Computer Vision – TP8
Statistical Classifiers

Miguel Coimbra, Hélder Oliveira

U. PORTO FC

Outline

- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation

Topic: Statistical Classifiers

- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation

Statistical PR

- I use **statistics** to make a decision
  - I can make **decisions** even when I don't have full a priori knowledge of the whole process
  - I can make **mistakes**
- How did I **recognize** this pattern?
  - I **learn** from previous observations where I know the classification result
  - I **classify** a new observation

Features

- Feature F_i $F_i = [f_i]$
- Feature F_i with N values.

$$F_i = [f_{i1}, f_{i2}, \ldots, f_{iN}]$$

- Feature vector F with M features.

$$F = [F_1 \mid F_2 \mid \ldots \mid F_M]$$

- Naming conventions:
  - Elements of a **feature vector** are called **coefficients**
  - **Features** may have one or more **coefficients**
  - **Feature vectors** may have one or more **features**

Classifiers

- A **Classifier C** maps a class into the feature space

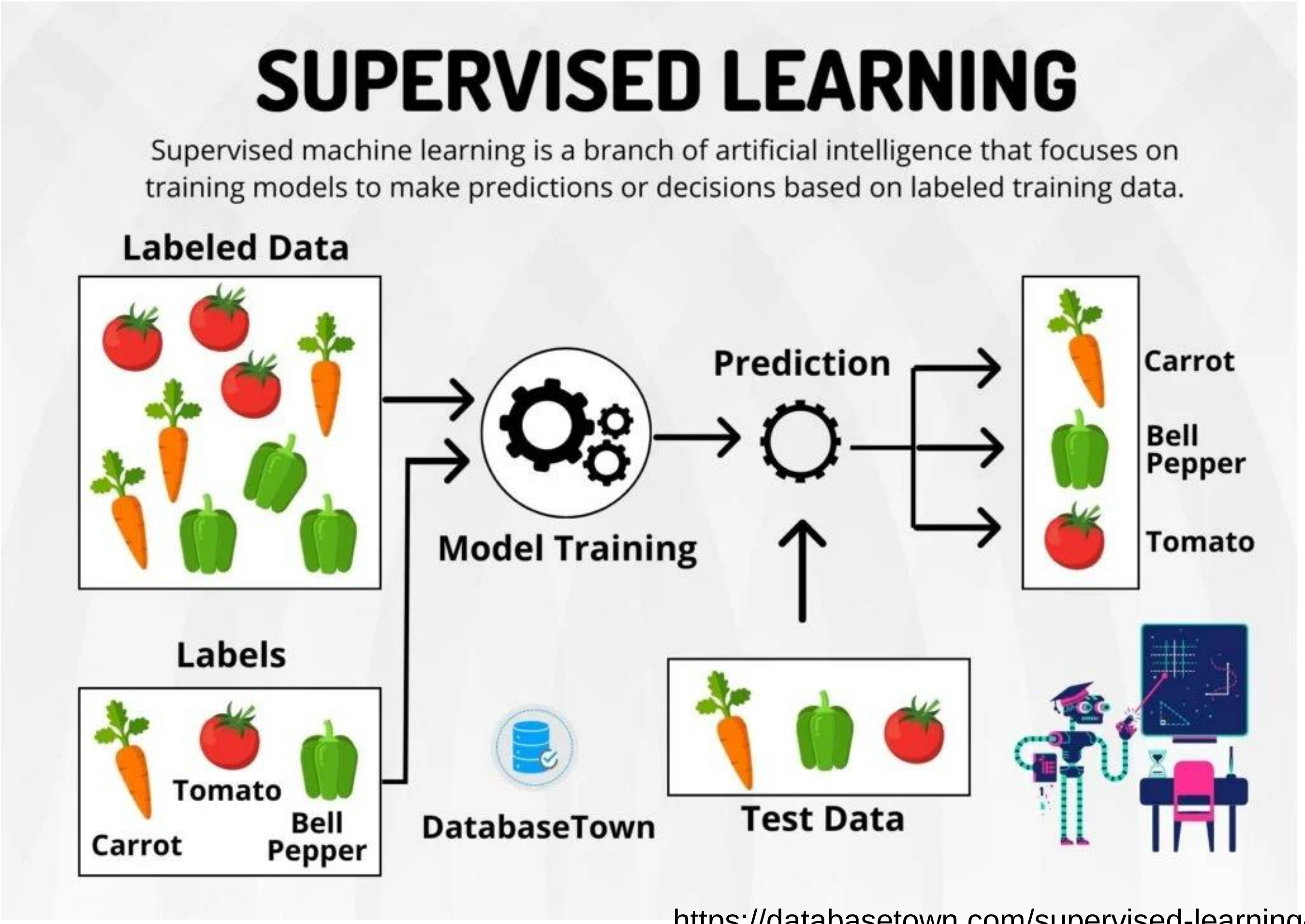
$$C_{\text{Spain}}(x, y) = \begin{cases} true & , y > K \\ false & , otherwise \end{cases}$$

- Various types of classifiers
  - Nearest-Neighbours
  - Support Vector Machines
  - Neural Networks
  - Etc...
- How do I train these classifiers using statistics?

Learning from Statistics

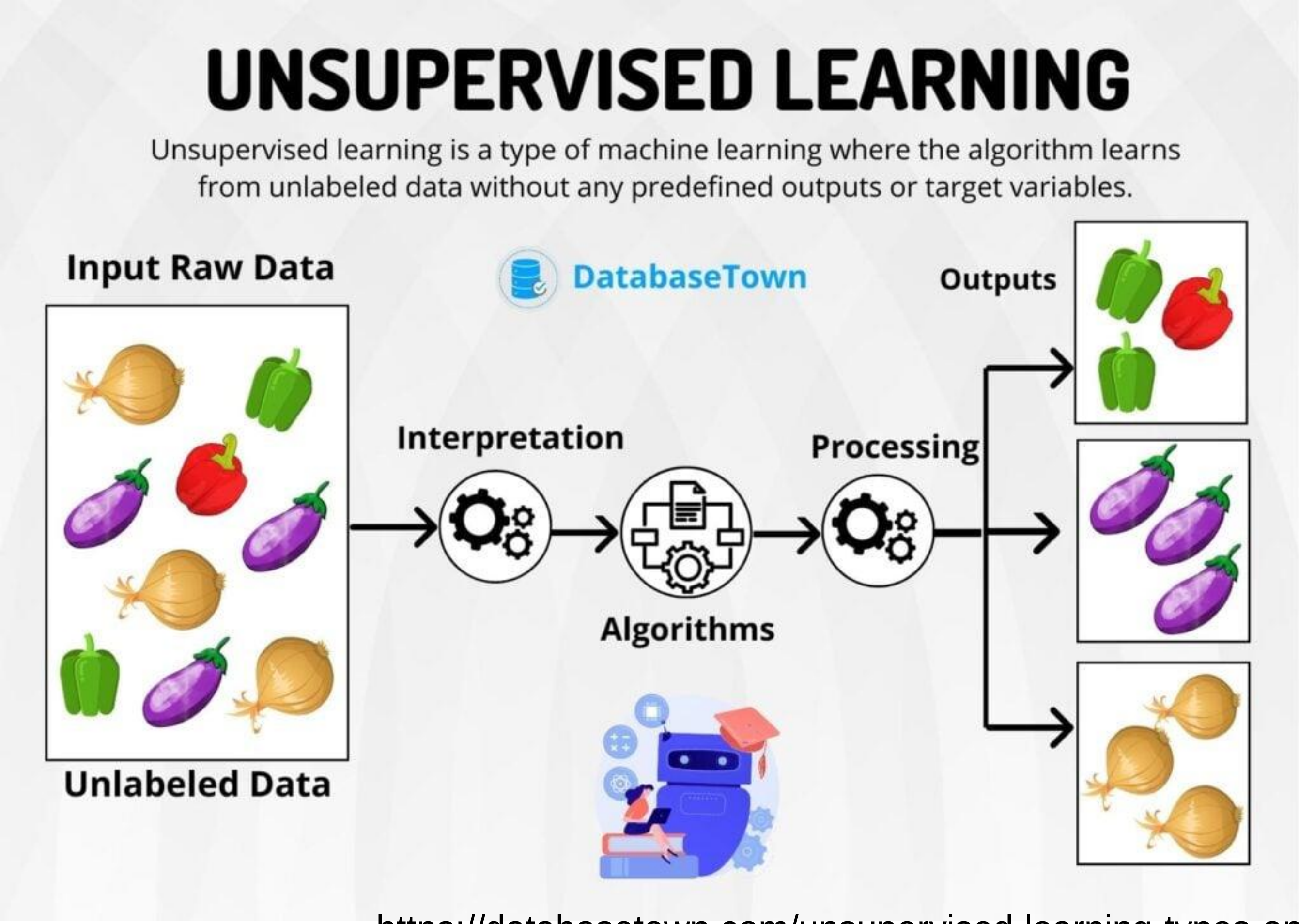
- Supervised Learning
  - Training examples have ‘ground truth’, i.e., their correct class is labelled
  - Easier to learn, requires annotation (expensive, non-trivial)
  - *More popular today in Computer Vision*
- Unsupervised Learning
  - Training examples do not have associated class labels
  - Harder to learn, no annotation means easier access to large datasets
- Semi-supervised Learning
  - Combines training examples with and without labels
  - Compromise between the other alternatives
  - *Hot topic today in Computer Vision (weakly supervised learning)*

Supervised Learning



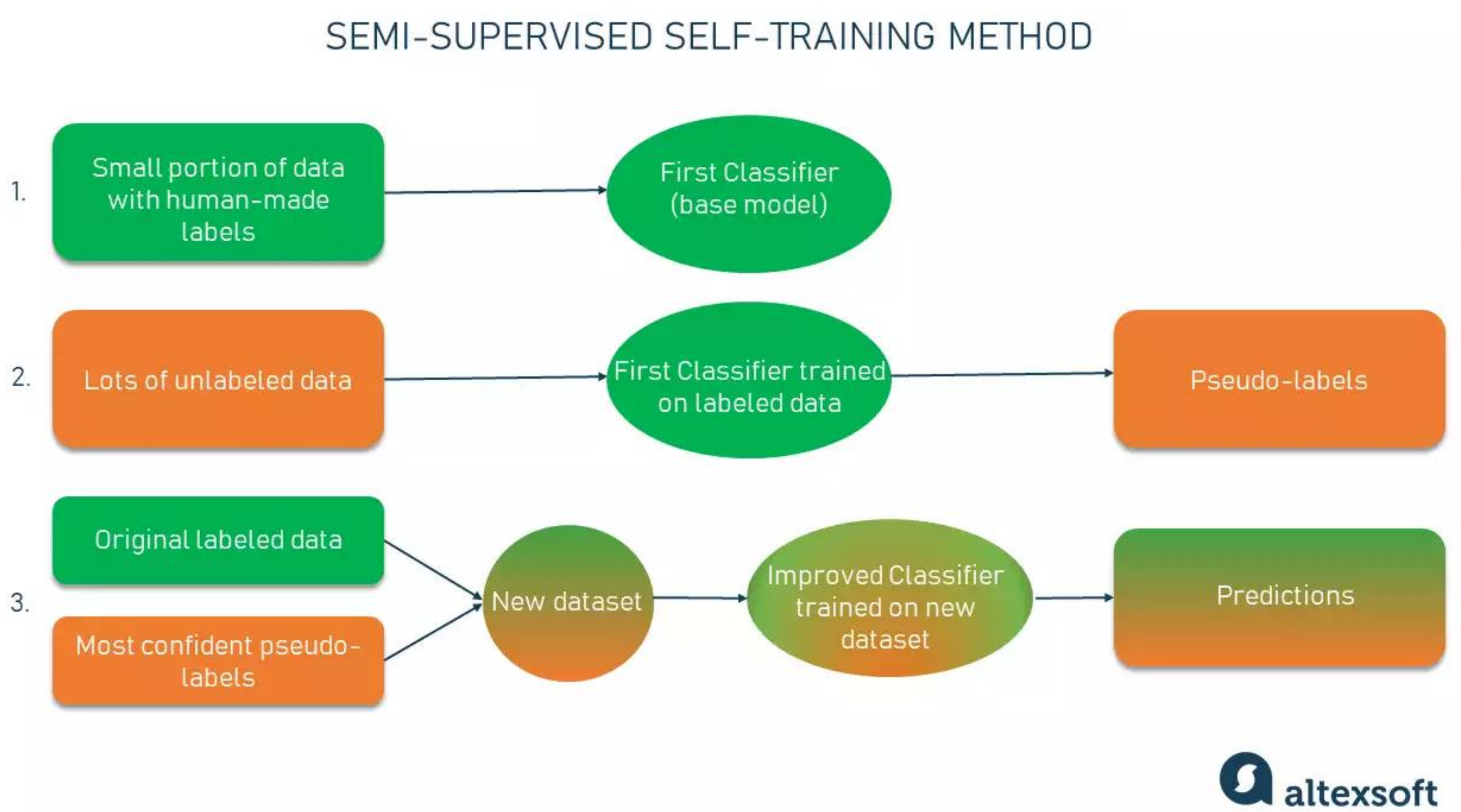
https://databasetown.com/supervised-learning-algorithms/

Unsupervised Learning



https://databasetown.com/unsupervised-learning-types-applications/

Semi-Supervised Learning



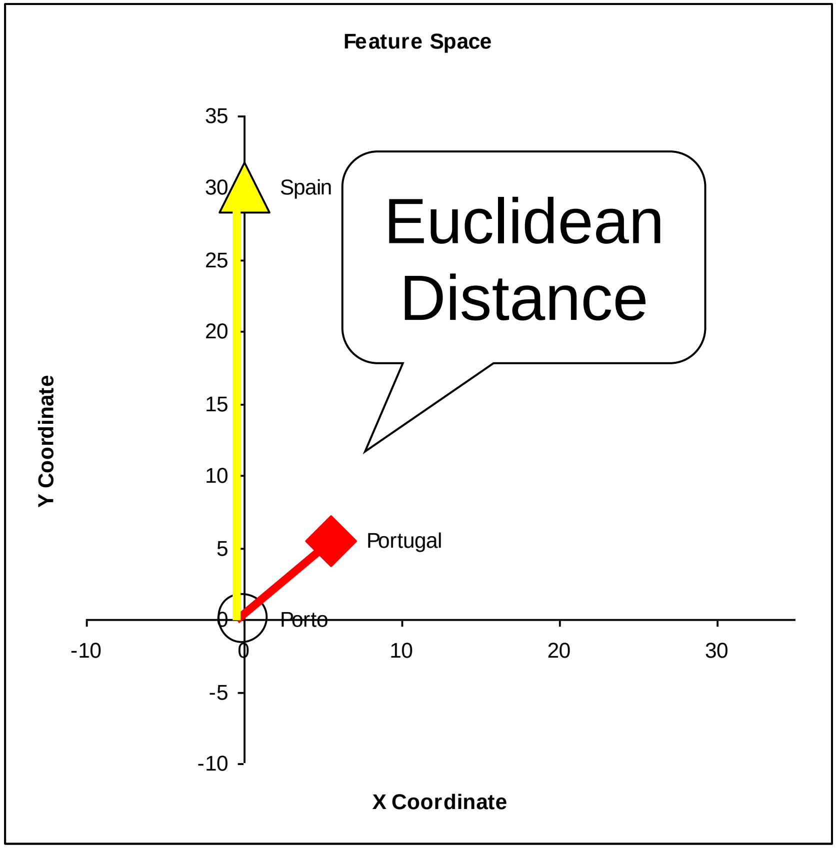
https://www.altexsoft.com/blog/semi-supervised-learning/

Example: Distance to Mean

- I can represent a class by its mean feature vector

$$C = \overline{F}$$

- To classify a new object, I choose the class with the closest mean feature vector
- **Different distance measures!**



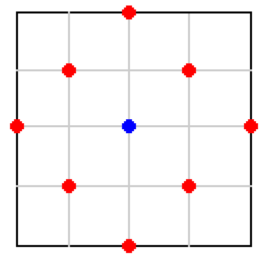
Possible Distance Measures

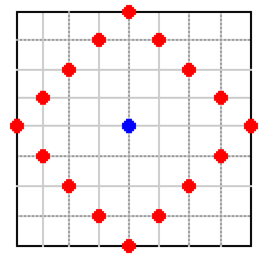
- L1 Distance

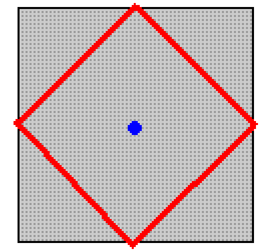
$$L1(x,y) = \sum_{i=1}^{N} |x_i - y_i|$$

- Euclidean Distance (L2 Distance)

$$L2(x,y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$





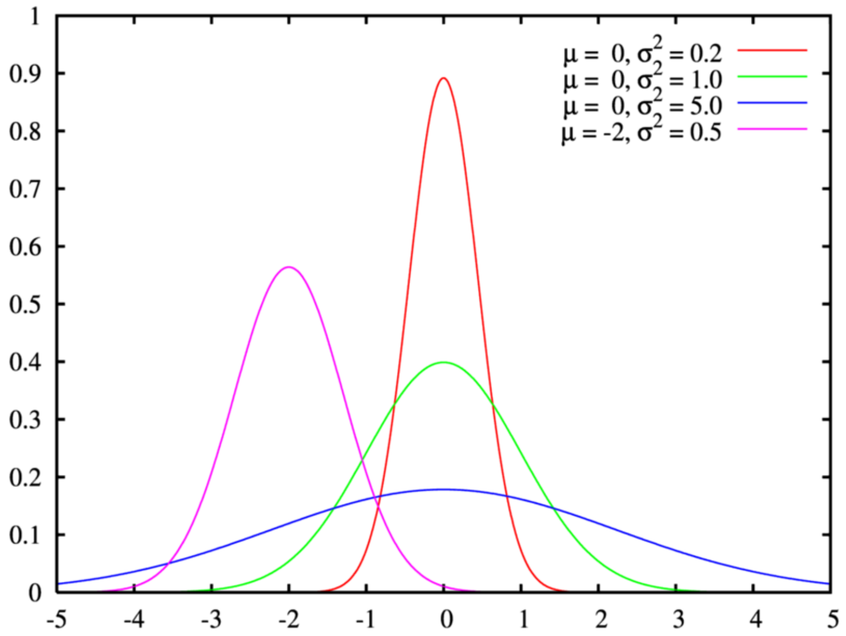


L1 or Taxicab Distance

Gaussian Distribution

- Defined by two parameters:
  - Mean: μ
  - Variance: σ^2
- Great approximation to the distribution of many phenomena.
  - *Central Limit Theorem*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-u)^2}{2\sigma^2}\right)$$



Multivariate Distribution

- For N dimensions:

$$f_X(x_1,\ldots,x_N) = \frac{1}{(2\pi)^{N/2}\,|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

- Mean feature vector:

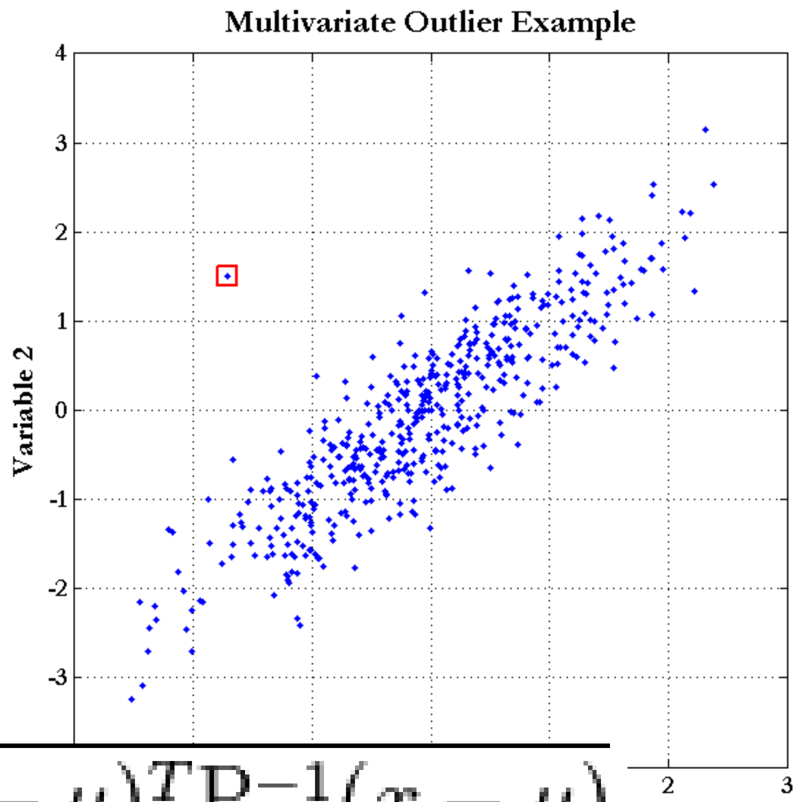
$$\mu = \overline{F}$$

- Covariance Matrix:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \qquad \mu_i = \mathrm{E}(X_i) \qquad \Sigma_{ij} = \mathrm{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big]$$

Mahalanobis Distance

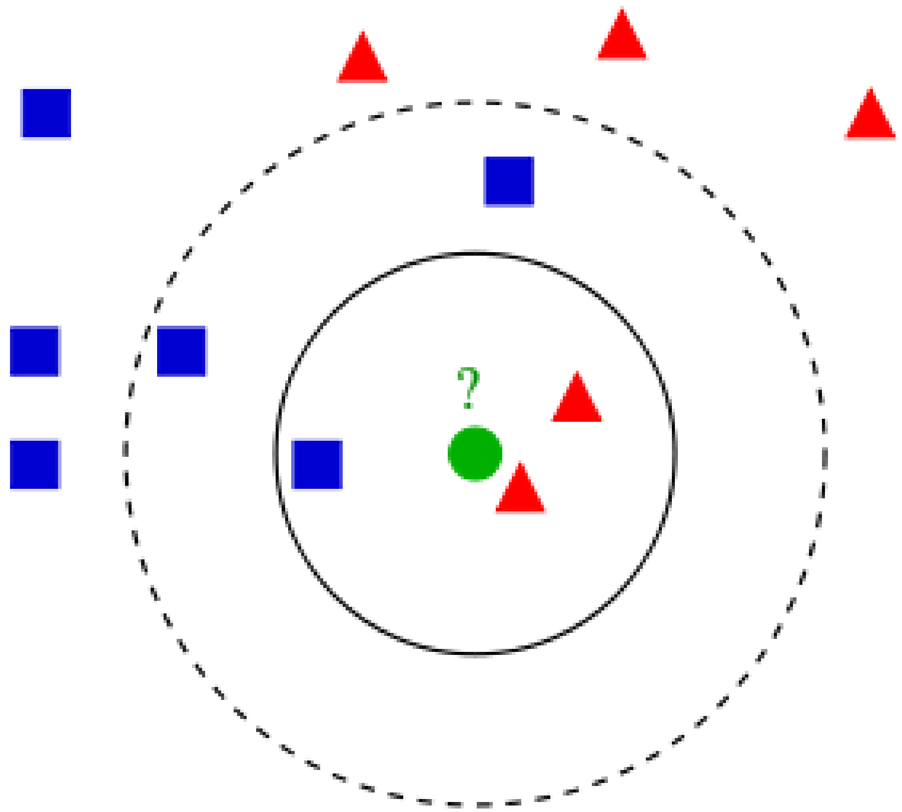
- Based on the covariance of coefficients
- Superior to the Euclidean distance



$$D_M(x) = \sqrt{(x-\mu)^T P^{-1} (x-\mu)}.$$

SL Example: K-Nearest Neighbours

- Algorithm
  - Choose the closest K neighbours to a new observation
  - Classify the new object based on the **class** of these K objects
- Characteristics
  - Assumes no model
  - Does not scale very well...



Other Classifier Examples

- Supervised Learning
  - Linear Regression
  - Logistic Regression
  - Decision Trees
  - Random Forests
  - Support Vector Machines
  - Neural Networks
- Unsupervised Learning
  - K-Means Clustering
  - Hierarchical Clustering
  - Principal Component Analysis
  - Association Rule Mining

Topic: Generalization

- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation

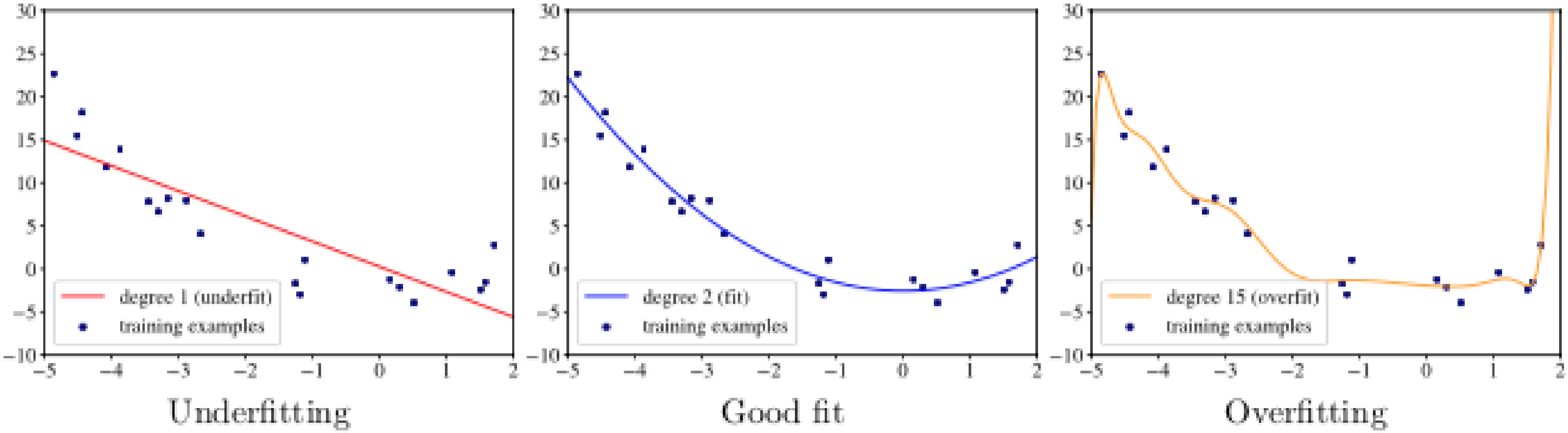
Generalization

- Classifiers are optimized to reduce training errors
  - (supervised learning): we have access to a set of training data for which we know the correct class/answer

- What if test data is different from training data?

Underfitting and Overfitting

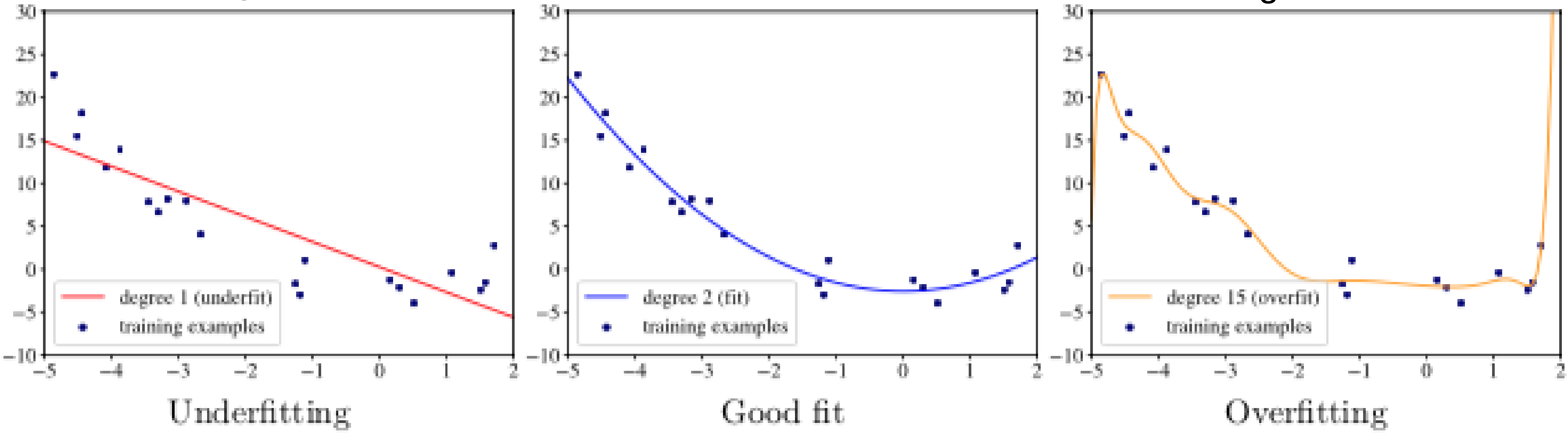
- Is the model too simple for the data?
  - Underfitting: cannot capture data behavior
- Is the model too complex for the data?
  - Overfitting: fit perfectly training data, but will not generalize well on unseen data



Bias and variance

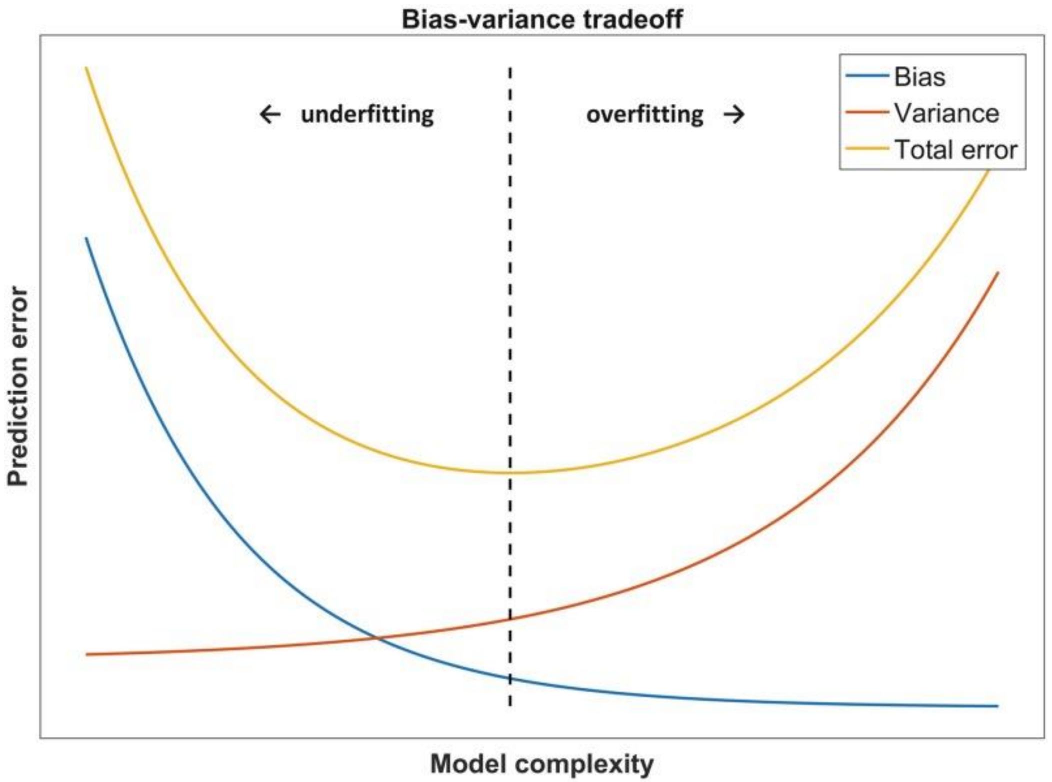
- Bias
  - Average error in predicting correct value
- Variance
  - Variability of model prediction

High bias | Low bias, low variance | High variance



Bias-variance tradeoff

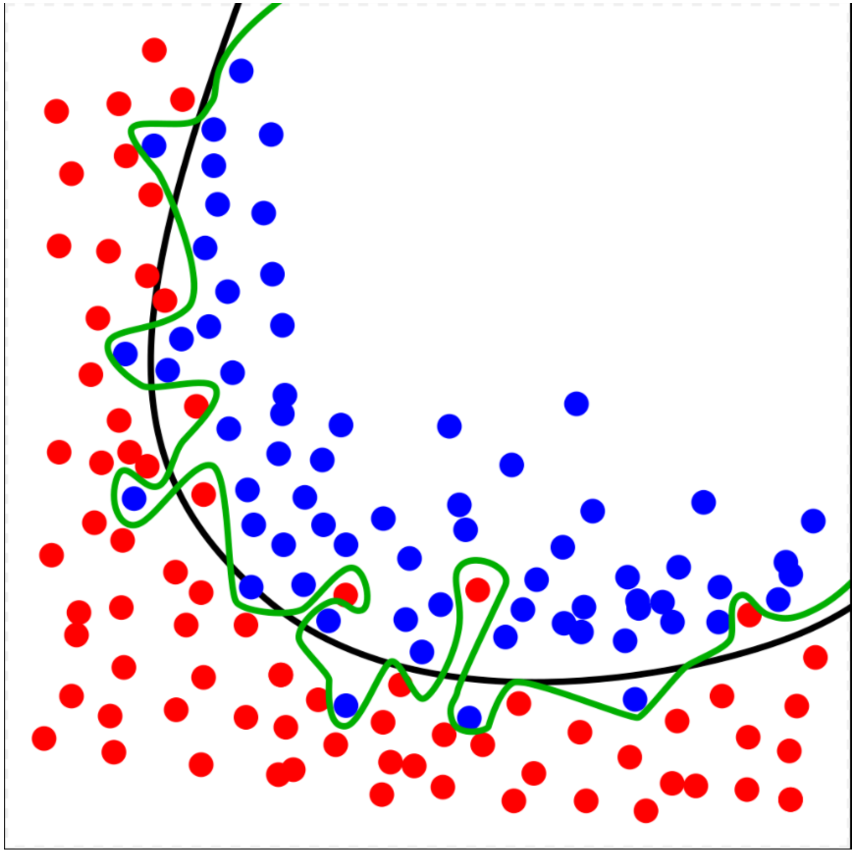
- total err = bias^2 + variance + irreducible err



Topic: Overfitting

- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation

Overfitting



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

- **Overfitting**
  - Analysis that corresponds too closely or exactly to a particular set of data
  - May fail to fit to additional data or predict future observations reliably

Overfitted Models

- Mathematical model that **contains more parameters than can be justified by the data**
- Model will unknowingly **extract some of the residual variation (i.e., the noise)** as if that variation represented underlying model structure

Everitt B.S., Skrondal A. (2010), *Cambridge Dictionary of Statistics, Cambridge University Press.*

Burnham, K. P.; Anderson, D. R. (2002), Model Selection and Multimodel Inference (2nd ed.), Springer-Verlag.

Strategies to Address Overfitting: Regularization

"Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error."

Ian Goodfellow, Yushua Bengio, Aaron Courville, "Deep Learning", London: The MIT Press, 2017

Weight regularization

- Reduce the generalization error by imposing constraints on the weights

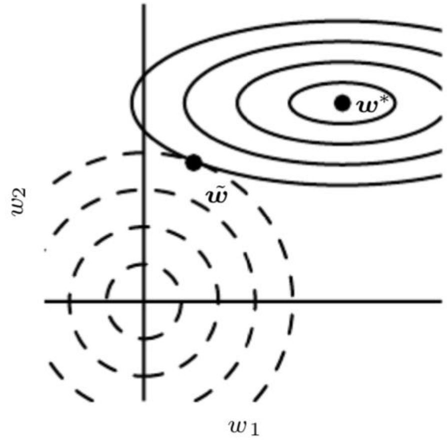
- Modifies the loss function in order to force some structure on the learned weights

$$L'(\theta, \{(x_i, y_i)_i\}) = L(\theta, \{(x_i, y_i)_i\}) + \gamma\Omega(\theta)$$

- Different Ω, different effect on the weights

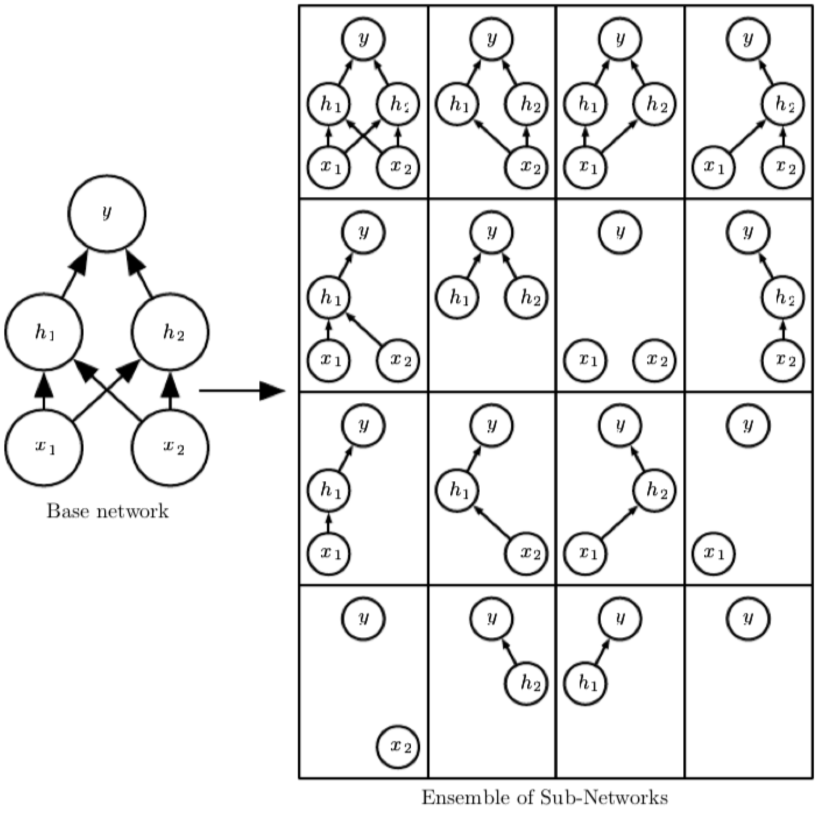
Weight decay

- Weight decay: $\Omega(\theta) = \|\theta\|_2^2$
  - Drives the weights closer to the origin
  - Weight components that do not impact significantly the loss function are decayed



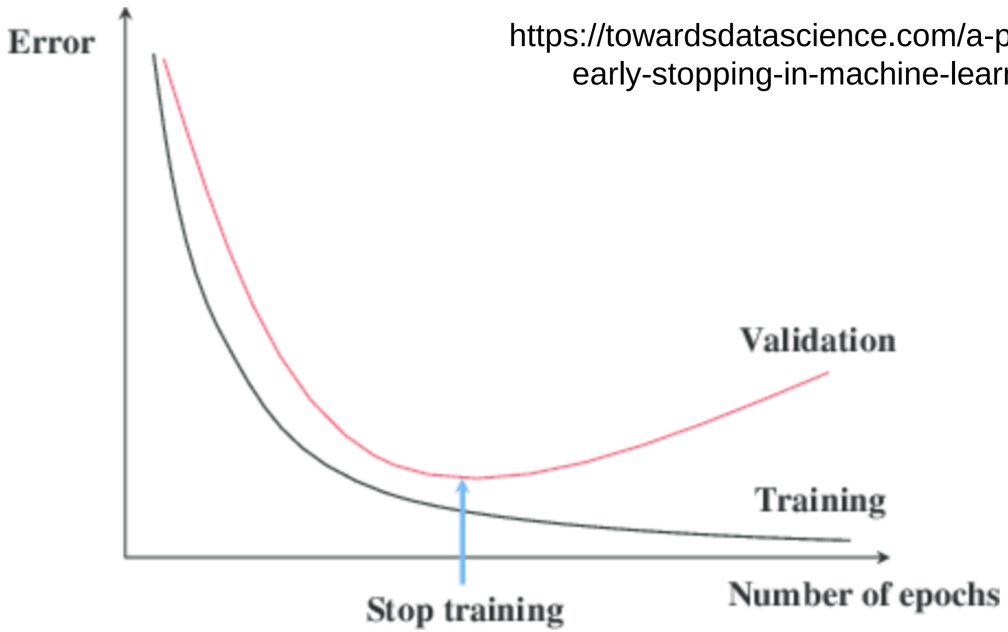
Strategies to Address Overfitting: Dropout

- During training, randomly switch off a fraction of the input or hidden units
- It avoids giving too much relevance to some training features
- It approximates bagging and ensemble learning over all sub-models (Monte-Carlo sampling)



Strategies to Address Overfitting: Early stopping

- Retain the model which performs best on the validation set (hopefully, test set too)

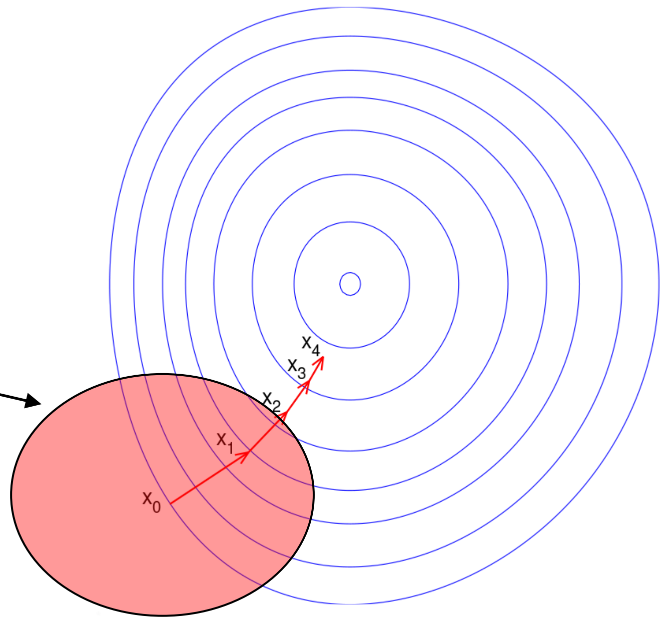


https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd

Early stopping

- Regularization effect: constraint on the number of training steps

Region of
feasible weights

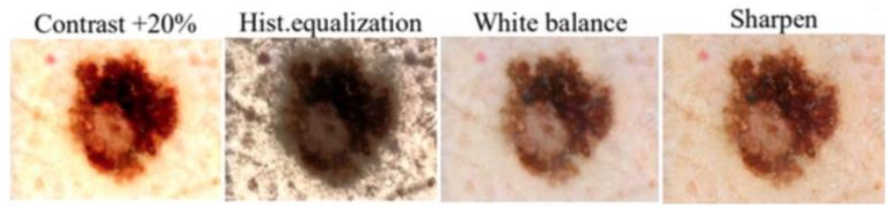


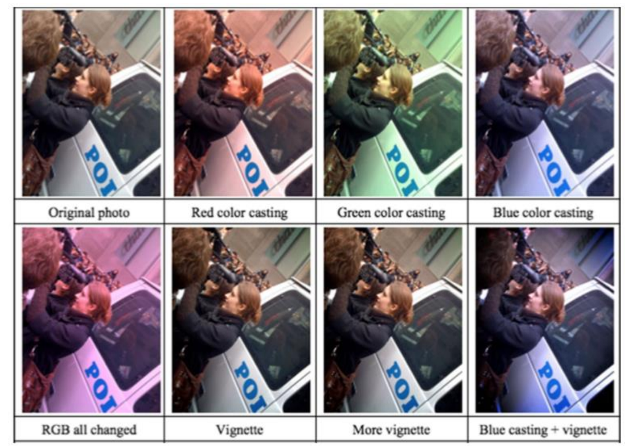
Strategies to Address Overfitting: Data augmentation

- Create fake data and add it to the **training dataset** (only training!)
- Especially useful for imaging data
- New data created from transformations of existing training data:
  - Different transformations may be more meaningful in different domains
  - A transformation should not change class meaning

Data augmentation

- Transformations:
  - Translating
  - Rotating
  - Cropping
  - Flipping
  - Color space
  - Adding noise
  - Image mixing
  - Generative Adversarial Networks (GANs)
  - Etc.

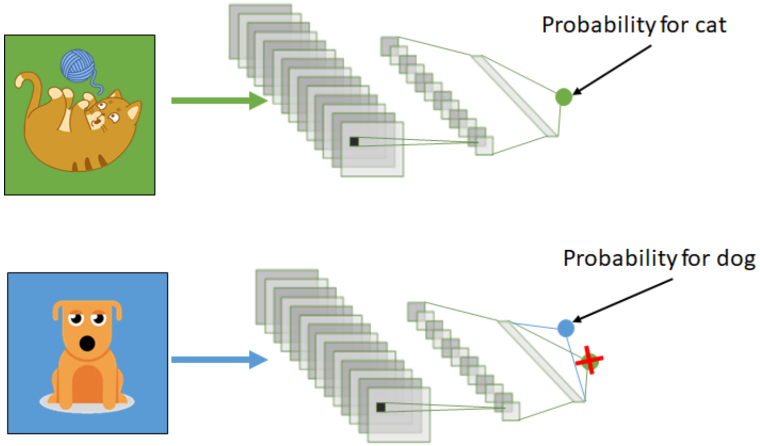




Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of Big Data. 2019 Dec;6(1):1-48.

Strategies to Address Overfitting: Transfer learning

- Main idea:
  - Features to perform a task T1 may be relevant and useful for a different task T2



https://towardsdatascience.com/transfer-learning-3e9bb53549f6

Transfer learning

- When is it useful:
  - Reduced number of training samples for the considered task
  - Large number of training samples for a related task
  - Low-level features could be common to both tasks!

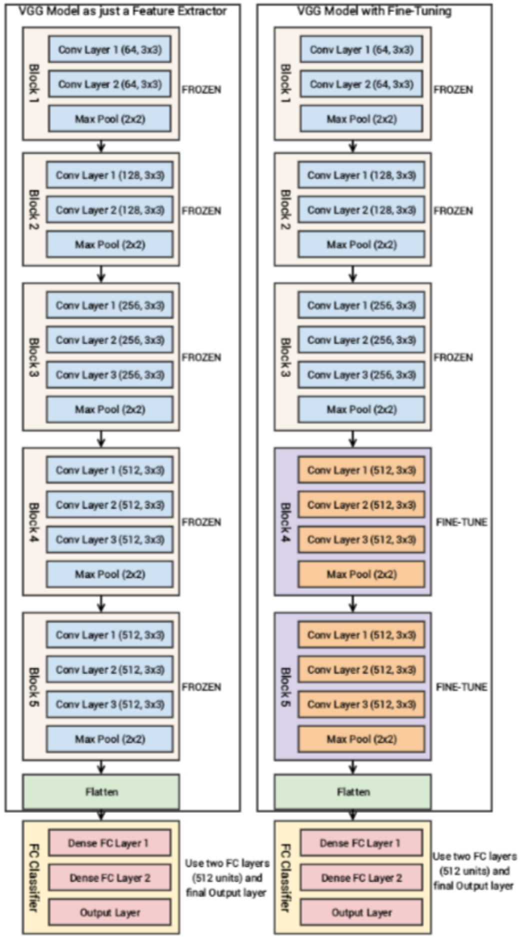
- Example:
  - Image classification
  - NNs pre-trained on the ImageNet dataset (~14 million images, ~20,000 categories)

Transfer learning schemes

https://towardsdatascience.com/a-comprehensive-hands-on-guide-to- transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

- Feature extraction:
  - Keep convolutional layers frozen
  - Pre-trained networks works as feature extractor
  - Train fully connected/classification layers

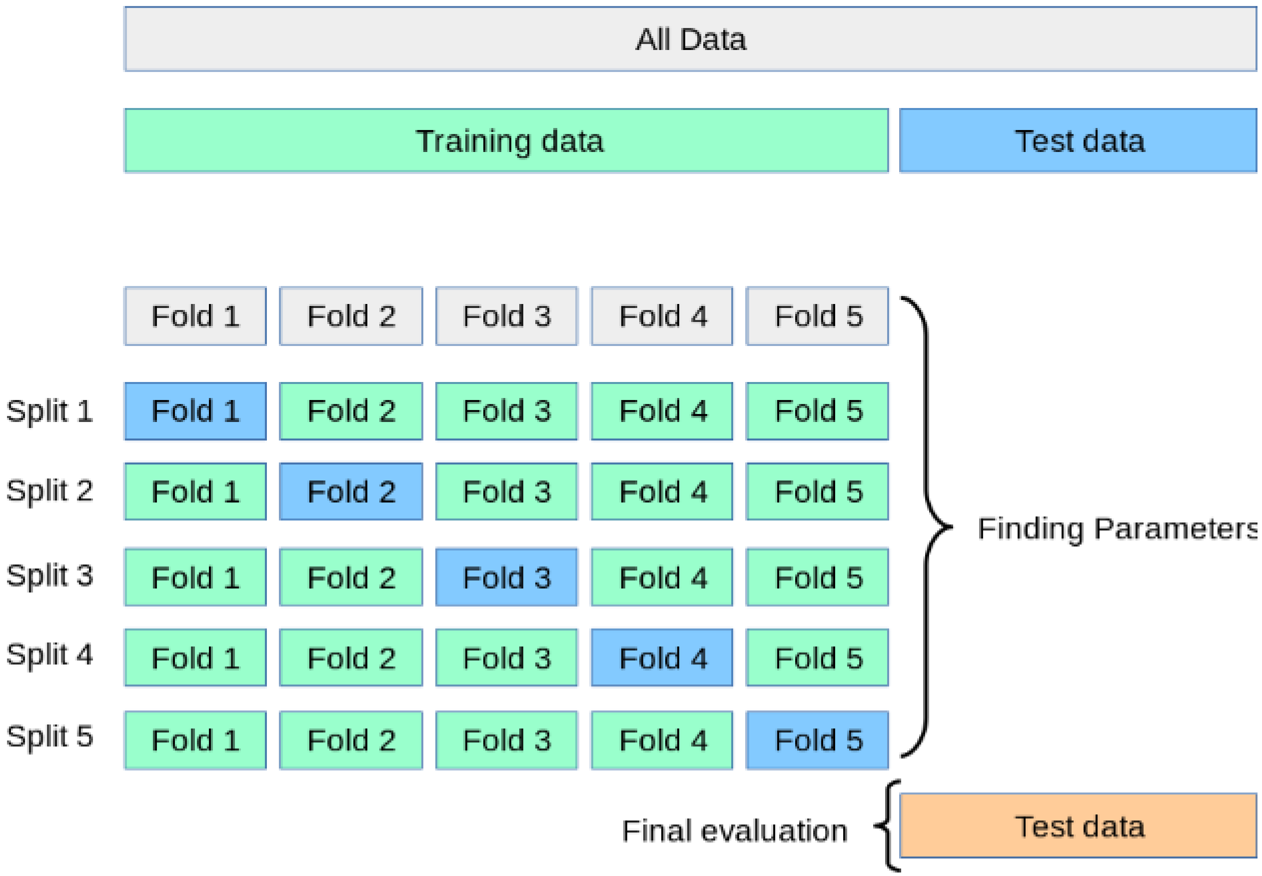
- Fine-tuning:
  - Use pre-trained weights as starting point for training
  - Can keep frozen first convolutional layers (mostly edge/geometry detectors)



Topic: Cross-Validation

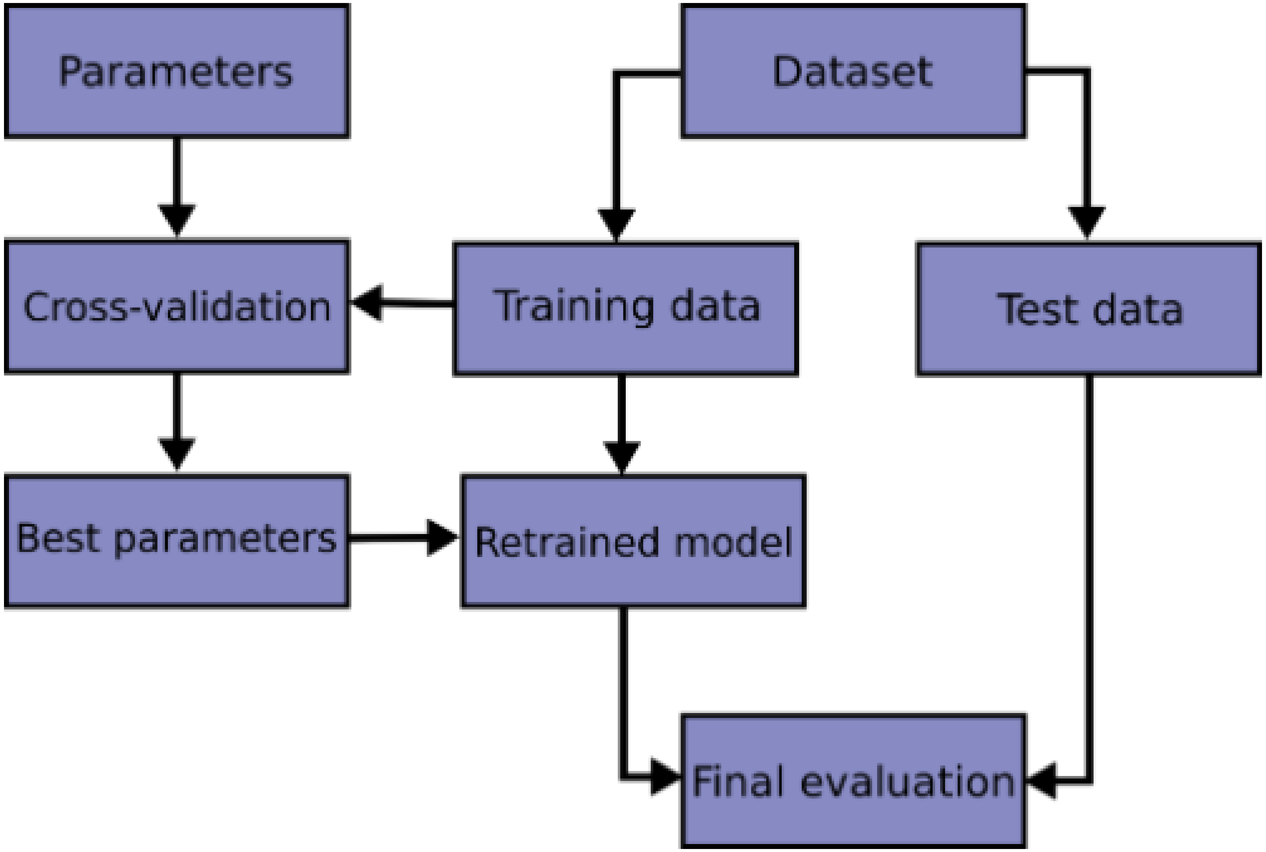
- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation

Cross-validation



https://scikit-learn.org/stable/modules/cross_validation.html

Cross-validation



https://scikit-learn.org/stable/modules/cross_validation.html

Cross-validation (other options)

- K-fold
- Repeated K-fold
- Leave One Out
- Leave P Out
- Random permutations cross-validation

Summary

- Statistical Classifiers
- Generalization
- Overfitting
- Cross-Validation