

Neural Networks and Deep Learning

Lecture 11: Explainable AI: SHAP, LIME, gradient methods and the stability of explanations

Prof. dr. Daniel Traian PELE

Bucharest University of Economic Studies (ASE), Department of Statistics and Econometrics
Romanian Academy, Institute for Economic Forecasting (IPE)
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Learning objectives

- Explain why banks, insurers and regulators require explanations of a neural network, and classify explanation methods of explainable artificial intelligence (XAI).
- Compute Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME) with full formulas, and read them for one customer and for a portfolio.
- Use gradient methods (saliency, integrated gradients, SmoothGrad) and check their axioms on a real fraud model.
- Test whether explanations agree, are faithful to the model and stay stable across seeds, samples and time.
- Write a counterfactual explanation that a rejected applicant can act on.

Abbreviations

AI	artificial intelligence
AOPC	area over the perturbation curve
ASE	Bucharest University of Economic Studies
AUC	area under the ROC curve
EBA	European Banking Authority
EU	European Union
EUR	euro
GDPR	General Data Protection Regulation
GNN	graph neural network
HMEQ	home-equity loan data set
ICE	individual conditional expectation
IG	integrated gradients
IPE	Institute for Economic Forecasting, Romanian Academy
JS	Jensen-Shannon divergence
KL	Kullback-Leibler divergence
LIME	local interpretable model-agnostic explanations
MAD	median absolute deviation
MLP	multilayer perceptron
PCA	principal component analysis
PD	probability of default
PDP	partial dependence plot
PICBE	Proceedings of the International Conference on Business Excellence
PR-AUC	area under the precision-recall curve
ReLU	rectified linear unit
ROC	receiver operating characteristic
SHAP	Shapley additive explanations
SR	Supervisory Guidance on Model Risk Management (Federal Reserve SR 11-7)
TCAV	testing with concept activation vectors
ULB	Université Libre de Bruxelles
USD	US dollar
XAI	explainable artificial intelligence
XGBoost	extreme gradient boosting

Real data in this lecture

Case	Data	Models and split
Credit	Home-equity loan data set (HMEQ), 5 960 loans, 19.9 % defaults (Baesens et al., 2016)	multilayer perceptron (MLP), logistic regression, extreme gradient boosting (XGBoost); 60/20/20 stratified
Fraud	Credit-card transactions of the Université Libre de Bruxelles (ULB) data set, two days of September 2013; 281 + 211 frauds in the sample (Dal Pozzolo et al., 2015)	MLP and logistic regression; train on day 1, test on day 2
Housing	California housing, 20 640 block groups of the 1990 census	MLP regression (partial dependence)

- Every number is computed by analysis_L11.py
  - Seeds from 20261108
  - Explanations computed on an Apple M5 Max processor.
- Fraud variables V1–V28 are principal component analysis (PCA) components released by the data owner: explanations name components, not raw fields.

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

Regulators require explanations of automated decisions

- **European Union (EU) Artificial Intelligence (AI) Act** (Regulation 2024/1689): creditworthiness scoring of natural persons is high-risk (Annex III)
  - Transparency to deployers (art. 13) and a right to an explanation of individual decisions (art. 86).
- **General Data Protection Regulation (GDPR)**, art. 22 and recital 71: safeguards against decisions based solely on automated processing.
- **European Banking Authority (EBA)**, guidelines on loan origination and monitoring (2020): models used for credit decisions must be explainable.
- **Model risk**: the Federal Reserve Supervisory Guidance on Model Risk Management (SR 11-7) asks for conceptual soundness, independent validation and ongoing monitoring.
- **Business reasons**: finding leakage, trust of credit officers, explaining a rejection.
- **But**: an explanation is a model of the model, and it can be wrong.

- **In practice:** an explanation that changes when nothing important changed is a compliance risk, not a safeguard.

(Regulation (EU) 2024/1689; SR 11-7)

A taxonomy of explanation methods

Method	Scope	Needs	This lecture
Coefficients, scorecards	global, intrinsic	a linear model	logistic benchmark
SHAP	local $\rightarrow$ global	model calls (Kernel); trees (TreeSHAP)	credit MLP, XGBoost
LIME	local	model calls	same applicant
Saliency, integrated gradients (IG), SmoothGrad	local	gradients	fraud MLP
Partial dependence plot (PDP), individual conditional expectation (ICE)	global effect	model calls	housing MLP
Counterfactuals	local, actionable	gradients or search	rejected applicant
Permutation importance	global	labels + model calls	fraud MLP

- Post-hoc methods explain a trained black box
  - Interpretable models are transparent by design (Rudin, 2019).

(Rudin, 2019)

Credit case study: comparison of three scoring models

- **In one sentence:** the network ranks defaulters better than the logistic scorecard (higher area under the receiver operating characteristic (ROC) curve (AUC) and area under the precision-recall curve (PR-AUC)); can we explain it as well?

$$\mathbf{h}_1 = \mathrm{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad \mathbf{h}_2 = \mathrm{ReLU}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$
$$f(\mathbf{x}) = \mathbf{w}_3^\top \mathbf{h}_2 + b_3, \qquad \mathrm{PD}(\mathbf{x}) = \sigma\big(f(\mathbf{x})\big)$$

Model (test)	AUC	PR-AUC	Brier
Logistic	0.909	0.785	0.080
MLP, 10 seeds	0.935	0.842	0.068
XGBoost, 10	0.959	0.889	0.058

- Base rate (PR-AUC of a random score) 0.200.

- **where**
  - $\mathbf{x}$ = 18 standardised features, median-imputed, with a flag for a missing debt-to-income ratio (DEBTINC)
  - f = log-odds of default
  - 64 and 32 rectified linear units (ReLU), dropout 0.1
  - PD = probability of default (PD).

- **In practice:** the logistic coefficients are the explanation
  - For the MLP the explanation must be computed.

(Baesens, Roesch & Scheule, 2016; own computation)

Experiment yourself: notebook §1

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

Shapley values: a fair split of the prediction among features

- **In one sentence:** each feature receives its average marginal contribution over all orders in which features could join the prediction.

$$\phi_j = \sum_{S \subseteq \{1,\dots,d\} \setminus \{j\}} \frac{|S|!\,(d - |S| - 1)!}{d!} \Big[v(S \cup \{j\}) - v(S) \Big]$$

- **where**
  - d = number of features
  - $v(S)$ = value of the coalition S (what the model predicts when only the features in S are known)
  - The weight is the share of orderings in which j joins right after S.

- **The four axioms** (Shapley, 1953) that make ϕ unique
  - Efficiency: $\sum_j \phi_j = v(\{1..d\}) - v(\emptyset)$
  - Symmetry: equal contributions, equal shares
  - Dummy: a feature that never changes v gets 0
  - Additivity: $\phi(v_1 + v_2) = \phi(v_1) + \phi(v_2)$
- **Cost**: 2^d coalitions
  - With 18 features, 2^{18} model values per applicant: we need approximations.

- **In practice:** efficiency is what a credit officer needs: the parts add up exactly to the score.

(Shapley, 1953)

SHAP: the value of a coalition of features

- **In one sentence:** SHAP sets $v(S)$ = expected model output when the features outside S are drawn from a background sample.

$$v(S) = \mathbb{E}_{\mathbf{x}_{\bar{S}} \sim \mathcal{B}}\left[f(\mathbf{x}_S, \mathbf{x}_{\bar{S}})\right]$$

$$f(\mathbf{x}) = \phi_0 + \sum_{j=1}^{d} \phi_j(\mathbf{x}), \qquad \phi_0 = \mathbb{E}_{\mathcal{B}}[f]$$

- **where**
  - $\mathcal{B} = 100$ training applicants (background)
  - Interventional expectation: features outside S are replaced independently of $\mathbf{x}_S$
  - Explanations are on the log-odds scale, where effects add.

- **Additivity on our data**: largest error $|\phi_0 + \sum_j \phi_j - f(\mathbf{x})| = 2.9\text{e-}06$ over 400 test applicants.
- **Linear model**: $\phi_j = \beta_j(x_j - \mathbb{E}x_j)$ exactly (linear SHAP).
- **Off-manifold risk**: with correlated features, independent replacement creates unrealistic applicants.

- **In practice:** the background sample defines the "average applicant"
  - Document it like any model input.

(Lundberg & Lee, 2017)

Experiment yourself: notebook §2

Kernel SHAP and TreeSHAP: how SHAP is computed

- **In one sentence:** Kernel SHAP fits a weighted linear regression on sampled coalitions; for trees the exact values come in polynomial time.

$$\min_{\phi} \sum_{\mathbf{z} \in \{0,1\}^d} \pi(\mathbf{z}) \left[v(\mathbf{z}) - \phi_0 - \sum_j \phi_j z_j \right]^2$$

$$\pi(\mathbf{z}) = \frac{d-1}{\binom{d}{|\mathbf{z}|} \, |\mathbf{z}| \, (d - |\mathbf{z}|)}$$

- **where**
  - $\mathbf{z}$ = coalition indicator
  - The Shapley kernel π makes the weighted least-squares solution equal the Shapley values (Appendix A)
  - We sample 600 coalitions per applicant.

- **Kernel SHAP**: model-agnostic
  - 16 s for 400 applicants.
- **TreeSHAP**: exact for tree ensembles such as XGBoost (Lundberg et al., 2020).
- **Deep networks**: Deep/Gradient SHAP combine Shapley values with gradients (Section 3).

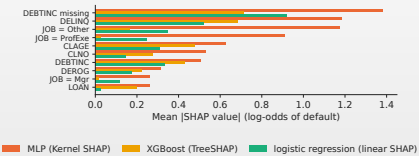
- **In practice:** Kernel SHAP has sampling noise: fix the seed and the number of samples in the model documentation.

(Lundberg & Lee, 2017; Lundberg et al., 2020)

Experiment yourself: notebook §2

Global importance: the missing debt-to-income ratio dominates

- **In one sentence:** mean absolute SHAP value per feature, for the network and the two benchmarks.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: 400 test applicants, 100 background applicants.

- Top for all three models: DEBTINC_missing
  - MLP: then DELINQ and JOB_Other
  - Spearman correlation of importances MLP–XGBoost 0.55, MLP–logistic 0.79.
- Debt-to-income ratio (DEBTINC) missing for 21.3 % of loans
  - Default rate 62.0 % when missing vs 8.6 % when reported.

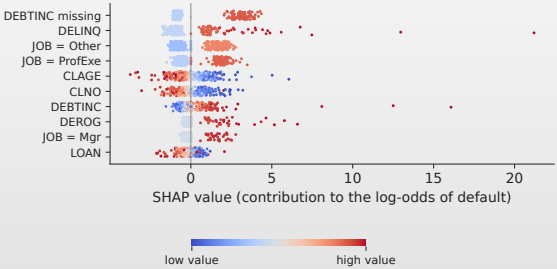
- **In practice:** an explanation can reveal that a model scores the paperwork rather than the borrower
  - Decide whether that is acceptable.

(own computation)

NN_L11_shap_global

The beeswarm: direction and spread of every effect

- **In one sentence:** each dot is one applicant; colour = rank of the feature value; position = its SHAP value.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: 400 test applicants; colour = rank of the feature value.

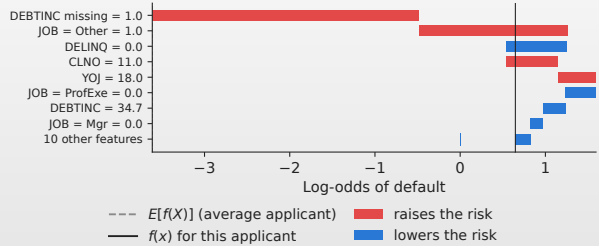
- Delinquent credit lines (DELINQ) and derogatory reports (DEROG): high values push the risk up, as credit theory expects.

(own computation)

NN_L11_shap_beeswarm

Local explanation of a rejected applicant: SHAP waterfall

- **In one sentence:** from the average log-odds ϕ_0 to this applicant's score, one feature at a time.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: test applicant (row 2411); values in original units.

- Test applicant (row 2411): PD 65.6 %, log-odds $0.648 = -3.613 + 4.261$
  - Largest pushes: DEBTINC_missing ($+3.13$), JOB_Other ($+1.74$), DELINQ (-0.71).

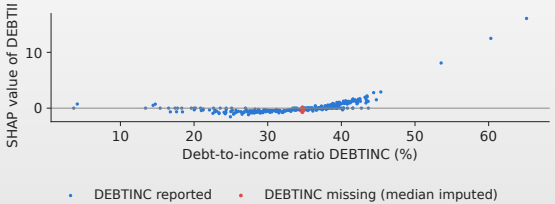
- **In practice:** the adverse-action notice lists the top reasons in this order, in words the customer understands.

(own computation)

NN_L11_shap_waterfall

Dependence plot: the network learned the missing-value signal

- **In one sentence:** SHAP value of DEBTINC against its value; red points are imputed at the training median.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: 400 test applicants, MLP.

- Imputed applicants sit at one DEBTINC value
  - Their risk is carried by the missing flag (mean SHAP +3.13), not by DEBTINC itself.

- **In practice:** imputation choices change explanations
  - Keep a missing-value flag so that the reason can be named.

(own computation)

NN_L11_shap_dependence

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

LIME: a simple model that is faithful near one point

- **In one sentence:** perturb the applicant, weight the perturbations by closeness, fit a sparse linear model to the network's answers.

$$\xi(\mathbf{x}) = \arg\min_{g \in \mathcal{G}} \sum_{\mathbf{z}} \pi_{\mathbf{x}}(\mathbf{z}) \Big[f(\mathbf{z}) - g(\mathbf{z}) \Big]^2 + \Omega(g)$$

$$\pi_{\mathbf{x}}(\mathbf{z}) = \exp\left(- \frac{\|\mathbf{z} - \mathbf{x}\|^2}{\sigma^2} \right)$$

- **where**
  - $\mathcal{G}$ = linear models $g(\mathbf{z}) = \beta_0 + \beta^\top \mathbf{z}$
  - Ω = complexity penalty
  - $\mathbf{z} \sim \mathcal{N}(\mathbf{x}, \mathbf{I})$ in standardised units
  - Kernel width $\sigma = 0.75\sqrt{d} = 3.18$ by default.

- **Local slope**, not a split of the prediction: the coefficients do not add up to $f(\mathbf{x})$.
- **Three choices** decide the answer: neighbourhood, kernel width, number of samples.
- **Our applicant**: LIME's top feature is DELINQ, SHAP's is DEBTINC_missing.

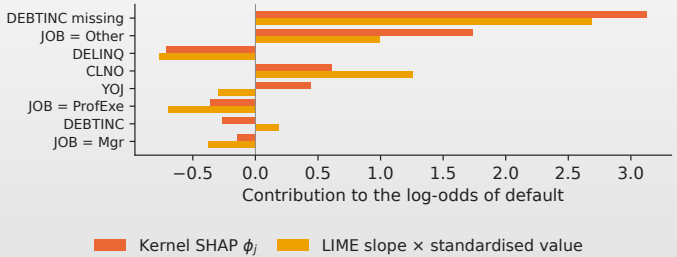
- **In practice:** LIME answers "what moves the score near this customer?", SHAP answers "why is this customer far from average?".

(Ribeiro, Singh & Guestrin, 2016)

Experiment yourself: notebook §3

LIME and SHAP for the same applicant

- **In one sentence:** LIME slope times the standardised value, next to the SHAP contribution, for the eight largest SHAP features.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: test applicant (row 2411); LIME with 5,000 samples.

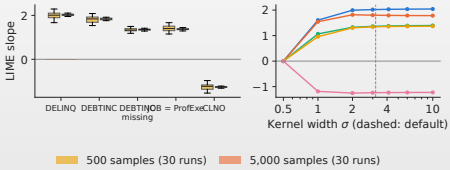
- Spearman correlation of the absolute contributions: 0.80
  - A binary flag is far from its local linear approximation.

(own computation)

NN_L11_lime_vs_shap

Sensitivity of LIME to its settings

- **In one sentence:** left: 30 runs with different random seeds; right: the five slopes as the kernel width changes.



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: test applicant (row 2411); right panel: same five features, 5,000 samples.

- Same top feature: 73 % of runs with 500 samples, 100 % with 5,000
  - Jaccard overlap of the top-5 sets 0.72 vs 1.00
  - Coefficient of variation 10 % vs 2 %.

- **In practice:** fix and report the seed, sample size and kernel width of every LIME explanation that reaches a customer or an auditor.

(own computation)

NN_L11_lime_instability

Counterfactual explanations: minimal changes that reverse the decision

- **In one sentence:** find the closest applicant, in actionable features only, whom the model would accept.

$$\mathbf{x}' = \arg\min_{\mathbf{x}'} \; \lambda \left(f(\mathbf{x}') - t \right)_+^2 + \sum_{j \in \mathcal{A}} \frac{|x'_j - x_j|}{\mathrm{MAD}_j}$$

- **where**
  - t = target log-odds (-0.5, below the 0.5 PD threshold)
  - $\mathcal{A}$ = actionable features (LOAN, DEBTINC, NINQ, DELINQ, DEROG, CLNO, MORTDUE)
  - Median absolute deviation (MAD) scales the distance
  - Raw values kept ≥ 0, counts rounded.

Feature	now	needed
MORTDUE	69 115.0	57 456.5
DEBTINC	34.7	33.4
LOAN	14 500.0	15 700.0

- PD 65.6 % → 39.2 % after these changes.
- Documentary counterfactual: reporting DEBTINC (at the median 34.7) gives PD 5.1 %.

- **In practice:** "bring the missing income document" is a better answer to the customer than any list of SHAP values.

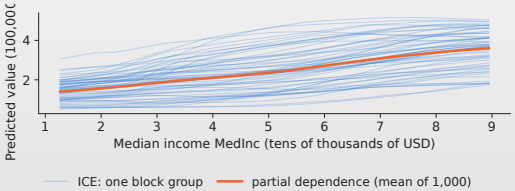
(Wachter, Mittelstadt & Russell, 2017)

Experiment yourself: notebook §4

Global effects: partial dependence and ICE

- **In one sentence:** the PDP averages the prediction over the data while one feature is set to a grid value; ICE keeps one curve per observation.

$$\mathrm{PD}_j(z) = \frac{1}{n}\sum_{i=1}^{n} f(z, \mathbf{x}_{i,-j}), \qquad \mathrm{ICE}_{i,j}(z) = f(z, \mathbf{x}_{i,-j})$$



Source: StatLib, California housing, 1990 census (Pace and Barry, 1997). Data: MLP; 60 test block groups (ICE), 1,000 for the PDP.

- California housing MLP (test R^2 0.819, linear 0.620): the predicted value rises from 1.40 to 3.60 (100,000 USD) over the income grid
  - Spread of the ICE curves at the top 0.95.

(Friedman, 2001; Goldstein et al., 2015)

NN_L11_pdp_ice

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

Fraud case study: a network trained on day 1 and explained on day 2

- **In one sentence:** gradients are free for a network: one backward pass gives the sensitivity of the fraud score to every input.

$$\text{saliency}_j(\mathbf{x}) = \left|\frac{\partial f(\mathbf{x})}{\partial x_j}\right|, \qquad \text{G}\times\text{I}_j(\mathbf{x}) = (x_j - x'_j)\,\frac{\partial f(\mathbf{x})}{\partial x_j}$$

- **where**
  - f = fraud log-odds of an MLP on V1–V28 and log amount
  - $\mathbf{x}'$ = baseline = mean legitimate transaction of day 1
  - Class weight $\sqrt{\text{imbalance}}$, early stopping on weighted validation PR-AUC.

- **Test, day 2** (full-data weights): MLP PR-AUC 0.815, logistic 0.742
  - AUC 0.961 vs 0.964.
- **Saturation**: in a flat region of f the gradient is near zero even for a decisive feature.
- **ReLU networks**: the gradient is piecewise constant, so saliency jumps between linear regions.

- **In practice:** a fraud analyst gets a ranked list of the inputs that drove the alert, in milliseconds per transaction.

(Simonyan, Vedaldi & Zisserman, 2014; Dal Pozzolo et al., 2015)

Experiment yourself: notebook §5

Integrated gradients and the completeness of attributions

- **In one sentence:** integrate the gradient along the straight path from a baseline to the input; the attributions sum to $f(\mathbf{x}) - f(\mathbf{x}')$.

$$\mathrm{IG}_j(\mathbf{x}) = (x_j - x'_j) \int_0^1 \frac{\partial f\left(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}')\right)}{\partial x_j}\, d\alpha$$
$$\approx (x_j - x'_j) \frac{1}{m} \sum_{k=1}^{m} \frac{\partial f\left(\mathbf{x}' + \frac{k-1/2}{m}(\mathbf{x} - \mathbf{x}')\right)}{\partial x_j}$$

- **where**
  - m = Riemann steps (midpoint rule, 64 here)
  - Completeness $\sum_j \mathrm{IG}_j = f(\mathbf{x}) - f(\mathbf{x}')$ follows from the fundamental theorem of calculus for paths.

- **Axioms** (Sundararajan et al., 2017)
  - Completeness; sensitivity: a feature that changes f gets non-zero credit
  - Implementation invariance: equal functions, equal attributions
- **Link to Shapley**: IG is the Aumann-Shapley value of the continuous game along the path.
- **Baseline matters**: "mean legitimate transaction" answers "why is this not a normal payment?".

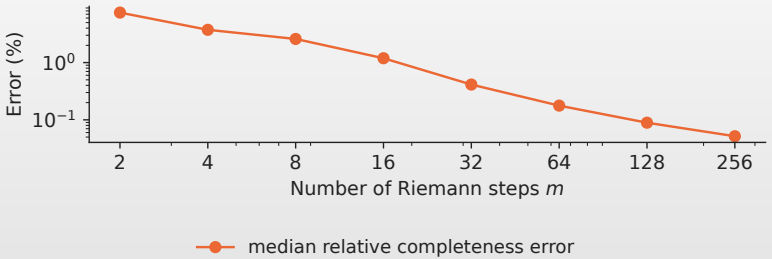
- **In practice:** write the baseline into the model documentation: a different baseline is a different question.

(Sundararajan, Taly & Yan, 2017)

Experiment yourself: notebook §5

Completeness holds once the integral is computed accurately

- **In one sentence:** median of $|\sum_j \mathrm{IG}_j - (f(\mathbf{x}) - f(\mathbf{x}'))|/|f(\mathbf{x}) - f(\mathbf{x}')|$ over the frauds of day 2.



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: 211 frauds of day 2; baseline = mean legitimate transaction of day 1.

- 3.7 % error with 4 steps, 0.18 % with 64, over 211 frauds
  - The error falls roughly as $1/m^2$, as for the midpoint rule.

(own computation)

NN_L11_ig_completeness

SmoothGrad and concept-level explanations

- **In one sentence:** average gradients over noisy copies of the input to remove ReLU jitter; or ask whether a human concept moves the output.

$$\mathrm{SG}_j(\mathbf{x}) = \frac{1}{n}\sum_{k=1}^{n} \frac{\partial f(\mathbf{x}+\varepsilon_k)}{\partial x_j}, \quad \varepsilon_k \sim \mathcal{N}(\mathbf{0}, s^2\mathbf{I})$$

$$S_C(\mathbf{x}) = \nabla h(\mathbf{x})^\top \mathbf{v}_C$$

- **where**
  - $n = 50$, $s = 0.15$ standard deviations
  - Testing with concept activation vectors (TCAV): $\mathbf{v}_C =$ normal of a linear classifier separating examples of concept C in a hidden layer h
  - The TCAV score is the share of inputs with $S_C > 0$.

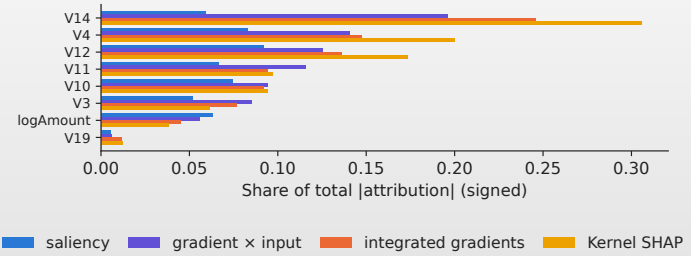
- **SmoothGrad** sharpens image saliency maps and stabilises tabular gradients (next section).
- **TCAV** speaks the language of the business: "night-time transaction", "new merchant" — if such concepts are labelled.
- **Our data**: V1–V28 are anonymised components, so concept tests are not possible here.

- **In practice:** concept explanations need concepts labelled when the data are collected.

(Smilkov et al., 2017; Kim et al., 2018)

Four explanation methods for one fraudulent transaction

- **In one sentence:** signed attributions of one day-2 fraud (median fraud score), as shares of the total absolute attribution.



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: one fraud of day 2 (median fraud score).

- Top feature: Kernel SHAP V14, integrated gradients V14, saliency V12
  - Fraud score 99.9 %: saliency is blind to the size of the deviation from the baseline.

(own computation)

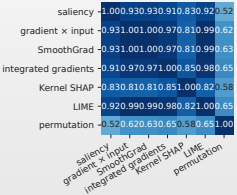
NN_L11_fraud_local

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

Agreement between explanation methods

- **In one sentence:** Spearman correlation of the global importances (mean |attribution|) of seven methods on the same frauds.



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: Spearman correlation of global importances; 211 frauds of day 2.

- IG–SHAP 0.85, saliency–SHAP 0.83, LIME–SHAP 0.82, permutation–SHAP 0.58
  - All methods put V12 first.

- **In practice:** disagreement is information: permutation importance measures what the model needs, attributions what it uses.

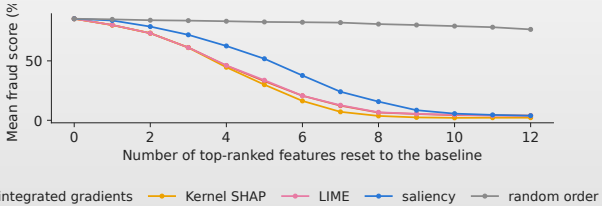
(Breiman, 2001; own computation)

NN_L11_method_agreement

Faithfulness of explanations: deletion test

- **In one sentence:** reset the top-k features of each fraud to the baseline; a faithful explanation makes the score fall fast.

$$\mathrm{AOPC} = \frac{1}{K}\sum_{k=1}^{K}\left[f(\mathbf{x}) - f(\mathbf{x}^{(k)})\right]$$



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: 211 frauds of day 2.

- Area over the perturbation curve (AOPC) in points of fraud score: IG 56.2, SHAP 58.4, LIME 56.1, saliency 48.0, random order 3.9.

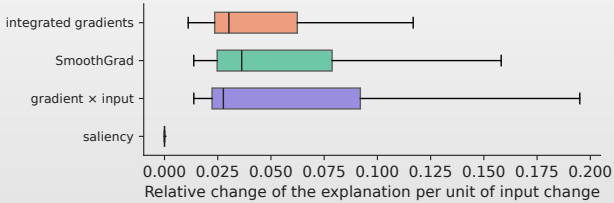
(Samek et al., 2017)

NN_L11_deletion_curve

Local robustness of explanations to small input changes

- **In one sentence:** relative change of the attribution vector per unit of a small perturbation (Lipschitz-type ratio).

$$\hat{L}(\mathbf{x}) = \frac{\|\mathbf{a}(\mathbf{x}+\varepsilon) - \mathbf{a}(\mathbf{x})\| / \|\mathbf{a}(\mathbf{x})\|}{\|\varepsilon\|}, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, 0.05^2\mathbf{I})$$



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: 60 frauds of day 2 x 10 perturbations (sigma = 0.05 standard deviations).

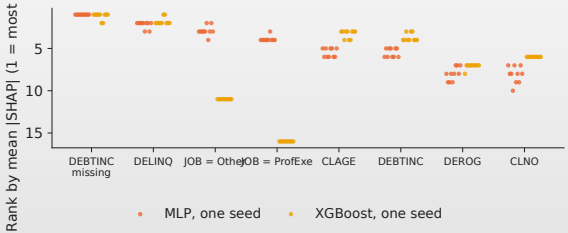
- Medians: saliency 0.000, gradient × input 0.028, IG 0.030, SmoothGrad 0.036
  - Within a linear region of a ReLU network the gradient does not move.

(Alvarez-Melis & Jaakkola, 2018)

NN_L11_local_robustness

The Rashomon effect: equally accurate models with different explanations

- **In one sentence:** rank of each feature by mean |SHAP| for 10 retrained networks and 10 boosted ensembles (different seeds only).



Source: Baesens, Roesch and Scheule (2016), HMEQ data. Data: 10 training seeds; 200 test applicants.

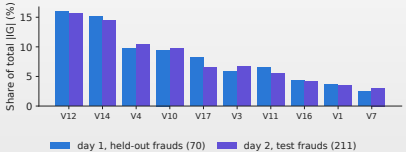
- MLP test AUC between 0.929 and 0.941
  - Mean pairwise Kendall τ (rank agreement of importances, 1 = identical order): MLP 0.83, XGBoost 0.97
  - Same top feature for one applicant in 78 % (MLP) and 83 % (XGBoost) of seed pairs.

(own computation)

NN_L11_seed_stability

Explanations over time: day 1 against day 2

- **In one sentence:** share of the total |IG| by feature, for held-out frauds of the training day and for the frauds of the next day.



Source: ULB Machine Learning Group and Worldline, credit-card transactions (OpenML 1597). Data: model trained on day 1; integrated gradients.

- Spearman 0.98, Jensen-Shannon (JS) divergence (built on the Kullback-Leibler (KL) divergence) 0.003, 4 of the top 5 shared
  - One day later the explanation is stable here.

- **In practice:** monitor the distribution of attributions monthly, next to the score distribution
  - A drift in reasons can precede a drift in accuracy.

(own computation) NN_L11_time_shift

Stability scorecard of this lecture

Source of change	Measure	Result
Explainer randomness (LIME)	same top feature over 30 runs	73 % (500), 100 % (5,000 samples)
Training seed (Rashomon)	Kendall τ of global importances	MLP 0.83, XGBoost 0.97
Training seed, one applicant	same top feature	MLP 78 %, XGBoost 83 %
Method	Spearman of importances	0.52 – 1.00
Input noise	median relative change	IG 0.030, SmoothGrad 0.036
Time (day 1 $\rightarrow$ day 2)	Spearman, JS divergence	0.98; 0.003

⊡ **In practice:** report every explanation with its stability: which seed, which sample, which period, and how much it moves.

(own computation)

Outline

- Reasons for explaining neural networks
- Shapley values and SHAP
- LIME, counterfactuals and global effects
- Gradient methods for neural networks
- Agreement, faithfulness and stability of explanations
- Explainability in model governance

An explainability audit checklist

- **Purpose**: who reads the explanation (customer, analyst, validator, regulator) and what decision it supports.
- **Settings**: background sample, baseline, seed, number of samples, kernel width — versioned with the model.
- **Checks**: additivity/completeness, faithfulness (deletion), stability (seeds, time), agreement of two methods.
- **Fairness**: attributions of protected attributes and of their proxies (age, sex, postcode)
  - An explanation shows use, not causation.
- **Customer output**: few reasons, actionable counterfactuals, no raw component names (V14 means nothing to a customer).
- **Alternative**: if an interpretable model is almost as accurate (here logistic AUC 0.909 vs MLP 0.935), consider using it.

In practice: the validation report should contain the explanations and their stability tests, not only the accuracy.

(SR 11-7; Rudin, 2019)

Quiz 1: adding up SHAP values

- For our applicant the base value is -3.613 and the SHAP values sum to 4.261.
  - What is the model's log-odds of default, and why can you be sure?

Answer

- 0.648: by the efficiency axiom $f(\mathbf{x}) = \phi_0 + \sum_j \phi_j$
  - The PD is $\sigma(0.648) = 65.6\,\%$.

Quiz 2: an unstable LIME

- Two analysts explain the same rejected applicant with LIME and report different top reasons.
  - Is the model inconsistent?

Answer

- Not necessarily: LIME samples perturbations
  - With 500 samples the top feature matched in 73 % of our runs
  - Fix the seed, raise the samples, check against SHAP.

Quiz 3: choosing an IG baseline

- A colleague uses the all-zero vector as the integrated-gradients baseline for standardised fraud features.
  - What question does that explanation answer?

Answer

- "Why does this transaction differ from the average transaction (frauds included)?"
  - Our baseline, the mean legitimate transaction, asks why it is not a normal payment.

Wrap-up

- **SHAP**: Shapley values with an exact additivity check
  - The missing debt-to-income ratio drives the credit network.
- **LIME**: a local slope that depends on seed, sample size and kernel width
  - Useful, but report its settings.
- **Gradients**: integrated gradients satisfy completeness (0.18 % error at 64 steps) and are as faithful as SHAP at a fraction of the cost.
- **Stability**: retraining with another seed reorders features (Kendall τ 0.83 for the MLP)
  - Explanations must be tested like models.
- **Next**: Lecture 12, large language models, where explanations become text.

Research from our group: XAI in fraud and energy markets

- **SHAP stability in graph-augmented fraud detection**: on the IEEE-CIS fraud data, SHAP contributions of an XGBoost model with graph neural network (GNN) embeddings are compared under random and forward-time validation, with a Gaussian control of the same dimension and covariance
  - Whether the graph model looks more stable depends on how the change is measured, a careful protocol for explanation stability (Diaconu & Pele, 2026, submitted to Expert Systems with Applications; code: QuantLet GNN_XAI_Stability).
- **Explainable early warning of electricity price spikes** in Romania (Andrei, Pele, Cramer, Amza & Diaconu, 2026, Proceedings of the International Conference on Business Excellence (PICBE)) and **SHAP-driven anomaly detection** of electricity prices (Andrei & Pele, 2025, Preprints).
- **Multi-horizon explainability** of price forecasts: which drivers matter at which horizon in Romania's liberalised market (Andrei, Pele & Cramer, 2026, PICBE).
- Findings as stated in the papers; numbers on the other slides come from our own data.

(Bucharest University of Economic Studies (ASE); Romanian Academy, Institute for Economic Forecasting (IPE))

Explore it visually

- **Neuronpedia**
  - Open-source platform for the interpretability of language models: sparse autoencoders split the activations of a layer into features, and each feature page lists the texts on which it fires most.
  - Compare the two views: SHAP and LIME attribute one prediction to the inputs, while these features describe what the inside of the model represents.

Further reading and resources

- **C. Molnar, Interpretable Machine Learning**, free online: christophm.github.io/interpretable-ml-book.
- **SHAP library** and examples: shap.readthedocs.io.
- **Rudin (2019)**: the case for interpretable models in high-stakes decisions.
- **This course**: code, data, notebooks — github.com/danpele/NEURAL_BIZ.
- **Experiment yourself**: notebooks/L11_notebook.ipynb
  - The seminar explains a German credit model, with a fairness check.
- **Replication of our stability study**: github.com/QuantLet/GNN_XAI_Stability.

References (1/3)

- Alvarez-Melis, D. & Jaakkola, T. S. (2018). On the robustness of interpretability methods. ICML Workshop on Human Interpretability in Machine Learning. arXiv:1806.08049
- Andrei, A.-V. & Pele, D. T. (2025). Explainable AI for electricity price anomaly detection: A SHAP-driven approach in Romania's energy market. *Preprints*. doi:10.20944/preprints202508.0410.v1
- Andrei, A.-V., Pele, D.-T. & Cramer, A.-A. (2026). Multi-horizon explainability in energy price forecasting: Disentangling market dynamics and renewable intermittency in Romania's liberalized electricity market. *Proceedings of the International Conference on Business Excellence*, 20(1), 104–117. doi:10.2478/picbe-2026-0014
- Andrei, A.-V., Pele, D.-T., Cramer, A.-A., Amza, A. & Diaconu, D. (2026). Explainable AI-based early warning system for electricity price spikes: Evidence from the Romanian market. *Proceedings of the International Conference on Business Excellence*, 20(1), 4820–4836. doi:10.2478/picbe-2026-0352
- Baesens, B., Roesch, D. & Scheule, H. (2016). *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS*. Wiley.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. doi:10.1023/A:1010933404324
- Dal Pozzolo, A., Caelen, O., Johnson, R. A. & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. IEEE SSCI, 159–166. doi:10.1109/SSCI.2015.33
- Diaconu, D. & Pele, D. T. (2026). SHAP stability across random and forward-time validation in graph-augmented fraud detection. Submitted to *Expert Systems with Applications*. Code: github.com/QuantLet/GNN_XAI_Stability

References (2/3)

- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. doi:10.1214/aos/1013203451
- Goldstein, A., Kapelner, A., Bleich, J. & 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. doi:10.1080/10618600.2014.907095
- Kim, B. et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). ICML. arXiv:1711.11279
- Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. arXiv:1705.07874
- Lundberg, S. M. et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2, 56–67. doi:10.1038/s42256-019-0138-9
- Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. KDD, 1135–1144. doi:10.1145/2939672.2939778
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1, 206–215. doi:10.1038/s42256-019-0048-x
- Samek, W., Binder, A., Montavon, G., Lapuschkin, S. & Müller, K.-R. (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11), 2660–2673. doi:10.1109/TNNLS.2016.2599820

References (3/3)

- Shapley, L. S. (1953). A value for n-person games. In *Contributions to the Theory of Games II*, 307–317. Princeton University Press.
- Simonyan, K., Vedaldi, A. & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. ICLR Workshop. arXiv:1312.6034
- Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M. (2017). SmoothGrad: Removing noise by adding noise. arXiv:1706.03825
- Sundararajan, M., Taly, A. & Yan, Q. (2017). Axiomatic attribution for deep networks. ICML, 3319–3328. arXiv:1703.01365
- Wachter, S., Mittelstadt, B. & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2), 841–887. doi:10.2139/ssrn.3063289
- Regulation (EU) 2024/1689 (Artificial Intelligence Act), *Official Journal of the European Union*, L, 12 July 2024. data.europa.eu/eli/reg/2024/1689/oj
- Board of Governors of the Federal Reserve System (2011). SR 11-7: Supervisory guidance on model risk management. federalreserve.gov

Appendix A: recovery of Shapley values by the Shapley kernel

- Restrict $g(\mathbf{z}) = \phi_0 + \sum_j \phi_j z_j$ to satisfy $g(\mathbf{0}) = v(\emptyset)$ and $g(\mathbf{1}) = v(\{1..d\})$ (infinite weight on the empty and full coalitions).
- The weighted least-squares normal equations with $\pi(\mathbf{z}) \propto (d-1)/[\binom{d}{|\mathbf{z}|}|\mathbf{z}|(d-|\mathbf{z}|)]$ have the unique solution $\phi_j =$ Shapley value (Lundberg & Lee, 2017, Theorem 2).
- Kernel SHAP samples coalitions from π (pairing $\mathbf{z}$ with $\mathbf{1}-\mathbf{z}$) and solves the constrained regression
  - The error falls as $1/\sqrt{\text{samples}}$.
- For a linear f and independent features, $v(S) = \beta_0 + \sum_{j\in S}\beta_j x_j + \sum_{j\notin S}\beta_j \mathbb{E}x_j$, so $\phi_j = \beta_j(x_j - \mathbb{E}x_j)$.

(Lundberg & Lee, 2017)

Appendix B: integrated gradients as a path method

- For a path $\gamma(\alpha)$ from $\mathbf{x}'$ to $\mathbf{x}$: $f(\mathbf{x}) - f(\mathbf{x}') = \int_0^1 \nabla f(\gamma(\alpha))^\top \gamma'(\alpha)\, d\alpha = \sum_j \int_0^1 \partial_j f(\gamma)\, \gamma_j'\, d\alpha$.
- The straight line $\gamma(\alpha) = \mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}')$ gives $\gamma_j' = x_j - x_j'$: each term is IG_j, hence completeness.
- Among path methods, the straight line is the only one that is symmetric in features (Sundararajan et al., 2017).
- Midpoint Riemann sum error is $O(m^{-2})$ for smooth f
  - ReLU kinks make it slower, as our completeness chart shows at small m.

(Sundararajan, Taly & Yan, 2017)