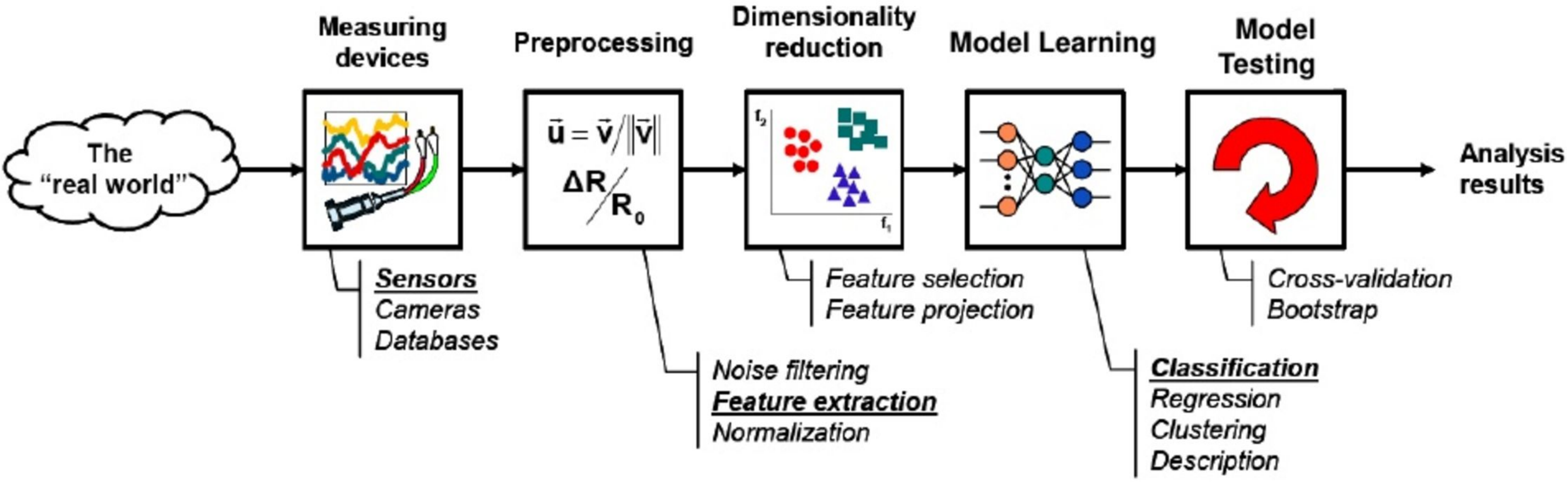
Simulate unseen data by splitting data into train / test (hold-out) sets

- Train our model on part of the data
- Test the performance on the rest
- (Optional validation set)

Decrease variance with cross-validation

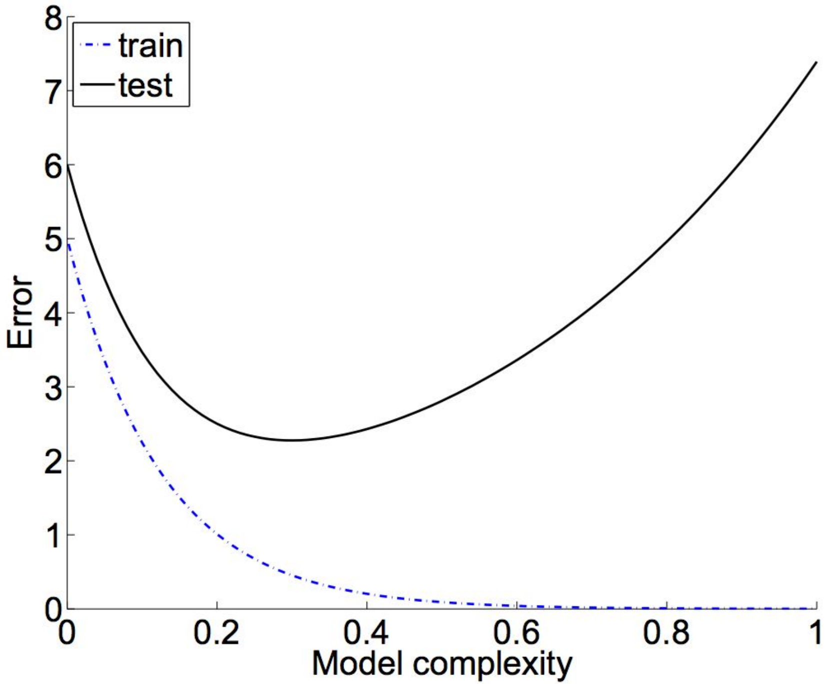


The Learning Process



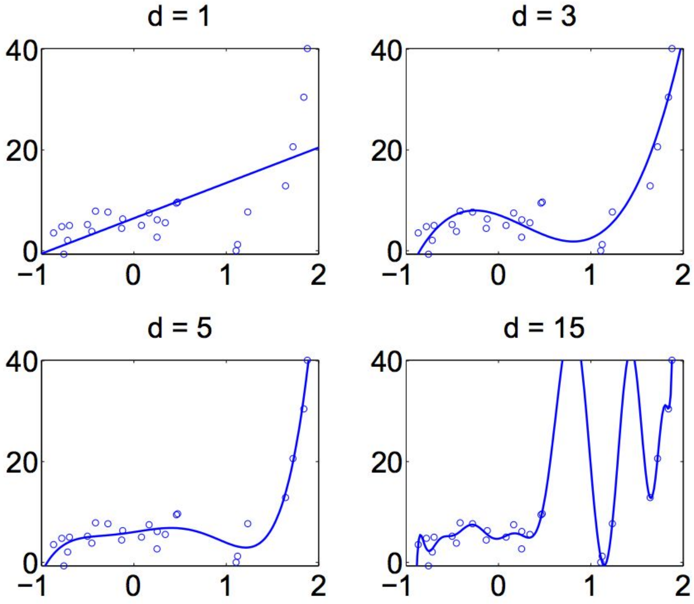
https://www.slideshare.net/liorrokach/introduction-to-machine-learning-13809045

Overfitting



Overfitting

Important to limit model complexity (example: polynomial degree)



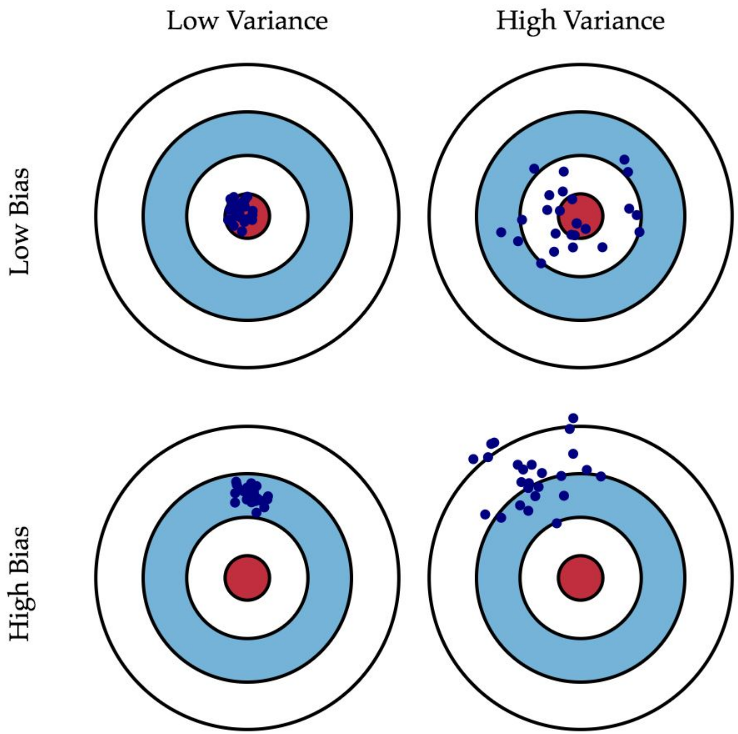
Bias-Variance tradeoff

Bias - stems from bad assumptions of our model

Variance - due to sensitivity to variations in the data

Low model complexity - high bias
low variance

High model complexity - low bias
high variance



http://scott.fortmann-roe.com/docs/BiasVariance.html

Error (Confusion) Matrix

True Positive - correct “Yes” classifications
True Negative - correct “No” classifications
False Positive - “No” classified as “Yes”
False Negative - “Yes” classified as “No”

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Many measures are derived using simple functions on these terms

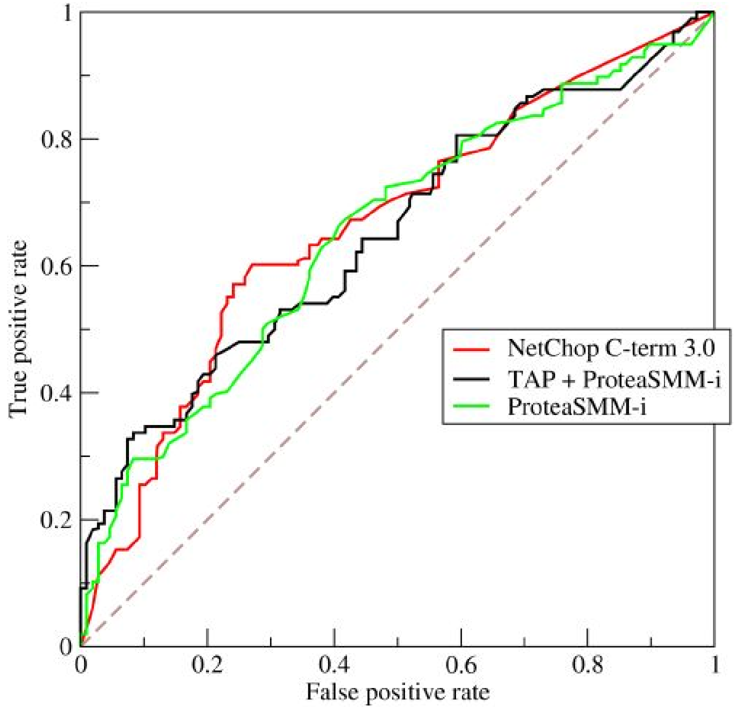
ROC & AUC

Recall or TPR (True Positive Rate) = TP/(TP+FN)

Fall-out or FPR = FP/(FP+TN)

AUC - Area Under Curve

Random model AUC = 0.5



https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Precision-Recall

Precision - TP/(TP+FP)