Hi! PARIS

HEC PARIS

INSTITUT POLYTECHNIQUE DE PARIS

PARIS ARTIFICIAL INTELLIGENCE FOR SOCIETY

Machine Learning I & II

Hi! PARIS DataBootcamp 2025

HEC PARIS

INSTITUT POLYTECHNIQUE DE PARIS

Inria

Intermediate Track

Agenda of the course

Machine Learning I (9:00AM-10:30PM)

- Introduction to Machine Learning
- Data Preprocessing (+ demo)

Machine Learning II (10:45PM-12:15PM)

- Model Training (+ demo)
- Model Evaluation (+ demo)
- Improve the performance of your model (+ demo) ➔ if time

Introduction to Machine Learning

What is Machine Learning (ML) ?

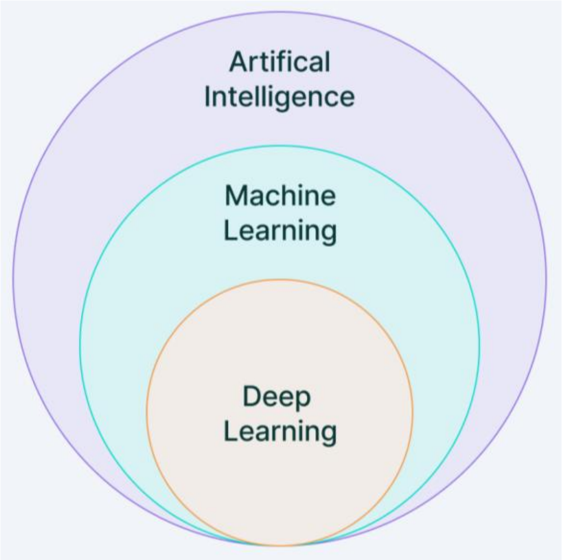
"Machine Learning is the field of study that gives computers the ability to learn without explicitly being programmed."

Arthur Samuel, Pioneer of AI in the 1950's

What is Machine Learning (ML) ?

"Machine Learning is the field of study that gives computers the ability to learn without explicitly being programmed."

Arthur Samuel, Pioneer of AI in the 1950's



- In Machine Learning, algorithms make **autonomious decisions**
- Contains **Deep Learning**, a subfield focused on neural networks

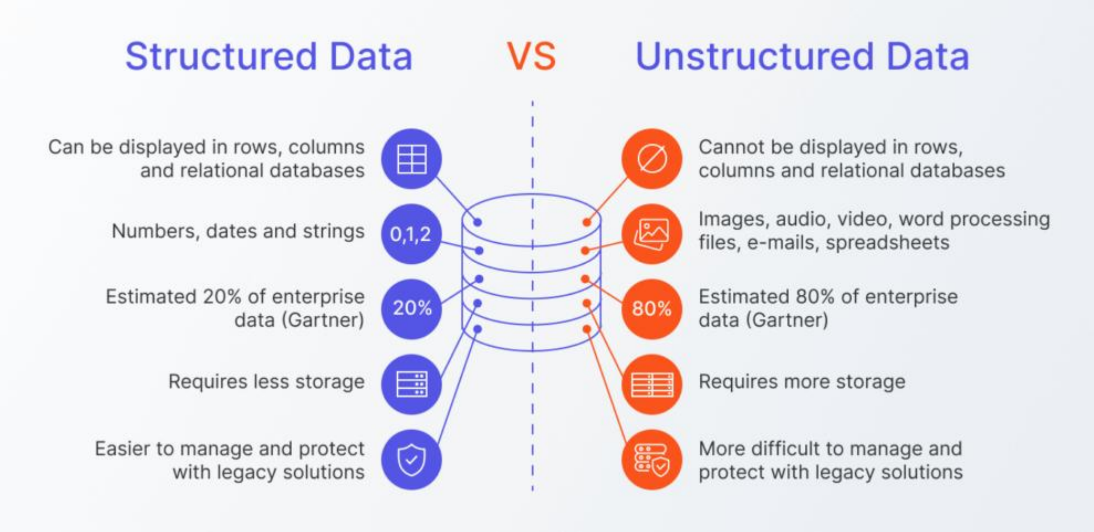
What is Machine Learning (ML) ?

Algorithms learn how to make decisions by **analyzing large amounts of data**, that can be stored in different formats

What is Machine Learning (ML) ?

Algorithms learn how to make decisions by **analyzing large amounts of data**, that can be stored in different formats

Machine Learning ➔ structured data



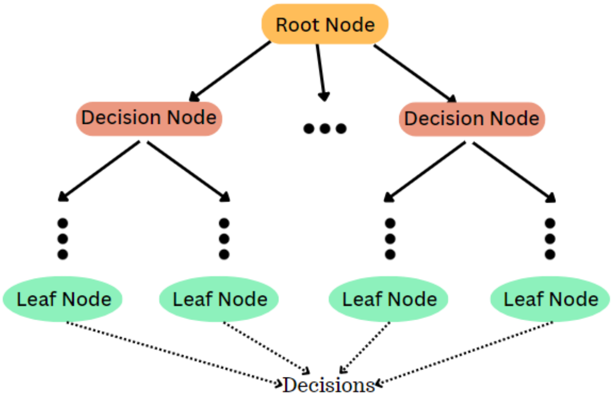
Deep Learning ➔ unstructured data

What is Machine Learning (ML) ?

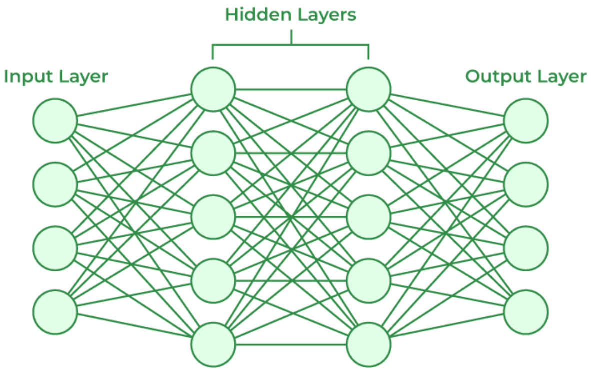
Model training: A model is given many examples and improves its results by adjusting its parameters

What is Machine Learning (ML) ?

Model training: A model is given many examples and improves its results by adjusting its parameters



Decision tree



Neural networks

Feature vs Target variables

Feature variables: Attributes of the data used to make predictions.

- *Shouldn't include the desired output (target).*

Feature vs Target variables

Feature variables: Attributes of the data used to make predictions.

- *Shouldn't include the desired output (target).*

Target variable: Output variable that the model aims to predict.

- *Data is considered "labeled" when the target is available.*
- *Data is "unlabeled" when it isn't.*

Let’s give an example

Build a Machine Learning model that can **predict whether a passenger survived (or not) the titanic**

The dataset is **structured**

Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000

Let’s give an example

Feature variables

Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000

Target

The target variable is available ➔ **Labeled dataset**

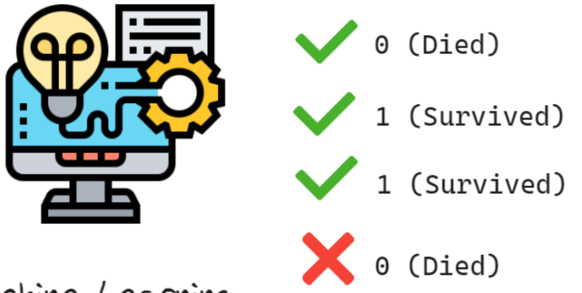
Let’s give an example

Target

Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000

Training data

Model training



Machine Learning model

Let’s give an example

Step 1: Training

Target

Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000

Training data

Model training

Machine Learning model

0 (Died)
1 (Survived)
1 (Survived)
0 (Died)

Step 2: Final prediction

New passenger

Survived: ?
Pclass: 1
Age: 29
Sex: Male
...

Trained model

0 (Died)

1 (Survived)

Continuous vs Categorical data

Continuous data: Numerical data that can take any value within a defined range

- Examples: *Weight, price, temperature*

Categorical data: Data with a finite number of possible values/categories

- Categories are ordinal or nominal
- Examples: *Yes/No, 0/1*,….

Continuous vs Categorical data

	Gender	Height	Weight	Index	Status
0	Male	174	96	4	Obesity
1	Male	189	87	2	Normal
2	Female	185	110	4	Obesity
3	Female	195	104	3	Overweight
4	Male	149	61	3	Overweight
5	Male	189	104	3	Overweight
6	Male	147	92	5	Extreme Obesity
7	Male	154	111	5	Extreme Obesity
8	Male	174	90	3	Overweight
9	Female	169	103	4	Obesity

Height and *Weight* are **continuous**

Gender, Index and Status are **categorical**

Main types of Machine Learning

Machine Learning

Supervised Learning

Train a model with supervision/examples (labeled data)

- Classification
- Regression

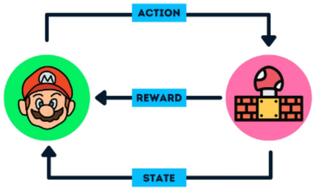
Unsupervised Learning

Discover patterns without prior knowledge (unlabeled data)

- Clustering
- Dimension reduction

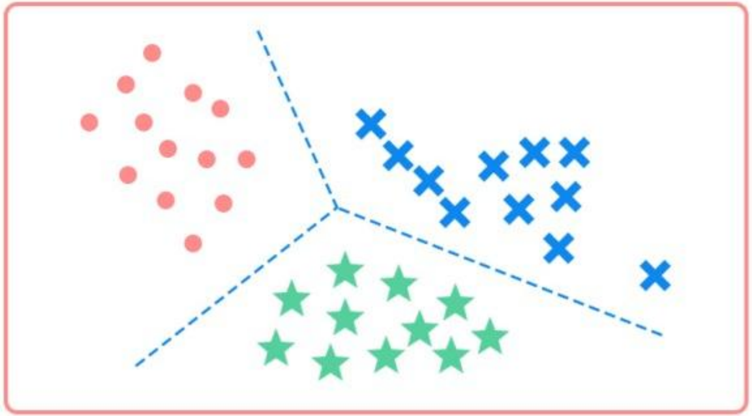
Reinforcement Learning

Teach an agent to interact in an environment by giving rewards and penalties



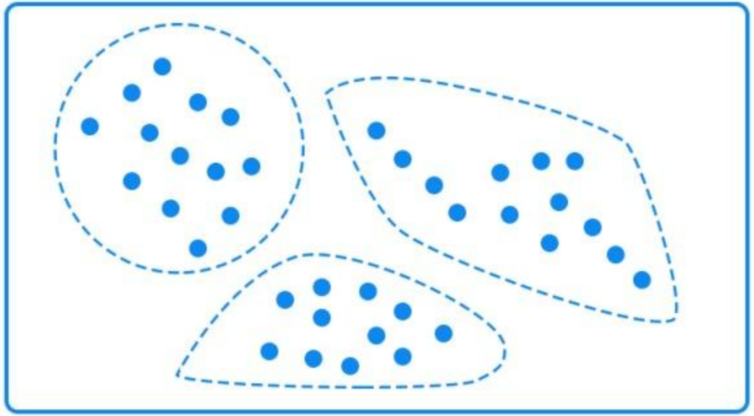
Main types of Machine Learning

Supervised Learning



- Knowledge of the values to predict
- Model learns using **labeled data** (examples)

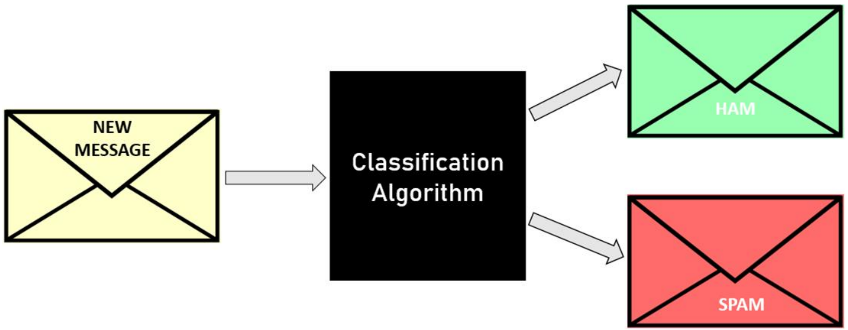
Unsupervised Learning



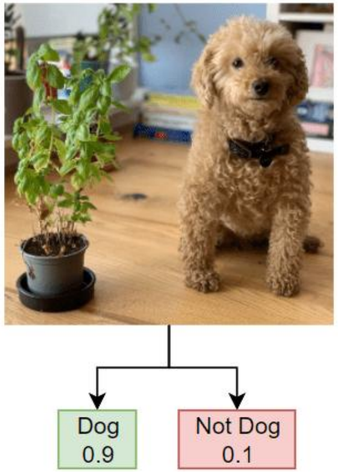
- No prior knowledge on what to predict
- Let the model discover groupings / patterns on its own
- **unlabeled data**

Supervised Learning

1. <u>**Classification:**</u> Task of predicting a **categorical target**



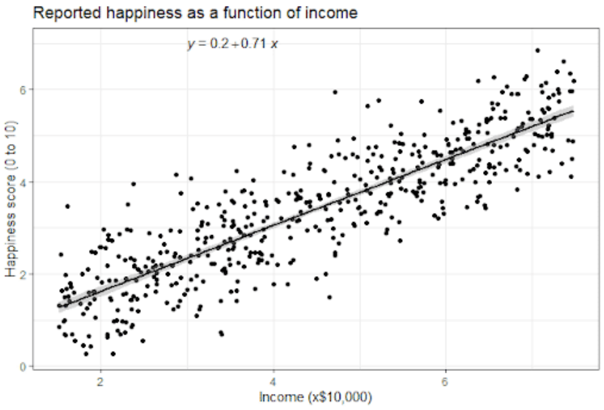
Spam classification (Spam, ham)



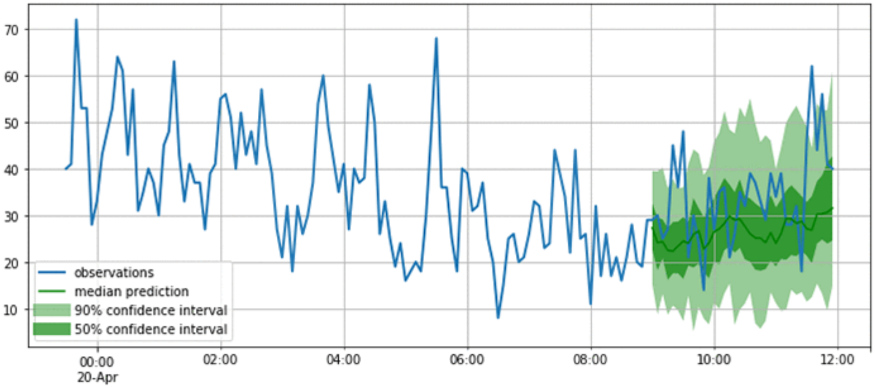
Object classification (Dog, not dog)

Supervised Learning

2. <u>Regression:</u> Task of predicting a **continuous target**



Income prediction
(based on a happiness score)



Weather forecasting
(based on historical data)

Supervised Learning

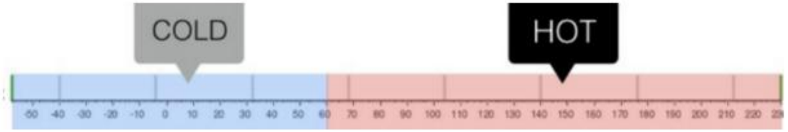
Regression

What will be the temperature tomorrow?

Fahrenheit

Classification

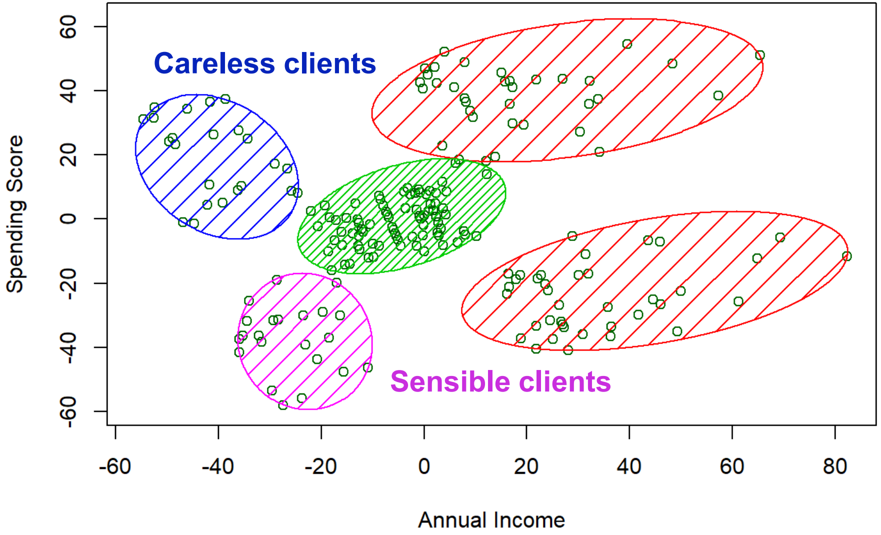
Will it be hot or cold tomorrow?



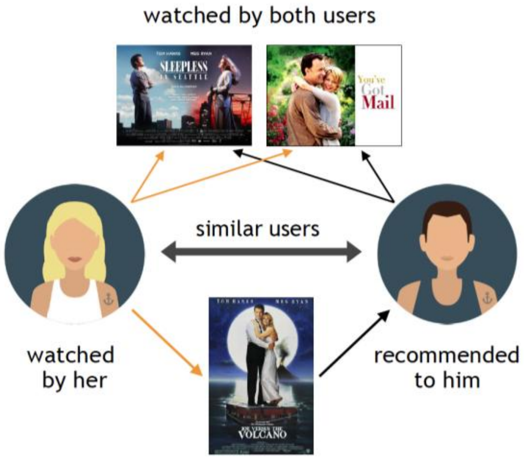
Fahrenheit

Unsupervised Learning

1. **Clustering:** Group data together based on distances/similarities



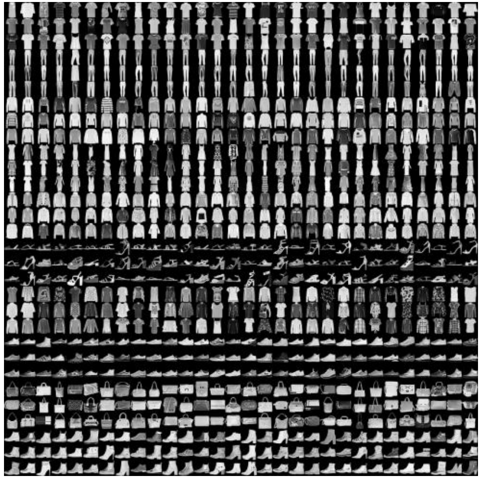
Client segmentation



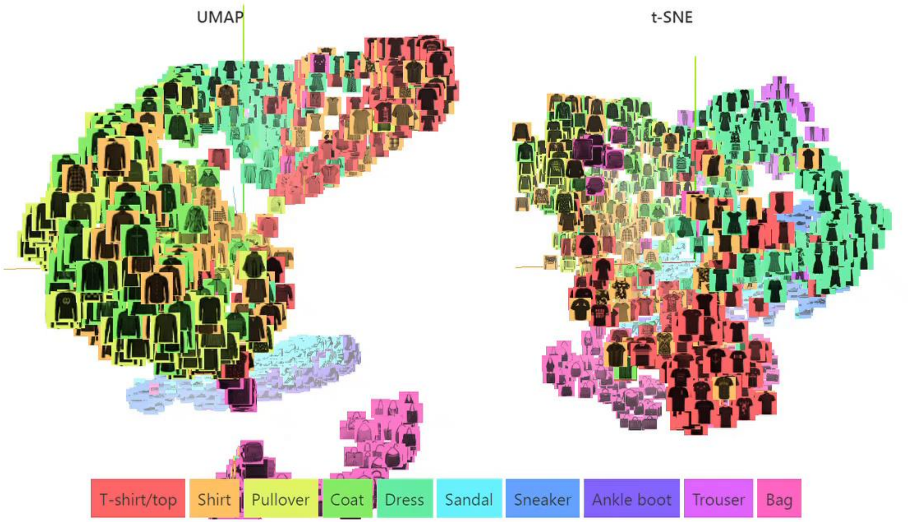
Recommend movies to similar users

Unsupervised Learning

2. **Dimension reduction**: Represent data in a lower number of features to simplify visualization and interpretability

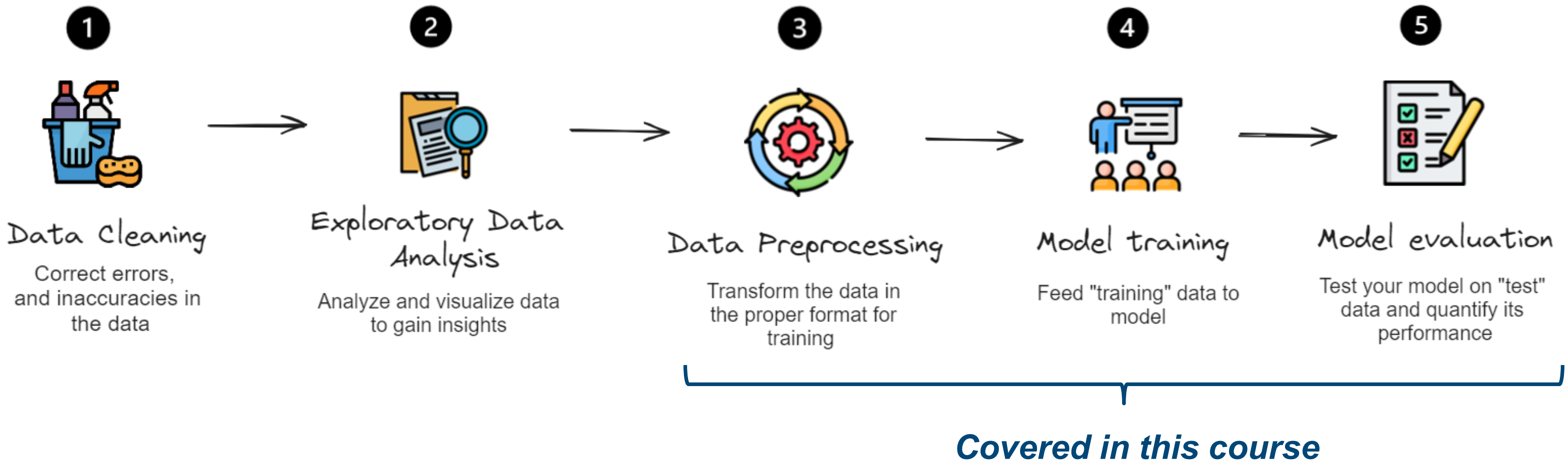


Fashion MNIST Dataset



Machine Learning in practice

In practice, Machine Learning is more then building algorithms. Models need **clean and properly formated data** to generate good results.



Data Preprocessing

What is Data Preprocessing

Data Preprocessing is the process of transforming data it into a clean and suitable format for training.

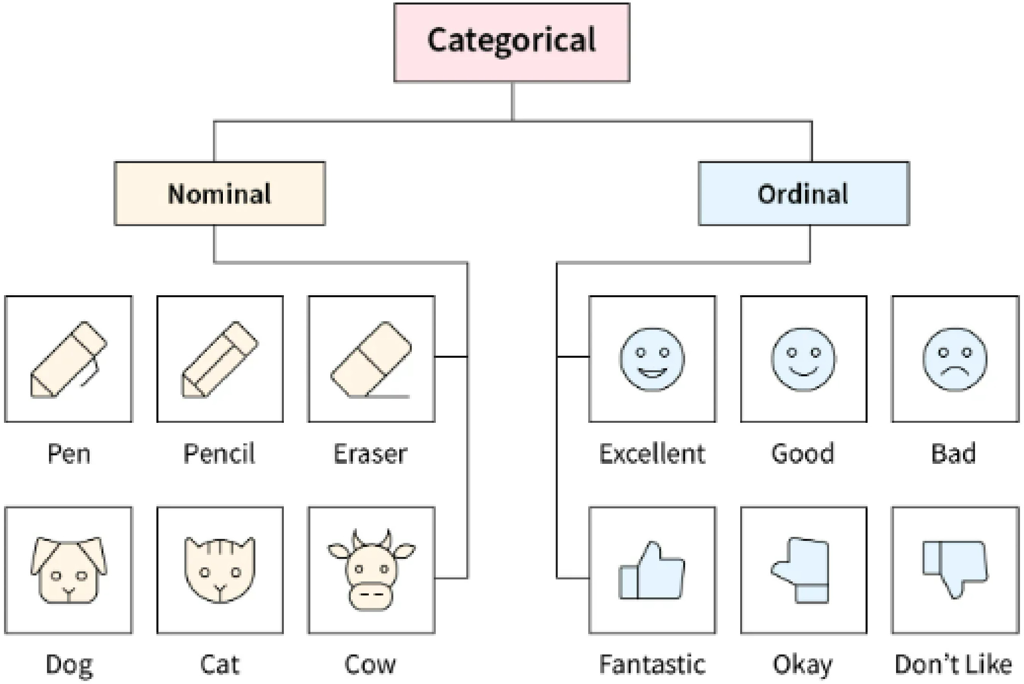
Examples

- **Categorical Encoding** (for categorical variables)
- **Feature Scaling** (for continuous variables)
- **Feature Selection** (not covered today)

Ordinal vs Nominal variables

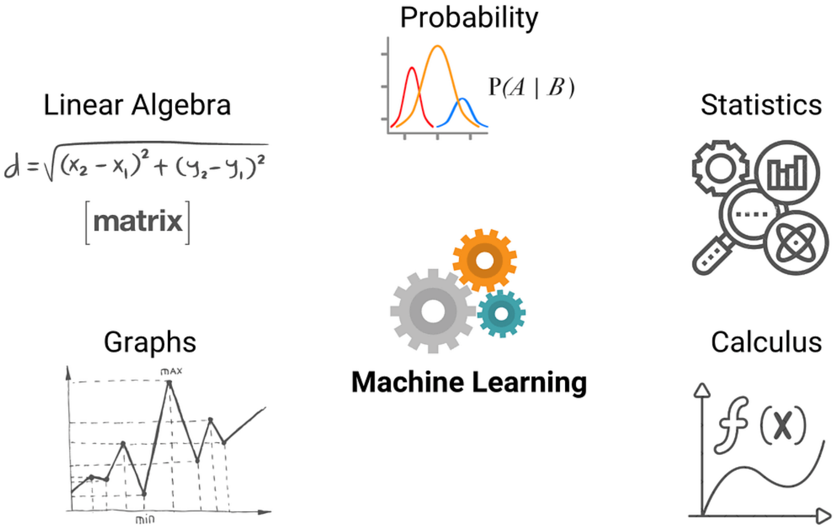
- **Ordinal**: Categorical variable whose categories have an inherent order/hierarchy
- **Nominal (unordered)**: Categorical variable whose categories have have no hierarchy

Ordinal vs Nominal variables



Categorical encoding

Categorical encoding means **converting categorical variables into numerical values** that can be understood and processed by a model.



Math behind Machine Learning models

Categorical encoding

1 Label Encoding

Replace categories by numerical values (0,1,2,…)

Each class gets attributed a numerical value within the same column

Grades
A
B
C
D
Fail

Grades	Encoded
A	4
B	3
C	2
D	1
Fail	0

Grades is an ordinal categorical variables

Categorical encoding

2 One-Hot Encoding

Create a new binary variable (0 or 1) for each class

Each class has its own column with 0 or 1

Places
New York
Boston
Chicago
California
New Jersey

New York	Boston	Chicago	California	New Jersey
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Places is a nominal categorical variables

The variable is 1 if the place is New York, else 0

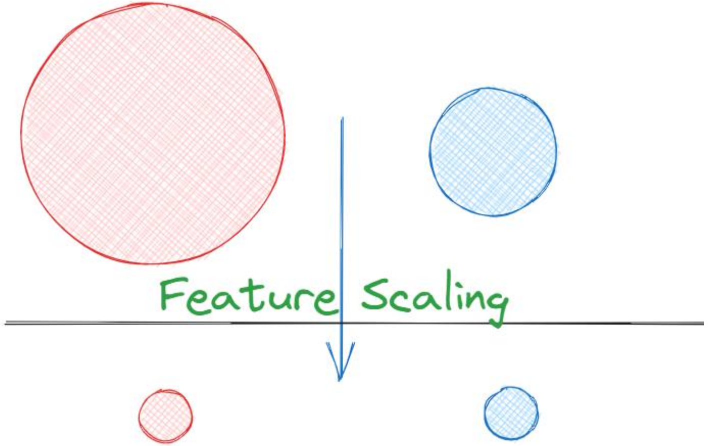
Categorical encoding

Important notions:

- Label Encoding should only be used on **ordinal feature variables**
- One-Hot Encoding can lead to a **huge increase in the number of columns** when the number of classes is large.

Feature scaling

Feature scaling means **scaling numerical variables to a similar range of values**



Why is it useful ?

- Improve a model's performance
- Decrease training time
- Not useful for every model (usually distance-based ones)

Feature scaling

1 Normalization (MinMax Scaling)

Scale values to a specific range, usually between 0 and 1

Age	Normalized Age
44	0.80952381
27	0
30	0.142857143
38	0.523809524
40	0.619047619

Salary	Normalized Salary
73000	0.838709677
47000	0
53000	0.193548387
62000	0.483870968
57000	0.322580645

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The normalized values of **Age** and **Salary** are between 0 and 1

Feature scaling

2 **Standardisation** (Standard Scaler)

Scale a variable to a normal distribution, with a 0 mean and 1 standard deviation

input	standardized
0	-1.46385
1	-0.87831
2	-0.29277
3	0.29277
4	0.87831
5	1.46385

$$x_{new} = \frac{x - \mu}{\sigma}$$

The standardized input
follows a normal distribution

Normalization vs Standardisation

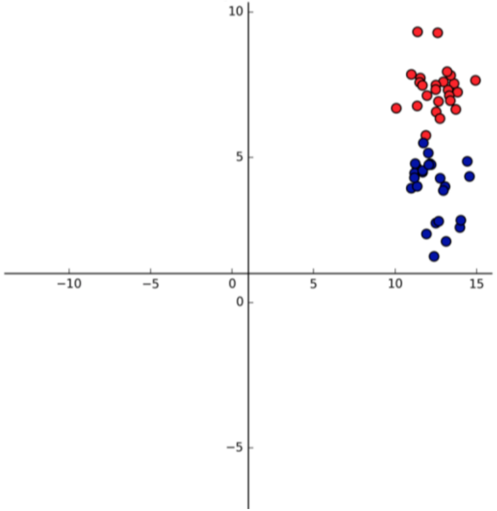
1. **Normalization**
   - Better for variables without a normal distribution
   - More sensitive to outliers

2. **Standardisation**
   - Better for variables that have close to a normal distribution
   - Less sensitive to outliers

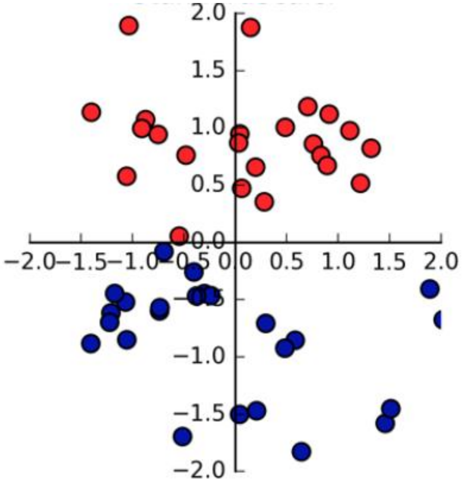
The best way to choose is to try both and compare results

Normalization vs Standardisation

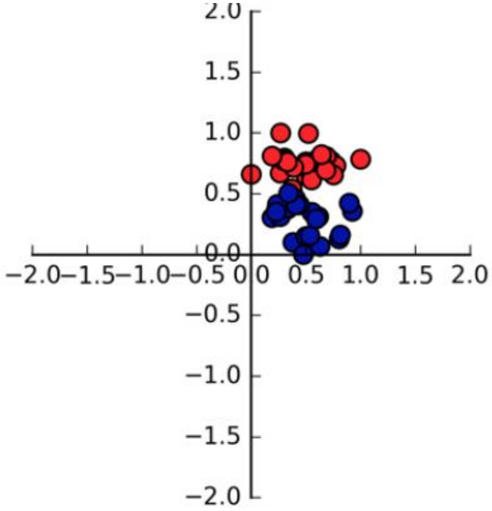
No scaling



Standardisation

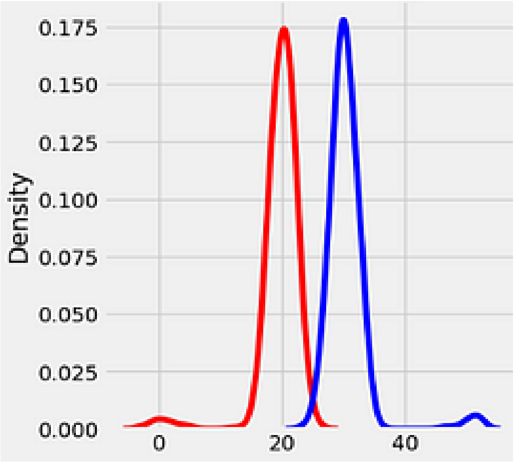


Normalization

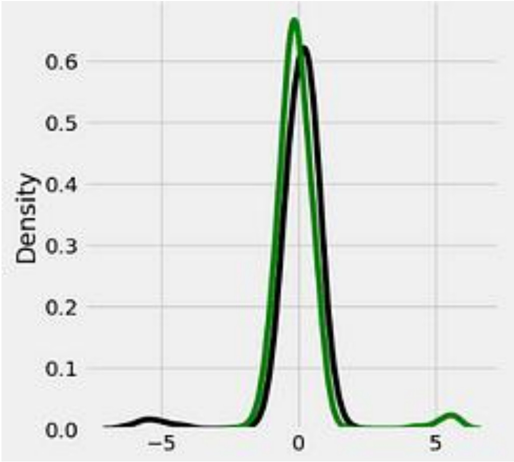


Normalization vs Standardisation

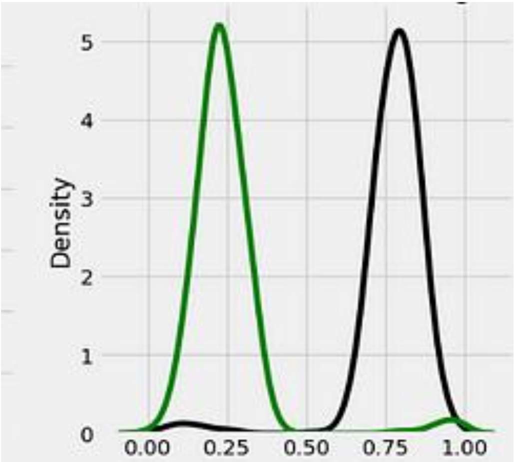
No scaling



Standardisation

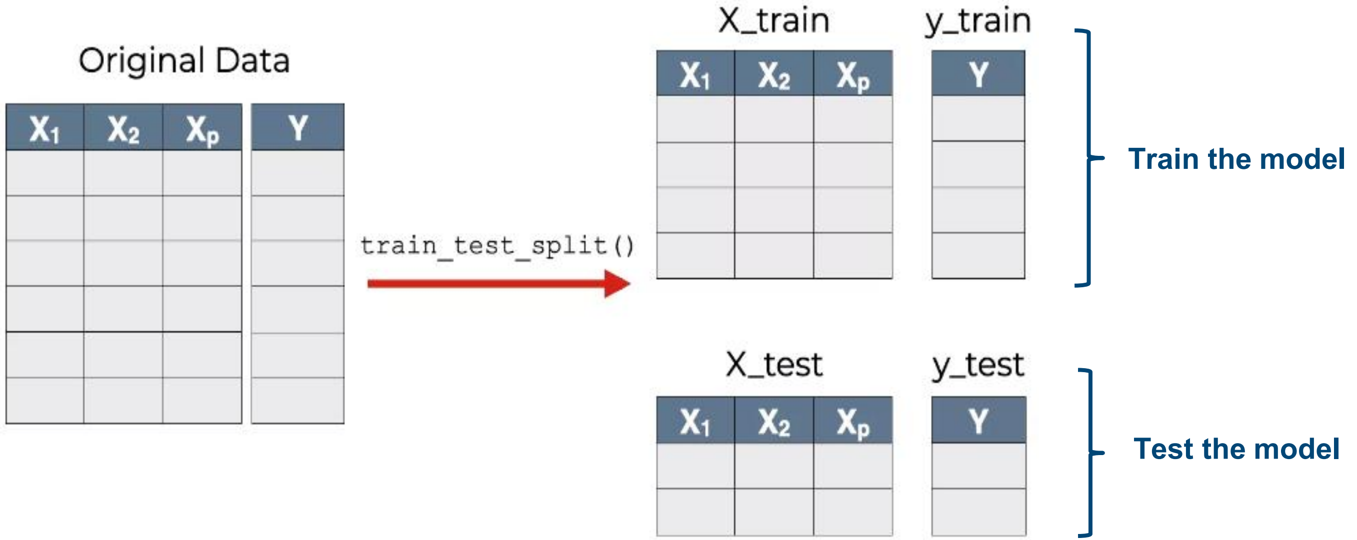


Normalization



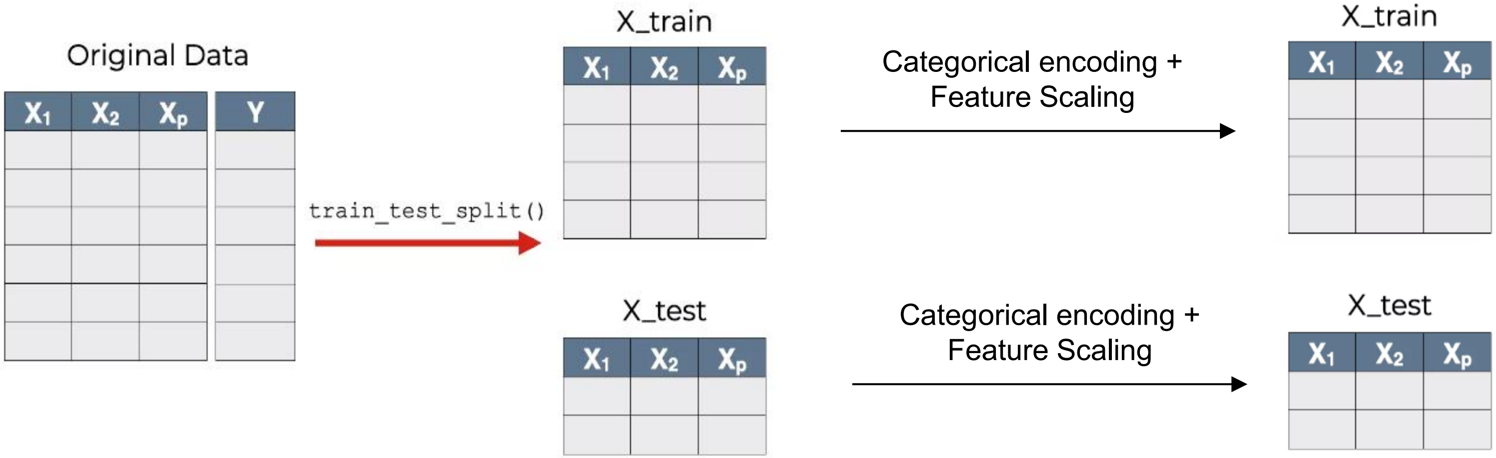
Training vs test data

A dataset is split into a **training** and a **test set** to evaluate a model on new/unseen data



Training vs test data

Data preprocessing **should be applied separately to each set** to prevent information in the training data to "leak" to the test data



Data preprocessing is only applied to the feature variables (X) !

Data Preprocessing with Scikit-learn

Demo of how to use scikit-learn for data preprocessing

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

Model training

Supervised Learning models

Most supervised learning models have **variants for Regression and Classification** (but not all).

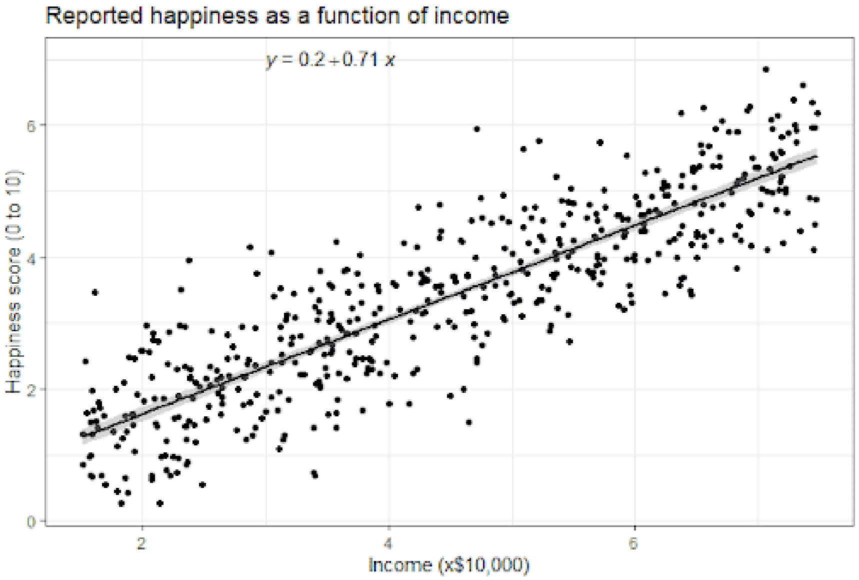
Models	Classification	Regression
Linear Regression	No	Yes
Logistic Regression	Yes	No
K nearest neighbors	Yes	Yes
Decision Trees	Yes	Yes
Random Forest	Yes	Yes
Boosting	Yes	Yes

Linear Regression

Find the best **linear equation** (or line) that describes the relationship between the features and target variable

Linear Regression

Find the best **linear equation** (or line) that describes the relationship between the features and target variable

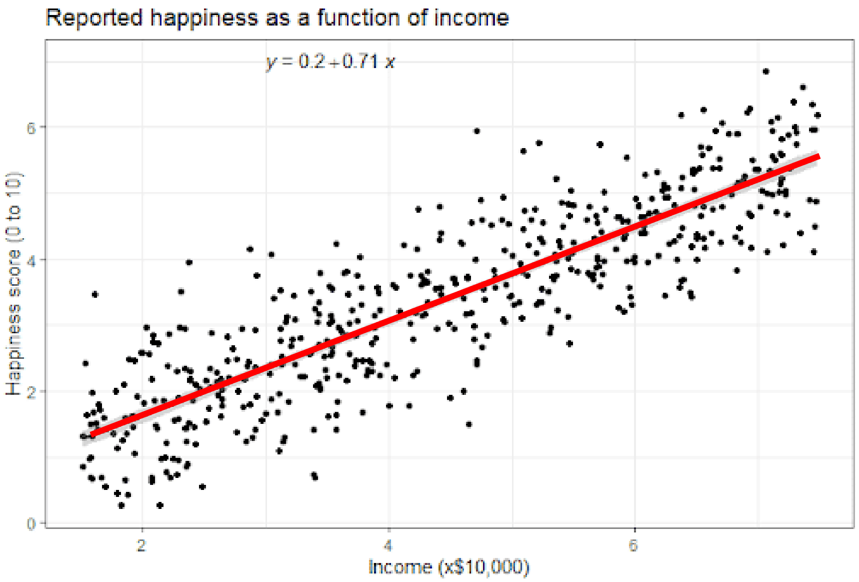


Feature (X): Income

Target (Y): Happiness score (0 to 10)

Linear Regression

Find the best **linear equation** (or line) that describes the relationship between the features and target variable

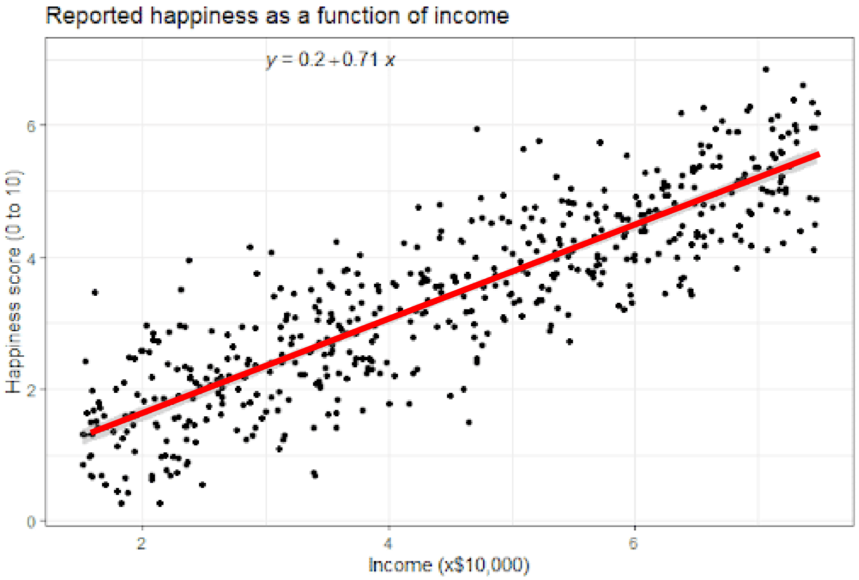


Linear equation

Y = **a** + X * **b**

Linear Regression

Find the best **linear equation** (or line) that describes the relationship between the features and target variable

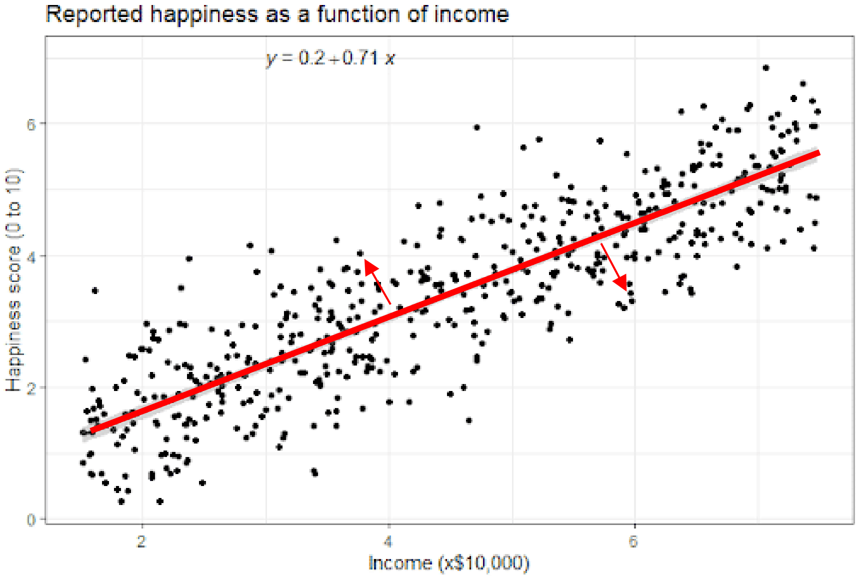


Linear equation

Happiness Score = **a** + *Income* * **b**

Linear Regression

Find the best **linear equation** (or line) that describes the relationship between the features and target variable



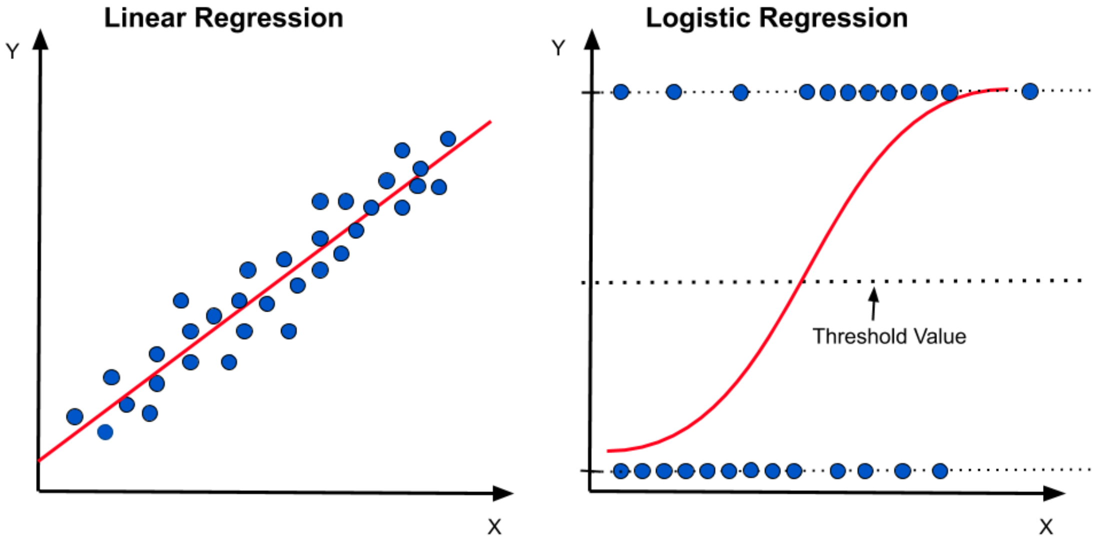
Find optimal parameters by **minimizing distances** with the estimated line

Logistic Regression

Estimate the probability that an **observation belongs to two possible classes** (binary classification)

Logistic Regression

Estimate the probability that an **observation belongs to two possible classes**



Logistic Regression

Logistic Regression is an **adaptation of Linear Regression** for binary classification tasks

$$\underbrace{Y = a + X * b}_{\textit{Linear regression}} \quad \oplus \quad \underbrace{\sigma(z) = \frac{1}{1 + \exp(-z)}}_{\textit{Sigmoid function}}$$

Logistic Regression

Logistic Regression is an **adaptation of Linear Regression** for binary classification tasks

$$\underbrace{Y = a + X * b}_{\text{Linear regression}} \quad + \quad \underbrace{\sigma(z) = \frac{1}{1 + \exp(-z)}}_{\text{Sigmoid function}} \quad = \quad \underbrace{P(Y = 0)}_{\text{Probability of the target being 0}}$$

Linear regression

Sigmoid function

Probability of the target being 0

Logistic Regression

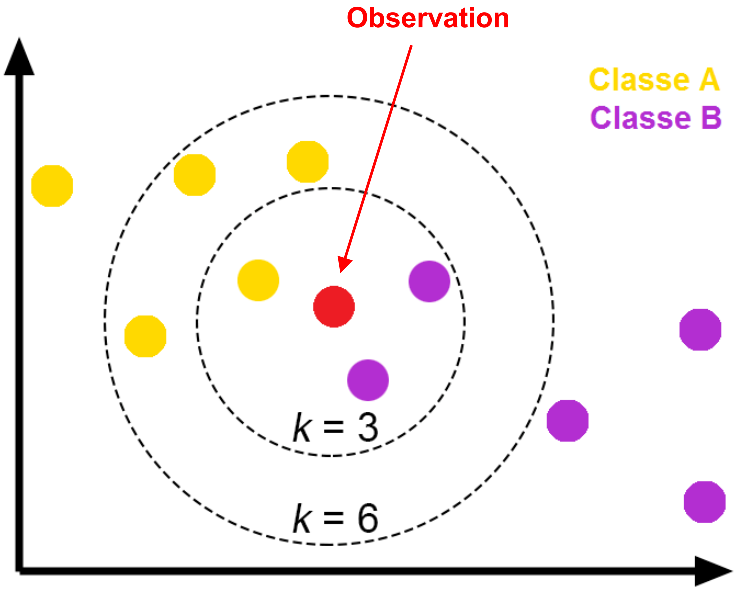
Final predictions are made by setting a threshold value (usually 0.5)

$P(Y = 0) \geq$ **0.5** ➡ ***The model predicts 0***

$P(Y = 0) <$ **0.5** ➡ ***The model predicts 1***

K nearest neighbors (KNN)

Predict the output of an observation using **the data points that are closest to it** (in its **neighborhood**)

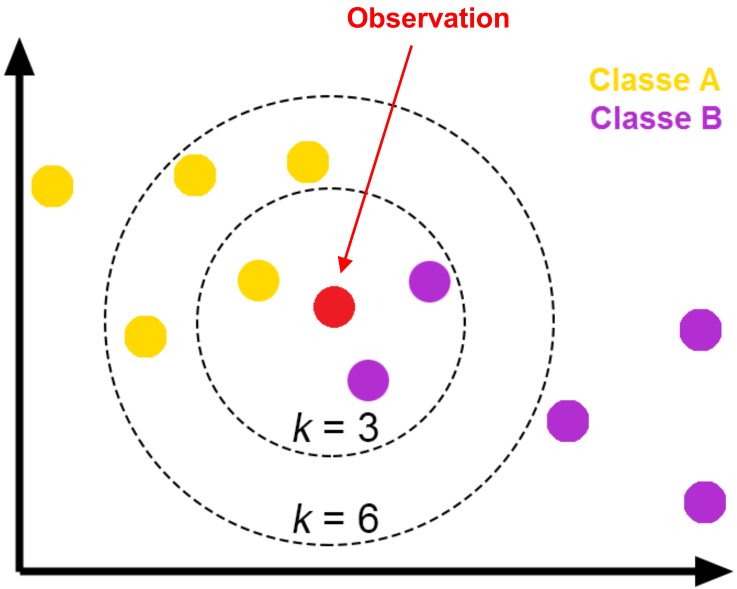


K nearest neighbors (KNN)

Predict the output of an observation using **the data points that are closest to it** (in its **neighborhood**)

k indicates how many points belong to a neighborhood

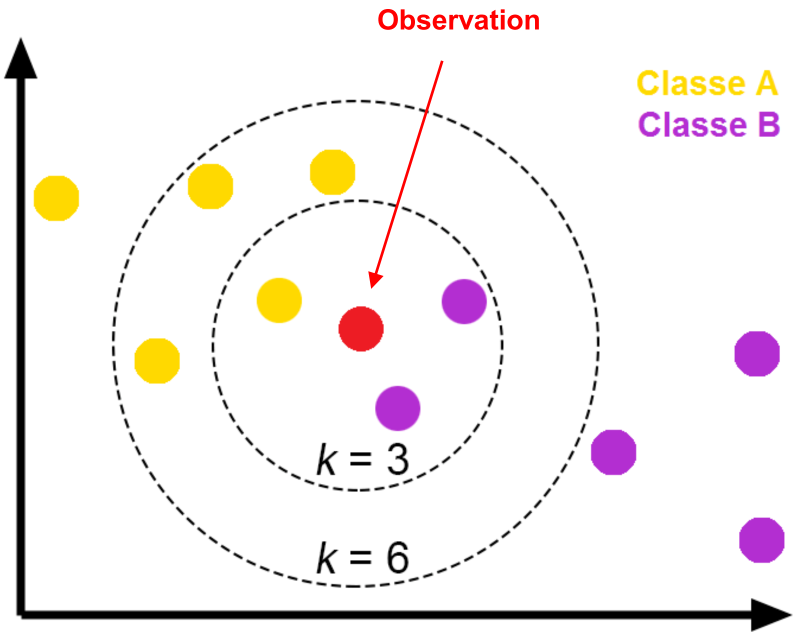
➔ “hyperparameter” that must be chosen



K nearest neighbors (KNN)

Final prediction:

- **Regression**: Compute the mean value of neighbor points
- **Classification**: Assign the most commonly found class in the neighborhood



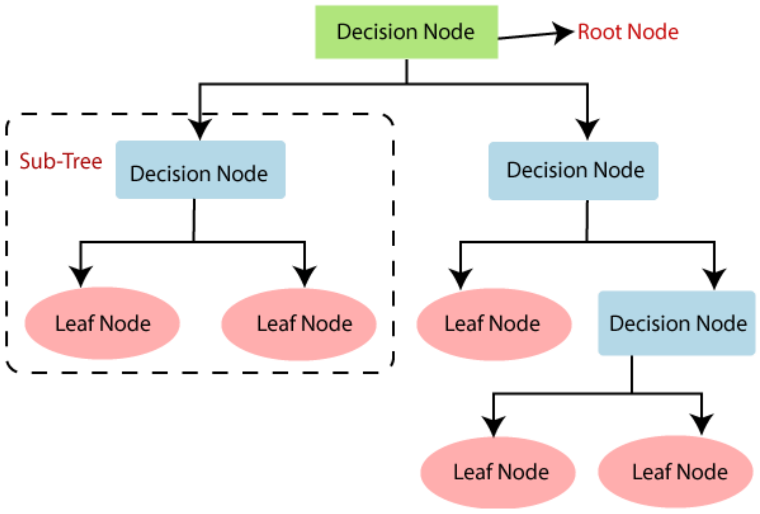
Decision trees

Build a decision tree that makes predictions by **splitting data into homogenous groups** (group points with similar caracteristics)

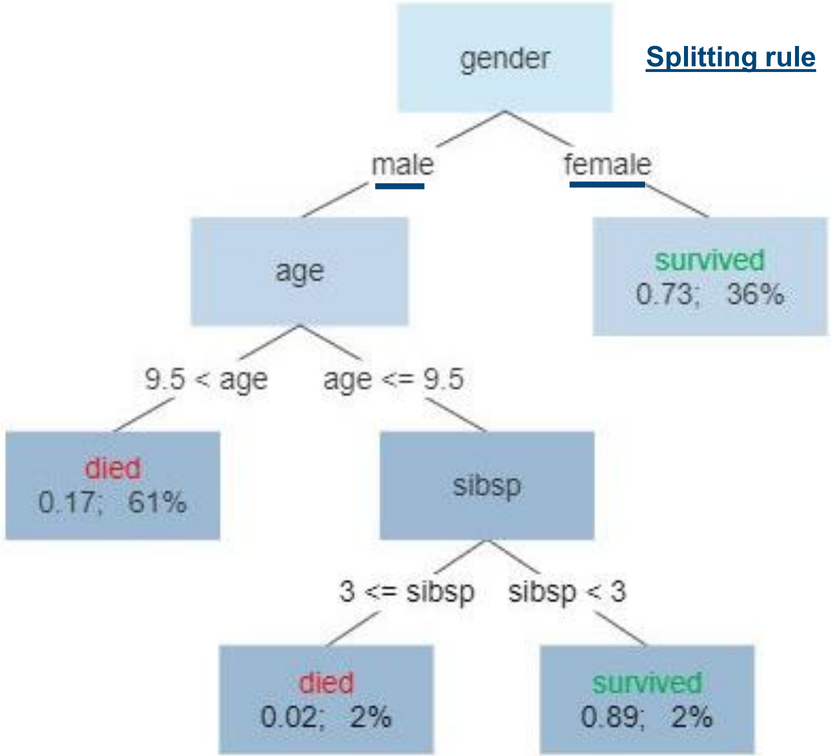
Decision trees

Build a decision tree that makes predictions by **splitting data into homogenous groups** (group points with similar caracteristics)

➔ The model splits data using **binary decision rules**



Decision trees



Splitting rule

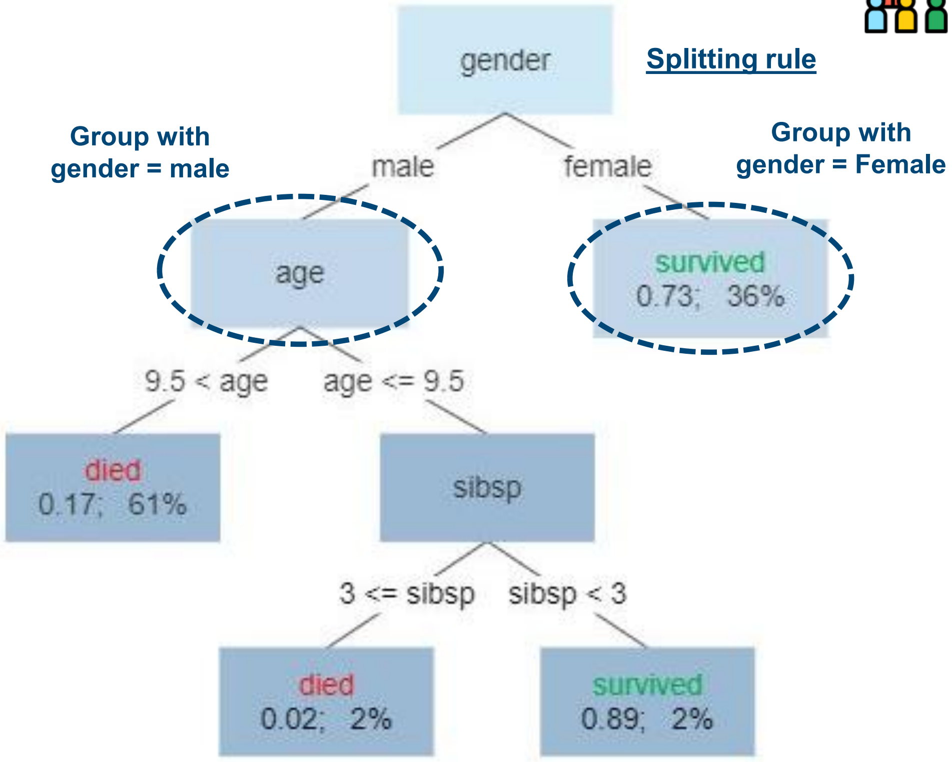
Example of a decision rule:

What gender is the passenger ?

Decision trees

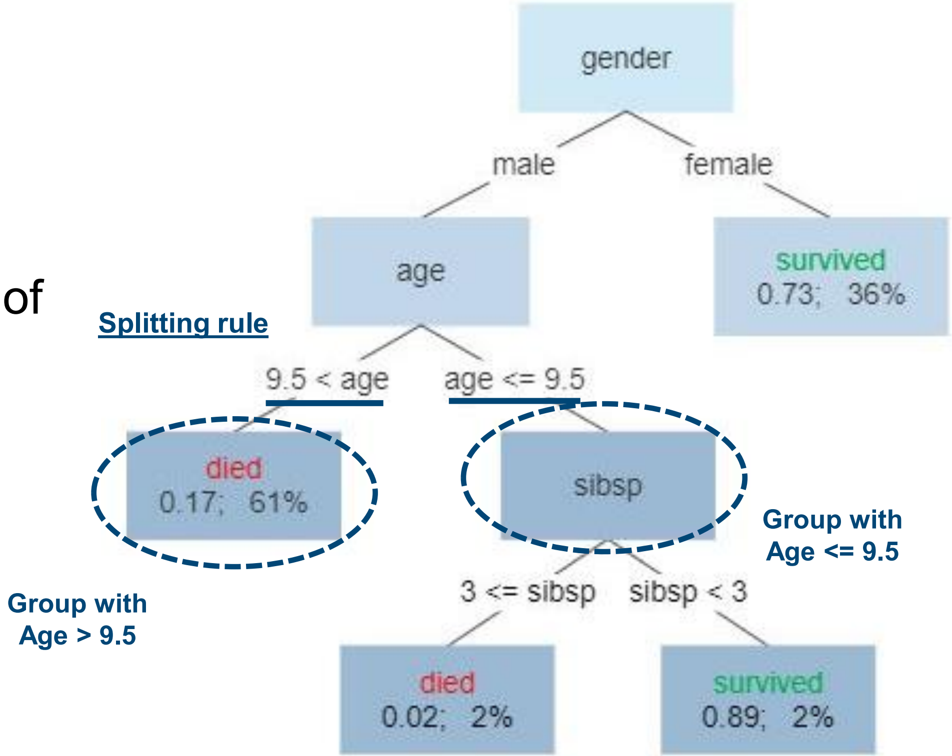
<u>Example of a decision rule:</u>

- <u>Gender = Male:</u> left split
- <u>Gender = Female</u>: right split



Decision trees

Continue till you reach the end of the tree (leaf node)



Decision trees

How does the tree stop splitting ?

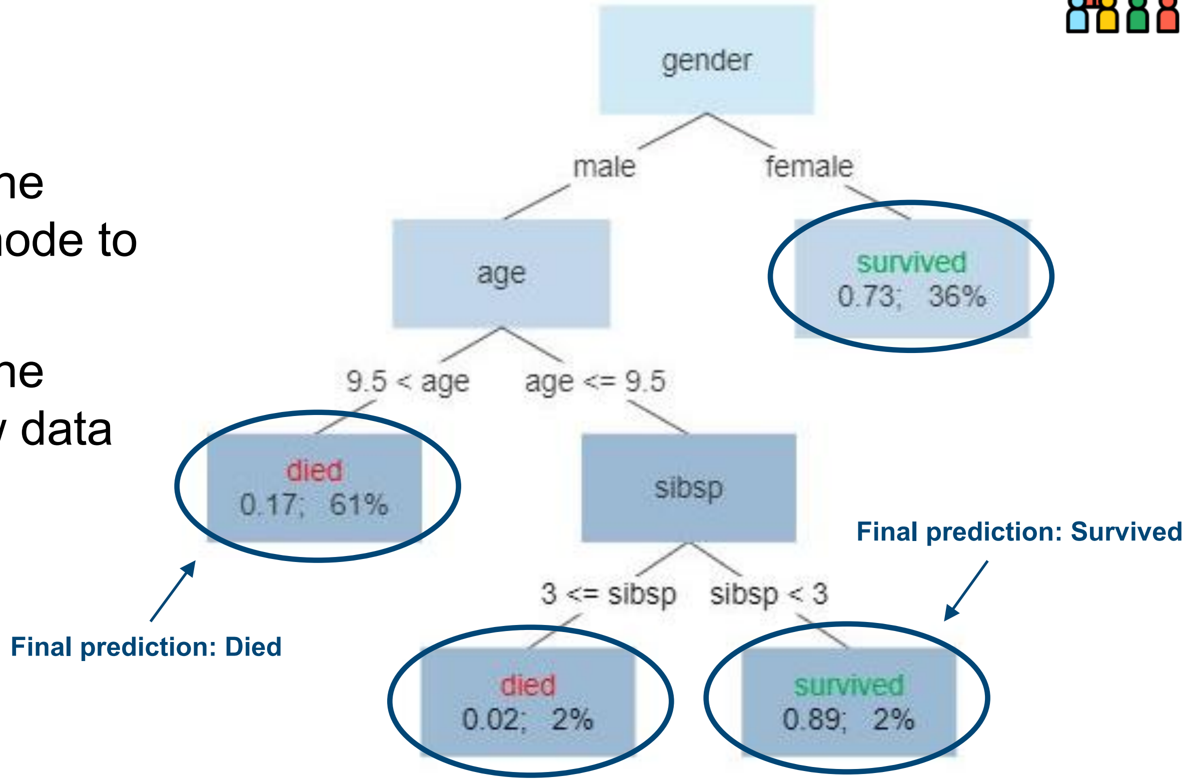
- Set a **max tree depth**
- Set a **miniumum number of points** in a leaf/final node

➔ "hyperparameters" that must be chosen

Decision trees

Final prediction

- **Regression**: Compute the average value in a leaf node to make a prediction
- **Classification**: Assign the majority class to the new data point



Random Forest

Build multiple <u>decorrelated trees</u> than aggregate final results to make a prediction on a new data point

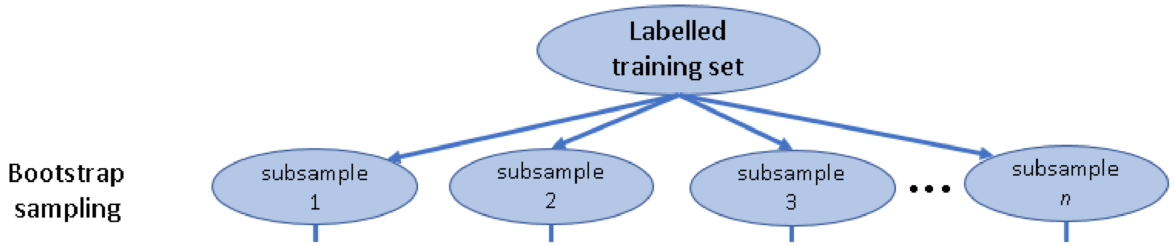
Random Forest

Build multiple <u>decorrelated trees</u> than aggregate final results to make a prediction on a new data point

➔ Prevent overfitting with regular Decision Trees

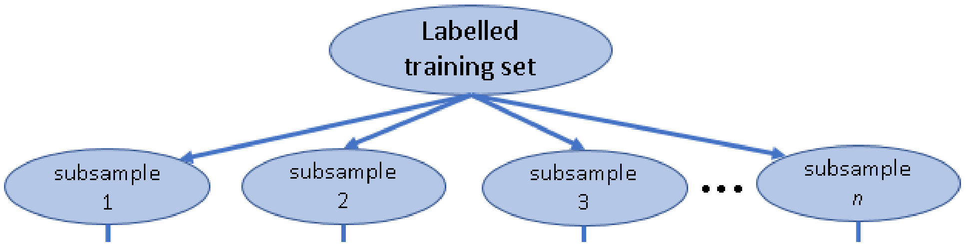
Random Forest

Step 1: Generate bootstrap samples →

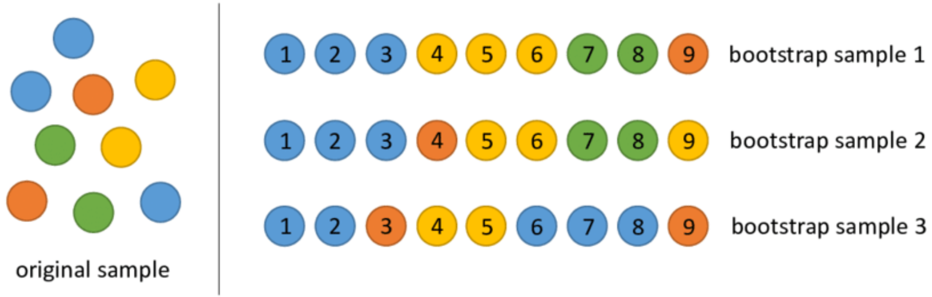


Random Forest

Step 1: Generate bootstrap samples → Bootstrap sampling

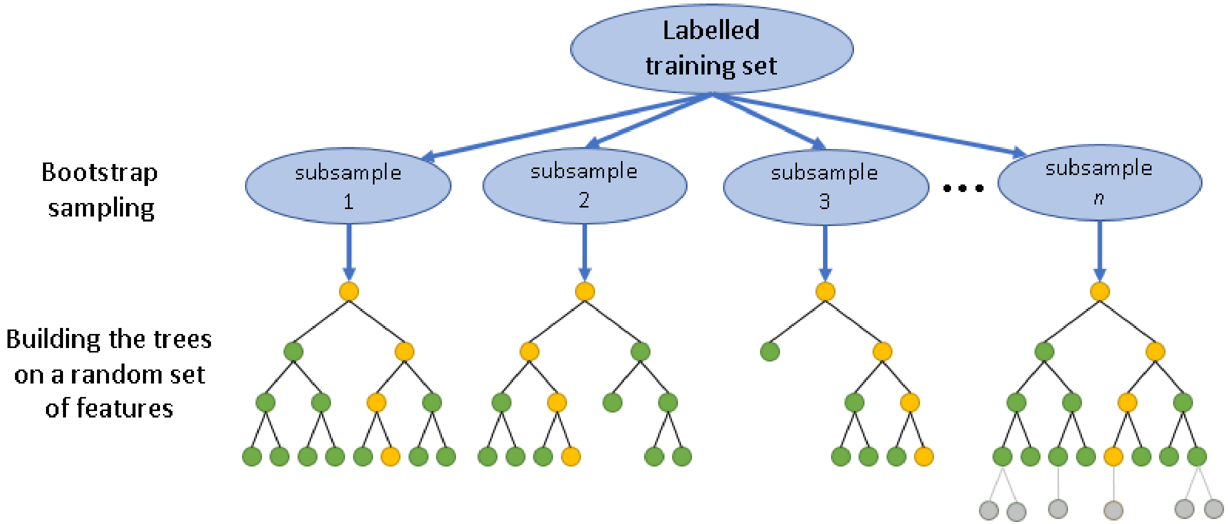


➔ *Boostraping means resampling multiple times the original data*

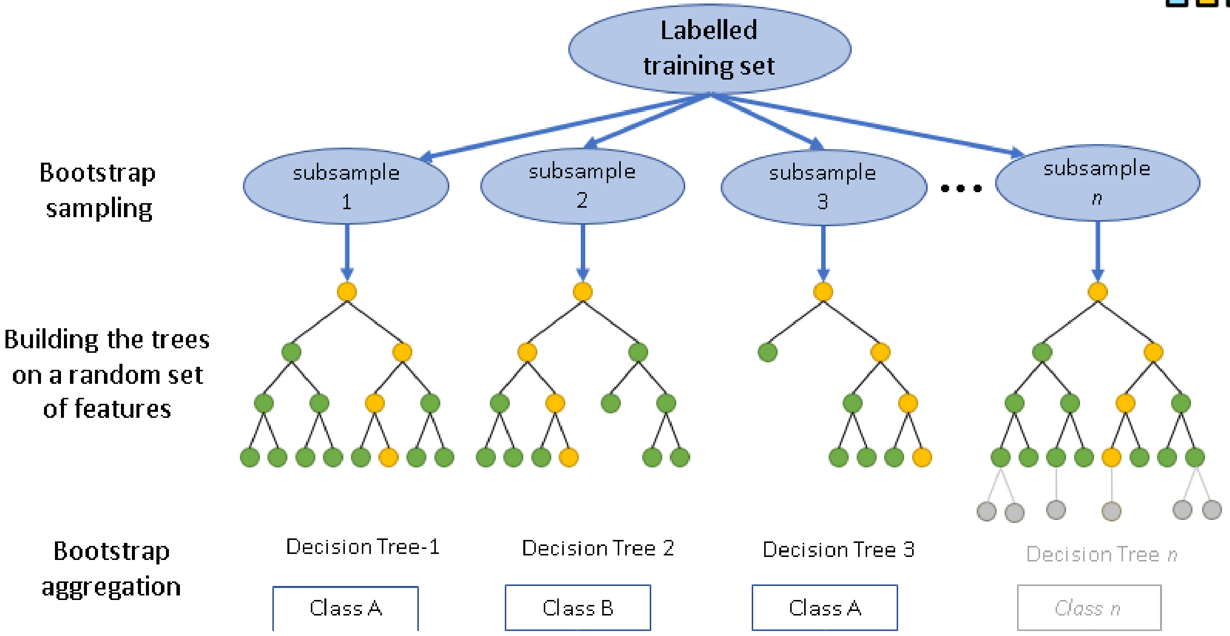


Random Forest

Step 2: Build trees using the bootstrap samples

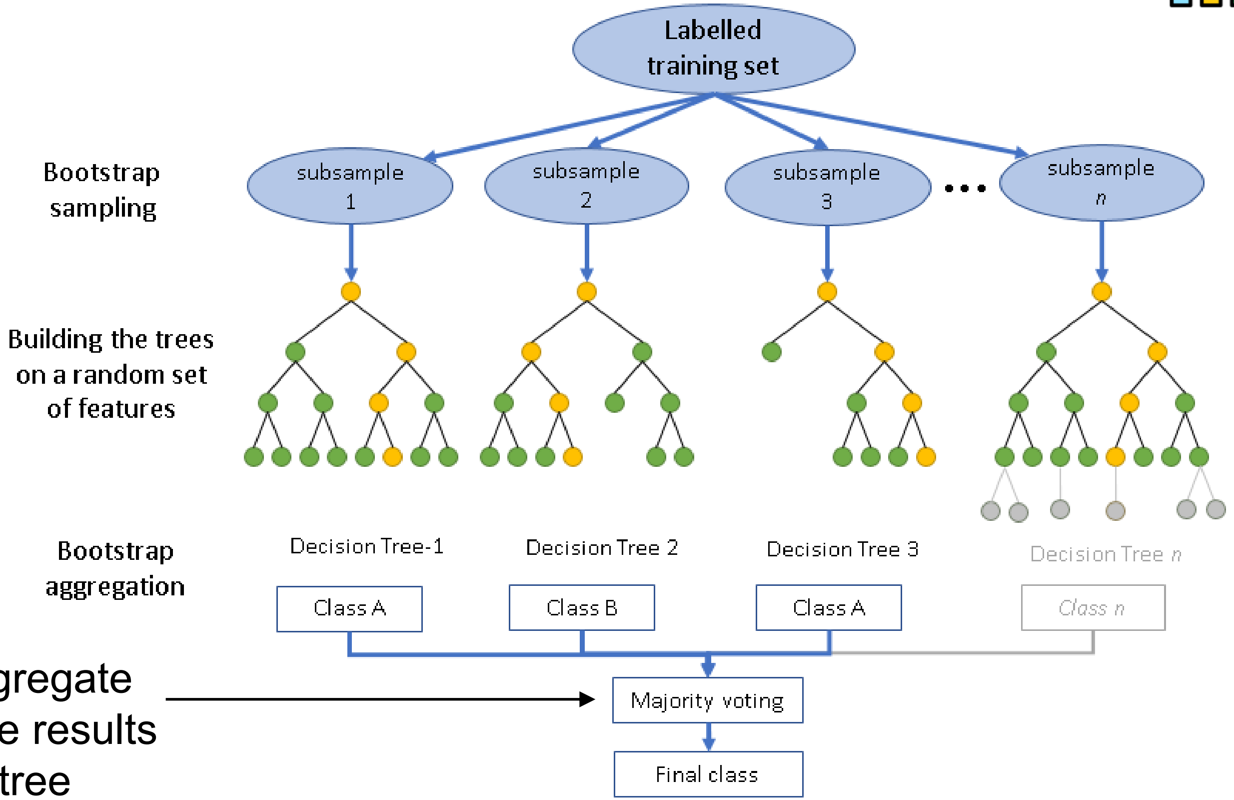


Random Forest



Step 3: Each tree makes its own prediction

Random Forest

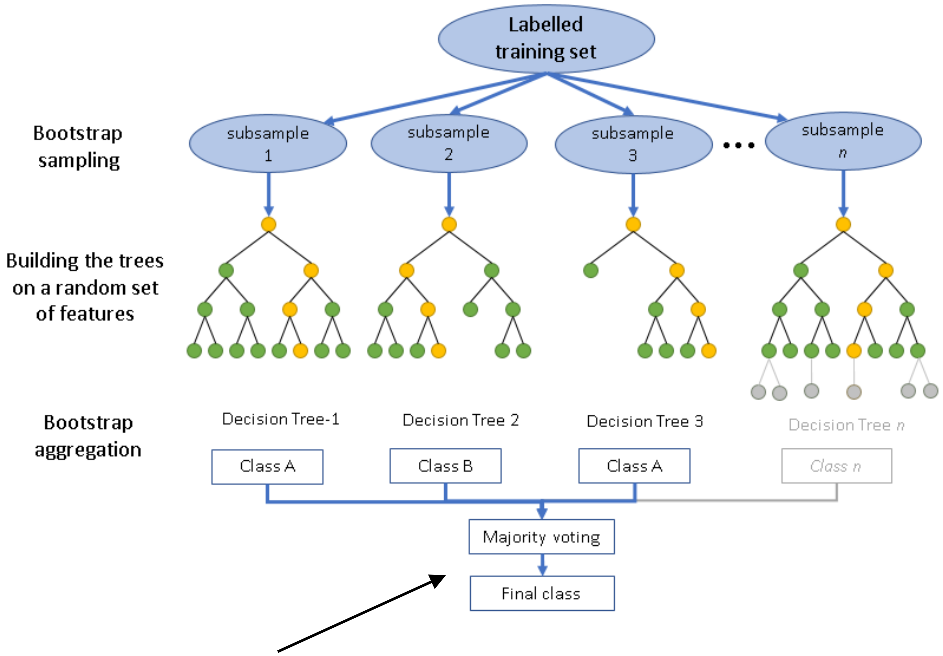


Step 4: Aggregate (combine) the results of each tree

Random Forest

Final prediction

- **Regression**: Compute the average value of each tree predictions
- **Classification**: Apply a majority vote between each class predicted



Majority voting aggregation for classification

Boosting

Build multiple trees <u>sequentially</u> by adapting from previous errors, instead of building them independently

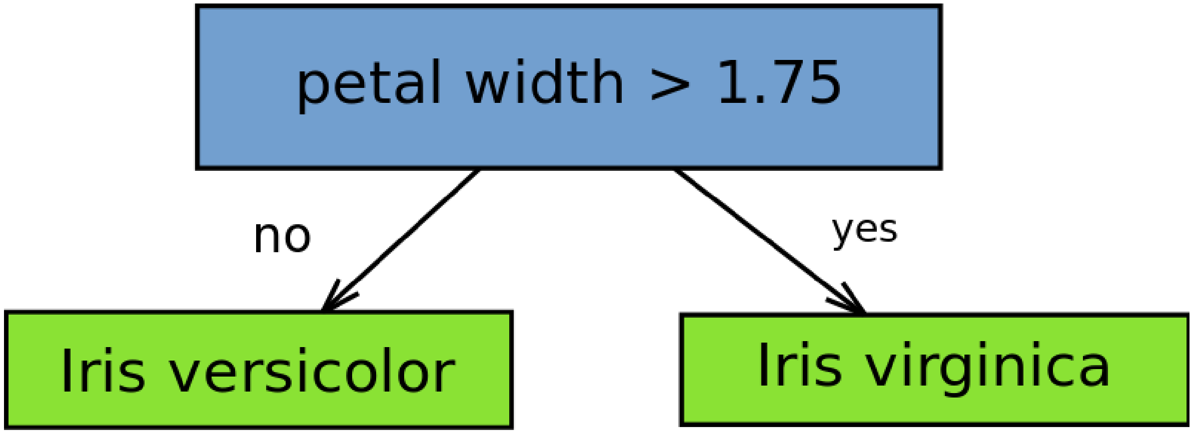
Boosting

Build multiple trees **sequentially** by adapting from previous errors, instead of building them independently

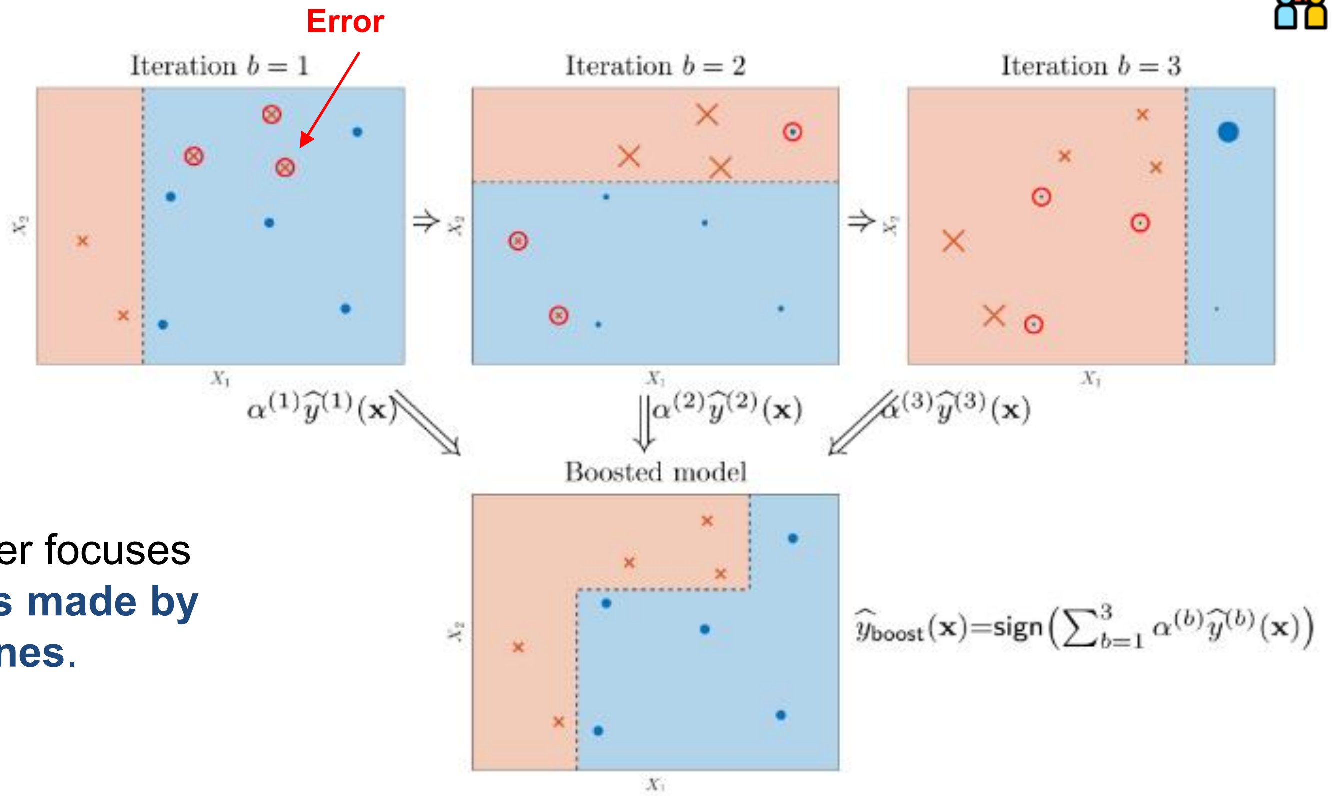
➔ Any **weak learner** can be used, not only decision trees

Boosting

Weak learners are models that perform slightly better than a random guess (with a 50% accuracy)



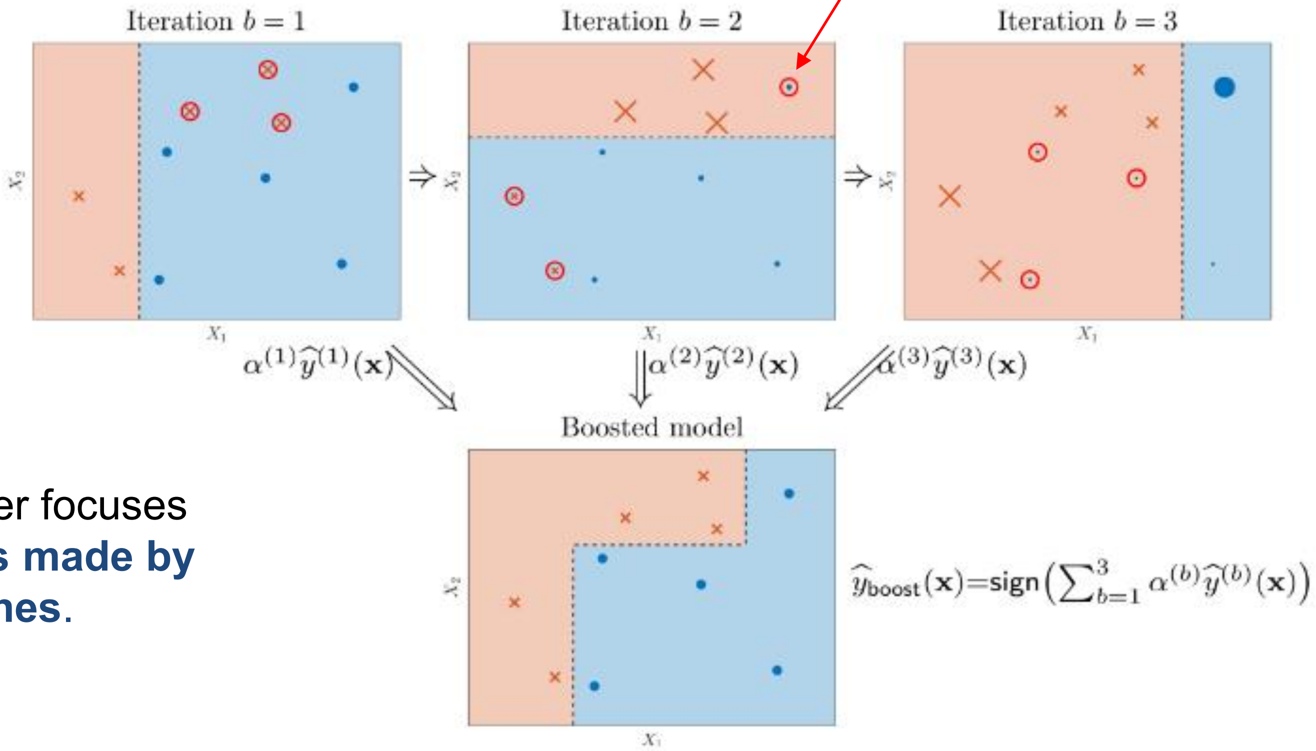
Boosting



Each new learner focuses on the **mistakes made by the previous ones**.

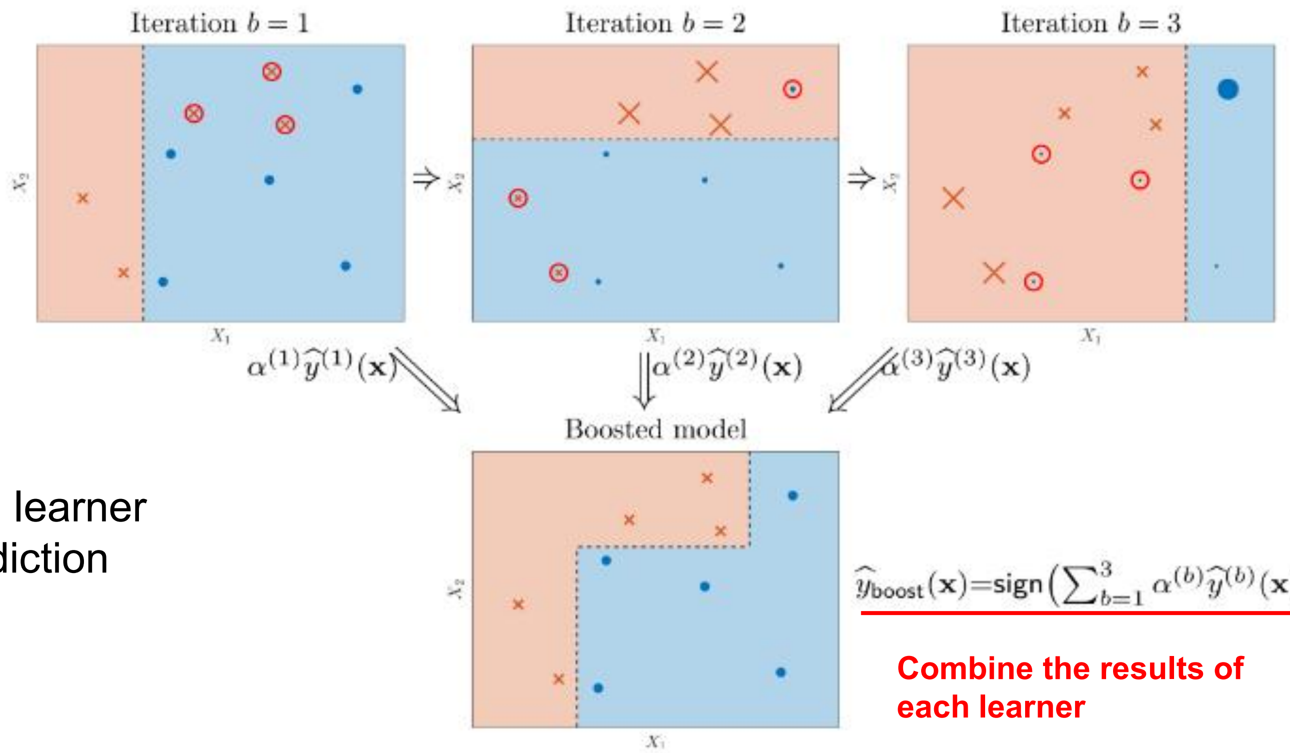
Boosting

Focus on the previous errors



Each new learner focuses on the **mistakes made by the previous ones**.

Boosting



Aggregate each learner for the final prediction

$$\widehat{y}_{\text{boost}}(\mathbf{x}) = \text{sign}\left(\sum_{b=1}^{3} \alpha^{(b)}\widehat{y}^{(b)}(\mathbf{x})\right)$$

Combine the results of each learner

Different types of Boosting

- **AdaBoost**: Adjust the weights of training data
- **Gradient Boosting**: Correct residual errors
- **Extreme Gradient Boosting (XGBoost)**: Fast implementation of Gradient Boosting

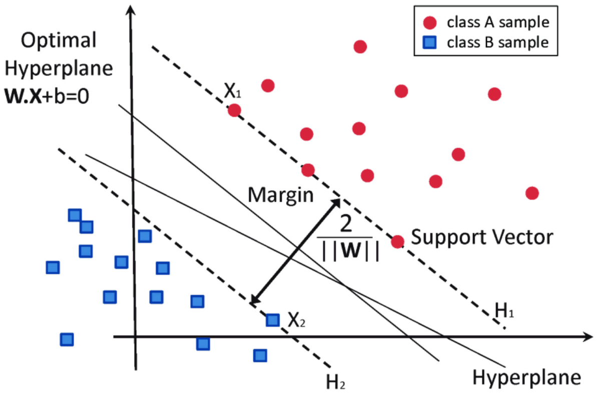
XGBoost

Summary of Supervised models

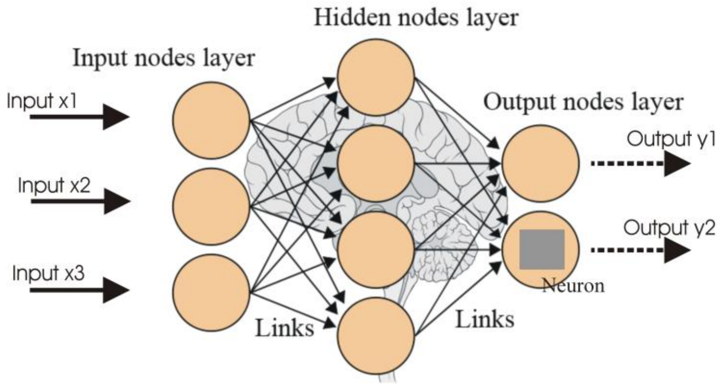
Models	Complexity	Hyperparameters
Linear Regression	Low	None
Logistic Regression	Low	• **C**: regularization strength
K nearest neighbors	Low	• **n_neighbors:** number of neighbors (k)
Decision Trees	Medium	• **max_depth:** depth of the tree • **min_samples_leaf:** min number of points in a leaf node)
Random Forest	High	• Same as Decision Tree • **n_estimators**: Number of trees • **max_features**: Number of features for building trees
Boosting	High	• Same as Decision Tree + n_estimators • **learning rate**: Scale each trees contribution • **Subsample**: Fraction of samples used to build individual trees

Other supervised models (classification)

- **Support Vector Machine (SVM):** Find the line that best separates classes by maximizing the margin between them.
- **Neural networks (MLP)**: Layers of "neurons" that process input data by adjusting weights, mimicking the way the human brain works.



Support Vector Machine (SVM)



Neural Network

Model training with Scikit-learn

Demo of how to use scikit-learn for model training

https://scikit-learn.org/stable/supervised_learning.html

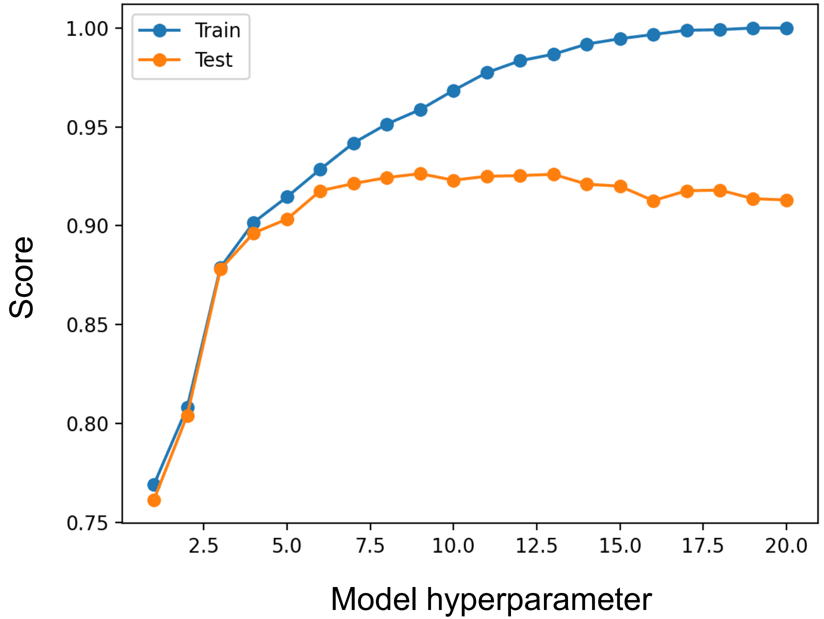
Model evaluation

Evaluating a Machine Learning model

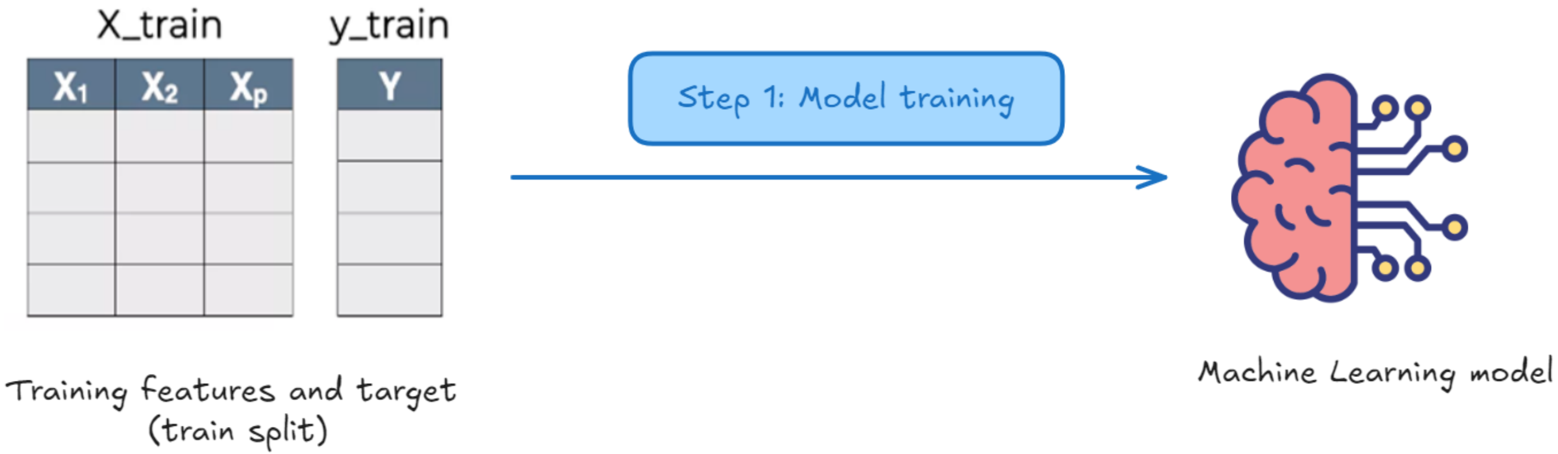
- Evaluation in Machine Learning is performed by **comparing the predicted values of a model with true values**

Evaluating a Machine Learning model

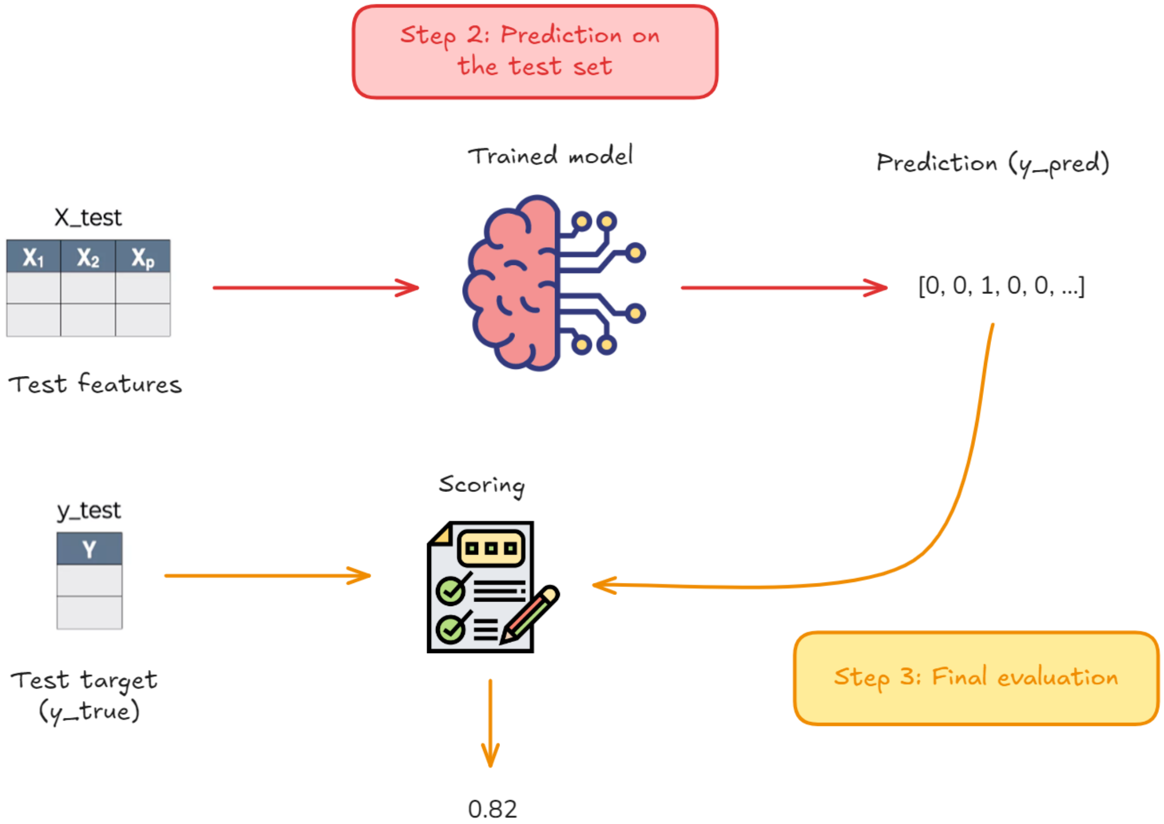
- Evaluation is done on the **test data**. This ensure the model is **reliable** on data it hasn't seen during training.



Evaluating a Machine Learning model



Evaluating a Machine Learning model



Evaluation for Classification models

Accuracy

Ratio of correct predictions to the total number of predictions

$$Accuracy = \frac{Nbr\ of\ correct\ predictions}{Total\ nbr\ of\ predictions}$$

➔ Accuracy is always between 0 and 1

➔ Reflects the **overall performance of a model**

Evaluation for Classification models

Example of Accuracy

Our classification model predicts two classes (0 and 1)
Let's test its performance on **119 new observations**

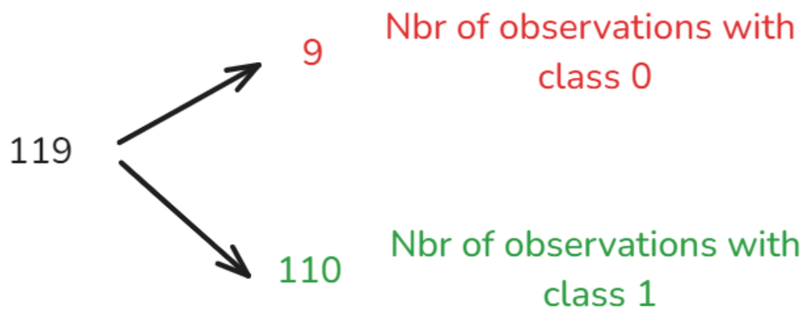
- **105 are correct predictions**
- 14 are incorrect predictions

➔ **Accuracy = 105/119 = 0.88 (88%)**

Evaluation for Classification models

Example of Accuracy

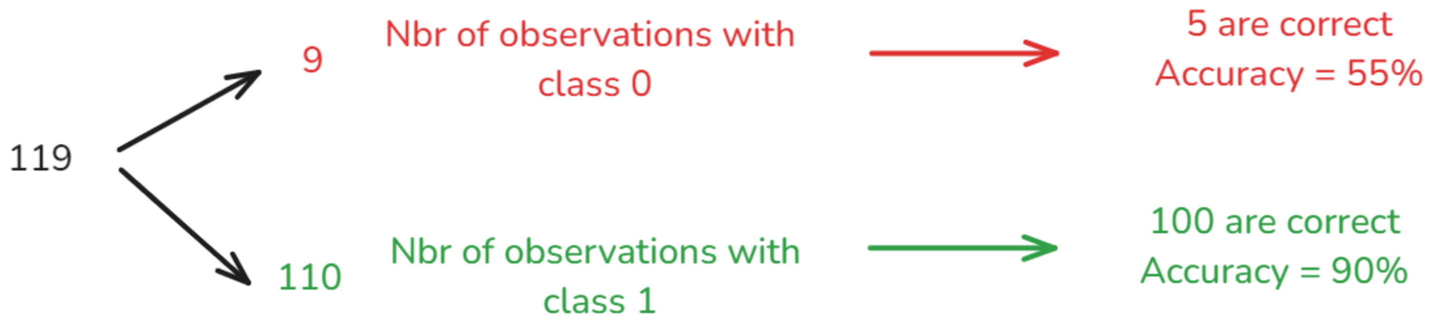
Now let’s study its performance separatly on each predicted class



Evaluation for Classification models

Example of Accuracy

Now let’s study its performance separatly on each predicted class



Accuracy can be misleading if the target classes are not in the same proportion

Evaluation for Classification models

Confusion Matrix

Summarize the performance of a classifier for each predicted class

		Predicted label	
		Positive (1)	**Negative (0)**
True label	**Positive (1)**	**5** True positive	**4** False negative
	Negative (0)	**10** False positive	**100** True negative

Evaluation for Classification models

Confusion Matrix

Summarize the performance of a classifier for each predicted class

Nbr of correct predictions for 1

Predicted label

True label	Positive (1)	Negative (0)
Positive (1)	**5** **True positive**	**4** False negative
Negative (0)	**10** False positive	**100** **True negative**

Nbr of correct predictions for 0

Evaluation for Classification models

Confusion Matrix

Summarize the performance of a classifier for each predicted class

Predicted label

		Positive (1)	Negative (0)
True label	**Positive (1)**	**5** True positive	**4** **False negative**
	Negative (0)	**10** **False positive**	**100** True negative

Nbr of false predictions for 0

Nbr of false predictions for 1

Evaluation for Classification models

Confusion Matrix

Summarize the performance of a classifier for each predicted class

True label \ Predicted label	Positive (1)	Negative (0)
Positive (1)	**5** True positive	**4** False negative
Negative (0)	**10** False positive	**100** True negative

$$\frac{5}{5+4} = 0.55$$ **Class 1 accuracy**

$$\frac{10}{10+100} = 0.90$$ **Class 0 accuracy**

Evaluation for Classification models

Precision

Proportion of predicted positives that were actually correct

$$\frac{TP}{TP + FP}$$

Recall

Proportion of true positives that were correctly identified

$$\frac{TP}{TP + FN}$$

Learn more about Precision and Recall

https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

Evaluation for Classification models

F1-score

Combine precision and recall into a single metric (harmonic mean)

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Evaluation for Classification models

Example of Precision and Recall

Let's take the same example

		Predicted label	
		Positive (1)	**Negative (0)**
True label	**Positive (1)**	**5** True positive	**4** False negative
	Negative (0)	**10** False positive	**100** True negative

Evaluation for Classification models

Example of Precision and Recall

		Predicted label	
		Positive (1)	**Negative (0)**
True label	**Positive (1)**	**5** True positive	**4** False negative
	Negative (0)	**10** False positive	**100** True negative

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{5}{15} = 0.33$$

When the model predicts positive (1), it is correct 33% of the time

Evaluation for Classification models

Example of Precision and Recall

		Predicted label: Positive (1)	Predicted label: Negative (0)
True label	Positive (1)	5 True positive	4 False negative
	Negative (0)	10 False positive	100 True negative

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{5}{9} = 0.55$$

The model correctly identifies 55% of all positive classes

Evaluation for Classification models

Example of F1 score

$$Precision = 0.33$$

$$Recall = 0.55$$

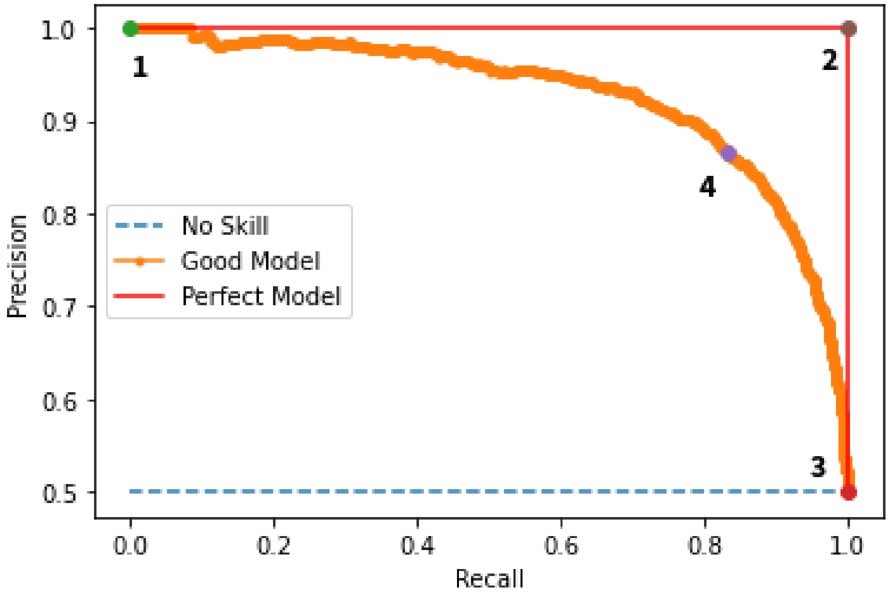
$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \mathbf{0.41}$$

The F1 score is significantly lower than accuracy
(41% vs 88%)

The model has a hard time predicting class 1 correctly

Evaluation for Classification models

Precision and Recall Curve



A « good model » has its curve pointing to the **top right of the graph**

➔ **Tradeoff between Precision and Recall**

Summary of evaluation methods

Metric	Type	Caracteristics	Bounded	Goal
Accuracy	Classification	Overall performance of the model	Yes [0,1]	Maximise
Confusion Matrix	Classification	Study the performance on all predicted classes	x	x
Precision, Recall and F1 score	Classification	Focus on the performance on the positive class (1)	Yes [0,1]	Maximise

Summary of evaluation methods

Metric	Type	Caracteristics	Bounded	Goal
Accuracy	Classification	Overall performance of the model	Yes [0,1]	Maximise
Confusion Matrix	Classification	Study the performance on all predicted classes	x	x
Precision, Recall and F1 score	Classification	Focus on the performance on the positive class (1)	Yes [0,1]	Maximise

For regression models, metrics such as the **Mean Squared Error** or the **R2 score** are used instead

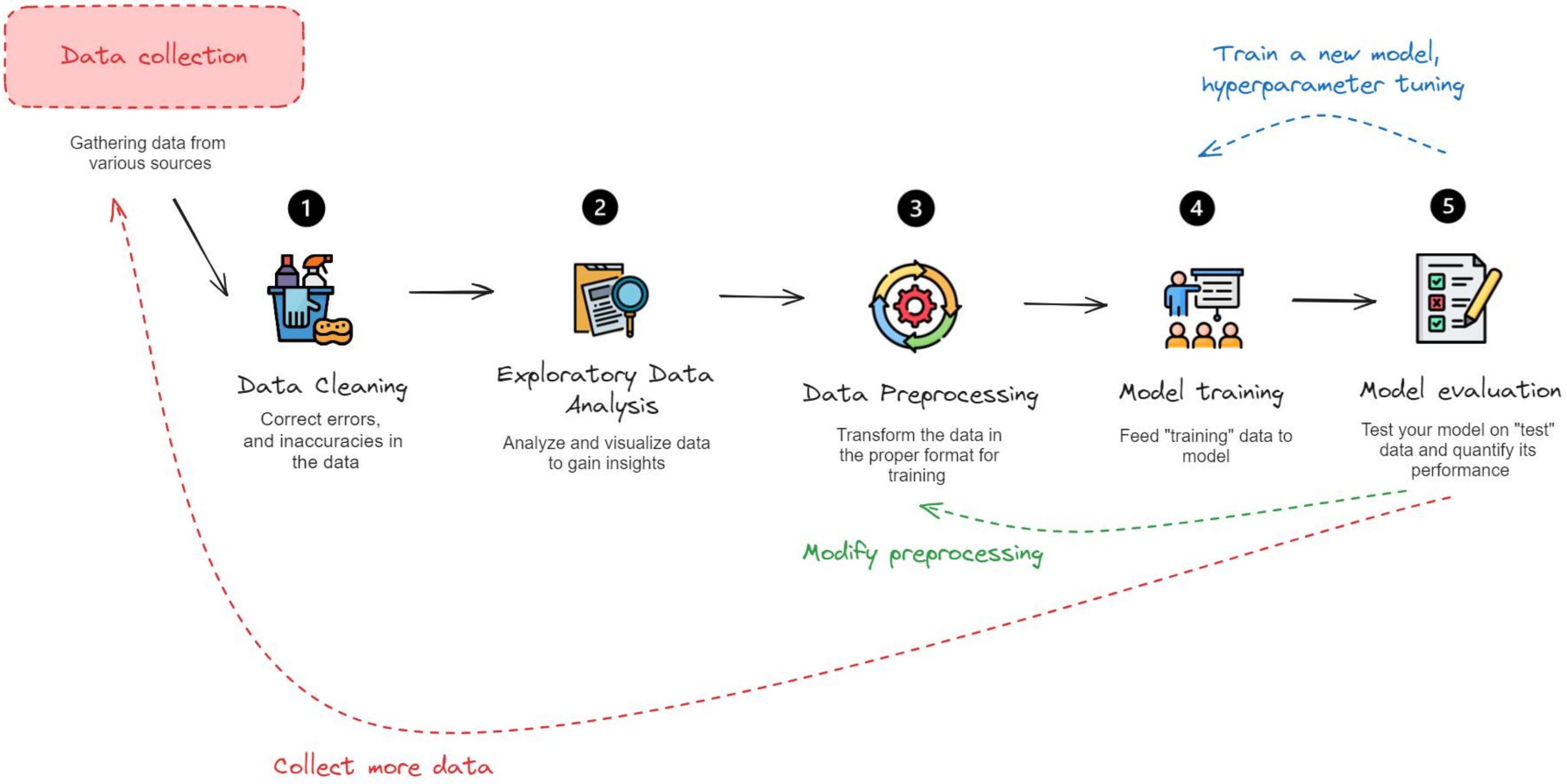
Model Evaluation with Scikit-learn

Demo of how to use scikit-learn for model evaluation

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

Improve the performance of a model

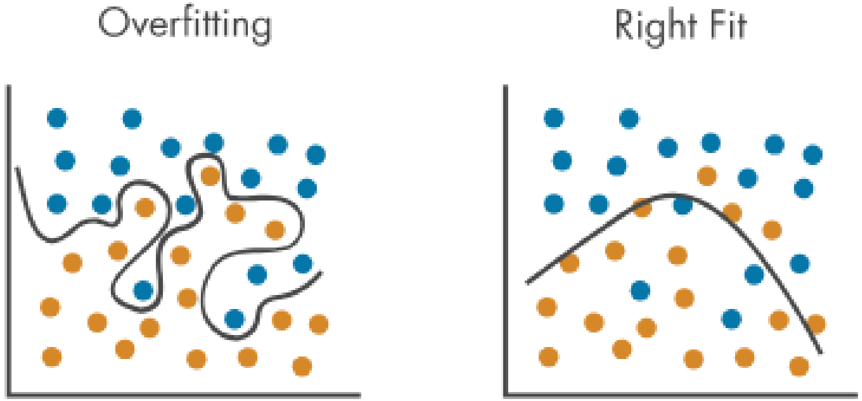
Improve the performance of a model



Common reasons for low performances

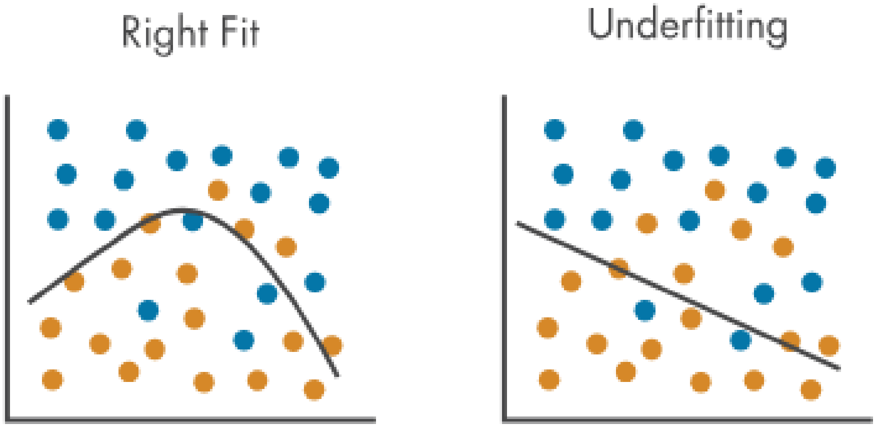
- **Overfitting:** The model has learned too closely on the training data and can't generalize to new data (test set)

➔ This can lead to unstable predictions



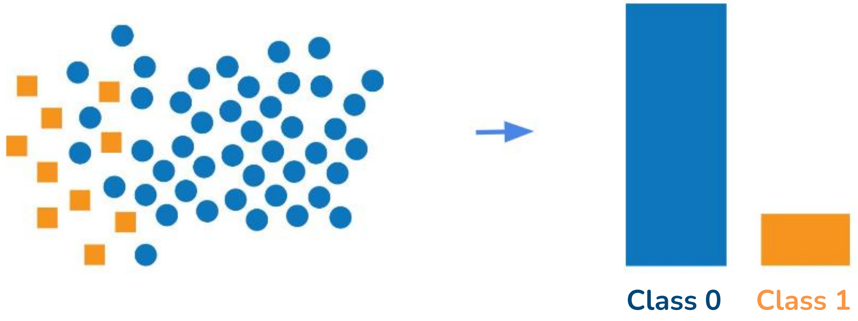
Common reasons for low performances

- **Underfitting:** The model hyperparameters you selected for training aren't optimal for your task

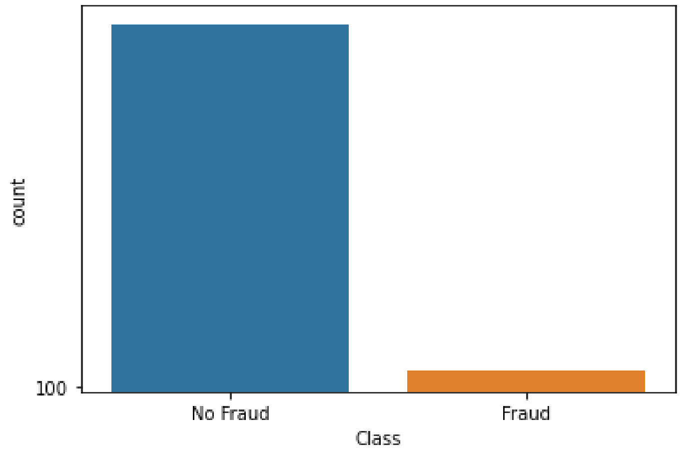


Common reasons for low performances

- **Unbalanced data:** The classes to predict in the target variable aren't in the same proportion.



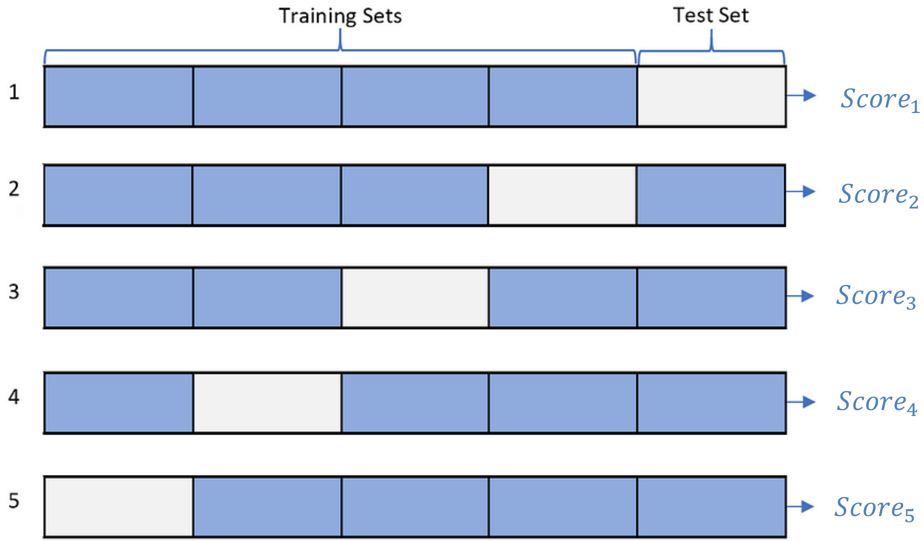
Example of imbalanced data with fraud detection



Reduce overfitting

K-Fold cross validation

Evaluate a model on K number of train/test splits (usually K=5)

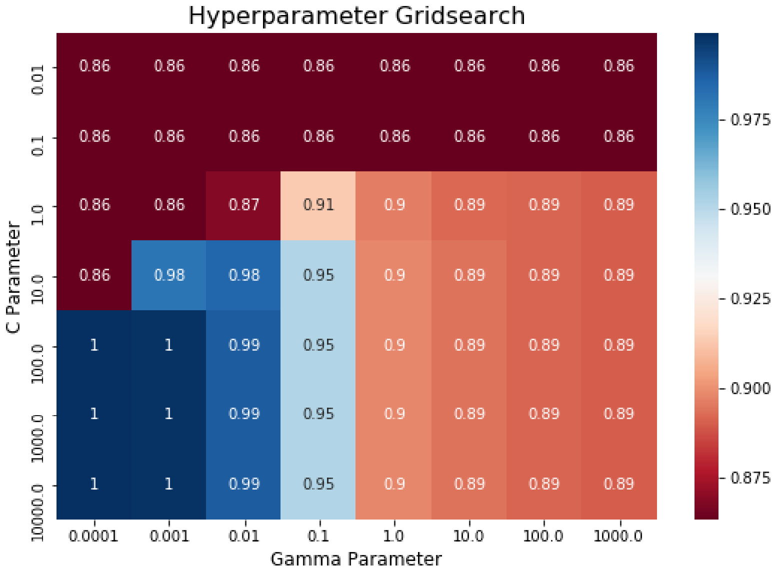


Overall score: $\frac{1}{5}\sum_{i=1}^{5} Score_i$

Hyperparameter tuning

Grid Search

Find the optimal hyperparameters of a model by testing every possible combination



Improve the performance on unbalanced data

Most models have a hyperparameter called **class_weight** to help with unbalanced data.

class_weight : *dict or 'balanced', default=None*

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one.

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

Improve the performance on unbalanced data

Most models have a hyperparameter called **class_weight** to help with unbalanced data.

class_weight : *dict or 'balanced', default=None*

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one.

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

When equal to « ***balanced*** », errors on the minority class will be penalized more heavily during training

Thank you for listening ! 🤗

Do you have any questions ?

HEC PARIS

INSTITUT POLYTECHNIQUE DE PARIS

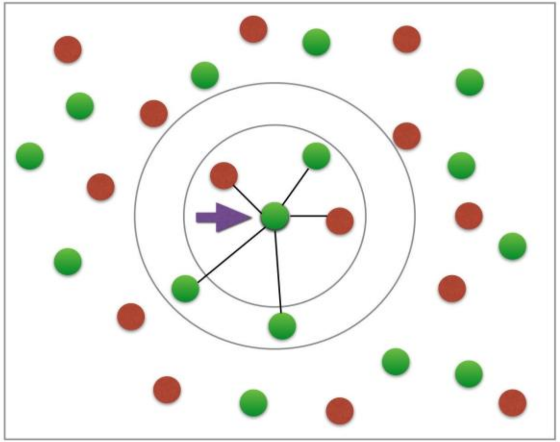
Hi! PARIS

PARIS ARTIFICIAL INTELLIGENCE FOR SOCIETY

Annexe

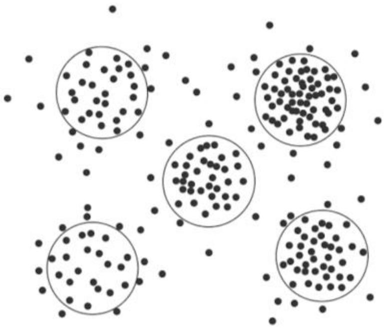
How does scaling improve performances ?

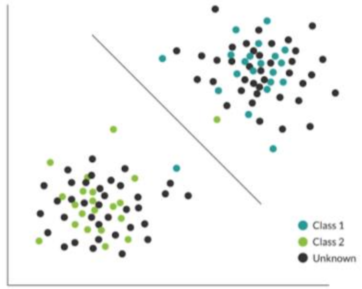
- **Distance-based models**: Models that rely on distances to measure similarity between points and make predictions
- Numerical variables with a large range will contribute more to the decrease in distance ➔ **false sense of importance**



How to choose a model to train ?

- **What is the <u>business objective</u>, the end goal ?**
  - Predict known values or discover unknown patterns ? (*Supervised vs Unsupervised Learning*)
  - Predict a continuous or categorical value ? (*Regression vs Classification*)





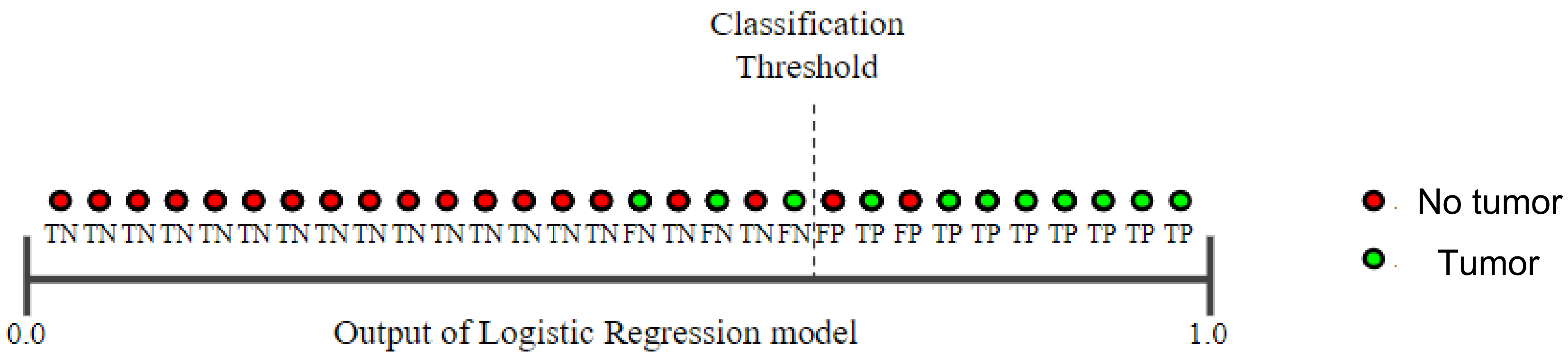
How to choose a model to train ?

- Size of the training data ?
- Accuracy vs interpretability of the output ?
- Access to powerful computing ressources ?

Evaluation for Classification models

Tradeoff between Precision and Recall

Improving Precision tends to reduce Recall (and vice versa)



Evaluation for Regression models

Mean squared error:

Measure the distance between the predicted value and the true value

$$\text{MSE} = \underbrace{\boxed{\frac{1}{n}\sum_{i=1}^{n}}}_{\text{Mean}}\Big(\underbrace{\boxed{Y_i - \hat{Y}_i}}_{\text{Error}}\Big)^{\overbrace{\boxed{2}}^{\text{Squared}}}$$

➔ **Average errors of every predicted observation/point**

Evaluation for Regression models

Root mean squared error:

Compute the square root of the mean squared error

$$RMSE = \sqrt{\boxed{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}}$$

Mean squared error

➔ **More intutive as it is the same unit as the original data**

Evaluation for Regression models

Important information:

- A lower MSE and RMSE signals a better perforemance
- MSE and RMSE are **not bounded between 0 and 1**
- Should only be used to compare models trained on the same data

Evaluation for Regression models

R2 score:

Measure the **goodness of fit** of a regression model (how well it explained the variations in the target variable Y)

$$\textbf{R2 Squared} = 1 - \frac{\textbf{SSr}}{\textbf{SSm}}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

Evaluation for Regression models

R2 score:

Measure the **goodness of fit** of a regression model (how well it explained the variations in the target variable Y)

→ R2 is **bounded between 0 and 1** (not like MSE and RMSE)

→ A higher R2 signals a better model

Summary of evaluation methods

Metric	Type	Caracteristics	Bounded	Goal
Accuracy	Classification	Overall performance of the model	Yes [0,1]	Maximise
Confusion Matrix	Classification	Study the performance on all predicted classes	x	x
Precision, Recall and F1 score	Classification	Focus on the performance on the positive class (1)	Yes [0,1]	Maximise
Mean squared error, Root mean squared error	Regression	Measure the distance between predictions and true values	No	Minimise
R2 score	Regression	Evaluate how well the features explain the variations of the target	Yes [0,1]	Maximise

Random Search

Try randomly chosen combinations of hyperparameters to find well performing (but not necessary optimal) ones