Advanced Topic A04: *SHAP Values: Game Theory Meets Explainability*

Data Science with Python – BSc Advanced Lectures

Joerg Osterrieder

10 Minutes

Motivation: Why Did the Model Reject the Loan?

Accuracy is not enough – regulators want reasons

- EU AI Act, GDPR, and US fair lending require explanations for automated decisions
- "Your XGBoost model rejected the loan" is not legally sufficient
- Feature importances are global and average; regulators want per-decision attribution
- SHAP provides a mathematically principled, locally faithful explanation

Explainability is not optional in regulated industries – it is a legal requirement

The Shapley Value: Fair Credit Allocation

A 1953 solution to a cooperative game theory problem

- Lloyd Shapley asked: how should the payout from a cooperative game be divided fairly among players?
- Players can form coalitions; each coalition generates a value
- The Shapley value allocates payout so that each player receives their average marginal contribution across all possible coalition orderings
- Four axioms guarantee uniqueness: efficiency, symmetry, dummy, additivity

Shapley (1953): A value for n-person games; Nobel Prize in Economics 2012

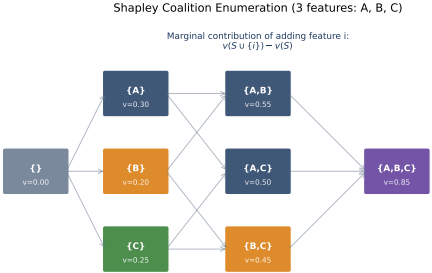
The Coalition Formula

Average marginal contribution over all coalitions

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \big[v(S \cup \{i\}) - v(S) \big]$$

Where:

- F: set of all features; S: a coalition not containing feature i
- $v(S)$: model prediction using only features in S (others marginalised)
- $v(S \cup \{i\}) - v(S)$: marginal contribution of adding feature i



Four Axioms: Why Shapley Values Are Unique

The only attribution method satisfying all four simultaneously

- **Efficiency:** $\sum_i \phi_i = f(x) - f(\bar{x})$ (attributions sum to prediction minus baseline)
- **Symmetry:** two features with identical marginal contributions receive the same attribution
- **Dummy:** a feature that never changes the prediction receives attribution zero
- **Additivity:** attributions for a sum of models equal the sum of their attributions

Any attribution method satisfying all four axioms must be the Shapley value

The Computational Problem

Exact Shapley values are intractable for large feature sets

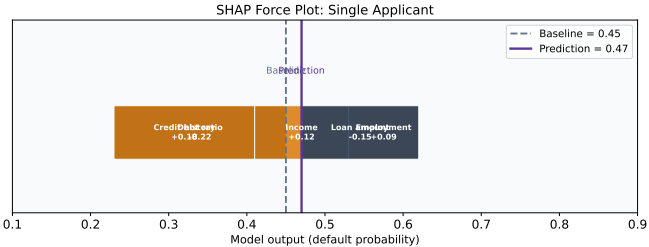
- With $n = 30$ features: $2^{30} \approx 10^9$ subsets to evaluate
- Each evaluation requires a model forward pass: impossible in practice
- Two practical solutions: **KernelSHAP** (model-agnostic, samples coalitions, fits weighted linear model) and **TreeSHAP** (exact polynomial-time algorithm for tree-based models)
- TreeSHAP is $O(TLD^2)$ where T = trees, L = leaves, D = depth

Lundberg & Lee (2017): A unified approach to interpreting model predictions

Force Plot: Per-Prediction Attribution

How each feature pushes the prediction up or down

- The force plot shows a single prediction relative to the baseline (average prediction)
- Features pushing prediction up (positive SHAP): shown in one colour
- Features pushing prediction down (negative SHAP): shown in another colour
- Width of each bar is proportional to the magnitude of the SHAP value

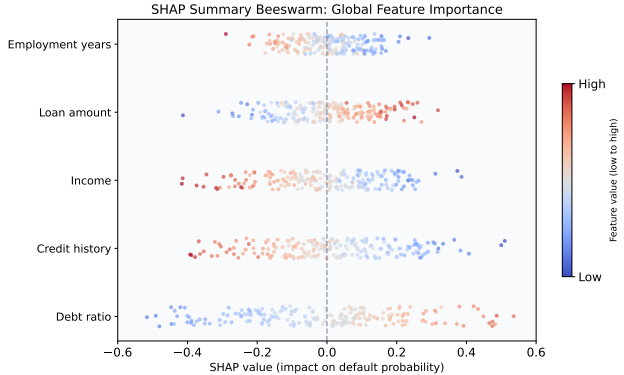


Force plot: each applicant gets their own explanation, satisfying per-decision requirements

Summary Beeswarm: Global Feature Importance

SHAP values across all predictions in the dataset

- Each dot is one data point; x-axis is the SHAP value for that feature
- Colour represents the feature value (high = AMBER, low = NAVY)
- Features sorted by mean absolute SHAP value (most important at top)
- Reveals both direction of effect and heterogeneity across the dataset



The beeswarm shows what global feature importance plots hide: sign and distribution

KernelSHAP: Model-Agnostic Approximation

Sampling coalitions instead of enumerating them

- KernelSHAP treats the model as a black box and samples random subsets of features
- Each sampled coalition $(S, f(S))$ becomes a training point for a weighted linear model
- The weight is the Shapley kernel: $\pi(S) = \frac{|F|-1}{\binom{|F|}{|S|}|S|(|F|-|S|)}$
- Solving the weighted least squares problem recovers approximate Shapley values
- With 100–200 samples: accuracy comparable to exact values for $n \leq 50$ features

KernelSHAP: any model, any language, any framework; slower than TreeSHAP by 10–100x

Interaction SHAP Values

Pairwise feature interactions beyond main effects

- SHAP interaction value ϕ_{ij}: the extra effect of features i and j appearing together beyond their individual contributions
- Captures non-linear dependencies: "high debt ratio only matters when income is low"
- Available via TreeSHAP's `tree_shap_interactions` method
- Visualised as a matrix; diagonal = main effects, off-diagonal = interactions

Interaction values decompose the model into main effects and pairwise terms

SHAP Dependence Plots

How one feature's SHAP value varies with its own value

- Plot SHAP value for feature i vs feature i's raw value
- Slope shows direction and magnitude of the marginal effect
- Colour a second feature to reveal interactions (automatic colouring by the feature most correlated with the displayed SHAP values)
- Non-linear shapes reveal threshold effects, diminishing returns, or reversals

Dependence plots replace partial dependence plots with a locally faithful alternative

Waterfall Plot: Step-by-Step Prediction Decomposition

Building up the prediction from the baseline feature by feature

- The waterfall plot displays a single prediction as a sequence of additive steps
- Start from $E[f(x)]$ (the baseline – average prediction over the training set)
- Each row adds one feature's SHAP value; the final value is the model output $f(x)$
- Features sorted by magnitude so the most influential appear first
- Directly answers: "What moved this prediction away from the average?"

Waterfall plots are favoured by regulators for adverse action notices

SHAP for Deep Learning: DeepSHAP and GradientSHAP

Extending Shapley explanations to neural networks

- TreeSHAP's polynomial trick does not apply to neural networks
- **DeepSHAP**: uses DeepLIFT backpropagation; compares activations to a reference
- **GradientSHAP**: samples reference inputs and computes integrated gradients; approximates Shapley under an independence assumption
- Both are faster than KernelSHAP for large networks but are approximations
- In finance: explain fraud scores, default probabilities, or NLP-based sentiment models

For tabular data, prefer TreeSHAP; for neural networks, GradientSHAP is a practical default

Limitations of SHAP

SHAP is principled but not perfect

- KernelSHAP assumes feature independence for out-of-coalition features: problematic when features are correlated (income and wealth always move together)
- SHAP explains the model, not the world: a high SHAP for "neighbourhood" reflects the model's reliance on neighbourhood, not necessarily a causal relationship
- TreeSHAP is fast but tied to tree models; deep learning SHAP approximations (DeepSHAP, GradientSHAP) are less exact

SHAP = model explanation; for causal explanation, combine with DoWhy (A03)

Code Pointer

Three lines to explain any model

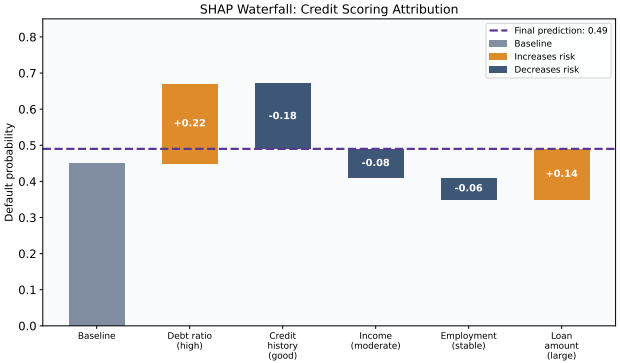
- `pip install shap`
- `explainer = shap.TreeExplainer(model)` (for XGBoost/LightGBM/RandomForest)
- `shap_values = explainer(X)`
- `shap.plots.beeswarm(shap_values)`
- For neural nets: `shap.DeepExplainer` or `shap.GradientExplainer`

SHAP integrates with scikit-learn, XGBoost, LightGBM, CatBoost, and PyTorch

Finance Application: Credit Scoring Attribution

Explaining individual lending decisions to applicants

- Regulatory requirement (GDPR Art. 22, ECOA): adverse action notices must list the top reasons for a denial
- SHAP provides the top-k features with largest magnitude for each applicant
- Features with positive SHAP: drove the score up (favourable)
- Features with negative SHAP: drove the score down (reasons for denial)



SHAP adverse action notices are now standard in model governance frameworks

Portfolio Risk Attribution with SHAP

Explaining factor contributions to portfolio volatility

- Apply SHAP to a risk model predicting portfolio variance from factor exposures
- Each factor (momentum, value, quality, rates) receives a SHAP value per date
- Positive SHAP: this factor increased predicted risk on that date
- Aggregating SHAP values over time reveals which factors chronically drive tail risk
- Risk managers use this for factor hedging decisions and regulatory stress-test narratives

SHAP converts a black-box risk model into an auditable factor-contribution dashboard

Practical Takeaway

When to use SHAP and when not to

- **Use SHAP when:** you need per-decision explanations, model debugging, regulatory compliance, or feature selection guidance
- **Do not use SHAP when:** you need causal attribution (use DoWhy instead), or when features are highly correlated (independence assumption is violated)
- For correlated features: use interventional SHAP or causal SHAP variants
- Always validate explanations against domain knowledge before presenting to regulators

SHAP is a debugging and communication tool, not a causal analysis method

Model Governance: SHAP in Production

Beyond one-time explanations: monitoring feature importance over time

- Compute SHAP values on a rolling window of live predictions
- Track mean absolute SHAP per feature: sudden shifts signal data drift or model degradation
- Alert when a previously minor feature becomes dominant (regime change signal)
- Regulatory model risk management (SR 11-7) requires ongoing monitoring: SHAP dashboards satisfy this requirement with minimal extra infrastructure
- Major banks now include SHAP drift charts in their monthly model performance reports

SHAP as a monitoring signal: detect model failure before it reaches the P&L

Summary and Open Questions

Three things to remember

- Shapley values are the unique attribution satisfying efficiency, symmetry, dummy, and additivity; they distribute prediction credit fairly among features
- TreeSHAP computes exact values in polynomial time; KernelSHAP approximates for any model; both are available in the `shap` library
- Finance: SHAP is the standard tool for adverse action notices and model governance; combine with causal analysis for full regulatory compliance

Open questions:

1. Can SHAP handle correlated features exactly?
2. How do we explain foundation models with billions of parameters?
3. Can SHAP values be manipulated adversarially?

Further reading: Lundberg & Lee (2017); Lundberg et al. (2020) From local to global