ENSEMBLE LEARNING & EXPLAINABLE AI

Darrel Danadyaksa Poli

COMPFEST 18 Academy · 18 Agustus 2026

BANK APP now

Your loan application was declined.

Ref: `model_v3.2`

No further explanation is available.

This is not hypothetical

- **Apple Card, 2019**: couples with shared finances given very different credit limits. Investigated by the New York DFS.
- **Amazon hiring, 2018**: a model scrapped after it learned to penalise the word “women's”.
- **EU AI Act · OJK POJK 22/2023**: “the model decided” is not a defence.

Sources: NY DFS Apple Card Report (2021) · Reuters (2018) · EU AI Act (2024) · OJK POJK 22/2023

Two regulators and one investigation, all asking the same question: why.

ACCURATE

ACCOUNTABLE

University teaches the top one.
Work demands both.

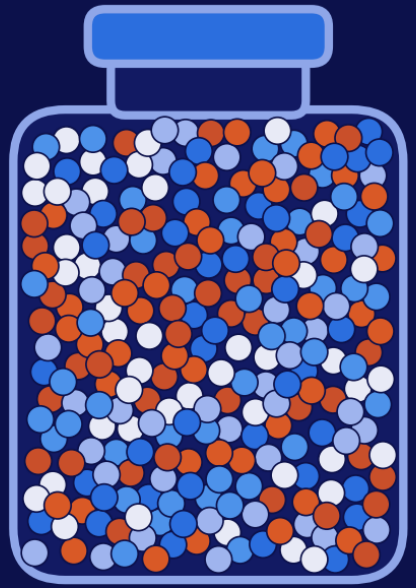
My Experience

- **NVIDIA AI Research Intern**: formal language evaluation
- **Meeting.ai AI Engineer Intern**: ASR stress-testing (10+ streams), speaker verification knowledge distillation & active learning
- **Full-Stack & Agentic AI**: architected `Marie Chan` AI (Go + Django, OpenRouter); co-founded `SafetyPin` startup (Django + Flutter)
- **Teaching & Peer Support @ UI**: TA for AI, Software Engineering Project, OS & Calculus; Certified Peer Counselor

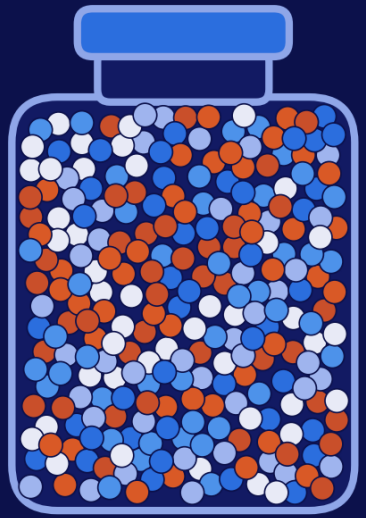
By the end of this session you can

- explain why a crowd of weak models beats one strong model
- choose between bagging and boosting, and say why
- never again trust raw `feature_importances_`
- turn one prediction into a sentence a human can act on

Guess how many. Chat, now.



387



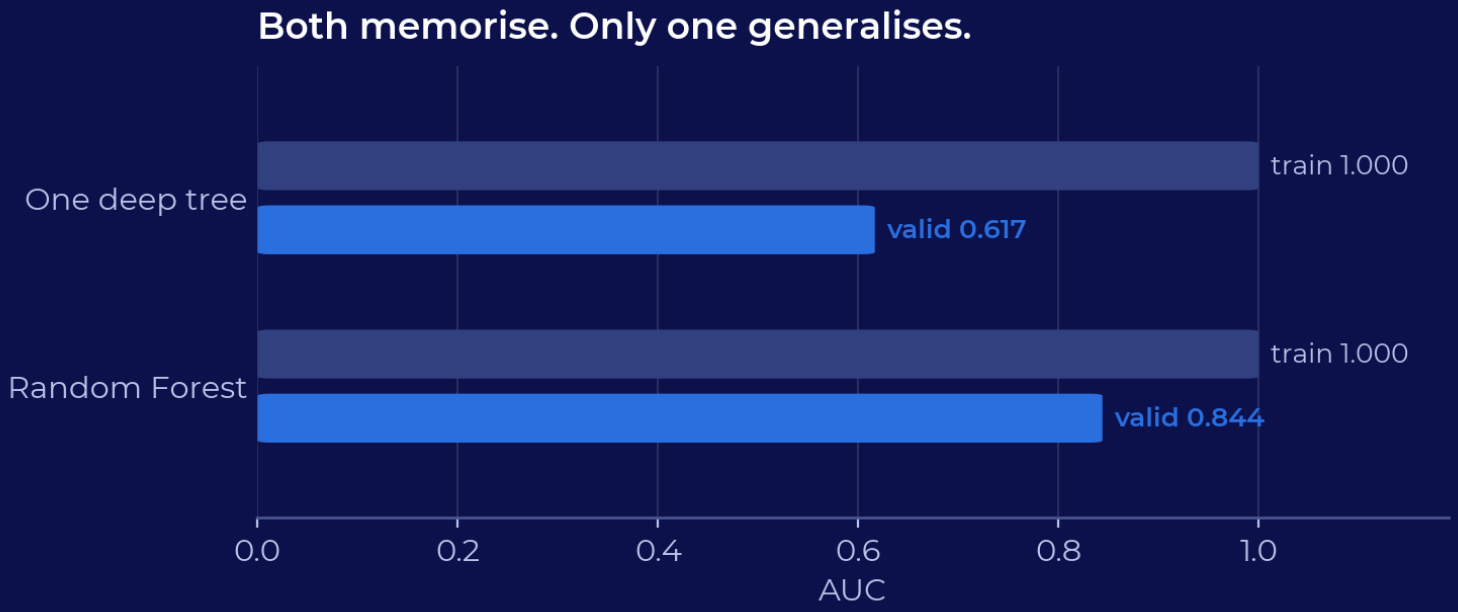
No individual was expert. The errors cancelled. That is the whole idea.

Why averaging works

$$\text{Var}(\text{mean of } n \text{ models}) = \rho\sigma^2 + \frac{(1-\rho)\,\sigma^2}{n}$$

- the right term vanishes as n grows. **The left one does not.**
- ρ is how correlated the models' errors are

One deep tree, on 150,000 applicants



ENSEMBLE LEARNING

Many weak models beat one strong one

BAGGING
parallel · different data · errors cancel

BOOSTING
sequential · each fixes the last one's errors

STACKING
a model learns how to combine

THREE
ways

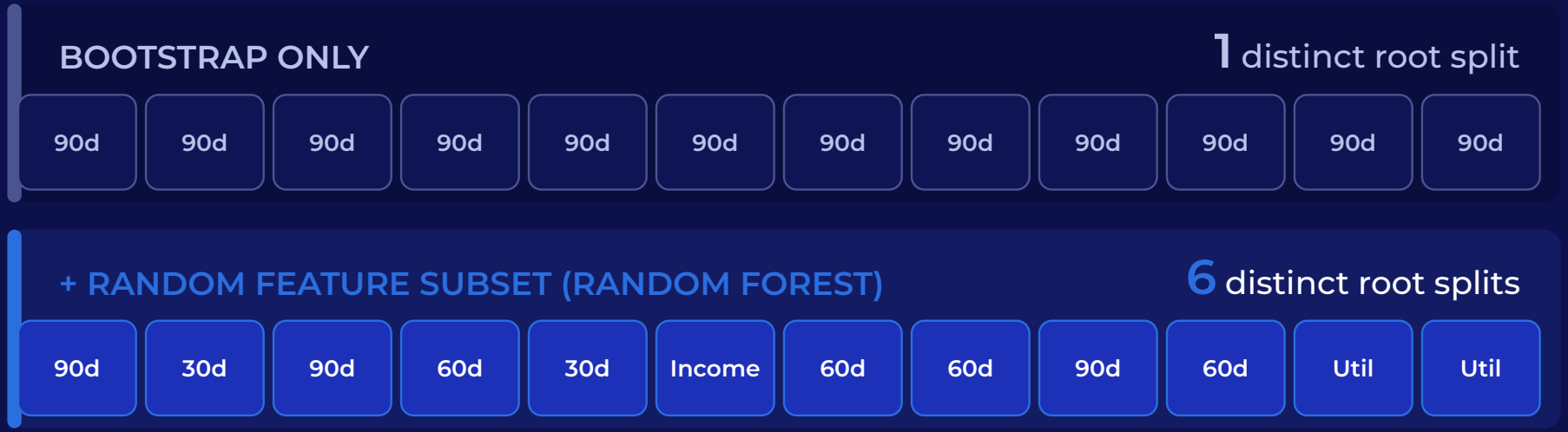
Bagging = Bootstrap + Aggregating

- draw n samples **with replacement** from the training set
- grow one tree per sample, independently, in parallel
- average their predictions
- each tree sees slightly different data, so each is wrong differently (back to ρ)
- the $\approx$37% of rows left out of each sample give you a free out-of-bag score

Random Forest = bagging + one more trick

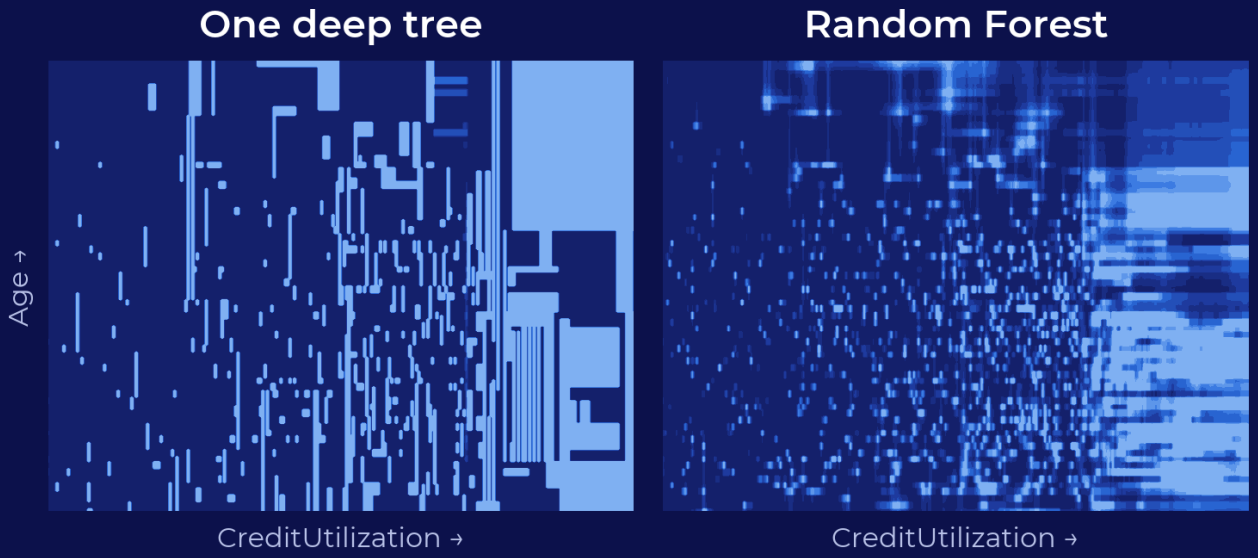
- at every split, a tree may only consider $\sqrt{p}$ **randomly chosen** features
- this deliberately makes each individual tree **worse**
- what you buy is a lower ρ, and the equation says that is what actually pays
- textbooks credit the bootstrap for the diversity. Let us check that.

Why Random Forest needs subsampling



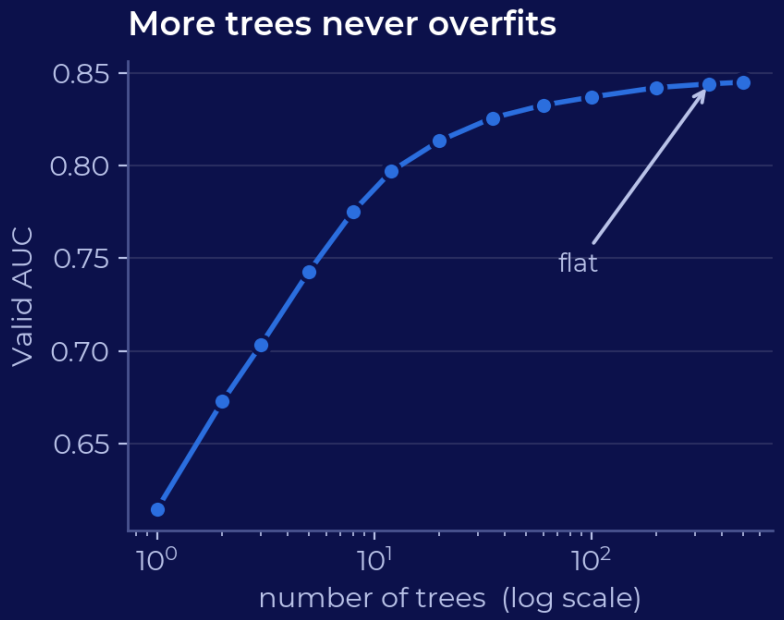
12 trees on 12 bootstrap samples of the same 105,000 rows.
90d = `Late90Days` · 60d = `Late60to89Days` · 30d = `Late30to59Days` · Util = `CreditUtilization` · Income = `MonthlyIncome`

One tree is jagged. Three hundred are smooth.



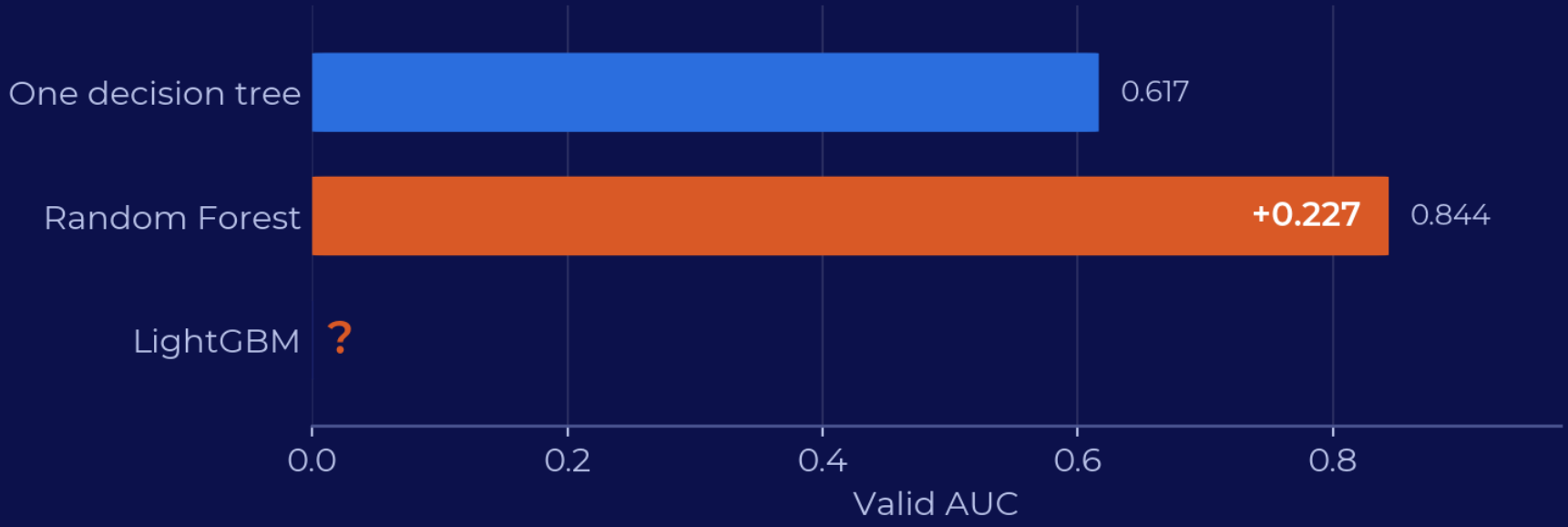
lighter = higher predicted risk

Is 3,000 trees better than 300?

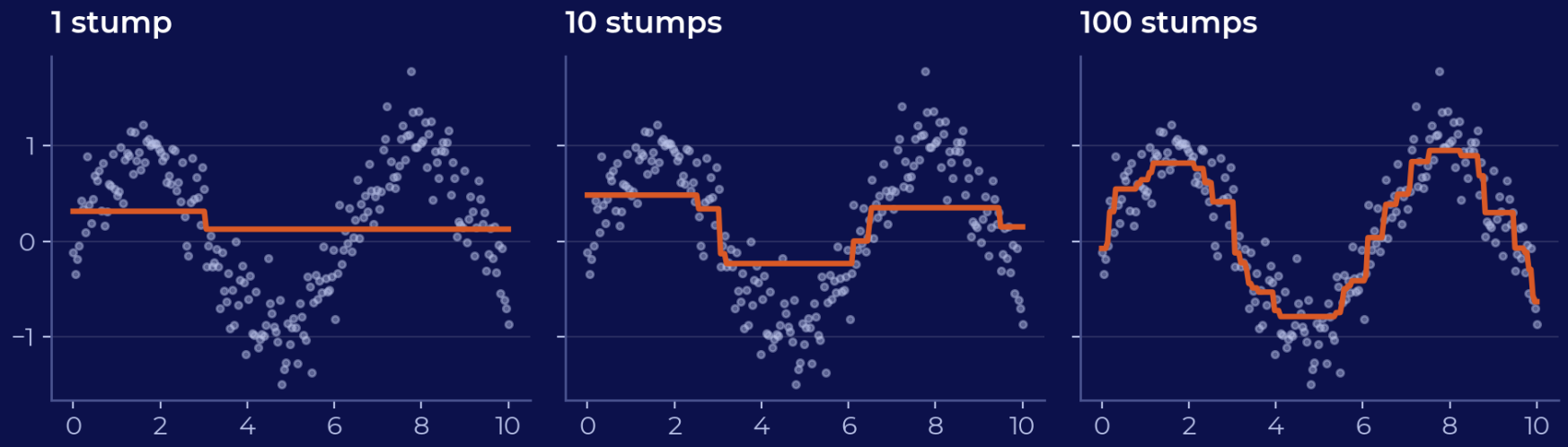


It stops helping. It never starts hurting.

Scoreboard: averaging bought +0.227



Boosting: each model fixes the last one's mistakes



Boosting, by hand

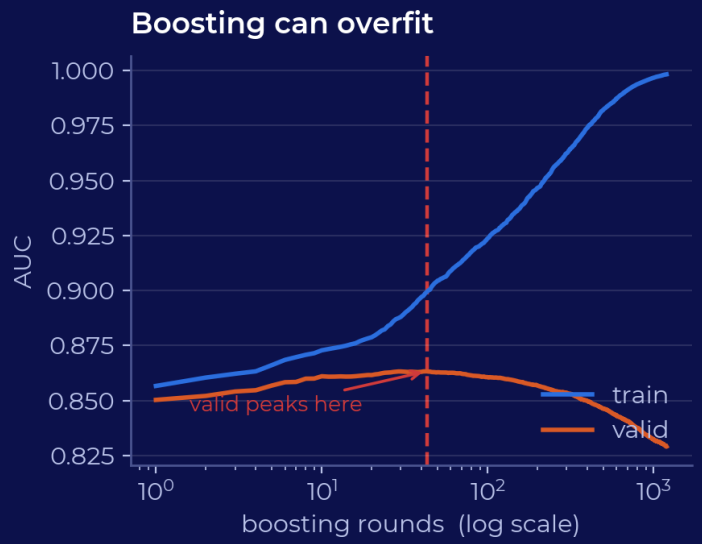
	y	round 1	error left	round 2	error left
A	10	7.0	+3.0	8.5	+1.5
B	6	7.0	−1.0	6.5	−0.5
C	12	7.0	+5.0	9.5	+2.5
D	4	7.0	−3.0	5.5	−1.5

Round 2 never sees **y**. It is trained to predict the error column.

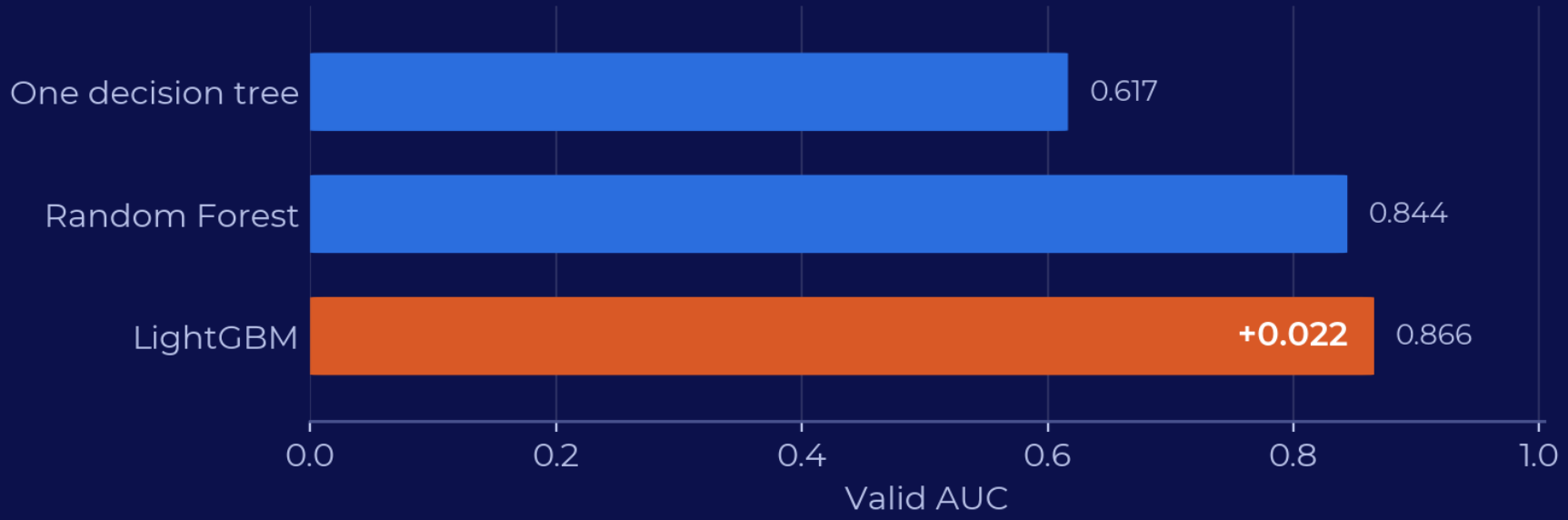
Thirty years of the same idea

- **AdaBoost** (1995): re-weight the rows you got wrong
- **Gradient Boosting**: fit the gradient of any loss
- **XGBoost**: regularisation + a fast implementation
- **LightGBM**: leaf-wise growth, histogram binning, much faster
- **CatBoost**: categorical features without the drama

The same two charts, opposite behaviour



Scoreboard: boosting bought +0.022



Bagging is safe, boosting usually wins

	Bagging / RF	Boosting	Stacking
How it helps	Errors cancel	Errors get fixed	A model learns the blend
Training	Parallel	Sequential	Two levels
Overfits with more models?	No	Yes	Leaks without OOF
Tuning needed	Little	A lot	A lot
Robust to noisy labels	More	Less	Depends on level 0
Typical tabular winner	Good	Usually wins	Competition margins

Quick check: three models, one vote

Three models. Each one is right **70% of the time**, and each is wrong in different places. You take a majority vote, best of three. **How often is the vote right?**

- **A** 70%: a vote cannot beat the models inside it
- **B** 78%: a small but real gain
- **C** 90%: most of the errors disappear
- **D** 100%: two right answers always outvote one wrong one

Type A, B, C or D in chat.

Quick check: three models, one vote

Three models. Each one is right **70% of the time**, and each is wrong in different places. You take a majority vote, best of three. **How often is the vote right?**

- **A** 70%: a vote cannot beat the models inside it
- **B** 78%: a small but real gain
- **C** 90%: most of the errors disappear
- **D** 100%: two right answers always outvote one wrong one

78.4%: when at least two of the three are right.

all three right: $0.7^3 = 0.343$ · exactly two right: $3 \times 0.7^2 \times 0.3 = 0.441$ · together **0.784**

Only because they are wrong in different places. Three copies of one model still vote 70%.

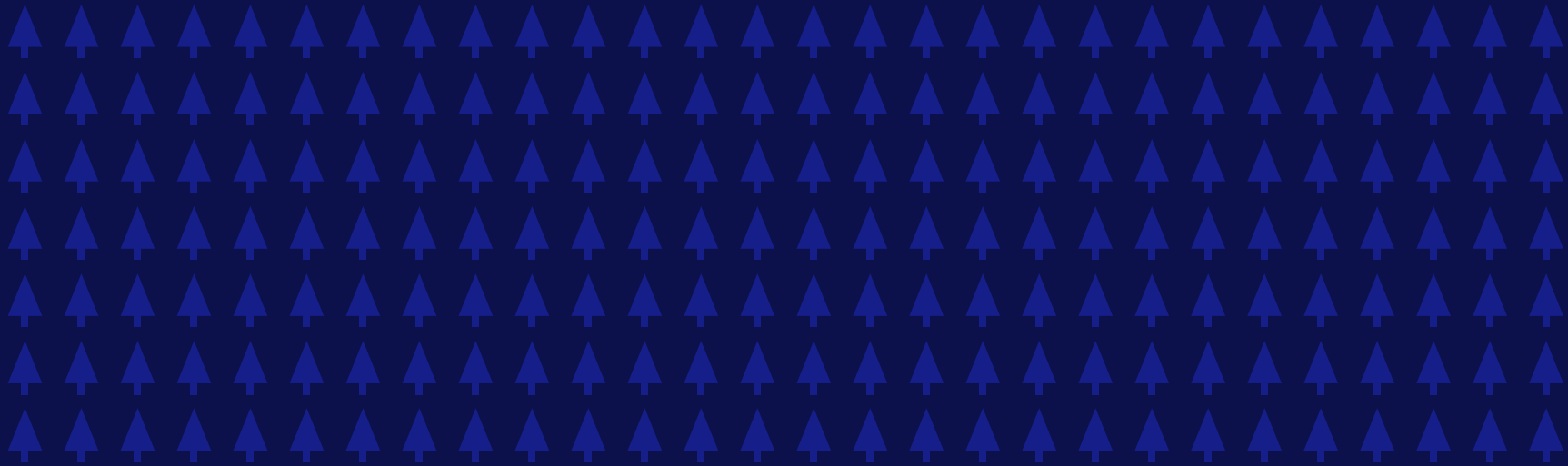
Stacking: learn the combination

- level 0: Random Forest, LightGBM, logistic regression, kNN
- their predictions become the features of a level-1 model
- mixing model families buys the most expensive diversity there is, logistic regression is wrong in completely different places than a tree
- the trap: train the meta-model on in-fold predictions and you have leakage. Use out-of-fold predictions.

Ensembles are not magic

- garbage in, garbage out: averaging will not fix wrong labels
- 500 trees is 500$\times$ the memory and the latency
- diversity cannot be forced: ten LightGBMs with different seeds $\approx$ one LightGBM
- +0.3% AUC is not worth 40$\times$ slower inference
- deep learning still wins on images, text and audio. Boosting wins on tables.

196 trees. AUC 0.866.



Now explain it to the applicant you rejected.

EXPLAINABLE AI

From “what” to “why”

BANK APP now

Why?

Sent 19.02. Still unanswered.

You built the model. You cannot answer it.

Who is asking?

One model, three audiences, three different right answers.

THE APPLICANT

“Why me, and what do I change?”

Needs one case, in a sentence.

THE REGULATOR

“Prove this is not discriminatory.”

Needs the whole model, on the record.

YOU

“Why is it bad on this segment?”

Needs enough detail to debug.

The whole field, on two axes

	GLOBAL whole-model behaviour	LOCAL one prediction
INTRINSIC reads directly	linear coefficients	one tree's path
POST-HOC explained after	permutation importance, PDP	LIME, SHAP

Our boosted forest lives in the bottom-right box.

Everyone's first move

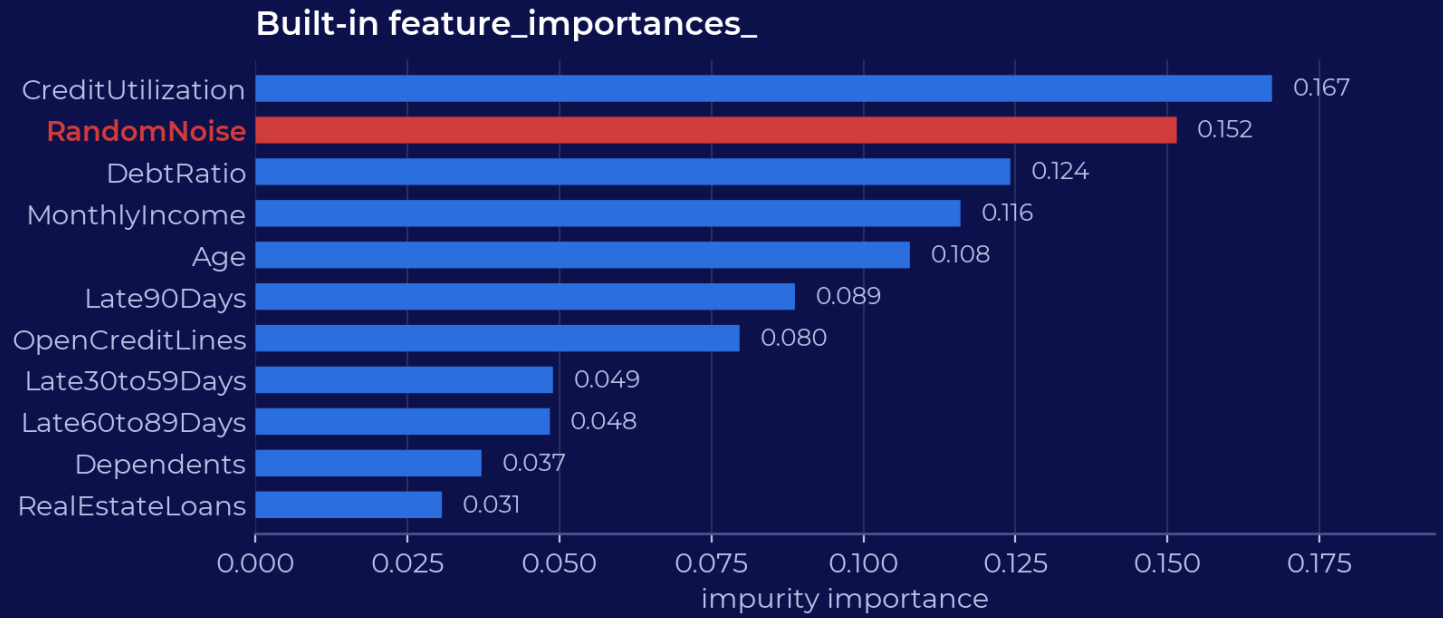
Rank the features by the model's built-in `feature_importances_`. One line of code.

```
df["kolom_sampah"] = np.random.rand(len(df))
```

I have just added a column of **pure noise**. Zero information about who defaults.

Of 11 features, what rank does it get?

Rank 2 of 11



A column of pure noise outranks eight real features.

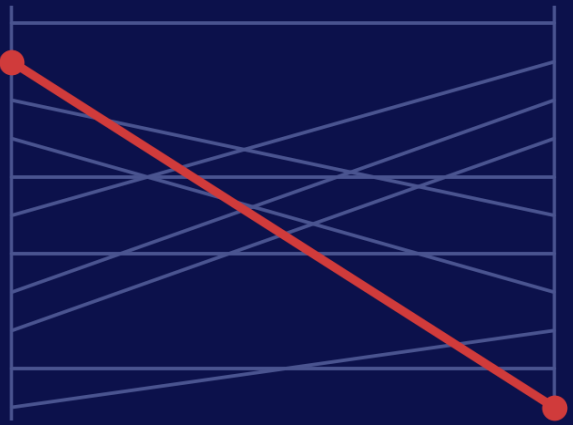
Why it lies

- **It is biased toward high-cardinality features.** A continuous column offers thousands of split points, offering thousands of chances to reduce impurity by luck. A random float is the champion of this.
- **It is computed on training data.** So it measures how hard the model **memorised**, not how useful the feature is.
- Not a bug: this is genuinely what the number means.

Shuffle it on held-out data instead

Gini impurity (in-sample)

1. CreditUtilization
2. RandomNoise
3. DebtRatio
4. MonthlyIncome
5. Age
6. Late90Days
7. OpenCreditLines
8. Late30to59Days
9. Late60to89Days
10. Dependents
11. RealEstateLoans



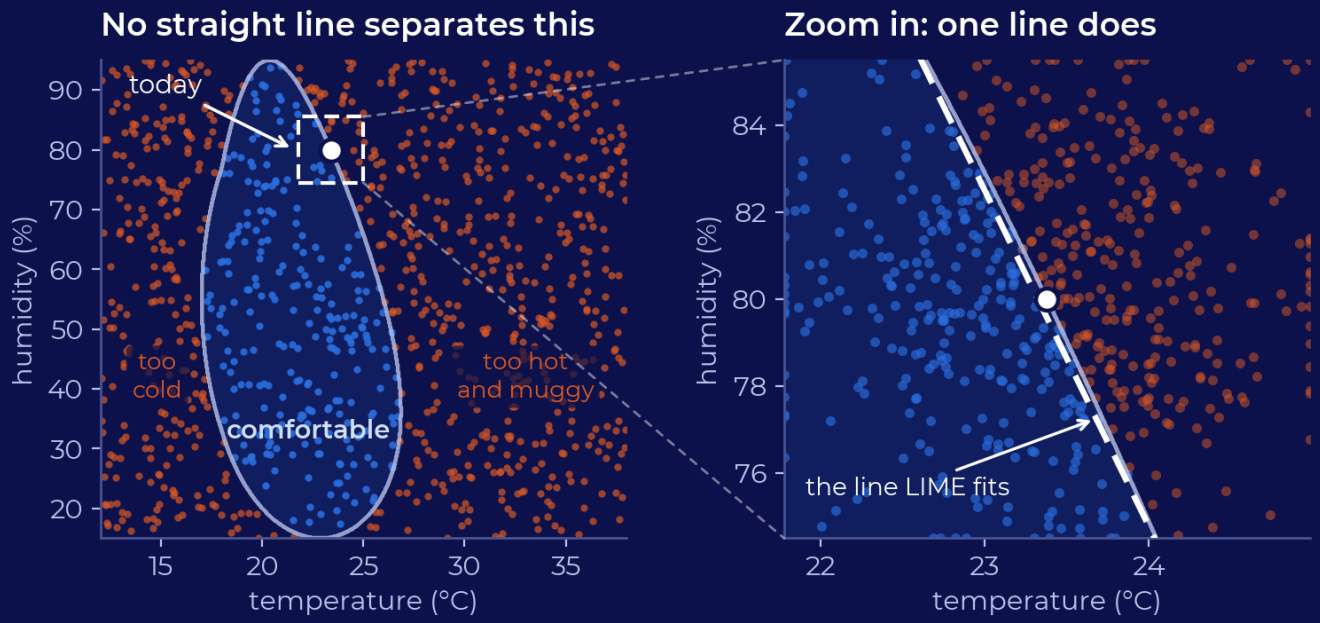
Permutation importance (held-out)

1. CreditUtilization
2. Late90Days
3. Late30to59Days
4. Late60to89Days
5. Age
6. DebtRatio
7. OpenCreditLines
8. MonthlyIncome
9. RealEstateLoans
10. Dependents
11. RandomNoise

LIME: zoom in far enough and anything is linear

- perturb the data around **one** point, fit a simple model there
- the wolf-vs-husky classifier was reading the **snow**, not the animal
- a model that is right for the wrong reason collapses on real data: explanation is debugging, not just compliance
- unstable: run it twice, get two different explanations

Why “local” is the whole trick



Globally curved. Around one point, a straight line is close enough.

LIME, by hand

	temp	humidity	model says	distance	weight
today	23.4 °C	80%	0.50	0.00	1.00
fake day 1	22.5 °C	78%	0.78	0.63	0.52
fake day 2	24.0 °C	83%	0.18	0.62	0.53
fake day 3	24.5 °C	76%	0.29	1.05	0.16
fake day 4	23.1 °C	85%	0.38	0.94	0.24

550 fake days in all. Weight falls off with distance, so the far ones barely count.

Fit a line through **model says**, weighted by **weight**:

$$g = 0.48 - 0.29 \times (\text{°C above today}) - 0.038 \times (\text{\% above today})$$

The same table, as four symbols

symbol	what it was on the last slide
x	today: the one row we are explaining: 23.4 °C, 80%
Z	the 550 fake days, written as offsets $z_i - x$
y	the **model says** column: $y_i = f(z_i)$
w_i	the **weight** column: $w_i = \exp\left(-D(x, z_i)^2 / 2\sigma^2\right)$

D is distance in standardised units; $\sigma = 0.55$ is the kernel width, how wide “nearby” is allowed to be.

The same table, as three matrices

The five rows above, then 545 more. Column one is the intercept.

$$Z = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -0.86 & -1.82 \\ 1 & 0.59 & 2.73 \\ 1 & 1.09 & -4.40 \\ 1 & -0.31 & 5.04 \\ \vdots & \vdots & \vdots \end{bmatrix} \quad y = \begin{bmatrix} 0.50 \\ 0.78 \\ 0.18 \\ 0.29 \\ 0.38 \\ \vdots \end{bmatrix} \quad W = \begin{bmatrix} 1.00 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0.52 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0.53 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0.16 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0.24 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Z is $N \times 3$, y is $N \times 1$, and W is $N \times N$ but diagonal: nothing off the diagonal, so no fake day is ever compared to another.

There is a formula for that line

No search, no gradient descent. Weighted least squares, closed form:

$$\hat{\beta} = (Z^\top W Z)^{-1} Z^\top W y$$

- $g(z) = z^\top \hat{\beta}$, and $\hat{\beta} = (0.484,\ -0.293,\ -0.0379)$, the three numbers under the table. That vector **is** the explanation.
- Intercept 0.484 against the model's own 0.50 at today.

...and the loss it is the answer to

That $\hat{\beta}$ is exactly the one that minimises (the paper writes w_i as $\pi_x(z_i)$):

$$L(f, g, \pi_x) = \sum_{i=1}^{N} w_i \big(f(z_i) - g(z_i)\big)^2 + \Omega(g)$$

- $\Omega(g) = \lambda \|\beta\|^2$ only adds $+\lambda I$ inside that inverse.
- "Keep just K features" is not differentiable, so LIME picks the columns first, then solves this.

LIME, in general

The same loss, with the numbers gone and g free to be any simple model:

$$\xi(x) = \arg\min_{g \in G} \sum_{z} \pi_x(z) \big(f(z) - g(z)\big)^2 + \Omega(g)$$

- $f(z)$: what the model answered; the **model says** column. g is the line, and its coefficients **are** the explanation.
- $\pi_x(z)$: closeness to today; the **weight** column, and the whole meaning of **local**.
- $\Omega(g)$: a charge for complexity. Two features need no pruning; a thousand columns do.
- Nothing here opens the model. It is only ever asked for answers: which is why LIME works on anything.

SHAP: a fair split of the credit

Three people finish a project and are paid 90 million. What is a fair split? Shapley (1953): average each person's marginal contribution over **every possible joining order**.

Swap “people” for **features** and “money” for **the distance of this prediction from the average one**.

$$\text{base value} + \sum \text{SHAP values} = \text{this exact prediction}$$

Shapley, by hand

worth alone: A 30, B 20, C 10 · **pairs:** AB 60, AC 50, BC 40 · **all three:** 90

joining order	A	B	C	hands out
A B C	30	30	30	90
A C B	30	40	20	90
B A C	40	20	30	90
B C A	50	20	20	90
C A B	40	40	10	90
C B A	50	30	10	90
average	40	30	20	90

Three rows, in slow motion

Every cell is one subtraction: the room after you, minus the room before.

A B C

A finds $\emptyset$
$v(A) - v(\emptyset) = 30 - 0 = \mathbf{30}$

B finds $\{A\}$
$v(AB) - v(A) = 60 - 30 = \mathbf{30}$

C finds $\{A, B\}$
$v(ABC) - v(AB) = 90 - 60 = \mathbf{30}$

B C A

B finds $\emptyset$
$v(B) - v(\emptyset) = 20 - 0 = \mathbf{20}$

C finds $\{B\}$
$v(BC) - v(B) = 40 - 20 = \mathbf{20}$

A finds $\{B, C\}$
$v(ABC) - v(BC) = 90 - 40 = \mathbf{50}$

C A B

C finds $\emptyset$
$v(C) - v(\emptyset) = 10 - 0 = \mathbf{10}$

A finds $\{C\}$
$v(AC) - v(C) = 50 - 10 = \mathbf{40}$

B finds $\{A, C\}$
$v(ABC) - v(AC) = 90 - 50 = \mathbf{40}$

$v(\emptyset) = 0$; $v(A) = 30$, $v(B) = 20$, $v(C) = 10$; $v(AB) = 60$, $v(AC) = 50$, $v(BC) = 40$; $v(ABC) = 90$.

Shapley, in general

The table you just read, written once instead of six times:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!} \left(v(S \cup \{i\}) - v(S) \right)$$

- the bracket is one cell of the table: what i adds to a room that already holds S
- the fraction counts how many joining orders build exactly that room: it is the "average over six orders" written properly
- 2^n subsets. Ten features is 1,024 and fine; a hundred features is past counting.
- Shapley (1953): this is the **only** split that is efficient, symmetric, dummy-proof and additive. Not one reasonable option: the only one.

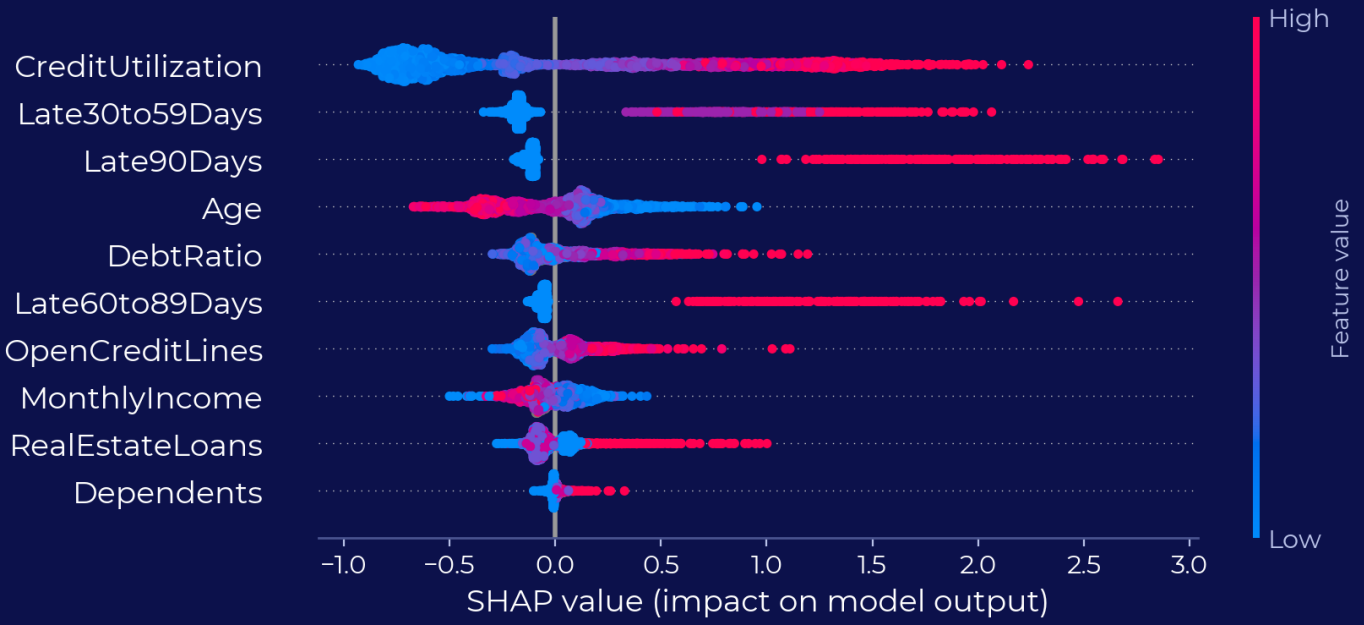
From players to features

the game	your model
a player	a feature
the payout, 90	how far this prediction sits from the average one
a coalition S	the features you are allowed to look at
$v(S)$	the model's output knowing only those, the rest averaged out
ϕ_i	the SHAP value of feature i
efficiency	**local accuracy**: base value $+ \sum \phi_i =$ this prediction, exactly

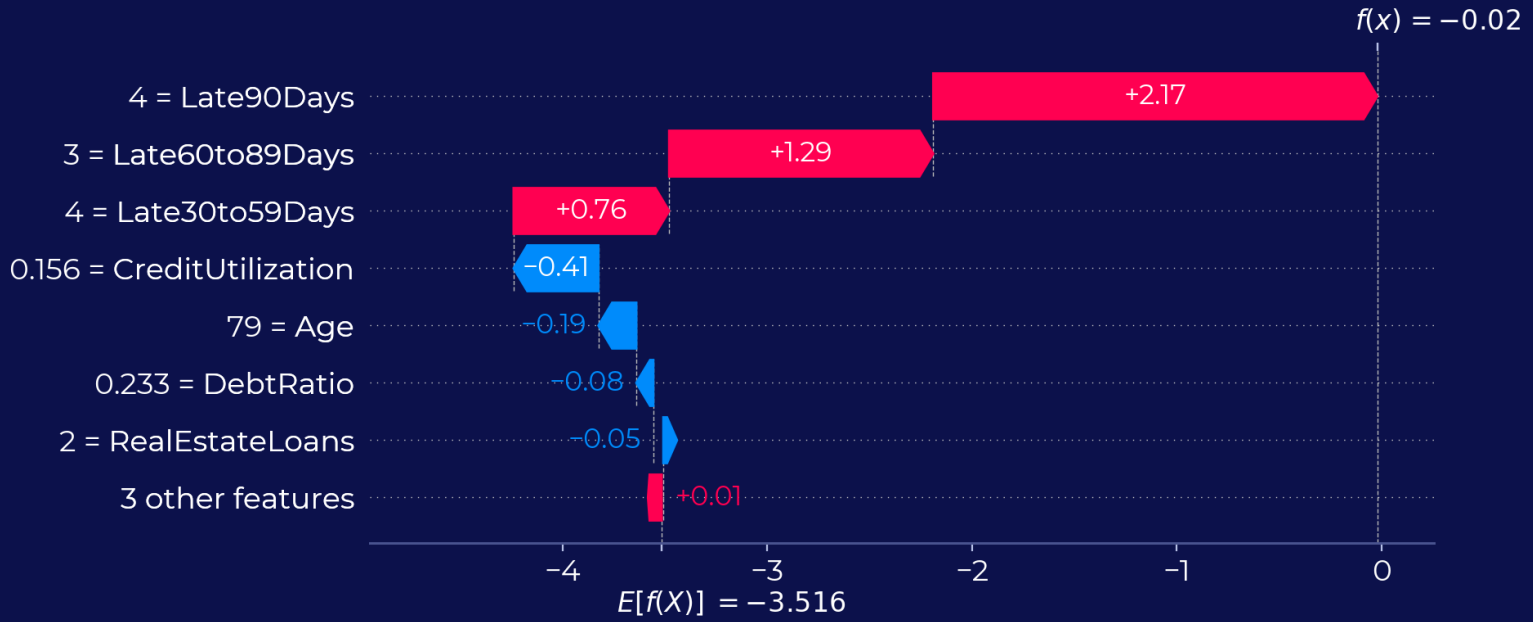
Why SHAP fits our models in particular

- exact Shapley values need all 2^p feature subsets (infeasible)
- **TreeSHAP** computes them exactly, in polynomial time, for tree ensembles
- **local accuracy**: the contributions sum to the prediction
- **consistency**: a feature that matters more never scores less
- the hardest models to explain have the fastest exact explainer. That is not a coincidence in tonight's title.

How to read this



One applicant, every contribution



From plot to sentence

BANK APP now

Your loan application was declined.

Main factors: **4 payments 90+ days late**, **3 payments 60–89 days late**, and 4 payments 30–59 days late.

Counting in your favour: you use only **16% of your available credit**, and you have a long credit history.

Estimated risk 49.5%, about 7× the average applicant.

Where explanations mislead

- **Explanation is not causation.** SHAP explains the **model**, not the world. “Lower your debt” is a statement about the model, not proven advice.
- **Correlated features split the credit.** Our three late-payment columns each took a share, so none looks as important as the behaviour really is.
- **Explanations can be attacked.** Slack et al. (2020) built biased models whose LIME and SHAP output looks clean.

“Stop explaining black box models for high-stakes decisions and use interpretable models instead”

Cynthia Rudin, 2019

Monday morning

- **Baseline**: one simple model, write the number down
- **Ensemble**: Random Forest first (cheap, safe), boosting if the number really matters
- **Validate honestly**: early stopping, out-of-fold, never trust a training score
- **Explain**: permutation importance globally, SHAP locally
- **Check**: does the explanation make sense to someone who knows the domain? If not, the model is wrong, not them.

The same model, twice

19.02

BANK APP now

Your loan application was declined.

Ref: `model_v3.2`

No further explanation is available.

20.50

BANK APP now

Your loan application was declined.

4 payments 90+ days late; 3 payments 60–89 days late.

In your favour: only 16% credit utilisation.

Estimated risk 49.5%, about 7× average.

Where to go next

- **Read**: *Interpretable Machine Learning*, Christoph Molnar. Free online. This is the book.
- **Run**: tonight's notebook: every chart here, reproducible. Swap in your own data; the five steps hold anywhere.
- **Try**: `shap`, `sklearn.inspection`, `interpret` (EBM)
- **Papers**: Rudin (2019); Lundberg & Lee (2017)

QUESTIONS

Diversity, not count

feature subsampling gave the trees their disagreement, not the bootstrap

Built-in importance lies

pure noise ranked 2nd of 11: use permutation importance or SHAP

Explanation is the product

if you cannot explain it, you are not finished

AI Engineer & Researcher - Darrel Danadyaksa Poli