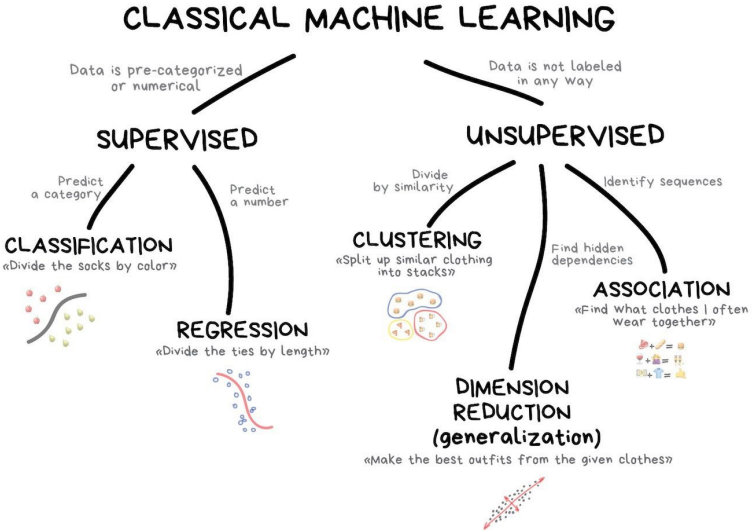
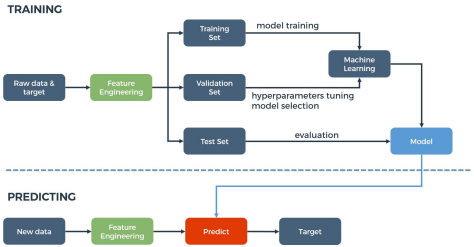

CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
Divide by similarity
Find hidden dependencies
Identify sequences
CLUSTERING
«Split up similar clothing into stacks»
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

vas3k

Data



Training set

- representative set of examples used for training, where the target value is known

Validation set

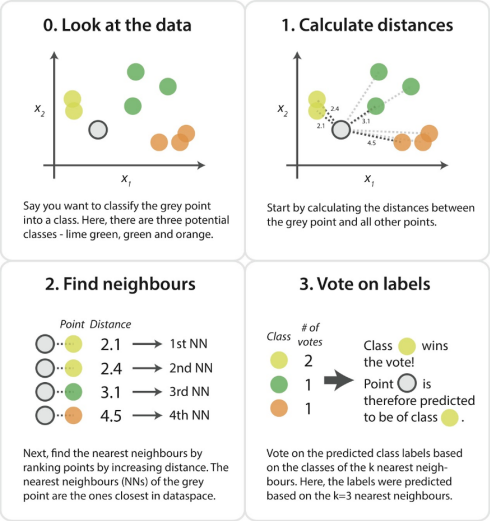
- representative set of examples used to tune the architecture of a learning algorithm and estimate prediction errors

Independent test set (blind test)

- independently assess the performance of a predictive model
- never used during the training process
- error on the blind test provides an unbiased estimate of the generalization error

Julien Jouganous *et al.* via medium.com

K-nearest neighbours (KNN)



- given training data,
  $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
  and a test point x_u
- prediction rule:
  look at the K most similar training examples to x_u
- for classification:
  assign the majority class label (majority voting)
- for regression:
  assign the average response
- The algorithm requires
  - parameter K: number of nearest neighbors to look for
  - distance function: to compute similarities between points

Jeorge Leonel via medium.com

Naïve Bayes

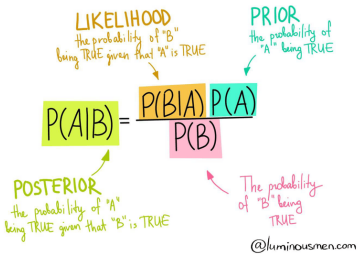
Conditional probability:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Probability of A and B

Probability of A given B

Probability of B



Data Science. Bayes theorem, luminousmen.com

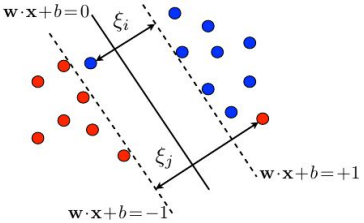
Maximum a posteriori (MAP):

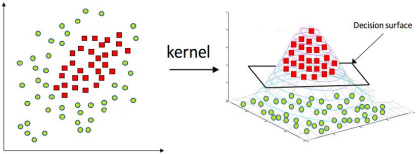
$\arg\max P(A \mid B)$ for each class A

$$\arg\max P(A \mid B) = \arg\max P(B \mid A) \times \frac{P(A)}{P(B)}$$
$$= \arg\max P(B \mid A) \times P(A)$$

- uses Bayes's rule
- $P(B)$ will be the same for all classes
- for a set of features x_i we calculate the joint probability: $B = (x_1, x_2, \ldots, x_n)$
- $\arg\max P(x_1, x_2, \ldots, x_n \mid A) \times P(A)$
  $P(x_1, x_2, \ldots, x_n \mid A) = \prod_i P(x_i \mid A)$
- simply the product of individual probabilities
- assumption: features are independent given a class

What if our data is not linearly separable?





- Use a soft SVM
  - Add terms to objective function to penalise points on the "wrong side" of boundary, but don't forbid them
- Use a kernel function
  - Transform input data into a separable space
  - Kernel SVM allows non-linear boundaries in the original space and is a popular approach

Polynomials of degree exactly d: $K(u,v) = (u \cdot v)^d$

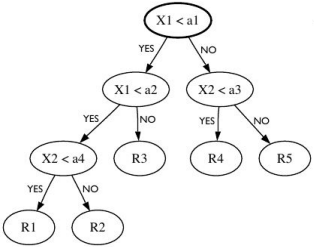
Polynomials of degree up to d: $K(u,v) = (u \cdot v + 1)^d$

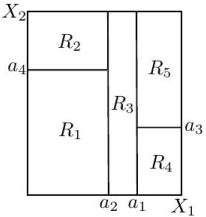
Gaussian kernels: $K(\vec{u},\vec{v}) = \exp\left(-\frac{\|\vec{u}-\vec{v}\|_2^2}{2\sigma^2}\right)$

Sigmoid: $K(u,v) = \tanh(\eta u \cdot v + v)$

Mohri 2012, *Foundations of machine learning*; *What is the kernel trick* via medium.com

How to build a decision tree from data?





1. Choose a decision point yielding best purity
2. Partition data into corresponding subsets
3. Reiterate with resulting subsets
4. Stop when regions are approximately pure

Impurity in classification

- misclassification
- Gini impurity: probability of incorrectly classifying a randomly chosen data point

Impurity in regression

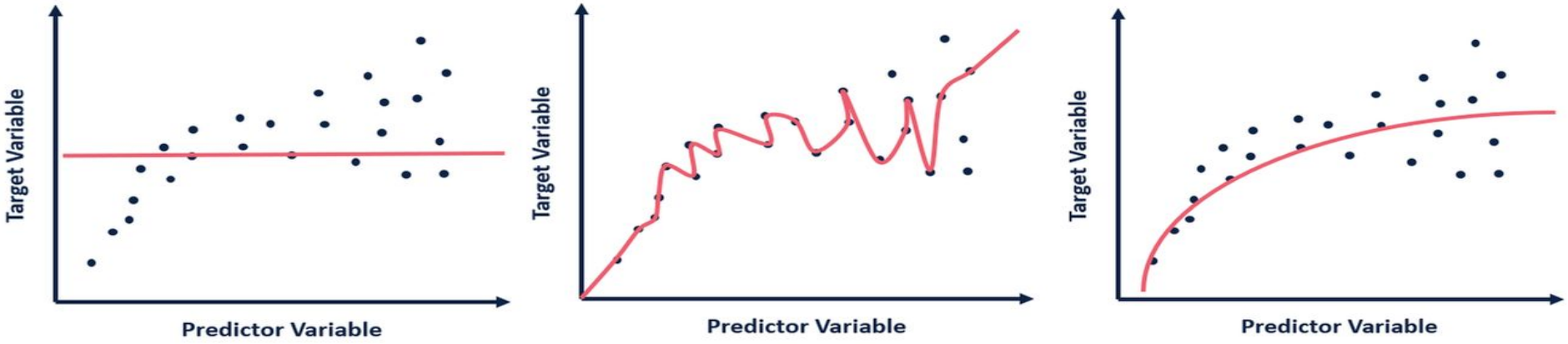
- mean squared error

$$F(R) = \sum_{x_i \in R} (y_i - \langle y \rangle)^2$$

Mohri 2012, *Foundations of machine learning*;

Performance Estimation

- **Overfitting** and **underfitting**

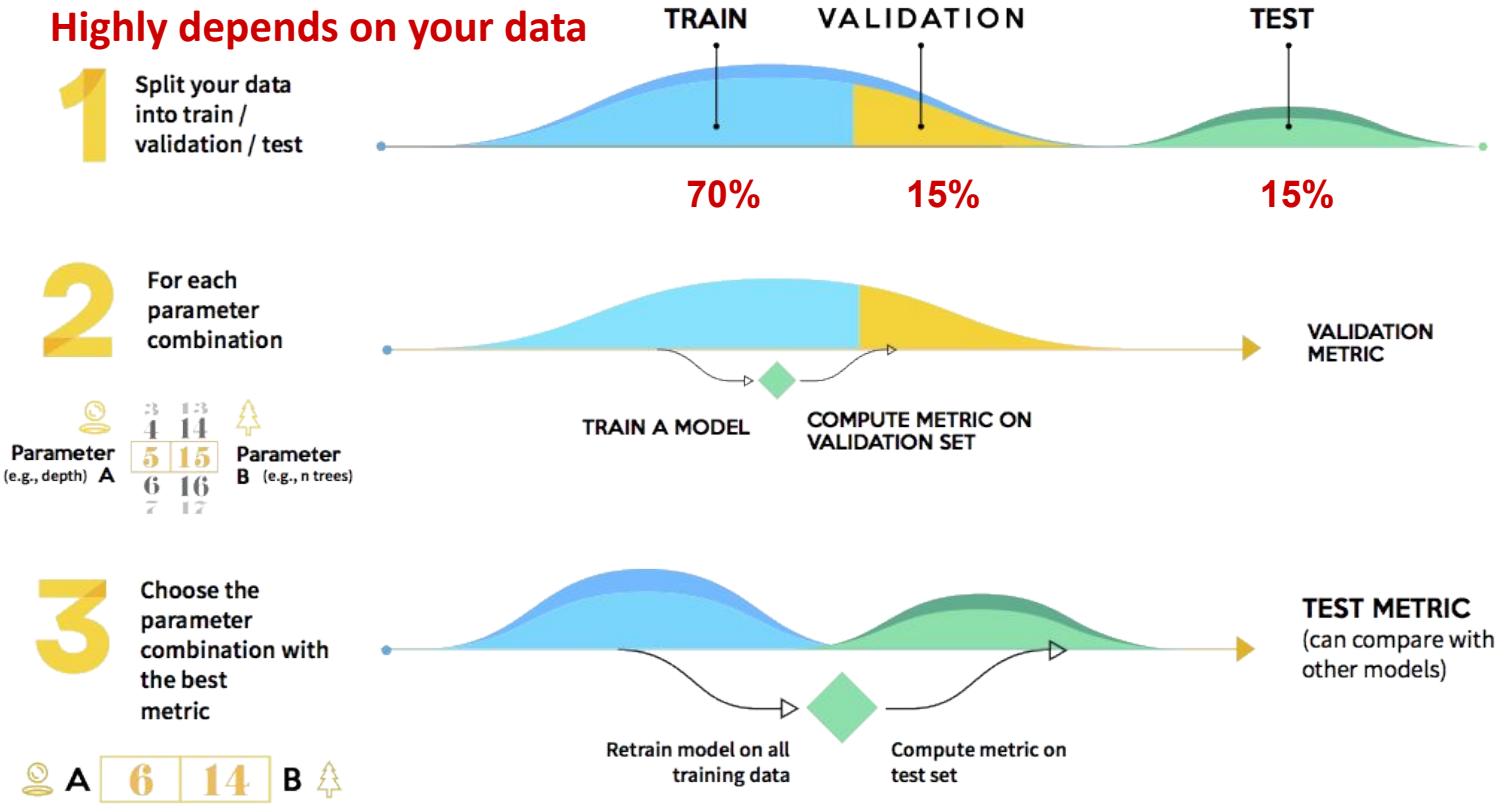


- **Bias**
  - Difference between prediction and real outcome
- **Variance**
  - Variability of predictions

https://community.alteryx.com/t5/Data-Science/Bias-Versus-Variance/ba-p/351862

HOLDOUT STRATEGY

Highly depends on your data



https://www.kdnuggets.com/wp-content/uploads/dataiku-kfold-strategy.jpg

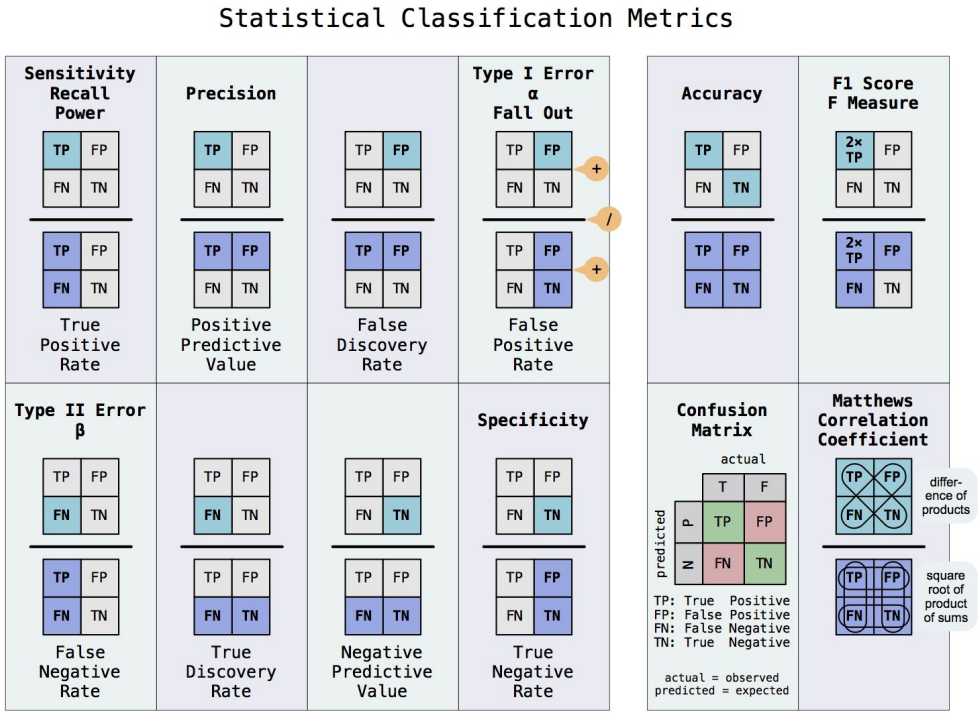
Predictive Performance Metrics

- To select the **best performing model** we need o be able to **compare them**

- **Predictive performance metrics**
  - On the validation set
  - On an independent test set
    - Make sure they are consistent

- There are multiple metrics for **classification** and **regression**

- **Classification**
  - We can derive several metrics from a **confusion matrix**



https://www.bluemontlabs.com/images/statistical-classification-metrics.png

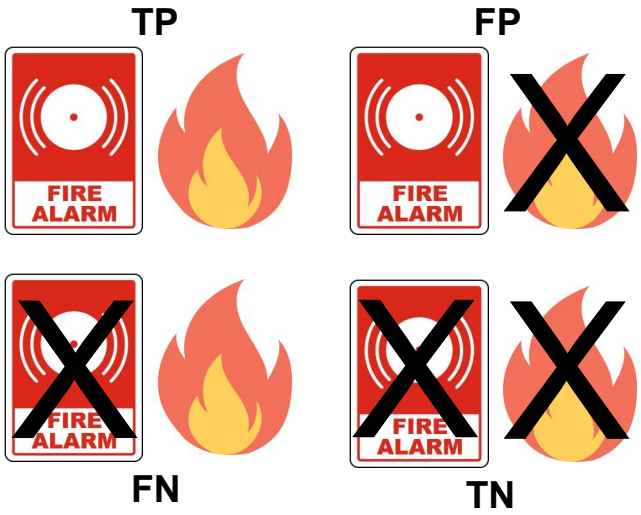
Type I and II errors

- **Which Type Error is worse?**
  - **Fire alarm**

- **Type I:**
  - Fire alarm rings when there is no fire

- **Type II:**
  - Fire alarm fails to ring when there is fire

- **Which Type Error is worse** in this case?



Which models is the best one?

Metric	MODEL1	MODEL2
Recall	0.6667	**0.8333**
Specificity	**0.8333**	0.6667
Precision	**0.8000**	0.7143
Accuracy	0.7500	0.7500
F1 Score	0.7273	**0.7692**
MCC	0.5071	0.5071

MODEL 1	Actual disease	Actual healthy
Predicted disease	200	50
Predicted healthy	100	250

MODEL 2	Actual disease	Actual healthy
Predicted disease	250	100
Predicted healthy	50	200

- **Model 1**
  - To minimise Type I Error (better precision)
- **Model 2**
  - To minimise Type 2 Error (better recall)
- **Benign *vs.* malignant cancer hypothetical cases**