DIGITAL FINANCE

This project has received funding from the Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101119635

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

State Secretariat for Education, Research and Innovation SERI

Funded by the European Union

MARIE CURIE ACTIONS

DIGITAL

Tutorial on Explainable Reinforcement Learning (XRL)

Explaining black-box trading models

DIGITAL

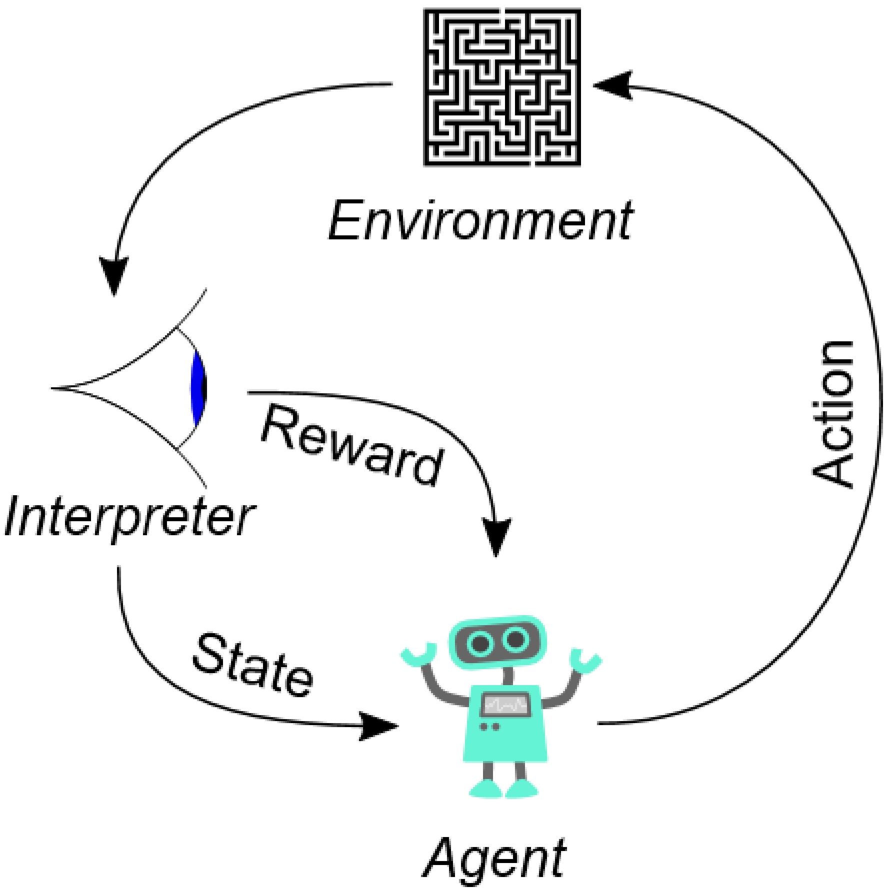
Funded by
the European Union

Why XRL in finance?

- Finance demands trust and transparency
- Black-box models (RL, deep nets) hard to interpret
- Regulators and risk managers require explainability
- Goal: bridge performance and interpretability

DIGITAL

Recap: Reinforcement Learning



- Framework for sequential decision making
- Agent interacts with environment
- Observes **state** S, chooses **action** a, receives **reward** $R(s,a)$
- Learns **policy** π to maximize expected long-term reward

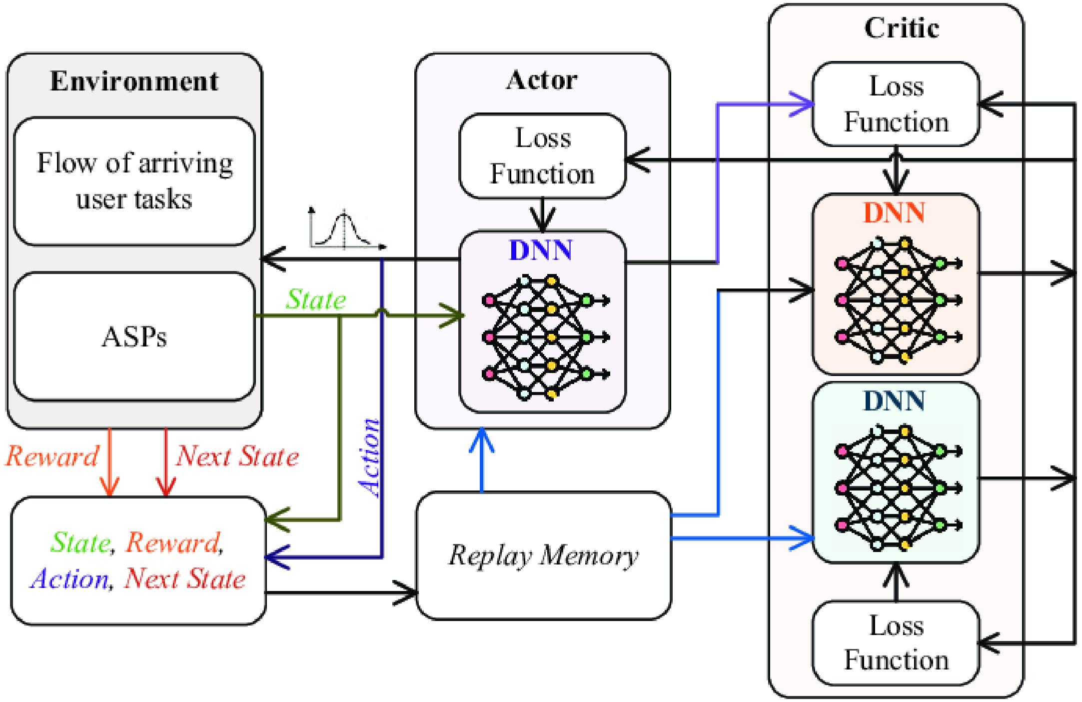
- **Value-based methods**: capture expected long-term reward in value function $V(s')$
- **Policy-based methods**: directly adjust policy π_θ parameterized by θ

- Example in finance: portfolio allocation, trading strategies, market making

DIGITAL

Soft Actor Critic (SAC) algorithm

- State-of-the-art RL algorithm
  - In family of deep deterministic policy gradient algorithms
  - Differentiates updates through the critic
- Learns stochastic policy with entropy regularization
- Balances exploration and exploitation
  - Relatively high degree of exploration due to entropy bonus, suitable for environments with much uncertainty
- Strong benchmark for continuous finance tasks



DIGITAL

The Explainability Gap

- RL policies are black-box neural networks
  - SAC has opaque architecture with multiple neural networks
  - Complex interplay between policy, (long-term) rewards and state trajectory
- Hard to understand 'why' actions are chosen
  - Reward signal may not be directly observable
  - Decision made based on complex latent representations
  - Environment itself is noisy and nonstationary
- In finance: unacceptable for risk oversight
  - Need human-readable explanations

Decision Trees as Surrogate Explainers

- Train a decision tree on (state, action) pairs generated by the agent
- Tree outputs simple if–then rules that approximate policy
- Useful for quick audits and communicating strategy to non-technical stakeholders

DIGITAL

How to Read Decision Trees

1. Pick a leaf — follow splits from root to leaf → yields an if–then rule
2. Translate numeric thresholds to readable statements (e.g., 'volatility > 0.02')
3. Check how many samples follow this rule (support) and the surrogate's fidelity there
    - **Support:** the subset of data over which the explanation is valid.
    - **Fidelity:** how accurately the explanation matches the original model’s predictions.

- Use example states to validate ('show me a trade where this rule triggers')

DIGITAL

Example of Decision Tree Output

```
|--- GLD_mom_20 <= 0.00
|   |--- GLD_mom_20 <= -0.03
|   |   |--- GLD_mom_20 <= -0.05
|   |   |   |--- value: [0.11, 0.48, 0.19]
|   |   |--- GLD_mom_20 >  -0.05
|   |   |   |--- value: [0.12, 0.45, 0.21]
|   |--- GLD_mom_20 >  -0.03
|   |   |--- SPY_dd_60 <= -0.05
|   |   |   |--- value: [0.17, 0.39, 0.24]
|   |   |--- SPY_dd_60 >  -0.05
|   |   |   |--- value: [0.14, 0.43, 0.23]
|--- GLD_mom_20 >  0.00
|   |--- SPY_vol_20 <= 0.02
|   |   |--- GLD_dd_60 <= -0.04
|   |   |   |--- value: [0.17, 0.39, 0.24]
|   |   |--- GLD_dd_60 >  -0.04
|   |   |   |--- value: [0.20, 0.36, 0.24]
|   |--- SPY_vol_20 >  0.02
|   |   |--- SPY_vol_20 <= 0.03
|   |   |   |--- value: [0.16, 0.44, 0.19]
|   |   |--- SPY_vol_20 >  0.03
|   |   |   |--- value: [0.11, 0.51, 0.16]
```

DIGITAL

SHAP: Local & Global Feature Attribution

- SHAP assigns additive contributions of features to a prediction
- Local explanations: why this single decision was made at time t
  - Useful in nonstationary environments, possible to show evolution over time
- Global explanations: which features matter across many decisions
  - Long-term perspective on explainable decisions

DIGITAL

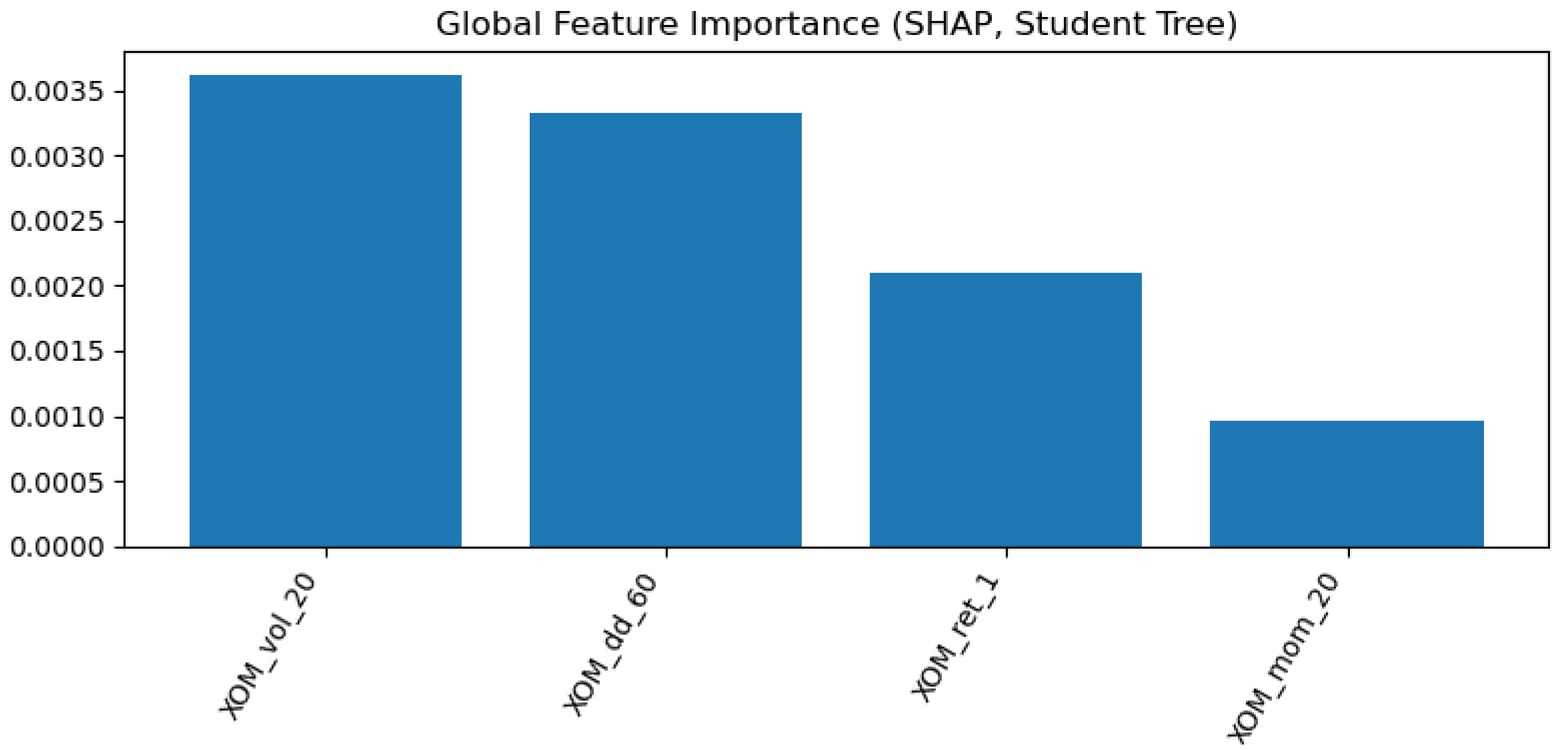
SHAP in Practice — Options

- If surrogate is a tree: use shap.TreeExplainer (fast, exact for many tree models)
  - **TreeExplainer:** fast, exact feature attributions for tree-based models (e.g., XGBoost, Random Forest).
- If explaining the NN policy directly: use shap.KernelExplainer or DeepExplainer (slower)
  - **KernelExplainer:** model-agnostic, perturbs inputs to estimate feature contributions, slower.
  - **DeepExplainer:** uses NN structure for faster, accurate feature attributions, NN-only.

- Typical workflow: collect states → choose model → compute SHAP → show summary & force plots

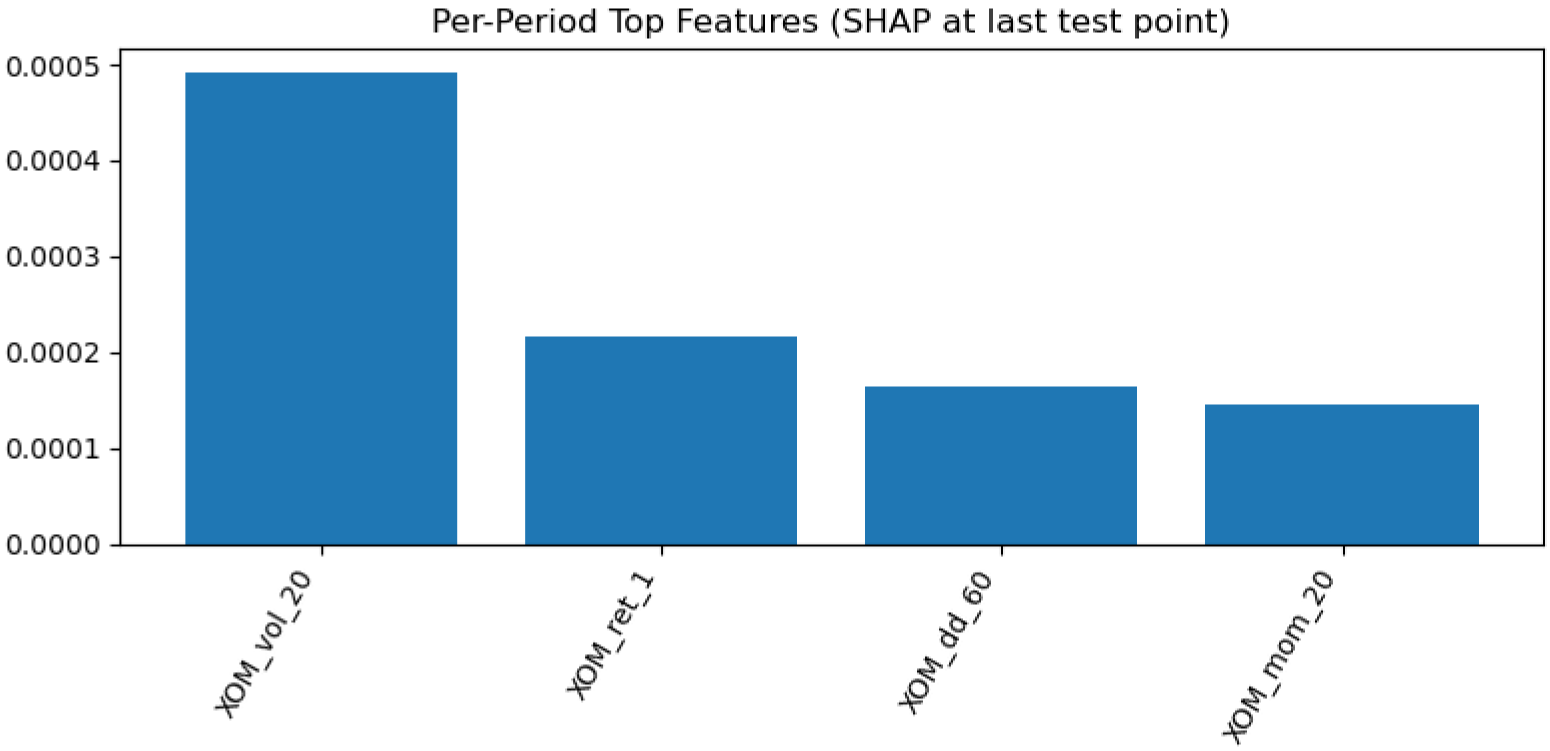
DIGITAL

SHAP values (global average)



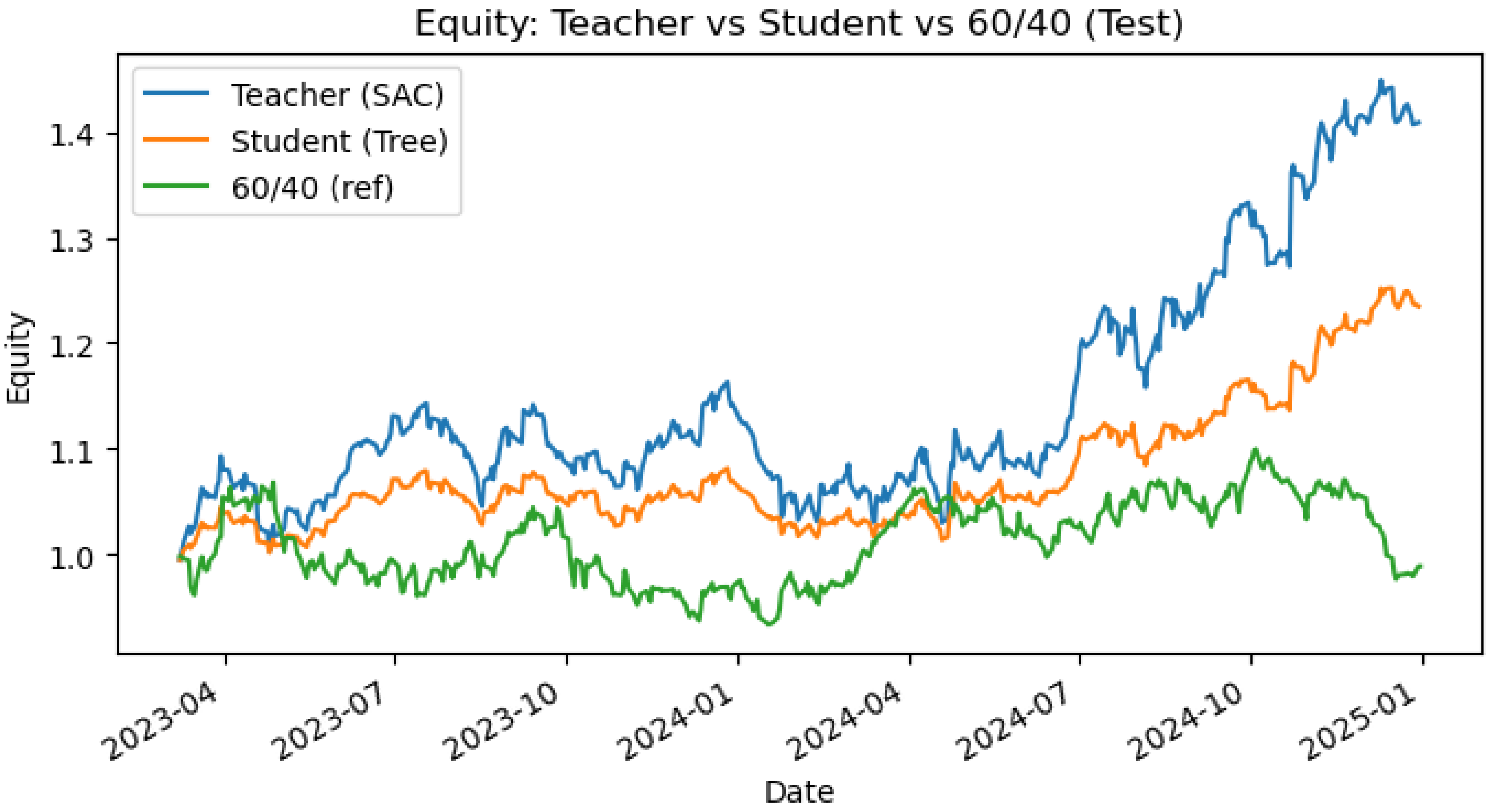
DIGITAL

SHAP values (local at time t)



DIGITAL

Example performance



DIGITAL

Demo Walkthrough — Notebook Steps

1. Load environment and train SAC agent
2. Generate rollouts → collect (state, action) pairs
3. Fit decision tree surrogate (report fidelity metrics)
4. Compute SHAP values for surrogate (or for NN with KernelExplainer)
5. Visualize: tree plot, SHAP summary, local force plot, action trace vs rule triggers

DIGITAL

Key Takeaways

- Surrogate trees simplify, measure fidelity and be honest about gaps
- SHAP helps both local and global interpretation but has limits (correlated features, runtime)
- Explainability is a communication tool, not a proof of correctness

DIGITAL

DIGITAL

Funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Horizon Europe: Marie Skłodowska-Curie Actions. Neither the European Union nor the granting authority can be held responsible for them.

DIGITAL

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101119635