SOCIETY OF ACTUARIES®

# 2021 LIFE MEETING

August 30-September 1

20
21

# SOA Antitrust Compliance Guidelines

Active participation in the Society of Actuaries is an important aspect of membership. While the positive contributions of professional societies and associations are well-recognized and encouraged, association activities are vulnerable to close antitrust scrutiny. By their very nature, associations bring together industry competitors and other market participants.
The United States antitrust laws aim to protect consumers by preserving the free economy and prohibiting anti-competitive business practices; they promote competition. There are both state and federal antitrust laws, although state antitrust laws closely follow federal law. The Sherman Act, is the primary U.S. antitrust law pertaining to association activities. The Sherman Act prohibits every contract, combination or conspiracy that places an unreasonable restraint on trade. There are, however, some activities that are illegal under all circumstances, such as price fixing, market allocation and collusive bidding.

There is no safe harbor under the antitrust law for professional association activities. Therefore, association meeting participants should refrain from discussing any activity that could potentially be construed as having an anti-competitive effect. Discussions relating to product or service pricing, market allocations, membership restrictions, product standardization or other conditions on trade could arguably be perceived as a restraint on trade and may expose the SOA and its members to antitrust enforcement procedures.

While participating in all SOA in person meetings, webinars, teleconferences or side discussions, you should avoid discussing competitively sensitive information with competitors and follow these guidelines:

- Do not discuss prices for services or products or anything else that might affect prices
- Do not discuss what you or other entities plan to do in a particular geographic or product markets or with particular customers.
- Do not speak on behalf of the SOA or any of its committees unless specifically authorized to do so.
- Do leave a meeting where any anticompetitive pricing or market allocation discussion occurs.
- Do alert SOA staff and/or legal counsel to any concerning discussions
- Do consult with legal counsel before raising any matter or making a statement that may involve competitively sensitive information.

Adherence to these guidelines involves not only avoidance of antitrust violations, but avoidance of behavior which might be so construed. These guidelines only provide an overview of prohibited activities. SOA legal counsel reviews meeting agenda and materials as deemed appropriate and any discussion that departs from the formal agenda should be scrutinized carefully. Antitrust compliance is everyone’s responsibility; however, please seek legal counsel if you have any questions or concerns.

# Presentation Disclaimer

*Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the participants individually and, unless expressly stated to the contrary, are not the opinion or position of the Society of Actuaries, its cosponsors or its committees. The Society of Actuaries does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented. Attendees should note that the sessions are audio-recorded and may be published in various media, including print, audio and video formats without further notice.*

# Agenda

Data Science Techniques Implementation Process

Data Science Technique Applications:

**1. Financial data modeling:**

- Insurance Company Fund mapping
- Volatility modeling

**2. Data quality example:**

- Outlier detection

**3. Policyholder behavior modeling example:**

- LTC incidence rate experience study

**4. Runtime Reduction examples:**

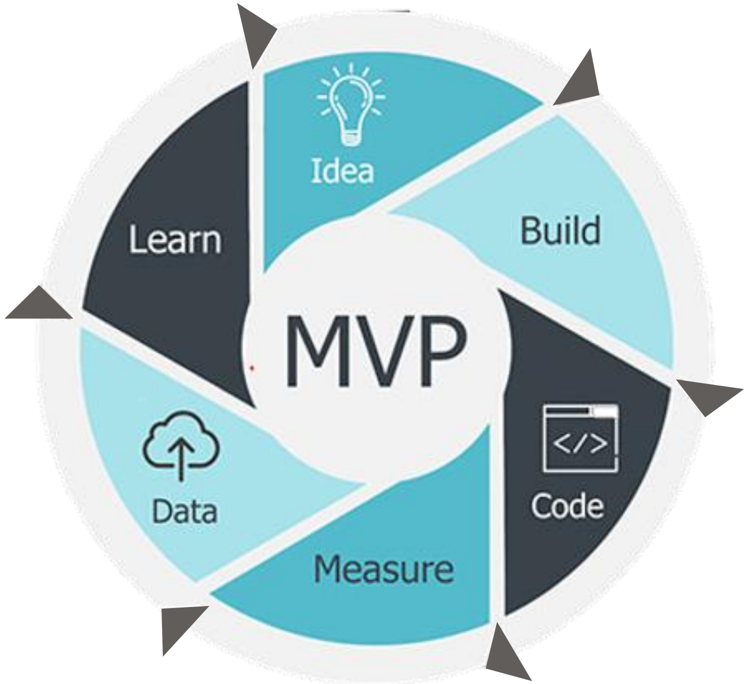
- Inforce compression
- Output prediction

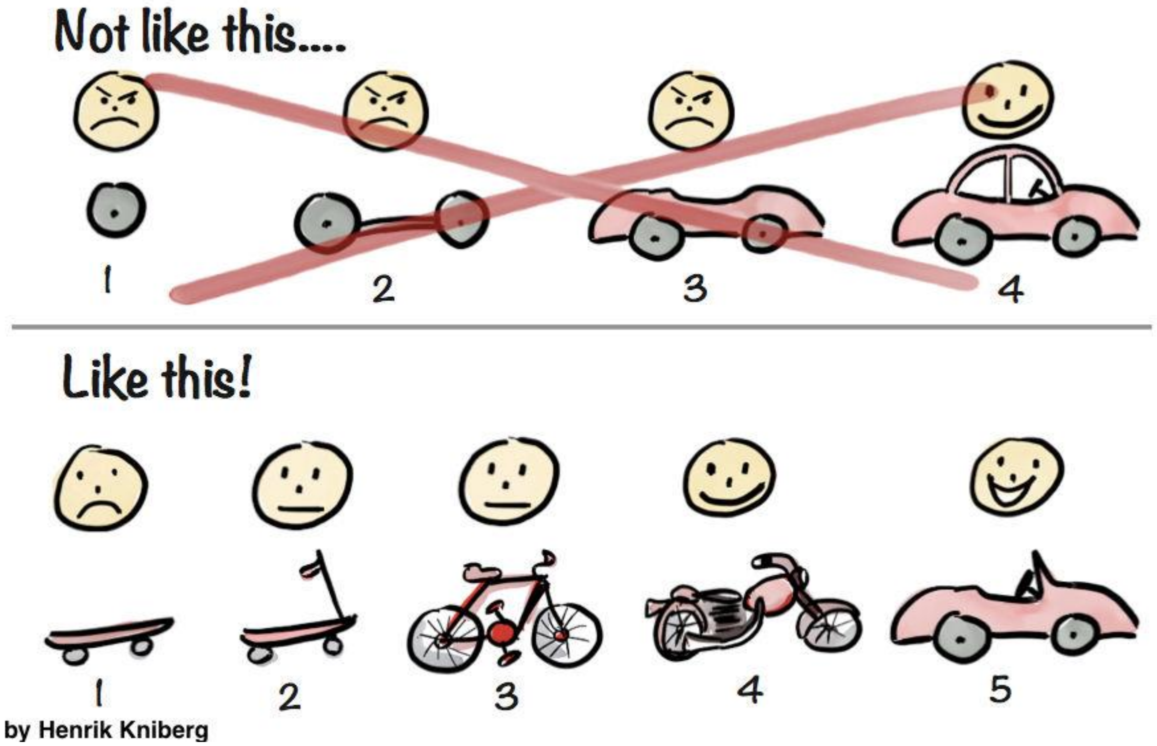
Case study: Neural Network application for Liability Prediction

Github sample code repo: https://github.com/yjing0926/SOA_Sample_Code.git

# Data Science Techniques Implementation

- MVP (Minimum Viable Product): Functional | Reliable | Usable | Design
- Start small and iterate



# Financial Data Modeling

Application (time series | supervised ML):

- Use regression to map insurance company mutual funds to tradeable indices for liability hedging or to the AAA ESG funds for Statutory reserve

MVP evolution:

**OLS regression**

- Converted from Excel to Python
- Expanded metrics from adj $R^2$ to adj $R^2$ and RMSE
- Train/test set split
- Cross validation

**Regularized regression**

- Lasso, Ridge, ElasticNet, etc
- Hyperparameter tuning using GridSearch
- Outlier treatment

**Streamline analyses**

- Visualization
- Different lookback period and return frequency
- Rolling window

**API connection**

- Streamline data collection
- Streamline end to end process

Benefits: efficient process | stable results | easy implementation & experiments

# Financial Data Modeling

Application (time series | supervised ML):

- Simulate risk-managed target volatility fund returns using volatility signals

Evaluation metric is LLF (loglikelihood)

MVP evolution:

**GARCH with simulation** →
- Garch(1,1) volatility model to simulate volatility signal

**Improve distribution** →
- Normal vs t distribution

**Research other volatility models** →
- EWMA
- FIGARCH
- EGARCH

**Hyperparameter tuning**
- P, q, d parameter tuning

Benefits: variety of readily available open source solutions

# Data Quality Validation

Application (time series | unsupervised ML):

- Seriatim inforce data scrubbing to detected unlabeled data anomalies

The workflow of inforce checks as below:

**Input data** → **Univariate layer** → **Multivariate layer** → **Output results**

- **Input data**
  - Input time series data of a policy
- **Univariate layer**
  - Apply Isolation Forest to each attribute time series
- **Multivariate layer**
  - Apply Isolation Forest to multi-attribute time series
- **Output results**
  - Output the detection result of this policy

Benefits:

- First line of defense; early detection of data issues
- Parallel processing enabled for scalability
  - GridStep, Spark/Hadoop, GCP, etc
- Ability to deal with high dimensional data involving a large number of attributes
- Linear time complexity at a low memory requirement

White paper by Hyunsu Kim, Michael Leitschkis (2019, December). See the forest for the trees. At https://www.milliman.com/en/Insight/See-the-forest-for-the-trees
Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413-422). IEEE.

# Policyholder Behavior Modeling

Application (discrete labelled data | supervised ML):

- LTC termination rates assumption setting

**Data Exploration pre modeling**

- Univariate review
- Multivariate review
- Transformation

**Modeling**

- Model selection
- Feature selection

**Model Evaluation**

- Standard analyses
- Cross validation

Data exploration pre modeling:

- Natural log of exposure/target to predict rate of termination
- Univariate review: looking for non-linear relationship and transform those variables
- Multivariate review: looking for highly correlated fields and avoid using them together

# Policyholder Behavior Modeling

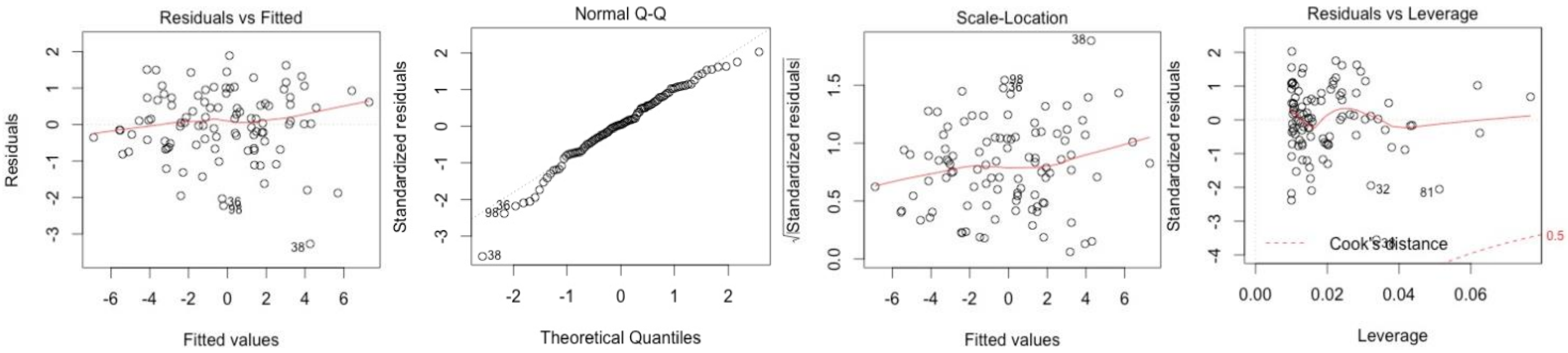
Modeling:

- Use Poisson GLM because the underlying is discrete counts
- Feature selection:
  - Un-transformed fields
  - Transformed fields, e.g. square, cube, ordinal, encoding
  - Interaction fields (requires in-depth business knowledge), e.g. age * duration
  - New fields (requires in-depth business knowledge), e.g. reverse ordinal

# Policyholder Behavior Modeling

Model Evaluation:

- Standard analyses:




- Cross validation: check for model stability
- Train set / holdout set, compare A/E, MSE, NLL, AIC, BIC
- Parsimonious model: check p-value of variables, use penalized GLM, use feature selection model

# Runtime Reduction

Application (discrete unlabelled data | unsupervised ML):

- Compress seriatim inforce using Kmeans clustering

**Data pre processing**

- Data transformation
- Feature selection including creating new features such as ITMness

**Kmeans clustering**

- Perform Kmeans clustering
- Fine Kmeans parameters and feature weights

**Data post processing**

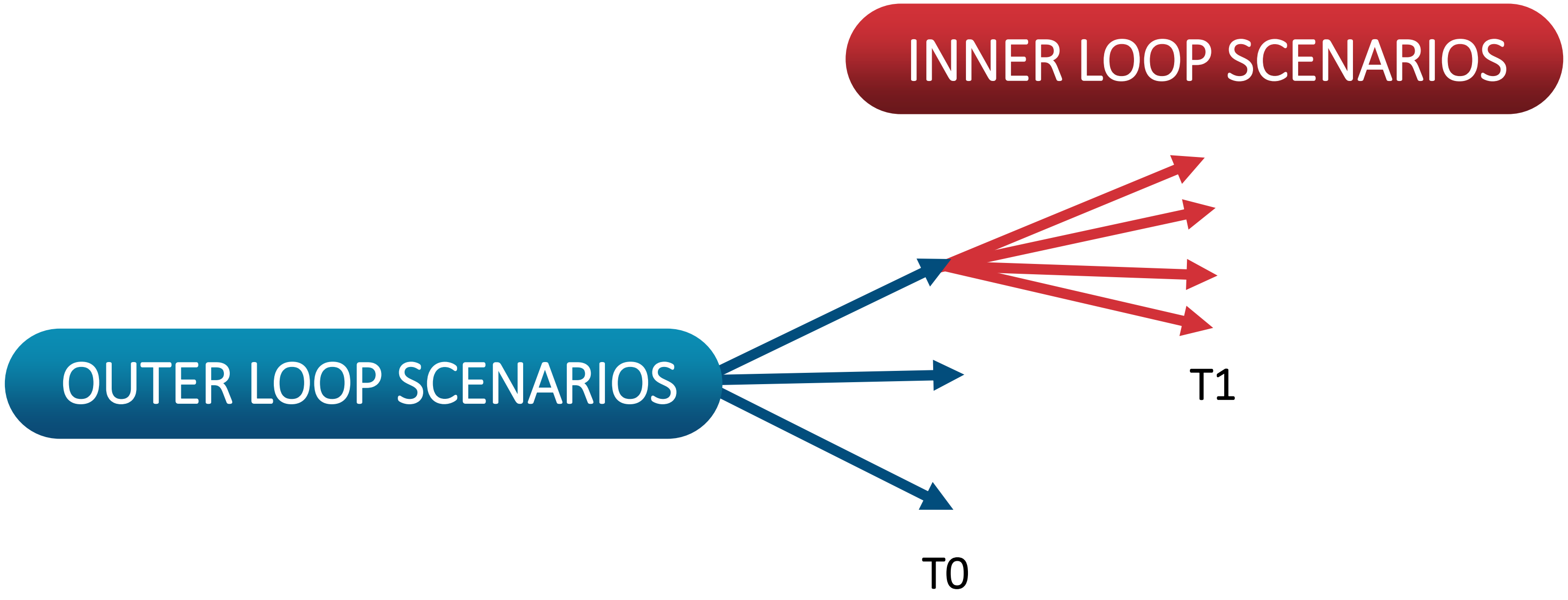
- Reverse transformation
- Determine how to summarize compressed data

- Compressed 500k policies to 400 records with <4% Option Value difference
- For variables that models are very sensitive to, either manually group or put more weights on those variables during Kmeans. For example, for VA, withdrawal delay
- Blending assumptions for each cluster also helps reduce mismatch

20
21

# Runtimes Issues with Variable Annuities

- **Variable annuities (VA)** are a complex product consisting of an embedded exotic put option on the equity market
  - Typical approach for risk neutral valuation is through Monte Carlo simulation
  - **Problem**: Runtime becomes onerous for insurance applications that require valuations along each timestep of a scenario (i.e. nested stochastic)



# Solutions to Runtime Issues

## Common Solutions

- Decrease the number of scenarios
- Decrease inforce size by grouping policies into clusters
- Simplified analytic approximation (i.e. implicit method for hedging)

## Alternative Solution

- Train a machine learning model to predict the liability value

# Potential Applications

- Explicit CDHS modeling under VM21
  - Requires projected risk-neutral liability values and corresponding Greeks (i.e. delta, rho) that will be hedged
- Projection of capital
  - Regulatory requirements may require projection of metrics (such as CTE98) under different real world scenarios
- Financial planning / forecast of balance sheets

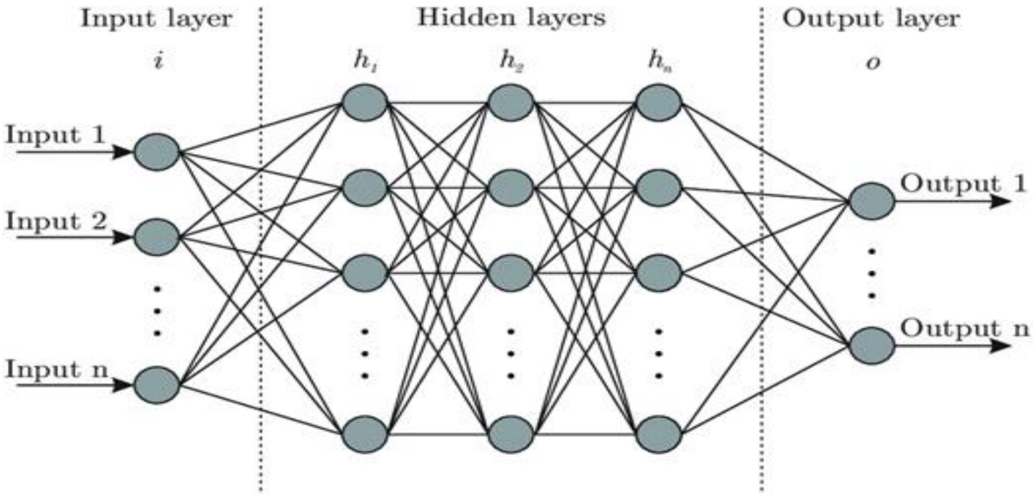
# Potential Machine Learning Model Choices

- Linear Regression
- Decision Trees
- Support Vector Machines
- Random Forests
- Gradient Boosting Machines
- Neural Networks

For this application, we used neural networks to predict the liability value of a variable annuity based on policy features and market data

# Neural Network: Overview

- Consists of:
  - **Input layer** – reads in features of the data
  - **Hidden layers** - perform nonlinear calculations
  - **Output layer** – uses an activation function that maps to the desired output format
- Architecture allows the capability to learn complex, nonlinear relationships between the input features and the output




| INPUTS | OUTPUTS |
|---|---|
| Age | Liability Value |
| Withdrawal Delay | |
| Interest Rates | |

# Neural Network: Feature Selection

- The features of the input data reflect key policy attributes (age, withdrawal delay) and market data (interest rates, moneyness)
- Dimension reduction techniques were used to avoid a high dimensional dataset
  - Principal component analysis for interest rate curves
  - Ignore policy features with an immaterial impact on the liability value
- Final input dataset had less than 30 features

# Neural Network: Generating Training Data

Generate policies that capture how the insurance inforce evolves over time

- Step 1: For categorical features, perform stratified sampling based on the inforce
  - Product type (GMWB, GMDB, Combo)
  - Mortality tables
- Step 2: For numerical features, randomly sample data to match the expected distribution of the inforce over a projection period
  - *Example:* Randomly sample ages from 30 to 100, with more weight given to more common ages (i.e. 60-80)

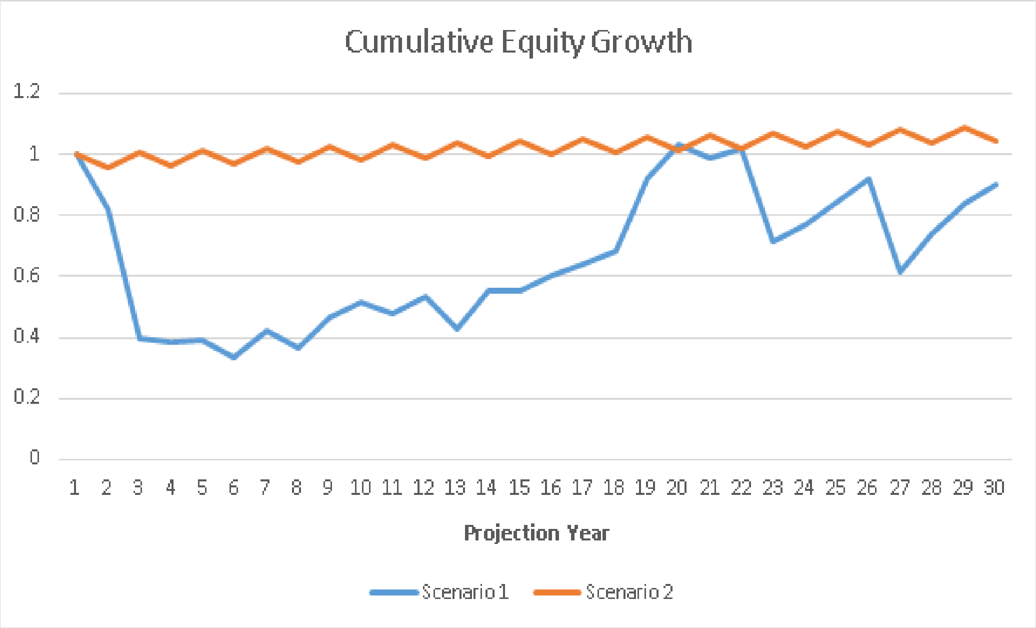
# Neural Network: Training Process

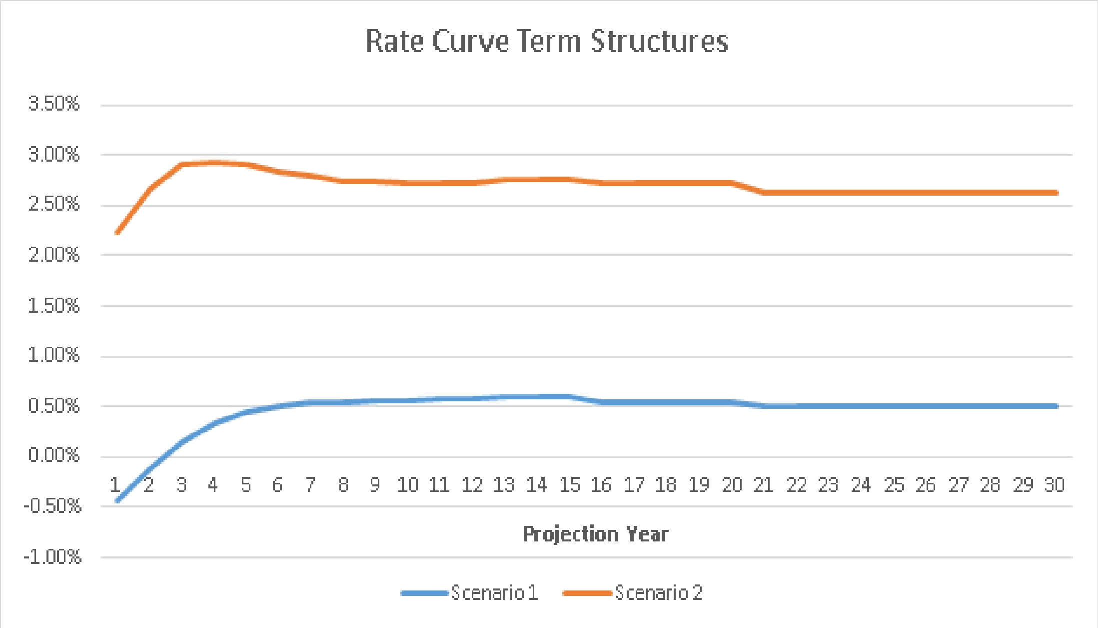
- Perform Monte Carlo simulation on the training data to calculate the target risk neutral liability value (RN LOV) for each policy
  - RN LOV = PV(Claims) - PV(Premiums)
- Objective is to solve for the parameters of the neural network that minimizes the MSE (mean squared error) between the neural network's predictions and the Monte Carlo RN value of the training data
- The training process requires the most runtime overhead

# Neural Network: Validation/Testing

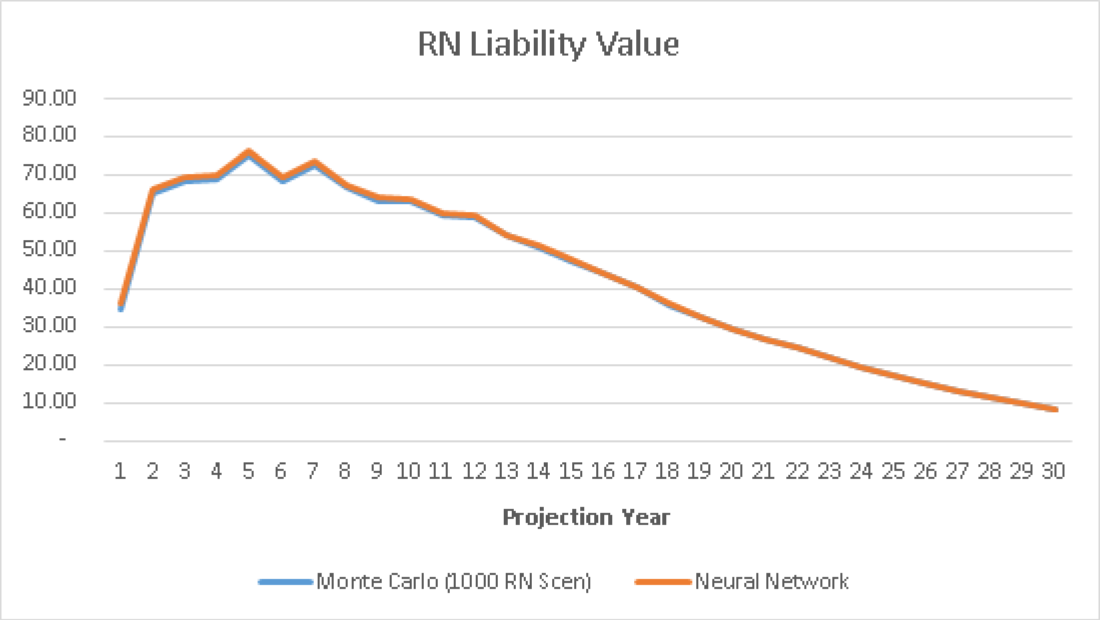
- Metrics used for evaluating the performance of the model
  - Base RN liability value
  - Equity Delta of RN LOV
- The above values were key inputs to the hedging strategy
- Apply trained neural network to predict base RN value and Delta for each policy in the inforce along two scenarios (for 30 years) reflecting different equity returns and interest rates
  - Scenario 1: Large equity crash, low rates
  - Scenario 2: Equities steadily grow with seesaw pattern, high rates
- Compare results to the values obtained using Monte Carlo simulation

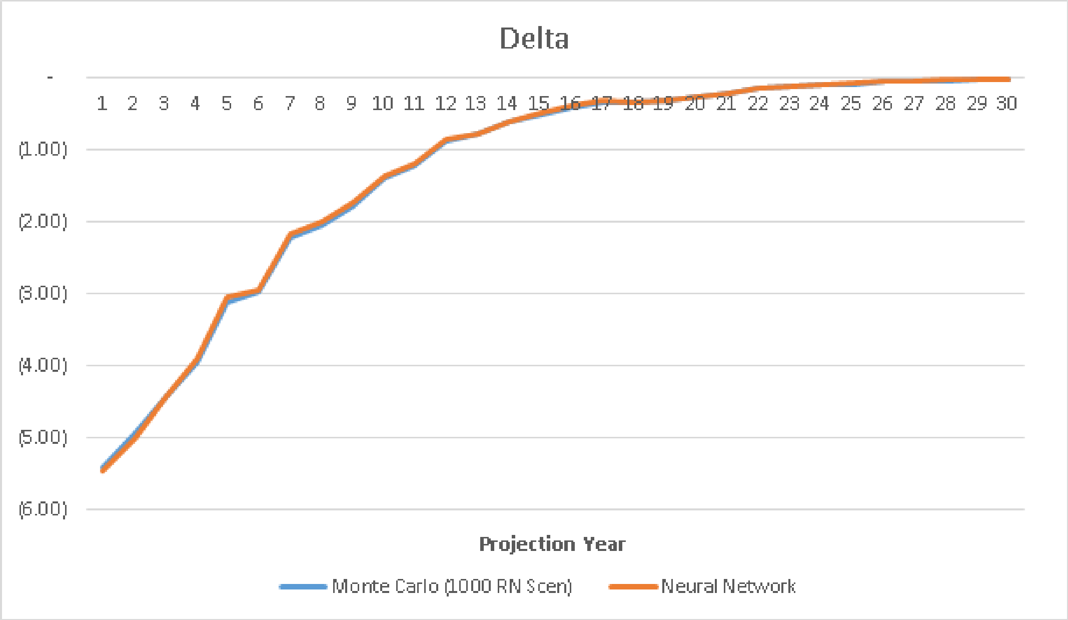
# Equity Return and Rate Curves



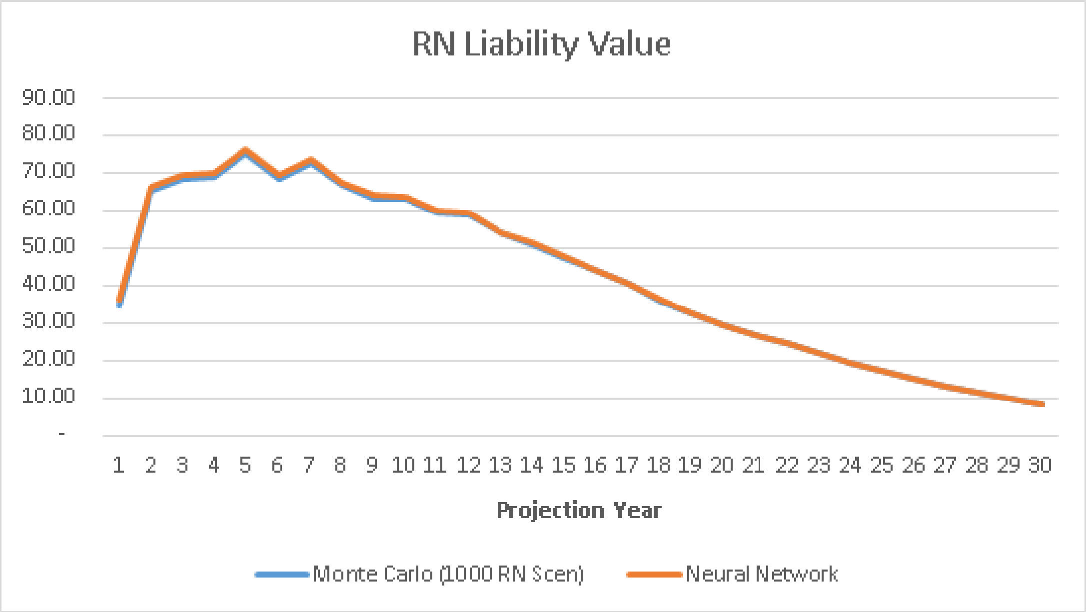



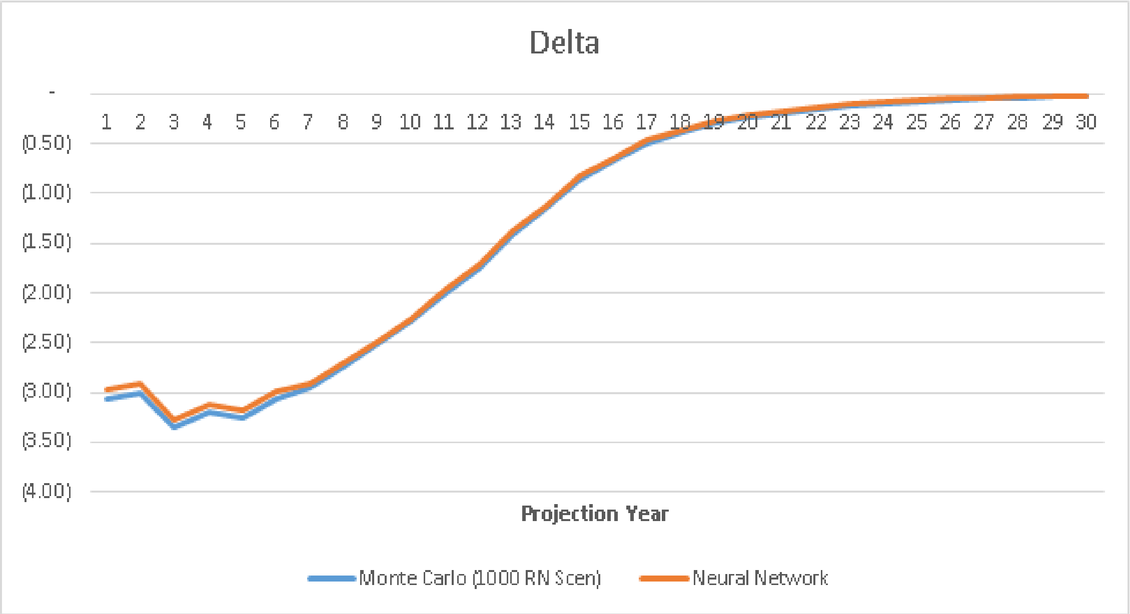
# Scenario 1 Results Comparison



# Scenario 2 Results Comparison



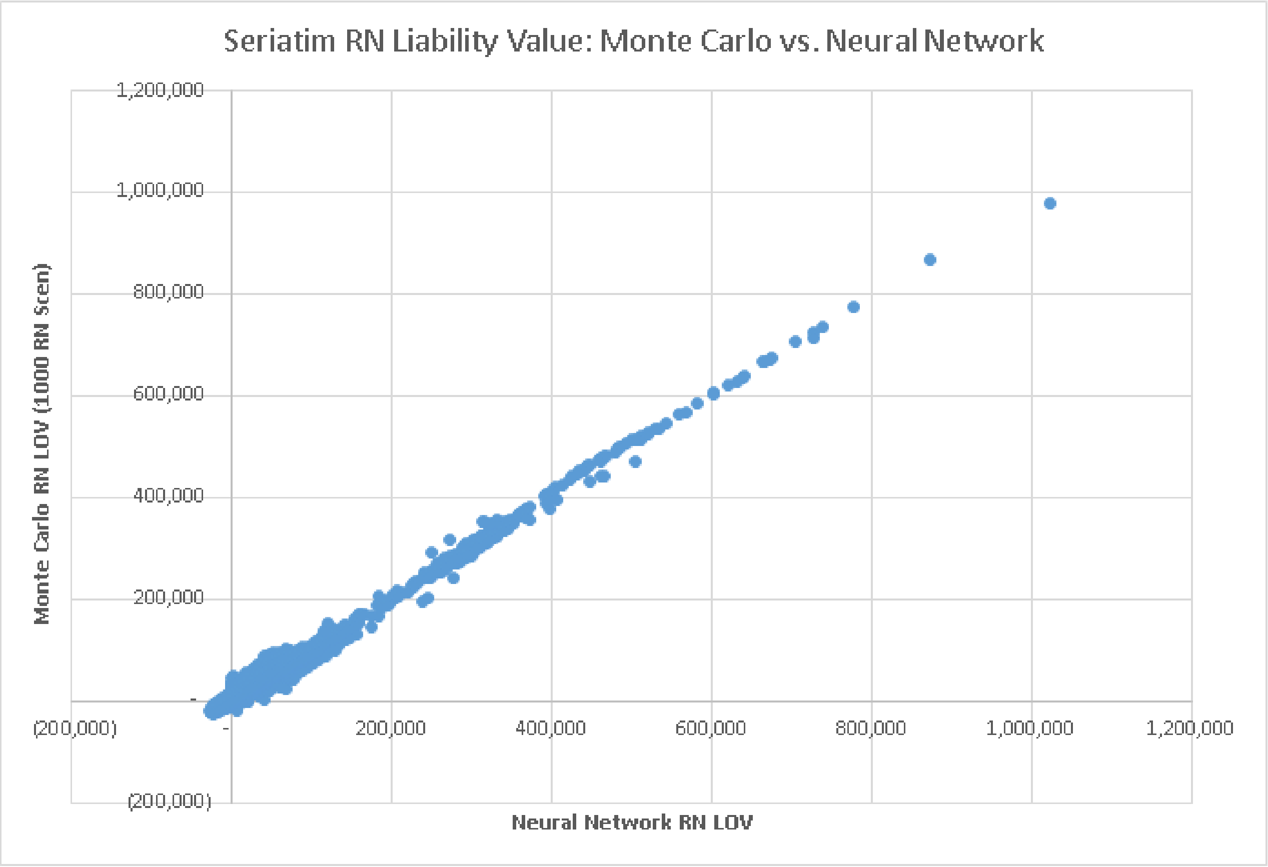



# Runtime Comparison (CPU Hours)

| | NEURAL NETWORK | MONTE CARLO SIMULATION – 1000 RISK NEUTRAL SCENARIOS |
|---|---|---|
| RN Projection for 30 years, 4 shocks along a single scenario | 30 minutes | ~9000 hours |

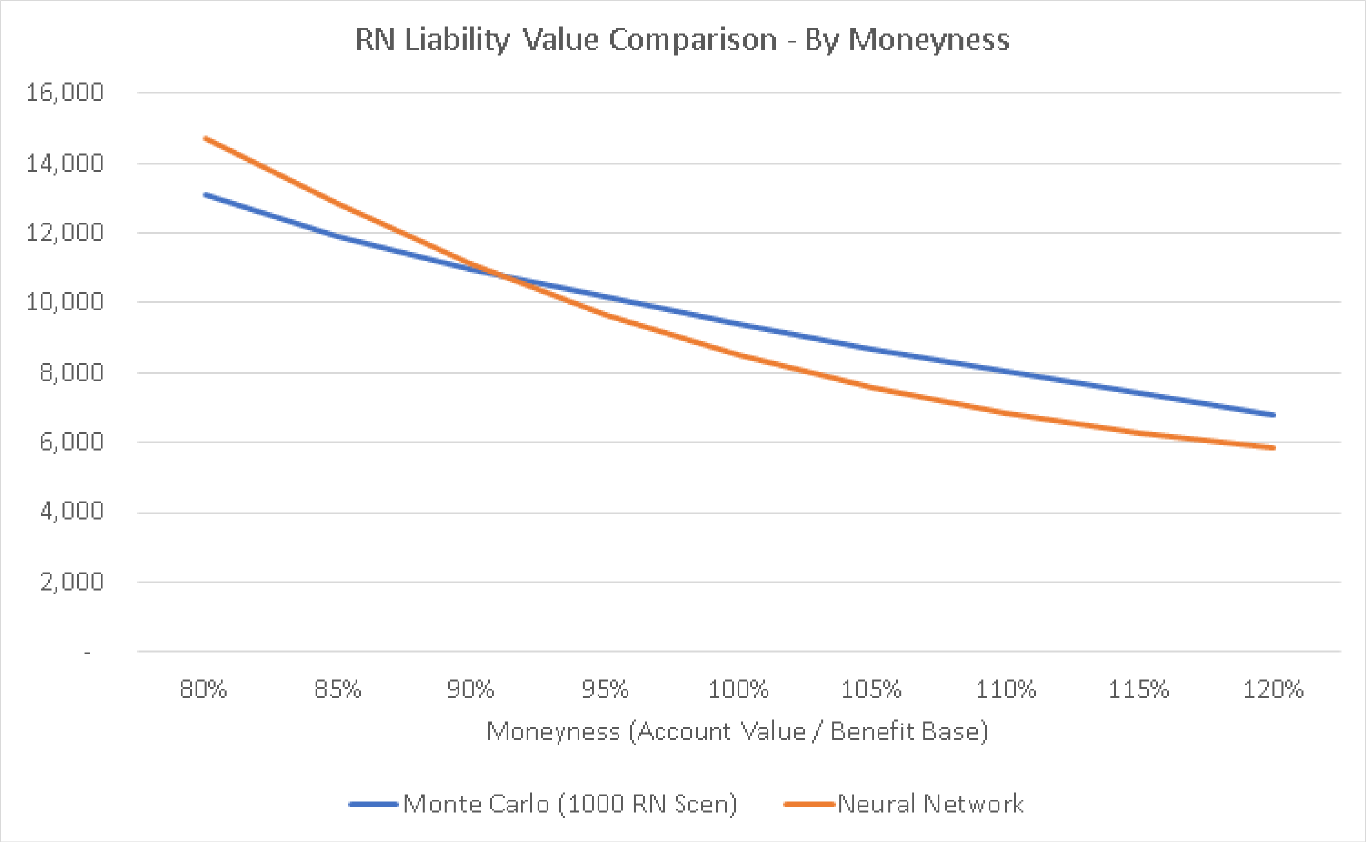
# Seriatim RN LOV

Neural Network vs. Monte Carlo



# Seriatim RN LOV

## Vary by Moneyness



# Resources

- https://towardsdatascience.com/

20
21

20
21