PREDICTIVE MODELING

A SEMINAR FOR REGULATORS

NAIC Insurance Summit – June 6, 2019

The Academy

- The American Academy of Actuaries is a 19,500-member professional association whose mission is to serve the public and the U.S. actuarial profession. For more than 50 years, the Academy has assisted public policymakers on all levels by providing leadership, objective expertise, and actuarial advice on risk and financial security issues. The Academy also sets qualification, practice, and professionalism standards for actuaries in the United States.

Seminar Host

- Roosevelt Mosley, MAAA, FCAS
  - Chairperson, Academy's Automobile Insurance Committee

Today's Agenda (Selective)

- Session 1 – Predictive Modeling "Cooking Show"
- Session 2 – Generalized Linear Models (GLMs)
  - Deep Dive Into GLMs
  - Going Beyond GLMs
- Session 3 – Practical Examples of Predictive Models
- Session 4 – Public Policy Discussion

***Note – only slides from Session 2 are available.**

GENERALIZED LINEAR MODELS:

A DEEP DIVE FOR INSURANCE APPLICATIONS

Predictive Modeling – Session 2A

Introductions

- Nathan Hubbell, FCAS
  - 2VP, Business Insurance R&D
  - Travelers
- Jeff Kinsey, MAAA, FCAS
  - P&C Actuarial Director, Research Unit
  - State Farm

Agenda

- Generalized Linear Models vs. Traditional Linear Models
- Model Building Process Key Steps

Credits

- Anderson, et al. A Practitioner's Guide to Generalized Linear Models
- Goldburd, et al. Generalized Linear Models for Insurance Rating

Linear Models

- Formulas

$$Y_i = E(Y_i) + \varepsilon_i$$

The actual value equals the expected value plus a residual

$$E(Y_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots$$

The expected value equals a combination of relevant factors

Linear Models

- Assumptions
  - **Random Component**: Each component of Y is independent and is normally distributed. The mean of each component is allowed to differ, but they all have common variance.
  - **Systematic Component**: The covariates are combined to give the linear predictor.
  - **Link Function**: The relationship between the random and systematic components is specified via a link function. For a linear model, the link function is the identity function.

Linear Models

- Limitations in Practice
  - Normality and constant variance are often not applicable for insurance applications
  - Normality assumptions means both positive and negative values are possible
  - Additive effects are not common

Generalized Linear Models

- Formulas

$$Y_i = E(Y_i) + \varepsilon_i$$

The actual value equals the expected value plus a residual

$$E(Y_i) = g^{-1}(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots)$$

The expected value equals a function of the combination of relevant factors

Generalized Linear Models

- Assumptions
  - **Random Component**: Each component of Y is independent and is from one of the exponential family of distributions
  - **Systematic Component**: The covariates are combined to give the linear predictor
  - **Link Function**: The relationship between the random and systematic components are specified via a link function that is differentiable and monotonic

Generalized Linear Models

- Exponential Family of Distribution

Distribution	Variance	Common Uses
Normal	Constant	--
Poisson	Varies with mean	Claim Frequency
Gamma	Varies with mean squared	Claim Severity
Tweedie	“Combo” of Poisson and Gamma	Loss Ratio

Generalized Linear Models

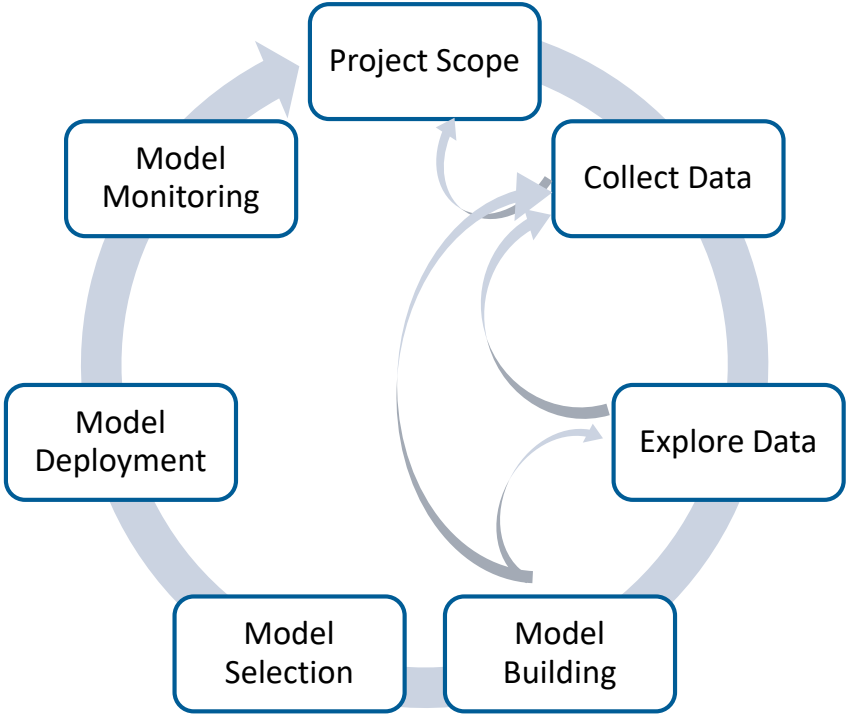
- Link Functions
  - Log link function is commonly used for insurance applications
  - Allows the covariate effects to be multiplicative rather than additive

$$E(Y_i) = e^{(\beta_1 X_{i1} + \beta_2 X_{i2} + \cdots)} = \left(e^{\beta_1 X_{i1}}\right) * \left(e^{\beta_2 X_{i2}}\right)$$

Generalized Linear Models

- Other Link Functions in Practice
  - Logit – used for binary target
    - Survival on Titanic, Customer Lapse
  - Identity – keeps “traditional” additive effects

Model Building Process



Model Building Process

Project Scope

Collect Data

Explore Data

Model Building

Model Selection

Model Deployment

Model Monitoring

- Somewhere between Collecting and Exploring…
  - Split data for specific purposes
    - One set to build models
    - Another to assess models
  - May elect to create three splits
  - Cross validation

Model Building Process

- Project Scope
- Collect Data
- Explore Data
- Model Building
- Model Selection
- Model Deployment
- Model Monitoring

- Model Selection
  - Performance of model compared to current or other challenger models
  - Perform well on all segments of business?

Model Building Process

Project Scope

Collect Data

Explore Data

Model Building

Model Selection

Model Deployment

Model Monitoring

- Model Deployment
  - How will the model be used?
  - Segmentation of business in “tiering” plan?
  - Inform “traditional” rating plan factors?
    - Move all the way to indicated? Weight with current rate plan?

Model Building Process

Project Scope

Collect Data

Explore Data

Model Building

Model Selection

Model Deployment

Model Monitoring

- Model monitoring
  - Ensures model continues to perform as expected
    - Changes in variables? Changes in performance?
    - Over time? Compared to model build?

Model Building Process

Project Scope

Collect Data

Explore Data

Model Building

Model Selection

Model Deployment

Model Monitoring

- Other Considerations
  - Model validation conducted?
  - Experience/credentials of modelers

Questions?

American Academy of Actuaries

Objective. Independent. Effective.™

BEYOND GLMS

Presenters

- Mark Jones, MAAA, ACAS
  - PWC
- Mike Woods, FCAS, CSPA
  - Allstate

Machine Learning/Artificial Intelligence Overview

Advantages

- **Higher-capacity models** can represent more complex data generating processes possibly achieving greater accuracy cost-effectively
- Flexible architecture capable of consuming a broad **range of data sources**
- Large range of functional application suitable for the **varied needs** of insurers

Disadvantages

- **Opaqueness** of many model types incompatible with a business need for "reasons"
- Many architectures require large amounts of **data** for sufficient training and validation
- Can require a much higher degree of **knowledge** in numerical analysis, programming and hardware to skillfully apply
- Higher-capacity models **overfit** easier
- Requires **significant resources** to tune the performance

A General ML/AI Framework

"A computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P***, if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E.***" — T.M. Mitchell

P depends on ***T***

Accuracy Error LogLoss MSE

E is continually generated by the process

Unsupervised $p(x)$ Supervised $p(y|x)$

Semi-supervised $p(.)$ Reinforcement $p(x(t))$

Typical tasks T

Depends on the purpose and determines how observations are processed

- Synthesis or Sampling
- Imputation
- De-noising
- Density Estimation
- Classification
- Regression
- Transcription or Translation
- Structured Output
- Anomaly Detection

Model Generalization

For a good model

1. Training error e_{train} is small
2. Gap between train and test error e_{train} - e_{test} is small

Generalization

Distinguishes machine learning from ***optimization***

Generalization error is the expected value of error over the range of possible inputs the machine will encounter

The inputs are assumed to be generated from a ***data generating process*** and *iid* according to a common ***data-generating distribution***

For a randomly selected model:

$$E[e_{train}] = E[e_{test}]$$

Model Capacity

capacity

a model with higher capacity has the ability to model more relationships between more variables than a model with a lower capacity

the ability to fit a wider range of functions

how complex a relationship it can model

Capacity and Generalization

- **Underfitting** occurs when model is not able to obtain as low an error value as is possible on the training set while maintaining generalizability
- **Overfitting** is the analogous situation where the model fits to noise present in the data and not the underlying data generating process

There is greater risk when a model's capacity greatly exceeds the complexity of the task.

Considered best practice by many that the first model built should always be designed to overfit.

Best to understand the boundaries of the model sooner rather than later.

The more a model has

capacity

the more opportunity it has to

overfit

Modifying Capacity Through Regularization

- Regularization discourages learning a more complex or flexible model, so as to avoid the risk of overfitting
- This can be achieved by adding a penalty term in the loss function that adds cost for increased complexity
- Modification made to the learning algorithm to reduce generalization error but not training error
- There are other ways to regularize:
  - Dropout
  - Bagging
  - Early stopping

L1 Regularization penalizes based on the absolute value of the parameters. Penalty may force some parameters to zero and so can be used for feature selection.

L2 Regularization penalizes based on the square value of the parameter. The penalty forces all parameters to be non-zero and so forces all information to be retained.

Modifying Capacity Through Hyperparameters

- Hyperparameters are the parameters of the model that are set prior to learning
- They are not optimized as part of the training process because they would overfit every single time
- Generally, hyperparameters affect capacity

Tree depth

Learning rate

Min observations per node

Hidden layers

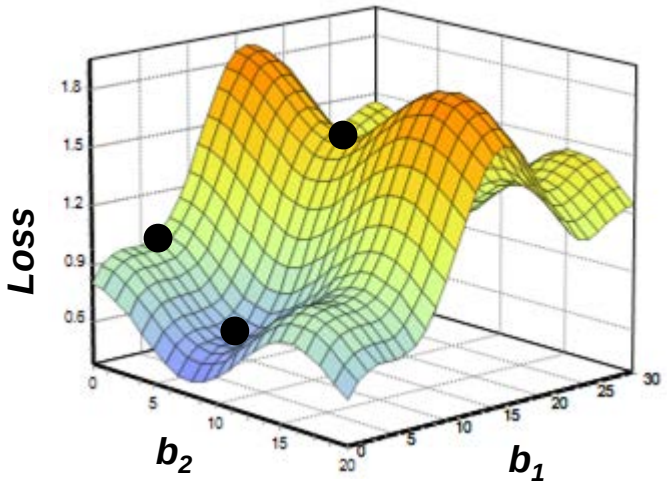
Folds

Number of trees

Sampling rate

Tweedie power

Modifying Capacity Through Hyperparameters



Hyperparameter Tuning

- Hyperparameters define a n-dimensional loss surface that may not be well-behaved
- Selecting the optimal set of hyperparameters is not a trivial task
- The final selection may have a significant impact on the model's performance and generalization error

Common approaches

Grid searching

Random searching

Bayesian optimization

Grid and Random Searching

Define a n-dimensional grid with potential combinations of hyperparameters $\mathbf{H} \subseteq R^n$

Fit model and test performance on elements in **H**

Exhaustive: Use all combination in **H**

Random: Use some random subset $\mathbf{H}_1$ of **H**

Refine the grid $\mathbf{H}^1 \subseteq \mathbf{H}$ based on the regions of minimum loss / best performance

Repeat until the desired level of loss / performance is reached

Consider other stopping criteria

Resource- and time-consuming

May require a more simplified version of the model to be fit for each combination of hyperparameters

Does not guarantee the global minimum

Bayesian Hyperparameter Estimation

- Evaluations of the loss function (minimized in training) is resource-intensive — minimize the number of calls to this function
- Grid & random searching is uninformed by past evaluations
- BHE can be used to leverage historical calls to the loss function to make more informed decisions on future hyperparameters to test

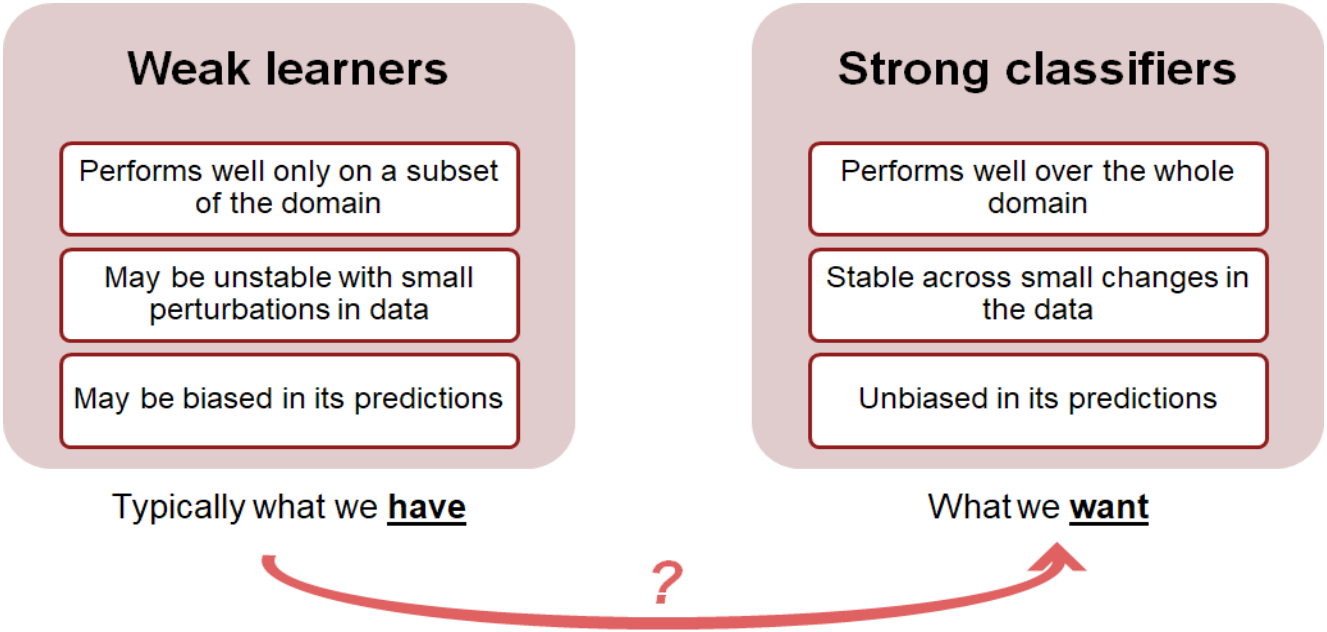
- Define a “surrogate” probability distribution of the loss function given the hyperparameters
- Find the parameters that maximize the expected improvement in the loss function
- Evaluate the loss function for the selected parameters
- Update the surrogate with the new information
- Repeat 2 - 4 until stopping criteria met

May produce better test set performance, in fewer iterations than a grid or random search

Relatively simple to understand but may be difficult to communicate

Relies on a selection of a prior “surrogate” for the loss function

Weak Learners and Strong Classifiers



Bagging – Bootstrap Algorithm

Algorithm

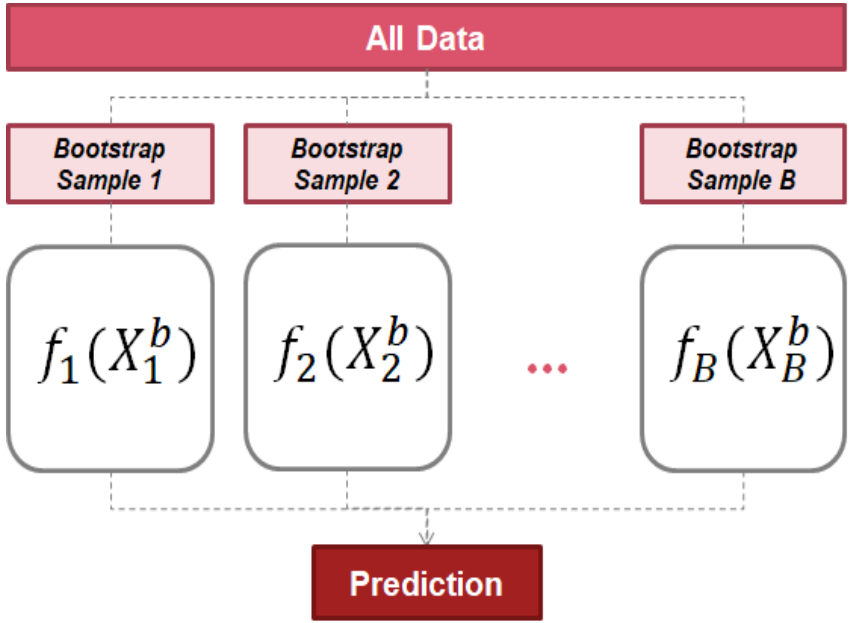
1. Create bootstrap resample of data
2. Fit model on each resample
3. Scoring:
   - Classification: Majority vote
   - Regression: Mean/Median score

Advantages

- Produces more stable predictions – i.e. reduces variance
- Less likely to overfit data

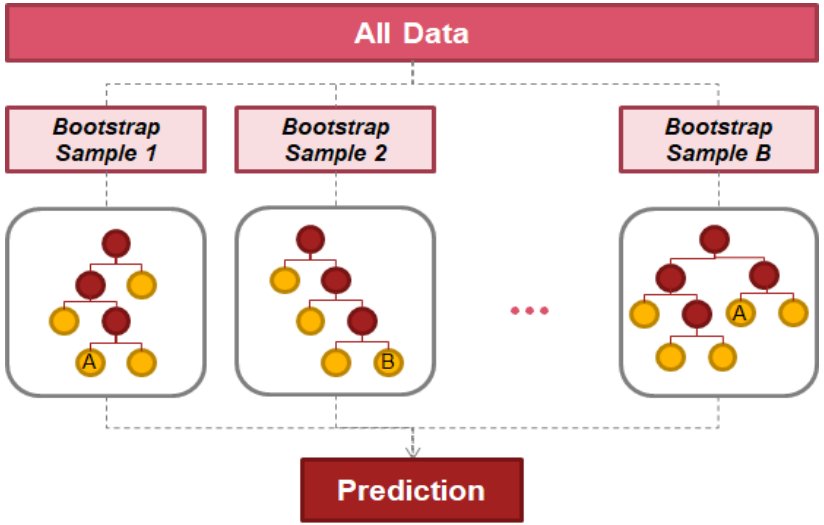
Disadvantages

- Generates a “black box”



Random Forests – Bagging Decision Trees

- Introduced by Leo Breiman (2001)
- Uses bagging to improve decision trees
- De-correlates trees by sampling
  - Data with replacement
  - Columns/features at each node
- Produces out-of-bag error rates
- Produces variable importance measure



Random Forests – Parameters

Tree Parameters	Impact and Considerations
Number of features to select at each node	• Prevent **overfitting** • Produce more diverse trees & discover **hidden** relationships
Maximum depth	• Deeper trees allow for more complex **interactions** • Deeper trees allow for **less biased** predictions • Shallower trees result in a lower propensity to **overfit**
Minimum observations per node	• Fewer observations per node increase **node purity** • More observations per node reduces propensity to **overfit**

Bagging Parameters	Impact and Considerations
Number of trees	• More trees result in a lower prediction **variance** • More trees may increase the propensity to **overfit** • **Stopping criteria** may limit the actual number of trees fit
Sampling rate	• A lower sampling rate results in a lower propensity to **overfit**

Boosting

Algorithm

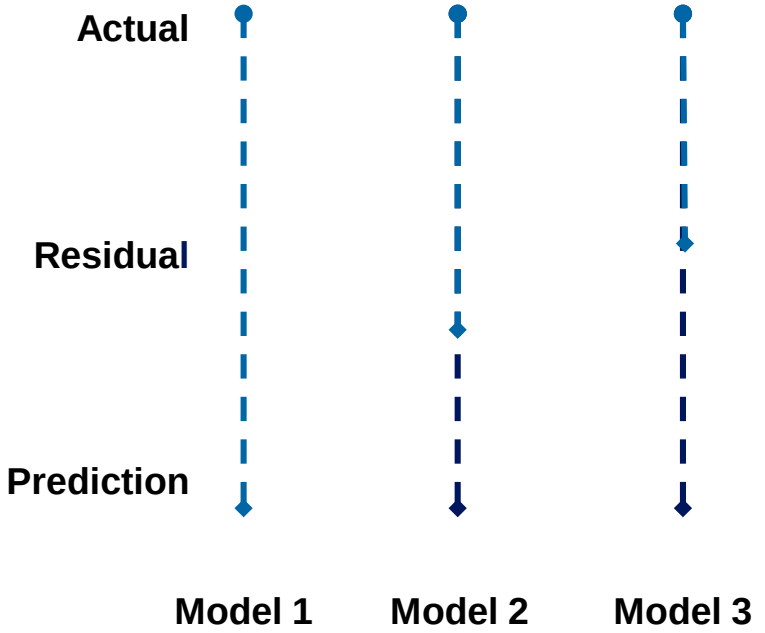
- Rather than fitting models to bootstrap samples of the data – boosting fits sequential models focusing on areas of poor performance
- Subsequent models correct errors of previous models

Advantages

- Decrease bias in predictions

Disadvantages

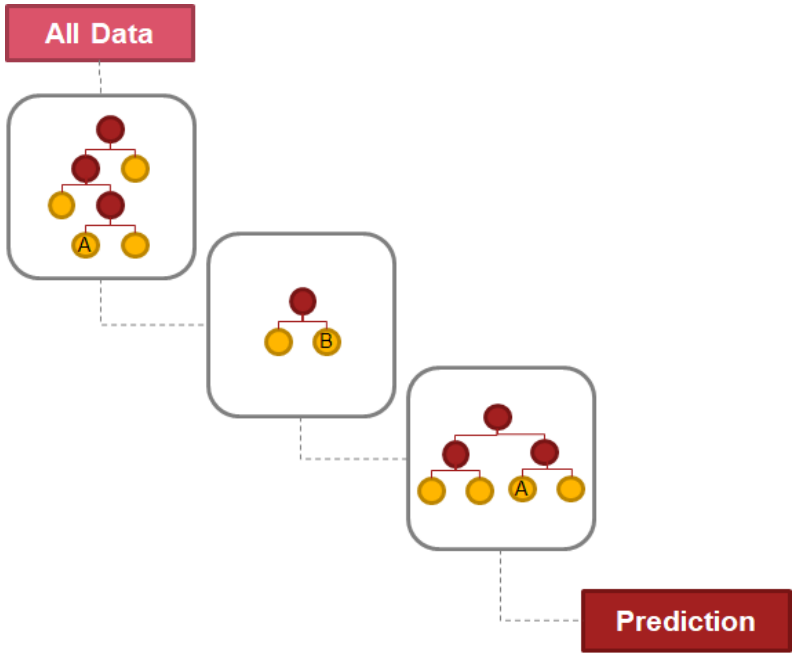
- May overfit the data
- Generates a “black box”
- May be sensitive to outliers and noise



Gradient Boosted Machines (GBM)

Boosting Decision Trees

- Introduced by Jerome Friedman (1999)
- Uses boosting to improve decision trees
- XGBoost algorithm most common
  - Stochastic gradient descent
  - Feature sub-sampling
- ➓ LightGBM is latest and greatest
  - Much faster
  - Different approach to finding splits
  - Feature bundling
- ➓ GBMs can vary significantly by implementation



Gradient Boosted Machines

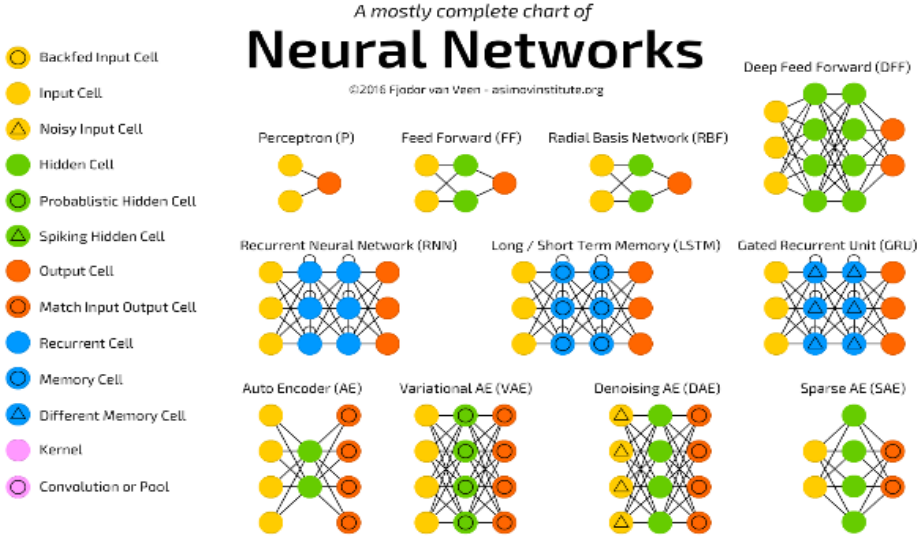
Parameters

Tree Parameters	Impact and Considerations
Number of features to select at each node	• Prevent **overfitting** • Produce more diverse trees & discover **hidden** relationships
Maximum depth	• Deeper trees allow for more complex **interactions** • Deeper trees allow for **less biased** predictions • Shallower trees result in a lower propensity to **overfit**
Minimum observations per node	• Fewer observations per node increase **node purity** • More observations per node reduces propensity to **overfit**

Boosting Parameters	Impact and Considerations
Learning rate	• A lower learning rate decreases the **impact** of any one individual tree • Slows down the **rate** of fitting
Number of trees	• More trees result in a lower prediction **variance** • More trees may increase the propensity to **overfit** • **Stopping criteria** may limit the actual number of trees fit
Sampling rate	• A lower sampling rate results in a lower propensity to **overfit**

Neural Networks
Briefly

- Highly flexible architecture based (loosely) on organic neural connections where layers of neurons are connected (fully or partially) to one another
- The capacity of a NN is its ability to extract feature representations via a composition of functions
- Activation functions for each layer determine each nodes state
- Weights for each layer are optimized via stochastic gradient descent and backpropagation
- Specialized architectures (e.g., LSTM & Convolutional) exist for specific tasks (e.g., sequence prediction & image recognition)



Advanced Models - Regulatory Concerns

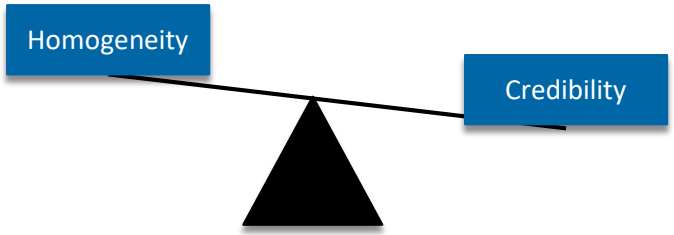
- Protected Classes
- Fitting to Noise
- Monotonicity of Continuous Variable Levels
- Intuitiveness of Discrete Variable Levels
- Reason Codes

Protected Classes

- Examine data that machine learning algorithm is allowed to use
  - Assess whether data could lead to identification of protected classes
- Examples
  - Pricing model
    - Examine list of variables considered by model
  - Image-based models
    - Ensure model only uses image data of subject matter

Fitting to Noise

- Every risk classification system must strike balance between homogeneity and credibility



- Determine if advanced model type has parameters that dictate credibility of decisions
  - GBM
    - Minimum Rows: determines minimum number of observations needed for a split

Monotonicity of Continuous Variables

- Monotonicity: “varying in such a way that it either never decreases or never increases”
- Certain continuous variables should have a monotonicity constraint applied, such as:
  - Number of Accidents in Last Three Years
  - Insurance Limit
- Modeler should apply monotonicity constraint in modeling process
  - GBM
    - Certain implementations of GBMs, such as XGBoost, allow for monotonicity constraints to be specified

Intuitiveness of Discrete Variable Levels

- Example
  - Multi-Policy Discount (MPD) on an Auto Policy
    - Want to check for intuitive relationship between four levels of the MPD
      - No other policies
      - Personal Umbrella Policy
      - Homeowners Policy
      - Homeowners & Personal Umbrella Policy

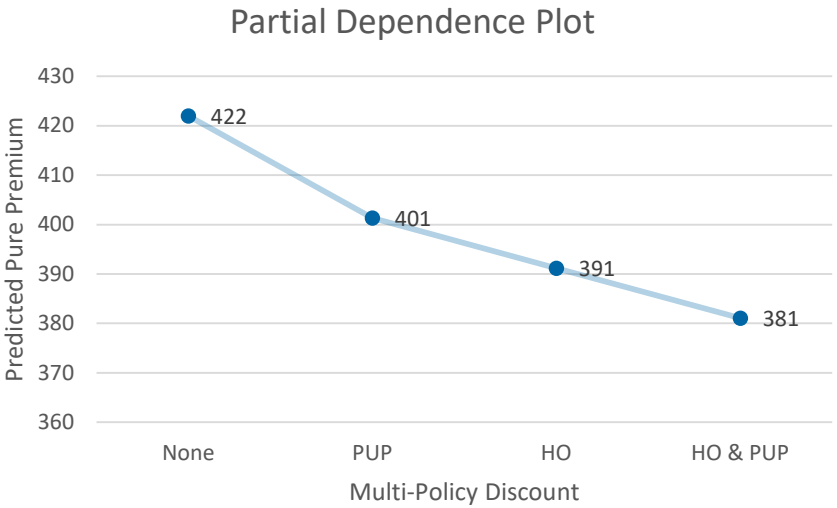
- Solutions
  - Partial Dependence Plot
  - Individual Conditional Expectations Plot

Partial Dependence Plot

- Partial dependence function

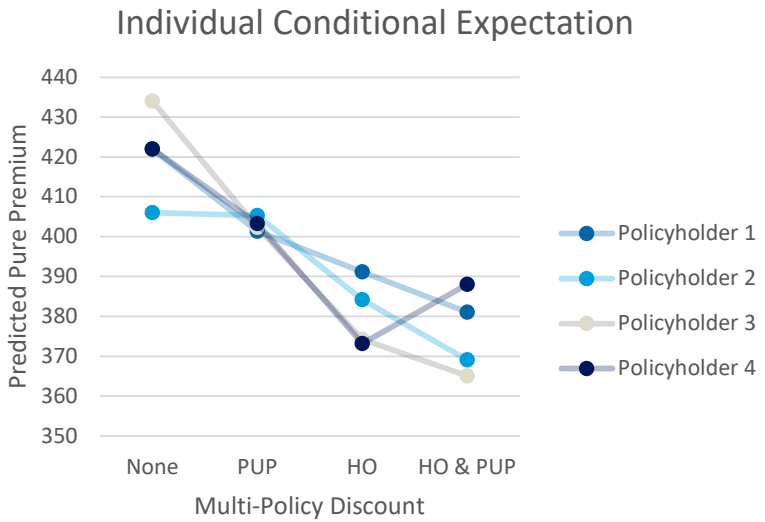
$$f_{x_s}(x_s) = \frac{1}{n}\sum_{i=1}^{n}\left(f(x_s, x_c^{(i)})\right)$$

$\boldsymbol{f}$ = machine learning model
$\boldsymbol{x_s}$ = feature set to be plotted
$\boldsymbol{x_c^{(i)}}$ = actual feature values from dataset in which we are not interested
n = number of observations



Individual Conditional Expectation

- Individual Conditional Expectation (ICE) plot is similar to PDP
  - PDP showed average effect of a feature across dataset
  - ICE shows effect of feature on each observation in dataset



Reason Codes

- How to educate insureds on how to lower premium?
- How to inform insureds why premium went up at renewal (when applicable)?
  - Insurance companies can prepare information, using PDPs and ICEs, on variables that are likely to increase or decrease cost of insurance for individual insureds

Beyond GLMs

Questions?

PRACTICAL EXAMPLES OF PREDICTIVE MODELING

Presenter

- Dorothy Andrews, MAAA, ASA
  - Chairperson, Academy's Data Science and Analytics Committee

PREDICTIVE MODELING

PANEL DISCUSSION OF PUBLIC POLICY QUESTIONS

Moderator

- Rich Gibson, MAAA, FCAS
  - Senior Property/Casualty Fellow, American Academy of Actuaries

Panelists

- Birny Birnbaum
  - Center for Economic Justice
- Kevin Dyke, MAAA, FCAS
  - Michigan Dept. of Insurance and Financial Services
- Mike Woods, FCAS
  - Allstate

Predictive Modeling – Public Policy

Discussion

Contact Us

For more information, contact:

Marc Rosenberg, senior casualty policy analyst

rosenberg@actuary.org or (202) 785-7865