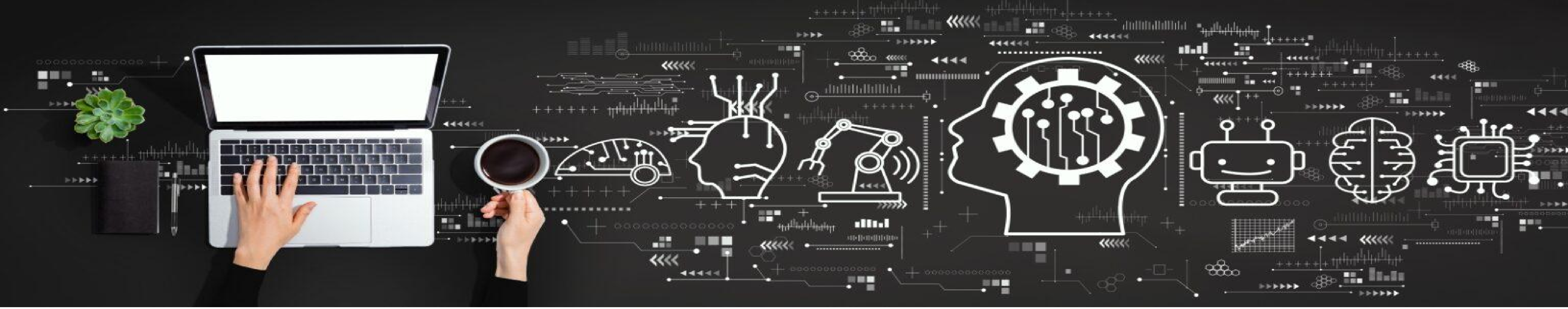


The Fundamentals of Machine Learning

2

Hichem Felouat
hichemfel@gmail.com

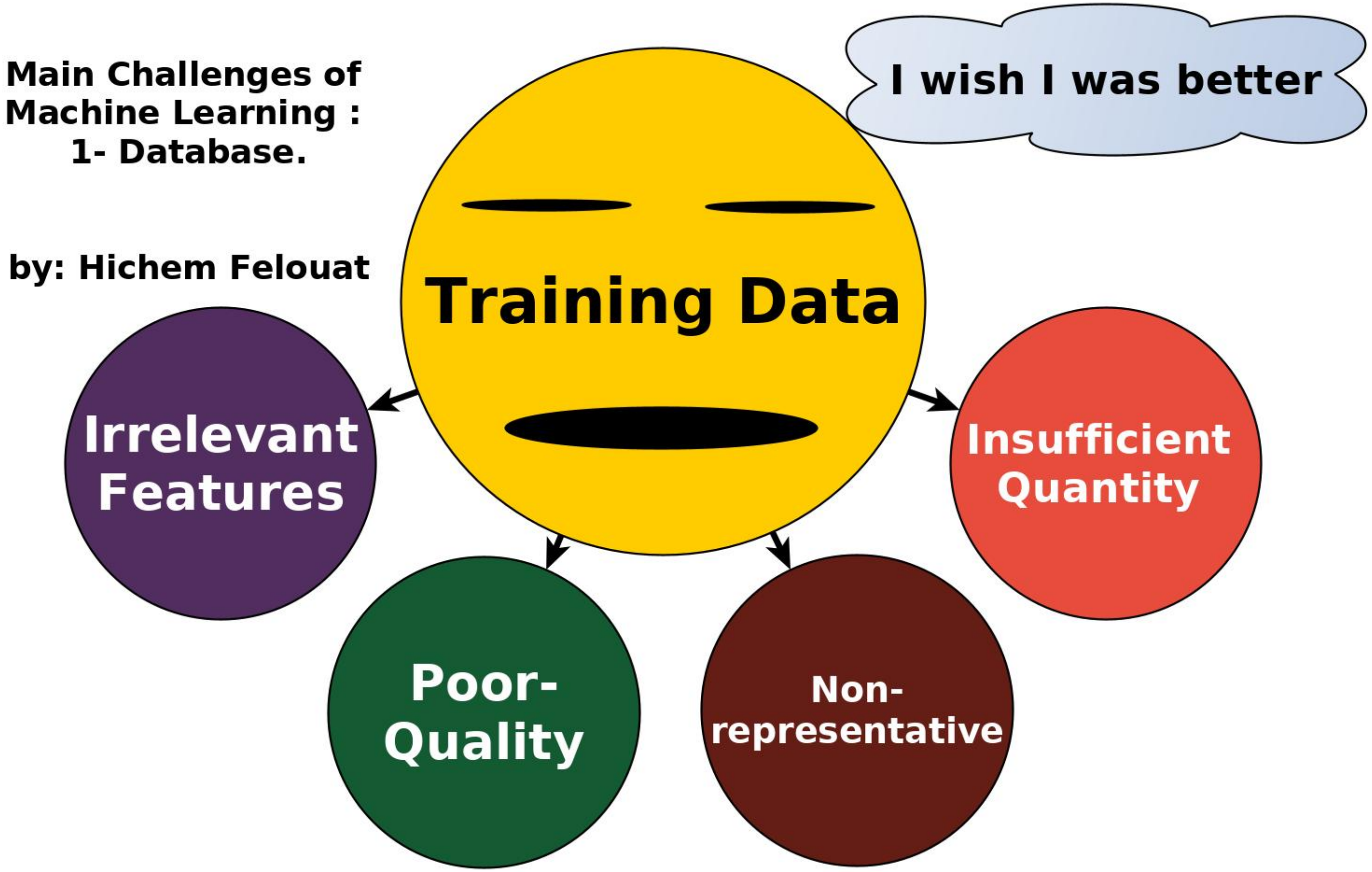
in https://www.linkedin.com/in/hichemfelouat

Main Challenges of Machine Learning

In short, since your main task is to select a learning algorithm and train it on some data, the two things that can go wrong are **"bad data"** and **"bad algorithm"**.

Main Challenges of Machine Learning

1- Database



Main Challenges of Machine Learning

1- Database

1- Insufficient Quantity of Training Data :

Machine Learning takes a lot of data for most Machine Learning algorithms to work properly. Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples (unless you can reuse parts of an existing model).

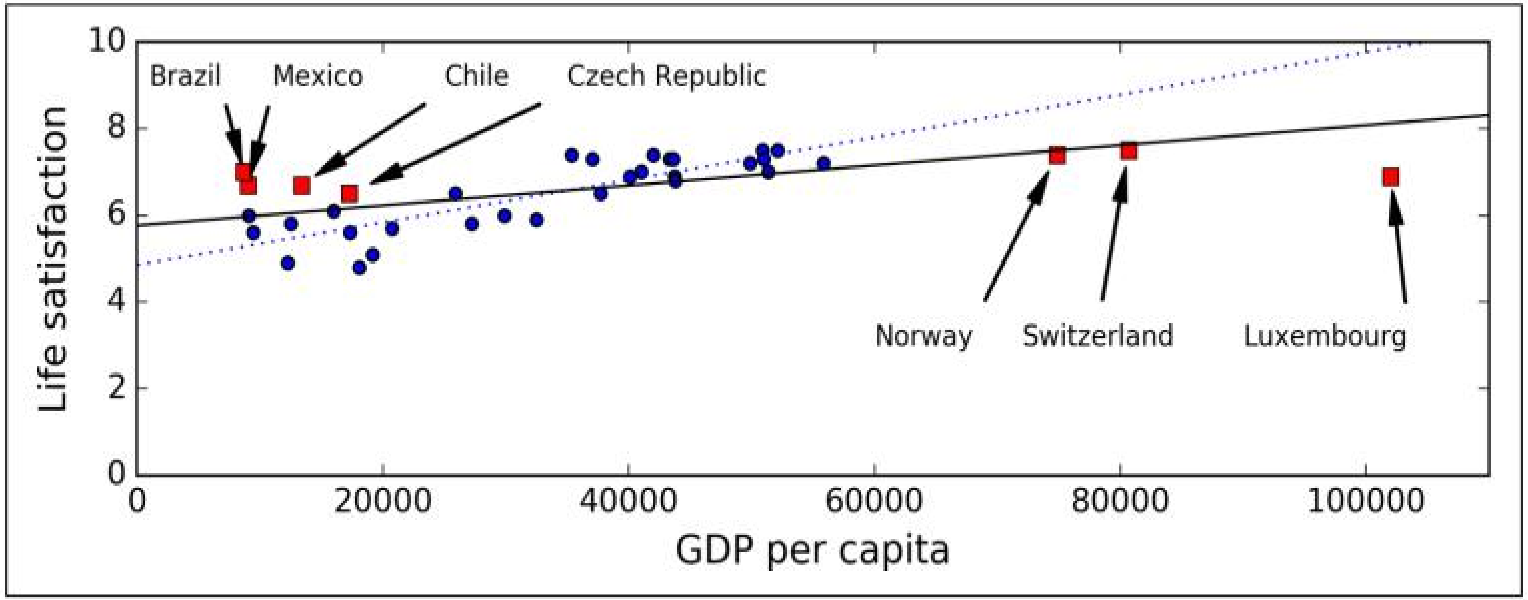
Main Challenges of Machine Learning

1- Database

2) Non-representative Training Data:

In order to generalize well, it is crucial that your training data be representative of the new cases you want to generalize to. This is true whether you use instance-based learning or model-based learning.

Does money make people happier?



Main Challenges of Machine Learning

1- Database

3) Poor-Quality Data:

If your training data is full of errors, outliers, and noise (e.g., **due to poor quality measurements**), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well. **It is often well worth the effort to spend time cleaning up your training data. The truth is, most data scientists spend a significant part of their time doing just that.** For example:

1) **If some instances are clearly outliers, it may help to simply discard them or try to fix the errors manually.**
2) **If some instances are missing a few features (e.g., 5% of your customers did not specify their age), you must decide whether you want to ignore this attribute altogether, ignore these instances, fill in the missing values (e.g., with the median age), or train one model with the feature and one model without it, and so on.**

Main Challenges of Machine Learning

1- Database

4) Irrelevant Features:

Your system will only be capable of learning if the **training data contains enough relevant features** and not too many irrelevant ones. A critical part of the success of a Machine Learning project is coming up with a good set of features to train on. This process, called ***feature engineering***, involves:

1) **Feature selection:** selecting the most useful features to train on among existing features.
2) **Feature extraction:** combining existing features to produce a more useful one (dimensionality reduction algorithms can help).
3) **Creating new features** by gathering new data.

Main Challenges of Machine Learning
2- Algorithm

1) Overfitting the Training Data:

Overfitting happens when a model **learns the detail and noise in the training data** to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

The model performs well on the training data, but it does not generalize well.

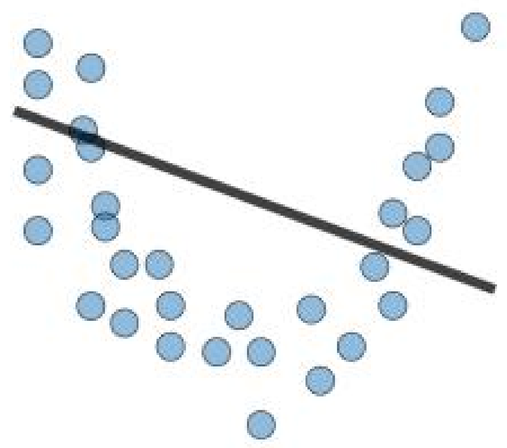
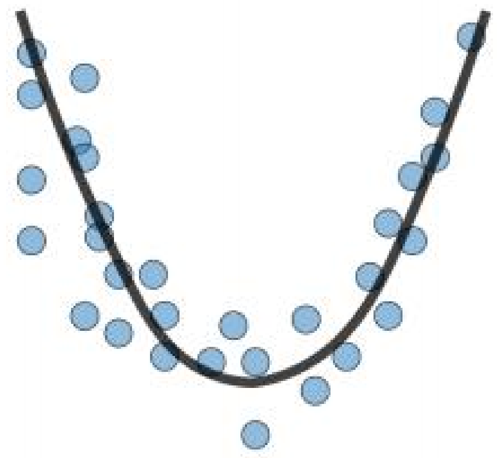
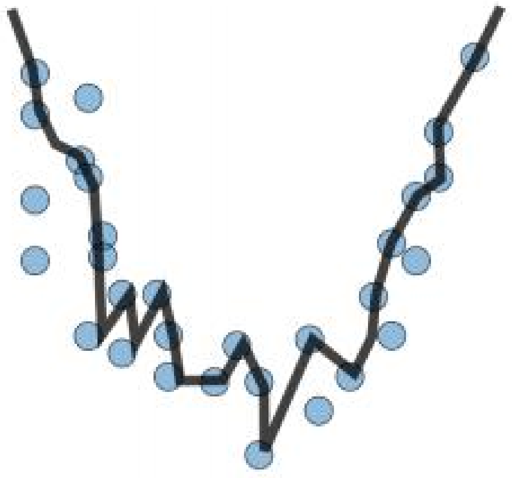
Main Challenges of Machine Learning

2- Algorithm

2) Underfitting the Training Data:

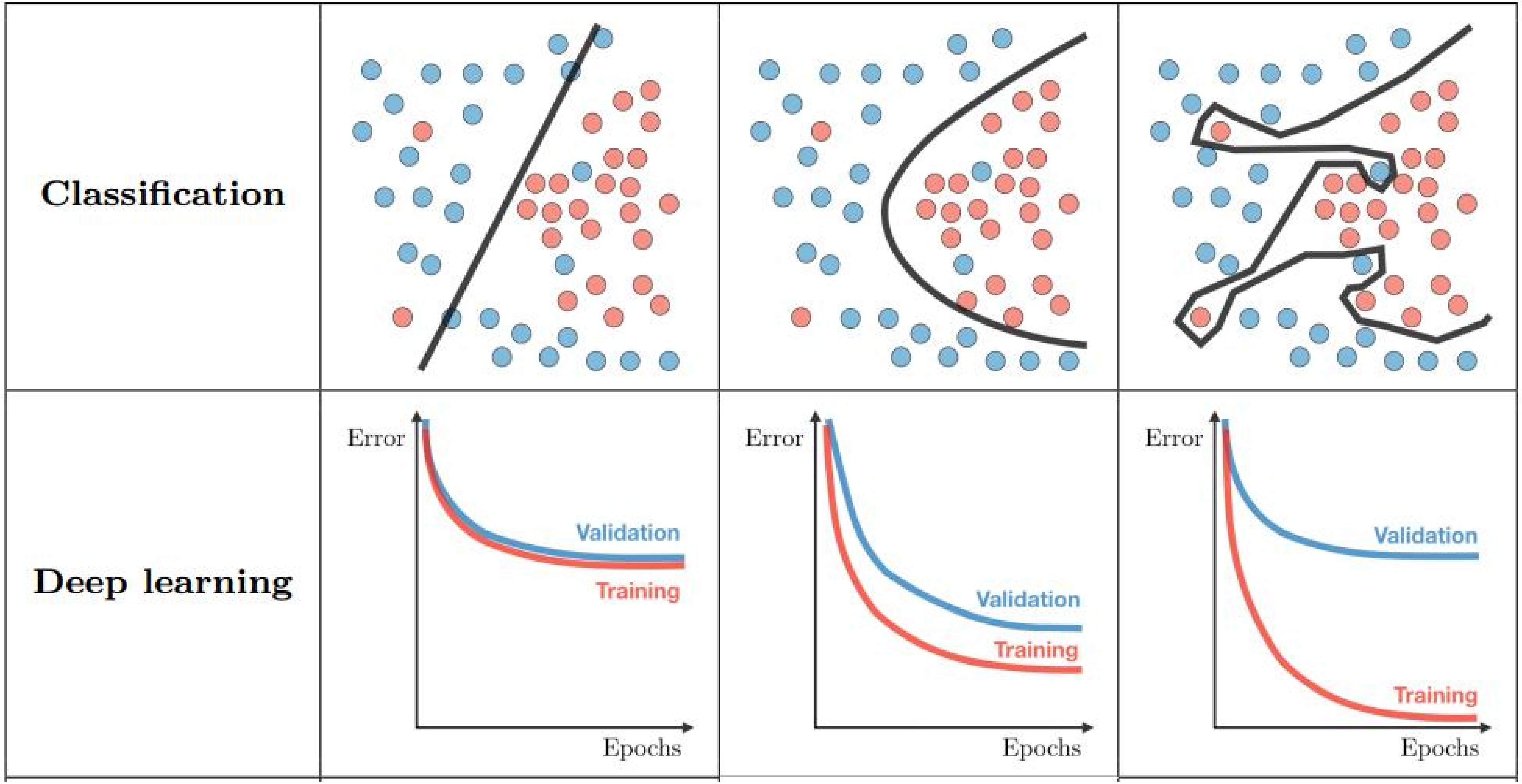
Underfitting is the opposite of overfitting: it occurs when **your model is too simple** to learn the underlying structure of the data.

Main Challenges of Machine Learning
2- Algorithm

	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			

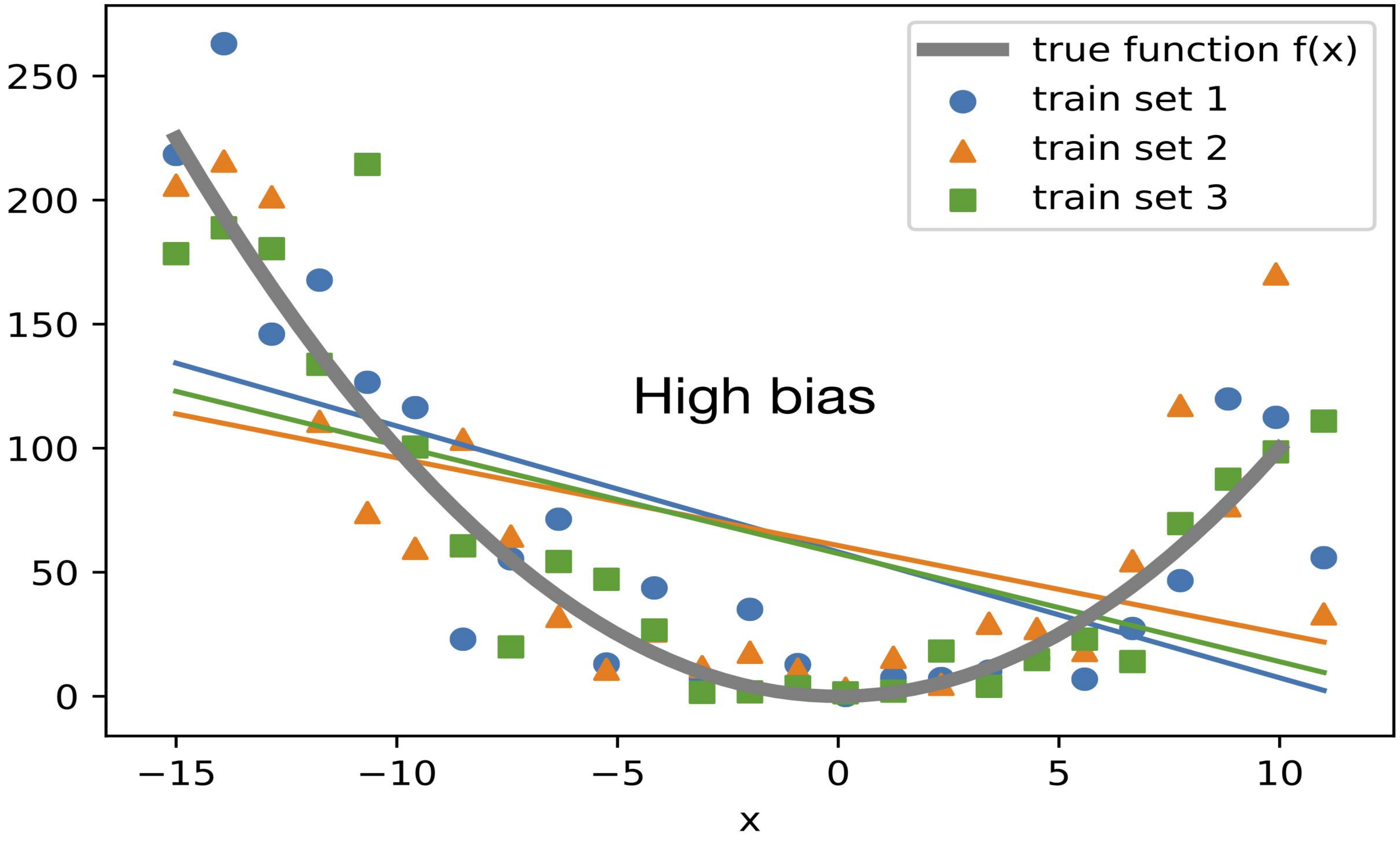
Main Challenges of Machine Learning

2- Algorithm

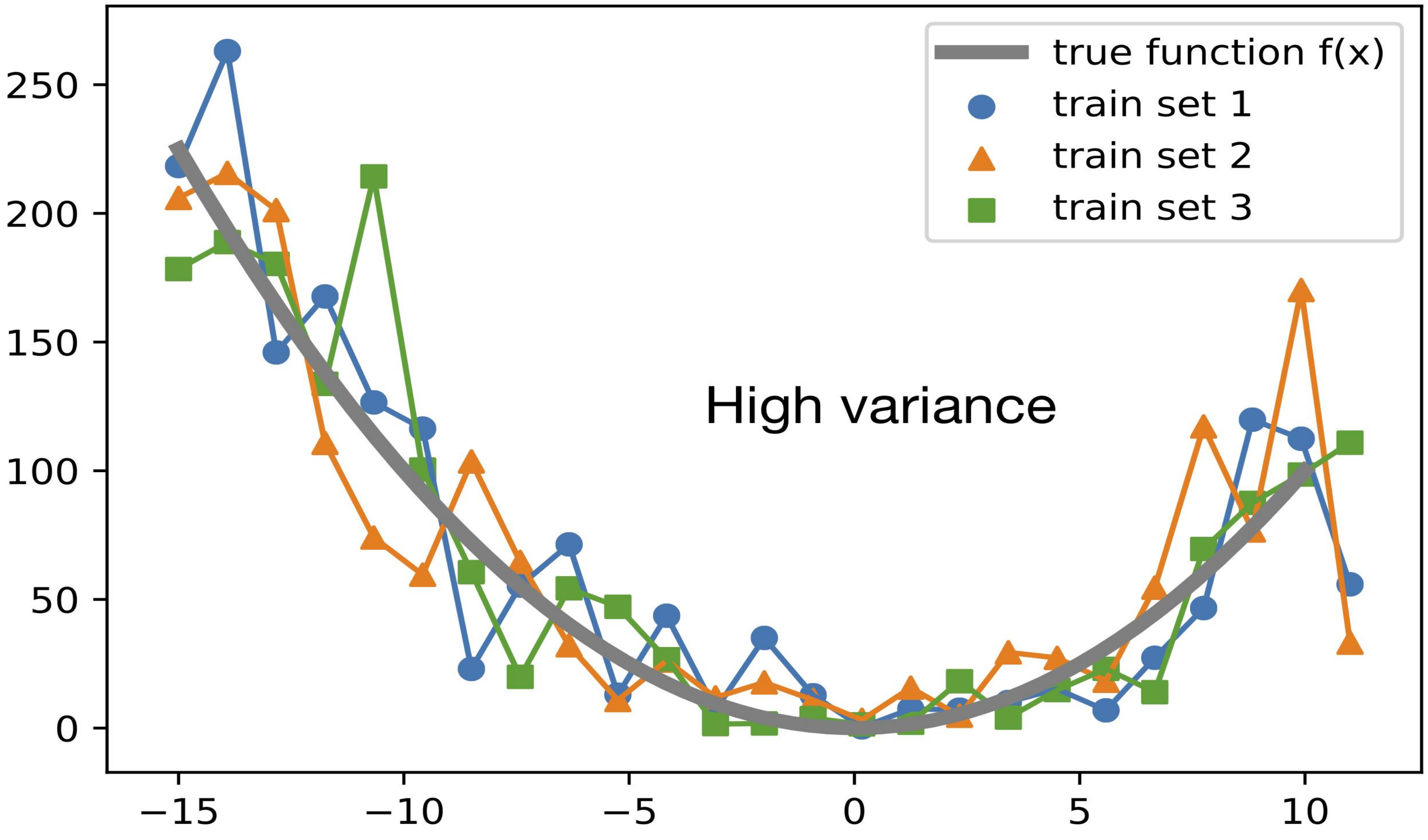


Main Challenges of Machine Learning

2- Algorithm



Main Challenges of Machine Learning
2- Algorithm



How to Avoid Underfitting and Overfitting

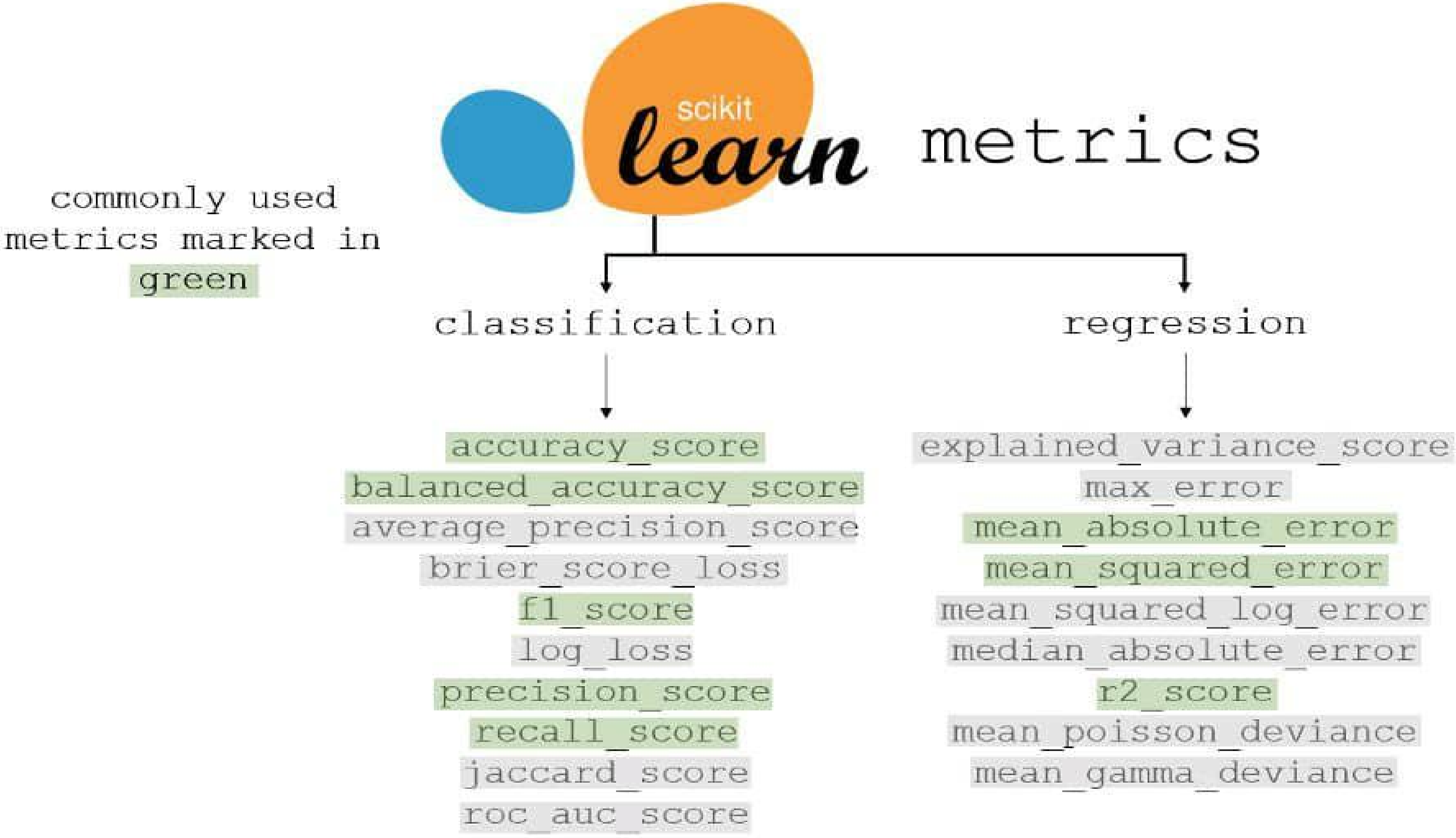
Underfitting :

- Complexify model
- Add more features
- Train longer

Overfitting :

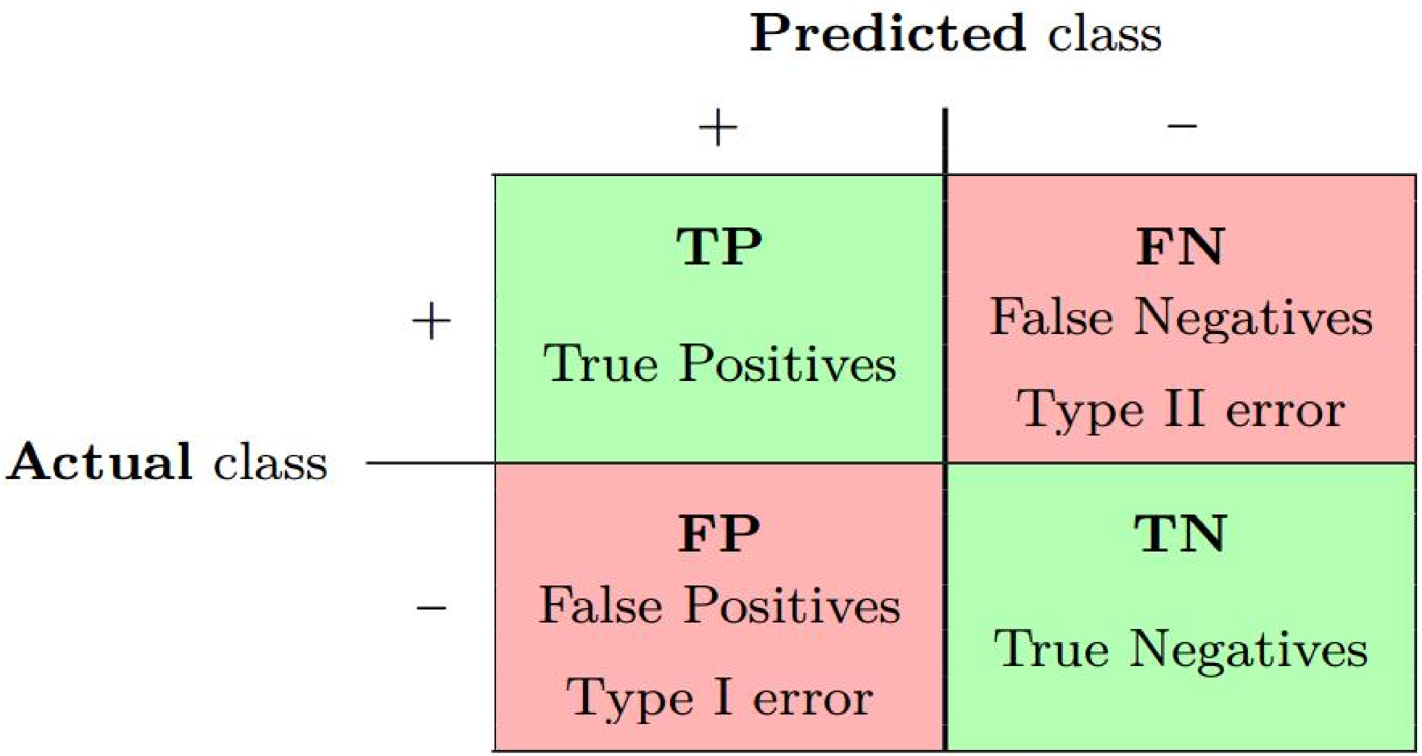
- validation
- Perform regularization
- Get more data
- Remove/Add some features

Evaluation Metrics



Common Classification Model Evaluation Metrics : Confusion Matrix

The confusion matrix is used to describe the performance of a classification model on a set of test data for which true values are known.



Common Classification Model Evaluation metrics : Main Metrics

Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

Common Classification Model Evaluation metrics : Main Metrics

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Recall, sensitivity
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

Common Classification Model Evaluation Metrics : Confusion Matrix

Multilabel Classification:
- Accuracy
- Hamming loss

Hamming loss

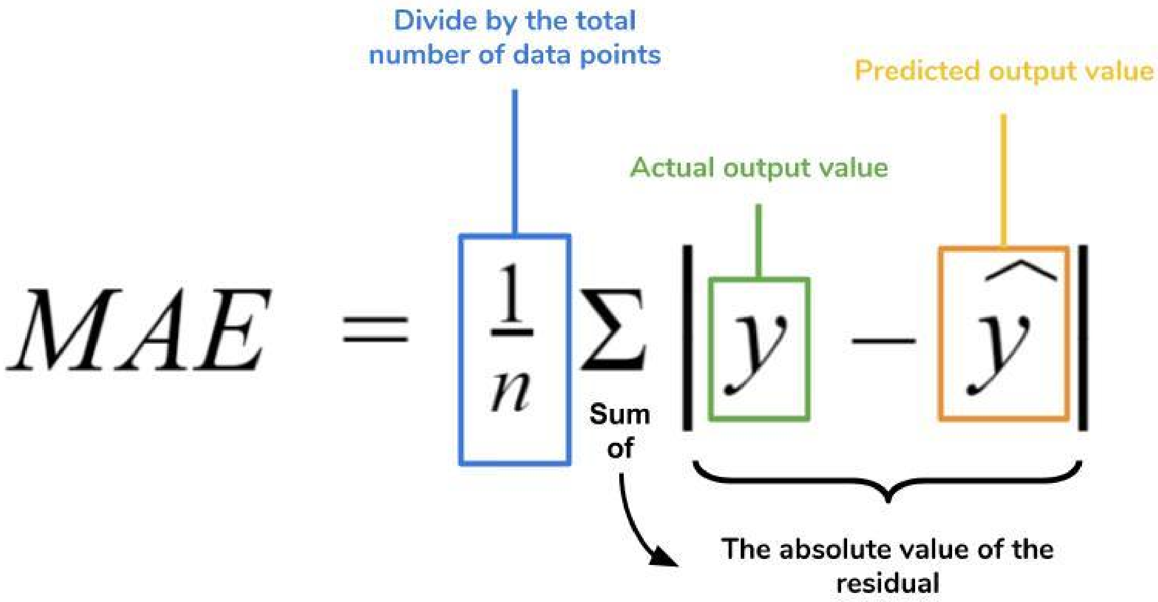
True labels / Predicted labels

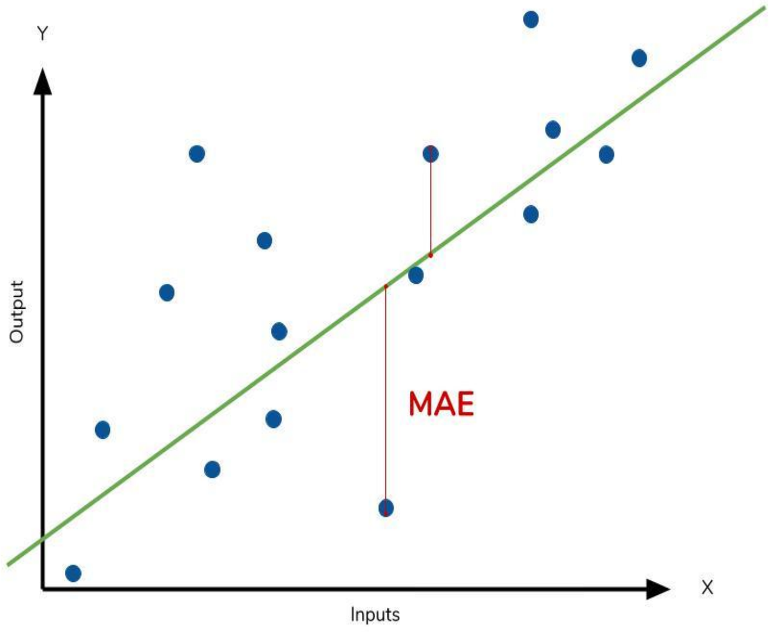
TEXT	SERVICE	FOOD	ANECDOTES	PRICE	AMBIENCE	SERVICE	FOOD	ANECDOTES	PRICE	AMBIENCE
but the staff was so horrible to us	1	0	0	0	0	0	1	0	0	0
to be completely fair the only redeeming facto...	0	1	1	0	0	1	1	0	0	0
the food is uniformly exceptional with a very ...	0	1	0	0	0	0	0	0	1	0
where gabriela personaly greets you and recomm...	1	0	0	0	0	1	0	0	0	0
for those that go once and dont enjoy it all i...	0	0	1	0	0	1	0	0	0	0

Total number of predictions (TNP) = 25
Total number of incorrect predictions (TNIP) = 8

Accuracy = TNIP/TNP = 8/25 = 0.32

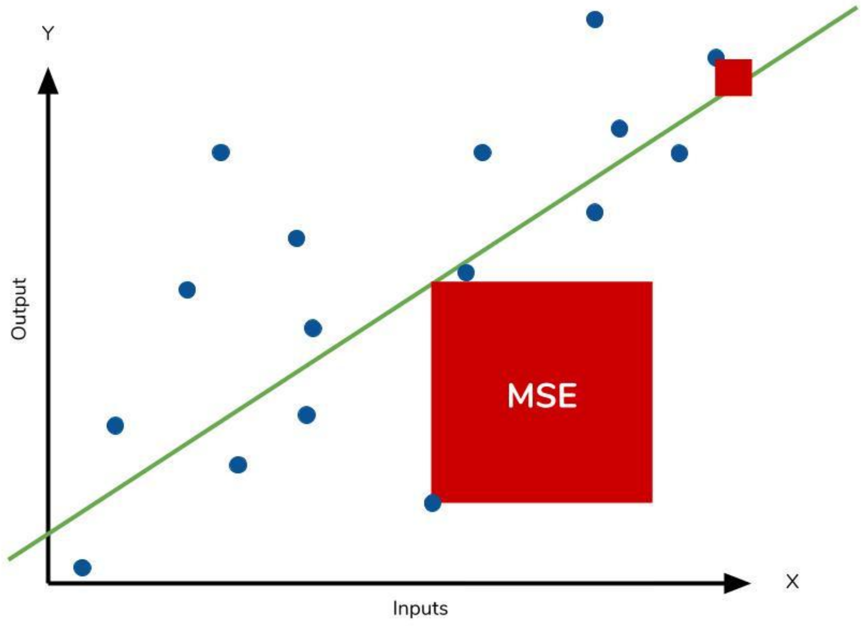
Common Regression Model Evaluation metrics : Mean Absolute Error





Common Regression Model Evaluation metrics : Mean Square Error

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

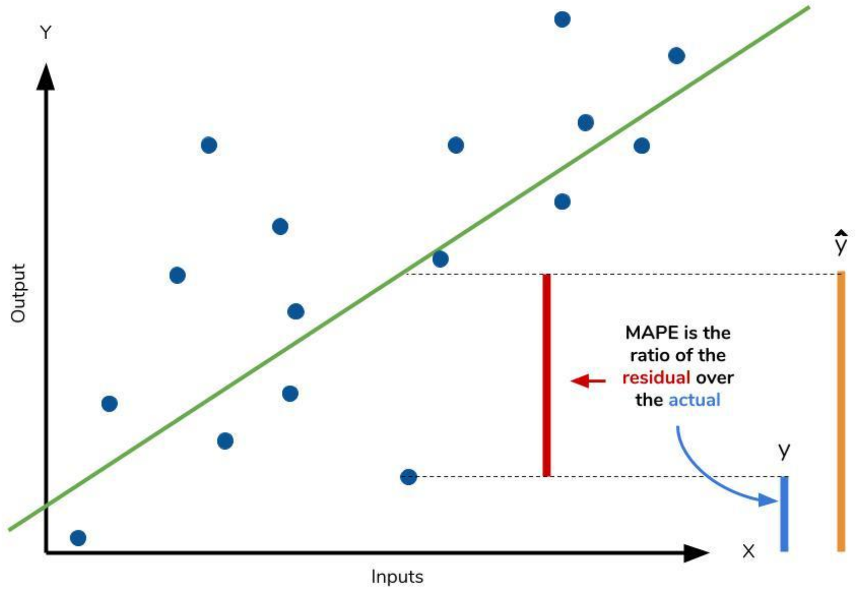


Common Regression Model Evaluation metrics : Mean Absolute Percentage Error

Multiplying by 100% converts to percentage

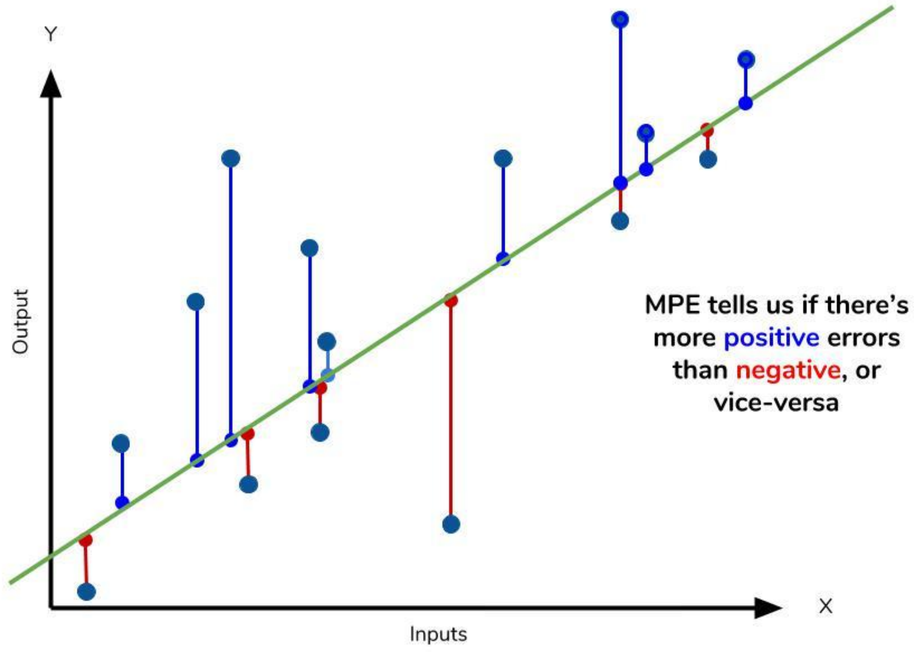
$$MAPE = \frac{\boxed{100\%}}{n}\sum\left|\frac{\overbrace{y-\hat{y}}^{\text{The residual}}}{\boxed{y}}\right|$$

Each residual is scaled against the actual value



Common Regression Model Evaluation metrics : Mean Percentage Error

$$MPE = \frac{100\%}{n} \Sigma \left(\frac{y - \hat{y}}{y} \right)$$



Common Regression Model Evaluation metrics : Mean Percentaae Error

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

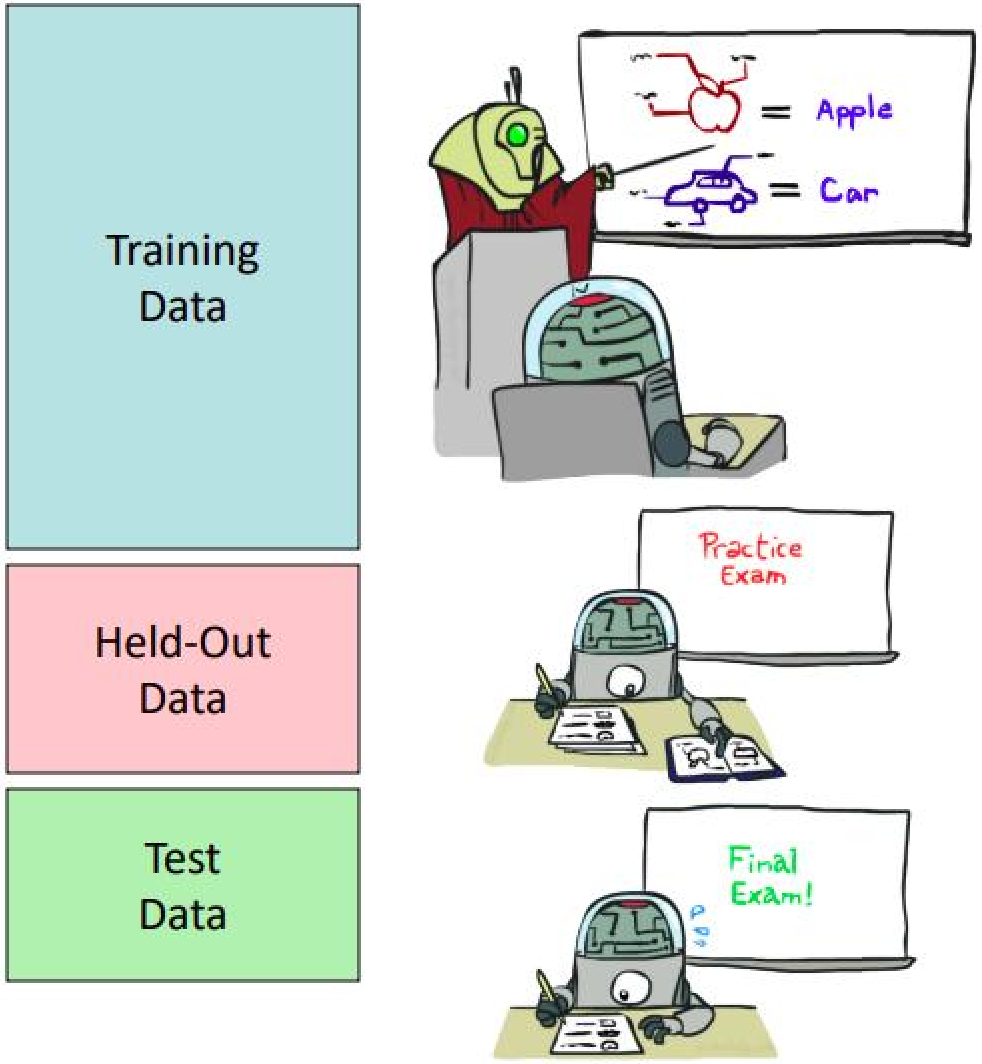
Where,

$\hat{y}$ $-$ *predicted value of* y

$\bar{y}$ $-$ *mean value of* y

Testing and Validating

It is common to use 80% of the data for training and hold out 20% for testing.



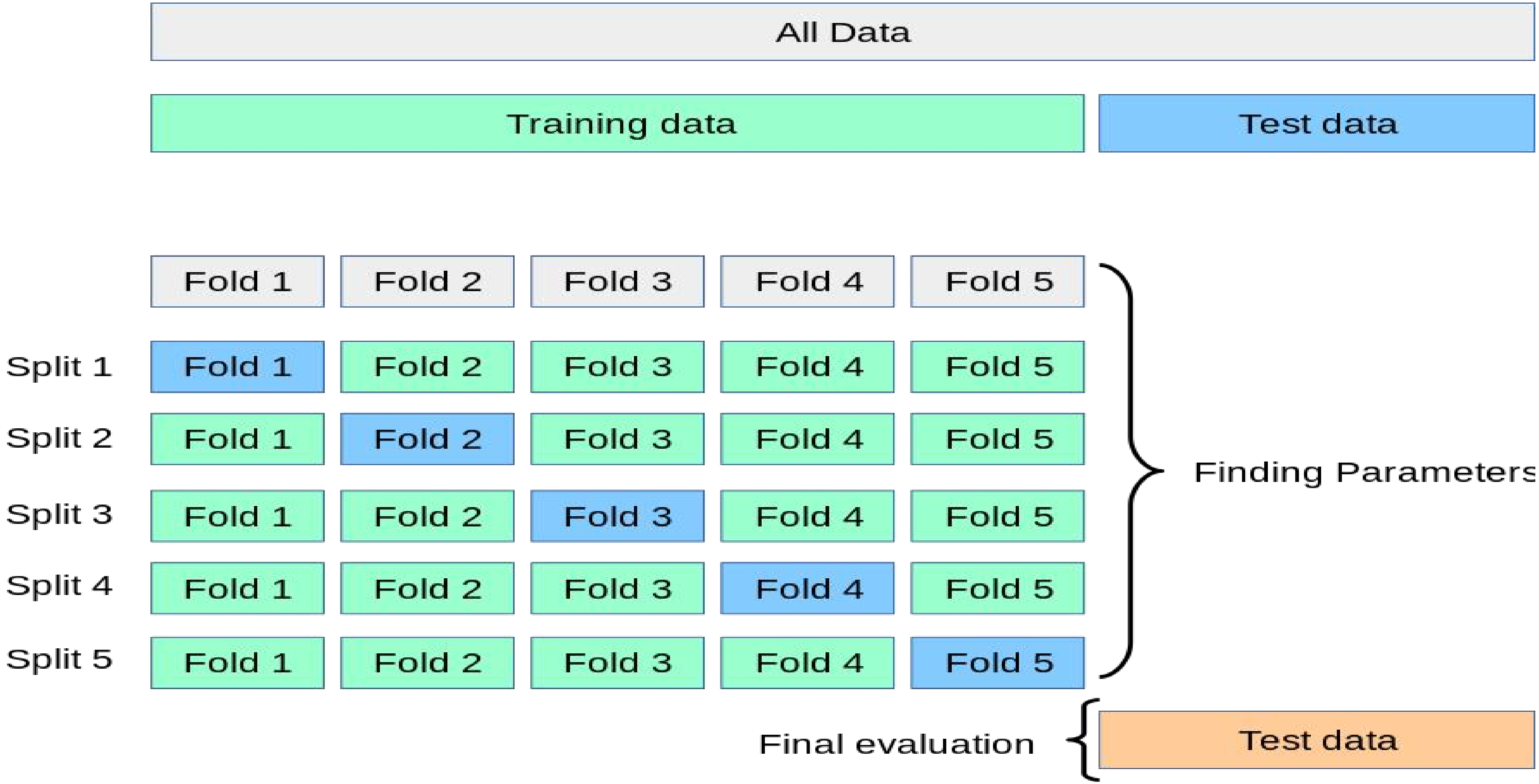
If the training error is low (i.e., your model makes few mistakes on the training set) but the generalization (testing) error is high, it means that your model is overfitting the training data.

Testing and Validating : Cross-Validation

Cross-Validation (CV) : **the training set is split into complementary subsets**, and each model is trained against a different combination of these subsets and validated against the remaining parts.

Once the model type and hyperparameters have been selected, a **final model is trained using these hyperparameters on the full training set**, and the generalized error is measured on the test set.

Testing and Validating : Cross-Validation

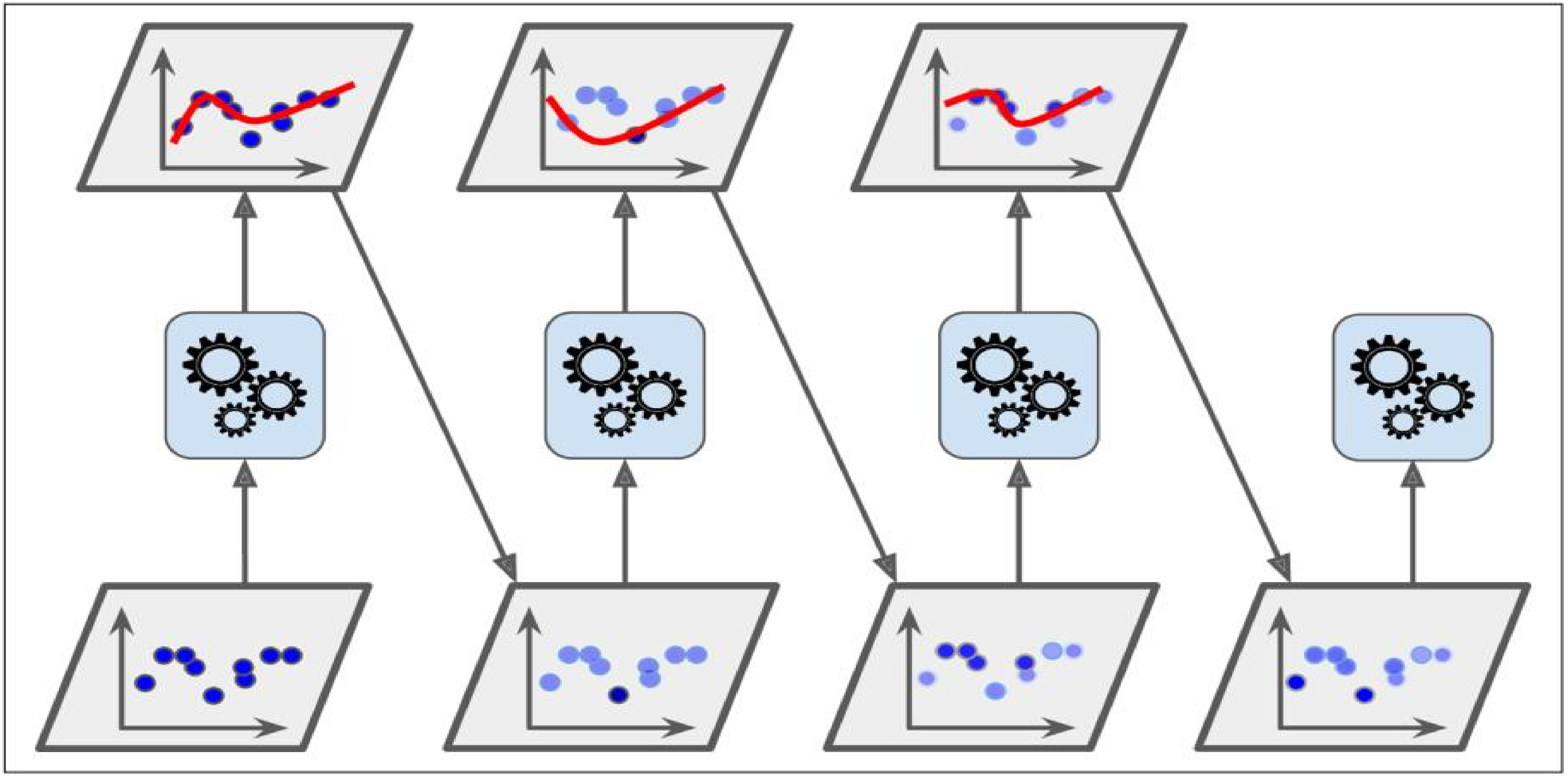


Boosting

Boosting refers to any Ensemble method that can combine **several weak learners into a strong learner**.

The general idea of most boosting methods is to train predictors sequentially, each trying to correct its predecessor. There are many boosting methods available, but by far the most popular are **AdaBoost** (Adaptive Boosting) and **Gradient Boosting**.

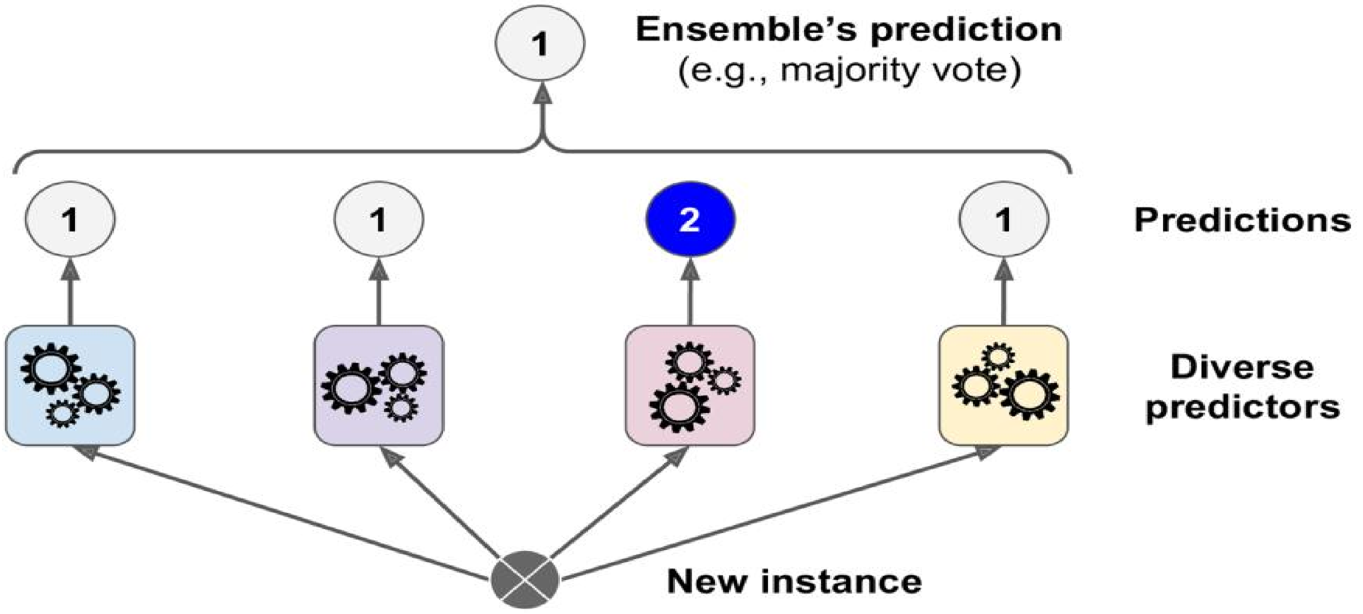
Boosting



AdaBoost sequential training with instance weight updates

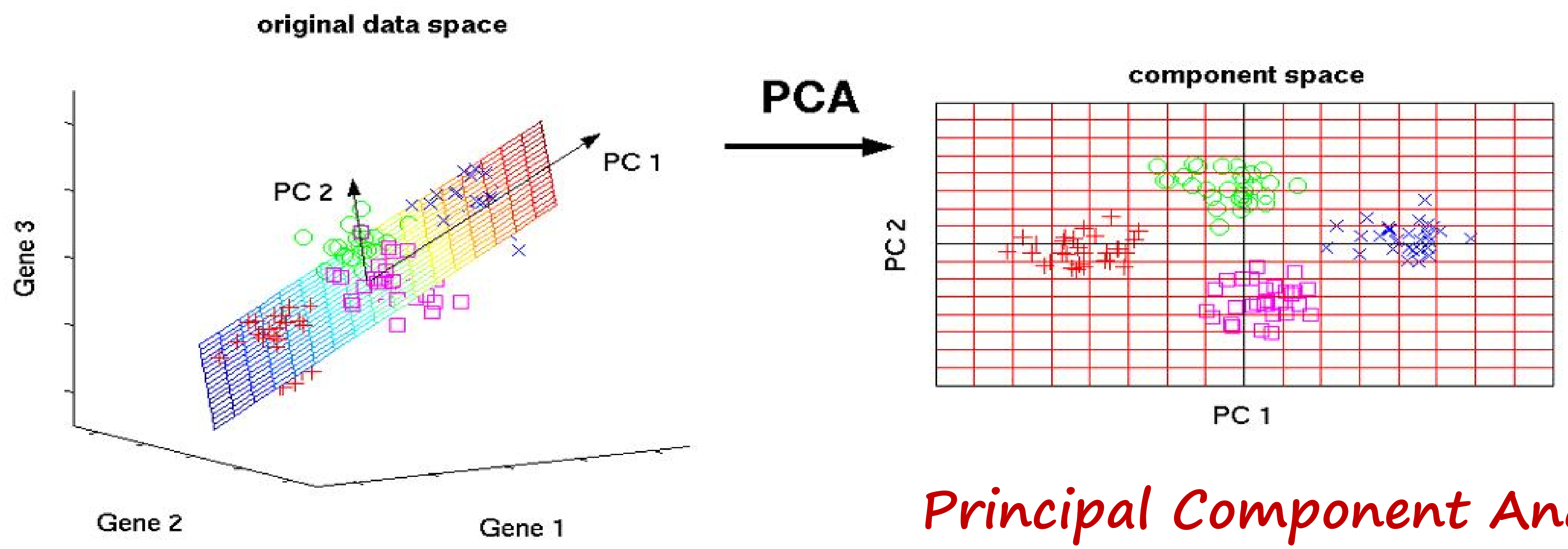
Voting Classifiers

The Voting Classifier: is a meta-classifier for combining similar or conceptually different machine learning classifiers for classification via majority or plurality voting. (For simplicity, we will refer to both majority and plurality voting as majority voting).



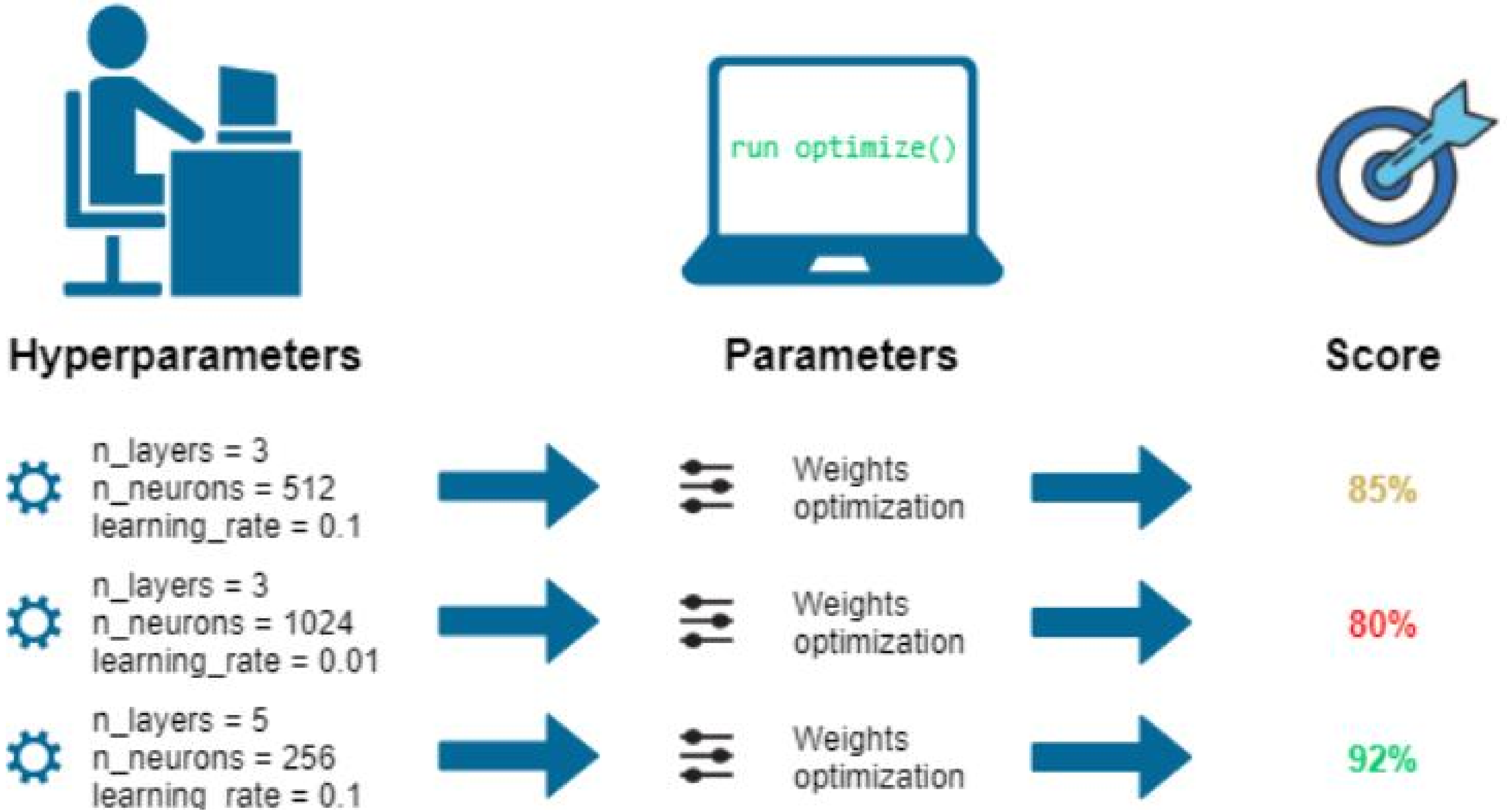
Dimensionality Reduction

Many Machine Learning problems involve thousands or even millions of features for each training instance. Not only does this make training extremely slow, but it can also make it much harder to find a good solution. This problem is often referred to as the curse of dimensionality.



Principal Component Analysis

Hyperparameter Tuning



Steps to Build a Machine Learning System

1. Data collection.
2. Improving data quality (data preprocessing: drop duplicate rows, handle missing values and outliers).
3. Feature engineering (feature extraction and selection, dimensionality reduction).
4. Splitting data into training (and evaluation) and testing sets.
5. Algorithm selection (Regression, Classification, Clustering ...).
6. Training.
7. Evaluation + Hyperparameter tuning.
8. Testing.
9. Deployment

Thank you for your attention

Hichem Felouat …