Reproductibilité des résultats et génie logiciel empirique

A tour on Machine Learning and Reproducibility

Paul Temple, Mathieu Acher & Olivier Barais

EJCP 2023

About me

- 2008 – 2013 IN (IUT Lannion + ESIR)
- 2015 – 2018 PhD (DiverSE, IRISA)
- 2019 – 2022 Post-doc (UNamur, Belgique)
- 2022 – ... :)

About me

- 2008 – 2013 IN (IUT Lannion + ESIR)
- 2015 – 2018 PhD (DiverSE, IRISA)
- 2019 – 2022 Post-doc (UNamur, Belgique)
- 2022 – ... :)

Research interests:

- Applying ML to Soft. Var.
- Applying Soft. Var. to ML
- Applying Soft. Test. to ML
- Sec. of ML

About me

- 2008 – 2013 IN (IUT Lannion + ESIR)
- 2015 – 2018 PhD (DiverSE, IRISA)
- 2019 – 2022 Post-doc (UNamur, Belgique)
- 2022 – ... :)

Research interests:

- Applying ML to Soft. Var.
- Applying Soft. Var. to ML
- Applying Soft. Test. to ML
- Sec. of ML

Research at DiverSE (`https://www.diverse-team.fr/`)

options:

- no-mbtree (T or F)
- nr ([100..1000])
- qblur ([0; 1])

$\Rightarrow$

step = 0.0001

$\rightarrow$ **18 millions** of configurations

options:

- no-mbtree (T or F)
- nr ([100..1000])
- qblur ([0; 1])

step $= 0.0001$

$\rightarrow$ **18 millions** of configurations

$\Rightarrow$

no-mbtree = T
qblur = 0.33
nr = 400

no-mbtree = F
qblur = 0.90
nr = 200

options:

- no-mbtree (T or F)
- nr ([100..1000])
- qblur ([0; 1])

step = 0.0001

→ **18 millions** of configurations

⇒

no-mbtree = T
qblur = 0.33
nr = 400

no-mbtree = F
qblur = 0.90
nr = 200

What happens when you want to use new values of parameters?

Performance prediction in software variability

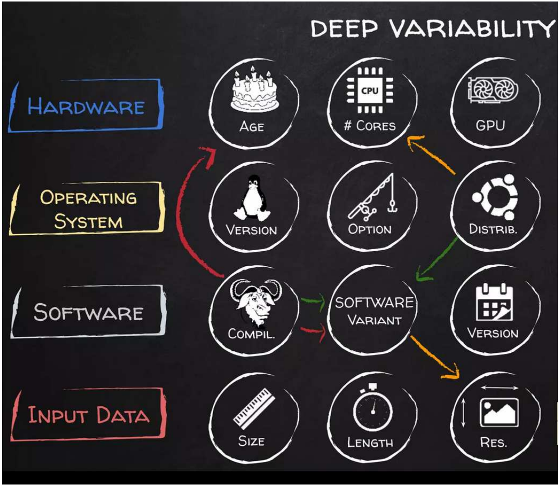
no-mbtree = F
qblur = 0.90
nr = 200

encoding time = 5 min

encoding time = 2 h

encoding time = 10 h

Performance prediction in software variability



Source:
Deep Software Variability: Towards Handling Cross-Layer Configuration
Luc LESOIL, Mathieu ACHER, Arnaud BLOUIN, Jean-Marc JEZEQUEL
https://doi.org/10.1145/3442391.3442402
https://hal.inria.fr/hal-03084276v2/file/vision_paper_deep_software_variability.pdf

Configurable systems? A few examples

Configurable systems? A few examples

Configurable systems? A few examples

Configurable systems? A few examples

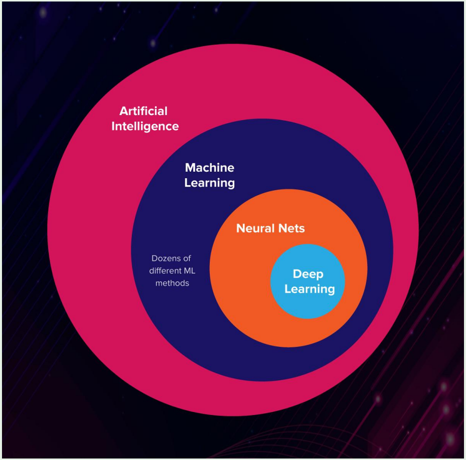
A quick introduction to Machine Learning

Back to basics

AI vs ML vs DL ???

Intro DL from P. Pérez (Valeo.ai)

AI vs ML vs DL ???



Source: `https://serokell.io/blog/ai-ml-dl-difference`

Intro DL from P. Pérez (Valeo.ai)

Back to basics

Quick formalization

- A set of data $\mathcal{X}$
- Data are described via d features $\rightarrow \mathcal{X} \in \mathcal{R}^d$
- A set of n target labels $\mathcal{Y}$

Back to basics

Quick formalization

- A set of data $\mathcal{X}$
- Data are described via d features $\rightarrow \mathcal{X} \in \mathcal{R}^d$
- A set of n target labels $\mathcal{Y}$

<u>Goal</u>: x $\rightarrow \mathcal{Y}$

$\mathcal{R}^d \rightarrow \mathcal{N}^n$ or $\mathcal{R}^d \rightarrow \mathcal{R}^n$

Back to basics

Quick formalization

- A set of data $\mathcal{X}$
- Data are described via d features $\rightarrow \mathcal{X} \in \mathcal{R}^d$
- A set of n target labels $\mathcal{Y}$

<u>Goal</u>: x $\rightarrow \mathcal{Y}$

$\mathcal{R}^d \rightarrow \mathcal{N}^n$ or $\mathcal{R}^d \rightarrow \mathcal{R}^n$

Main ML families:

Back to basics

Quick formalization

- A set of data $\mathcal{X}$
- Data are described via d features $\rightarrow \mathcal{X} \in \mathcal{R}^d$
- A set of n target labels $\mathcal{Y}$

Goal: x $\rightarrow \mathcal{Y}$

$$\mathcal{R}^d \rightarrow \mathcal{N}^n \text{ or } \mathcal{R}^d \rightarrow \mathcal{R}^n$$

Main ML families:

- Supervised
- Unsupervised
- Semi-supervised
- Reinforcement

Back to basics

Tasks:

Back to basics

Tasks:

- Classification
- Regression

Back to basics

Tasks:

- Classification
- Regression

Algorithm examples:

Back to basics

Tasks:

- Classification
- Regression

Algorithm examples:

- Regressors
- SVM
- Decision Trees
- k-nearest neighbors
- Boosting
- Neural Networks
- ...

The training process

The training process

Data preparation

- Data are gathered
- Data are sanitized

The training process

Data preparation

- Data are gathered
- Data are sanitized
- Features are engineered

The training process

Data preparation

- Data are gathered
- Data are sanitized
- Features are engineered

Model training

- Model is trained
- Model is validated

The training process

Data preparation

- Data are gathered
- Data are sanitized
- Features are engineered

Model training

- <u>Model is trained</u>
- Model is validated

Model exploitation

- Model is deployed

Model training

Feeding batches to the model

- Bunch at once
- Batches are representative
- Stratified sampling

sklearn user guide on sampling

Model training

Feeding batches to the model

- Bunch at once
- Batches are representative
- Stratified sampling

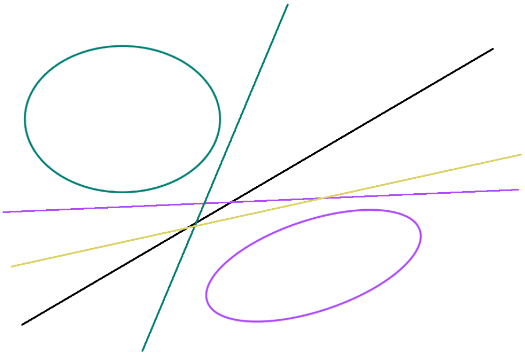
Optimization

- Solve the optimization problem
- Back-propagation

sklearn user guide on sampling

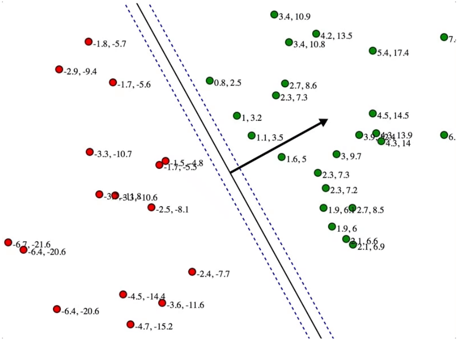
Example with SVMs

Which separation is the best?



Example with SVMs

How SVMs optimize



Source:`https://jeremykun.com/2017/06/05/formulating-the-support-vector-machine-optimization-problem/`

Model evaluation

Confusion Matrix

		Actual	
		Class +1	Class -1
Predicted	Class +1	TP	FN
	Class -1	FP	TN

Model evaluation

Confusion Matrix

		Actual	
		Class +1	Class -1
Predicted	Class +1	TP	FN
	Class -1	FP	TN

Metrics

- Accuracy
- Precision // Recall
- F1-score

Model evaluation

Confusion Matrix

		Actual	
		Class +1	Class -1
Predicted	Class +1	TP	FN
	Class -1	FP	TN

Metrics

- Accuracy
- Precision // Recall
- F1-score
- ...

Time to train models!

Retrieve results

4. Experimental Approach and Results

The primary objective of this study was to investigate the use of Support Vector Machine (SVM) and decision tree classifiers for the detection of cardiac issues. The focus of our approach was to establish a robust model capable of predicting cardiac conditions with a target accuracy that surpasses the current state of the art of 62 %.

4.1 Data Acquisition and Pre-processing

The dataset we used is the well known *heart disease* dataset which describes a number of 300 patients with 14 indicators including age and sex but also cholesterol level, blood pressure among otheres. Following the acquisition, we conducted rigorous data cleaning, addressed missing values, and normalized the data to ensure compatibility with the SVM and decision tree classifiers.

4.2 Model Implementation and Parameter Tuning

The SVM and decision tree classifiers were built and meticulously tuned to optimize their performance. This included searching for optimal parameters values such as the cost and gamma for the SVM, and maximum depth and minimum samples split for the decision tree. Cross-validation techniques were used throughout this process to mitigate overfitting and validate the performance of our models.

4.3 Results

After an extensive period of training and evaluation, the SVM classifier, with optimized parameter yielded the best results. Although the decision tree classifier demonstrated satisfactory performanc it was slightly outperformed by the SVM.

Remarkably, our SVM model achieved an approximate accuracy rate of 80%. This significant finding suggests that SVM, when correctly parameterized and cross-validated, can indeed serve as powerful tool for the prediction of cardiac conditions.

While the robustness of our model on this dataset is encouraging, we acknowledge that the results' generalizability needs to be further evaluated on diverse datasets and in different medical scenarios

In conclusion, this pioneering study provides a foundation for future research in leveraging machin learning techniques such as SVM and decision tree classifiers for healthcare applications, particularly in predicting cardiac conditions. Our research underscores the potential of these model to revolutionize medical diagnosis, and lays groundwork for more comprehensive and reliable AI-driven solutions in healthcare.

Try to reach 80% accuracy!

Retrieve results

- Did you succeed?

Retrieve results

- Did you succeed?
- What did you miss?

Retrieve results

- Did you succeed?
- What did you miss?
- Did you try any strategy?

Retrieve results

- Did you succeed?
- What did you miss?
- Did you try any strategy?
- Did you try different algorithms?

Optimize results

4. Experimental Approach and Results

The primary objective of this study was to investigate the use of Support Vector Machine (SVM) and decision tree classifiers for the detection of cardiac issues. The focus of our approach was to establish a robust model capable of predicting cardiac conditions with a target accuracy that surpasses the current state of the art of 62 %.

4.1 Data Acquisition and Pre-processing

The dataset we used is the well known *heart disease* dataset which describes a number of 300 patients with 14 indicators including age and sex but also cholesterol level, blood pressure among otheres. Following the acquisition, we conducted rigorous data cleaning, addressed missing values, and normalized the data to ensure compatibility with the SVM and decision tree classifiers.

4.2 Model Implementation and Parameter Tuning

The SVM and decision tree classifiers were built and meticulously tuned to optimize their performance. This included searching for optimal parameters values such as the cost and gamma for the SVM, and maximum depth and minimum samples split for the decision tree. Cross-validation techniques were used throughout this process to mitigate overfitting and validate the performance of our models.

4.3 Results

After an extensive period of training and evaluation, the SVM classifier, with optimized parameter yielded the best results. Although the decision tree classifier demonstrated satisfactory performanc it was slightly outperformed by the SVM.

Remarkably, our SVM model achieved an approximate accuracy rate of 80%. This significant finding suggests that SVM, when correctly parameterized and cross-validated, can indeed serve as powerful tool for the prediction of cardiac conditions.

While the robustness of our model on this dataset is encouraging, we acknowledge that the results' generalizability needs to be further evaluated on diverse datasets and in different medical scenario

In conclusion, this pioneering study provides a foundation for future research in leveraging machin learning techniques such as SVM and decision tree classifiers for healthcare applications, particularly in predicting cardiac conditions. Our research underscores the potential of these model to revolutionize medical diagnosis, and lays groundwork for more comprehensive and reliable AI-driven solutions in healthcare.

What's the best accuracy measure you can find?

Optimize results

- What is the best result?

Optimize results

- What is the best result?
- Did you try any strategy?

Optimize results

- What is the best result?
- Did you try any strategy?
- Do you think it can be reproduced?

Coming back to missing details

Hyperparameters

- Usually comes first to mind
- Communicate values?

Coming back to missing details

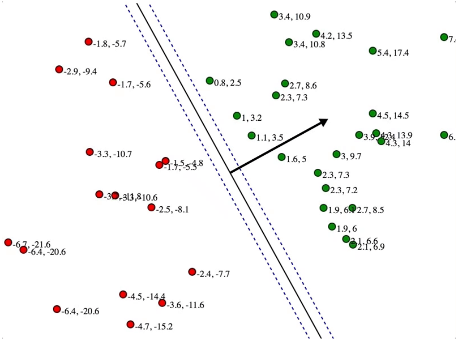
Hyperparameters

- Usually comes first to mind
- Communicate values?

Data

- Shuffled again and again
- Order of presentation can affect performances

Coming back to missing details



Optimizing is complex

Random everywhere

- Random separation between training and test sets
- Random in data presentation
- Random initialization of (hyper)parameters/weights
- ...

Optimizing is complex

Random everywhere

- Random separation between training and test sets
- Random in data presentation
- Random initialization of (hyper)parameters/weights
- ...

Why so many?

Optimizing is complex

Random everywhere

- Random separation between training and test sets
- Random in data presentation
- Random initialization of (hyper)parameters/weights
- ...

Why so many?

For repeating...

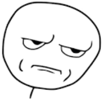
Optimizing is complex

Random everywhere

- Random separation between training and test sets
- Random in data presentation
- Random initialization of (hyper)parameters/weights
- ...

Why so many?

For repeating...



Consequences of random everywhere

Reporting results is hard

- Everything related to random $\rightarrow$ unstable

Consequences of random everywhere

Reporting results is hard

- Everything related to random → unstable
- Gain stability and confidence → repeat

Consequences of random everywhere

Reporting results is hard

- Everything related to random → unstable
- Gain stability and confidence → repeat
- Accounting for (un)stability → average is not sufficient

Consequences of random everywhere

- Reporting results is hard

The role of seeds

- Can be fixed
  - Favor reproducibility
  - Limit flakiness when testing

To seed or not to seed?

Consequences of random everywhere

- Reporting results is hard

The role of seeds

- Can be fixed
  - Favor reproducibility
  - Limit flakiness when testing
- Not a good practice for model deployment (replicability)
  - Lower confidence in generalization
  - Probably not optimal for new datasets

To seed or not to seed?

Consequences of random everywhere

- Reporting results is hard

The role of seeds

- Can be fixed
  - Favor reproducibility
  - Limit flakiness when testing
- Not a good practice for model deployment (replicability)
  - Lower confidence in generalization
  - Probably not optimal for new datasets
- Ok for model testing
  - Fix seeds for every run
  - Log seeds
  - Report seeds and results

To seed or not to seed?

Consequences of random everywhere

- Reporting results is hard
- Seeds can be fixed

Not always harmful

- Taxonomy of use of randoms

Consequences of random everywhere

- Reporting results is hard
- Seeds can be fixed

Not always harmful

- Taxonomy of use of randoms
- Ok for → model selection, ensemble creation, and sensitivity analysis
- Avoid for → reproducibility (does not help with GPU under TensorFlow), performance comparison, optimizing performances

We need to talk about random seeds

ML processes are random by nature

- Different libraries exist

ML processes are random by nature

- Different libraries exist
- Implement different techniques
- Default values may not be the same

ML processes are random by nature

- Different libraries exist
- Implement different techniques
- Default values may not be the same

$\Rightarrow$ Harm reproducibility

$\rightarrow$ Need to improve communications of results

Sum-up

- ML is random by nature

Sum-up

- ML is random by nature
- Seeds can be used (but with care)

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability
- Need to think about what elements are needed for reproducibility
- Do not forget Deep Software Variability

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability
- Need to think about what elements are needed for reproducibility
- Do not forget Deep Software Variability

Sorry... No magic solution for today

But...

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability
- Need to think about what elements are needed for reproducibility
- Do not forget Deep Software Variability

Sorry... No magic solution for today

But...

- Some papers are there

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability
- Need to think about what elements are needed for reproducibility
- Do not forget Deep Software Variability

Sorry... No magic solution for today

But...

- Some papers are there
- You can think about it and help building the litterature =)

Sum-up

- ML is random by nature
- Seeds can be used (but with care)
- Repeat to account for (un)stability
- Need to think about what elements are needed for reproducibility
- Do not forget Deep Software Variability

Sorry... No magic solution for today

But...

- Some papers are there
- You can think about it and help building the litterature =)
- You can force colleagues to be careful (e.g., reviewing)