UNITED STATES MILITARY ACADEMY
WEST POINT

Interpretable Machine Learning for Requirements Development

AI4SE & SE4AI Workshop 2024

Arlington, VA

Presented by: Stephen Gillespie

17 September 2024

Importance of Requirements

> If one were to ask practitioners of any [domain engineering], they would characterize themselves as ***problem solvers***. Systems engineers, in contrast, are ***problem staters***.
>
> A. Wayne Wymore, Model-Based Systems Engineering, 1993

- We state problems through requirements.
- Poorly developed and defended requirements have led to programs that:
  - Don't meet stakeholder needs.
  - Have increased cost and delayed schedule.
  - Get cancelled.
- Examples:
  - Future Combat System (FCS) : Cancelled after 12 years and $32 Billion expended. Contradictory requirements across systems.
  - Joint Tactical Radio System (JTRS): Cancelled after 15 years and $6 Billion expended. Requirements defied physics, system was supposed to be mobile, but weighed 207 lbs, etc.
  - Future Attack Reconnaissance Aircraft (FARA): Cancelled after 5 years and $2 Billion expended. Requirement and survivability for crewed aviation reconnaissance questionable relative to drones.
- Defining requirements that are **feasible** and **defensible** is necessary for program success.

Feasible and Defensible Requirements

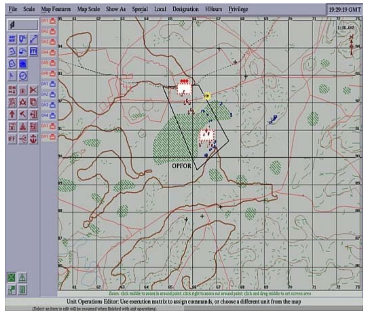
User input and desire is often insufficient to define feasible & defensible requirements.

Models & Simulations (M&S) are ways to assess requirements; but there is no single gold standard.

Live / Virtual Experimentation

System M&S

Wargames



Operational Simulations

Challenges Associated with M&S

- Models can produce large volumes of data that is challenging to understand innately.
  - Heterogeneous.
  - Unbalanced.
  - Non-linear, interdependent.
  - Open for multiple interpretations.
- Analyze and simplify model input (i.e., requirements) with outputs, i.e., Measures of Effectiveness (MOEs).
- Let:
  - y represent our MOE (i.e., model output) (e.g., P_k)
  - $\boldsymbol{x} =< x_1, x_2, \ldots x_r >$ represent our r requirements set to varying levels (e.g., sensing range or speed).
  - Want to understand relationship between MOE and requirements, i.e., $y = f(\boldsymbol{x})$
- Estimate $\hat{y} = f(\boldsymbol{x})$, using statistical or machine learning (ML) methods.

Understanding Requirements and Their Impact

Understanding requirements is not a math problem. It is a communication problem.

$$\hat{y} = f(\boldsymbol{x})$$

Understanding this function can justify and defend requirements for **decision makers**.
Understanding this function helps people synthesize information across a variety of sources.

Requirements ($\boldsymbol{x}$) → **Black Box Methods** Random Forest, Neural Network, … → MOE ($\hat{y}$)

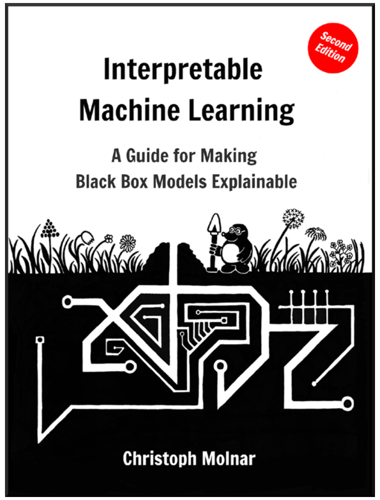
Black box methods are great:

- Don't require significant assumptions.
- Can represent complex relationships.
- Can be highly predictive.

But:

- Challenging to visualize.
- Challenging to quantify.
- Challenging to understand.
- Challenging to trust.

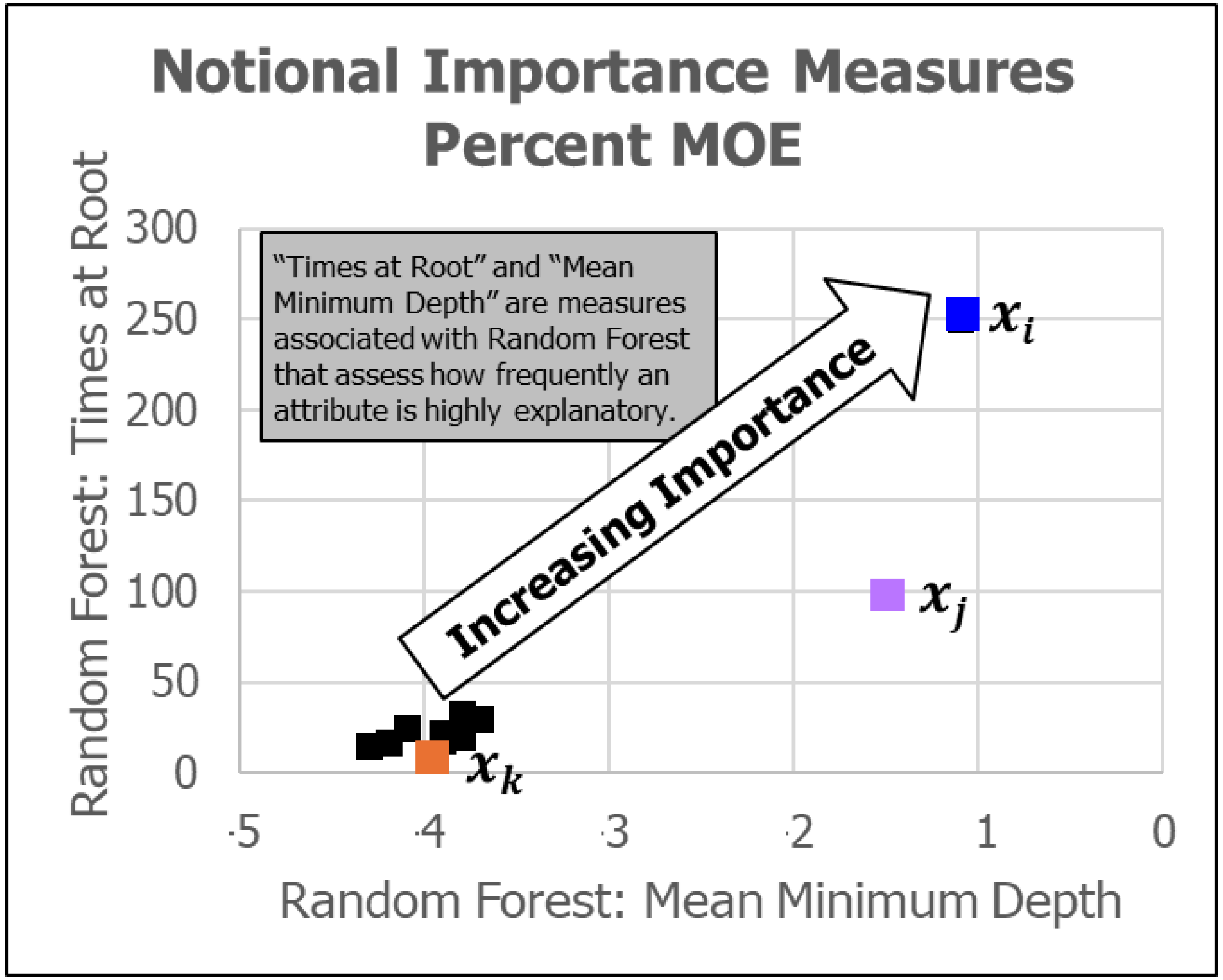
Prediction alone doesn't lead to insight.
Prediction alone doesn't lead to trust.
Results need to be interpretable.



Interpretable Machine Learning

Not All Interpretations Are Useful

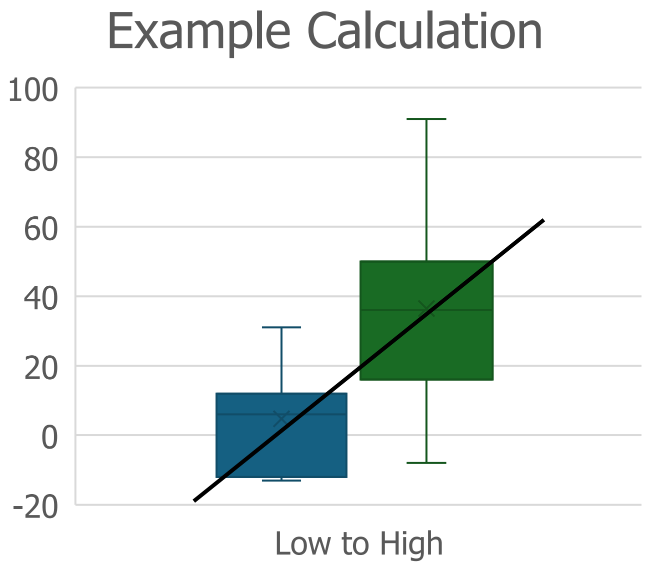
ML interpretations that require domain expertise to understand generally do not help decision makers.

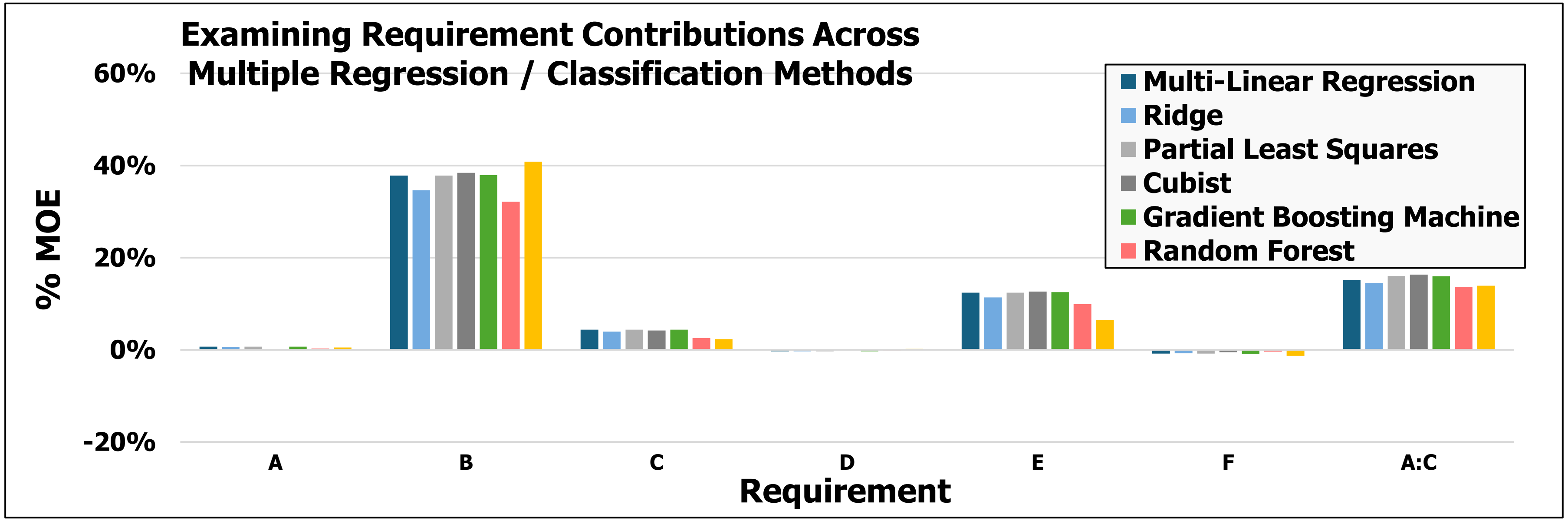


- Requires domain expertise to interpret.
- Specific to black box method (e.g., random forest).
- Quantifying differences is not apparent:
  - The most important feature (requirement) may not have a practical impact on your MOE.
  - Ordering of features doesn't necessarily provide absolute impact among

It's OK to Simplify

"Linear Impact Measure": Measuring median difference between low level of feature and high level of feature identifies and quantifies feature (requirement) importance.

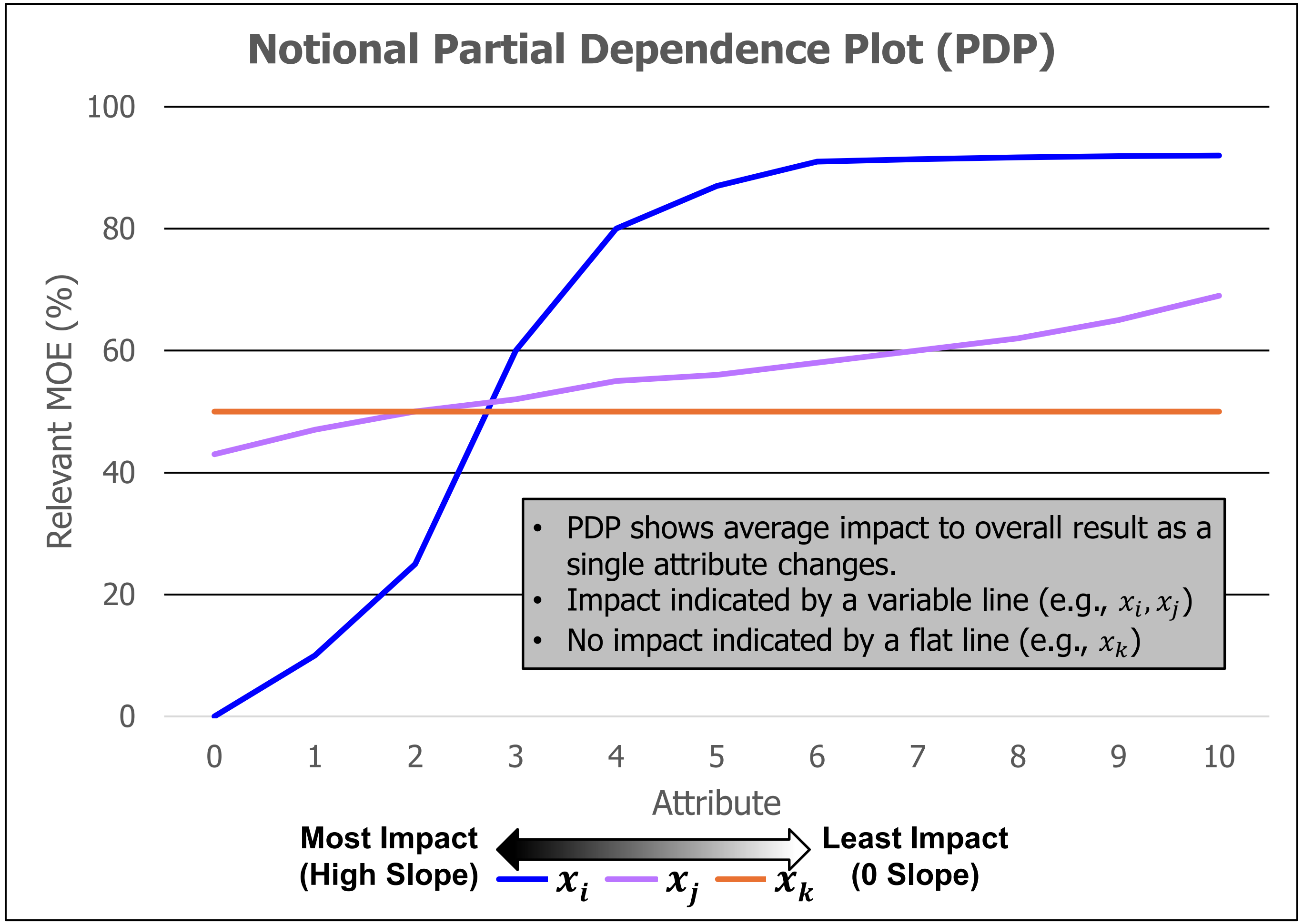




- Assess multiple requirements and set at threshold and objective (or other relevant extremes) levels.
- Take difference of median result between objective and threshold across all combinations of other requirements to assess individual requirement impact.
- Allows one to identify and quantify practically significant effects.
- Works across a range of different methods.

Partial Dependence Plots Identifying Requirements

Partial Dependence Plots (PDPs) show the median impact of varying one requirement (i.e., attribute or feature) relative to all others. Enables prioritization of requirements and visualization of impact.

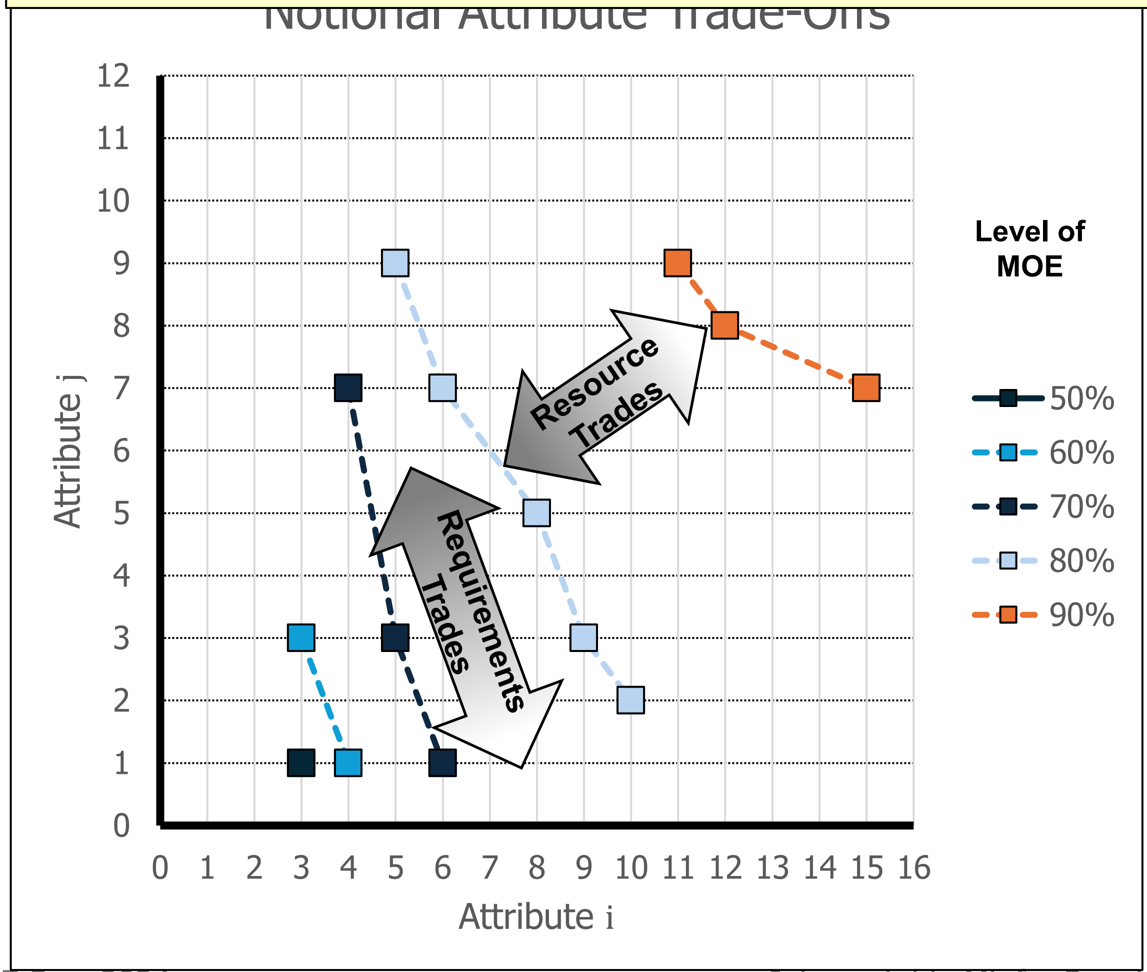


Given $\hat{y} = f(\boldsymbol{x})$ we can prioritize attributes as more or less impactful and visualize the impact.

- If f is explicit (e.g., $f = \beta_0 + \sum \beta_i x_i + \epsilon$) one can directly assess and graph impact of x_i using typical mathematical techniques, e.g., if $\beta_1 > \beta_2$ one can assess that x_1 has greater importance than x_2.
- If f is "implicit" (e.g., a Random Forest model contained in a computer object) one can use other measures associated with the method from which f was derived.
- Requirements **decisions**:
  - x_k may not need to be a requirement at all (no impact).
  - x_j may be a requirement, but threshold can be set relatively low (slight linear returns).
  - x_i should be a requirement with a threshold set at ~5 (diminishing returns afterwards).

Optimization on Sets of Requirements

Given $\hat{y} = f(\boldsymbol{x})$, a set of constraints, and a goal, one can identify optimal design points. This can highlight trades among design considerations (e.g., x_i vs x_j) or operational outcomes (e.g., survivability vs. lethality).



Method

- Attributes i and j were most significant for enabling dis-integration.
- Team optimized for a given level of MOE (y^*):
  - Minimize i and j subject to:
  - $\sum_1^6 x_i \leq L_{max}$
  - $\hat{y} \geq y^*$ = level of MOE.
- Limitation: used point estimates of y.

Key Points

- Clear relationship among i, j and y.
  - Requirements trade: increase x_i to decrease x_j for same y^*.
  - Resource trade: increase x_i, x_j to increase y.
- Analysis is independent of defined alternatives and conducted early.
- Analysis allows for "what-ifs" like:
  - What if we modify a current capability?
  - What if we accept different levels of dis-integration?
  - What if we change a requirement?

Conclusion

- Setting requirements for a program has a major financial and operational impact.
- Setting requirements is a mix of judgement and engineering.
- ML techniques can enable better prediction of requirements data.
- Interpretable ML techniques can enable better understanding of ML outputs.
- Combining well understood ML outputs with decision maker judgment can lead to more sophisticated and technically accurate setting of requirements.