PIMP for Feature Importance

A practical guide to understanding what drives your data science models

Attendance: "Permutation"

What is PIMP?

Permutation Importance in Machine Learning

A way to determine which features have the MOST impact on your model's predictions

---

How: Measure performance loss when you randomly shuffle a feature's values

Why Use PIMP?

Model-Agnostic

Works with any machine learning model—from linear regression to deep neural networks

Captures Complexity

Identifies feature interactions and non-linear relationships that simpler methods miss

Stakeholder-Friendly

Easy to interpret and explain to non-technical audiences

Common Use Cases

Feature Selection

Reduce dimensionality while preserving model performance

Model Debugging

Validate assumptions and catch unexpected dependencies

Business Insights

Understand which factors truly drive outcomes

Data Quality

Identify problematic features or data issues

How PIMP Works: The Process

Train & Baseline

Train your model and calculate baseline performance on validation data

Shuffle Feature

For each feature, randomly shuffle its values while keeping all other features unchanged

Measure Impact

Calculate model performance with the shuffled feature

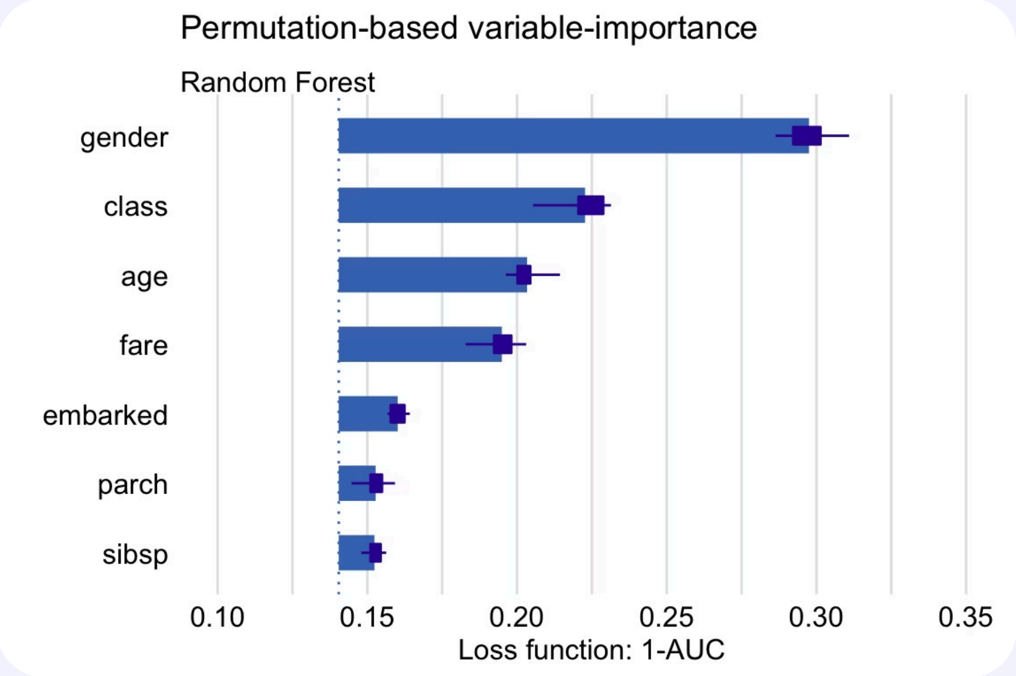
Calculate Importance

Feature importance = Baseline performance − Shuffled performance

Repeat & Stabilize

Repeat multiple times (3-50 iterations) to get stable, reliable estimates

Interpreting PIMP Scores



High Positive Importance

Model relies heavily on this feature. Shuffling it significantly hurts performance.

Action: Keep and protect this feature

Near-Zero Importance

Feature provides little value to predictions.

Action: Consider removing to reduce complexity

Negative Importance

Feature may be causing overfitting or adding noise.

Action: Strong candidate for removal

PIMP vs Other Methods

Coefficient-Based Importance

Limitation: Only works for linear models

PIMP advantage: Works for any model and captures non-linear effects

Tree-Based Importance

Limitation: Biased toward high-cardinality features

PIMP advantage: Reflects actual predictive performance

SHAP Values

Limitation: Computationally expensive at scale

PIMP advantage: Faster for large datasets; SHAP excels at instance-level insights

Best practice: Use PIMP alongside other methods for comprehensive model understanding

Best Practices

√ Do

- **Use Validation Data**

  Always use held-out validation data, never training data

- **Repeat Permutations**

  Run 10-50 iterations for stable estimates

- **Consider Cost**

  Account for computational expense on large datasets

Don't

- **Confuse with Causation**

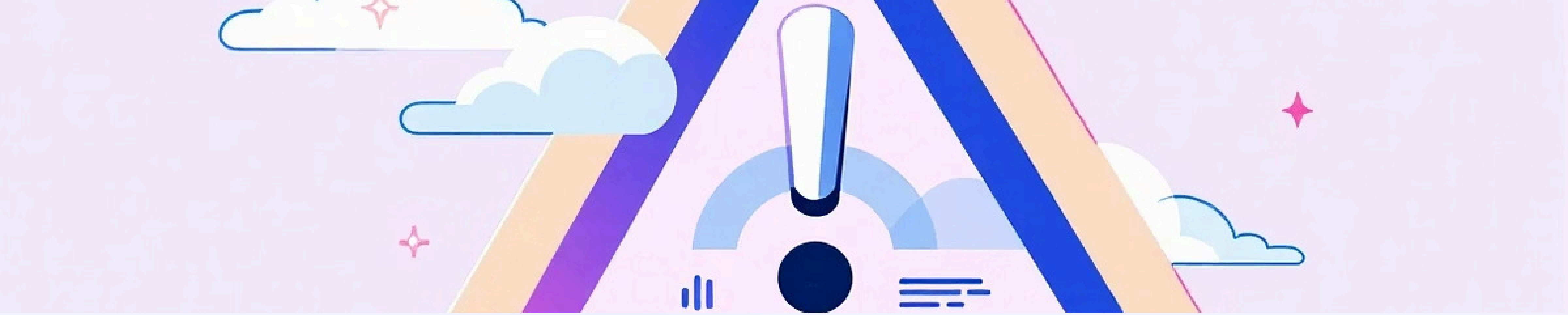
  Importance ≠ causality; correlation can mislead

- **Ignore Domain Knowledge**

  Statistical results must align with business understanding

- **Rely on PIMP Alone**

  Never use PIMP scores as sole decision factor for high-stakes choices



Common Pitfalls

Unrealistic Permutations

Shuffling may create feature combinations that don't exist in reality

Solution: Validate with domain expertise

Computational Cost

Can be expensive for large datasets or complex models

Solution: Use strategic sampling

> "The best defense against pitfalls is combining PIMP with domain knowledge and triangulating with other explainability methods"

Key Takeaways

Performance-Based Measurement

PIMP measures feature importance by evaluating performance drop when features are shuffled

Universal Applicability

Model-agnostic approach captures real predictive value across any ML algorithm

Essential Toolkit

Powerful tool for feature selection, model interpretation, and debugging

---

Simple concept, powerful insights

Google Colab

Thanks NDL!

1

Next Meeting: 10/26

Topic: SHAP