DCML-CPS - Module 6

Supervised ML

Tommaso Zoppi
University of Florence - Florence (Italy)
tommaso.zoppi@unifi.it

Course Map

1. Basics and Metrology
2. Monitoring

Monitoring

Testing

3. Fault Injection
4. Robustness Testing

5. Data Analysis
6. Supervised ML
7. Unsupervised ML
8. Meta-Learning
9. Error/Intrusion Detection

Anomaly Detection

Tools & Libs

Deep Learning

Supervised Learning

- Classifiers were usually meant to be supervised
  - Use labels in data during training

    They NEED ground truth!

  - This way, they learn both normal behaviour and specific alterations due to known errors/attacks
- Non-Sliding Algorithms are very famous and usually build the core of any Machine Learning course.
  - Here we are presenting the baseline idea of some of them, without expanding on the insights
  - Just enough to use them for meaningful analyses!

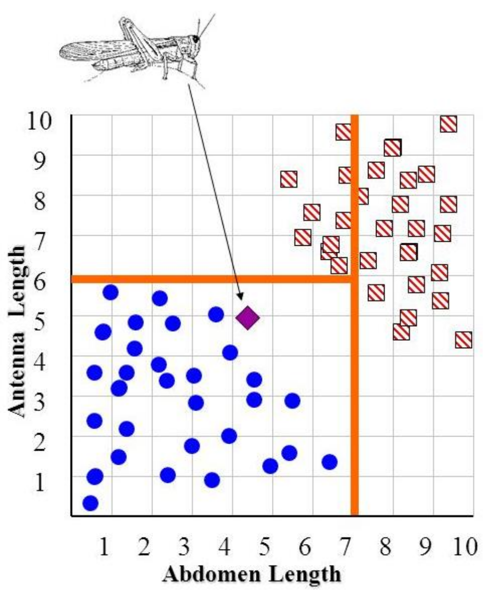
Supervised Algorithms

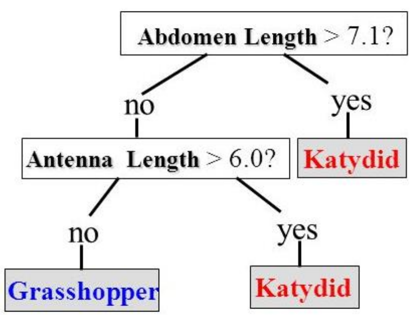
- In the followings we will see an overview of the following supervised algorithms
  - Tree-Based
    - Decision Tree, Random Forest
  - Neighbour-based
    - kNN
  - Statistical
    - Naïve Bayes, LDA, Logistic Regression
  - Neural Networks
    - MultiLayer Perceptron

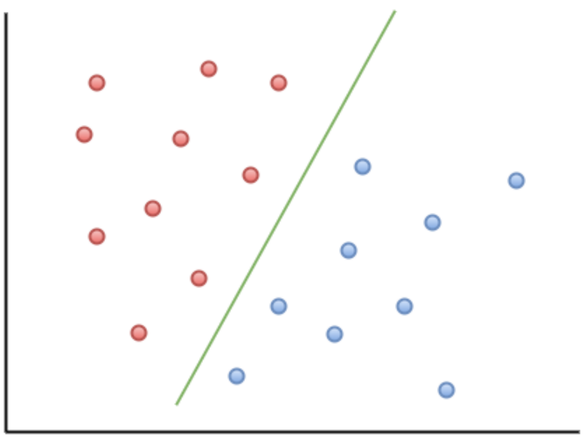
Tree-Based

Classification

- Most supervised algorithms are suitable for binary decisions
- They aim at learning a linear or non-linear boundary
  - To differentiate between normal and anomalous data points

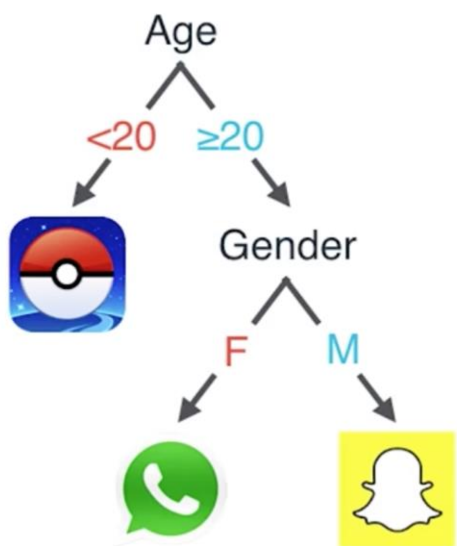






Classification: Decision Tree

- We start from the baseline of Tree-based classification.
- Decision trees aim at partitioning the input space, labeling each partition according to its class
  - Each internal node of a tree specifies a "split" based on a feature



Ways to “split” in Decision Trees

- Gini Index: the **gini impurity** is calculated using the following formula:
  - Where p_j is the probability of class j.
  - The gini impurity measures the frequency at which data points will be mislabelled if randomly labeled.
  - The minimum value of the Gini Index is 0.
    - This happens when the node is **pure**, this means that all the contained elements in the node are of one unique class.
  - Thus, the optimum split is chosen by the features with less Gini Index

$$GiniIndex = 1 - \sum_j p_j^2$$

Ways to "split" in Decision Trees

- Entropy: The **entropy** is calculated using the following formula:
  - Where, as before, p_j is the probability of class j.
  - Entropy is a measure of information that indicates the disorder of the features with the target.
  - Similar to the Gini Index, the optimum split is chosen by the feature with less entropy.
  - It gets its maximum value when the probability of the two classes is the same and a node is pure when the entropy has its minimum value, which is 0.

$$Entropy = -\sum_{j} p_j \cdot log_2 \cdot p_j$$

Building a Decision Tree - I

- Example from start to finish
  - Problem: will you play outside depending on the current weather, temperature, humidity and wind?

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Partially taken from https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/

Building a Decision Tree - II

- First split: Gini to be calculated for each feature
- Weather: Sunny 3/10, Cloudy 3/10, Rainy 4/10
  - When sunny, 1/3 you play, 2/3 you don't
    - Gini(sunny) = $1-((1/3)^2 + (2/3)^2) = 4/9$
  - When cloudy, you always play
    - Gini(cloudy) = $1-((1)^2) = 0$
  - When rainy, $\frac{1}{4}$ you play, $\frac{3}{4}$ you don't
    - Gini(rainy) = $1-((1/4)^2 + (3/4)^2) = 6/16$

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Gini(weather) =

p(sunny)*gini(sunny) + p(cloudy)*gini(cloudy) + p(rainy)*gini(rainy) =

3/10 * 4/9 + 3/10 * 0 + 4/10 * 6/16 = 2/15 + 3/20 = 14/60 = 7/30

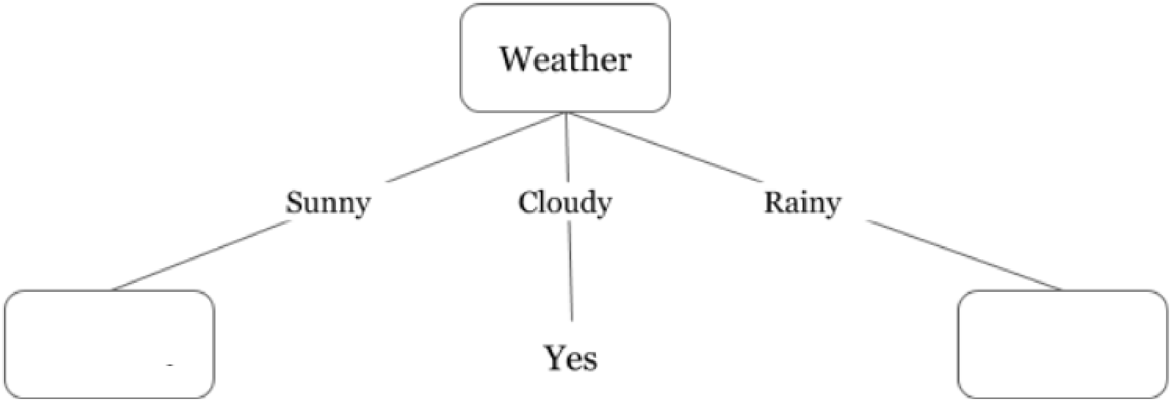
Building a Decision Tree - III

- First split: Gini to be calculated for each feature
  - Gini has to be calculated for others

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

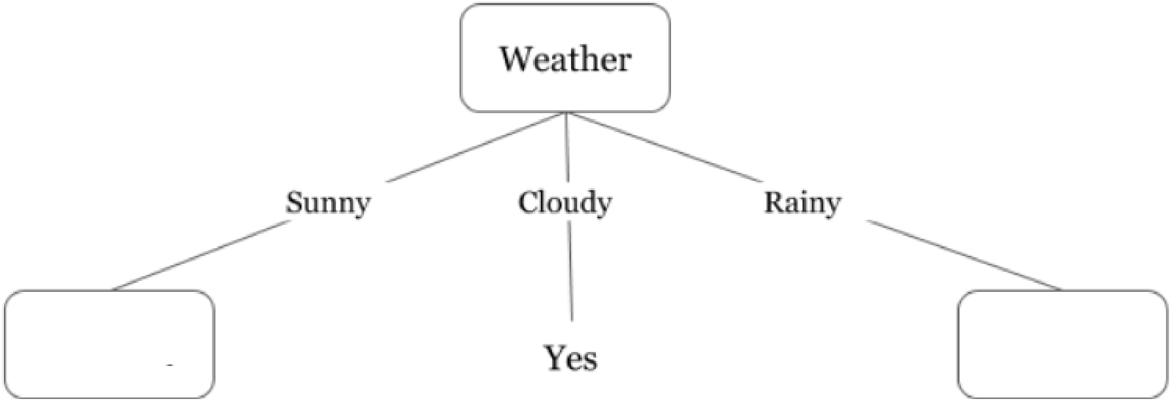
Building a Decision Tree - IV

- First split: Gini to be calculated for each feature
  - Gini(weather) = 7 / 30
  - Gini(Temperature) = 11 / 25
  - Gini(Humidity) = 10 / 21
  - Gini(Wind) = 5 / 12
- Gini(weather) is the lowest, therefore the first layer of the tree is



Building a Decision Tree - V

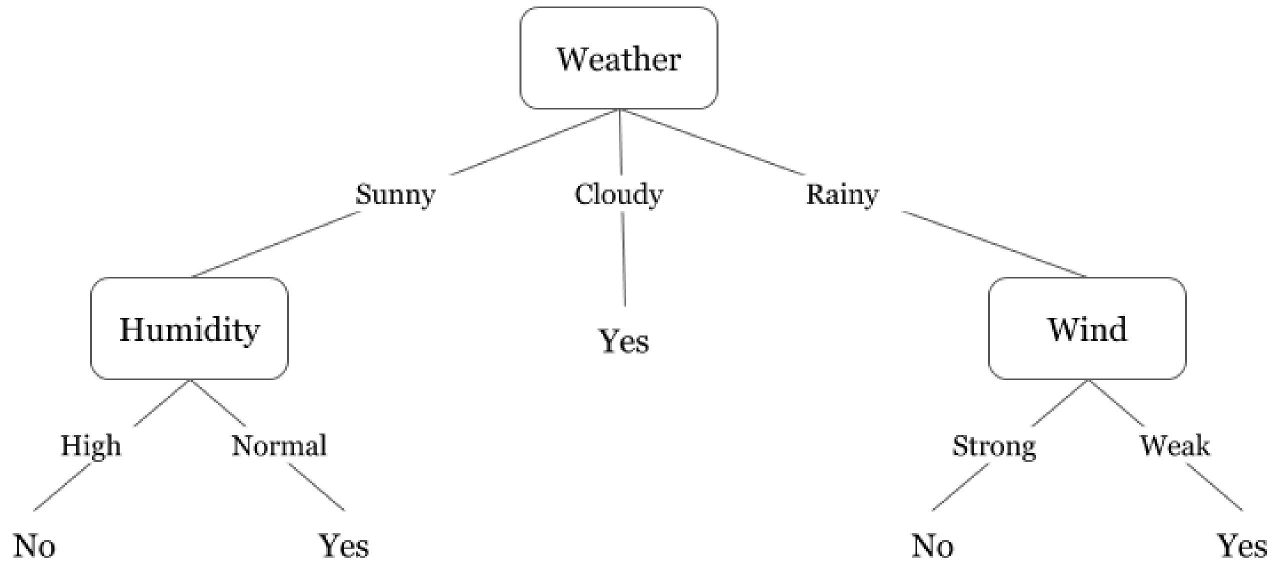
- The process iterates for all sub-branches which do not have a clear label
  - "Cloudy" branch is already ok
- We calculate Gini for the other 3 features
  - Only for "sunny" data
  - Only for "rainy" data



Building a Decision Tree - VI

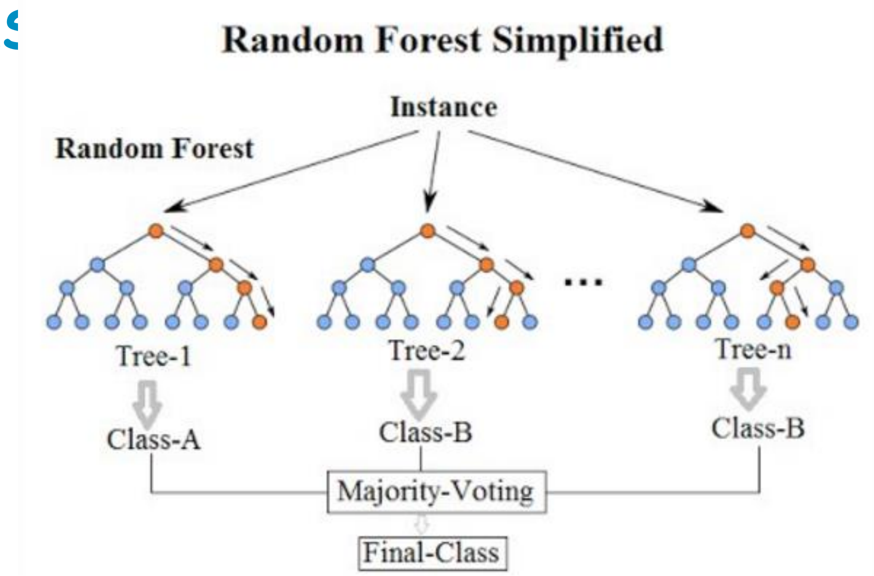
- For sunny data, the lowest gini is Humidity
- For rainy data, the lowest gini is Wind

▶ Then, the process ends because there is no need to split anymore



From Trees to Random Forests

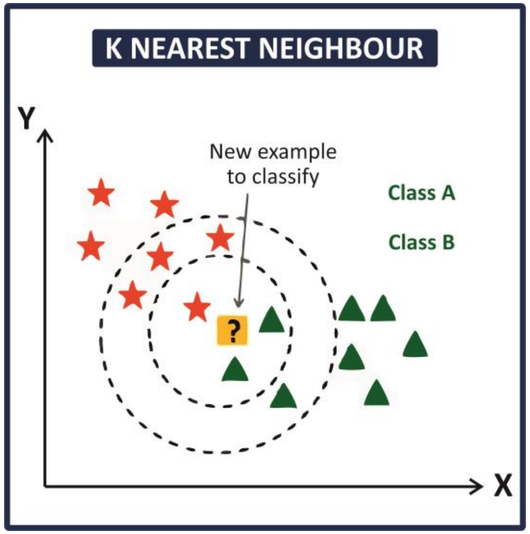
- Random Forests build multiple decision trees
  - Each tree uses a slightly different subset of training set
  - Classifier result is build as a majority voting of individual ans



Neighbour-Based

Neighbour-Based

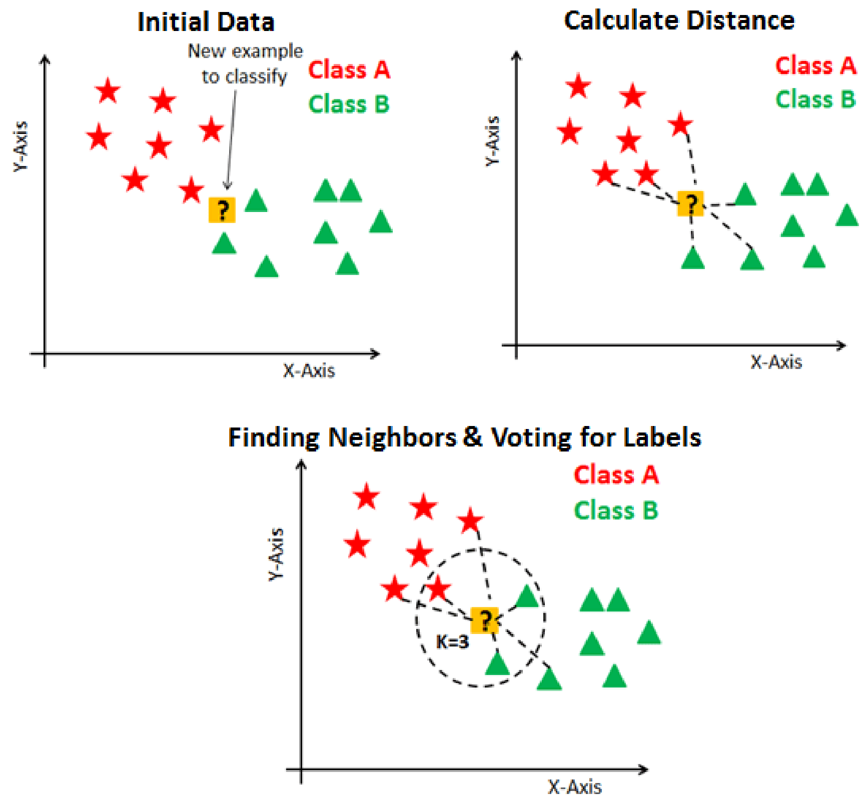
- Assigns the class to a novel data point depending on the class the majority of its "neighbours" belong to
  - Neighbourhood is generally derived through Euclidean distance



From: http://test.basel.in/product/knn-naive-bayes-classifier-using-excel/

Neighbour-Based: kNN

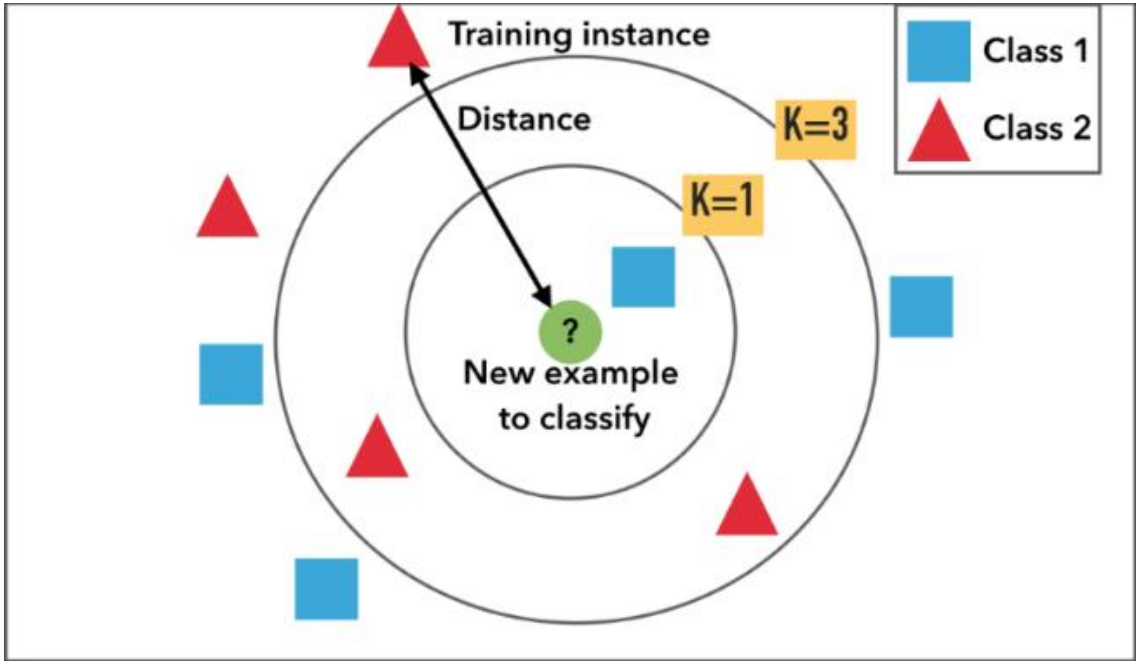
- Typical algorithm is the kNN (k-th Nearest Neighbour)
  - Calculates the k nearest (lower Euclidean distance) neighbours
  - Uses their labels to decide on a new data point



https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

Neighbour-Based: kNN

- The parameter k has big impact
  - Example below:
    - K=1 -> new example classified as SQUARE (C1)
    - K=2 -> new example classified as undefined (k should be even)
    - K=3 -> new example classified as TRIANGLE (C2)



Statistical

Statistical Algorithms

- Statistical algorithms
  - exploit distributions or statistical indexes
  - to first model the data and then
  - predict classes for novel data points
- They are very different among themselves
- We will see 3 different algorithms based on different statistical mechanisms

Naïve Bayes Classifier

- Based on the Bayes Theorem
  - Briefly, during training it aims at learning a statistical model that minimizes the probability of misclassif

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k).$$

  - For each class C_k (K=2 in binary classification), the predicted class ý is the one that maximises the product of n probabilities that a feature value of the data point belongs to that class (n = # feat)
    - Other details are out of scope in this course

Devroye, L.; Gyorfi, L. & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer. ISBN 0-3879-4618-7.

Example (from Wikipedia) - I

- Problem: classify whether a given person is a male or a female based on the measured features.

Person	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Example (from Wikipedia) - II

- The classifier created from the training set using a Gaussian distribution assumption would be (given variances are *unbiased* sample variances)

Person	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033×10^{-2}	176.25	1.2292×10^{2}	11.25	9.1667×10^{-1}
female	5.4175	9.7225×10^{-2}	132.5	5.5833×10^{2}	7.5	1.6667

- The following example assumes equiprobable classes so that P(male)= P(female) = 0.5. This prior probability distribution might be based on prior knowledge of frequencies in the larger population or in the training set.

Example (from Wikipedia) - III

$$\text{posterior (male)} = P(\text{male})\, p(\text{height} \mid \text{male})\, p(\text{weight} \mid \text{male})\, p(\text{foot size} \mid \text{male})$$

$$\text{posterior (female)} = P(\text{female})\, p(\text{height} \mid \text{female})\, p(\text{weight} \mid \text{female})\, p(\text{foot size} \mid \text{female})$$

- Need to calculate both
  - And understanding what is bigger
  - Also, p(male) = p(female) = 0.5 (50%)

- Data point to classify

Person	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

Example (from Wikipedia) - IV

Person	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033×10^{-2}	176.25	1.2292×10^{2}	11.25	9.1667×10^{-1}
female	5.4175	9.7225×10^{-2}	132.5	5.5833×10^{2}	7.5	1.6667

- Data point to classify

Person	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789,$$

$$p(\text{weight} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(130-\mu)^2}{2\sigma^2}\right) = 5.9881 \cdot 10^{-6}$$

$$p(\text{foot size} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(8-\mu)^2}{2\sigma^2}\right) = 1.3112 \cdot 10^{-3}$$

$$\text{posterior numerator (male)} = \text{their product} = 6.1984 \cdot 10^{-9}$$

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Example (from Wikipedia) - V

Person	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033×10^{-2}	176.25	1.2292×10^{2}	11.25	9.1667×10^{-1}
female	5.4175	9.7225×10^{-2}	132.5	5.5833×10^{2}	7.5	1.6667

- Data point to classify

Person	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

$$p(\text{height} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(6-\mu)^2}{2\sigma^2}\right) \approx 1.5789,$$

$$p(\text{weight} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(130-\mu)^2}{2\sigma^2}\right) = 5.9881 \cdot 10^{-6}$$

$$p(\text{foot size} \mid \text{male}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(8-\mu)^2}{2\sigma^2}\right) = 1.3112 \cdot 10^{-3}$$

$$\text{posterior numerator (male)} = \text{their product} = 6.1984 \cdot 10^{-9}$$

Example (from Wikipedia) - VI

- Data point to classify

Person	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

- The same goes for

$$p(\text{height} \mid \text{female}) = 2.23 \cdot 10^{-1}$$
$$p(\text{weight} \mid \text{female}) = 1.6789 \cdot 10^{-2}$$
$$p(\text{foot size} \mid \text{female}) = 2.8669 \cdot 10^{-1}$$
$$\text{posterior numerator (female)} = \text{their product} = 5.3778 \cdot 10^{-4}$$

- Overall, posterior(female) > posterior(male)
  - Therefore the data point is classified as FEMALE

Linear Discriminant Analysis - I

- Another Statistical Classifier
  - Based on Fisher Linear Discriminant
  - (Very briefly) Fisher Linear Discriminant projects data points to a vector which maximises "discriminant" capabilities

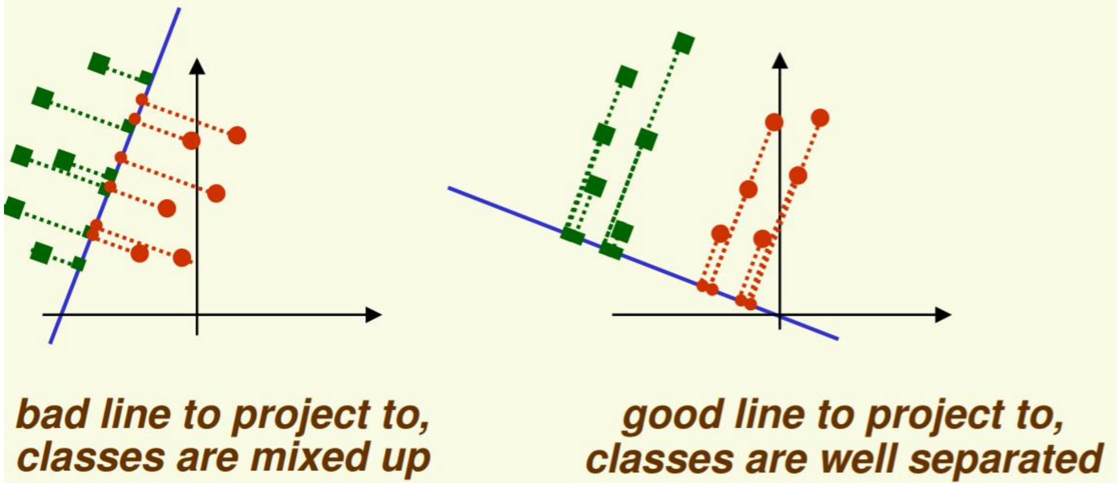


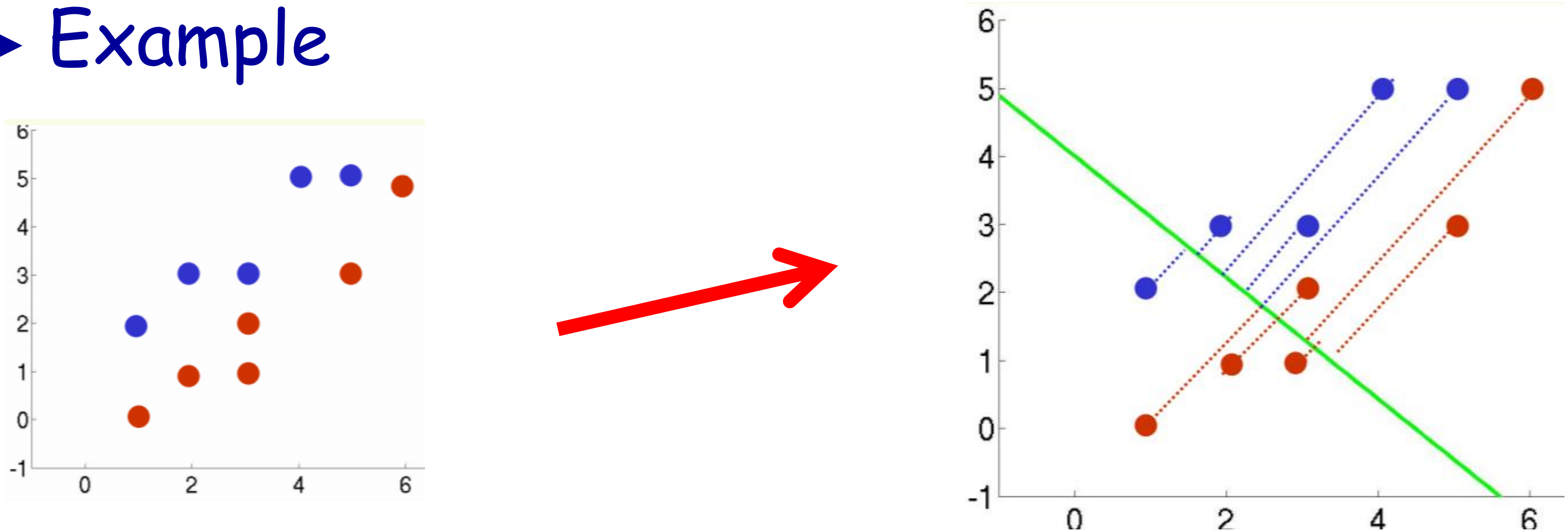
Image from https://www.csd.uwo.ca/~oveksler/Courses/CS434a_541a/Lecture8.pdf

Linear Discriminant Analysis - II

- Another Statistical Classifier
  - Based on Fisher Linear Discriminant
  - (Very briefly) Fisher Linear Discriminant projects data points to a vector which maximises "discriminant" capabilities
  - Once found, the vector is used as reference to calculate average /std of data points projected onto the vector, for each class (two classes in binary classification)
  - This is used to predict class for a new data point
    - Again, no need to go deeper in this course, the main idea is enough

Linear Discriminant Analysis - III

- Example

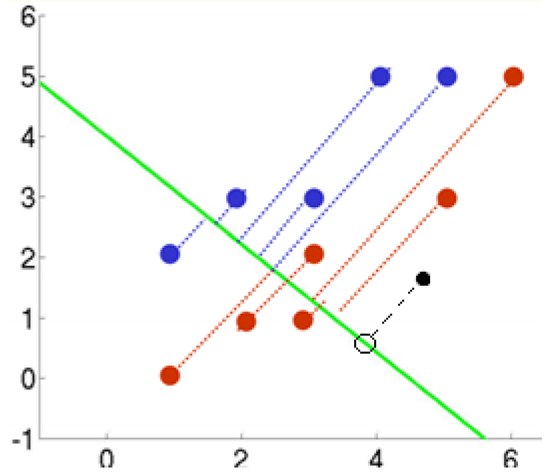


- Green line is the Fisher Discriminant
- Once found, the discriminant allows calculating average/std for blue dots and for red dots

Example from https://www.csd.uwo.ca/~oveksler/Courses/CS434a_541a/Lecture8.pdf

Linear Discriminant Analysis - IV

- Example (cont.)
  - Green line is the Fisher Discriminant
  - Once found, the discriminant allows calculating average/std for blue dots and for red dots
  - Intuitively
    - a new (black) data point will be projected to the green line and
    - we will understand if it is closer to blue or red dots
    - Closer to red -> red class



Logistic Regression - I

- Key observation here is that logistic regression is a statistical model that uses a logistic function to model a binary dependent variable
  - Slightly different from linear regression, which is usually used to predict real values rather than classes
  - In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors")
  - The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling

Logistic Regression - II

- Example from Wikipedia (slightly modified)
  - Problem: A group of 20 students spends between 0 and 6 hours studying for an exam: some of them succeeded, others did not. How does the number of hours spent studying affect the probability of the student passing the exam?
- Train Data

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

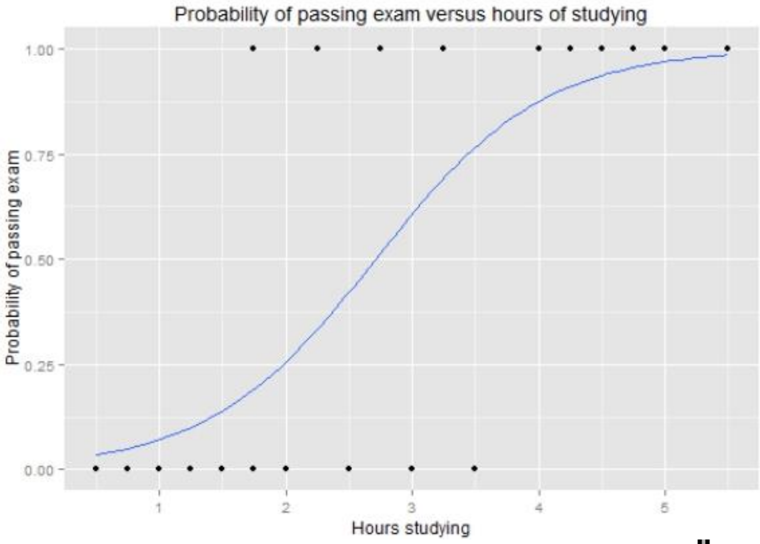
Logistic Regression - III

▶ Train Data

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

- The logistic regression analysis gives the following output
- Which traces the distribution on the right

	Coefficient	Std.Error	z-value	P-value (Wald)
Intercept	−4.0777	1.7610	−2.316	0.0206
Hours	1.5046	0.6287	2.393	0.0167



Logistic Regression - IV

- Now, such distribution follows the formula

$$\text{Probability of passing exam} = \frac{1}{1 + \exp(-(1.5046 \cdot \text{Hours} - 4.0777))}$$

	Coefficient	Std.Error	z-value	P-value (Wald)
Intercept	−4.0777	1.7610	−2.316	0.0206
Hours	1.5046	0.6287	2.393	0.0167

- Which is the one that we can use to predict new class labels
  - Hours = 5 → probability = 0.97, or rather class 1 (pass)
  - Hours = 2 → probability = 0.26, or rather class 0 (fail)
  - ….

Neural Networks

Neural Networks (I)

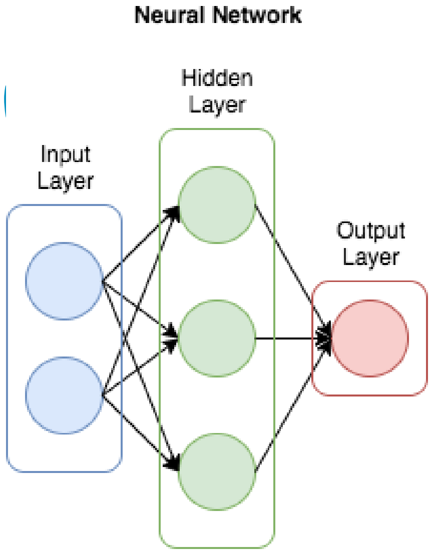
- (Artificial) Neural Networks (A)NNs are classifiers inspired by the biological neural networks that constitute animal brains.
  - A NN is based on a collection of
    - connected units or **nodes** called artificial neurons, which loosely model the neurons in a biological brain.
    - Each connection (**edge**), like the synapses in a biological brain, can transmit a signal to other neurons.

Neural Networks (II)

- (Artificial) Neural Networks (A)NNs are classifiers inspired by the biological neural networks that constitute animal brains.
  - Neurons and edges have a weight that adjusts as learning goes
  - Neurons are aggregated into layers.
    - Different layers may perform different transformations on their inputs.
    - Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

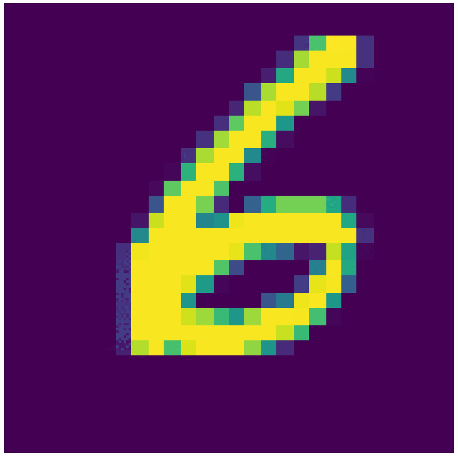
Neural Networks Explained

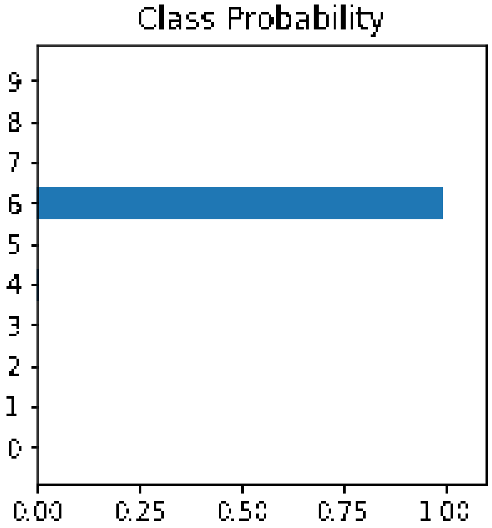
- The input layer provides the interface of the network
  - Input data is sent here
- The hidden layer allows executing non-linear combinations of inputs trough subsequent weighted sums
- Output layer(s) produce the result, us number
  - e.g., % of belonging to class B for binary classification
  - More than 2 classes -> more neurons in output layer



Neural Networks: Multiple Outputs

- In case of multiple classes, multiple neurons in the output layer are needed
  - For example, lets recognize numbers 0-9
  - Each neuron outputs a probability

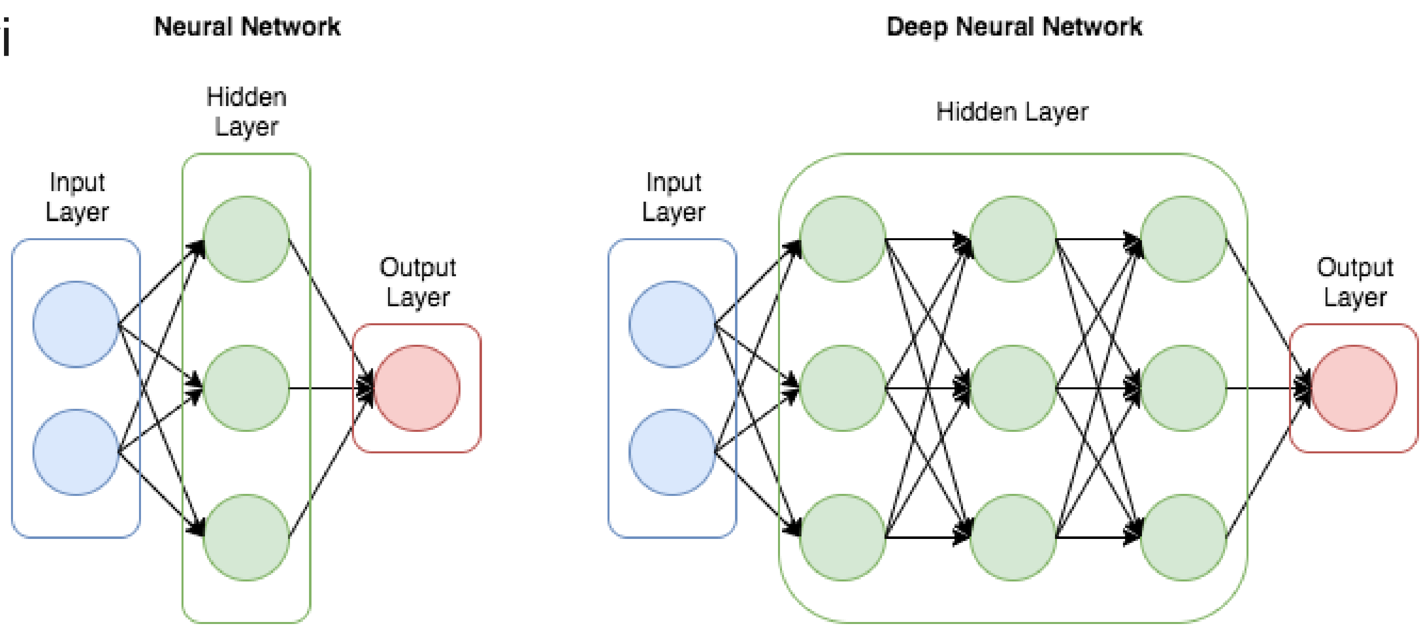




From: https://towardsdatascience.com/training-neural-network-from-scratch-using-pytorch-in-just-7-cells-e6e904070a1d

Deep Neural Networks

- Briefly, a deep NN has multiple hidden layers (more than 1)
  - Other more precise characterizations are currently used, but are too detailed for this part of the course
    - You wi



Neural Networks: Train Functions

- Training a Neural Network translates to assigning adequate weights to edges
  - Weights are initialized randomly
  - Subsequent training **epochs** aim at reducing the **loss**
    - Which is the difference of NN outputs with respect to ground truth
  - The impact each rain epoch has on weights is guided by learning rate
    - The higher the rate, the bigger the potential change of weights
  - Different **train functions** obey to different rules or heuristics to minimize loss by changing weights of edges involving hidden layers

MultiLayer Perceptron

- A multilayer perceptron (MLP) is a class of artificial neural network
  - A MLP consists of at least three layers of nodes:
    - an input layer,
    - a hidden layer and
    - an output layer
  - Except for the input nodes, each node is a neuron that uses a nonlinear activation function.
    - non-linear activation distinguish MLP from a linear perceptron
  - MLP relies on backpropagation for training
    - backpropagation computes the gradient of the loss function
    - Baseline for reinforcement learning

MLP: Backpropagation

- MLP relies on backpropagation for training
  - backpropagation computes the gradient of the loss function
  - Baseline for reinforcement learning