Verisk

Tree-based models

NAIC June 29th, 2021

Caolan Kovach-Orr, Ph.D., CSPA
Vahid Meimand, Ph.D.

SERVE | ADD VALUE | INNOVATE

Caolan Kovach-Orr, PhD, CSPA

Senior Data Science Manager:

Data Engineering & New Technologies

BACKGROUND

Caolan Kovach-Orr leads the DENT team for ISO Insurance Analytics, Verisk Analytics. He is a Ph.D. data scientist with 12+ years of experience in big data, analytics, machine learning, and new product development. Caolan combines deep subject matter expertise, technical leadership, and a passion for engineering novel solutions to difficult problems.

Before joining Verisk, Dr. Kovach-Orr was a research data scientist who leveraged high performance cluster computing to investigate and understand how variation affects the risk of collapse for ecosystems threatened by environmental change.

PROFESSIONAL DESIGNATIONS AND ACTIVITIES

Caolan is a CAS Certified Specialist in Predictive Analytics. He takes an active role in presenting at conferences and educating Industry professionals (actuaries, regulators, data scientists) on advances in data science, machine learning, and insurance modeling.

EDUCATION

- Postdoctoral Researcher, McGill University, Canada
- PhD in Theoretical Biology, McGill University, Canada
- BSc in Ecology & Evolutionary Biology, Rutgers University, New Brunswick, NJ
- Minor in Geographic Information Systems, Rutgers University, New Brunswick, NJ

SELECTED PROJECTS

Risk Analyzer Commercial Auto Liability Symbols (RACA)
Developed a methodology to interpret Machine Learning Algorithms, which allowed Verisk to attain regulatory approval for the US's first Machine Learning Ratemaking product based on traditional variables (e.g., horsepower, airbags, etc.)

AWS Insurance Analytics Environment
Engineered CPU, GPU, and Hadoop environments for use by the Insurance Analytics teams. These environments are capable of providing a 1000-5000x speed up over SAS Grid while simultaneously reducing costs by 2+ orders of magnitude (>100x cheaper)

Voice of the Customer (NLP)
Enhancing the value of Verisk customer feedback by using advanced data engineering techniques, Natural Language Processing, and dashboard reporting

Ethical AI/ML
Leading efforts to identify and remove bias from data so that Protected Classes are not unfairly impacted by Verisk products

INTERESTS AND EXPERTISE

- Data Science | Machine Learning | Analytics
- Data Engineering | Architecture
- Rating Models | Underwriting | Internal Projects
- Big Data Architecture & Solutions
- Accelerated Computing (GPU & FPGA)
- Optimization | Design
- Complex Systems | Systems Modeling
- Solution Engineering

- Insurance: Personal and Commercial
- Academia | Consulting
- Grant and proposal writing

Vahid Meimand, PhD

Senior Lead Data Scientist

BACKGROUND

Vahid Meimand leads a data science team at ISO insurance analytics where he uses data and analytics to build products in the insurance underwriting domain. He received his Ph.D. in engineering at John Hopkins University. Vahid has significant experience in applying advanced statistical learning theory, machine learning algorithms, and visualization techniques to drive insight from data in insurance, infrastructure, energy, and health sectors. Prior to Verisk, Vahid was Lecturer at Purdue University in the school of mechanical engineering. He also performed consulting work at NBM Technology as a Senior/then Principal Engineer for 5 years. There, he led projects on surveying railroad tracks and inspecting bridges using computer vision and machine learning algorithms.

PROFESSIONAL DESIGNATIONS AND ACTIVITIES

- Johns Hopkins Data Science Specialization Certificate
- Currently pursuing CSPA and ACAS credentials

EDUCATION

- PhD in Civil Engineering, John Hopkins University
- MS in Civil Engineering, Johns Hopkins University and University of Tehran
- BS, Civil Engineering, University of Tehran

SELECTED PROJECTS

Hazard Detection Computer Vision

Hazard detection is a product to facilitate underwriting process by allowing insurer's customers do self inspect their properties. It uses imagery data from personal properties as the input and determines if any risk/hazard from a set list exist in the given property. Deep learning convolution neural networks are trained on a huge set images collected from properties across the US.

Roof Age Model

Roof Age product include the roof age and an assigned confidence score for about 80 million residential home across the US. A predictive model is built to estimate the roof age for 75% of the homes for which the roof age is not know through different sources.

Connected Homes Model-Ready Data

Working with the IoT team to extract and transfer smart homes data. The data is provided by Vivant (a home security company) and is transferred to various insurance carriers. Tasks include receiving, cleansing, filtering, and matching large amount of data from both sides.

BOP Rating Factors

Updating the rating factors for the Business Owner Policies using transactional data provided by insurers. Also updating the grouping of the different classes of businesses. GLM is the main modeling technique used.

INTERESTS AND EXPERTISE

- Data Science | Analytics | Machine Learning
- Supervised and unsupervised predictive analytics
- Deep learning and Computer Vision
- Statistics
- Big data

Agenda

- Introduction & Background
- CART: Simple Trees
- Beyond CART: Bagging & Boosting
  - AKA Random Forest & Gradient Boosting
- Interpretation
- Evaluation of Tree based models

Traditional Insurance Ratemaking

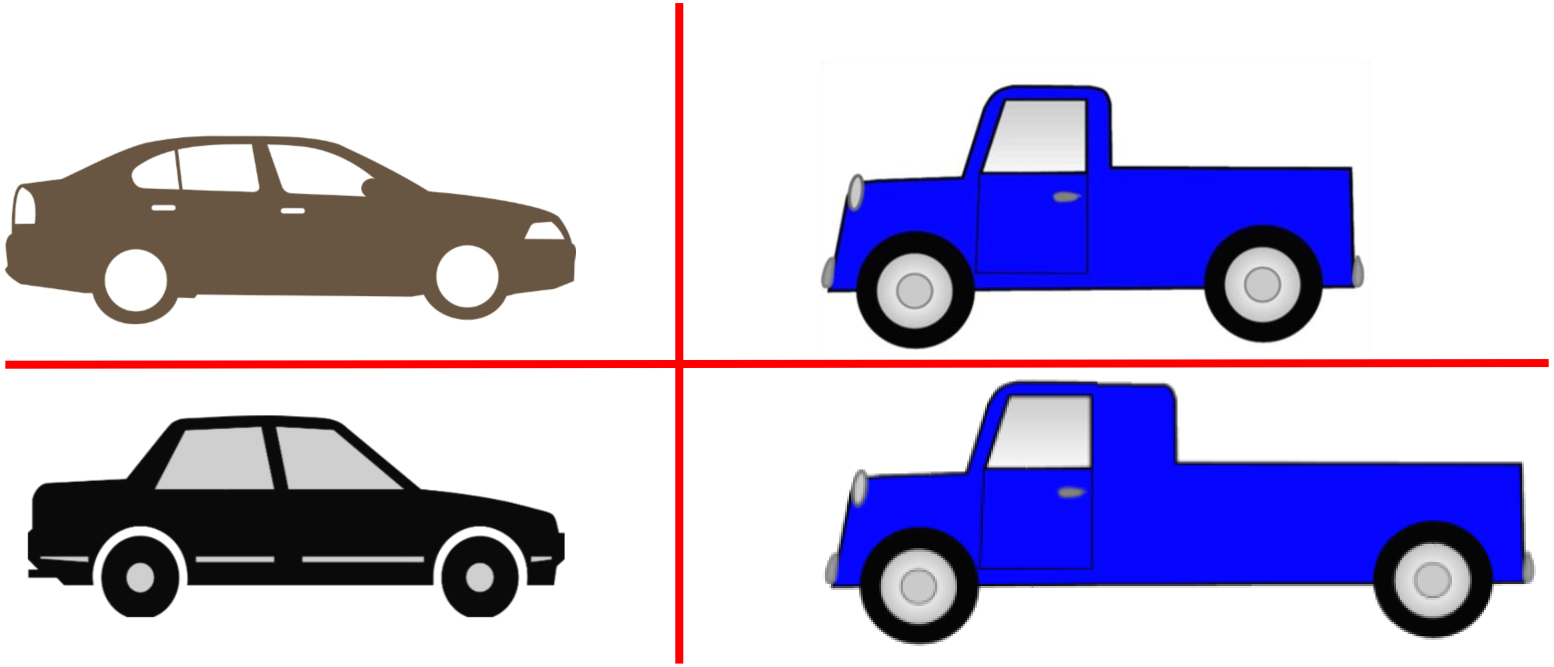
Generalized Linear Models

- Relate Multiple Predictors to Frequency, Severity, or Pure Premium
- Not Necessarily "Linear"

- Well established for Ratemaking & Underwriting
- Proven Track Record

	Linear Models
	GLM
Interpretability	🟩
Likelihood of Overfitting	⬜
Sensativity to Collinearity	🟥
Relative Predictive Power Given Strong Signal	🟩
Relative Predictive Power Given Weak Signal	⬜
Relative Predictive Power Given Complex Variable Behaviors (Interactions + Mixtures + Non-linear Effects)	🟥
Skill Required	🟥
Speed of Development	🟥

Segmentation, Body Style, Vehicle Weight, Wheelbase, Engine Size, MSRP, Fuel Type, etc.

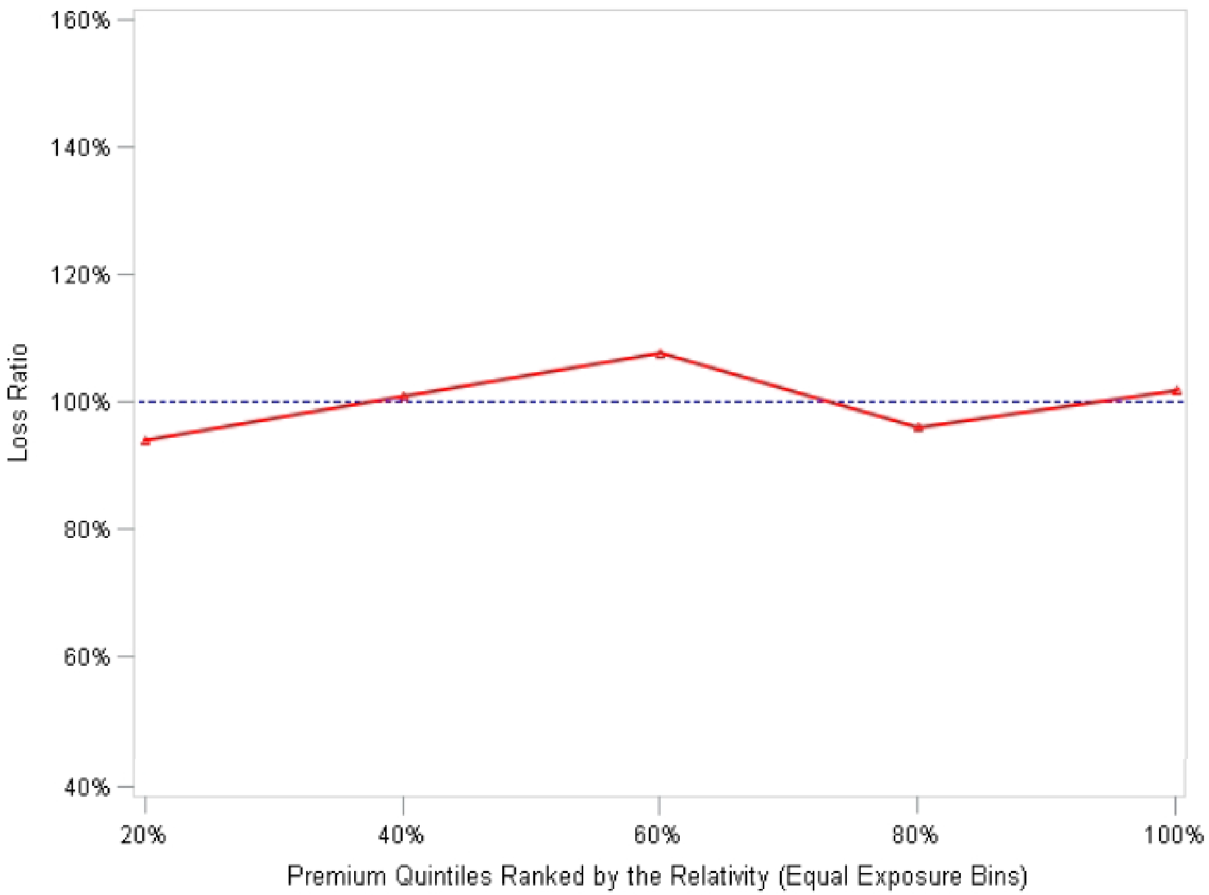


This Photo by Unknown Author is licensed under CC BY-SA

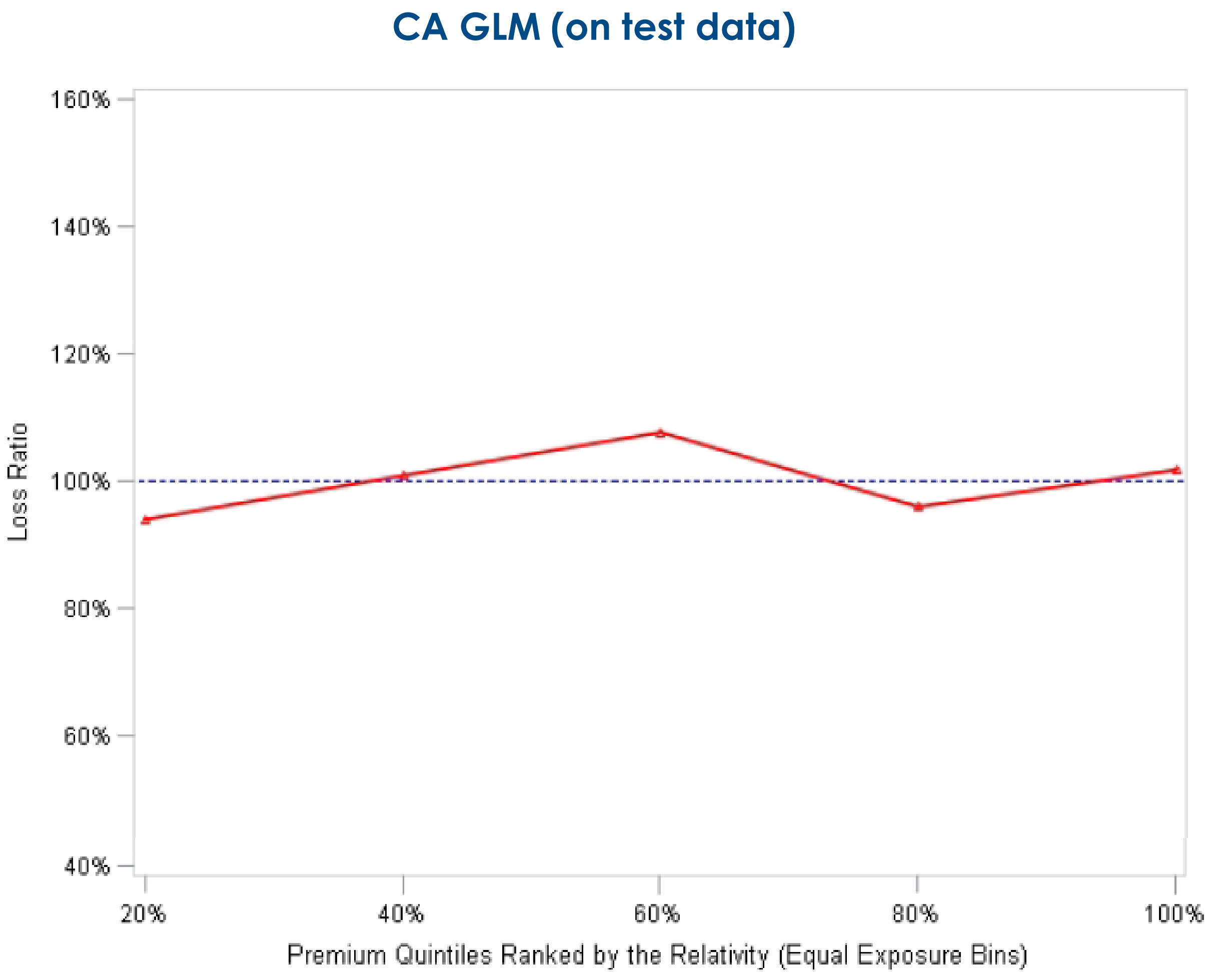
This Photo by Unknown Author is licensed under CC BY-NC

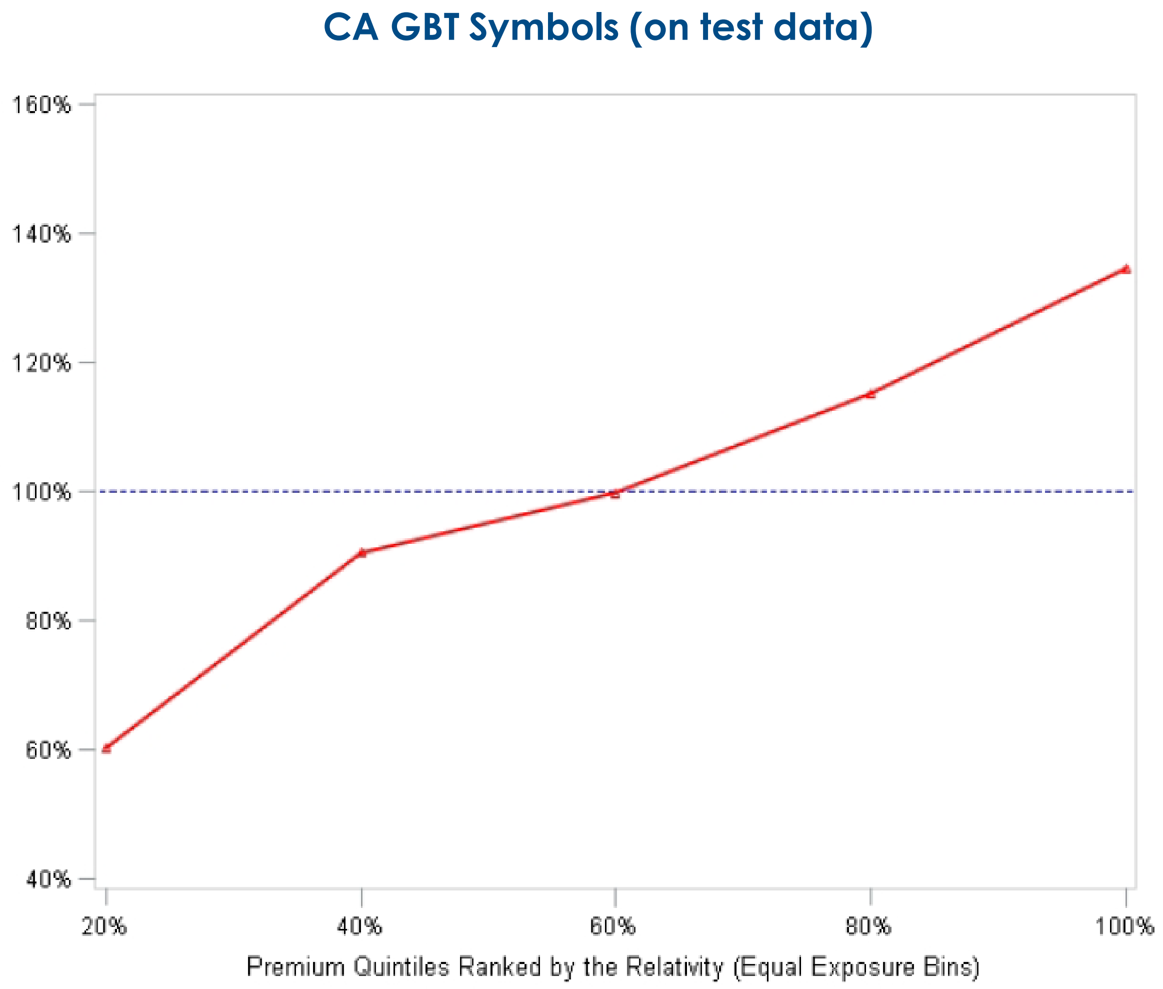
GLMs and GBTs applied to Comm Auto Liability

CA GLM (on test data)



GLMs and GBTs applied to Comm Auto Liability


CA GLM (on test data)


CA GBT Symbols (on test data)

Level Setting

- Not artificial intelligence- we build a mc uct
  - Doesn't adapt, change, evolve, etc. (
  - Supervis model
- Exactly the
- We can o
- This isn't c
  - First
  - First
    - Use
  - First
    - Use

DELTA BOOSTING: A BOOSTING APPLICATION IN ACTUARIAL SCIENCE

Simon CK Lee[*1] and Sheldon Lin[†2] and Katrien Antonio[‡1,3]

[1] KU Leuven, Bel
[2] University of Toront
[3] University of Amsterdam,
May 1, 2015

DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Claims Reserving using Boosting and Generaliz Models

MARCUS AHLGREN

Expert Systems with Applications
Volume 39, Issue 3, 15 February 2012, Pages 3659-3667

Gradient boosting trees for auto insurance loss cost modeling and prediction

Leo Guelman

Balancing robust statisti Gradient

Leo Guelman, Simon Lee, and Helen Gao

Royal Bank of Canada - RBC Insurance

March, 2012

Insurance Premium Prediction via Gradient Tree-Boosted Tweedie Compound Poisson Models

Yi Yang[*], Wei Qian[†] and Hui Zou[‡]

April 22, 2016

Breiman, L. (June 1997). "Arcing The Edge" (PDF). *Technical Report 486*. Statistics Department, University of California, Berkeley.

CART: Classification And Regression Trees

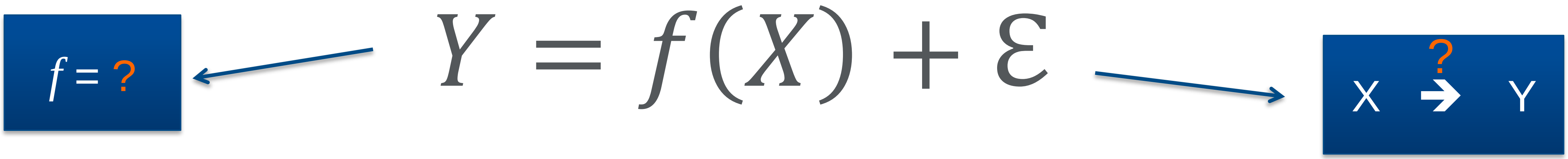
- Non-parametric method for Classification And Regression
- Segments the predictor space into simple regions (terminal nodes) following a set of splitting rules

- **Pop**: 2,500,000
- ower
- 150 vs. >=150
- 000

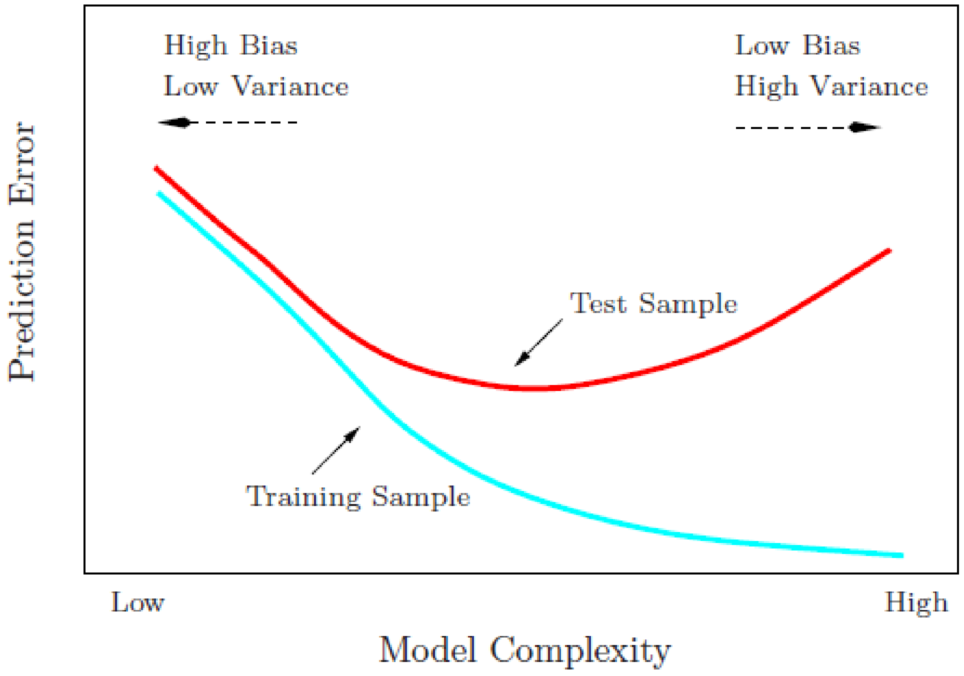
- **Pop**: 350,000
- **Par**: Segmentation Code
- **Threshold:** Compact Van vs. All Others
- **Left**: 50,000
- **Right**: 300,000

	Linear Models
	GLM
Interpretability	🟩
Likelihood of Overfitting	⬜
Sensativity to Collinearity	🟥
Relative Predictive Power Given Strong Signal	🟩
Relative Predictive Power Given Weak Signal	⬜
Relative Predictive Power Given Complex Variable Behaviors (Interactions + Mixtures + Non-linear Effects)	🟥
Skill Required	🟥
Speed of Development	🟥

Bias-Variance trade-off



$$Y = f(X) + \varepsilon$$



Source: Elements of Statistical Learning II

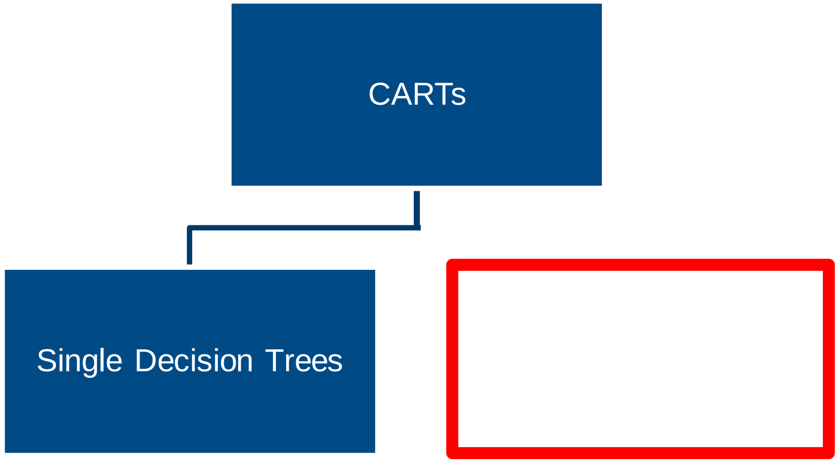
$$EPE = E\{Y - \hat{f}(x)\}^2$$

$$= E\{Y - f(x)\}^2$$ (Irreducible Error)

$$+ \left\{E\left(\hat{f}(x)\right) - f(x)\right\}^2$$ (Bias)

$$+ E\left\{\hat{f}(x) - E\left(\hat{f}(x)\right)\right\}^2$$ (Variance)

Beyond CART: Overcoming the limitations of the Single Decision Tree

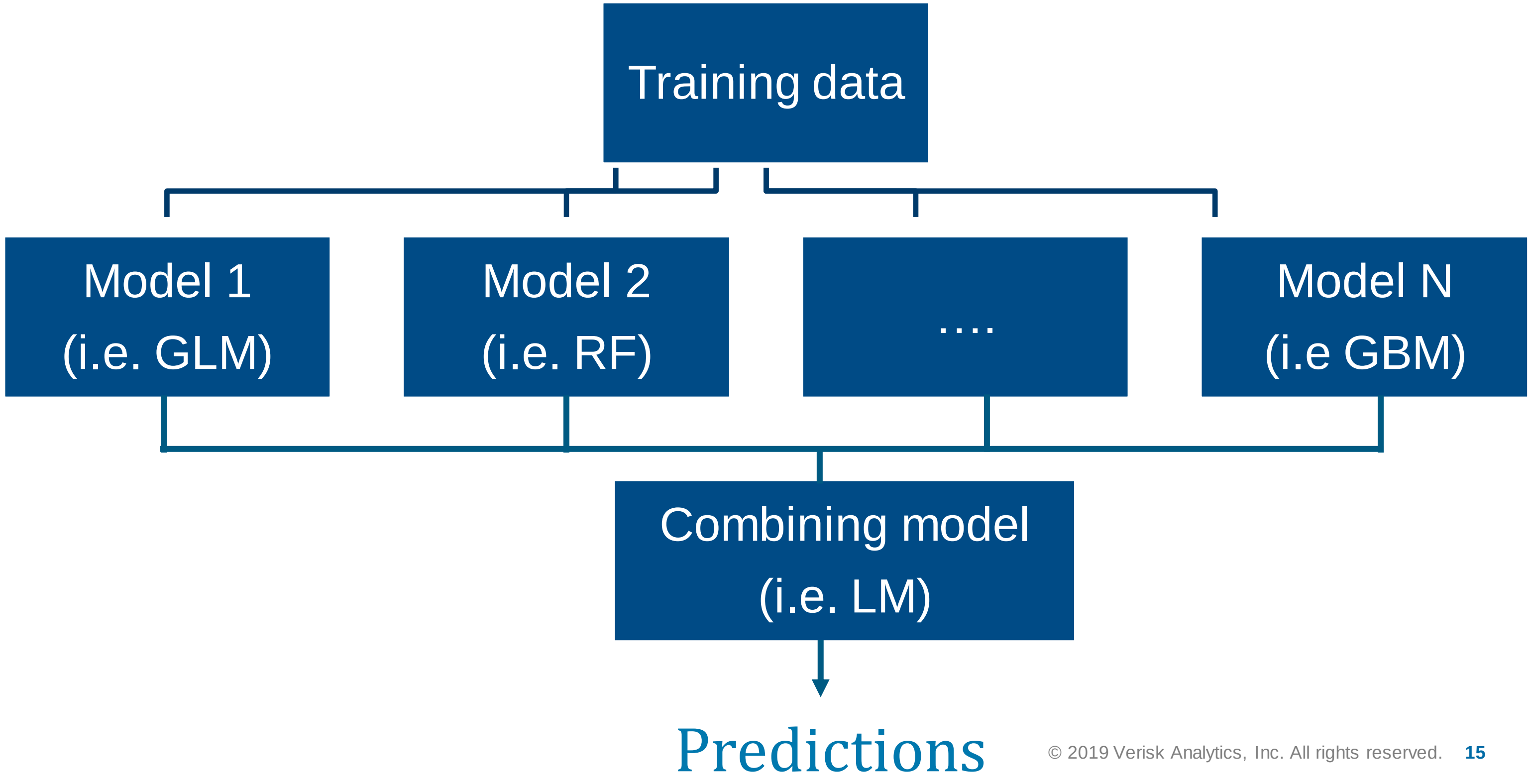


Single decision tree:

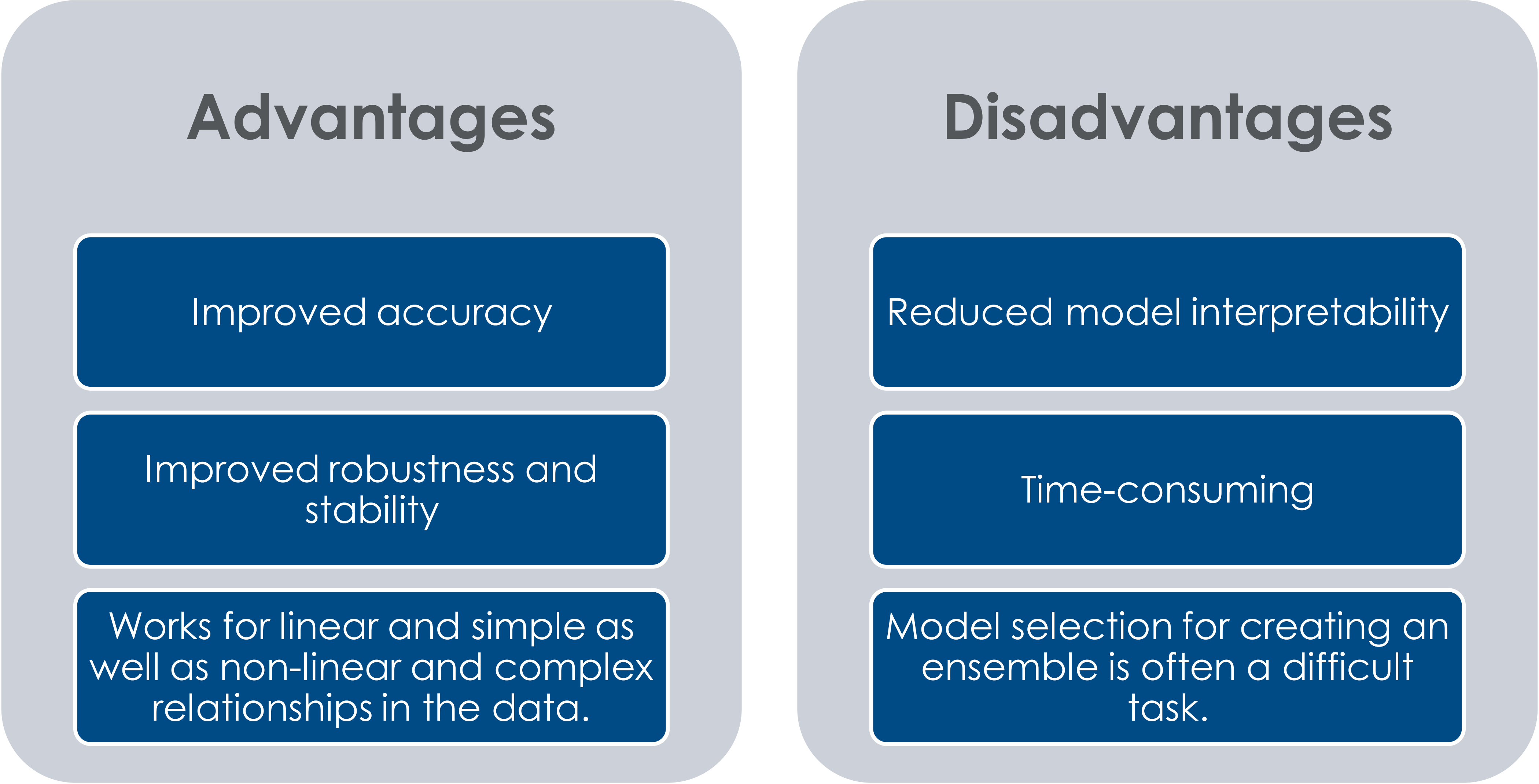
- Low bias, high variance
- Predictive power is not great

Ensemble Learning

- **Ensemble learning can be broken down into two tasks:**
  - Developing a population of base learners from the training data
  - Combining them to form the composite predictor.
- **Bagging and Boosting are two examples of ensemble learning**



Ensemble Learning



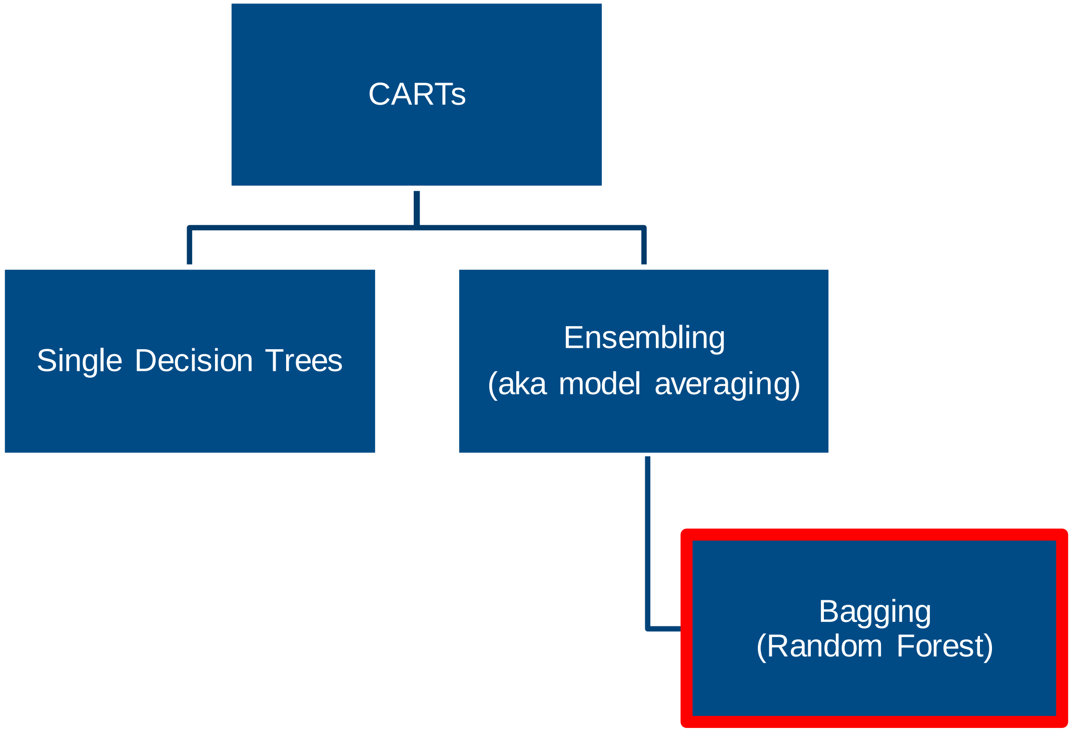
Advantages

- Improved accuracy
- Improved robustness and stability
- Works for linear and simple as well as non-linear and complex relationships in the data.

Disadvantages

- Reduced model interpretability
- Time-consuming
- Model selection for creating an ensemble is often a difficult task.

Beyond CART: Overcoming the limitations of the Single Decision Tree



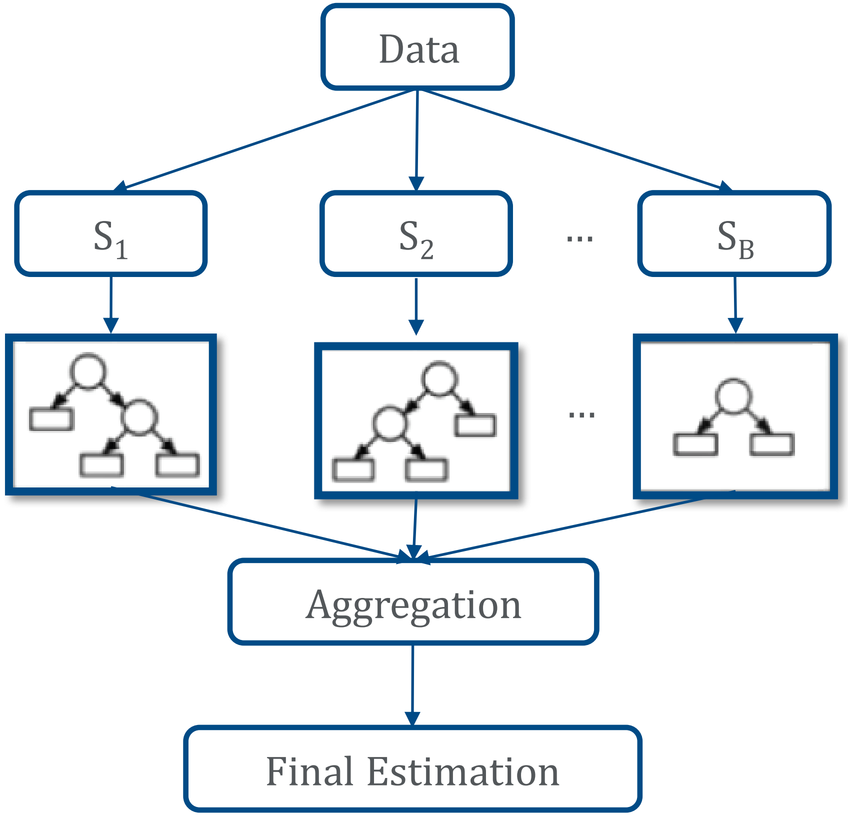
Bagging (Random Forest)

SERVE | ADD VALUE | INNOVATE

Bagging

- Bagging (bootstrap aggregating) averages prediction over a collection of bootstrap samples, thereby reducing its variance.
- B models are created using B bootstrap samples of the data, Then bagging estimate is denoted by:

$$\hat{f}^{B}_{bag} = \begin{cases} \frac{1}{B}\sum_{b=1}^{B} T_b(x) & regression \\ Majority\,Vote & classification \end{cases}$$



Bagging was introduced by: Leo Breiman, 1994.

Why Bagging is such a great idea?

A decision tree is a very low bias but high variance model

We need to reduce the variance, how?

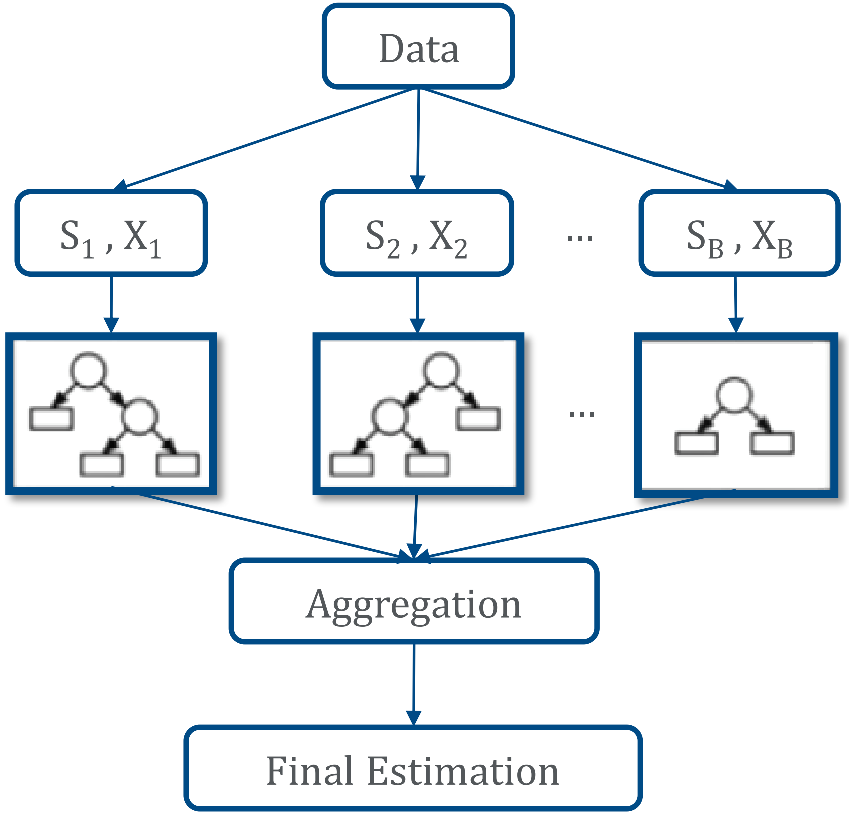
- Simple idea: aggregate many trees which is still a low bias model

If the trees are completely uncorrelated, variance is divided by the number of trees

If the trees are fully correlated, we gain nothing

Usually, we are somewhere in between

Bagging to random forest



Random Forest was introduced by: Leo Breiman, 2001.

S_1 to S_B: B bootstrap samples from data
X_1, X_B: B random subsamples of the predictors to decease the correlation between the trees

$$\hat{f}_{rf}^{B} = \begin{cases} \frac{1}{B}\sum_{b=1}^{B} T_b(x) & regression \\ Majority\,Vote & classification \end{cases}$$

$$Var_{rf} = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Tuning Parameters:

- Fraction of the predictors to randomly select for each tree
- Minimum terminal node size
- Number of trees – mainly for computational efficiency, rarely cause overfitting
- Depth of trees

Bagging/Random Forest Summary

- Bootstrap aggregation
  - Sampling from data
  - Sampling the variables (RF)
- Note for categorical variable:
  - Some packages can only handle numerical data (one hot-encoding)
  - Example: imagine your data has two types of houses, single family and other.
    - turn your categories into 0s and 1s (but what happens if you have 3 possible categories?).
    - Create a new column for each variable value and assign a 0 or 1
    - Create a new column for each variable value -1 “base” value (like in a GLM)
      - But now, stochastic sampling on the variables is not going to work at variable level

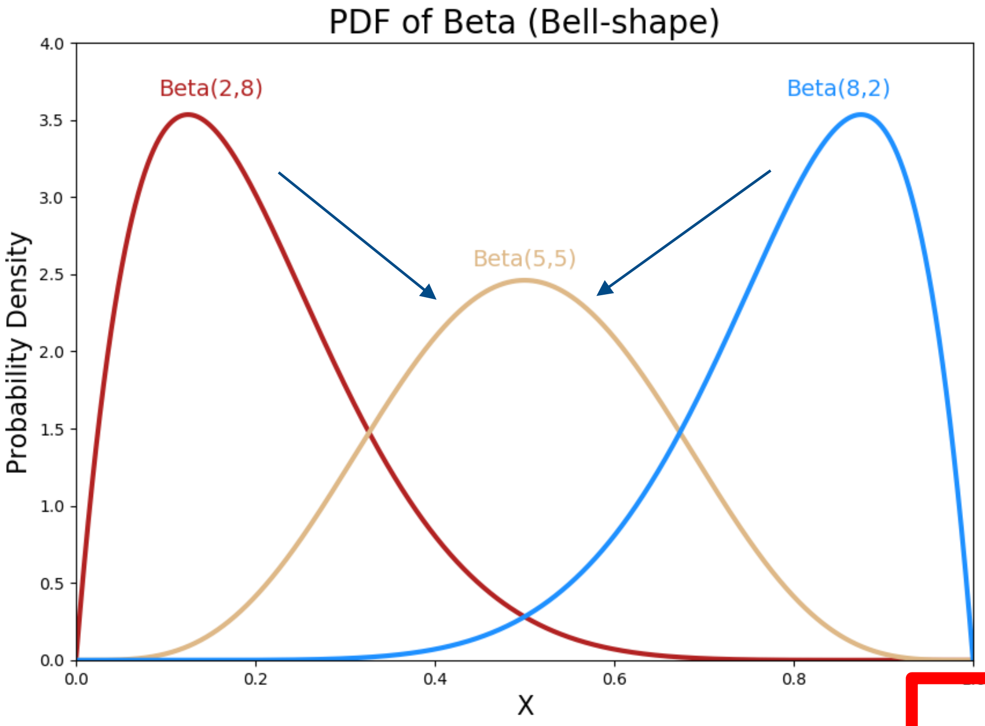
	Linear Models	CART	Bagging
	GLM	Single Decision Tree	Random Forest
Interpretability	🟩	🟩	🟧
Likelihood of Overfitting	⬜	🟥	🟨
Sensativity to Collinearity	🟥	🟨	⬜
Relative Predictive Power Given Strong Signal	🟩	⬜	🟨
Relative Predictive Power Given Weak Signal	⬜	⬜	🟩
Relative Predictive Power Given Complex Variable Behaviors (Interactions + Mixtures + Non-linear Effects)	🟥	⬜	🟩
Skill Required	🟥	🟩	🟧
Speed of Development	🟥	🟩	🟨

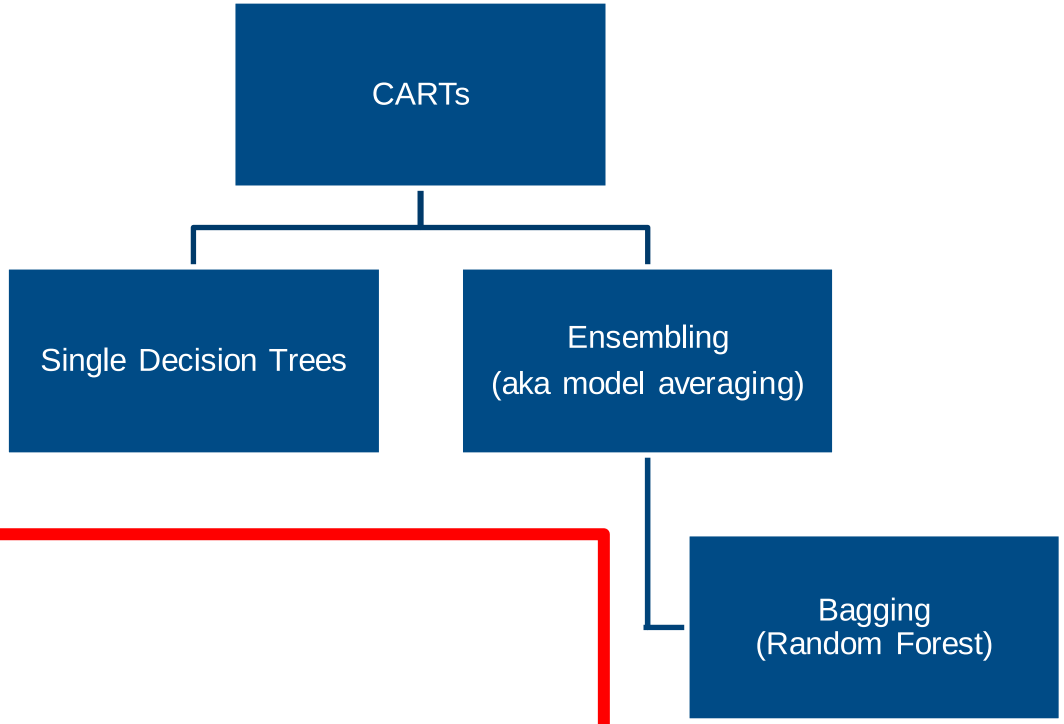
Boosting (GBT/GBM)

Verisk

SERVE | ADD VALUE | INNOVATE

Beyond CART: Overcoming the limitations of the Single Decision Tree





GBT = Gradient ***Boosted*** Trees

- Decision Trees: fine line between ‘underfitting’ and ‘overfitting’
  - Random Forest are often ‘underfitting’

- Solution: use the residuals of the first tree to reweight the data (greater weight given to higher residuals), this ‘reweighted’ data is used to create the next tree
  - Introduced by Jerome Friedman (1999)

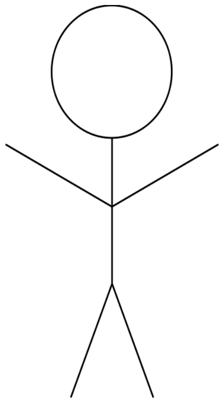
Residuals

Residuals

GBT = Gradient Boosted Trees

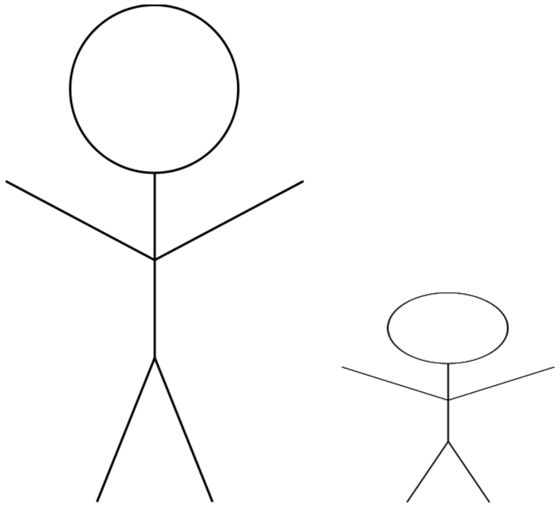
Analogy

Sam is OK at guessing weights



Sam: 182
Act: 180

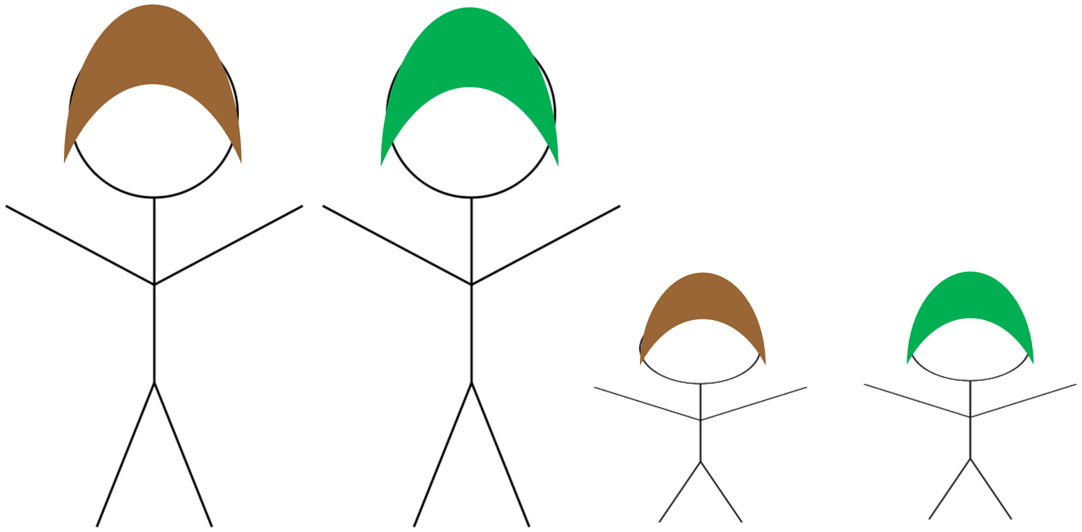
CKO notices Sam has a bias



Sam: 215
Act: 200

Sam: 130
Act: 145

Jane notices I have a partial bias



CKO: 195
Act: 200

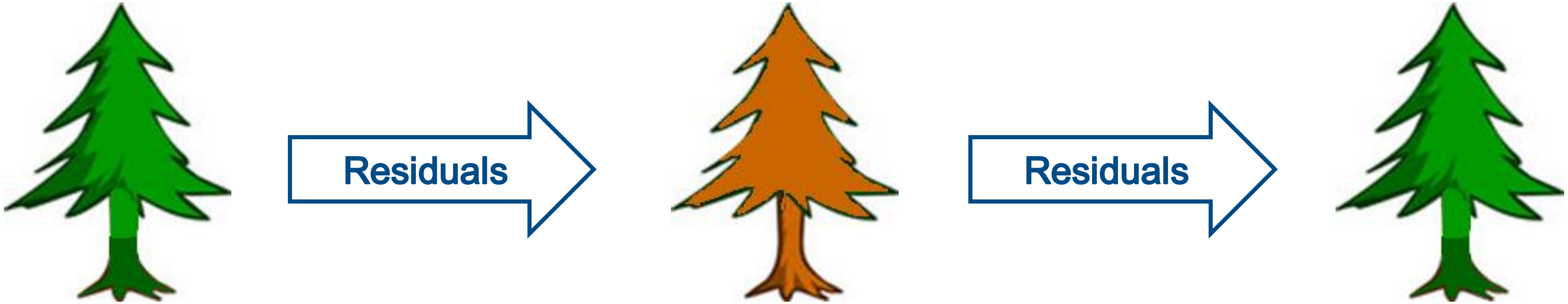
CKO: 205
Act: 200

CKO: 145
Act: 145

CKO: 145
Act: 145

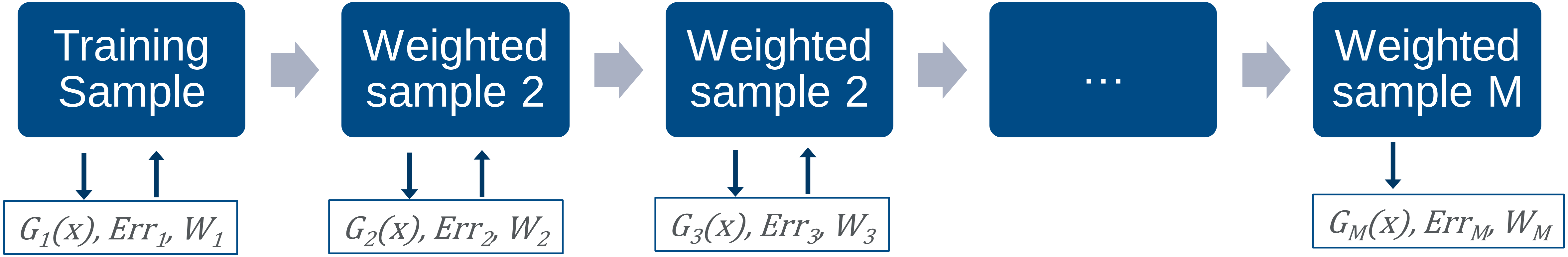
GBT = ***Gradient*** Boosted Trees

- A function that controls ‘reweighting’ based on residuals
- Prevents ‘overcompensation’



Boosting

The original boosting algorithm: AdaBoost (Freund and Schapire 1997)



Training Sample

Weighted sample 2

Weighted sample 2

…

Weighted sample M

$G_1(x), Err_1, W_1$

$G_2(x), Err_2, W_2$

$G_3(x), Err_3, W_3$

$G_M(x), Err_M, W_M$

$$G(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$$

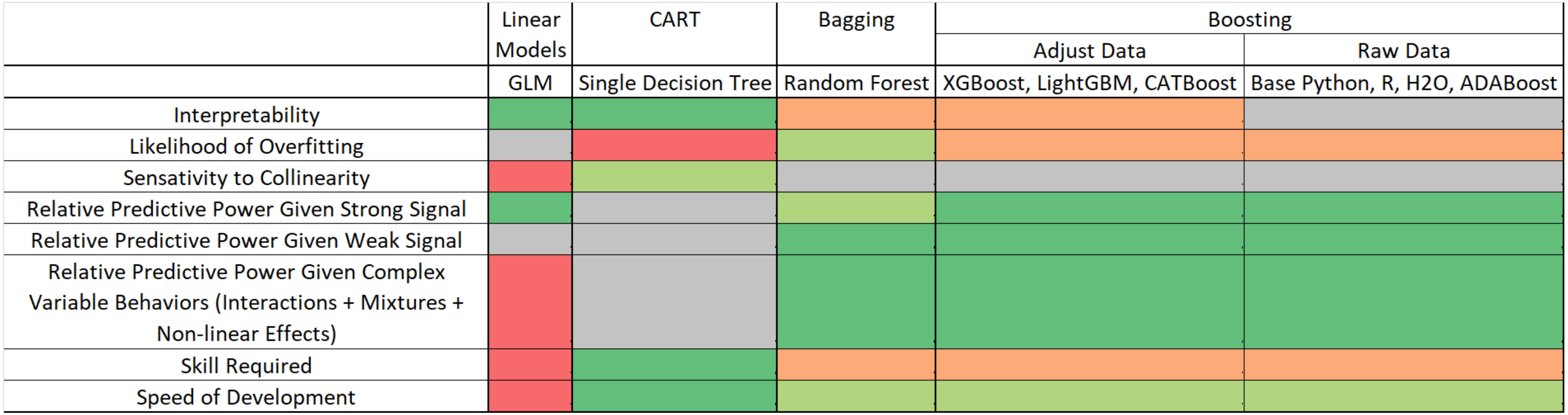
$$\overline{Err} = \frac{1}{N} \sum_{n=1}^{N} I(y_i \neq G(x_i))$$

Boosting in Practice: Hyperparameters

- Minimum Terminal Node Size
- Maximum Depth
- The number of splits per node
  - Controls the complexity of the boosted ensemble
  - Often very small numbers are used (d =2, 3, etc.)
- The number of trees
  - Boosting can overfit if too many trees are used, although this overfitting tends to occur slowly if at all.
- Stochastic sampling rates (column & rows) ~0.6 +- .2
- The Learning Rate
  - How quickly trees zero in on Strong signal
- The Learning Rate Shrinkage parameter
  - It is used to avoid early overfitting
  - Typical values are 0.01 or 0.001, and the right choice can depend on the problem.
  - A very small shrinkage parameter results in needing a very large number of trees to achieve good performance
- Hyperparameter Search: Cross Fold Validation

Gradient Boosted Trees

- Pros, Cons, & Limitations



	Linear Models	CART	Bagging	Boosting	
				Adjust Data	Raw Data
	GLM	Single Decision Tree	Random Forest	XGBoost, LightGBM, CATBoost	Base Python, R, H2O, ADABoost
Interpretability	🟩	🟩	🟧	🟧	⬜
Likelihood of Overfitting	⬜	🟥	🟨	🟧	🟧
Sensativity to Collinearity	🟥	🟨	⬜	⬜	⬜
Relative Predictive Power Given Strong Signal	🟩	⬜	🟨	🟩	🟩
Relative Predictive Power Given Weak Signal	⬜	⬜	🟩	🟩	🟩
Relative Predictive Power Given Complex Variable Behaviors (Interactions + Mixtures + Non-linear Effects)	🟥	⬜	🟩	🟩	🟩
Skill Required	🟥	🟩	🟧	🟧	🟧
Speed of Development	🟥	🟩	🟨	🟨	🟨

Evaluation – Model Performance

SERVE | ADD VALUE | INNOVATE

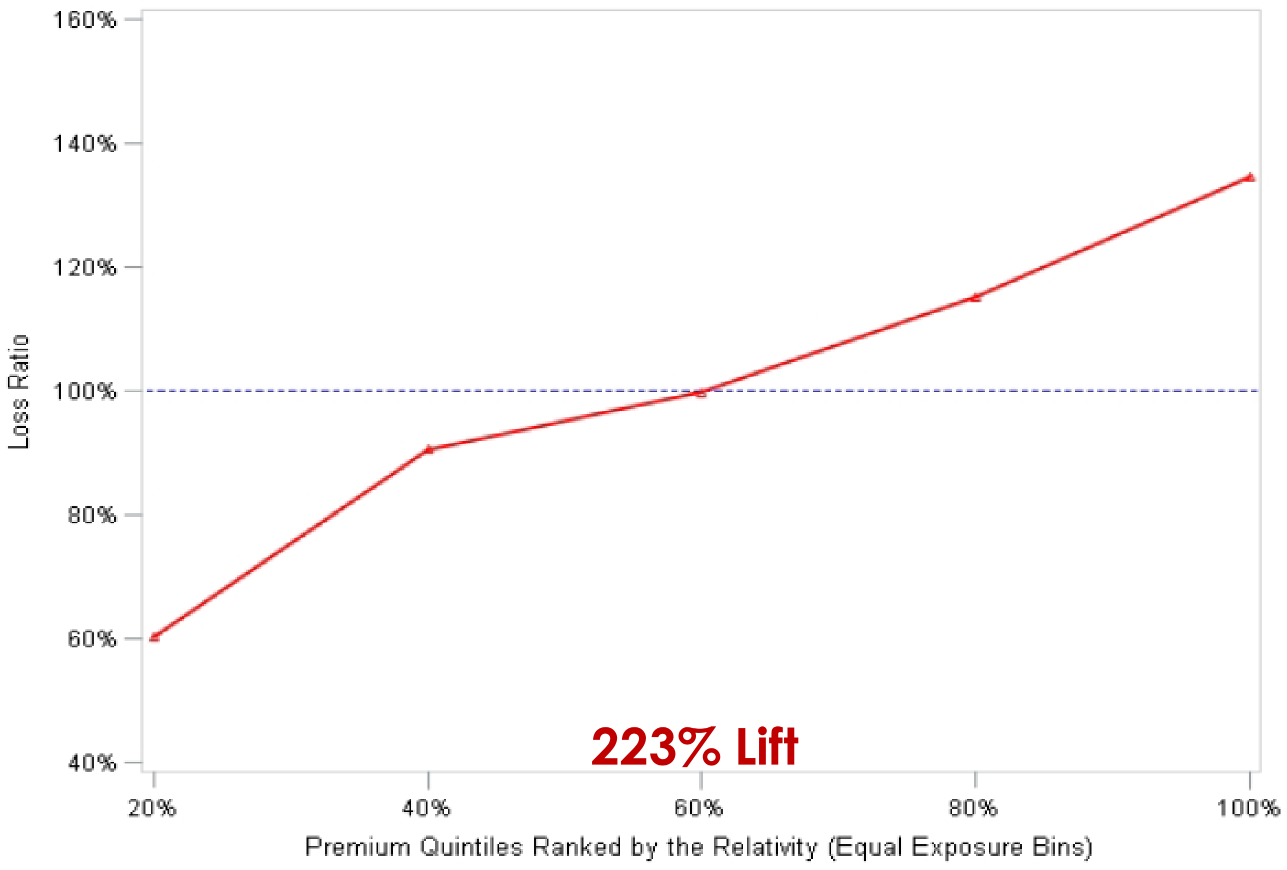
Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM

Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift

RACA GBT Symbols (on test data) – Random Holdout

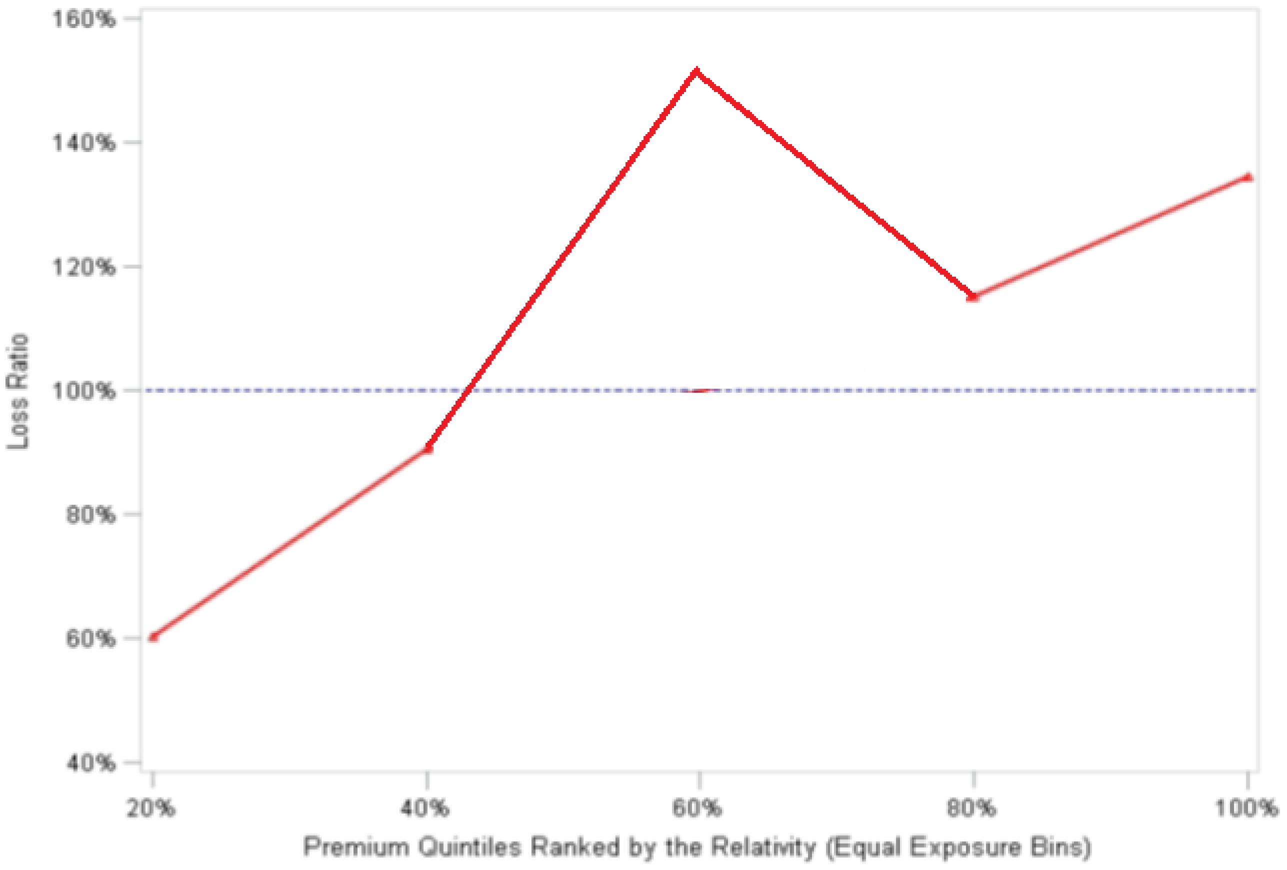


223% Lift

Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift (No Reversals)

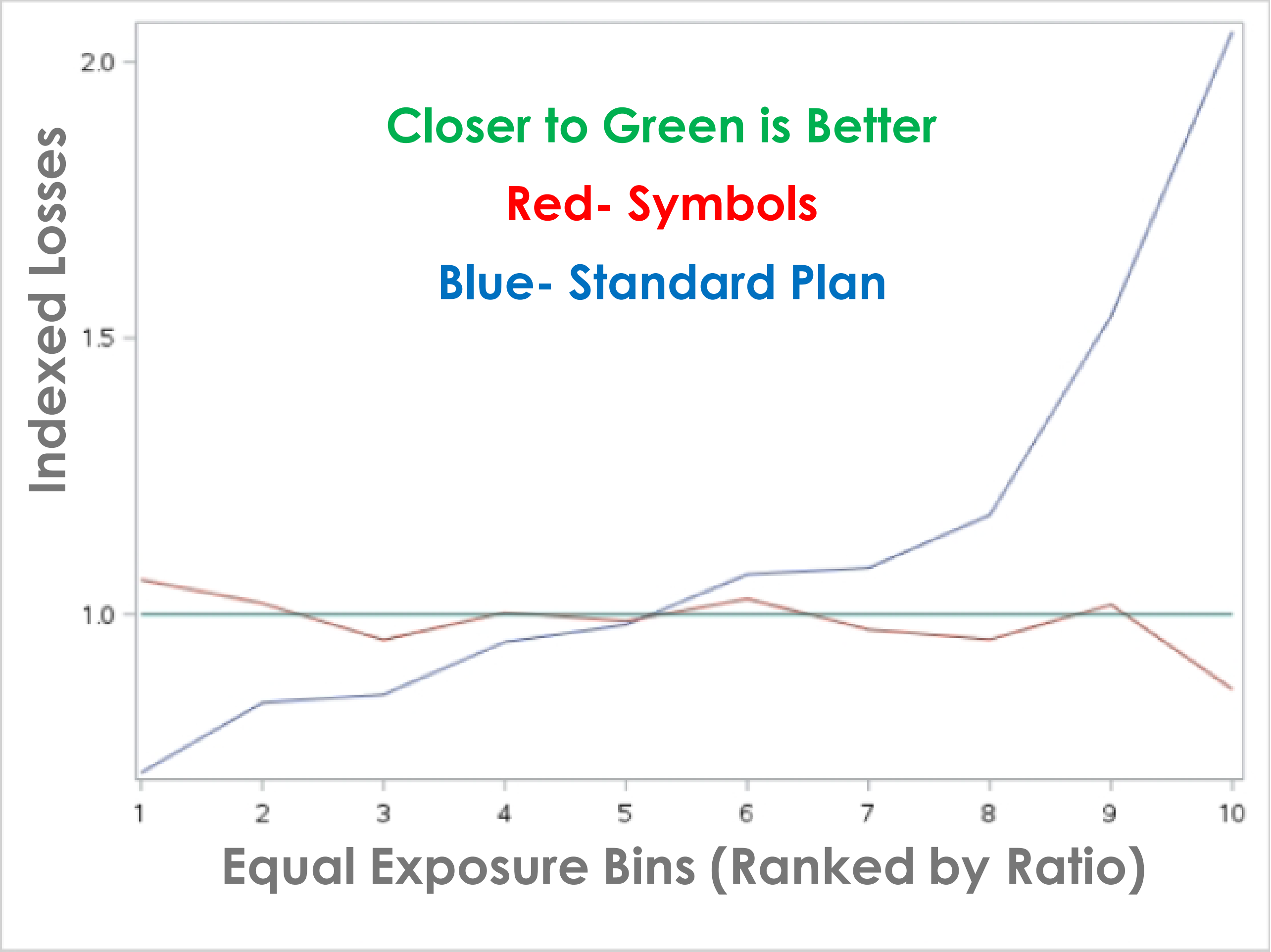
Simulated Bad Lift Chart



Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift (No Reversals), Head-to-Head

RACA GBT Symbols vs Standard Plan (on test data)

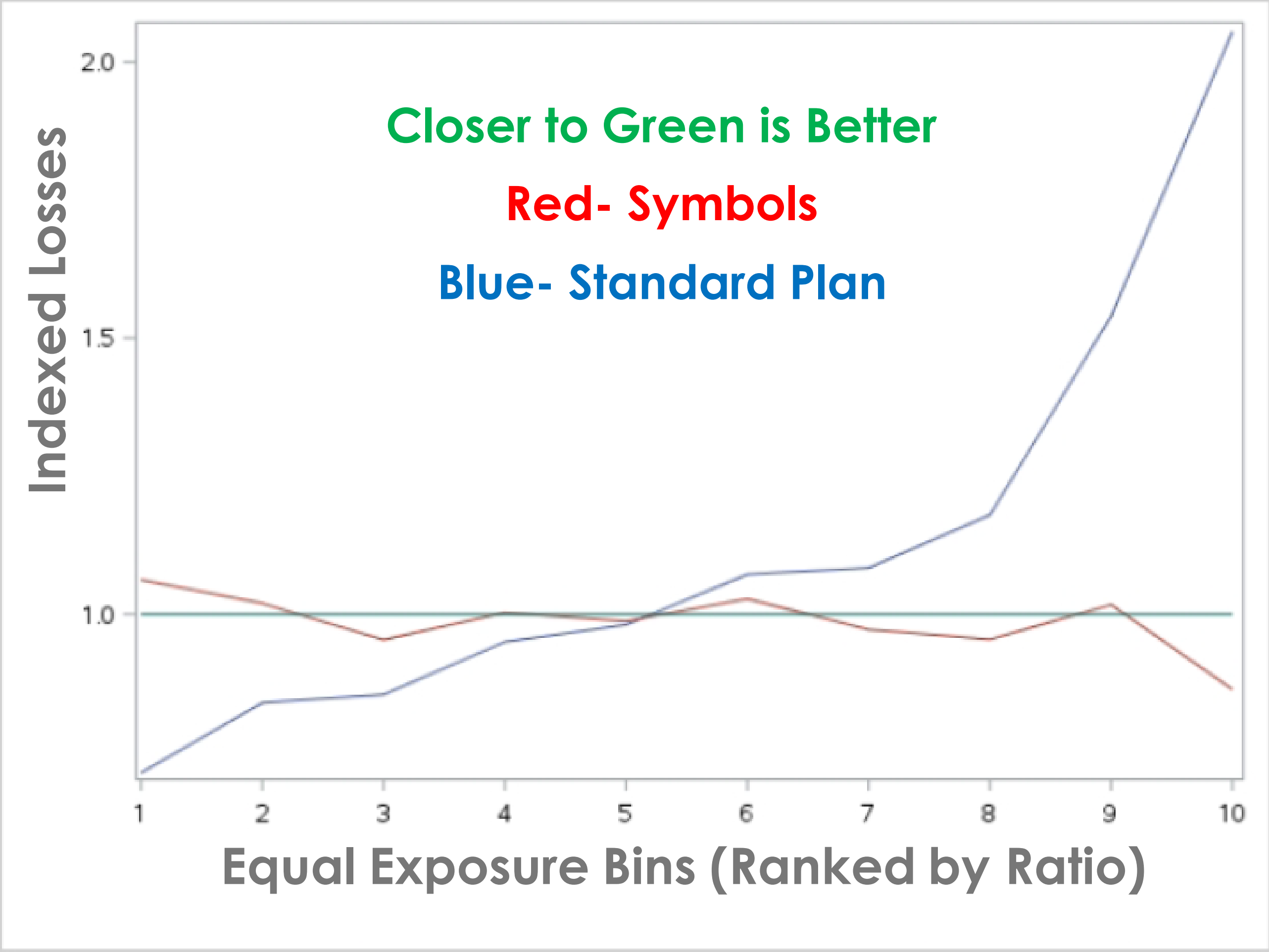


Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift (No Reversals), Head-to-Head

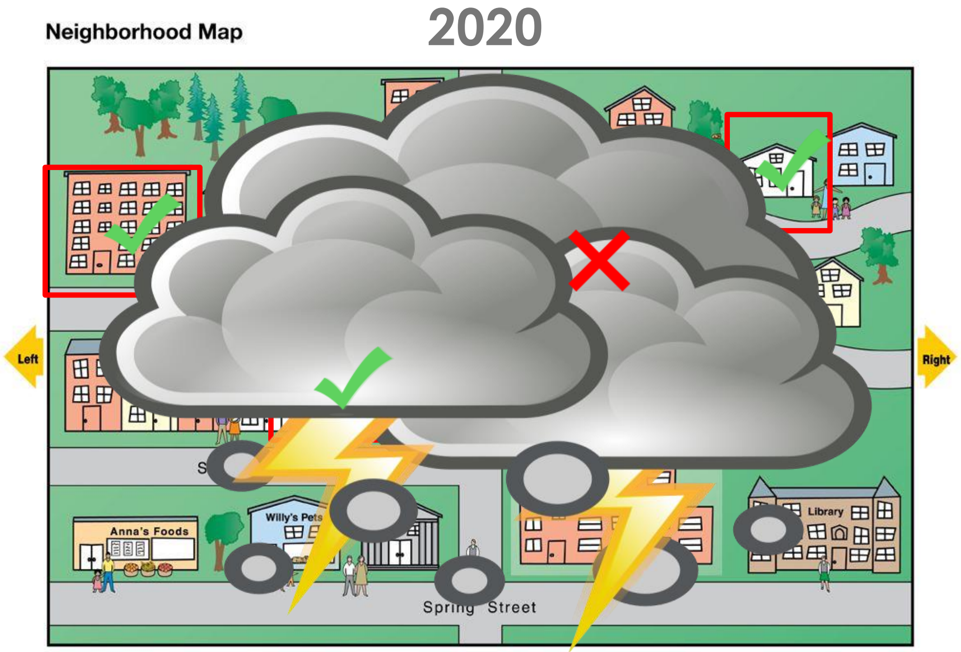
	Current	GBT Symbols	Improvement
Lift	*NA*	223%	223%
Gini	30.343	33.404	10%
MSE	2.594	0.398	652%
MSE Weighted	0.201	0.036	552%

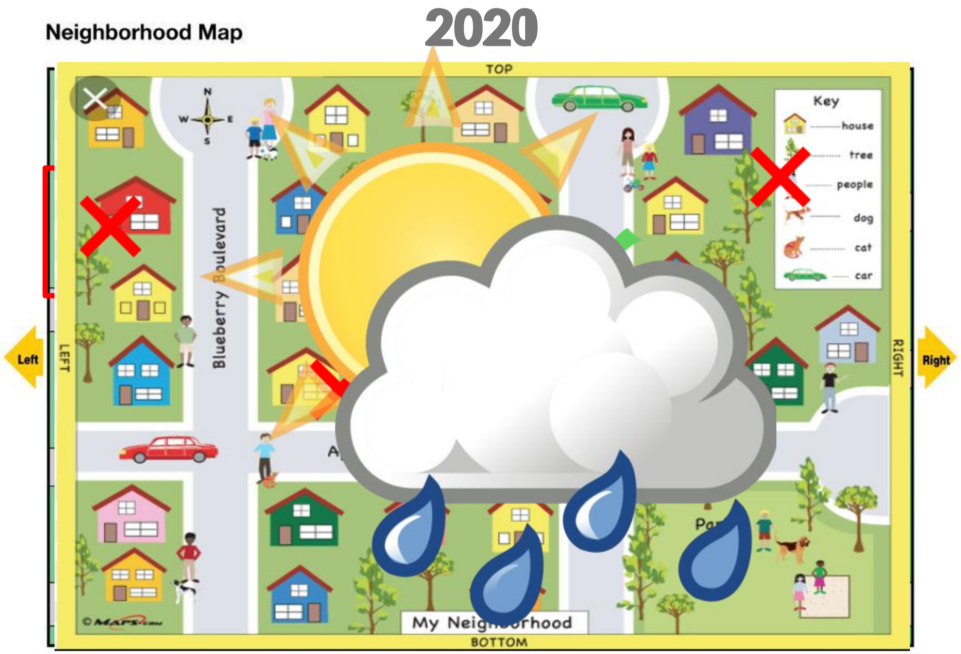
GBT Symbols vs Standard Plan (on test data)



Machine Learning Tree Performance

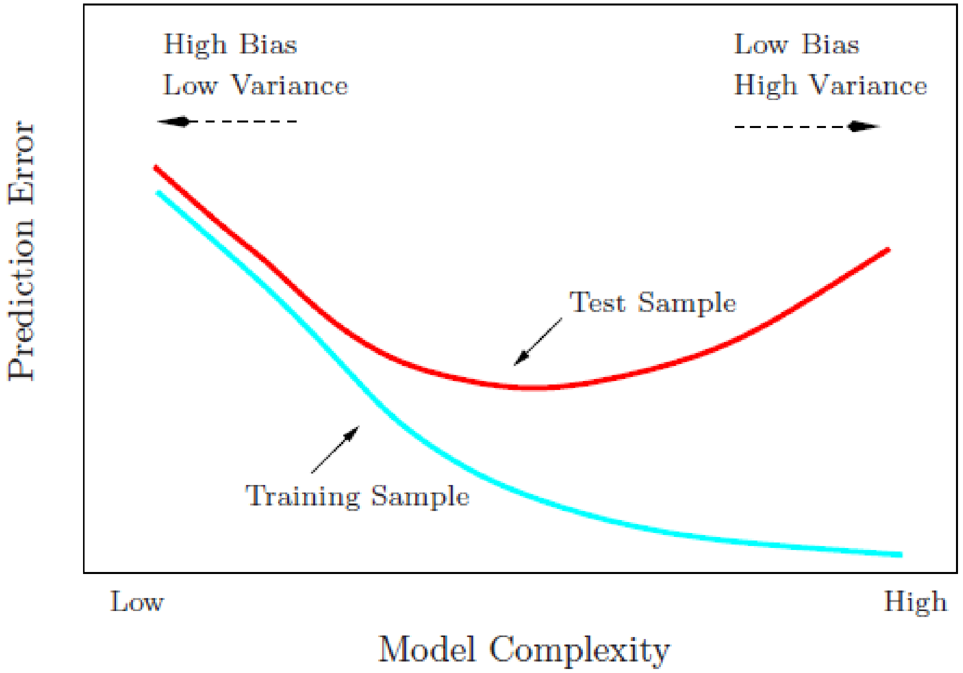
- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift (No Reversals), Head-to-Head
  - Trees can overfit, need “True Test” data
    - Out of Time (train on 2014-2018, test on 2019-2020)
    - Out of Geography





Machine Learning Tree Performance

- **Predictive Accuracy**
  - Same metrics for GLM, RF, and GBT/GBM
    - AUC, Gini, Lift (No Reversals), Head-to-Head
  - Trees can overfit, need “True Test” data
    - Out of Time (train on 2014-2018, test on 2019-2020)
    - Out of Geography
    - Multiple Random Samplings (different seeds)
  - Test set should be no more than 30% of total data
    - Hyperparameters are optimized based on data set size
  - Train vs Test Error



When to use Trees

SERVE | ADD VALUE | INNOVATE

When to use Trees

- Classification & Regression on Structured Data
  - Claims
  - Underwriting
  - Rating
- Real, but Hard to find Signal
  - Variable Interactions, Mixtures
  - Especially well suited for Low Frequency, High Severity Lines (Liability, Fire, etc.)
- Data
  - **Big** [enough] Data
    - Trees can overfit, be biased towards training data
    - Need enough data to reach credibility
  - Not **overly** concerned about Protected Classes
    - Data is orthogonal to Protected Classes
    - Underlying data has been transformed to remove bias
    - Risk of reengineering removed features

Evaluation - Regulation

Verisk

SERVE | ADD VALUE | INNOVATE

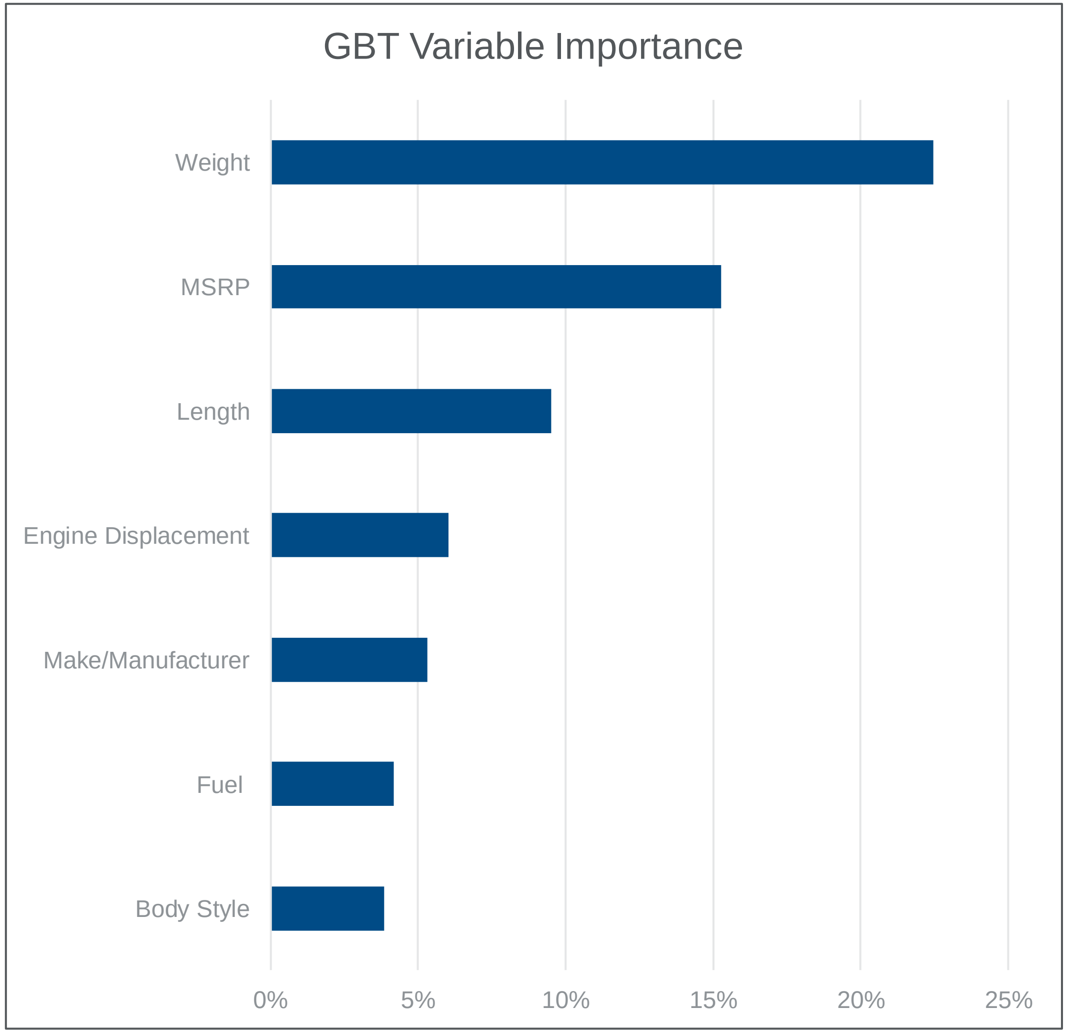
Machine Learning Tree Quality

- **Predictive Accuracy**
- **Protecting the Consumer**
  - Control data that goes into model
  - Interpretability
    - Do you understand “the story”?
      - Could a failure to understand “the story” cause an undesirable outcome?

Machine Learning Tree Quality

- **Predictive Accuracy**
- **Protecting the Consumer**
  - Control data that goes into model
  - Interpretability
    - Do you understand “the story”?
      - Could a failure to understand “the story” cause an undesirable outcome?
    - Variable Importance: Weighted measure of how many records are affected by each Variable throughout entire Model

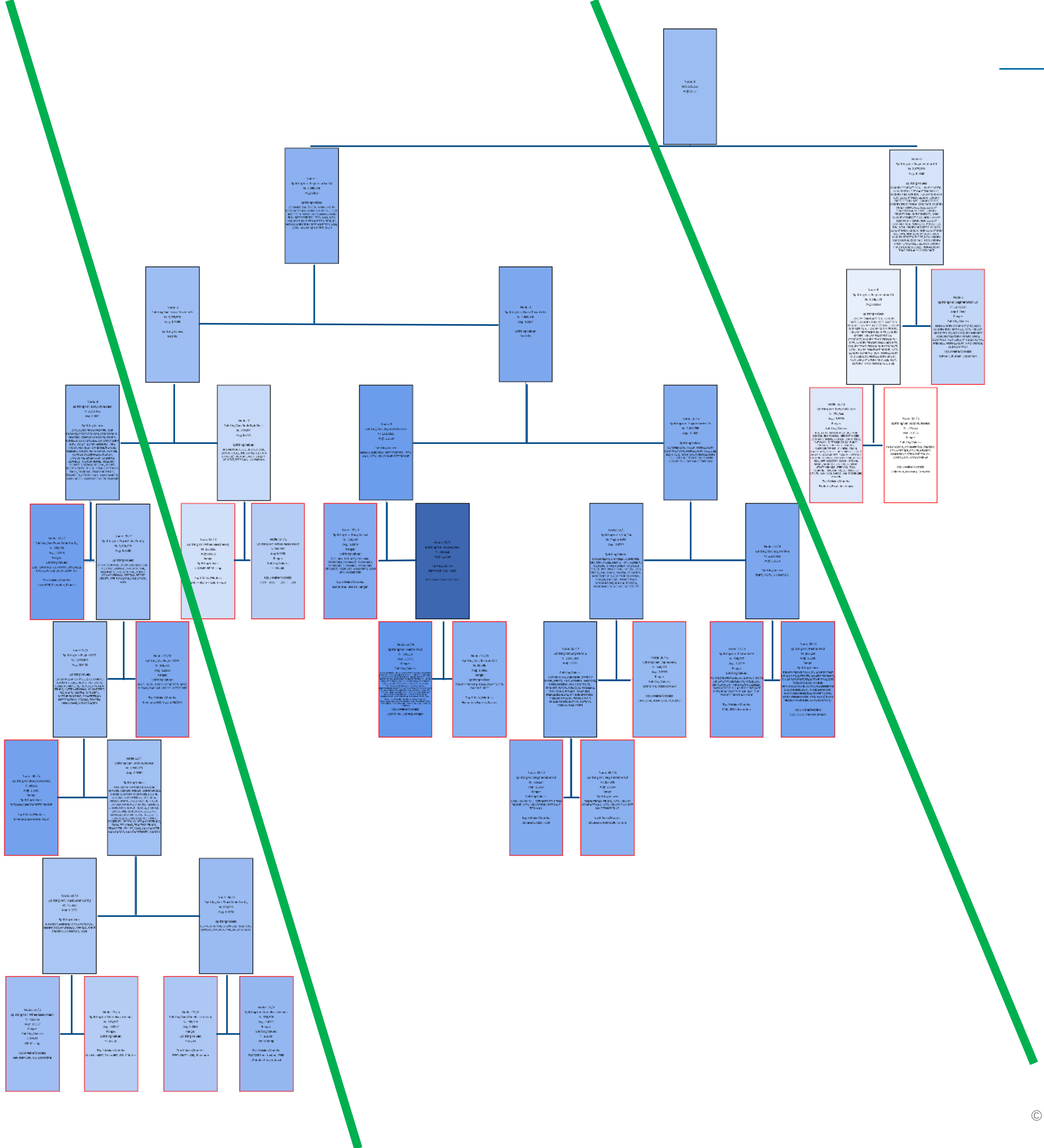
Variable Importance



Machine Learning Tree Quality

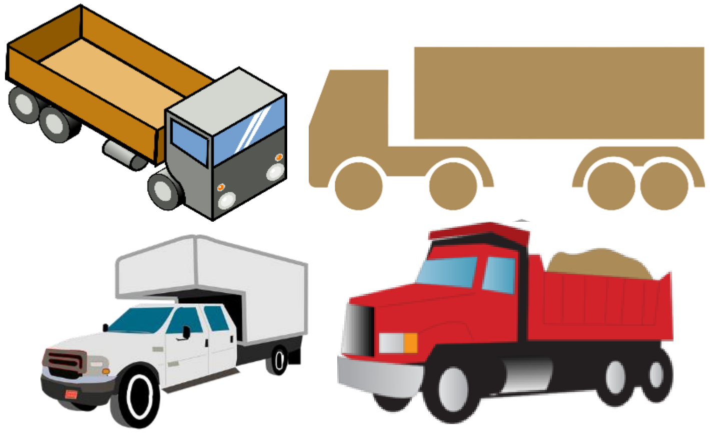
- **Predictive Accuracy**
- **Protecting the Consumer**
  - Control data that goes into model
  - Interpretability
    - Do you understand "the story"?
      - Could a failure to understand "the story" cause an undesirable outcome?
    - Variable Importance: Weighted measure of how many records are affected by each Variable throughout entire GBT
    - Interpretative trees (Surrogate model)
      - Fit a simple Decision Tree to the GBT model predictions – AKA Data Mine the GBT Predictions
      - Easy to interpret – tells a story

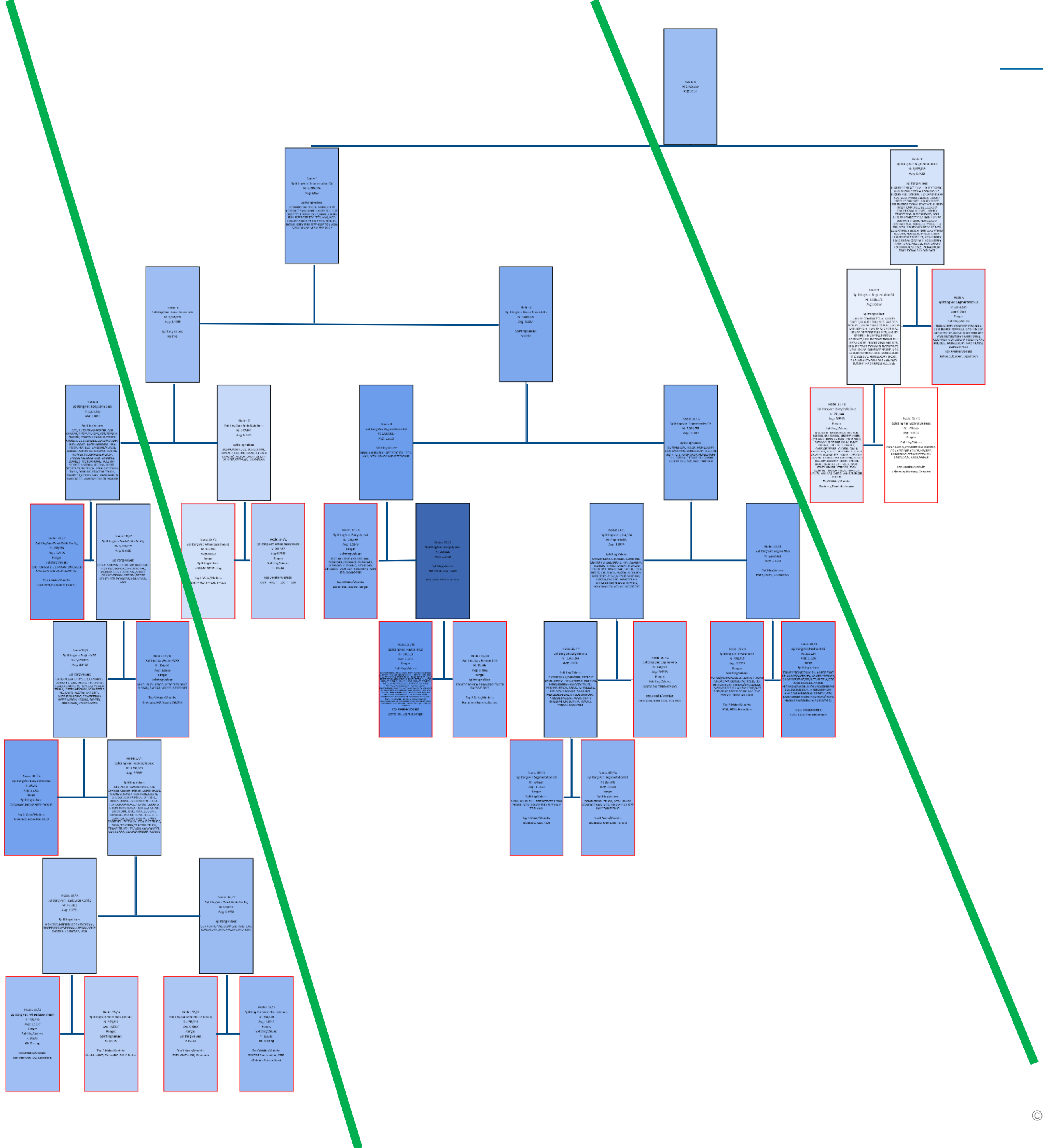
Interpretive Trees



Interpretive Trees

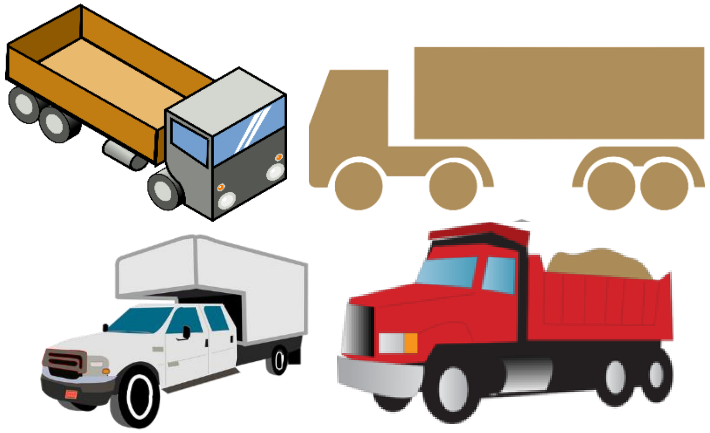
Material Movers

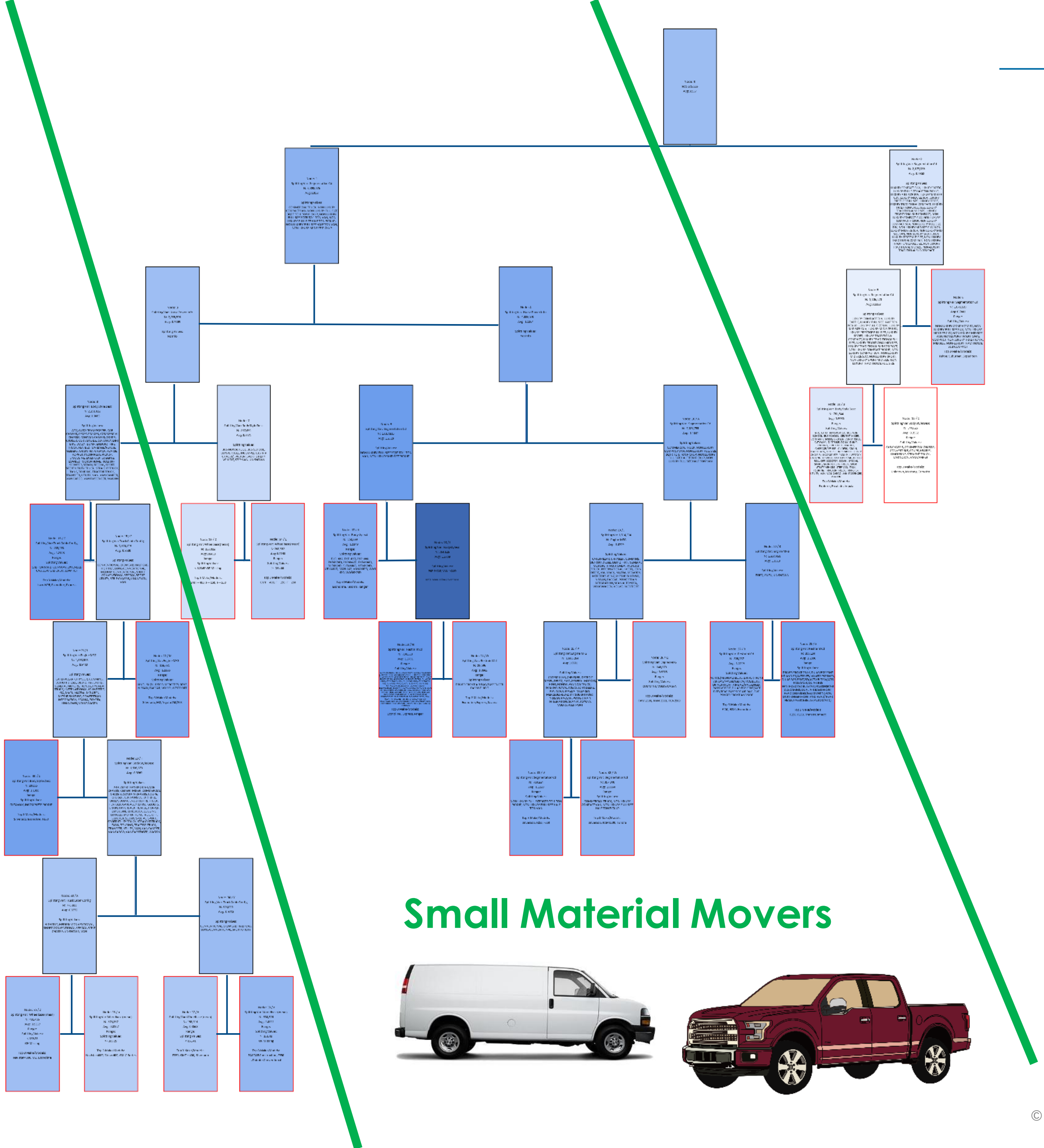




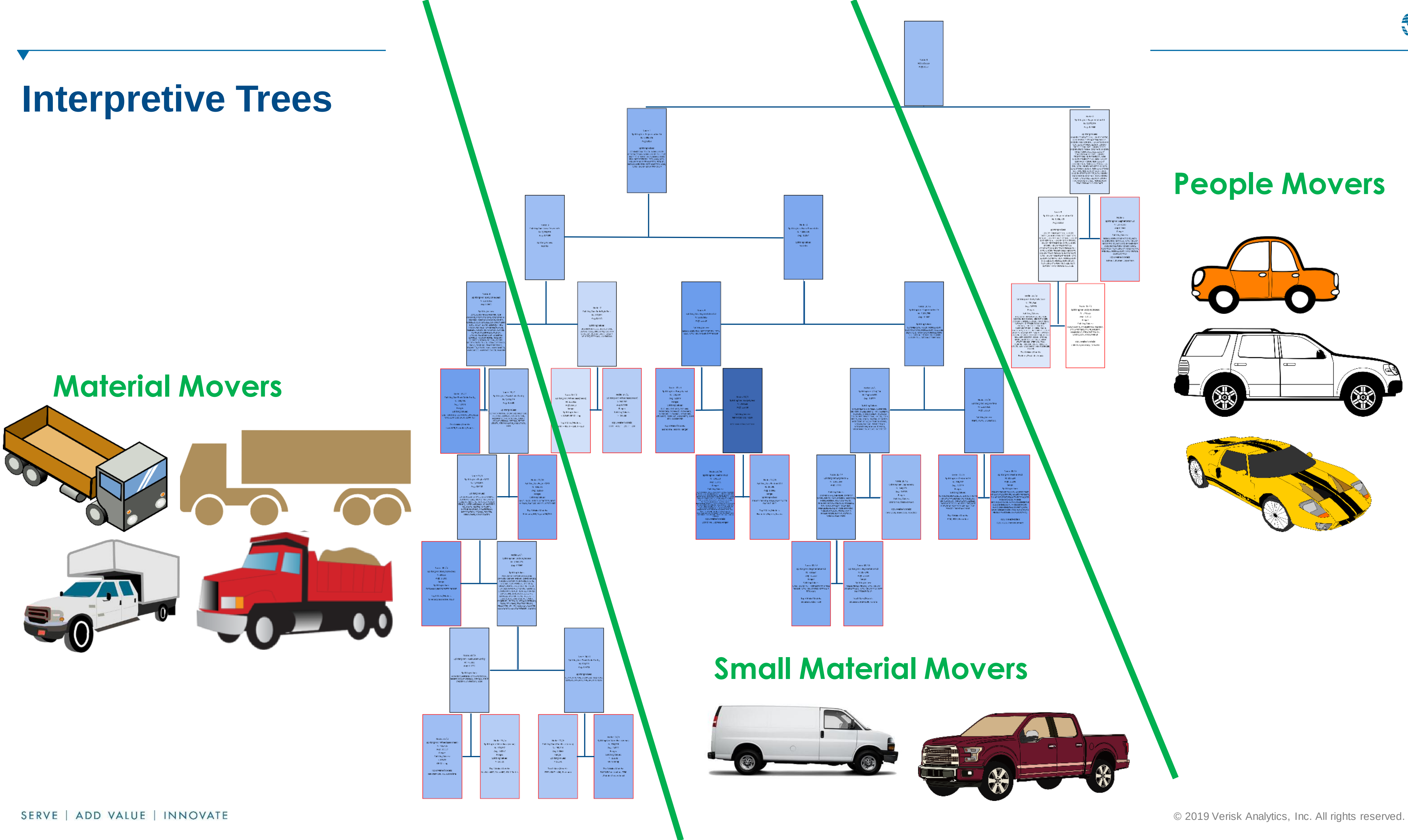
Interpretive Trees

Material Movers





Interpretive Trees



Machine Learning Tree Quality

- **Predictive Accuracy**
- **Protecting the Consumer**
  - Control data that goes into model
  - Interpretability
    - Do you understand “the story”?
      - Could a failure to understand “the story” cause an undesirable outcome?
    - Variable Importance: Weighted measure of how many records are affected by each Variable throughout entire GBT
    - Interpretative trees (Surrogate model)
      - Fit a simple Decision Tree to the GBT model predictions – AKA Data Mine the GBT Predictions
      - Easy to interpret – tells a story
    - Individual Conditional Expectation (ICE)
      - Show Correlation between Variable’s Value and GBT Prediction
      - Hard to interpret
    - Partial Dependence Plots
      - Requires rerunning of the model while iteratively setting each variable to a constant level
        - Long process, only takes a ~’univariate’ approach and leaves out 2+ way interactions
      - Hard to interpret
    - Shap Index
      - Really great for understanding ‘univariate’ approach
      - Not useful for 2+ way interactions & Mixtures

Machine Learning Tree Quality

- Things to look for
  - Can the Actuaries/Data Scientists tell a story that makes sense?
  - Hyperparameter Search
    - Good: Exhaustive/Factorial (Requires multiple rounds), Bayesian, Random
    - Bad: No optimization (even on Random Forest)
  - Truly “Random” Test Data
    - Weather Damage should always be tested using Out of Time/Out of Geography methods
  - “The Smell Test”
    - E.g., predictors=“engine size & driver age” or “# of windows per capita”
      - Partial (or preferably Repair*) questionable variables
        - Without ‘Repairing’ data, our understanding is that Trees can find signal related to undesirable predictor variables
      - Intuitive explanations for all important variables

*** Feldman et al 2015: certifying & removing disparate impact**

Machine Learning Tree Quality

- Common Insurance Data Science Mistakes
  - Non-sensical variables
  - Overfitting
  - Non-random test data (out of time/geography is key)
  - Minimum terminal node size is too small
    - For low frequency/high severity lines, recommend minimum size >5,000 records (could be 50,000)
      - Analogy to GLM Credibility Standards
  - Trees are too deep
    - Depth of 50= 2^50=1 billion terminal nodes ➔ overfitting
    - Recommended maximum depth : RF <=30 , GBT <=13
  - Not removing collinear predictors + one-hot encoding trap
    - Can negate stochastic column sampling in GBT & RF
  - Non-coherent predictions
    - Rates shouldn't double when you turn 25 only to go back down when your 26
- Some divergence from standard operating procedures is not necessarily "wrong", maybe just suboptimal

Summary

Trees are

- Not “AI”
- Well Established
- Faster & Easier than GLMs
- Very Accurate
- More interpretable than people realize

Thank you

Verisk

SERVE | ADD VALUE | INNOVATE