# Practical introduction to machine learning

Part 4 : Validation and interpretation

Rémi Flamary - CMAP, École Polytechnique

Master Data Science, Institut Polytechnique de Paris

October 11, 2023





# **Overview of MAP654I**

### 1. Data and Machine Learning problems

- Data properties and visualization
- Pre-processing
- Finding your Machine Learning problem

### 2. Unsupervised learning

- Clustering
- Density estimation and generative modeling
- Dictionary learning and collaborative filtering
- Dimensionality reduction and manifold learning

### 3. Supervised learning

- Bayesian decision and Nearest neighbors
- Linear models nonlinear methods for regression and classification
- Trees, forest and ensemble methods

### 4. Validation and interpretation

- Performance measures
- Models and parameter selection (validation)
- Interpretation of the methods

# Overview for the current part

Introduction	2
Performance measure	5
Unsupervised learning	6
Regression	7
Classification	8
Validation/Model selection	10
Spliting the data	10
50 types of validation	11
Validation with Scikit-learn	12
Interpretation of the model	14
Feature importance	15
Model explanation	16
Adversarial attack	17
Algorithmic bias	19
Conclusion	20

## Machine Learning in practice



### Selecting the model

- For a given ML problem several kinds of method can be applied.
- Even for a given method several parameters can greatly change its performance.
- Selecting the "best" method/parameters is called model selection or validation.
- An important question is which **performance measure** to use.

### Understanding the model

- Interpret the perfomance, identifying bad predictions, detect bias in the predictor/data.
- Most important variables for the model.
- Explaining a given prediction (what lead to this prediction).
- Robustness to noise, to adversarial attacks.

# Section

### Introduction

### Performance measure

Unsupervised learning Regression Classification

### Validation/Model selection

Spliting the data 50 types of validation Validation with Scikit-learn

#### Interpretation of the model

Feature importance Model explanation Adversarial attack Algorithmic bias

### Conclusion

## **Performance measure**

### **Unsupervised learning**

- Clustering
  - Supervised (actual clusters ar known, perf\_measure(y\_true,y\_pred)).
  - Unsupervised (actual clusters unknown, perf\_measure(X,y\_pred))).
- Dimensionality reduction performance is often the objective of the optimization problem (same as regression performance for invertible methods).

### Supervised learning

- Classification (default is accuracy/0-1 loss).
  - How accurate is the class prediction.
  - How separable are the classes in the score function space.
- Regression (prediction error)
  - Average prediction error.
  - Correlation (focus on the dynamic).

### Performance measures

- Performance measures provided below are functions of sklearn.metrics.
- Measures with  $\uparrow$  are better with large values and  $\downarrow$  with low or negative values.

### Warning

Always evaluate performance on data that was not used to train the model for supervised learning (also sometimes on unsupervised).

# **Clustering performance**

### Silhouette score (Rousseeuw, 1987)

- Score is the average of (b − a)/max(a, b) when a is the distance to the cluster and b the distance to the closest other cluster.
- Non-supervised measure between -1 (worst) and 1 (best).

### Rand Index $\uparrow$ , rand\_score [Rand, 1971]

- Ratio of samples belonging in the same clusters in the predicted and true clustering (similar to accuracy but invariant to class permutation).
- ▶ Supervised measure between 0 (worst) and 1 (perfect).
- ► Adjusted Rand Index ajusted\_rand\_score has score 0 when random prediction.

### Mutual Information $\uparrow$ , mutual\_info\_score [Vinh et al., 2010]

- Measure of the mutual information between the true and predicted clustering.
- Supervised measure  $\geq 0$  (where 0 is worst).
- Adjusted Mutual Information adjusted\_mutual\_info\_score has score 0 when random prediction and 1 when perfect.

### **Regression performances**

### Mean Square Error (MSE) ↓, mean\_squared\_error

- $MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i} (y_i \hat{y}_i)^2$ , classical convex and smooth loss in regression, can be used as performance measure on new data.
- Can be normalized by the mean square of the true labels (which computes the Signal to Noise Ratio, SNR).

#### Pearson Correlation Coefficient $\uparrow$ , np.corrcoef

- ▶  $r = cov(\mathbf{y}, \hat{\mathbf{y}}) / \sqrt{cov(\mathbf{y}, \mathbf{y})cov(\hat{\mathbf{y}}, \hat{\mathbf{y}})}$ , between -1 and 1 (random pred. is 0).
- Measure of linearity between the true and predicted labels (invariant to scaling).

### $R^2$ coefficient of determination $\uparrow$ , r2\_score

- ▶  $R^2 = 1 MSE(\mathbf{y}, \hat{\mathbf{y}}) / MSE(\mathbf{y}, \bar{\mathbf{y}})$  where  $\bar{\mathbf{y}}$  contains the mean of  $\mathbf{y}$ .
- ▶ 1 when perfect prediction, 0 when random prediction (can be negative).

Mean/Median absolute error  $\downarrow$ , mean\_absolute\_error,median\_absolute\_error

• 
$$MeanAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n} \sum_{i} |y_i - \hat{y}_i|$$

More robust to outliers in the data but non-smooth (harder to optimize).

# **Classification performances**

### Confusion matrix, confusion\_matrix

- Matrix C that counts for C<sub>i,j</sub> the number of samples that are from the true class i and are predicted as class j.
- For binary classification we have
  - ► C<sub>0,0</sub> True Negative (TN) and C<sub>1,1</sub> True Positive (TP).
  - C<sub>1,0</sub> False Negative (FN) and C<sub>0,1</sub> False Positive (FP).
- Used for many performance measures.

### Accuracy $\uparrow$ , accuracy\_score

- ▶ Ratio of correctly classified samples  $\frac{TP+TN}{n} = \frac{1}{n} \sum_{k} C_{k,k}$ .
- ▶ Balanced accuracy  $\frac{1}{p} \sum_{k} \frac{C_{k,k}}{\sum_{l} C_{l,k}}$  better when unbalanced classes.

### Area Under the Receiver Operating Curve (ROC) curve \, roc\_auc\_score

- Compute the Area under the curve plotting  $TPR = \frac{TP}{TP+FN}$  as a function of  $FPR = \frac{FP}{FP+TN}$  when varying the threshold on the score function.
- Estimates for binary classification the probability for a positive sample to have a larger score than a negative sample (measure of separability in the score space).



# Interpreting the performance

### Performance measure

- A performance measure even when computed on test data is a 1D (partial) measure of the quality of a model.
- Know the side effect of the performance measure (e.g. MSE is very sensitive to outliers, Pearson-s correlation coeff is invariant to scaling).
- Always compute other performance measures and compare them on a given model/data.

#### Interpreting the prediction

- Visualize the predictions (confusion matrix, scatterplot for regression).
- Search for bias in the predictions (some groups always badly predicted?).
- Look at mispredicted samples (bad label or systematic error).

### Warning

Always be careful what you wish for (in terms of performances). Optimizing a given criterion can/will have unintended effect.

# Section

### Introduction

### Performance measure

Unsupervised learning Regression Classification

### Validation/Model selection

Spliting the data 50 types of validation Validation with Scikit-learn

#### Interpretation of the model

Feature importance Model explanation Adversarial attack Algorithmic bias

### Conclusion

12

# Splitting the data



### Principle of Hold-Out cross-validation

- Split the training data in a training and validation sets (non overlapping).
- Train different models (different methods and/or parameters) on the train data.
- Evaluate performance on the validation data and select the method/parameters with best performance.

### **Final estimator**

- The validation is a method of selection for the method/parameters not the estimator.
- After selecting the optimal parameters, one should retrain the estimator on the whole training dataset using the optimal method/parameters.
- For methods that can have a large variability (neural network) the best classifier on validation set is often kept (also used for early stopping).

# Different ways to split the data



### Data splitting for cross-validation [Arlot and Celisse, 2010]

The training data is split in non-overlapping training/validation sets.

▶ Hold-Out uses a unique split and computes the performance on the validation set.

- More robust cross-validation approaches actually investigate several splits of the data and compute the average performance:
  - K-fold (split in K sets and use one split as test for all k)
  - Random sampling (aka **Shuffle split**) draws several random splittings.
  - Leave one out bootstrap draws training samples with replacement.
- Scikit-learn implementation : sklearn.model\_selection.cross\_validate

# Data splitting with Scikit-learn



- Scikit-learn implements iterator classes for data split in sklearn.model\_selection.
- KFold is the classical K-fold cross-validation.
- StratifiedKFold ensures a data split that preserves the proportion of classes.
- ShuffleSplit randomly selects a proportion of the samples for train/validation.
- TimeSeriesSplit preserves the temporal sequences and ensures that the validation data is in the future (see practical session 2).

Source :

https://scikit-learn.org/stable/auto\_examples/model\_selection/plot\_cv\_indices.html

### Validation with Scikit-learn



### Principle

- GridSearchCV takes a model and a grid of parameters as input and performs cross-validation.
- Both the best estimator (retrained on the whole data) and the best parameters can be recovered.
- Number of splits and type of data splitting can be chosen.
- For large number of parameters complexity is exponential, RandomizedSearchCV can be more efficient.



### Python code

```
from sklearn.svm import SVC
2 from sklearn.model_selection import
        GridSearchCV
3
  ngrid=21
4
  clf = SVC()
  param_grid={'C':np.logspace(-2,2,ngrid),
           'gamma':np.logspace(-2,2,ngrid),}
8
  cv = GridSearchCV(clf,param_grid)
9
  cv.fit(xn,y)
  # recover best parameters and estimators
14
  clf_opt = cv.best_estimator_
15 params_opt = cv.best_params_
```

# Section

### Introduction

### **Performance measure**

Unsupervised learning Regression Classification

### Validation/Model selection

Spliting the data 50 types of validation Validation with Scikit-learn

### Interpretation of the model

Feature importance Model explanation Adversarial attack Algorithmic bias

### Conclusion

14

# Interpretation of the model and data

### ML interpretation and model explainability [Molnar, 2020]

- Important question of understanding the model and the data.
- Interpretation: how does the model work?
- Explainability: why did it predict this?
- GDPR brought the "right to explanation" in European countries.

### Linear models

- Linear models are the simplest models and the importance of each variable is provided in the weights.
- Remember to standardize the data before interpretation because the weights depend on the scaling of the variables.
- Example in Scikit-learn documentation : https://scikit-learn.org/stable/ auto\_examples/inspection/plot\_linear\_model\_coefficient\_interpretation.html

### Feature selection [Guyon and Elisseeff, 2003]

- Can be seen as both pre-processing and promotion of interpretability.
- Can be done simultaneously with model estimation with linear models (Lasso).
- Wrapper methods perform a validation over the subset of variables (forward/backward methods add/remove variables one by one).

### Feature permutation importance



### Principle [Breiman, 2001]

- Computed by doing a random permutation for one feature (permute one column).
- The loss/gain of performance is computed on a held-out data and is a measure of the importance of this variable.
- Mean Decrease in Impurity (MDI) is an alternative for random forests.
- Correlated features will all be "important" even when non necessary.
- Computational complexity is high on high dimensional data.
- Scikit-learn : sklearn.inspection.permutation\_importance

# Local Interpretable Model-agnostic Explanations (LIME)



### Principle [Ribeiro et al., 2016]

- The image to interpret is segmented in homogeneous super-pixels.
- Generate perturbed samples where only some super-pixels contain the image information the other being replaced by their average value.
- Estimate weights for the perturbed samples with a kernel.
- Perform Ridge regression trying to predict the output of model f on the perturbed samples from the binary activation of the super-pixels: the weights give the importance of the superpixels in the decision.
- ▶ When LS is used instead of Ridge we recover SHAP [Lundberg and Lee, 2017].
- Python implementation in ELI5: https://eli5.readthedocs.io/

### **Adversarial attacks**



### Principle [Goodfellow et al., 2014]

- A model that generalizes should be robust to small perturbation of the samples.
- Adversarial attacks search for samples x̃ = x + p close to the true sample x of label y that maximize the change in the prediction of the model f :

$$\max_{\tilde{\mathbf{x}}, \|\mathbf{x} - \tilde{\mathbf{x}}\| \le \epsilon} L(y, f(\tilde{\mathbf{x}})), \quad \text{or,} \quad \max_{\tilde{\mathbf{x}}, \|\mathbf{x} - \tilde{\mathbf{x}}\| \le \epsilon} L(f(\mathbf{x}), f(\tilde{\mathbf{x}})) \tag{1}$$

- Virtual Adversary (right) does not require the true label [Miyato et al., 2018].
- Adversarial examples can be used for manipulating the output of a model [Brown et al., 2017], for evaluating its robustness and for regularization.
- Python implementation : https://adversarial-robustness-toolbox.readthedocs.io/

# Interpretability and explainability

### Main approaches

- Linear models and sparsity (Lasso)
- Global agnostic models
  - Partial Dependence Plot [Goldstein et al., 2015]
  - Feature permutation importance [Breiman, 2001]
- Local approximation (smooth)
  - Linear local approximation [Erhan et al., 2009, Shrikumar et al., 2017].
  - Integrated gradients [Sundararajan et al., 2017]
- Local model agnostic methods
  - Game theory: Shapley [Strumbelj and Kononenko, 2014], SHAP [Lundberg and Lee, 2017]
  - LIME [Ribeiro et al., 2016]
- By design of the model (attention mechanism [Vaswani et al., 2017])

### References

- Free book [Molnar, 2020]: https://christophm.github.io/interpretable-ml-book/
- Recent tutorial: https://explainml-tutorial.github.io/

### Python toolboxes

- Interpretability: https://eli5.readthedocs.io/
- Adversarial robustness: https://adversarial-robustness-toolbox.readthedocs.io/

# **Bias and Fairness in Machine Learning**

### Exemple : COMPAS score in Florida, study ProPublica



# **Machine Bias**

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

- Proprietary algorithm predicting recidivists.
- Used by judges to decide where a person stays in prison until trial.
- ► Overall Accuracy : 66%
- Classified "risky", not recidivist:
  - 23.5% of whites.
  - ▶ 44.5% of afro-americans.
- Classified "not risky", but recidivists:
  - 47.7% of whites.
  - ▶ 28.0% of afro-americans.
- Student project:
  - 65.8% accuracy, same bias with linear model.
  - 62% accuracy but perfect fairness (same rates for both groups) for corrected data.

### Algorithmic fairness (Survey [Mehrabi et al., 2021])

- Bias in the data will be learned by the model.
- Post processing or data repair methods (https://fairlearn.org/).
- There is a balance between fairness and performance.

# Section

Im	÷×	d			ы.	n
	u	u	u	L	u	

#### **Performance measure**

Unsupervised learning Regression Classification

#### Validation/Model selection

Spliting the data 50 types of validation Validation with Scikit-learn

#### Interpretation of the model

Feature importance Model explanation Adversarial attack Algorithmic bias

### Conclusion

19/24

20

# Conclusion



#### Last words

- Know the data (visualize it, talk with experts, pre-process it, check for bias).
- Know the problem (unsupervised, supervised, final goal).
- Know the methods (linear/nonlinear, trees, neural networks).
- ▶ Validate the methods and parameters (performance measure, cross-validation).
- ▶ Be critical and curious with the model (interpretation, explainability, adversaries).

### References

- Elements of statistical learning (free PDF online) [Friedman et al., 2001].
- Pattern recognition and machine learning [Bishop Christopher et al., 2006].
- Machine learning: a probabilistic perspective [Murphy, 2012].

## **References I**

```
[Arlot and Celisse, 2010] Arlot, S. and Celisse, A. (2010).
A survey of cross-validation procedures for model selection.
Statistics surveys, 4:40–79.
```

[Bishop Christopher et al., 2006] Bishop Christopher, M. et al. (2006). Pattern recognition and machine learning. Information science and statisticsNew York: Springer.

```
[Breiman, 2001] Breiman, L. (2001).
Random forests.
Machine learning, 45(1):5–32.
```

[Brown et al., 2017] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. arXiv preprint arXiv:1712.09665.

[Erhan et al., 2009] Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. University of Montreal, 1341(3):1.

[Friedman et al., 2001] Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). The elements of statistical learning, volume 1. Springer series in statistics New York.

# **References II**

[Goldstein et al., 2015] Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation.

journal of Computational and Graphical Statistics, 24(1):44-65.

[Goodfellow et al., 2014] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014).

Explaining and harnessing adversarial examples.

arXiv preprint arXiv:1412.6572.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Journal of machine learning research, 5(Mar):1157–1162.

[Lundberg and Lee, 2017] Lundberg, S. M. and Lee, S.-I. (2017).

A unified approach to interpreting model predictions.

In Proceedings of the 31st international conference on neural information processing systems, pages 4768–4777.

[Mehrabi et al., 2021] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM computing surveys (CSUR), 54(6):1–35.

[Miyato et al., 2018] Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

## **References III**

[Molnar, 2020] Molnar, C. (2020). Interpretable machine learning. Lulu. com.

[Murphy, 2012] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

 [Rand, 1971] Rand, W. M. (1971).
 Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66(336):846–850.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016).
" why should i trust you?" explaining the predictions of any classifier.
In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[Shrikumar et al., 2017] Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In International Conference on Machine Learning, pages 3145–3153. PMLR.

# **References IV**

[Strumbelj and Kononenko, 2014] Strumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

[Sundararajan et al., 2017] Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks.

In International Conference on Machine Learning, pages 3319-3328. PMLR.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017).

Attention is all you need.

In Advances in neural information processing systems, pages 5998-6008.

[Vinh et al., 2010] Vinh, N. X., Epps, J., and Bailey, J. (2010).

Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance.

The Journal of Machine Learning Research, 11:2837-2854.