

# Predictive Analytics

## A Primer for Pension Actuaries



Aging and  
Retirement

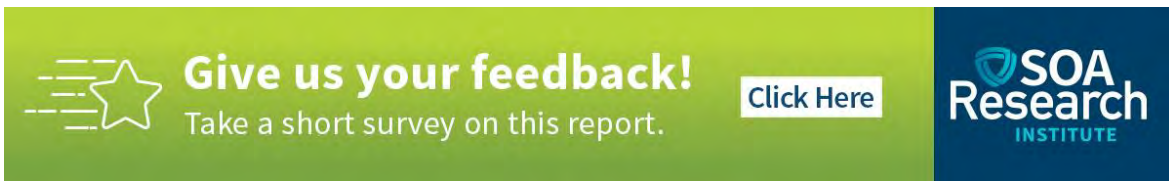


# Predictive Analytics

## A Primer for Pension Actuaries

**Authors** David R. Cantor, ASA, CFA, FRM  
Kailan Shang, FSA, CFA, PRM, SCJP

**Sponsor** Aging and Retirement Strategic  
Research Program Steering  
Committee

A horizontal banner with a green-to-blue gradient background. On the left is a white star icon with horizontal lines extending from its left side. To the right of the star is the text "Give us your feedback!" in a bold, white, sans-serif font, followed by "Take a short survey on this report." in a smaller, white, sans-serif font. Further right is a white rectangular button with the text "Click Here" in a blue, sans-serif font. On the far right of the banner is the SOA Research Institute logo, which consists of a blue shield icon and the text "SOA Research INSTITUTE" in white, sans-serif font.

**Caveat and Disclaimer**

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries Research Institute, Society of Actuaries, or its members. The Society of Actuaries Research Institute makes no representation or warranty to the accuracy of the information.

Copyright © 2021 by the Society of Actuaries Research Institute. All rights reserved.

## CONTENTS

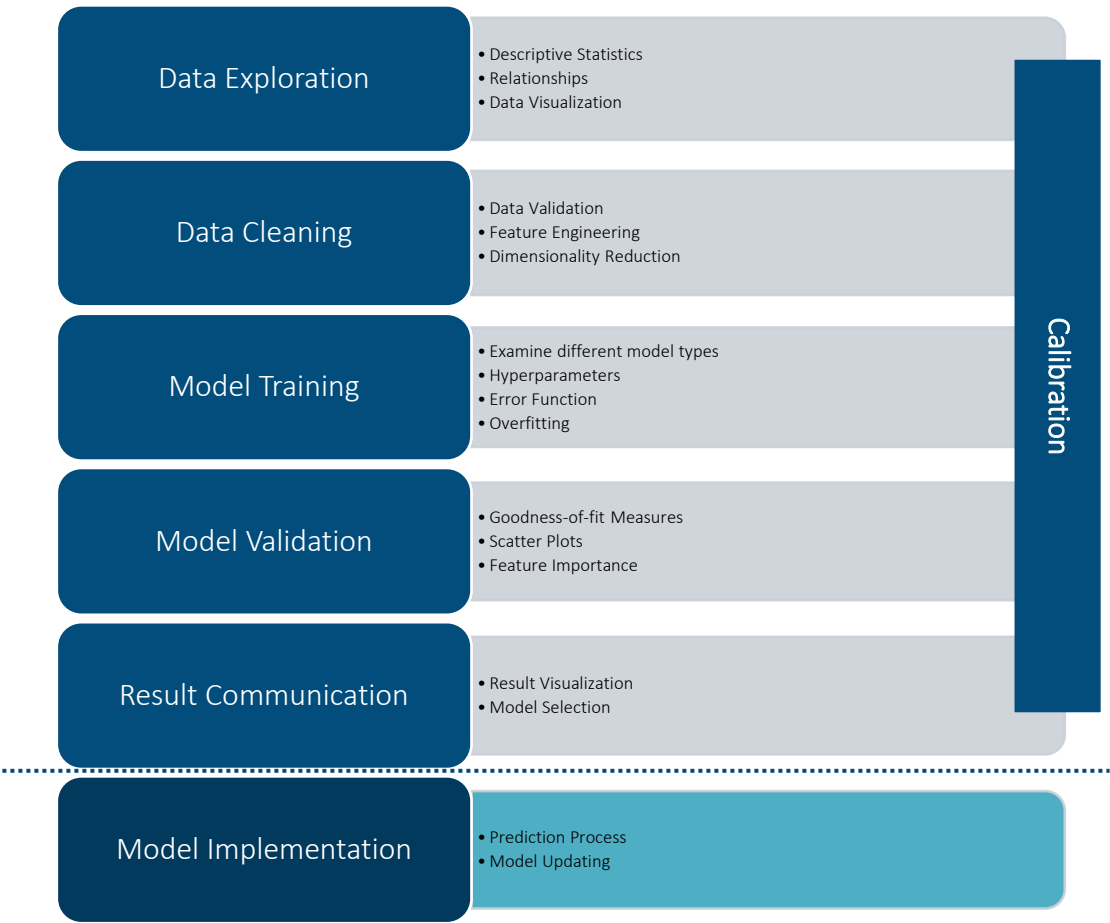
<b>Executive Summary .....</b>	<b>4</b>
<b>Section 1: Introduction .....</b>	<b>6</b>
<b>Section 2: Predictive Modeling with an Example .....</b>	<b>8</b>
2.1 Exploratory Data Analysis .....	10
2.2 Data Cleaning .....	15
2.3 Predictive Model.....	15
2.3.1 Supervised Learning .....	16
2.3.2 Unsupervised Learning.....	19
2.4 Model Training and Validation .....	20
2.5 Result Communication .....	23
2.6 Model Implementation.....	24
<b>Section 3: Literature Review .....</b>	<b>27</b>
<b>Section 4: Case Study: De-risking Activity Prediction .....</b>	<b>31</b>
4.1 Data Preparation .....	32
4.2 Predictive Model .....	35
4.3 Model Training, Validation and Selection .....	38
<b>Section 5: Conclusion .....</b>	<b>41</b>
<b>Section 6: Acknowledgments.....</b>	<b>43</b>
<b>References .....</b>	<b>44</b>
<b>Appendix A: Predictive Modeling in Condensed Form .....</b>	<b>47</b>
A.1 Exploratory Data Analysis .....	48
A.2 Data Cleaning .....	54
A.2.1 Missing Data Treatment.....	55
A.2.2 Data Normalization .....	55
A.2.3 Feature Engineering.....	56
A.2.4 Dimensionality Reduction .....	57
A.3 Predictive Model.....	60
A.3.1 Supervised Learning .....	60
A.3.2 Unsupervised Learning .....	68
A.3.3 Reinforcement Learning .....	72
A.4 Model Training.....	73
A.4.1 Error Function.....	73
A.4.2 Overfitting.....	75
A.4.3 Optimization Algorithm .....	78
A.4.4 Hyperparameters .....	80
A.5 Model Validation.....	81
A.5.1 Regression Model Validation.....	82
A.5.2 Classification Model Validation .....	83
A.5.3 Feature Importance .....	85
A.5.4 Unsupervised Learning Model Validation.....	88
A.5.5 Reinforcement Learning Model Validation.....	88
A.6 Result Communication .....	89
A.7 Model Implementation.....	90
<b>Appendix B: Open-Source Python Program .....</b>	<b>92</b>
<b>About The Society of Actuaries Research Institute.....</b>	<b>93</b>

## Executive Summary

The growing availability of data has changed the landscape of analytics on data processing, predictive models, and granularity of analysis. Changes are happening in the pension and retirement field to utilize the data and predictive models for better analysis and decision-making.

This report introduces predictive analysis to pension actuaries in a concise and practical way. We cover the three types of models used in predictive analytics: supervised learning, unsupervised learning, and reinforcement learning. By using a simple case study on relative mortality prediction at the U.S. county level using demographic and economic information, we introduce the standard predictive modeling process, as shown in Figure E.1. With illustrations and explanations of all the connected components in the process, actuaries get a high-level picture of how a real-world application of predictive analysis can be applied by those in the pension/retirement domain.

**Figure E.1**  
PREDICTIVE MODELING SAMPLE PROCESS



To explore the existing applications to the pension/retirement field, a thorough review of existing applications is conducted, with a focus on mortality modeling, pension plan risk transfer, liability driven investment and asset allocation, and retirement decision-making and defined contribution plans. At the same time, areas that predictive modeling may be applied to improve the pension industry are identified.

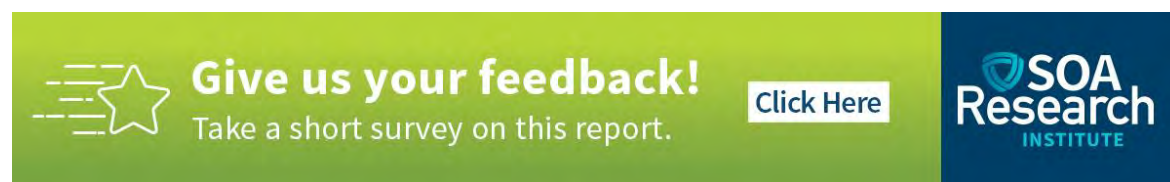
To further demonstrate the potential application of predictive analysis, another but more complicated case study is used. Here we use predictive analytics to predict de-risking activity. We use 11 years of IRS Form 5500 data to



predict whether an individual single-employer plan will have de-risking activities in the next year given the current plan information reported in Form 5500 and its schedules. This prediction task can be formulated as a classification problem and solved. This provides an additional useful case study to complement the regression type problem used in the introductory example on relative mortality prediction.

Through these new and carefully designed examples, we keep the focus on important concepts as opposed to technical details. At the same time, these relevant examples can be used as a foundation, and hopefully inspiration, for other applications in the pension/retirement field.

More methods and models are discussed in the Appendix to reinforce and expand what is covered in the case studies and core of the report. Python codes used for the case studies are also made available for educational purpose and hosted at [GitHub - Society-of-actuaries-research-institute/AR135-Predictive-Analytics-for-Retirement](https://github.com/Society-of-actuaries-research-institute/AR135-Predictive-Analytics-for-Retirement).

To our knowledge, our paper is the first to contribute directly to the applications of predictive modeling to pension and retirement problems.

A horizontal banner with a green-to-blue gradient background. On the left, there is a white star icon with motion lines. To its right, the text "Give us your feedback!" is written in a bold, white font, followed by "Take a short survey on this report." in a smaller white font. Further right, a white rectangular button contains the text "Click Here" in blue. On the far right, the SOA Research Institute logo is displayed, featuring a blue shield icon and the text "SOA Research INSTITUTE" in white and blue.

 **Give us your feedback!**  
Take a short survey on this report. [Click Here](#) 

## Section 1: Introduction

Like many other industries, the pension industry is experiencing changes brought about by predictive analytics, the availability of more data, and advanced technologies. Social insurance programs, employer sponsored pension plans, and individual retirement planning are adapting to these new developments. In general, predictive analytics can help understand and predict demographic changes, financial behavior and facilitate better retirement decision-making.

Predictive analytics is statistical analysis aimed at making predictions about future or unobservable outcomes using data and techniques such as statistical modeling and machine learning. Actuaries have been working with predictive models such as linear regression and generalized linear models (GLMs) for a long time. However, with the advancement of better computing technologies, a few things have changed in the past few decades.

- Algorithms used for traditional statistical models changed with much larger data volume available than before. For example, in linear regression  $Y = X\beta + \varepsilon^1$ , the ordinary least squares method will estimate  $\beta$  as  $(X'X)^{-1}X'Y^2$ . However, when the dataset is large, calculating  $(X'X)^{-1}$  becomes challenging and it is more likely to encounter singularity or near singularity issues where the inverse matrix cannot be calculated. What this means is the solution cannot be found. Other methods are widely used instead, such as the gradient descent method which is an iterative optimization algorithm that gradually adjusts  $\beta$  to minimize the prediction error  $\varepsilon$ .
- Many models that were not practical in the past have become popular with the availability of increasing computing capabilities. For example, artificial neural networks (ANNs) were first developed by Rosenblatt (1958) to model information storage and organization in the brain but became popular only about two decades ago given increased computing power.
- In traditional statistical models, emphasis has been put on hypothesis tests such as the t-test and the F-test to evaluate model accuracy. With more data available, model validation has become more data driven where the entire dataset is usually split into training data and validation data. Validation data is not observable during model training but used to assess the accuracy of prediction. Hypothesis tests are less used, partially because some models have formats that are too complicated with too many parameters, and partially because data driven model validation is enough to assess model accuracy. The focus of predictive analytics tends to be on making accurate predictions whereas classic statistics tends to focus more on building models with intuition and clear explanatory variables.

Along with technological developments and increasing data availability, new buzzwords such as big data analytics, machine learning, deep learning and artificial intelligence (AI) have appeared. Although these buzzwords have their specific focus on data volume, model types, or applications, they have overlaps with predictive analytics. From a modeling perspective, they share the methods and overarching approach of data processing, examining different potential model types, training the potential models and then performing validation.

This report will start by introducing the typical predictive modeling process including data processing, model choice, model training, model validation, and results interpretation. Readers will be able to understand the kind of problems that predictive modeling can help solve, different tools and models that are available to perform the analysis, and ways to assess model accuracy and address important issues such as overfitting.

---

<sup>1</sup>  $Y$  is a vector that contains all the data records of the response variable.  $X$  is a matrix containing ones and the values of the explanatory variables.  $\beta$  is the parameters of the linear function and  $\varepsilon$  is the residual errors that cannot be explained by the linear function.

<sup>2</sup>  $X'$  is the transpose of matrix  $X$ .  $(X'X)^{-1}$  is the inverse of matrix  $X'X$ . These are standard linear regression formulas written in matrix form as found in elementary statistics textbooks.

We proceed as follows:

- Section 2 (Predictive Modeling with an Example) introduces predictive modeling with a mortality prediction example. Using U.S. county level data including mortality, demographic and economic information, this section shows a subset of tools and models used in predictive analytics but provides a comprehensive overview of a typical predictive modeling process including data processing, model choice, model training, model validation, result communication and model implementation. Readers will be able to understand the kind of problems that predictive modeling can help solve, different tools and models that are available to perform the analysis, and ways to assess model accuracy and address important issues such as overfitting.
- Section 3 (Literature Review) discusses the areas that predictive modeling may be applied to improve the pension industry. It includes both existing and potentially future applications and research. This helps readers better understand the potential impact of predictive analytics on pension and retirement in the future.
- Section 4 (Case Study: De-risking Activity Prediction) uses Form 5500 data published by the U.S. Department of Labor and studies the demographic profile, fund contribution, asset allocation, funding status, demographic profile, and actuarial assumptions using plan level data to identify useful data, trend, and patterns for pension plan management. It builds a prediction system to identify plans that may have risk transfer activities in the near future. Supervised learning is applied with a detailed explanation of data preparation, model training, model validation, and model selection.
- Section 5 (Conclusion) summarizes the key points of this research and concludes the main body of the report.
- Appendix A (Predictive Modeling in Condensed Form) provides a comprehensive overview of predictive modeling. We encourage interested and advanced readers to review the Appendix. We have deliberately designed Appendix A to mirror the body of the report but with more details and examples. Therefore, topics can be read at a high-level in the body of the report and further explored in the parallel area of the Appendix.
- Appendix B (Open-Source Python Program) describes the Python programs built for this research that are publicly accessible.

## Section 2: Predictive Modeling with an Example

Using a simple yet relevant topic as an example, this section introduces some of the basic elements of a predictive analytical task including data, model selection, and prediction. Discussions of modeling choices are avoided in this section but covered in [Appendix A](#), to make this section as light as possible. The idea here is not to get obsessed with all the details but to understand the process and core concepts so that other problems can be approached in the same mechanical and systematic way.

Geolocation has been widely used in insurance pricing for a long time, such as life products that protect against death events, and non-life insurance products like auto insurance. For the pension industry, geolocation of plan participants can help evaluate the aggregate mortality rate of a pension plan. It may also be beneficial for individual retirement planning, recognizing that other factors such as age and health conditions may have a bigger impact on mortality. In this example, we try to predict the U.S. county level mortality rate compared to the national average mortality rate.

Two datasets are used in the relative mortality example:

- United States Mortality Rates by County 1980-2014 by Global Health Data Exchange<sup>3</sup>. It provides mortality rates of 3,142 counties due to 21 mutually exclusive causes of death.
- 2016 Planning Database by U.S. Census Bureau<sup>4</sup>. It provides information about urbanization, gender, age, race distribution, education level, health insurance, and household incomes for each county.

Table 1 lists the final variables used in this example. The *response variable* “MR\_Relative” which we want to predict is defined as the county level mortality rate divided by the national average mortality rate. The *explanatory variables* that are used to explain the response variable contain some high-level demographic and economic information. Some of the explanatory variables are created based on the original dataset to better represent the information we want to use for this analysis.

---

<sup>3</sup> Institute for Health Metrics and Evaluation (IHME). United States Mortality Rates by County 1980-2014. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2016.  
<http://ghdx.healthdata.org/record/ihme-data/united-states-mortality-rates-county-1980-2014>

<sup>4</sup> The U.S. Census Bureau, 2017. “2016 Planning Database”.  
<https://www.census.gov/data/datasets/2016/adrm/research/2016-planning-database.html>



**Table 1**  
**RELATIVE MORTALITY DATASET VARIABLES**

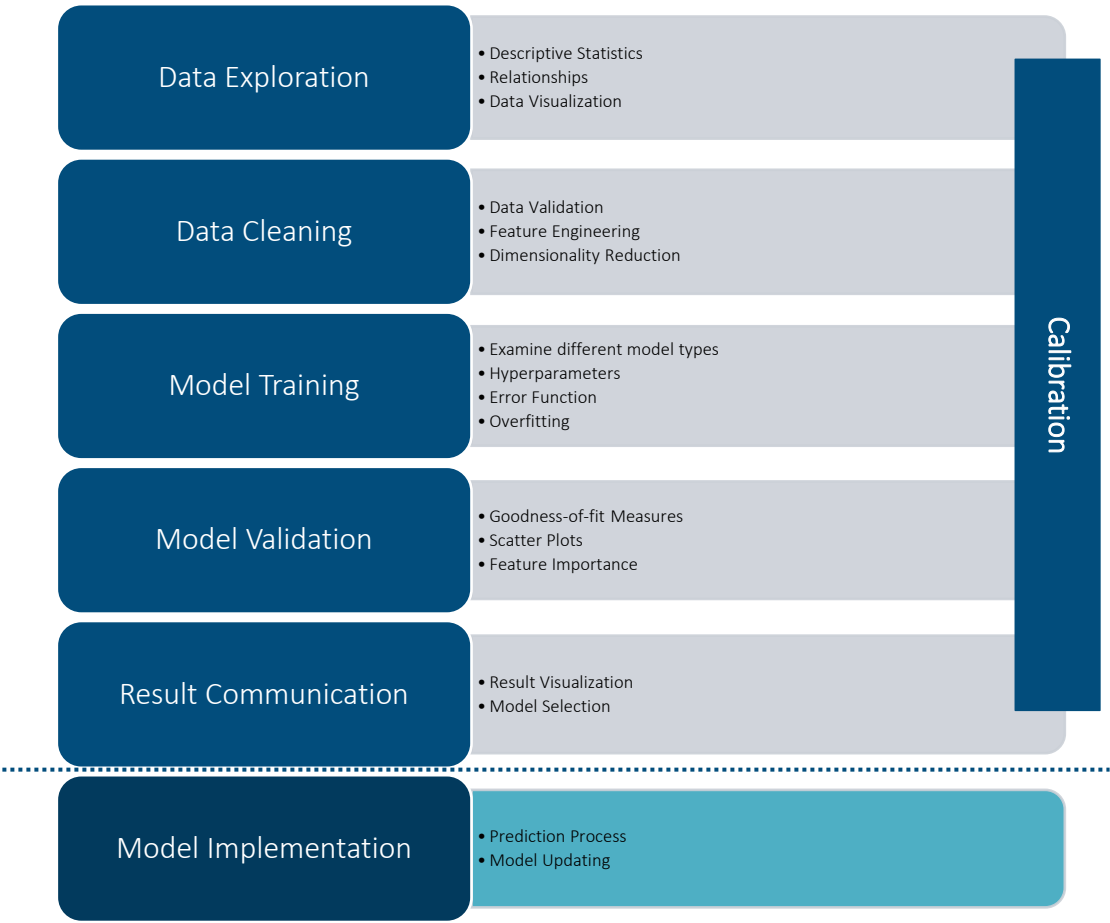
Variable	Note	Type
<b>MR_Relative</b>	<b>Relative mortality multiple</b>	<b>Response</b>
land_per_capita	Land area (sq.mi.) per person	Explanatory
urbanized_pop_pct	Perc. of population living in Area defined as an Urbanized Area (50,000 or greater)	Explanatory
urban_cluster_pct	Perc. of population living in Area defined as an Urban Cluster Area (2,500-49,999)	Explanatory
rural_pop_pct	Perc. of population living in Area outside of an Urban Area or Urban Cluster	Explanatory
male_pct	Perc. of male residents	Explanatory
female_pct	Perc. of female residents	Explanatory
age_5_pct	Perc. of age below 5	Explanatory
age_5_17_pct	Perc. of age 5-17	Explanatory
age_18_24_pct	Perc. of age 18-24	Explanatory
age_25_44_pct	Perc. of age 25-44	Explanatory
age_45_64_pct	Perc. of age 45-64	Explanatory
age_65_pct	Perc. of age 65 and over	Explanatory
hispanic_pct	Perc. of Hispanic origin population	Explanatory
white_pct	Perc. of White population	Explanatory
black_pct	Perc. of Black and Africa American population	Explanatory
aian_pct	Perc. of American Indian and Alaska Native population	Explanatory
asian_pct	Perc. of Asian population	Explanatory
nhopi_pct	Perc. of Native Hawaiian and Other Pacific Islander population	Explanatory
sor_pct	Perc. of some other race population	Explanatory
not_hs_pct	Perc. of people 25 years old and over who are not high school graduates	Explanatory
college_pct	Perc. of persons 25+ with Bachelor's degree or higher	Explanatory
poverty_pct	Perc. of people classified as below the poverty level	Explanatory
one_health_ins_pct	Perc. of people with one type of health insurance coverage	Explanatory
two_plus_health_ins_pct	Perc. of people with two or more types of health insurance coverage	Explanatory
no_health_ins_pct	Perc. of people with no health insurance coverage	Explanatory
med_hh_inc	Median household income	Explanatory
avg_hh_inc	Average household income	Explanatory
med_house_value	Median house value	Explanatory
avg_house_value	Average house value	Explanatory
state_avg	Average relative mortality at state level	Explanatory

One question to answer before performing the analysis is why do we need to predict the relative mortality multiple

in the first place? If the county level relative mortality multiple is already known, it can simply be applied to adjust the mortality assumption at the county level. For applications in the pension field, the relationship found at the county level can be applied at a more granular level such as to zip codes. For instance, say we know the relative mortality for Westchester County in New York. There are numerous zip codes within Westchester County. Mortality data is not publicly available at the zip code level. We can, however, collect explanatory variables at the zip code level and then map those to a representative county or set of counties. If we know the relative mortality of the count(ies) we can then use this as a proxy for the relative mortality at the more detailed zip code level.

Figure 1 shows a typical predictive modeling process, which is composed of two major parts: calibration and implementation.

**Figure 2**  
**PREDICTIVE MODELING SAMPLE PROCESS**



The rest of this section explains each component in the process using the example of relative mortality prediction.

**2.1 EXPLORATORY DATA ANALYSIS**

As the first step in predictive modeling, exploratory data analysis (EDA) is a model-free approach to summarize data and relationships among variables using descriptive statistics and visualization. The goal of EDA is to provide an overview of the data and spot any interesting trends or relationships that may be helpful for constructing predictive

models and validating the results. Pension actuaries already employ a version of EDA in their standard valuation actuarial reports when they plot metrics such as trends in funded status and population statistics.

Descriptive statistics, correlation, and data visualization are typical tools used during EDA to find prominent and simple patterns. Table 2 shows some descriptive statistics of the variables used as explained in Table 1.

**Table 2**  
**SAMPLE DESCRIPTIVE STATISTICS**

Variable	Mean	Standard Deviation	Min	1st Quartile	Median	3rd Quartile	Max
MR_Relative	1.10	0.18	0.39	0.98	1.09	1.22	2.09
land_per_capita	0.10	0.68	0.00	0.01	0.02	0.04	26.04
urbanized_pop_pct	30%	35%	0%	0%	6%	61%	100%
urban_cluster_pct	21%	21%	0%	4%	16%	34%	100%
rural_pop_pct	49%	29%	0%	26%	46%	69%	100%
male_pct	50%	2%	45%	49%	49%	50%	72%
female_pct	50%	2%	28%	50%	51%	51%	55%
age_5_pct	6%	1%	3%	6%	6%	7%	13%
age_5_17_pct	17%	2%	7%	16%	17%	18%	27%
age_18_24_pct	9%	3%	2%	8%	9%	10%	47%
age_25_44_pct	24%	3%	14%	23%	24%	26%	43%
age_45_64_pct	28%	3%	10%	26%	28%	29%	45%
age_65_pct	15%	4%	3%	13%	15%	17%	33%
hispanic_pct	9%	12%	0%	3%	5%	10%	96%
white_pct	76%	17%	3%	68%	80%	89%	99%
black_pct	9%	12%	0%	1%	5%	14%	78%
aian_pct	2%	6%	0%	0%	0%	1%	83%
asian_pct	2%	3%	0%	0%	1%	2%	43%
nhopi_pct	0%	0%	0%	0%	0%	0%	11%
sor_pct	0%	0%	0%	0%	0%	0%	3%
not_hs_pct	10%	4%	2%	7%	9%	12%	30%
college_pct	15%	6%	2%	11%	14%	18%	54%
poverty_pct	16%	5%	1%	13%	15%	18%	44%
one_health_ins_pct	66%	6%	36%	63%	67%	70%	84%
two_plus_health_ins_pct	18%	3%	3%	16%	17%	19%	38%
no_health_ins_pct	14%	5%	2%	11%	14%	16%	61%
med_hh_inc	49,598	12,000	21,756	42,163	47,786	54,564	132,203
avg_hh_inc	63,259	13,971	30,086	54,294	60,703	69,175	152,424
med_house_value	144,333	77,800	30,200	98,559	125,495	163,340	891,664
avg_house_value	120,483	54,312	0	90,227	107,708	134,596	850,772
state_avg	1.06	0.11	0.82	0.98	1.03	1.11	1.30

Descriptive statistics can also be represented as graphs to provide an overview of the data in a vivid way. Figure 2 shows the degree of urbanization, age mix and health insurance coverage.

**Figure 2**  
SAMPLE PIE CHARTS

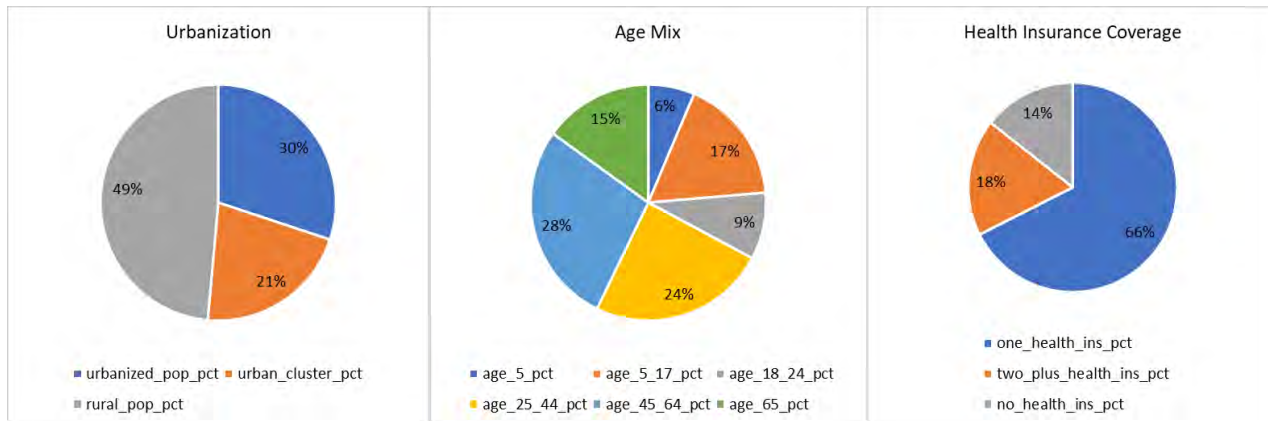
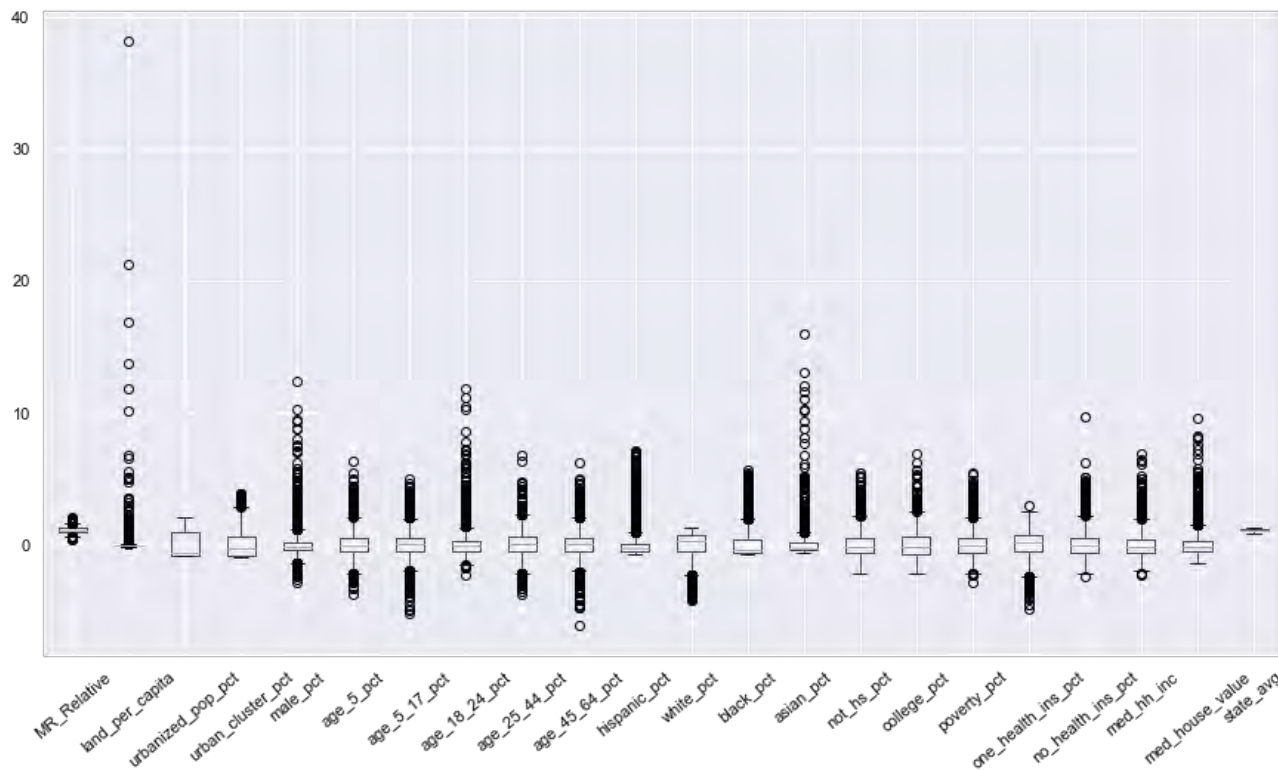


Figure 3 are the box plots that show the distributions of variables. Each box represents the data between the first and third quartile (Q1 and Q3), with the line in the middle as the median. The box extends to wider data range between  $[Q1 - 1.5(Q3 - Q1), Q3 + 1.5(Q3 - Q1)]$ , where available. For outliers outside this range, they are plotted as individual dots. It is clear that variables such as “land\_per\_capita”, “male\_pct”, “age\_18\_24\_pct” and “med\_house\_value” are all right skewed.

**Figure 3**  
SAMPLE BOX PLOTS

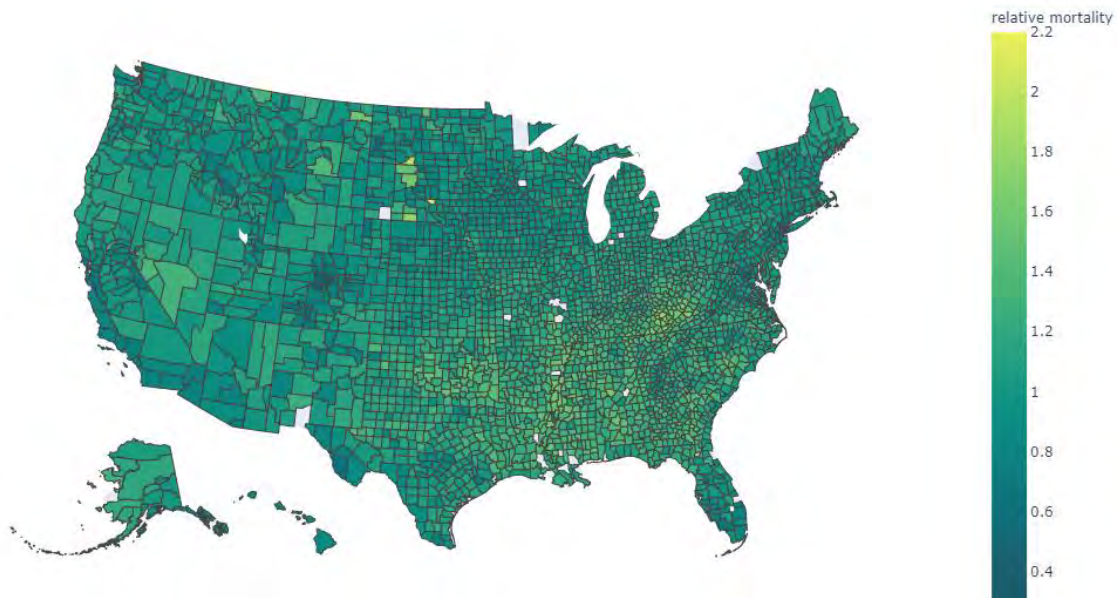


Descriptive statistics are helpful for gaining a high-level understanding of the data. However, data can behave significantly differently with the same or similar descriptive statistics. Visualization is a powerful tool to further study the details that may be missed in descriptive statistics. Figure 4 illustrates the relative mortality data on a map. It

can help us set some benchmark about areas where mortality rates are relatively higher compared to national average.

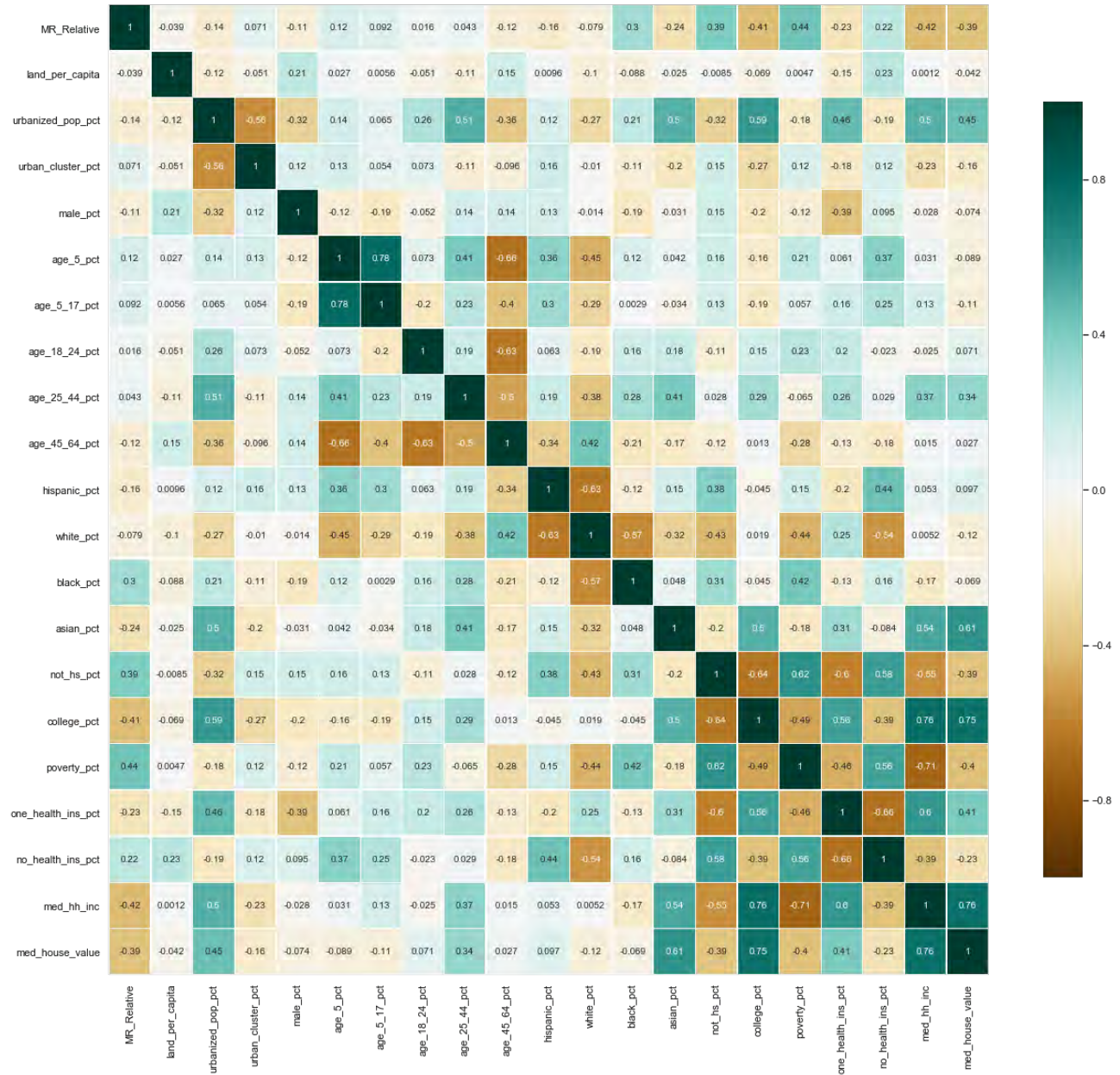
**Figure 4**

**U.S. MORTALITY LEVEL BY COUNTY**



To explore the relationship among variables, correlation matrices in the format of a heatmap can be used to quickly identify highly correlated pairs. Figure 5 shows the heatmap of some selected variables in the dataset. It is notable that the response variable is negatively correlated with household income (“med\_hh\_inc”), house value (“med\_house\_value”), college education (“college\_pct”) and positively correlated to poverty (“poverty\_pct”). Some explanatory variables such as household income (“med\_hh\_inc”) and house value (“med\_house\_value”) are highly correlated.

Figure 5  
SAMPLE HEATMAP



EDA is a general concept that contains all kinds of data exploration without formal modeling. What is described above is only a small portion of what is available in this field<sup>5</sup>.

<sup>5</sup> Two recent books on the subject include, “How Charts Lie: Getting Smarter about Visual Information” and “The Truthful Art: Data, Charts, and Maps for Communication”. Both are by Alberto Cairo and are recommended for readers who are more interested in this area.

## 2.2 DATA CLEANING

After the EDA, it is easier to define the prediction task and determine what data may be used. However, before the data can be fed into the predictive models, additional processing is often needed to adjust the data inputs to potentially improve model accuracy. Missing data treatment, data normalization, feature engineering, and dimensionality reduction are often used in the data cleaning process. In this simple example, part of the first three components is used.

Missing data is quite common, especially with large datasets. In the census dataset where demographic and economic information is stored, 33 out of 3,142 counties do not have household income and house value data, which are correlated with relative mortality level. Normally a few choices are available to treat data records with missing data. They can be removed, replaced with average value or value in a similar data record, or flagged in an indicator variable to represent that the data is missing. In this example, all 33 records are removed considering that the volume of the remaining data is still large enough for a meaningful analysis.

When explanatory variables have different levels of magnitude, they may need to be normalized so that the parameter calibration will not be dominated by a small portion of the variables, and therefore better reflect the relationship between response variable and explanatory variables. As shown in Table 2, certain variables such as “med\_hh\_inc” and “med\_house\_value” are much larger than most of the other variables. Therefore, it is necessary to perform data normalization to bring all explanatory variables into a similar data range. Standardized Scaling is used in this example. It is a commonly used method and a reasonable choice for cases with and without outliers, both of which exist in the dataset.

$$X' = \frac{X - \mu_x}{\sigma_x}$$

Where

$\mu_x$ : mean of X variable

$\sigma_x$ : standard deviation of X variable

In many situations, new explanatory variables are created based on existing explanatory variables and used for predictive analytics. One type of feature engineering is to transform categorical variables to dummy variables. For categorical variables such as occupation, even though numerical values may be used to represent categories, they need to be converted to dummy variables based on distinctive categories. Another type of feature engineering is to create new variables to reflect nonlinear relationships. For example, new variables

$X_1^2$ ,  $X_1^3$ ,  $\log(X_1)$ ,  $X_1X_2$ , and  $X_1/X_2$  may be created based on variables  $X_1$  and  $X_2$ .

In this example, new features are created from a different perspective. The original dataset contains the number of people belonging to a certain category. Given that the population of each county is different, using the number of people directly is not helpful. Therefore, most variables are transformed to represent the percentage of people belonging to a category. For example, male residents are converted into the percentage of male in the county. This is an important change to make the predictive models useful for future applications. The aggregated mortality rate of a pension plan is likely to be better explained by gender mix, rather than the number of male and female participants.

## 2.3 PREDICTIVE MODEL

Three types of models are used in predictive analytics: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is used to learn the relationships between the response variable and explanatory variables. Linear regression is a supervised learning model.

Unsupervised learning is to learn the patterns and relationships among explanatory variables, without any knowledge of the response variables.

Reinforcement learning is related to dynamic decision-making. It requires trial and error to actively learn from experiments that generate training data.

Together with the EDA explained in [Section 2.1](#), different learning methods can be categorized based on two criteria: 1) whether the response variable is used and 2) whether data is used in a fixed way or an interactive way, as shown in Table 3.

**Table 3**  
**LEARNING METHOD CLASSIFICATION**

		Response variable	
		Known/Used	Unknown/Unused
Data Input	Fixed	Supervised Learning	Unsupervised Learning
	Interactive	Reinforcement Learning	Exploratory Data Analysis

The relative mortality example is a typical supervised learning problem. This is because we have a response variable and we are using the data inputs as prediction variables, although unsupervised learning can be used as well to estimate relative mortality level.

### 2.3.1 SUPERVISED LEARNING

Supervised learning can be applied to two different broad types of problems.

- 1) Regression analysis is used to predict the value of a response variable such as the fund surplus next year.
- 2) Classification analysis is used to predict the probability that a variable is true such as whether a pension fund will be underfunded next year.

The relative mortality example is a task of regression analysis. Linear regression, Classification and Regression Tree (CART), Random Forests, and Gradient Boosting Machine (GBM) can all be used to solve the problem.

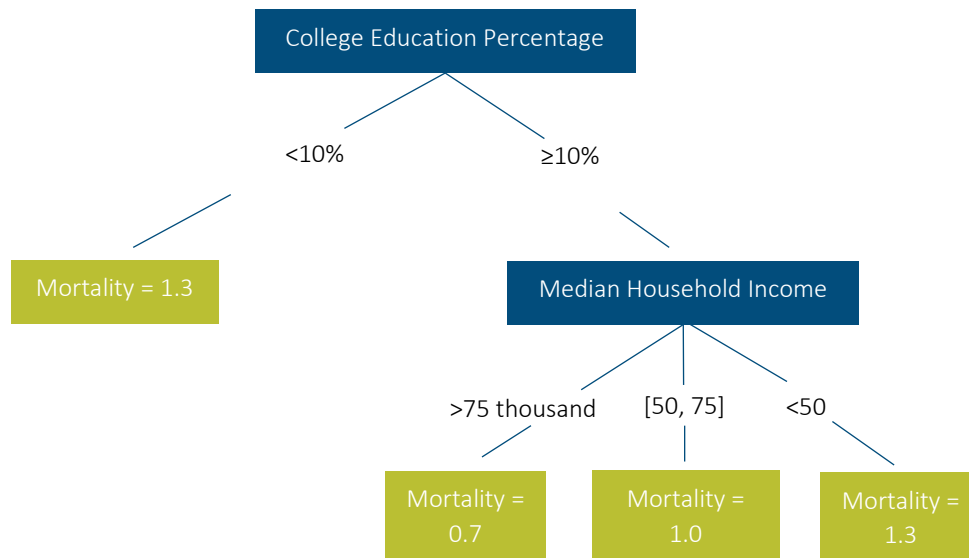
**Linear regression** is the simplest yet powerful parametric model. It assumes a linear relationship between explanatory variables and the response variable. Model parameters can be estimated by minimizing the squared errors. Here we would use the explanatory variables from Table 1 and use them to predict the response variable. Advanced forms of regressions are covered in the Appendix.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

**Classification and Regression Tree (CART)** takes a different approach than linear regression by switching from formulas to decision rules for prediction purposes. In a tree, leaves represent different subgroups and branches represent the rules to split into subgroups based on explanatory variables. The prediction is based on the value of the leaves that are in the same subgroup. Figure 6 shows an example using a tree-based model to determine the relative mortality level. The rules and conclusions in this example are straightforward and may not need any data to support.



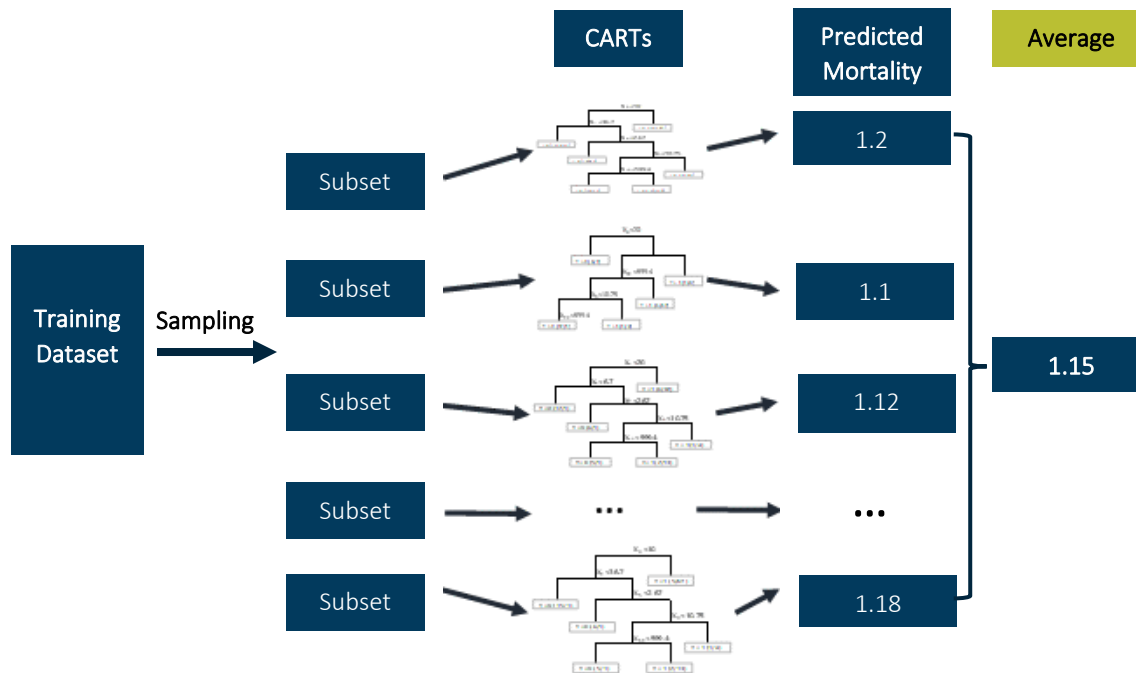
**Figure 6**  
SAMPLE TREE-BASED MODEL



CART models extend the concept above by building trees that split the data based on explanatory variables. At each split, a variable is used to separate the data into two subgroups. The variable is chosen to provide the best split that improves the purity of the data in the subgroups. A high similarity among data in a same subgroup means a high data purity.

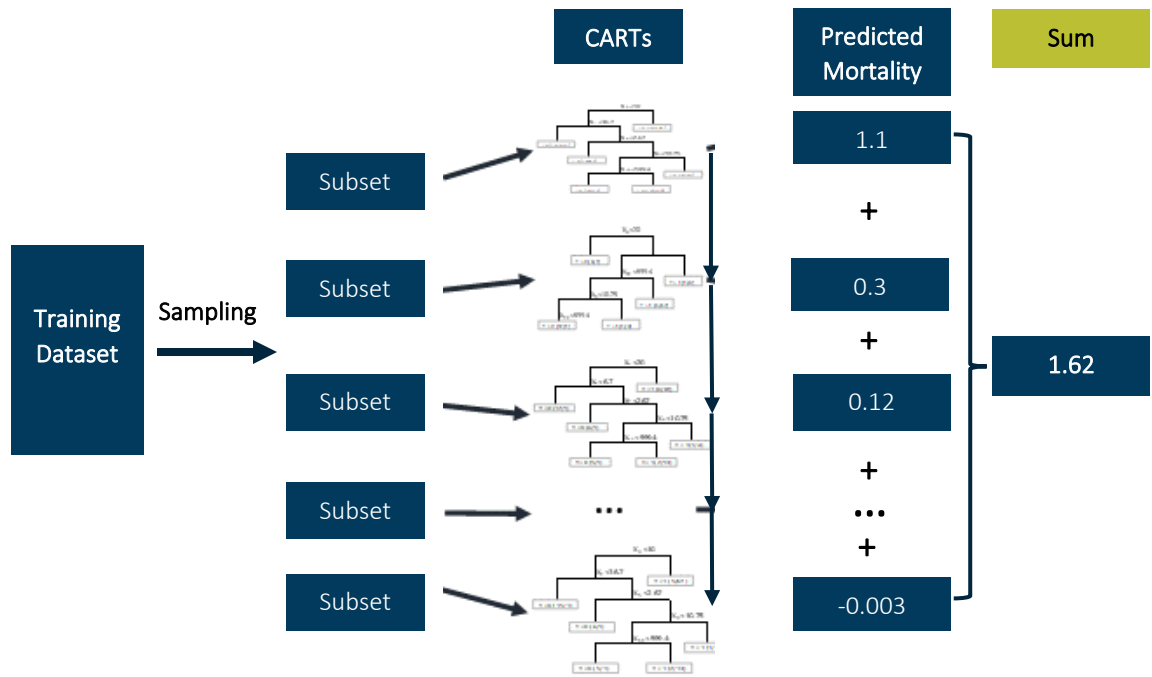
More advanced tree-based models are built upon CART. The famous **Random Forests models** are a random version of the CART models. Multiple subsets are sampled from the training dataset and each subset is used to build a CART model. Explanatory variables are sampled as well so that the relationship between the response variable and the explanatory variables will not be dominated by the most important ones. Less important explanatory variables can contribute to the final prediction as well. Figure 7 illustrates the structure of the Random Forests models used in this report. The final prediction is calculated as the average prediction by individual CART models.

Figure 7  
RANDOM FORESTS MODEL STRUCTURE



Gradient boosting machine (GBM) is another decision tree–based ensemble method. Each tree is a weak estimator trying to estimate the residual error that the estimation of previous trees has caused. Gradually with a sufficient number of decision trees, the estimation error will decline to a very low level. Unlike Random Forests models which use *parallel* trees to predict in aggregate (a concept known as “bagging”), GBM is a *sequential* tree model with the final prediction as the sum of predictions of all sequential trees, as shown in Figure 8.

**Figure 8**  
GBM MODEL STRUCTURE



In this illustration, we have a target Y variable with a value of 1.62. Using a standard GBM, we first fit a CART model using a subset of data and a subset of features (explanatory variables). This first CART model will give us a predicted value of 1.1. The remaining difference is 0.52, calculated as the difference between 1.62 and 1.1. We then fit another CART model to the difference of 0.52 and get an estimation of the 0.3. And this process keeps going until the difference is small enough, or there is no further improvement of the prediction.

### 2.3.2 UNSUPERVISED LEARNING

**Clustering** is one of the most popular model types in unsupervised learning. It categorizes data based on similarity. Similarity is usually measured by Euclidean distance with each dimension represented by an explanatory variable.

**Figure 9**  
K-MEANS EXAMPLE

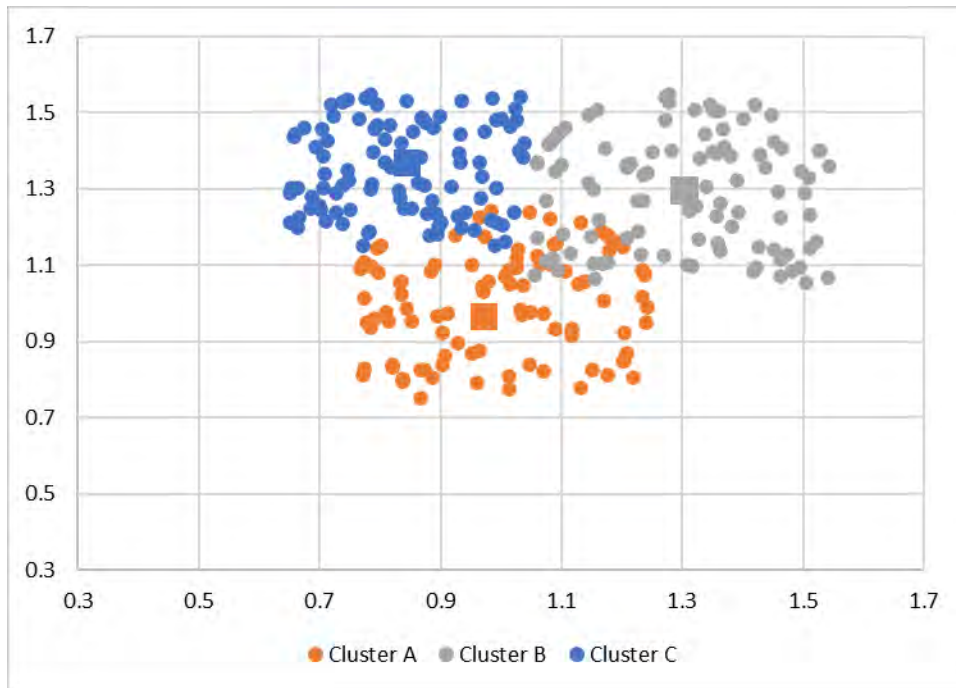


Figure 9 shows an example of k-means, a centroid clustering method. K-means partitions all data records into three groups based on similarity which is measured by Euclidean distance. The rectangles represent the centers of the three groups. A center in k-means is the average of all data points in that group. This can be used to classify all counties into different groups and calculate the average value of the response variable for each group. If the groups show different levels of relative mortality level, the model can be used to estimate the mortality level of a new county by assigning it to one of the groups based on similarity. The response variable is not included when creating the clusters which makes this an unsupervised learning exercise.

## 2.4 MODEL TRAINING AND VALIDATION

Model training is the process to calibrate model parameters based on the training data. Before model training, the clean dataset needs to be split into training data and validation data to facilitate model assessment. During model training, only training data, i.e., “in-the-sample data”, is observable by the model. Validation data, i.e., “out-of-sample data”, is then used to evaluate model performance. There are also advanced techniques like k-folds that extend this concept of training and validation. Sometimes, the split between training and validation data can be based on some key variables such as time. Similar to the idea of back testing, it is helpful to test whether the same relationships identified in early periods still hold in recent time. However, data splitting by variable(s) does not apply in this simple example.

An important issue to consider for model training is overfitting. When too many variables are unintentionally used to explain the random noises rather than the relationships, the model overfits the data and shows a very high accuracy of prediction with training data. However, a much lower prediction accuracy is usually observed using the validation data. Many methods such as regularization and using random data subsets and feature subsets are available to mitigate the risk of overfitting.

In our mortality prediction example, the error is defined as the difference between actual value  $y_{actual}$  and predicted value  $y_{pred}$  based on **Root-mean-squared error (RMSE)**: the square root of the mean of the square of all of the errors. Other definitions of errors can also be used as combinations of errors.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^N (y_{pred,i} - y_{actual,i})^2}{N}}$$

After the model training process to minimize the error function, calibrated models need to be assessed and compared using standard validation methods. It is important to know that validation data (out-of-sample data) needs to be used for a meaningful comparison so that the issue of overfitting can be identified.

To assess the goodness-of-fit of regression models, we can use an additional measure beyond just checking the RMSE metric that was part of the fitting procedure. A common alternative measure is coefficient of determination, also known as  $R^2$ .

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} = 1 - \frac{\sum_i (y_{pred,i} - y_{actual,i})^2}{\sum_i (y_{actual,i} - \bar{y}_{actual})^2}$$

This is applicable to not only linear regression but also other regression models. Table 4 shows the goodness-of-fit result of different models. K-means has the lowest performance as when classifying the data, it does not include the information of response variable. However, it does show some potential benefit of performing unsupervised learning even when the value of response variable is unknown. Tree based models have a better performance than the linear regression model as they can capture some nonlinear relationships. The level of  $R^2$  is not high but satisfactory because only demographic and economic data types are used in the prediction.

**Table 4**

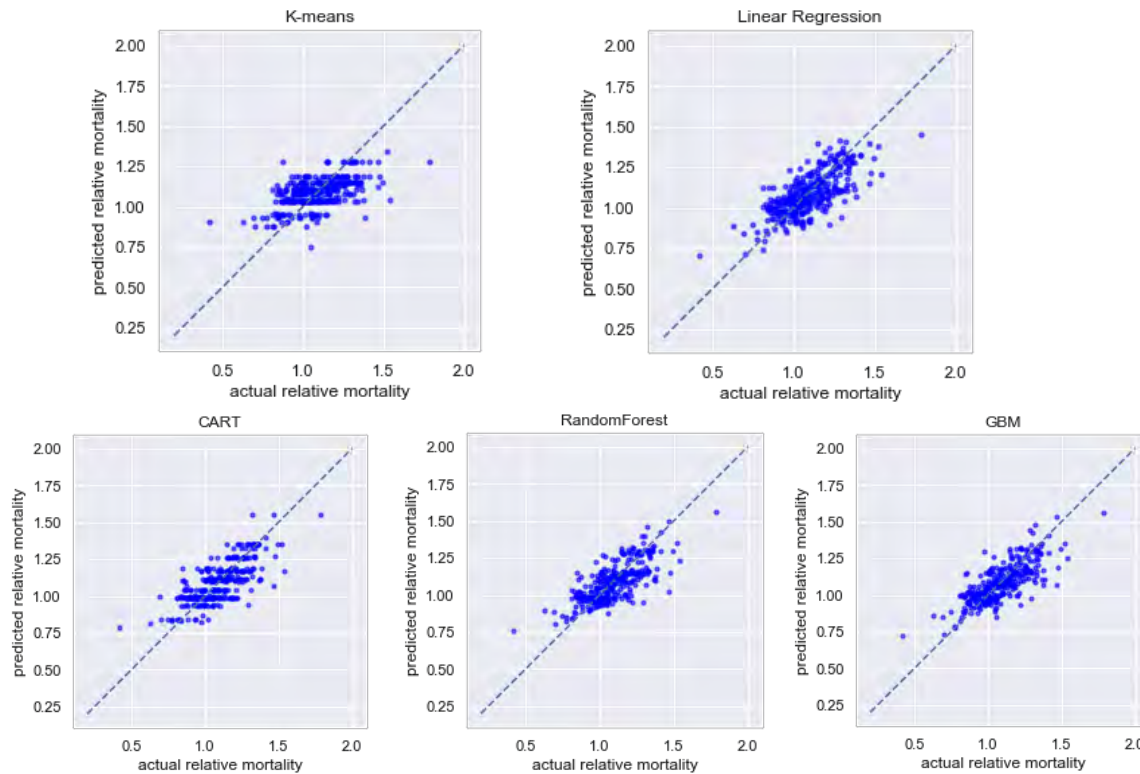
**GOODNESS-OF-FIT RESULTS**

Model	$R^2$	Model Specification*
K-means	30.3%	30 clusters
Linear Regression	51.9%	Standard linear regression with intercept
CART	54.4%	Maximum tree depth of 6
Random Forests	55.5%	500 trees
GBM	58.3%	1000 trees

Note: model training parameters were fine tuned to improve model accuracy.

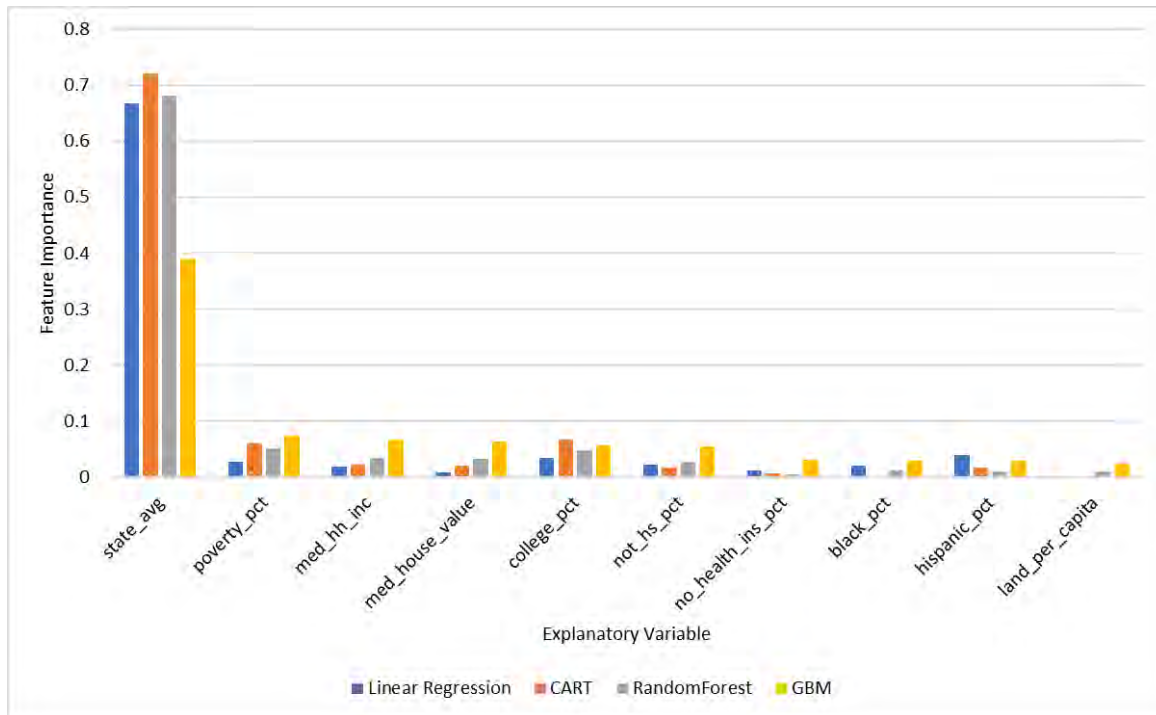
Models can be ranked based on goodness-of-fit measures at a high level. However, further analysis is usually desired to look at the actual predictions. Scatter plots of the actual values and predicted values are a good way to identify outliers and get comfortable with model accuracy. Figure 10 shows an example of a scatter plot to evaluate regression model accuracy. Dots lying on line  $y=x$  represent perfect estimation. Even if a model has a high  $R^2$ , scatter plots may help identify outliers which may be too important to ignore and may lead to a different model choice. As we described in the EDA section, plotting and visualizing the data is very useful. The scatter plots of the K-means and CART model have discrete predicted values. Both K-means and CART models use subgroup average as the estimate for any members in that subgroup. The GBM model shows the least volatility around line  $y=x$ , which is consistent with the highest  $R^2$  it has in this example.

**Figure 10**  
**SCATTER PLOT: REGRESSION MODEL VALIDATION**



In addition to scatter plots, feature importance analysis is also helpful for understanding what explanatory variables are driving the prediction. This is important from a sense-check and interpretability point of view. Figure 11 illustrates a typical feature importance analysis. Top 10 most important variables of the linear regression, CART, Random Forests, and GBM model are listed. For each of the models, the listed variables take more than 80% of the total weight in explaining the response variable. Although the state average relative mortality is important for all four models, GBM utilizes other variables in a better way to get higher model accuracy. The top explanatory variables are also consistent with what was indicated by the heatmap in [Section 2.1](#).

**Figure 11**  
FEATURE IMPORTANCE ILLUSTRATION



Performing feature importance analysis is beneficial in three ways.

- If some unexpected variables show in the list of important features, it helps identify potential issues with the model and data and requires further investigation before implementing the model.
- Important features can be used to set up key risk indicators and be frequently monitored for material changes.
- In the presence of overfitting, unimportant features may be removed.

## 2.5 RESULT COMMUNICATION

Predictive analytics contains many technical concepts that can be difficult to explain, especially with the growing number of models and their complexity. Material efforts are needed to be able to effectively communicate with the final decision-makers the results of predictive analytics. Based on our experience, a few recommendations are given below:

- As in any effective communication, knowing your audience is key. With an understanding of your audience's background, prior knowledge of and experience with predictive analytics, the communicator can carefully weigh on the content to be presented and the way they are presented.
- Relevance is important to attract interest. By linking the predictive analytics with something that the audience cares about, the chance of success will be higher. For example, when discussing a predictive model that estimates mortality, in addition to talking about model accuracy, a more relevant topic would be how the model can improve decision-making and the financial impact of switching to the model compared to maintaining the status quo. Actionable suggestions need to be embedded in predictive analytics result communication.
- No matter what the background of the audience is, it is often easier and more compelling to explain difficult concepts using graphs and/or tables. Result visualization is a powerful tool to deliver messages.

Also being able to model on the fly, in a dynamic way can really allow decision makers to gain intuition on what the models are doing and their impact on metrics of concern.

- Rather than communicating everything at one time, it is often better to present results gradually and sometimes repeatedly throughout a predictive analytics project. Even in one presentation, if a difficult technical detail is necessary to get the buy-in from the audience, it may be better to split the discussion into small pieces to explain. Short, bite-sized information sharing can be better than detailed long documentation.
- In addition to one-way communication, active involvement of stakeholders is very important to get them on board. Stakeholders should be encouraged to provide inputs to the process and be consulted about their interests, concerns, expectations and opinions. Domain knowledge and input from those with “boots on the ground” should be reflected in model building and then back in the communication of results.
- Actuaries should also consult with appropriate ASOPS. For example, ASOP 41 on actuarial communications and ASOP 56 on modeling are of particular relevance.
- To the extent end decision makers do not need to know the details of the model, communicating sufficiently in an Appendix or separate technical document is also recommended.

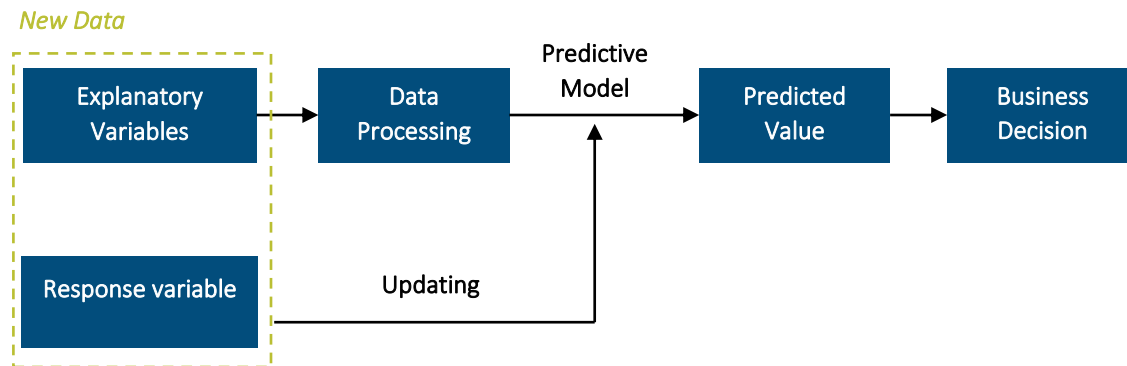
Once sufficient communication is made among stakeholders, a decision needs to be made regarding the best model to be used, if at all. Compared to existing decision-making rules, the financial impact of using the new model can also be quantified. By subtracting the cost of implementing the new model, the net impact can be used as a selection criterion. Costs of implementation include computing resources, database, program maintenance, training cost, and so on. Additional validation datasets can also be used to evaluate the impact of adopting the model. In addition to model accuracy and financial impact, model complexity and model risk are also important factors to consider. Given two models both of which have satisfactory prediction accuracy and financial impact, the model that is easier to understand, communicate and validate is likely to be chosen even though it has a lower accuracy. Simplicity also often times trumps complexity in designing a more robust, less potentially overfit model to deploy in real time. With the mortality example we have been using, all these considerations would need to be taken into account before deploying the model. Practitioners may choose their final model after weighing different criteria. A possible outcome from this analysis is that GBM will be adopted given its highest accuracy. Even though the model is complex, model accuracy will be a key concern for this specific case because the accuracy level measure by  $R^2$  is in the range between 50% and 60%. The CART model may be the second choice which is simpler than Random Forests and GBM to extract model parameters. However, its scatter plot in Figure 10 showed less randomness than other models which may not be desired for ranking counties’ mortality level.

## 2.6 MODEL IMPLEMENTATION

Once it is decided that a model will be used for real prediction task, a prediction and model updating process needs to be set up, as shown in Figure 12.



**Figure 12**  
SAMPLE PREDICTION PROCESS



When new data of explanatory variables arrives, it will be fed into the same data cleaning process used in model training. With the clean data, the selected predictive model can be used to estimate the value of the explained value. The estimated value can then be used to make business decisions. When the actual value of the response variable is available, new data records containing both explanatory and response variables can be added to the training dataset and used to update the predictive model when appropriate.

The updating process depends on many factors such as the volume of new data, the type of new changes, and the impact on decision-making.

- If the new data exhibits similar distributions and relationships to the existing data, model updating is not necessary. EDA can be used as preliminary analysis to evaluate whether a full-scale updating is needed. For example, if we selected the GMB model for our mortality prediction exercise, unless something structurally changed to the relationships, we would not feel the need to change the structure of the model. This, however, may not apply to this mortality case study as it is likely the frequency of data updating is low, and it is not a burden to recalibrate the model when new data is available. For other cases where data are more frequent such as capital market data, it is more realistic to have a recalibration frequency lower than the data frequency.
- A threshold of new data volume may be set to trigger the updating process. However, the determination of the threshold can be arbitrary. A rule of thumb is that the new data is more than 10% of existing data. The threshold may be determined using  $k$ -fold cross validation. By testing different values of  $k$ , the smallest  $k$  when an undesired difference is spotted among the  $k$  sets of training results can be found.  $1/k$  can be used as the threshold so that if the new data volume is  $1/k$  of existing data, an updating is needed. We further explain the  $k$ -folds method in the Appendix.
- When changes in a variable are spotted, if the volatility of that variable has been fully reflected in existing training dataset, an updating may not be necessary. This is because we already expect the variable to exhibit some amount of natural fluctuation. Alternatively, when the new data brings in values that are beyond what could be expected or there are outliers compared to existing values, an updating is needed. For example, with mortality prediction if a major change in the data (e.g., due to COVID-19) occurred, a model updating might be considered.
- The usage of the predictive modeling can also affect the updating cycle. If the model is used for pricing and the repricing follows a quarterly cycle, a quarterly model updating seems to be a reasonable choice. For the mortality example, an annual update might be considered given valuations are performed on an annual basis.

- The required efforts to update the model can also play a role. If automated processes can be set up for model updates, and computing resources are available, more frequent model updating can be implemented.

Model risk is also an important area to focus on during implementation. Although model risk can be vastly mitigated during model validation and model selection, efforts still need to be made to make sure the model has been applied correctly for prediction. If a complicated model such as a GBM model is used, it may be a good idea to use a simple model such as linear regression as a benchmark to make sure the predictions are not too far off. It is difficult to check a GBM model given the large number of parameters, but it is easy to check a linear model without programming.

## Section 3: Literature Review

In Section 3, we give pension practitioners insight into some relevant published research. To accomplish this aim, we conducted a literature review across sources noted below:

- American Academy of Actuaries
- Annals of Actuarial Science
- arXiv
- ASTIN Bulletin
- Boston Center for Retirement Research
- British Actuarial Journal
- Casualty Actuarial Society
- European Actuarial Journal
- Google search
- Insurance: Mathematics and Economics
- Journal of Economic Dynamics and Control
- Journal of Risk and Insurance
- Journal of Pension Economics and Finance
- North American Actuarial Journal
- Scandinavian Actuarial Journal
- Society of Actuaries
- SSRN

In addition to pulling information from the aforementioned sources, the Society of Actuaries Technology Section maintains a website of information of data analytics resources that pension practitioners may find useful <https://www.soa.org/sections/technology/technology-data-analytics-resources/>. Links are provided to open data sources, data integration, data visualization, advanced analytics, databases, programming tools, Excel add-ins, and free online open courses. Pension practitioners do not need to be afraid of using predictive models. With open-source platforms and many available resources<sup>6</sup>, it's fairly easy to interact with models to gain intuition and understanding with just a few lines of code.

A quote from “What data science means for the future of the actuarial profession” (British Actuarial Journal 2018) sums up the state of affairs quite well: “There is a general awareness and an appreciation of its (data science) importance although many actuaries may not have specific knowledge about the technicalities and the methodologies. Older actuaries tend to have less awareness of the approach and there tends to be a higher awareness amongst actuaries working in general insurance than in other areas and, *perhaps, lower awareness amongst pensions or life actuaries.*”

The literature review showed that the use of predictive analytics to actual retirement applications is fairly limited<sup>7</sup>. Practitioners need to be creative and inventive in applying the ideas from the literature to address their specific issues.

---

<sup>6</sup> Even twitter is a great place to learn about predictive analytics. There are numerous accounts that explain concepts in easy-to-read threads. A list of relevant twitter handles is available from the authors upon request.

<sup>7</sup> One other interesting example is the North American Actuarial Journal ran a special “Advances in Predictive Analytics” edition in 2020. Nine articles were shared – none in the pension domain.

We characterize prior research according to topics that are relevant to those operating in the pension domain. The literature on predictive analytics is vast. We have filtered the research to germane pieces.

Additional research is needed to identify, or create, predictive analytics textbooks specifically geared towards use by pension professionals. Future research may also be performed to compile and document the various actuarial packages already developed and used in common programming languages including but not limited to R and Python. The SOA syllabus for the Predictive Analytics examination is a very useful resource in this regard: <https://www.soa.org/Education/Exam-Req/edu-exam-pa-detail.aspx>. The textbooks and resources used on the syllabus are relevant for those practicing in the pension domain. We encourage practitioners to review these documents.

## General

The American Academy of Actuaries Big Data Task Force (2018) acknowledges that the use of predictive analytics in the pension area is limited but it is growing with the emergence of new roles for pension actuaries. The paper makes mention of two specific applications for pension practitioners. The first relates to mortality improvement assumptions for pension valuations whereby the assumptions are derived by extensive mortality data analysis, graduation to smooth out random noise, trend identification and pattern extrapolation. The second application relates to modeling embedded options in pension programs and potentially suboptimal choices made by plan participants. While the paper does not specifically address this, we draw the link between embedded options in pensions plans and certain options that exist for variable annuity policyholders. See Cantor (2011 and 2014) or Shang (2013) for additional details. The paper also notes applications related to assumption setting such as termination and retirement. By gathering data from broader company resources and/or the plan administrator (who may be able to share extensive industry data) more accurate predictions are a possibility. Plan design is another application that is referenced. Using advances in survey methodology, including data visualization and summarization, plan sponsors may be able to better assess the real needs of plan participants.

Chalk and McMurtie (2016) review many principles of predictive analytics that pension practitioners should be familiar with. These include the loss function, model evaluation measures, model validation, feature scaling, regularization and feature engineering. Their focus is on supervised learning only. Our paper is similar in many respects to this paper. Therefore, the two pieces can be read in tandem.

## Mortality modeling

Based on industry experience, most pension actuaries use the Society of Actuaries published mortality tables to perform their work. While the literature on applying predictive analytics to mortality tables is fairly rich, most of the papers are written for the insurance industry. Still, the concepts and techniques used in these papers could be ported to the pension domain, even if in a complementary manner to what is current practice.

Vincelli (2019), shows that by reverse-engineering the industry mortality table into a series of higher dimensional features and then using those features as inputs to a nonlinear predictive model (in this case a neural net), a company can better model relationships between mortality cells across the full spectrum of ages and durations when faced with sparse experience data. Vincelli's methodology could be extended to mortality tables pension practitioners use. The research also uses many of the steps outlined in Step 2 of this paper and therefore is itself a useful case study on how to apply predictive analytics to a pension type problem.

Claire-Koissi, Day and Whitledge (2019) provide a general overview to the field including data exploration and visualization and data analytics techniques (i.e., supervised and unsupervised; they do not cover reinforcement learning). Sample code is also provided for a few case studies. One case study in particular relates to mortality modeling. The authors also identify some useful R and Python packages with actuarial applications.

Nigri, Levantesi, and Marino (2020) use a recurrent neural network with a long short-term memory to forecast life expectancy and lifespan disparity, both independently and simultaneously at birth and age 65.

Perla, Richman, Scognamiglio and Wutrich (2021) forecast mortality rates using a simple shallow convolutional network model and find interesting results in terms of applicability.

### **Pension plan risk transfer**

Tian and Chen (2020) use K-Nearest Neighbor, Naïve Bayesian and Support Vector Machine algorithms to search the SEC database and identify various pension de-risking activities. They create the first historical database of these transactions. The database and approach can be very useful for those conducting empirical work on pension risk transfers.

Advanced mortality modeling techniques are also being used by insurance carriers to underwrite group annuity contracts which are used to transfer risk from plan sponsors to the insurer. Access to data, like nine-digit zip code, has also allowed insurance carriers to refine their plan population mortality estimates. Pension actuaries may want to adopt some of the same approaches so they are able to adequately negotiate on annuity pricing. Our example in Section 2 is a stylized illustration of how this might work.

Hadass, Laboure, Shen and Turner (2021) highlight the use of blockchain technology to share data in pension risk transfer deals occurring in the UK. While somewhat outside the scope of this paper, the idea that pension funds and insurance carriers are embracing new technologies and approaches is relevant.

Global retirement and investment consulting firm, Mercer UK, launched<sup>8</sup> in 2020 a pension risk transfer AI tool. The tool helps sponsors predict the outcome of a member options exercise program. Using anonymous data from completed member options exercises and the pension plan's own data, the machine learning algorithm determines the probability of a member accepting a tailored offer.

While not necessarily pension risk transfer, The Pension Regulator in the UK has developed a machine learning algorithm that can assess pension scheme risk on a monthly basis as opposed to the three-year evaluation process that was previously used<sup>9</sup>. The standard approach was to have a team of experts review 2,000 schemes a year. This led to results being outdated quite rapidly and exposes the UK insurance fund to risk. A well-known example is the collapse of the British company NHS in 2016. The scheme was risk assessed in 2009 and 2012 but subsequent valuations took too long allowing the fund to deteriorate without it being actively monitored. The machine learning model can perform risk assessment across all 6,000 pension schemes at once. The US PBGC and other pension regulators may want to consider adopting a similar approach.

### **Liability driven investment and asset allocation**

The use of predictive analytics is gaining increased traction amongst sophisticated investors. For example, the Journal of Financial Data Science was started in 2020 to publish investment research using advanced data techniques. A casual glance at other investment journals such as Financial Analyst Journal and Journal of Portfolio Management also shows an uptick in machine learning type articles. A good example is Das (2020) paper in the Journal of Investment Management. Das uses reinforcement learning and applies it to solving goals-based wealth management problems.

---

<sup>8</sup> <https://www.uk.mercer.com/newsroom/Mercer-launches-industry-first-AI-powered-tool-to-predict-outcome-of-member-options-exercises.html>

<sup>9</sup> <https://www.ipe.com/the-robot-regulator-uk-watchdog-reveals-use-of-machine-learning/10018449.article>

There are numerous other examples of published research attempting to apply predictive analytics to improve the results of specific investment strategies. For example, Yazdani (2020) uses machine learning to create recession indicators. Using ensemble machine learning models, with Random Forests performing best, recessions are predicted with high accuracy and greater reliability. James, Abu-Mostafa and Qiao (2019) use support vector machines to forecast recessions. They show how the predictions can be used to inform a dynamic asset allocation strategy. Pension funds can use these types of research findings to augment their portfolios and potentially improve investment performance. An extension would be to use this type of research to dynamically allocate between return seeking and liability hedging assets, which is a common way pension fund investments are designed.

Hegstrom (2016) discusses the effective communication of stochastic model results. This is a challenge for pension practitioners who need to communicate complicated results to decision-makers who may not be familiar with the concepts or have the time to engross themselves in the analysis. Hegstrom reviews histograms, kernel density plots, strip charts with jitters and violin plots.

Sasaki et al (2018) applied deep learning to detect investment styles of fund managers for the Japan Government Pension Investment Fund. While most pension practitioners are not performing investment manager research they are often involved, at least to some degree, with plan investments. The concepts in this paper can be applied to analyzing fixed income managers who are tasked with matching plan liabilities. DeMiguel et al (2021) applies machine learning to select portfolios of mutual funds. A pension investor could explore the techniques for their fund selection especially in the return seeking piece of their portfolio.

Shang (2021) applies deep reinforcement learning to the dynamic asset allocation problem for pension plan sponsors. He compares a reinforcement learning approach to the standard dynamic programming approach. He shows reinforcement learning can improve the design of investment strategies in terms of improving return and reducing risk.

While some of the above literature is not necessarily pension focused, many of the ideas and concepts can be ported over to the pension domain, either at the pension investment manager implementation level and/or at the portfolio level. Beyond the cited research articles, a very useful and recent textbook is by De Prado (2020).

### **Retirement decision making and defined contribution plans**

Pension actuaries are increasingly becoming more involved with solving broader retirement planning problems and with managing defined contribution plans<sup>10</sup>. Applications in this domain include:

- Predicting participants taking a lump sum – data can be gathered from the company’s warehouse or information from similar employers, provided by the company’s plan administrator or actuary, to better predict when a participant might take a lump sum.
- Predicting behaviors in a DC plan related to selecting investment options, taking loan withdrawals, and making other financial decisions.
- Communications – using chat-bots to more efficiently address participant questions related to their benefits. Communications are also being increasingly personalized and tailored through the use of predictive analytics and artificial intelligence.
- Pension dashboards – in the UK the government has made a commitment to support pension dashboards created by the pension industry. The dashboards allow participants to keep track of various pension

---

<sup>10</sup> The SOA has compiled a set of resources for retirement actuaries interested in DC plans. <https://www.soa.org/sections/retirement/defined-contribution-resources/>. We encourage interested readers to consult this link.

accounts and have an online snapshot of their retirement programs including defined benefit plans, defined contribution plans and the UK social security program. Israel and several European countries also provide some sort of pension dashboard.

- Cybersecurity enhancements via blockchain technology are increasingly being used by pension and defined contribution record keepers. This is highly relevant given the recent increase in cyberattacks, many of them very well publicized.

There is also an increased use of predictive analytics to retirement planning. Irlam (2020) uses reinforcement learning to solve a complex financial planning problem. He shows how using reinforcement learning creates substantial improvements over classic dynamic programming and Monte-Carlo simulations. Forsyth and Li (2019) use a neural network approach to solve for the optimal asset allocation phase of a defined contribution plan where a target objective function is used.

In short, while the current literature on predictive analytics is not tailored specifically for pension practitioners, one could imagine a future where the lifecycle of pension actuarial work fully embraces a predictive analytics approach. For example, we can consider what this might mean in the context of performing a standard pension service such as an annual actuarial valuation. To start the process, an automated email could be sent from the plan actuary to the plan sponsor requesting valuation census data. Data could be populated into a secure cloud-based portal. The actuary could extract the data and run preprogrammed automatic routines that perform data checks, impute missing values, clean the data and elevate specific issues regarding additional assistance. The data could also be run through various exploratory data visualization tools, such as the ones described in Section 2, that enable the actuary and client to see trends and analyze quickly any interesting features. Programming scripts can be run that would then automatically take the cleaned data and populate valuation software. Valuations can then be performed and results automatically populated into report templates and/or online dashboards or even an application a sponsor could download on their mobile device. Dashboards would enable daily funding level monitoring and funded status attribution analysis, among other relevant metrics. Additional information could be layered in, including but not limited to, pension asset information, forecasting, optimizations, and reporting modules. Some software providers have already embraced these principles<sup>11</sup>. We expect continued development and refinement in these areas. Embracing analytics and technology would make the entire pension valuation process more streamlined and quicker to complete. Clients would then have information sooner and be able to make more real-time and informed decisions. Taken to the extreme, one might even consider a time where there will be robo-actuaries similar to robo-advisors that exist in the investment domain.

## Section 4: Case Study: De-risking Activity Prediction

Pension risk transfer activities have proliferated in recent years. With defined contribution (DC) plans replacing defined benefit (DB) plans and transferring DB plan liabilities to third parties such as insurance companies, the trend of employer sponsored pension plans continues with shifting risks from plan sponsors to plan participants or counterparties. It is valuable to predict these de-risking activities to better prepare for the future landscape.

An existing example of analyzing de-risking activities is the analysis of single-employer pension plan risk transfers by the Pension Benefit Guaranty Corporation. In its 2020 report that summarizes the 2015-2018 PBGC Premium Filings, the PBGC studied risk transfer activities (RTA) and identified some key factors that affect the level of RTA. Plan size

---

<sup>11</sup> See Pfaroe (<https://pfaroe.moodyanalytics.com/>) and Winklevoss (<https://www.winklevoss.com/proval-ps/overview/>)

and PBGC variable-rate premiums were determined to be important variables while financial status of the plan was not as material a factor as perhaps expected.

In this case study, we use Form 5500 data<sup>12</sup> from 2010 to 2020 to study de-risking activities at the plan level using predictive models. This is different from existing studies given that the focus is not on individual key factors but rather the *combination* of these factors which can help identify more complex relationships. The ability to assess individual plans using predictive analytics can also help identify plans that are most likely to perform those de-risking activities and take a closer look to estimate the potential impact.

The goal is to predict whether a plan will conduct any de-risking activities in the next year and the explanatory variables are the demographic profile, fund contribution, asset allocation, funded status, and actuarial assumptions at the plan level at the end of the current year. Unlike the relative morality example in [Section 2](#) which is a regression problem, this case study is a classification problem that looks for a “Yes” or “No” answer to the question of “will this plan de-risk next year?”

#### 4.1 DATA PREPARATION

IRS Form 5500 data is selected given its size, accessibility, and relevance to the pension industry. The data is publicly available. IRS Form 5500 is an annual reporting requirement by the U.S. Department of Labor to disclose operations, funding and investment activities of about 800,000 retirement benefit plans. The database used for this case study contains 11 years of plan data from 2010 to 2020, with a total size of 4GB and 2.7 million records. Table 5 lists the major datasets used in the case study.

**Table 5**  
**FORM 5500 RAW DATA**

Dataset	No. of Variables	Information
Form 5500: Annual Return/Report of Employee Benefit Plan	135	Plan identification and basic plan information such as plan sponsor(s), plan type and number of plan participants
Schedule A: Insurance Information	90	Insurance, investment, annuity, and welfare contract information
Schedule H: Financial Information	158	Balance sheet and income statement of pension plans
Schedule SB: Single-Employer Defined Benefit Plan Actuarial Information	120	Funding status and actuarial assumptions

To narrow down the scope of the case study, the following criteria are used to filter the data:

- Only single-employer plans are included in the analysis.
- Total number of participants at the beginning of the plan year is no less than 1,000.

---

<sup>12</sup> The database was retrieved on June 1, 2021 from <https://www.dol.gov/agencies/ebsa/about-ebsa/our-activities/public-disclosure/foia/form-5500-datasets>. The link contains Form 5500 data by calendar year, form and schedule, and plan. The 2020 dataset was partially available at the time of download. The Data dictionary can be downloaded at <https://www.dol.gov/sites/dolgov/files/EBSA/about-ebsa/our-activities/public-disclosure/foia/form-5500-2019-data-dictionary.zip>



Single-employer plans are chosen given that de-risking activities are easier to implement compared to multi-employer plans. A minimum of 1,000 participants is used to focus on those with larger economic impacts. 31,354 out of 2.7 million records meet those criteria. Table 6 summarizes the de-risking plans by reporting year.

**Table 6**  
**DE-RISKING ACTIVITIES ESTIMATED FROM FORM 5500 FILTERED DATA**

Reporting Year	No. of plans	No. of plans with de-risking activities	% of de-risking plans
2011	3623	146	4.0%
2012	3631	173	4.8%
2013	3609	139	3.9%
2014	3540	165	4.7%
2015	3529	178	5.0%
2016	3467	183	5.3%
2017	3445	201	5.8%
2018	3371	196	5.8%
2019	3111	173	5.6%
2020 (Partial)	28	6	21.4%
Total	31354	1560	5.0%

Note, the increase in % of plans de-risking in 2020 is based on limited data and should not be interpreted as a true indication of increased activity.

Table 7 lists the data cleaning details applied to the raw data.

**Table 7**  
**FORM 5500 DATA CLEANING**

Item	Treatment	Comment
Response variable construction	Variable “risk_transfer” is created so that it is TRUE if the total number of participants reduced by more than 25%, and FALSE if otherwise.	Additional data sources can be used to directly identify plans with defined risk transfer activities. However, considering data privacy and the purpose of this case study, publicly available Form 5500 data is used to approximately identify these plans with de-risking activities.  The threshold “25%” plan participants reduction is chosen to largely align with the number of plans with 1,000 or more participants and risk transfer activities in PBGC (2020).
Yearly dataset combination	For each plan, previous year’s forms are used to retrieve values of explanatory variables based on multiple plan identification fields.	It is important to make sure that to predict the activities in year N, only year N-1 or earlier information is used as explanatory variables.
Missing Data Treatment	For subtotal variables, missing data is replaced with the sum of its components; Otherwise, missing data is replaced with zero.	The raw data has many missing data which corresponds to empty cells in the Form 5500 and its schedules. By applying filters such as the number of participants is no less than 1,000, a large portion of missing data has been addressed.
Feature Engineering	<ul style="list-style-type: none"> <li>• Many text fields such as plan name, identification number, addresses are removed from the raw data.</li> <li>• New variables are created such as funding ratio<sup>13</sup>, asset mix<sup>14</sup>, and participant mix<sup>15</sup>.</li> <li>• The State of the sponsor’s mailing address is converted to dummy variables each of which indicate whether it is a specific State.</li> </ul>	
Data Normalization	<ul style="list-style-type: none"> <li>• Many variables already have a value range of [0,1] and are kept untouched.</li> <li>• For variables with dollar amount, many of them have been converted to a percentage of subtotal variables, such as the asset mix and participant mix created in feature engineering.</li> </ul>	
Dimensionality Reduction	None	With only 443 variables in the clean dataset, it is manageable using normal computing capacities nowadays. Dimensionality reduction may lead to difficulty of interpreting the driving factors of de-risking activities. Here we are fortunate that we do not need to reduce the data.

<sup>13</sup> Funding ratio is calculated as the ratio of total asset and total liability in Schedule H.

<sup>14</sup> Asset mix variables such as “cash”, “govt bond”, “corp debt”, “public equity”, and “real estate” are created using Schedule H information to represent the asset allocation in terms of proportion of the total asset.

<sup>15</sup> Participant mix variables describe the portion of total participants in a plan, including active participants, retired participants, deceased participants whose beneficiaries are receiving benefits, etc.

In a real-world application of predictive analytics, data processing is usually the most challenging component given that practitioners can apply different data processing approaches and parameters that generate the clean data to be fed into the models. It has a big impact on model accuracy. This is also where domain knowledge can be valuable as it can inform the data processing piece of the project. For large datasets with many explanatory variables, it is important to have an efficient and at least semi-automated program to try different data processing choices. It is also notable that with over 400 explanatory variables, it is a daunting task for human beings to process that much information and identify relationships. Predictive analytics can help, but industry experts can still play a significant role improving model accuracy and validating the results.

## 4.2 PREDICTIVE MODEL

Although this case study is a classification problem, many models introduced before can be used. In addition to CART, Random Forests, and GBM models, we will also use Logistic regression and neural networks to perform the analysis.

**Logistic regression**, as a special case of **generalized linear models (GLMs)**, can be used to estimate the probability of the response variable based on the logistic function given below.

$$E(Y|X) = \mu = \frac{1}{1 + e^{-\eta}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

$E(Y|X)$  is the probability that a plan will perform de-risking activities or not. This simple yet sometimes powerful classification model is easy to understand and validate compared to other more complicated models. As part of the model, L2 regularization term  $\sum_{i=1}^n \beta_i^2$  is added to the error function to address the overfitting issue with many explanatory variables, as given below.

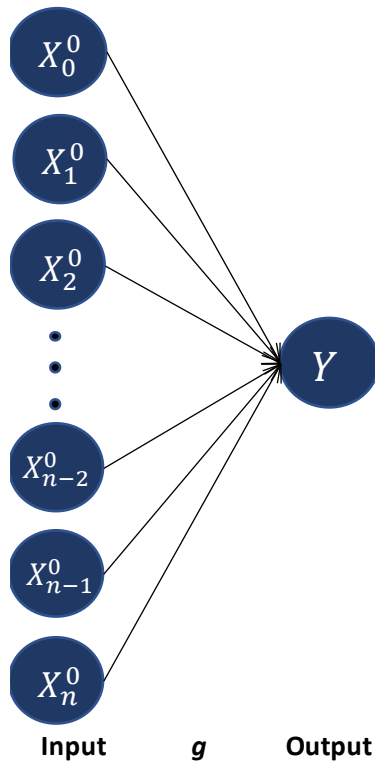
$$L = \frac{1}{m} \sum_{j=1}^m \left[ Y_j - \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1^j + \beta_2 X_2^j + \dots + \beta_n X_n^j)}} \right]^2 + \lambda \sum_{i=1}^n \beta_i^2$$

**CART, Random Forests, and GBM models** in [Section 2.3.1](#) are tree-based models. As described before, CART tries to find the best tree to make the prediction while Random Forests models use multiple trees with random components to make the prediction jointly. GBM models are using multiple trees as well but in a sequential way to minimize residual errors from the previous tree. In this classification task, the response variable becomes the probability of having a de-risking activity in the next year.

**Artificial neural networks (ANNs)** mimic biological neural networks to make predictions based on a large amount of data. Unlike traditional predictive models such as linear or Logistic regression, the mathematical function that describes the relationship between the response variable and explanatory variables is undefined. Rather, it uses multiple layers of linear, Logistic, or other simple functions to allow many more possible relationships. With enough data and appropriate training, ANN models are believed to mimic any complex relationship.

In its simplest format, an ANN with only an input layer, an output layer, and the Sigmoid activation function, is a Logistic regression, as shown in Figure 13.

**Figure 13**  
LOGISTIC REGRESSION IN THE FORM OF ANN



Output layer:  $Y = g(\theta^0 \times x^0)$  for  $j > 0$

$$X_0^0 = 0$$

$x^0$ : an  $(n+1)$  element column vector containing all the explanatory variables and the intercept  $x_0^0$ .

$\theta^0$ : an  $(n+1)$  element row vector containing the weights applied to all explanatory variables and the intercept to determine the value of output  $Y$ .

Subscript "0" stands for the first layer in the neural network, which is also used in more complicated networks discussed later.  $\theta^0 \times x^0$  is a linear combination of the neurons in the input layer. Depending on its value, the next step is to determine whether the output should be activated or not, like the way our brains work. Activation function  $g$  can be considered as a mechanism to bring in nonlinear relationships and bring the value range down to a manageable level. The Sigmoid function is defined as

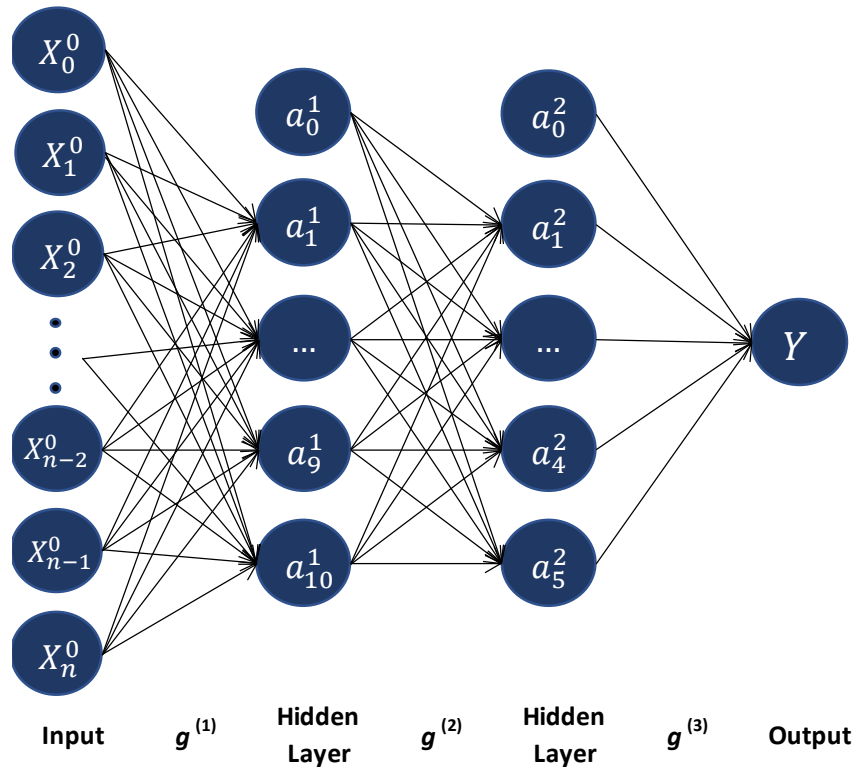
$$g(x) = \frac{1}{1 + e^{-x}}$$

This transforms to a pure Logistic regression function:  $Y = g(X) = \frac{1}{1 + e^{-X}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \dots + \theta_n X_n)}}$

Through this simple structure, we can see that an ANN can be decomposed into layers, linear functions, and activation functions. To make the structure more complicated for potentially better results, we can increase the number of layers and change the activation function types.

Figure 14 shows a simple ANN model with input, output, and two hidden layers.

**Figure 14**  
ANN MODEL STRUCTURE



The layers are linked together with activation functions  $g$ . Each neuron in the hidden layers and output layer is determined by the neurons in the previous layer.

*First hidden layer:*  $a_0^1 = 0$  and  $a_j^1 = g(\theta_j^0 \times x^0)$  for  $j > 0$

$x^0$ : an  $(n+1)$  element column vector containing all the explanatory variables and the intercept  $x_0^0$ .

$\theta_j^0$ : an  $(n+1)$  element row vector containing the weights applied to all explanatory variables and the intercept to determine the value of neuron  $a_j^1$ .

*Second hidden layer:*  $a_0^2 = 0$  and  $a_j^2 = g(\theta_j^1 \times a^1)$  for  $j > 0$

$a^1$ : an 11-element column vector containing all the neurons in the first layer.

$\theta_j^1$ : an 11-element row vector containing the weights applied to all the neurons in the first layer to determine the value of neuron  $a_j^2$ .

*Output layer:*  $Y = g(\theta^2 \times a^2)$

$a^2$ : a 6-element column vector containing all the neurons in the second layer.

$\theta^2$ : a 6-element row vector containing the weights applied to all the neurons in the second layer to determine the value of the output variable  $Y$ .

A neuron's value  $a_j^i$  depends on a linear combination of the neurons in the previous layer ( $\theta_j^{i-1} \times a^{i-1}$ ).

In addition to the Sigmoid activation function, other choices including Tanh, ReLu, SoftMax are available, as introduced in [Section A.3.1](#). The choice of activation function can be arbitrary and by trial and error. In this case study, Tanh is found the best option compared to others.

$$\text{Tanh}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

### 4.3 MODEL TRAINING, VALIDATION AND SELECTION

With all the model candidates, the next step is to train the models and perform validation. In this exercise, an 80/20 split is used to create a training dataset and a validation dataset. The training dataset (in-the-sample data) is used to calibrate the models. The validation dataset (out-of-sample data) is used to validate the data. Results shown in the section are based on validation dataset, which is essential for finding any overfitting issues.

When assessing whether a model is good for solving a classification problem, different measures from those used with regression are used. Precision, recall and the F-measure are popular measures based on the confusion matrix, as shown in Table 8.

**Table 8**  
**SAMPLE CONFUSION MATRIX**

	Predicted: True	Predicted: False
Actual: True	True Positive	False Negative
Actual: False	False Positive	True Negative

Precision measures the Type I error<sup>16</sup> and recall measures the Type II error. F-measure (or F-score) is the harmonic average of precision and recall and may be used as a high-level measure to rank the performance of different models.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall (True Positive Rate)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$F - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table 9 lists the confusion matrices and model accuracy results for each model type. Efforts were made to improve results under each model type by adjusting model hyperparameters such as regularization term, number of random trees, maximum depth of trees and number of hidden layers. For each pension plan in the validation dataset, the model will generate a probability of whether that plan would have a de-risking activity in the next year. If the probability is no less than 50%, that plan will stay in column “Predicted: True”. And if the plan did not have a de-risking activity in that year, it then belongs to False Positive, the combination of “Actual: False” and “Predicted: True”.

---

<sup>16</sup> Recall from classical statistics, a Type 1 error is a false positive where you reject a true hypothesis. A Type II error is a false negative and occurs when you fail to reject a false hypothesis.

**Table 9**  
**CLASSIFICATION MODEL VALIDATION RESULTS**

Model		Predicted: True	Predicted: False	Precision	Recall	F-measure
Logistic Regression	Actual: True	142	175	83.0%	44.8%	58.2%
	Actual: False	29	5925			
CART <sup>17</sup>	Actual: True	144	173	71.3%	45.4%	55.5%
	Actual: False	58	5896			
Random Forests <sup>18</sup>	Actual: True	123	194	98.4%	38.8%	55.7%
	Actual: False	2	5952			
GBM <sup>19</sup>	Actual: True	138	179	90.2%	43.5%	58.7%
	Actual: False	15	5939			
ANN <sup>20</sup>	Actual: True	131	186	90.3%	41.3%	56.7%
	Actual: False	14	5940			

Models generally showed a high precision and a low recall rate. The model accuracy is not perfect but satisfactory given that de-risking activities are not solely determined by plan information but many other factors such as management decisions and market conditions. Some randomness is expected as no one can be sure what will happen in the next year. The ability to identify around 40% of the plans with future de-risking activities and have a high precision showed the potential of predictive analysis to assess individual plans.

It is important to note that our goal is to identify those plans that are more likely to conduct de-risking activities in the near future. Therefore, our “True” cases in the confusion matrix are those with future de-risking activities. However, a common error found in using these standard validation routines are that “True” category is defined incorrectly. Table 10 shows the different results of precision, recall and F-measure if we flip the true and false definition.

---

<sup>17</sup> This CART model has a max tree depth of 20.

<sup>18</sup> This Random Forests model uses 500 trees with a max tree depth of 25.

<sup>19</sup> This GBM model has 500 sequential trees with a max tree depth of 20.

<sup>20</sup> This ANN model has two hidden layers with 100 and 25 neurons, respectively.

**Table 10**  
**CLASSIFICATION MODEL VALIDATION RESULTS**

Definition		Predicted: True	Predicted: False	Precision	Recall	F-measure
True: De-Risking	Actual: True	142	175	83.0%	44.8%	58.2%
	Actual: False	29	5925			
True: No de-risking	Actual: True	5925	29	97.1%	99.5%	98.3%
	Actual: False	175	142			

In many cases, using the wrong definition of the “True” category gives us very high model accuracy. This is not always easy to recognize as both “true” and “false” categories are used in the calculation. It is a prevalent issue as many of the classification tasks are dealing with imbalanced data. In our case, only 5% of the data has de-risking activities. If we simply predict that all plans do not have de-risking activities in the future, we can still get 95% precision using the no de-risking category as the “true” category.

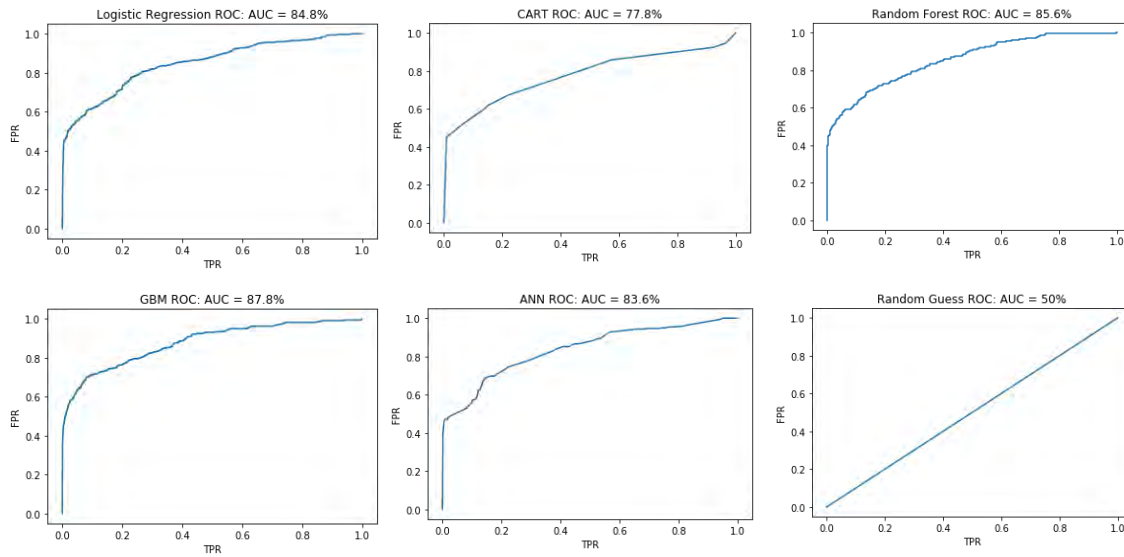
Another widely used measurement in classification problems is the Receiver Operating Characteristic (ROC) curve and the area under the curve (AUC). The ROC curve helps understand the tradeoff between the true positive rate (TPR) and the false positive rate (FPR) by varying the threshold that is used to determine whether a prediction is positive or negative.

$$fall - out (False Positive Rate) = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

The AUC tells the capability of the classification model to distinguish between two classes at different thresholds. In Table 10, a threshold of 0.5 is used by default to determine that whether we should predict it as a “Yes” or “No” given a predicted probability. In many cases, the threshold can change to get different tradeoffs of true positive rate and false positive rate. It is a more general measure of prediction accuracy for classification models. Figure 15 shows the ROC curves of each model used in the case study, in addition to the ROC of a random guess method. The closer the curve is to line TPR = 0 and line FPR =1, the better the classification model. AUC can be used to rank classification models as an aggregate measure. In this case, the GBM model has the highest AUC.



**Figure 15**  
ROC GRAPHS



Based on precision, recall, F-measure, and AUC, the GBM model is the best choice in terms of model accuracy. However, the marginal gain of using the GBM model compared to the simplest Logistic regression model is not material. In a real-world situation, it is very likely that the Logistic model will be chosen for implementation given that it is easy to understand and validate compared to the more complicated GBM model.

In conclusion, this case study shows the possibility and potential benefit to assess individual plans using large volumes of data using predictive analysis. With solid model training and model validation methods, credible and useful relationships can be identified and applied to improve existing strategies.

## Section 5: Conclusion

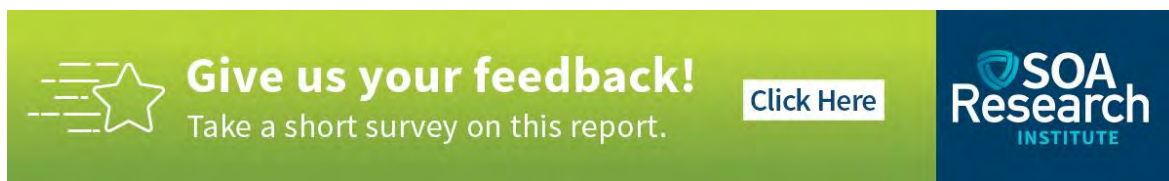
In this research, we reviewed the process to solve problems using a predictive analytics framework. To that end, we built two new illustrative case studies of predictive analytics to introduce predictive analysis to pension actuaries.

The first case study on relative mortality prediction explained important concepts and techniques of predictive analysis without touching many technical details. It encompasses the end-to-end process for a viable predictive analysis, including data collection, data cleaning, predictive models, model training, model validation, and model implementation.


The second case study uses a much larger and complex dataset to predict de-risking activities of single-employer pension plans. Unlike the first case study which is a regression analysis, the second case study is a classification task that utilizes different methods for model validation and model selection. The result also shows the potential benefits of using predictive analysis to assess individual plans, which is unlikely with traditional methods given the data volume.

The two case studies demonstrate that vast data is publicly available and together with predictive analysis, they can be used to extend our current work and make it into more production-ready real life usable models.


Our research also explored different areas in the pension and retirement field regarding the existing and future applications of predictive analysis. With examples and suggested areas for future applications, this research is expected to attract more interest in utilizing predictive analysis in the pension and retirement field.



The banner is a horizontal strip with a green-to-blue gradient. On the left, there is a white star icon with motion lines. To its right, the text 'Give us your feedback!' is written in a bold, white font, followed by 'Take a short survey on this report.' in a smaller white font. A white rectangular button with the text 'Click Here' is positioned to the right of the text. On the far right, the SOA Research Institute logo is displayed in white and blue.

 **Give us your feedback!**  
Take a short survey on this report.

[Click Here](#)

 **SOA**  
**Research**  
INSTITUTE

## Section 6: Acknowledgments

The authors would like to thank all members of the Project Oversight Group (POG) tasked with providing governance on this research project. This paper would not have attained its current level of relevance to practitioners without the POG's guidance, feedback and insightful input.

"Predictive Analytics in Retirement" POG members are the following:

- Gavin Benjamin
- Gordon Enderle
- Tariq Hussain
- Yazhan Lin
- Larry Pollack
- Anna Rappaport
- Steven Siegel (SOA Sr. Research Actuary)
- Haofeng Yu

The authors would also like to thank Barbara Scott for her effective coordination of this project, as well as the sponsorship and funding support of the Society of Actuaries through the Aging & Retirement Strategic Research Program.

## References

- American Academy of Actuaries, 2018. “Big Data and the Role of the Actuary”.  
<https://www.actuary.org/sites/default/files/2019-10/BigDataAndTheRoleOfTheActuary.pdf>
- Anscombe, Francis J., 1973. “Graphs in Statistical Analysis”. American Statistician. 27 (1): 17–21
- Cantor, Buchen, Forman, Gamzon, 2011. “Embedded Options in Pension Plans”. Society of Actuaries.  
<https://www.soa.org/globalassets/assets/files/research/projects/research-catalogue-survey-report.pdf>
- Cantor, 2014. “Valuation of Guarantees in Cash Balance Plans”. Society of Actuaries.  
<https://www.soa.org/globalassets/assets/files/research/projects/research-emb-opt-val-cash-report.pdf>
- Claire-Koissi, Day, Whitlege, 2019. “Emerging Data Analytics Techniques with Actuarial Applications”.  
<https://www.soa.org/resources/research-reports/2019/emerging-analytics-techniques-applications/>
- Chalk, McMurtie, 2016. “A Practical Introduction to Machine Learning Concepts for Actuaries”.  
[https://www.casact.org/sites/default/files/database/forum\\_16spforum\\_chalk\\_mcmurtrie.pdf](https://www.casact.org/sites/default/files/database/forum_16spforum_chalk_mcmurtrie.pdf)
- Das, Varma, 2019. “Dynamic Goals Based Wealth Management using Reinforcement Learning”.  
[https://srdas.github.io/Papers/GBWM\\_RL2.pdf](https://srdas.github.io/Papers/GBWM_RL2.pdf)
- Department for Work and Pensions, 2018. “Annual Report and Accounts 2017-2018”: 63  
[https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment\\_data/file/721224/dwp-annual-report-and-accounts-2017-2018.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/721224/dwp-annual-report-and-accounts-2017-2018.pdf)
- de Prado, 2020. “Machine Learning for Asset Managers”. <https://www.cambridge.org/core/books/machine-learning-for-asset-managers/6D9211305EA2E425D33A9F38D0AE3545>
- DeMiguel, Gil-Bazo, Nogales, Santos, 2021. “Can Machine Learning Help to Select Portfolios of Mutual Funds”.  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3768753](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3768753)
- Andrew, Theo, 2018. “Finnish AI testing successfully identifies future retirees facing disability pension”  
<https://www.europeanpensions.net/ep/Finnish-AI-successfully-identifies-future-retirees-facing-disability-pension.php>
- Forsyth, Li, 2018. “A Data Driven Neural Network Approach to Optimal Asset Allocation for Target Based Defined Contribution Pension Plans”. Insurance: Mathematics and Economics  
[https://cs.uwaterloo.ca/~paforsyt/Data\\_NN.pdf](https://cs.uwaterloo.ca/~paforsyt/Data_NN.pdf)
- Hadass, Laboure, Shen, Turner, 2021. “Fintech and Retirement Savings”. Society of Actuaries.  
<https://www.soa.org/globalassets/assets/files/resources/research-report/2021/fintech-retirement-savings.pdf>
- Hegstrom, 2016. “Effective Communication of Stochastic Model Results”. Society of Actuaries Data Visualization Call for Essays. <https://www.soa.org/resources/essays-monographs/2016-data-visualization-essays/>
- Institute for Health Metrics and Evaluation (IHME). “United States Mortality Rates by County 1980-2014”. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2016.  
<http://ghdx.healthdata.org/record/ihme-data/united-states-mortality-rates-county-1980-2014>
- Irlam, 2020. “Multi Scenario Financial Planning via Deep Reinforcement Learning AI”.  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3516480](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3516480)

- Irlam, 2020. "Financial Planning via Deep Reinforcement AI". Journal of Retirement [https://papers.ssrn.com/sol3/Delivery.cfm/SSRN\\_ID3201703\\_code2306779.pdf?abstractid=3201703&mirid=1](https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3201703_code2306779.pdf?abstractid=3201703&mirid=1)
- James, Abu-Mostafa, Qioa, 2019. "Machine Learning for Recession Prediction and Dynamic Asset Allocation." Journal of Financial Data Science. <https://jfds.pm-research.com/content/1/3/41>
- Lowe, Bradley, Nicholson, Inglis, Balaam, Lawson, Wasserman, 2021. "The pensions dashboard: an actuarial perspective". British Actuarial Journal <https://www.cambridge.org/core/journals/british-actuarial-journal/article/pensions-dashboard-an-actuarial-perspective/B4EE08846366B8476B6C43472D3CFE91>
- Marino, Levantesi, Nigri, 2021. "Deepening Lee-Carter for longevity projections with uncertainty estimation". <https://arxiv.org/abs/2103.10535>
- Pension Benefit Guaranty Corporation, 2020. "Analysis of Single-Employer Pension Plan Partial Risk Transfers" [https://www.pbgc.gov/sites/default/files/2020-risk-transfer-report.pdf?utm\\_medium=email&utm\\_source=govdelivery](https://www.pbgc.gov/sites/default/files/2020-risk-transfer-report.pdf?utm_medium=email&utm_source=govdelivery)
- Perla, Richman, Scognamiglio, Wuthrich, 2021. "Time-series forecasting of mortality rates using deep learning". Scandinavian Actuarial Journal. <https://www.tandfonline.com/doi/abs/10.1080/03461238.2020.1867232>
- Richman, 2020. "AI in Actuarial Science – a review of recent advances Part 1". Annals of Actuarial Science. <https://www.cambridge.org/core/journals/annals-of-actuarial-science/article/abs/ai-in-actuarial-science-a-review-of-recent-advances-part-1/65D135D28505261F431EBEC0220DF0B0>
- Richman, 2020. "AI in Actuarial Science – a review of recent advances Part 2". Annals of Actuarial Science. <https://www.cambridge.org/core/journals/annals-of-actuarial-science/article/abs/ai-in-actuarial-science-a-review-of-recent-advances-part-2/C35A295A1F3ECC3013EA4D953706694A>
- Rosenblatt, Frank, 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." Psychological Review, 65 (6): 386-408 <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf>
- Sasaki, T., Koizumi, H., Tajiri, T., Kitano, H., 2018. "A Study on the Use of Artificial Intelligence within Government Pension Investment Fund's Investment Management Practices (Summary Report)". Tokyo, Japan: Government Pension Investment Fund. [https://www.gpif.go.jp/en/investment/research\\_2017\\_1\\_en.pdf](https://www.gpif.go.jp/en/investment/research_2017_1_en.pdf)
- Shang, Huang, Su, 2013. "Pension Plan Embedded Option Valuation". Society of Actuaries. <https://www.soa.org/globalassets/assets/Files/Research/Projects/2013-pension-plan-embed-opt-val.pdf>
- Shang, Kailan, 2017. "Variable Selection in Predictive Modeling: Does it Really Matter?" *Society of Actuaries Predictive Analytics and Futurism Newsletter*, Issue 15: 48-53. <https://www.soa.org/globalassets/assets/library/newsletters/predictive-analytics-and-futurism/2017/june/2017-predictive-analytics-newsletter-issue-15.pdf>
- Shang, Kailan, 2021. "Deep Learning for Liability-Driven Investment." Society of Actuaries. <https://www.soa.org/globalassets/assets/files/resources/research-report/2021/liability-driven-investment.pdf>
- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research 15 (2014) 1929:1958.

The U.S. Census Bureau, 2017. “2016 Planning Database”.

<https://www.census.gov/data/datasets/2016/adrm/research/2016-planning-database.html>

Tian, Chen, 2020. “De-risking Strategies of Defined Benefit Plans: Empirical Evidence from the United States”.

Society of Actuaries. <https://www.soa.org/globalassets/assets/files/resources/research-report/2020/de-risking-strategies.pdf>

Tripp, 2018. “What data science means for the future of the actuarial profession”, British Actuarial Journal, Institute and Faculty of Actuaries, Sessional Research Event

<https://www.cambridge.org/core/journals/british-actuarial-journal/article/what-data-science-means-for-the-future-of-the-actuarial-profession/B42EF98A53ED8D6E42606FAE664021D2>

Vincelli, Marc, 2019. “A Machine Learning Approach to Incorporating Industry Mortality Table Features into a Company’s Insured Mortality Analysis”.

<https://www.soa.org/resources/research-reports/2019/2019-machine-learning-approach/>

Yazdani, 2020. “Machine Learning Prediction of Recessions: An Imbalanced Classification Approach”. Journal of

Financial Data Science. <https://jfds.pm-research.com/content/2/4/21>

## Appendix A: Predictive Modeling in Condensed Form

This appendix is provided for more advanced and interested readers.

A complete application of predictive modeling requires not only the models to be used, but also many other components including data, model selection, and prediction. This appendix is an extension of [Section 2](#), expanded in two areas:

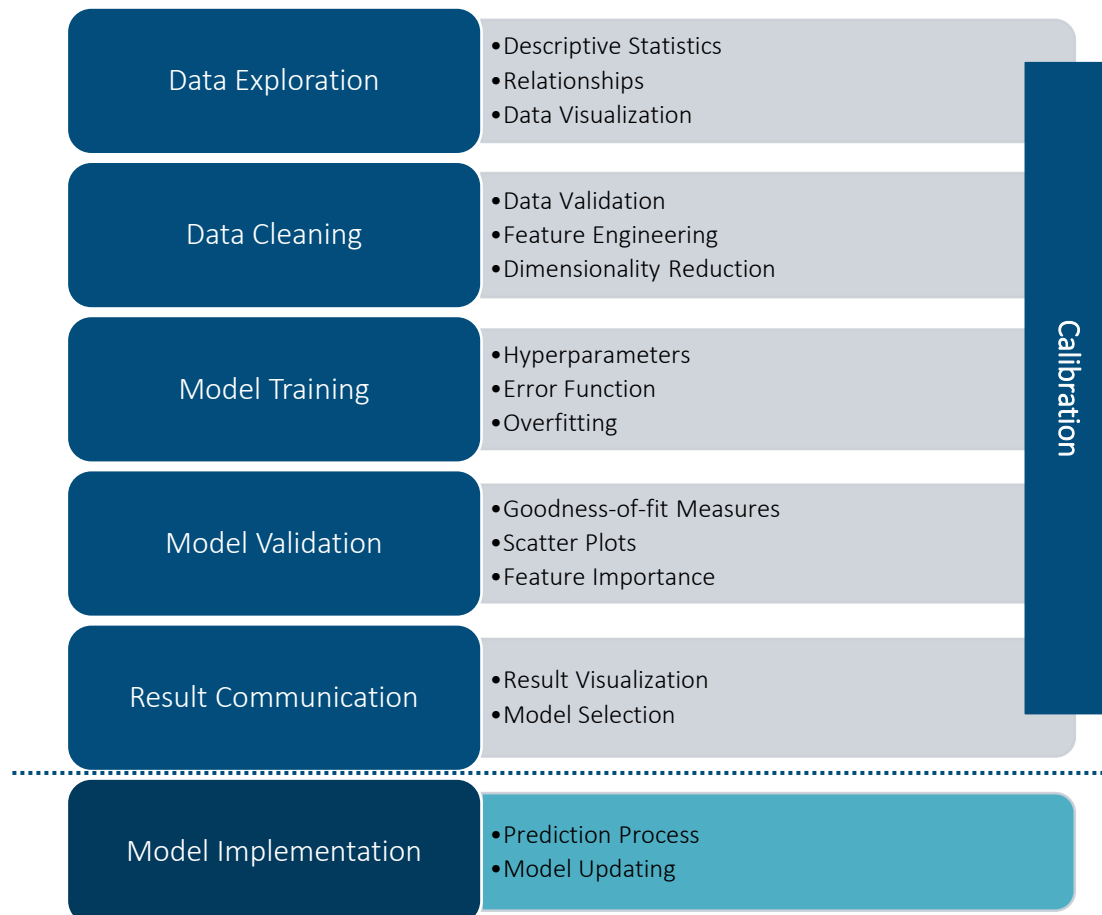
1. More choices of data processing, models and modeling approaches are introduced.
2. More examples are provided to illustrate these concepts.

We wanted to make this appendix self-contained. Therefore, some content from Section 2 and Section 4 is repeated.

Figure A.1 shows a typical predictive modeling process, which is composed of two major parts: calibration and implementation.

**Figure A.3**

### PREDICTIVE MODELING SAMPLE PROCESS



The rest of this section explains each component in the process, focusing on concepts, options, and usage.

## A.1 EXPLORATORY DATA ANALYSIS

As the first step in predictive modeling, exploratory data analysis (EDA) is a model-free approach to summarize data and relationships among variables using descriptive statistics and visualization. The goal of EDA is to provide an overview of the data and spot any interesting trends or relationships that may be helpful for constructing predictive models and validating the results. Pension actuaries already employ a version of EDA in their standard valuation actuarial reports when they plot metrics such as trends in funded status and population statistics.

Table A.1 lists some typical measures used in EDA, including descriptive statistics, relationships, and data visualization.

**Table A.2**  
**EDA COMPONENTS**

	Category	Detail
Descriptive Statistics	Moment	Mean, variance, skewness, kurtosis Summarize the data with its average value, risk, shape of distribution, and comparison to normality
	Range	Min, max
	Median	Indicate the shape of the distribution when compared to its mean
	Mode	Identify the most frequent value which can be useful for discrete distributions
	Quantile	How many values of a distribution are above a certain limit
	Conditional Tail Expectation	Indicate the average of extreme values
Relationships	Correlation Coefficient	Both temporal and contemporary relationships can be studied.
	Rank Correlation	Correlation measurement based on the <i>ranks</i> of data, rather than the data itself. Examples include Spearman's $\rho$ and Kendall's $\tau$
Visualization	Histogram	Graph of empirical distribution to be compared with the density function of a fitted distribution
	Boxplot	Graph of quantiles of variables using mean, upper quantile, and lower quantile. It is a parsimonious approach to represent the distributions
	Run chart	Graph of data in a time sequence to identify time trend
	Scatterplot	Graph of a pair of variables to identify relationships

To facilitate the introduction of predictive modeling, sample data is used for illustration purposes in Section A.2. Form 5500 data is selected given its size, accessibility, relevance to the pension industry, and usage in the case studies. Form 5500 is an annual reporting requirement by the U.S. Department of Labor to disclose operations, funding and investment activities of about 800,000 retirement benefit plans. The database contains 20 years of plan data from 2000 to 2019. It contains data of Form 5500 and its schedules including plan level information on insurance, financial transactions, financial reporting, actuarial assumptions, and service providers. The asset information contained in Schedule H is used in the appendix.

By looking at the Schedule H of Form 5500 data in 2019, for 21,001 plans with total assets greater than \$1,000, the asset mix data has the following descriptive statistics.



**Table A.2**  
**PLAN ASSET MIX DESCRIPTIVE STATISTICS**

Asset Class	Total Asset Mix	Mean	Standard Deviation	Min	1st Quartile	Median	3rd Quartile	Max
Cash	3.0%	6.7%	19.3%	-8.1%	0.0%	0.1%	2.9%	152.0%
Receivables	1.7%	3.4%	11.0%	-52.0%	0.0%	0.4%	2.1%	148.2%
Govt bond	6.7%	3.0%	11.4%	0.0%	0.0%	0.0%	0.0%	100.0%
Corporate debt	7.3%	3.5%	12.8%	-25.0%	0.0%	0.0%	0.0%	100.0%
Public equity	18.4%	9.0%	24.9%	-0.7%	0.0%	0.0%	0.0%	100.0%
Real estate	0.8%	0.2%	2.6%	0.0%	0.0%	0.0%	0.0%	99.9%
Loans	0.5%	0.9%	3.0%	0.0%	0.0%	0.0%	1.0%	98.7%
Employer related investment	2.1%	5.3%	21.1%	0.0%	0.0%	0.0%	0.0%	100.0%
Building	0.0%	0.1%	1.4%	0.0%	0.0%	0.0%	0.0%	75.3%
Partnership	2.9%	1.3%	8.3%	0.0%	0.0%	0.0%	0.0%	100.0%
Others*	56.6%	66.7%	40.7%	-61.6%	20.7%	93.4%	98.2%	101.1%

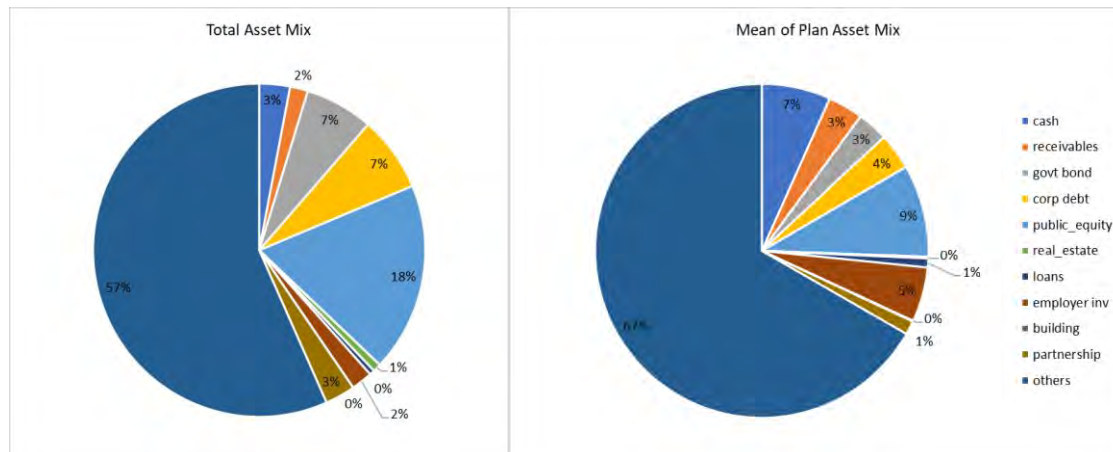
Note:

Others: Value of interest in common/collective trusts, pooled separate accounts, master trust investment accounts, 103-12 investment entities, registered investment companies, and funds held in insurance company general account.

Some asset classes have negative amount for certain pension plans which also indicates potential issues with the submitted forms and users may want to exclude them in further analysis.

Visualization can be used to represent the statistics. For example, Figure A.2 uses a pie chart to show the asset mix: asset mix of the total asset of 21,001 plans, and the arithmetic average of the asset mix of individual plans.

**Figure A.2**  
**PREDICTIVE MODELING SAMPLE PROCESS**



Descriptive statistics are helpful for gaining a high-level understanding of the data. However, data can behave significantly differently with the same or similar descriptive statistics. The famous Anscombe's quartet constructed by the statistician Francis Anscombe illustrates the idea with extreme yet powerful cases. Four sets of data are given in Table A.3. The four Y variables have the same mean and standard deviation. The relationship with X variables are

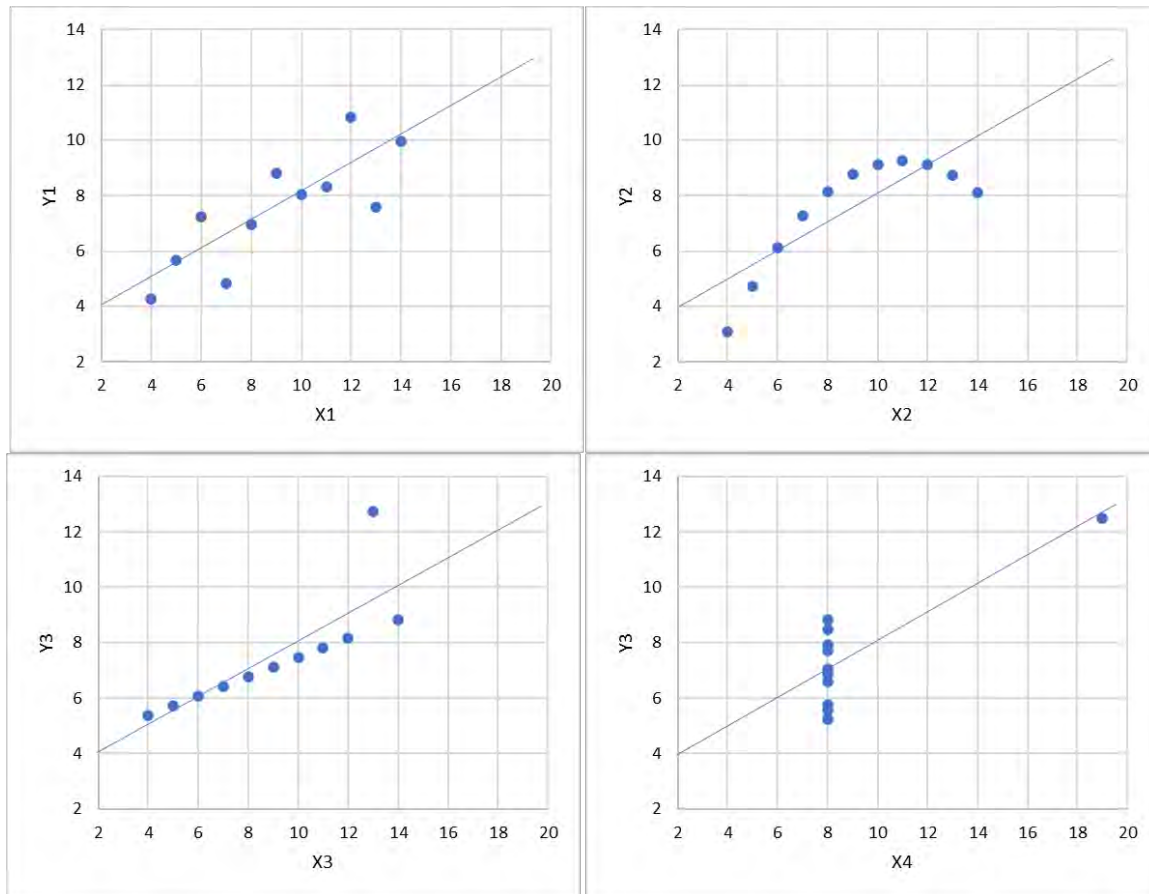
also similar with the same linear function with ordinary least squares and the coefficient of determination ( $R^2$  as explained in [Section A.5.1](#))

**Table A.3**  
**ANSCOMBE QUARTET**

$X_{1,2,3}$	$Y_1$	$Y_2$	$Y_3$	$X_4$	$Y_4$
10	8.04	9.14	7.46	8	6.58
8	6.95	8.14	6.77	8	5.76
13	7.58	8.74	12.74	8	7.71
9	8.81	8.77	7.11	8	8.84
11	8.33	9.26	7.81	8	8.47
14	9.96	8.1	8.84	8	7.04
6	7.24	6.13	6.08	8	5.25
4	4.26	3.1	5.39	19	12.5
12	10.84	9.13	8.15	8	5.56
7	4.82	7.26	6.42	8	7.91
5	5.68	4.74	5.73	8	6.89
Mean	7.50	7.50	7.50		7.50
Standard Deviation	2.03	2.03	2.03		2.03
Correlation with X	0.816	0.816	0.816	Correlation with $X_4$	0.816
Linear Regression	$Y=3+0.5X$				
$R^2$	0.666	0.666	0.666		0.666

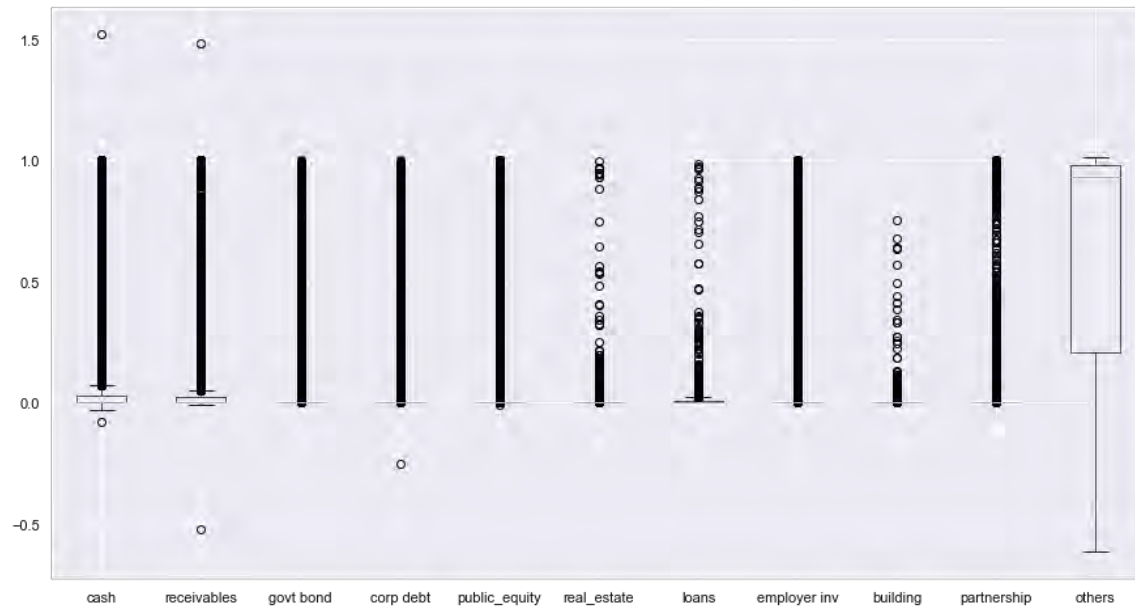
However, by plotting the scatter plots between X and Y variables, the patterns are totally different, as shown in Figure A.3. This illustrates that no matter what measures or models are used to analyze data, some details are lost which can be important. Visualization is helpful for identifying patterns that may not necessarily be captured by models.

**Figure A.3**  
SCATTER PLOTS OF ANSCOMBE QUARTET



Many visualization tools are available to explore data. Continuing with the 2019 Schedule H of Form 5500 data, Figure A.4 are the box plots that show the distributions of variables. Each box represents the data between the first and third quartile (Q1 and Q3), with the line in the middle as the median. The box extends to wider data range between  $[Q1 - 1.5(Q3 - Q1), Q3 + 1.5(Q3 - Q1)]$ , where available. For outliers outside this range, they are plotted as individual dots. For the asset mix data, expected for “others”, other asset classes show a right skewed distribution of allocation percentage.

**Figure A.4**  
SAMPLE BOX PLOTS



Heatmaps that show the correlation matrix among variables is also helpful for identifying relationships. Figure A.5 shows the heatmap of the Schedule H data. It is notable that government bond and corporate debt have a correlation of 0.28. Cash allocation has a negative correlation with all other asset classes.

**Figure A.5**  
SAMPLE HEATMAP

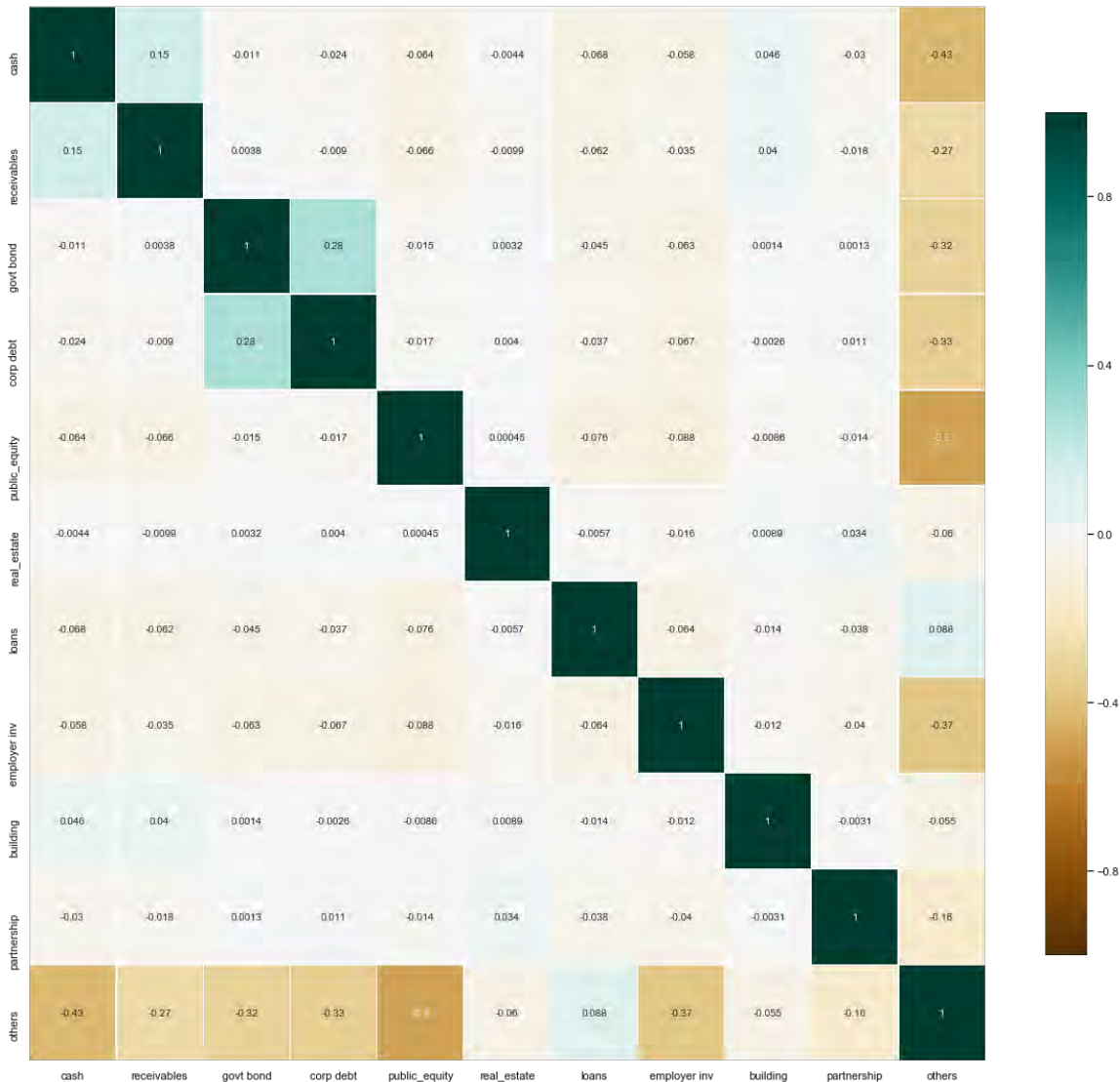
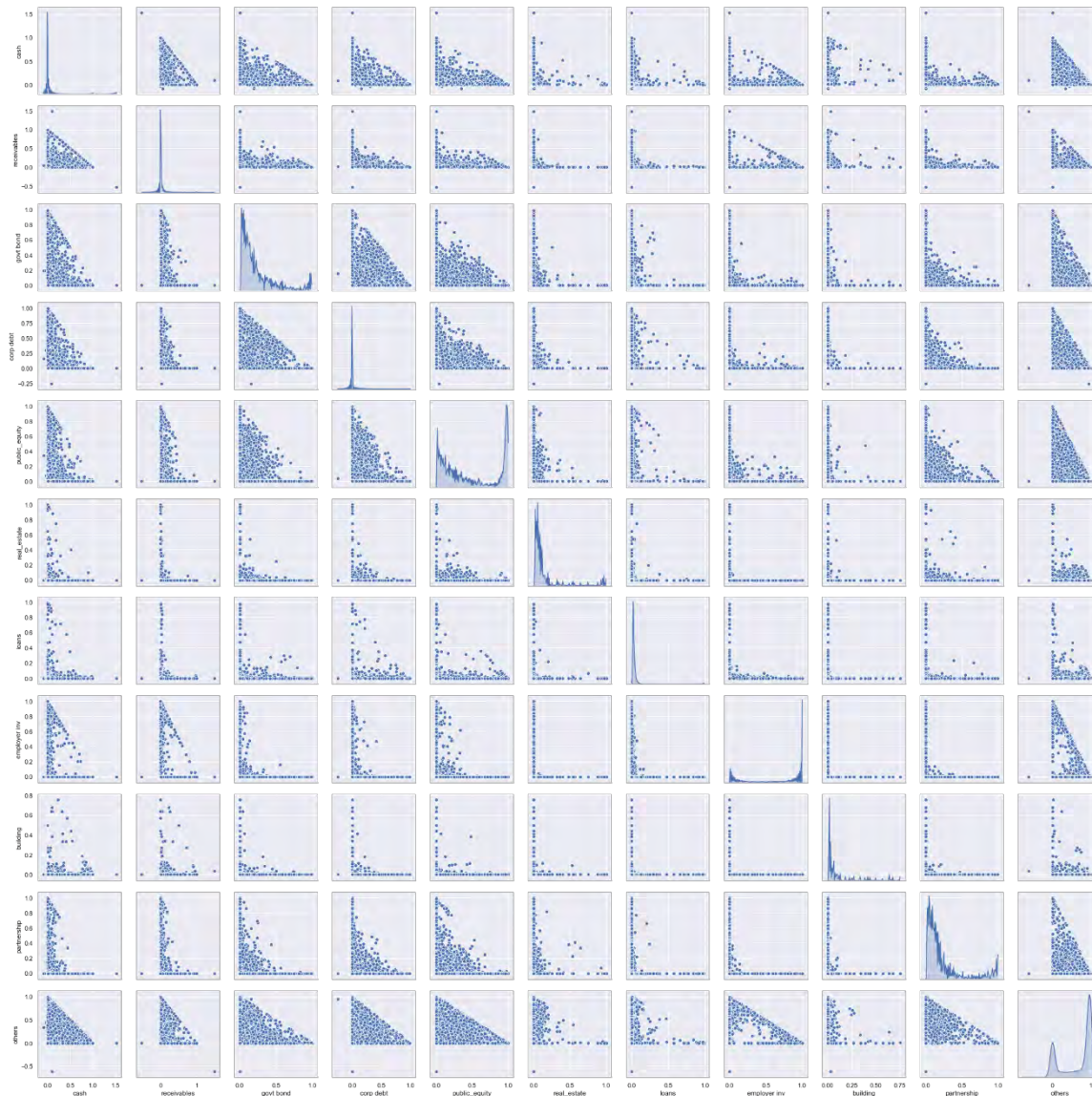


Figure A.6 shows an example of pair plots that contains scatter plots of all data variables, with the diagonal containing the histogram of each variable. The histograms show that some asset classes have a bimodal distribution. All the scatter plots have dots in the lower left triangles which is expected because the total of asset allocation percentages cannot exceed 1. There is no strong correlation observed in the scatter plots, as indicated in the heatmap as well. Some scatter plots such as the one of real estate and employer related investment have an L shape which means that these two asset classes are exclusive in most cases. Some scatter plots have many dots on their diagonal, which indicate some plans invest most, if not all, of their assets in two asset classes.

**Figure A.6**  
SAMPLE PAIR PLOTS



EDA is a general concept that contains all kinds of data exploration without formal modeling. What is described above is only a small portion of what is available in this field<sup>21</sup>. Pension actuaries may also want to gain familiarity with Datawrapper, Flourish, RAWGraphs, Power BI or Tableau.

## A.2 DATA CLEANING

After the EDA, it is easier to define the prediction task and determine what data may be used. However, before the data can be fed into the predictive models, additional processing is needed to adjust the data inputs to potentially

<sup>21</sup> Two recent books on the subject include “How Charts Lie: Getting Smarter about Visual Information” and “The Truthful Art: Data, Charts, and Maps for Communication”. Both are by Alberto Cairo.

improve model accuracy. This section will discuss some of the techniques used for data cleaning. Missing data treatment, data normalization, and feature engineering will be explained.

### A.2.1 MISSING DATA TREATMENT

Missing data is quite common, especially with large datasets. For example, the input for a field in the Form 5500 may be missing for a plan, which makes the plan's data not readily usable by the model.

This can be addressed by several approaches as follows.

- Remove the data record with missing data. This may be an option if the remaining data records are sufficient for predictive analytics.
- Fill the missing data with a unique value that has not been used by other records with full data. For example, we may replace the missing data in data field "" with 999999. This will indicate if there is missing data or not and may be helpful for prediction. In the Schedule H data, empty fields are normal for asset amounts of certain asset classes. It is straightforward to replace them with 0, which indicates no allocation to these asset classes.
- Fill the missing data with the average value of the data field. The average value can be conditional on the value of some other field(s). For example, the average value of real estate allocation percentage is 0.3% for multi-employer plans and 0.8% for single-employer plans.
- Fill the missing data with the value used by the most similar records. K-nearest neighbors may be used, and the missing data is replaced with the average of the k-nearest neighbors. The similarity may be measured by Manhattan distance, Euclidean distance, Cosine similarity, and so on.
- Fill the missing data with a predicted value. For example, a linear regression model may be used to estimate the value of liability discount rate based on asset mix and capital market data. The model parameters can be calibrated using data records without the missing data. If the model performance is satisfactory, the calibrated linear model can be used to estimate the missing data.

### A.2.2 DATA NORMALIZATION

When explanatory variables have different levels of magnitude, they may need to be normalized so that the parameter calibration will not be dominated by a small portion of the variables, and therefore better reflect the relationship between response variable and explanatory variables. Table A.4 lists some common normalization methods with discussions on their appropriate usage. In practice, trial and error is likely to be needed to choose the best method based on specific data and predictive model.

**Table A.4**  
**DATA NORMALIZATION METHODS**

Method	Definition	Usage
Min-Max Scaling	$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$ <p>Where  <math>X</math>: original value of X variable  <math>X'</math>: value after scaling  <math>X_{min}</math>: minimum value of X variable  <math>X_{max}</math>: maximum value of X variable</p>	When the data does not have many outliers and is approximately uniformly distributed, min-max scaling can be used to transform the value range to [0,1].
Decimal Scaling	$X' = \frac{X}{10^j}$ <p>Where  <math>j</math>: smallest integer such that <math> X' _{max} &lt; 1</math></p>	Similar to min-max scaling, decimal scaling transforms the variable to a value range of (-1,1).
Log Scaling	$X' = \log(X)$ <p>Where  <math>\log(X)</math>: natural logarithm of original value</p>	Used in cases where the original X variable has a long tail (a few extreme cases). $\log(X)$ will dampen the impact of extreme outliers and make the variable behave similar to a Normal distribution, which may be a desired property.
Standardized Scaling	$X' = \frac{X - \mu_x}{\sigma_x}$ <p>Where  <math>\mu_x</math>: mean of X variable  <math>\sigma_x</math>: standard deviation of X variable</p>	Most commonly used method and a reasonable choice for cases with and without outliers.
Truncating	$X' = \begin{cases} T_{min} & X \leq T_{min} \\ X & T_{min} < X < T_{max} \\ T_{max} & X \geq T_{max} \end{cases}$ <p>Where  <math>T_{min}</math>: threshold for left truncation  <math>T_{max}</math>: threshold for right truncation</p>	Even with a standardized scaling, extreme cases may still cause some undesired impact. Truncating method is able to truncate these extreme cases

### A.2.3 FEATURE ENGINEERING

In many situations, new explanatory variables are created based on existing explanatory variables and used for predictive analytics. One type of feature engineering is to transform categorical variables to dummy variables. For categorical variables such as occupation, even though numerical values may be used to represent categories, they need to be converted to dummy variables based on distinctive categories. For example, if there are five different occupations in the dataset, four dummy variables can be created as shown in Table A.5. Newly created dummy variables, rather than the original categorical variable, are expected to be fed into the predictive models.



**Table A.5**  
**DUMMY VARIABLE EXAMPLE**

Occupation	Occupation_1	Occupation_2	Occupation_3	Occupation_4
Category 1	1	0	0	0
Category 2	0	1	0	0
Category 3	0	0	1	0
Category 4	0	0	0	1
Category 5	0	0	0	0

Sometimes, a variable may be used as either a numerical variable or a categorical variable, depending on the response variable. For example, age may be used as a numerical variable if the response variable has a monotonic relationship with it. When predicting a life event such as education and marriage, age may be transformed to dummy variables each of which represents an age range.

Another type of feature engineering is to create new variables to reflect nonlinear relationships. The transformation can happen to both response variables and explanatory variables. Generalized linear models (GLMs), which will be explained in [Section A.3](#), are a generalization of linear models through a link function that transforms the response variable. New explanatory variables are valuable for linear regression models that try to capture nonlinear relationships. For example, new variables  $X_1^2$ ,  $X_1^3$ ,  $\log(X_1)$ ,  $X_1X_2$ , and  $X_1/X_2$  may be created based on variables  $X_1$  and  $X_2$ . However, this type of feature engineering may not be necessary for tree-based models that reflect nonlinear relationships directly, or artificial neural networks that have little restriction on the type of relationships it can approximately represent.

#### A.2.4 DIMENSIONALITY REDUCTION

Contrary to feature engineering, dimensionality reduction aims to reduce the number of explanatory variables to help the prediction. The curse of dimensionality refers to the issue when analyzing the data in high-dimensional spaces, the data becomes sparse, and the low density leads to statistical insignificance. It is also a cause of overfitting discussed in [Section A.4](#). It can be addressed by several approaches.

For a variable with a constant value for all data records, it can be removed as it does not have any predicting power.

For a pair of explanatory variables that have an extremely high correlation, either positive or negative, one of the pair can be removed from the analysis. This is often known as collinearity analysis in linear regression and is aimed at improving the robustness of model parameters. The absolute value of the correlation coefficient of each pair of explanatory variables can be compared against a threshold to determine whether a variable needs to be dropped.

Linear transformations such as principal component analysis (PCA) can also be used to reduce the dimension. By projecting each data record onto principal components that are perpendicular to each other, the first few principal components that contain the majority volatility in the data can be kept for predictive analytics. Principal components are derived by using the correlation matrix of explanatory variables and calculating the eigenvalues and eigenvectors. Eigenvectors with the highest corresponding eigenvalues are selected as the first few principal components. Using the 2019 Schedule H data as an example, Figure A.7 shows the cumulative portion of variance explained by principal components. The first five principal components explained 94.5% of the total volatility in the dataset. Instead of using 10 asset classes, five principal components may be used instead as inputs for model training.

**Figure A.7**  
SAMPLE PCA EXPLAINED VARIANCE

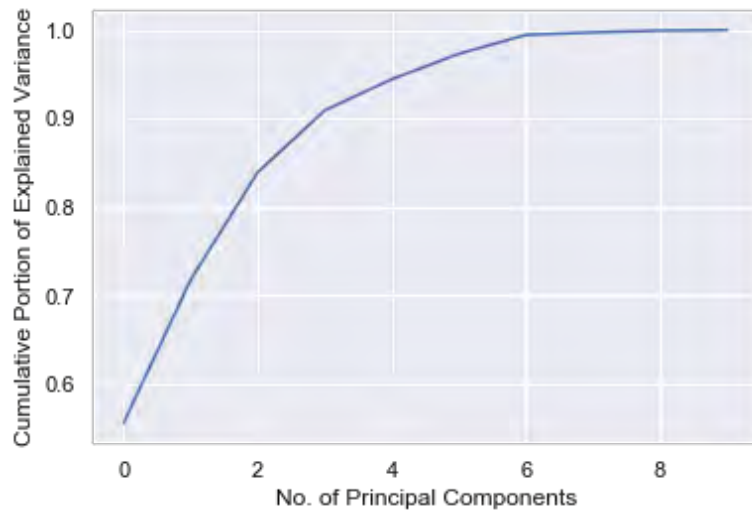
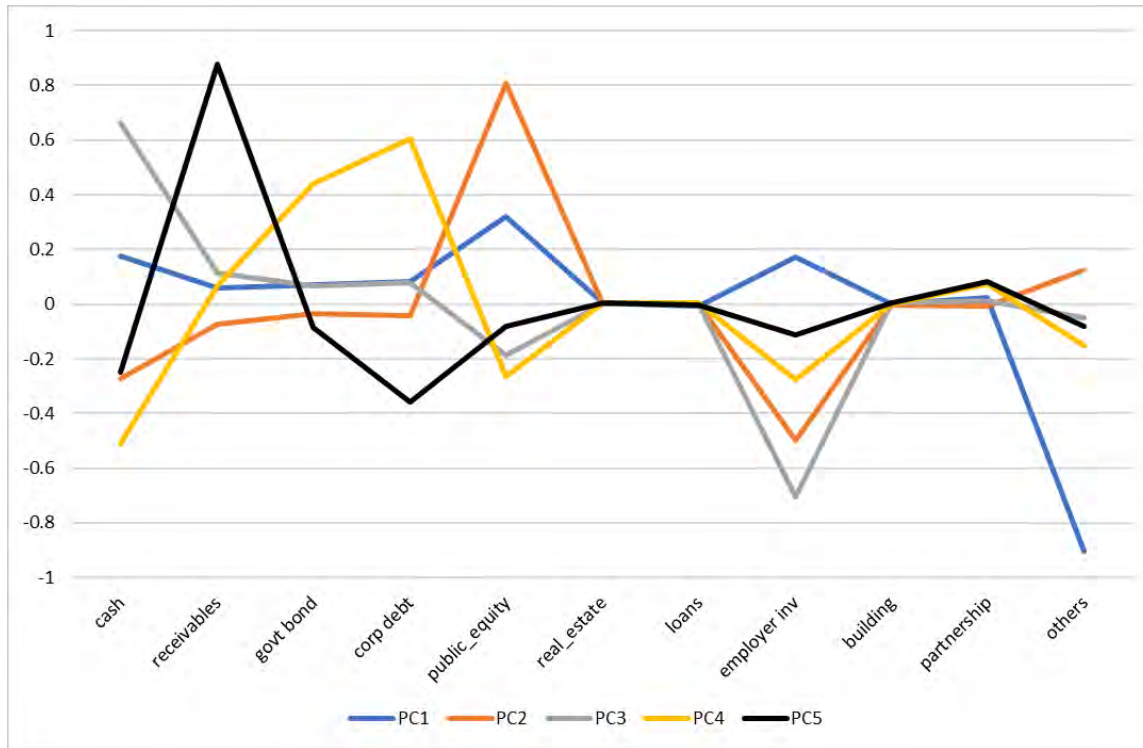


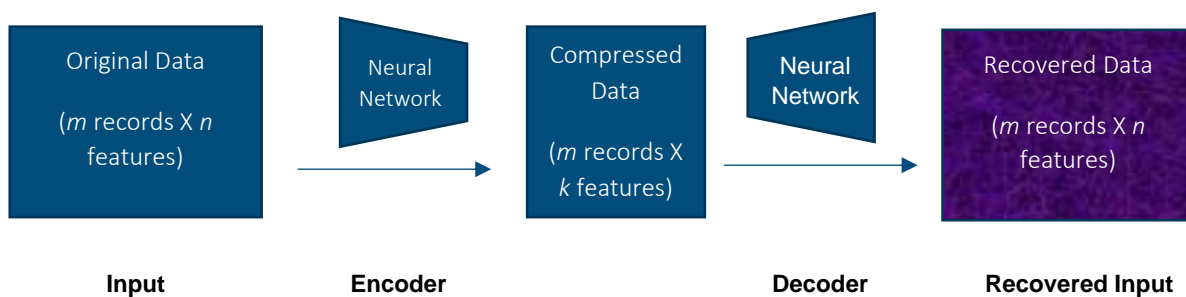
Figure A.8 shows the first five principal components (PCs), with PC1 representing an increase in equity and employer related investment and a drop in other investments, PC2 represents a spike in equity investment, PC3 represents an increase in cash holding and decrease in equity and employer related investment, PC4 represents a reduction of cash and an increase in bond allocations, and PC5 represents a spike in receivables and a reduction of corporate debt instruments. Each principal component is a vector containing shocks to weights on each asset class, represented by a line in Figure A.8.

**Figure A.8**  
SAMPLE PRINCIPAL COMPONENTS



Transformations that allow nonlinear operations can also be used to reduce the dimension. Neural network-based models such as the autoencoder is a popular choice. An autoencoder is composed of an encoder that is used to compress data, and a decoder that is used to recover data. Figure 9 shows the structure of an encoder. The neural networks as used in the autoencoder are explained in [Section A.2.3](#). The original data with  $n$  explanatory variables is compressed to  $k$  new explanatory variables using this encoder. Both the encoder and decoder need to be calibrated at the same time to minimize the difference between original data and recovered data. Similar to the PCA, autoencoder allows the compressed data to capture the majority volatility of the data. Unlike the PCA, autoencoder enables nonlinear operation but at the same time has a higher cost of model training.

**Figure 9**  
AUTOENCODER MODEL STRUCTURE



### A.3 PREDICTIVE MODEL

Three types of models are used in predictive analytics: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is used to learn the relationships between the response variable and explanatory variables. Linear regression is a supervised learning model. Unsupervised learning is to learn the patterns and relationships among explanatory variables, without any knowledge of the response variables. The PCA and autoencoder used for dimensionality reduction are examples of unsupervised learning. Reinforcement learning is related to dynamic decision-making. It requires trial and error to actively learn from experiments that generate training data. Together with the EDA explained in [Section A.1](#), different learning methods can be categorized based on two criteria: whether the response variable is used and whether data is used in a fixed way or an interactive way, as shown in Table A.5.

**Table A.5**  
**LEARNING METHOD CLASSIFICATION**

		Response variable	
		Known/Used	Unknown/Unused
Data Input	Fixed	Supervised Learning	Unsupervised Learning
	Interactive	Reinforcement Learning	Exploratory Data Analysis

The remainder of this section will introduce supervised learning, unsupervised learning and reinforcement learning with specific models. The intention is to introduce the available models and provide some insights into their usages.

#### A.3.1 SUPERVISED LEARNING

Supervised learning can be applied to two different types of problems. Regression analysis is used to predict the value of a response variable such as the fund surplus next year. Classification analysis is used to predict the probability that a variable is true such as whether a pension fund will become underfunded next year. Some models can be used for both regression and classification purposes with minor adjustments while others are more suitable for one of them. If a model has an output range of [0,1], it is probably more suitable for classification. A variety of models are introduced below but the focus is to describe different features. By no means does it covers all models and all the model details. The reader is encouraged to further read the resources in the references section of this paper or those noted in [Section 3](#).

**Linear regression** is the simplest yet powerful parametric model. It assumes a linear relationship between explanatory variables and response variable.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

Model parameters can be estimated by minimizing the squared errors. The simple linear regression also has many variants including Lasso, Ridge regression and Elastic Net with different methods of regularization to prevent overfitting, as discussed in [Section A.4.2](#). In addition to minimizing the squared errors, Lasso models add the sum of the absolute value of parameters into the error function. Ridge regression uses the sum of squared parameters as the regularization term, and Elastic Net models use both.

In addition to linear regression, **generalized linear models (GLMs)** are also widely used in the actuarial field. The GLM is a generalization of linear regression. It can be transformed to linear functions and allows different error distributions and value ranges of the response variable. A GLM contains a linear predictor, similar to linear regression.

$$\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

It then uses a link function  $g$  to link the response variable  $Y$  with  $\eta$ .

$$E(Y|X) = \mu = g^{-1}(\eta) = g^{-1}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)$$

Linear regression is a special case of GLM where the Gaussian distribution is assumed and an identity link function ( $\eta = \mu$ ) is used. Logistic regression estimates the probability of the response variable based on the logistic function given below. It has a link function  $\eta = \ln\left(\frac{\mu}{1-\mu}\right)$

$$E(Y|X) = \mu = \frac{1}{1 + e^{-\eta}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Table A.6 provides a list of popular GLMs with different distribution types, link functions, and value ranges for the response variable. Given the empirical distribution of the response variable, the appropriate link function and distribution type can be chosen to specify the GLM.

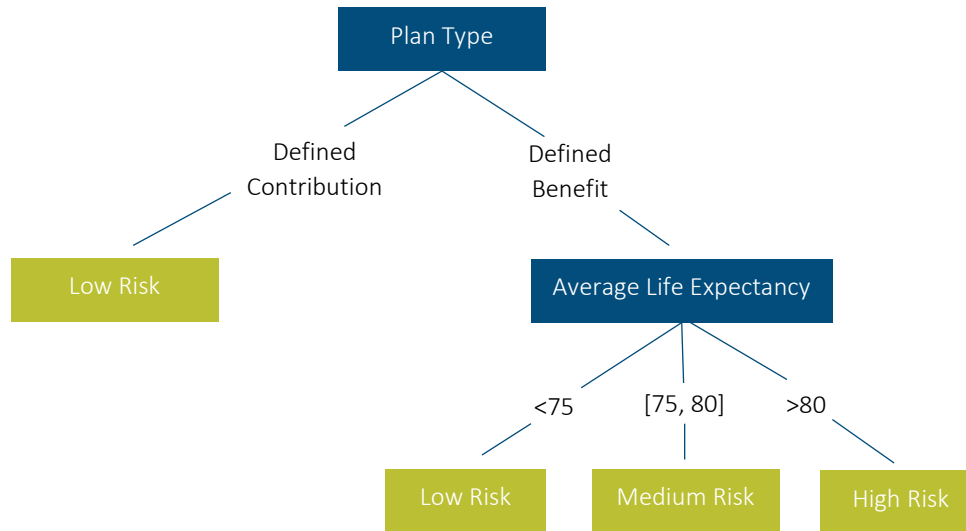
**Table A.6**  
GLM EXAMPLES

Link Function ( $g$ )	Inverse Link Function ( $g^{-1}$ )	Distribution Type	Value Range of Response variable	Alias
$\eta = \mu$	$\mu = \eta$	Gaussian	$(-\infty, \infty)$	Linear regression
$\eta = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{1}{1 + e^{-\eta}}$	Bernoulli	$\{0,1\}$	Logistic regression
$\eta = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{1}{1 + e^{-\eta}}$	Multinomial	A vector of K elements each element belonging to $[0, N]$ and the sum of all elements equals N.	Multinomial Logistic regression
$\eta = -\frac{1}{\mu}$	$\mu = -\frac{1}{\eta}$	Exponential/Gamma	$(0, \infty)$	Negative inverse
$\eta = \frac{1}{\mu^2}$	$\mu = \frac{1}{\sqrt{\eta}}$	Inverse Gaussian	$(0, \infty)$	Inverse Squared
$\eta = \ln(\mu)$	$\mu = e^\eta$	Poisson	Non-negative integers	Poisson Regression

GLM allows the relaxation of the Gaussian distribution assumption and can accommodate different value ranges of the response variable. Unlike linear regression whose model parameters can be solved using closed form formulas, the parameters of these models are normally solved using maximum likelihood estimation (MLE). This method maximizes the likelihood the data is observed based on the fitted parameters.

Unlike GLM, **tree-based models** switch from formulas to decision rules for prediction. In a tree, leaves represent different subgroups and branches represent the rules to split into subgroups based on explanatory variables. The prediction is based on the value of the leaves that are in the same subgroup. Figure A.10 shows an example using a tree-based model to determine plan sponsors' exposure to longevity risk. The rules and conclusions in this example are straightforward and may not need any data to support them. For a tree-based model where the rules are learnt from data, it becomes more complicated.

Figure A.10  
SAMPLE TREE-BASED MODEL



**Classification and Regression Tree (CART)** models are a basic form of tree-based models. CART models build trees to split the data based on explanatory variables. At each split, a variable is used to separate the data into two subgroups. The variable is chosen to provide the best split that improves the purity of the data in the subgroups. The Gini index is commonly used to represent the data dispersion. It is calculated as follows.

$$G(T) = \sum_{i=1}^n p_i(1 - p_i) = 1 - \sum_{i=1}^n p_i^2$$

Where

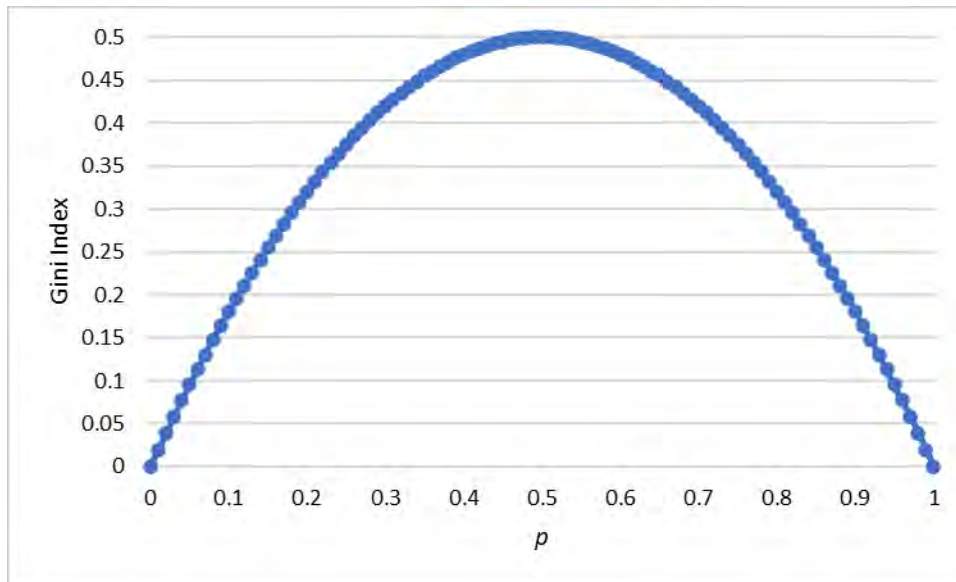
$p_i$ : the probability that the data belongs to category  $i$ .

$n$ : the number of categories in the data.

$T$ : the dataset based on which Gini index is calculated.

If the data is pure, meaning that it only has one value, the Gini index is zero. If the data is evenly dispersed, such as 50% probability for each of two possible values, the Gini index is 0.5. Figure A.11 shows the Gini index curve for data with only two categories. The Gini index reaches the maximum when the probabilities are even between two categories.

**Figure A.11**  
GINI INDEX CURVE



At each split, the increase in data purity in subsets is maximized when choosing the variable and the threshold for splitting.

$$\max_{x, \text{threshold}} G(T) - p(T_L)G(T_L) - p(T_R)G(T_R)$$

Where

$T_L$ : the data subgroup of the split's left branch.

$T_R$ : the data subgroup of the split's right branch.

$p$ : the portion of the data subgroup in the dataset before splitting.

$x$ : the variable to be used for the splitting.

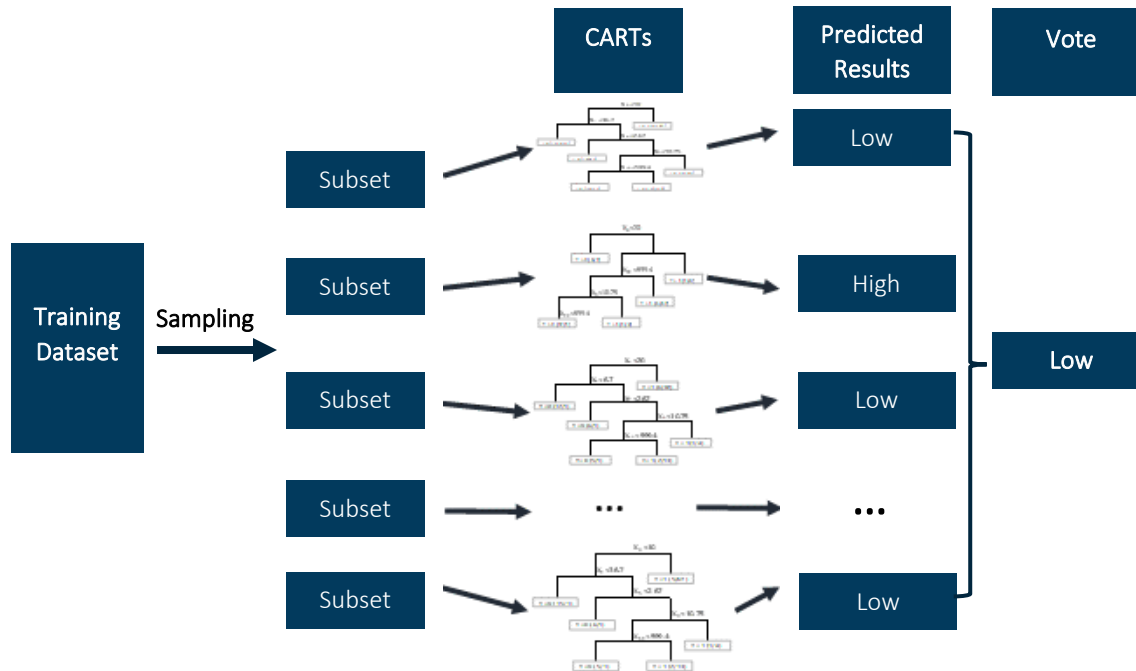
*threshold*: the threshold used to set the split based on the value of  $x$ .

Assuming that the data is evenly dispersed with 50% probability for each of the two categories, the Gini index  $G(T)$  before splitting is 0.5. If the split divides the data perfectly into the two categories, then the new Gini index is zero, as calculated below. The gain from the split is 0.5 at its maximum.

$$p(T_L)G(T_L) + p(T_R)G(T_R) = 0.5 \times 0 + 0.5 \times 0 = 0$$

More advanced tree-based models are built upon CART. The famous **Random Forests models** are a random version of the CART models. Multiple subsets are sampled from the training dataset and each subset is used to build a CART model. Explanatory variables are sampled as well so that the relationship between the response variable and the explanatory variables will not be dominated by the most important ones. Less important explanatory variables can contribute to the final prediction as well. Figure A.12 illustrates the structure of the Random Forests models used in this report. The final prediction is calculated as the average prediction by individual CART models.

Figure A.12  
RANDOM FORESTS MODEL STRUCTURE

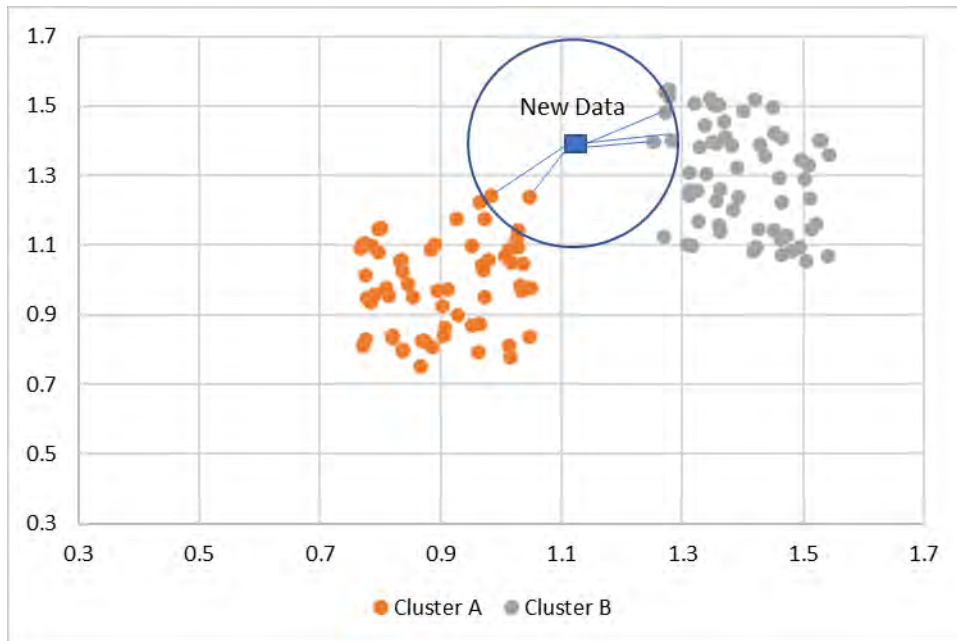


Gradient boosting machine (GBM) is another decision tree–based ensemble method. Each tree is a weak estimator trying to estimate the residual error that the estimation of previous trees has caused. Gradually with a sufficient number of decision trees, the estimation error will decline to a very low level. Unlike Random Forests models which use parallel trees to predict in aggregate (a concept known as “bagging”), GBM is a sequential tree model. GBM is usually proven to have better accuracy than many other methods when presented with nonlinear relationships. The model is also faster to train compared to artificial neural networks (ANNs).

As a nonparametric model, a ***k*-nearest neighbors (KNN)** model predicts the response variable based on the values of the *k* closest neighbors. The closeness can be measured by the Euclidean distance, Manhattan distance, cosine similarity, and so on. When predicting the response variable, whether it is a regression or classification problem, the value is determined based on the value of its *k* nearest neighbors. It can be an arithmetic average or a weighted average with the weight depending on the distance. Figure A.13 illustrates the concept of a KNN model with *k* equals 2.

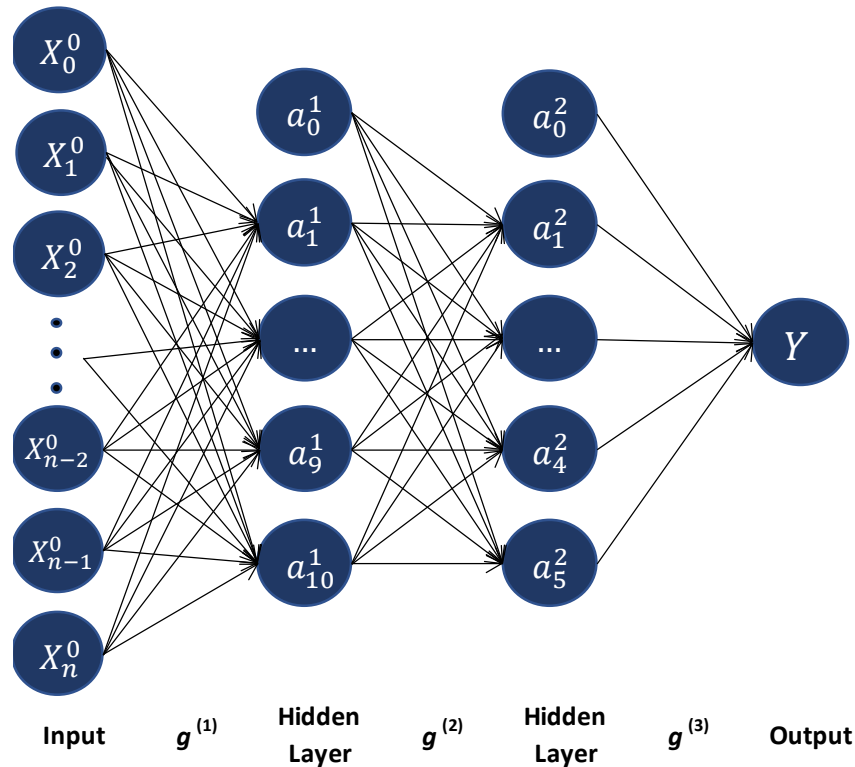


Figure A.13  
KNN ILLUSTRATION



**Artificial neural network (ANN) models** mimic biological neural networks to make predictions based on a large amount of data. Unlike traditional predictive models such as linear regression and logistic regression, the mathematical function that describes the relationship between the response variable and explanatory variables is unknown. Rather, it uses multiple layers of linear, logistic or other simple functions to allow many more possible relationships. With enough data and appropriate training, ANN models are believed to mimic any complex relationships. Figure A.14 shows a simple ANN model with input, output, and two hidden layers.

Figure A.14  
ANN MODEL STRUCTURE



The layers are linked together with activation functions  $g$ . Each neuron in the hidden layers and output layer is determined by the neurons in the previous layer.

*First hidden layer:*  $a_0^1 = 0$  and  $a_j^1 = g(\theta_j^0 \times x^0)$  for  $j > 0$

$x^0$ : an  $(n+1)$  element column vector containing all the explanatory variables and the intercept  $x_0^0$ .

$\theta_j^0$ : an  $(n+1)$  element row vector containing the weights applied to all explanatory variables and the intercept to determine the value of neuron  $a_j^1$ .

*Second hidden layer:*  $a_0^2 = 0$  and  $a_j^2 = g(\theta_j^1 \times a^1)$  for  $j > 0$

$a^1$ : an 11-element column vector containing all the neurons in the first layer.

$\theta_j^1$ : an 11-element row vector containing the weights applied to all the neurons in the first layer to determine the value of neuron  $a_j^2$ .

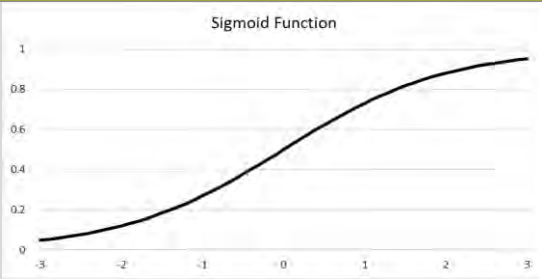
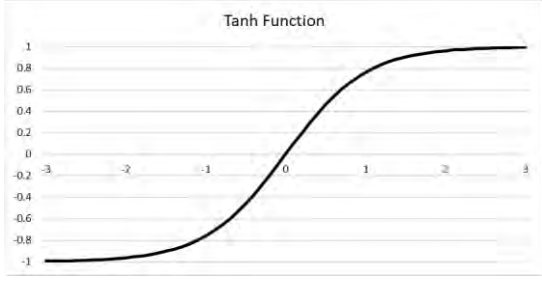
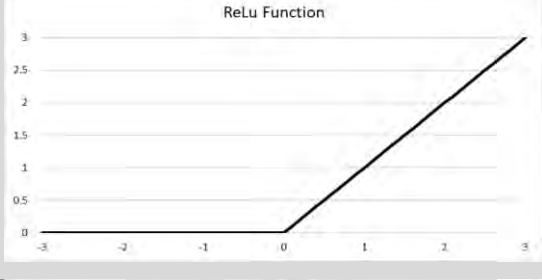
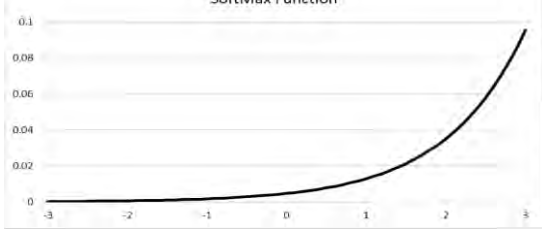
*Output layer:*  $Y = g(\theta^2 \times a^2)$

$a^2$ : a 6-element column vector containing all the neurons in the second layer.

$\theta^2$ : a 6-element row vector containing the weights applied to all the neurons in the second layer to determine the value of the output variable  $Y$ .

A neuron's value  $a_j^i$  depends on a linear combination of the neurons in the previous layer ( $\theta_j^{i-1} \times a^{i-1}$ ). Depending on its value, the next step is to determine whether the neuron should be activated or not, like the way our brains work. Activation function  $g$  can be considered as a mechanism to bring the range down to a manageable level and bring in nonlinear relationships. Table 7 lists four common activation functions.

Table A.7  
ACTIVATION FUNCTIONS

Name	Function	Output Range	Plot
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	(0,1)	
Tanh	$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$	(-1,1)	
ReLu	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$[0, \infty)$	
SoftMax	$f(x_j) = \frac{e^{x_j}}{\sum_{i=1}^N e^{x_i}}$ for $j = 1, 2, \dots, N$ $\sum_{i=1}^N f(x_i) = 1$	[0,1]	

The choice of activation function can be arbitrary and by trial and error. The output range can be used as a general criterion to narrow down the choices. When the output is a probability, the sigmoid function is a natural choice, as used in Logistic regression. The ReLu function is less smooth than some other activation functions but works well in practice using a large-scale neural network.

ANNs belong to the family of deep learning. **Recurrent neural networks (RNNs)** and **convolutional neural networks (CNNs)** are two other popular types of deep learning models. Unlike ANNs where the connection among neurons and layers are forward, RNNs allow recurrent connections in the hidden layers. When analyzing a pension plan's performance, it may be helpful to utilize all available historical data, which is difficult to realize using the feedforward neural networks as described in Figure A.14. The recurrent connections can be considered as a time

series analysis where past states affect the future states of the neurons in a hidden layer. RNN adds the dimension of time in the model.

CNN brings the spatial dimension into the picture. The model extracts features from the input data by analyzing the data block by block. It is widely used in image recognition. Instead of looking at the entire image at the same time, like what an ANN does, CNN looks at a small area (receptive field) at each time. This is similar to the human way of searching for a small object in a big picture. The idea of using small receptive fields is no stranger in actuarial analysis. For example, when setting the rate for auto insurance policies, location is an important pricing factor. Even though national data may be available, the most relevant information is from local experience data based on zip code, city, or even geolocation. Even though CNNs are intended for image data analysis, they can be used for sequential data as well. An example would be using transaction records to analyse customer behavior. Instead of using all transactions as explanatory variables all at once, using 10 consecutive transactions as a group and scan the entire record with all possible groups as inputs to the neural network model, CNN considers the spatial dimension which is the implication from consecutive transactions that are not explicitly considered in ANN or RNN.

### A.3.2 UNSUPERVISED LEARNING

The PCA and autoencoder mentioned in [Section A.2.4](#) are unsupervised learning. Other model types are available as well, including but not limited to clustering, association rules, and Bayesian network.

**Clustering** is one of the most popular model types in unsupervised learning. It categorizes data based on similarity. Similarity is usually measured by Euclidean distance but other measures such as cosine similarity may be used as well. Three types of clustering methods are introduced below: centroid models, connectivity models, and density models.

**Figure A.15**  
K-MEANS EXAMPLE

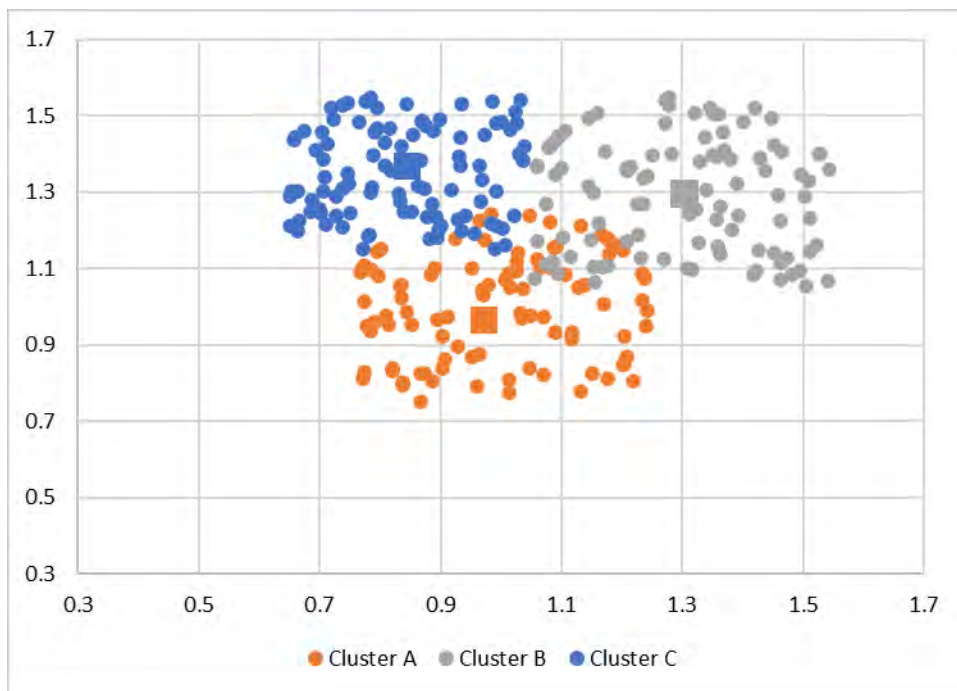


Figure A.15 shows an example of k-means, a centroid clustering method. K-means partitions all data records into three groups based on similarity which is measured by Euclidean distance. The rectangles represent the centers of

the three groups. A center in k-means is the average of all data points in that group. A variant of k-means is k-medoids where the center of a group is an actual data point rather than the average value. K-medoids may be helpful for categorical variables when the average value is not meaningful.

**Figure A.16**  
HIERARCHICAL CLUSTERING EXAMPLE

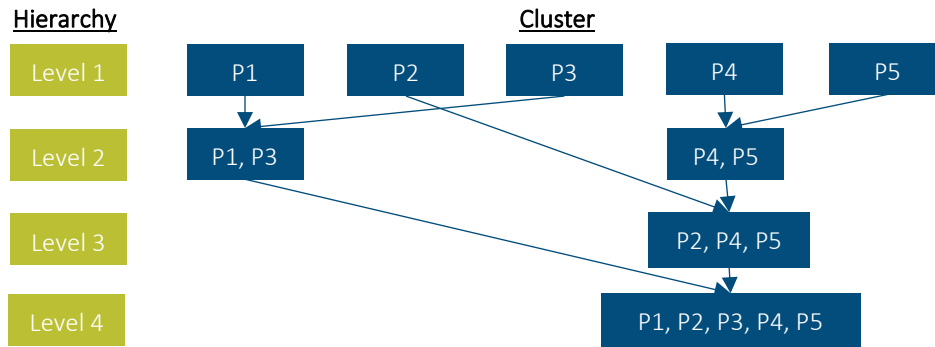
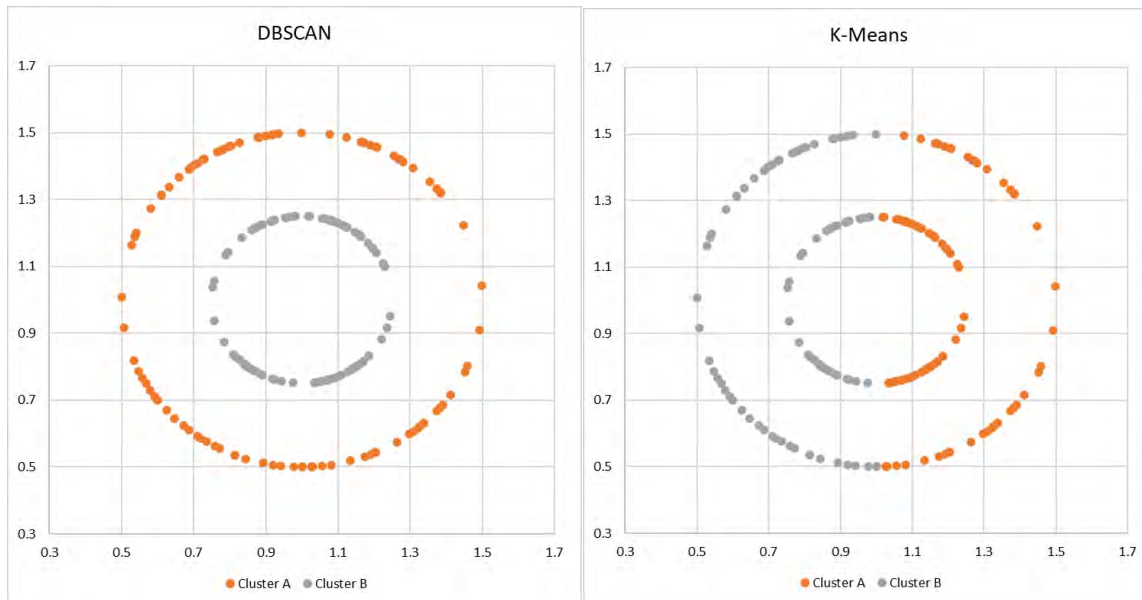


Figure A.16 shows an example of hierarchical clustering, a connectivity clustering model. Hierarchical clustering provides different levels of clustering. At the lowest level, every data point is a cluster by itself. It then moves up by combining data points based on similarity. At the highest level, all data points belong to one single cluster. Hierarchical clustering can provide different clustering results based on the desired number of clusters.

Both centroid and connectivity methods are good at identifying clusters that are well separated in a spherical shape. However, data may have outliers and different shapes that similarity itself is not sufficient. Density models can be used to address these issues. DBSCAN (Density Based Spatial Clustering of Applications with Noise) is an example of density models that not only requires similarity, but also minimum number of neighbors within a specified distance.

Figure A.17 compares DBSCAN and K-means when dealing with data that has high density on two circles. DBSCAN is able to take into consideration the zero density among the two circles and group data appropriately.

**Figure A.17**  
DBSCAN V.S. K-MEANS



Clustering helps provide a high-level picture of data variability in terms of category. They can be used to compress data and use cluster centers for further analysis. Sometimes the clusters exhibit different behaviors that are useful for risk rating and pricing.

**Association rule learning** aims to identifying strong association among explanatory variables in a dataset, rather than grouping data as in clustering analysis. By analyzing data records, it can find what combination of features is most frequently observed in the dataset. A popular example is analyzing shopping lists and identifying what items are usually shopped together. This technique can be helpful for pension plan analysis. For example, by analyzing investment choices of DC plan participants, interesting relationships may be identified regarding fund selection and will be useful for deciding the offering of funds.

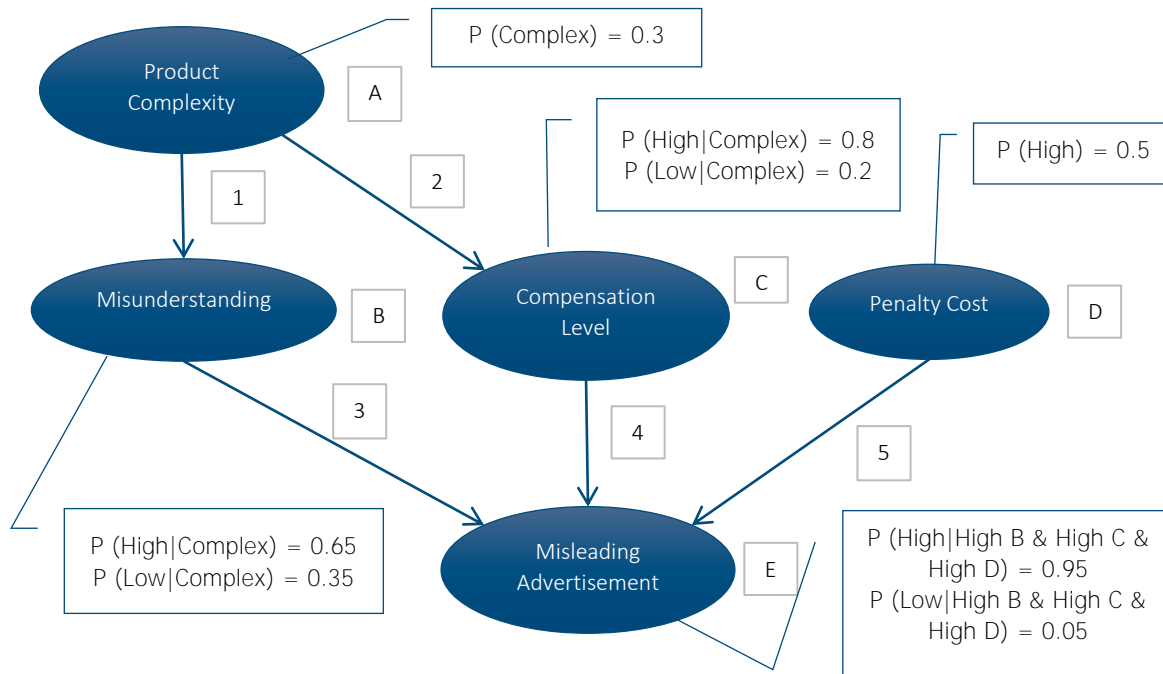
But how can the popularity be measured in a mathematical way? Support, confidence and lift are three common measures to use. Support measures how frequently the feature or combination of features appear in the data and is calculated as the number of observations of the feature divided by the total number of data records. Confidence measures the chance that the relationship is true in the data. Lift is the ratio of the support of the combination and the expected support assuming an independent relationship. A lift greater than 1 indicates the dependence is stronger than an independent relationship. Table A.8 shows a simple example of calculating the three measures. A lift of 14/15 means that the chance of having A and B together is not that strong. By setting a threshold of support and lift, credible relationships can be identified and evaluated for applications.

**Table A.8**  
ASSOCIATION RULE EXAMPLE

Data		Support	Confidence	Lift
ID	Items	Support(A) = 6/7 Support(B) = 5/7 Support(A, B) = 4/7	Confidence(A=>B) = Support(A, B)/Support(A) = 2/3	Support(A, B) Support(A)Support(B) = 14/15
1	A, B			
2	A		Confidence(B=>A) = Support(A, B)/Support(B) = 4/5	
3	A			
4	B			
5	A, B			
6	A, B			
7	A, B			

**Bayesian Network** is another approach to identify relationships in the data. It is a probabilistic graph that models the known conditional dependency among variables represented by directed edges in the graph of variables. Variables without connected edges are assumed to be independent. Figure A.19 shows an example of using Bayesian network to evaluate the chance of having a misleading advertisement. There are five nodes (A, B, C, D, and E) and five directed edges (1, 2, 3, 4, and 5) in the graph. For each node, the probability, either conditional or unconditional, can be learned from the data. As in all Bayesian models, prior knowledge plays a role. For Bayesian network, the required prior knowledge is the edges, either directed or undirected. These edges can be set up based on experience and common sense.

**Figure A.19**  
BAYESIAN NETWORK EXAMPLE



The advantage of a Bayesian network is that questions such as “if A is high, what is the chance that E is high” are answered through the joint probability specified in the Bayesian network.

### A.3.3 REINFORCEMENT LEARNING

Reinforcement learning seeks to make optimal sequential decisions under uncertainty. A famous example is AlphaGo Zero that uses reinforcement learning to study the ancient Chinese game of Go. It starts with basic rules of the game without data of human games. It is an improved version of AlphaGo that defeated a world champion of the Chinese Go game. Although reinforcement learning is not a straightforward predictive analytics application, predictive models such as ANNs can be used to represent the reward function in reinforcement learning. As the use of ANNs has increased, this has led to many significant advances in reinforcement learning.

Figure A.20

#### REINFORCEMENT LEARNING PROCESS FOR LIABILITY INVESTMENT STRATEGY

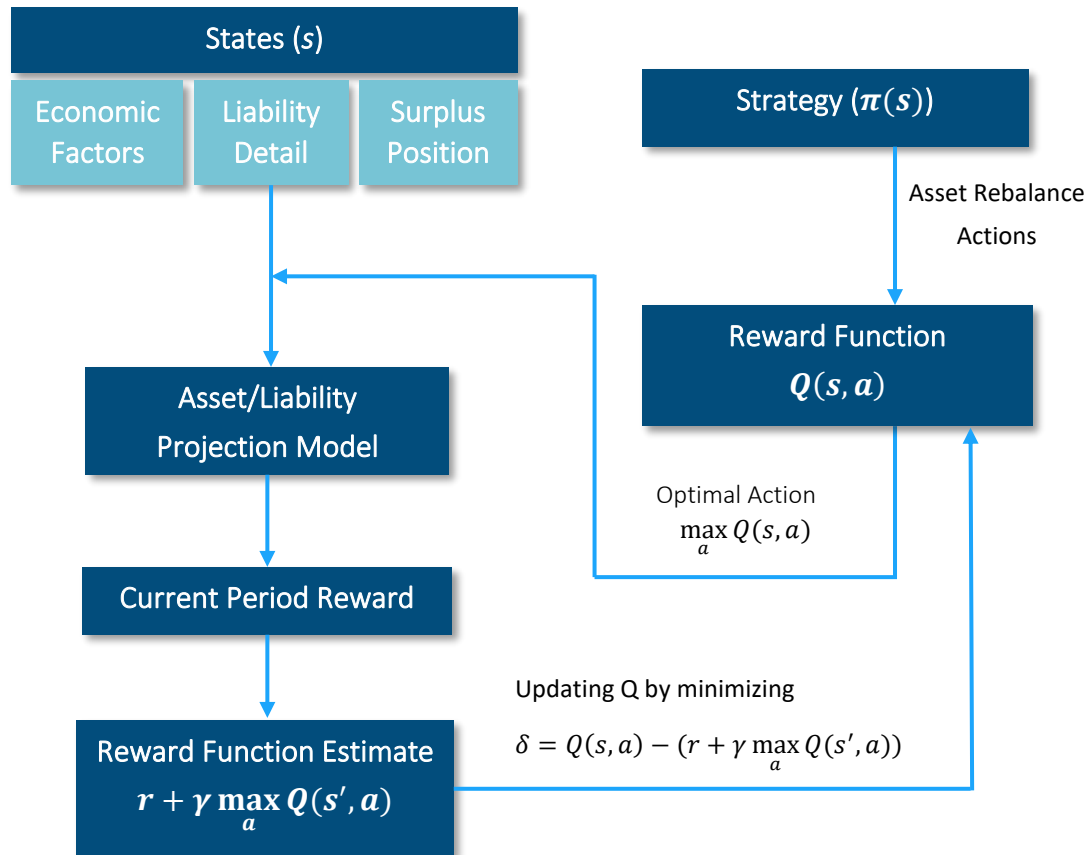


Figure A.20 shows an example of applying reinforcement learning to LDI strategy. The goal is to find the optimal dynamic investment strategy  $\pi^*(s)$  based on  $s$ , the states the decision-maker can observe at the time of decision-making. The states can be economic conditions, surplus position, and liability details. The optimality of the investment strategy is defined as the one that maximizes the reward function  $Q^*(s, a)$  determined by  $s$ , the states, and  $a$ , the rebalancing action determined by the strategy  $\pi^*(s)$  that maximizes the reward.

$$\pi^*(s) = \max_a Q^*(s, a)$$

The reward function  $Q^*(s, a)$  is difficult to define using a mathematical formula. The impact of an asset mix selection not only affects the current period performance but could also have long-lasting impact on the future surplus position. Instead of defining the reward function directly, it can be constructed in a recursive form.



$$Q^\pi(s, a) = r + \gamma Q^\pi(s', \pi(s'))$$

Where

$r$ : current period reward that can be observed. For example, it could be defined as changes in surplus amount or funding ratio.

$\gamma$ : discount factor to reflect timing difference.

$Q^\pi(s', \pi(s'))$ : reward function in the next period with new states  $s'$  and new asset rebalance action  $\pi(s')$ .

The reward function  $Q(s, a)$  is a nonlinear function that explains nonlinear relationships. In most cases, the exact function is not defined but learnt by minimizing the error  $\delta$  between the two sides of the recursive equation.

$$\delta = Q(s, a) - (r + \gamma \max_a Q(s', a'))$$

Using deep learning models such as ANNs and RNNs, the actual reward function may be approximated without the need of setting the exact function form before model training. In theory, it will work given enough data and training time. In rare cases, the reward function can be specified in advanced through supervised learning. An experimental environment is needed to generate different states so that the model can try different investment strategies and find the optimal strategy. The asset rebalance action that has the highest expected reward is chosen and used to determine its impact on current period performance. After trying more and more scenarios, the deep learning model that represents the reward function is updated and is expected to move closer to the real reward function. Unlike supervised and unsupervised learning, the training data is generated through experimenting.

## A.4 MODEL TRAINING

Model training is the process to calibrate model parameters based on training data. Before model training, the clean dataset needs to be split into training data and validation data to facilitate model assessment. During model training, only training data, i.e., “in-the-sample data”, is observable by the model. Validation data, i.e., “out-of-sample data”, is then used to evaluate model performance. A rule of thumb is to use an 80/20 split to randomly create the training dataset and the validation dataset, although more training data may be used in the presence of insufficient data.

With the availability of many open-source libraries, running predictive models is not a challenging task nowadays. However, to improve model accuracy and make sure the calibrated models are robust using validation data, it is important to select appropriate model training choices.

### A.4.1 ERROR FUNCTION

For supervised learning, reinforcement learning, and some unsupervised learning models such as autoencoder, an error function needs to be selected. The error is defined as the difference between actual value  $y_{actual}$  and predicted value  $y_{pred}$ .

- **Root-mean-squared error (RMSE)**: the square root of the mean of the square of all of the error.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^N (y_{pred,i} - y_{actual,i})^2}{N}}$$

Alternatively, mean square error (MSE) may be used to measure error as well.

- **Mean absolute error (MAE)**: the mean of the absolute value of all of the error.

$$MAE = \frac{\sum_{i=1}^N |y_{pred,i} - y_{actual,i}|}{N}$$

- **Weighted error measure:** error measures that assign different weights to different data records. This is useful when data records have different levels of importance. An example is weighted RMSE.

$$\text{Weighted RMSE} = \sqrt{\sum_{i=1}^N \frac{\text{weight}_i (y_{\text{pred},i} - y_{\text{actual},i})^2}{N}}$$

- **Huber loss:** a loss function that utilizes either squared error or absolute error depending on the magnitude of the error, as defined below. It is less sensitive to outliers than squared error loss.

$$\text{Huber Loss}_{\delta}(y_{\text{actual},i}, y_{\text{pred},i}) = \begin{cases} 0.5(y_{\text{pred},i} - y_{\text{actual},i})^2 & \text{for } |y_{\text{pred},i} - y_{\text{actual},i}| \leq \delta \\ \delta |y_{\text{pred},i} - y_{\text{actual},i}| - 0.5\delta^2 & \text{otherwise} \end{cases}$$

The average Huber loss of all data records can be used as the aggregate loss function.

- **Quantile loss:** a loss function that can be used to penalize either overestimation or underestimation.

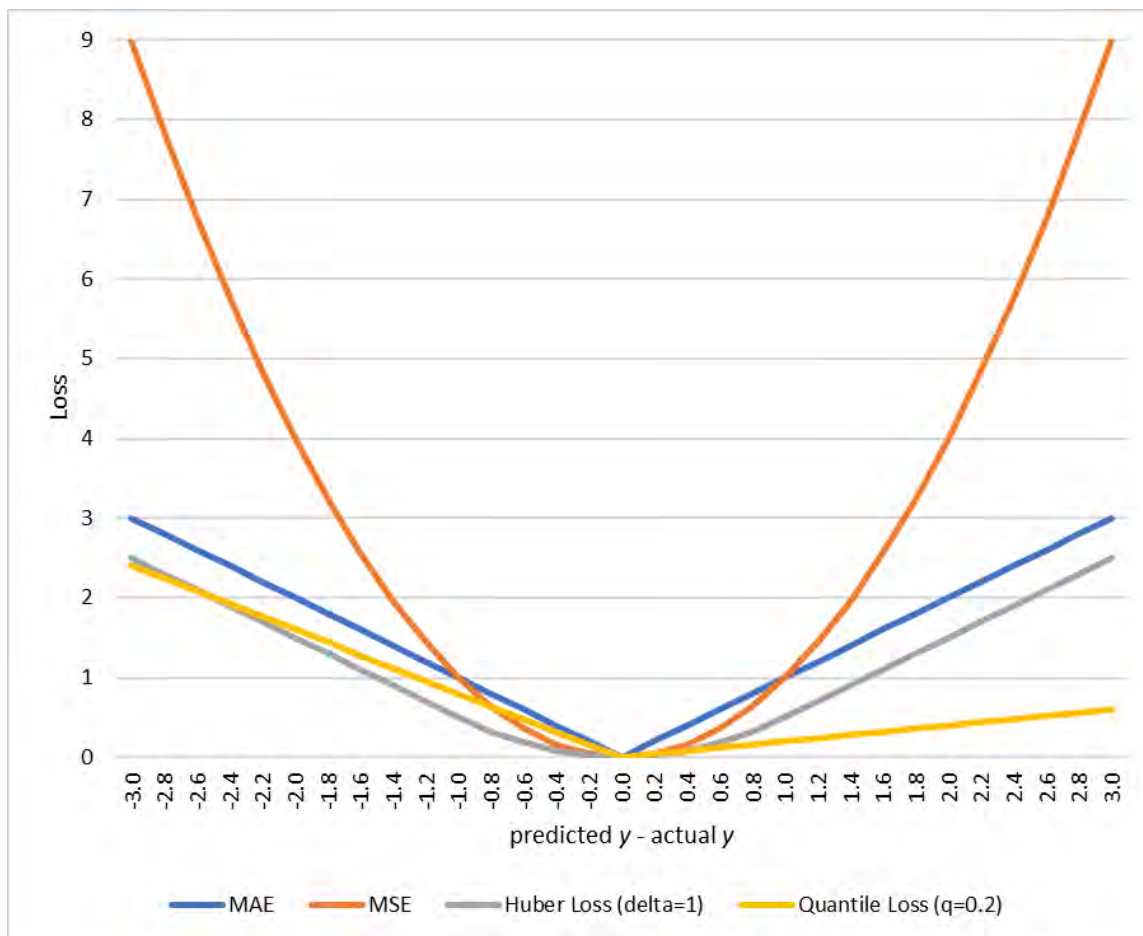
$$\text{Quantile Loss}_q(y_{\text{actual},i}, y_{\text{pred},i}) = \max\{q(y_{\text{pred},i} - y_{\text{actual},i}), (q - 1)(y_{\text{pred},i} - y_{\text{actual},i})\}$$

The average quantile loss of all data records can be used as the aggregate loss function.

Figure A.21 compares several error functions. MSE has the highest degree of penalizing outliers, followed by MAE and Huber loss ( $\delta=1$ ). Quantile loss with  $q=0.2$  penalizes underestimation more than overestimation. Error functions can be chosen depending on specific prediction tasks, considering the preferred treatment of outliers and overestimation/underestimation.

Figure A.21

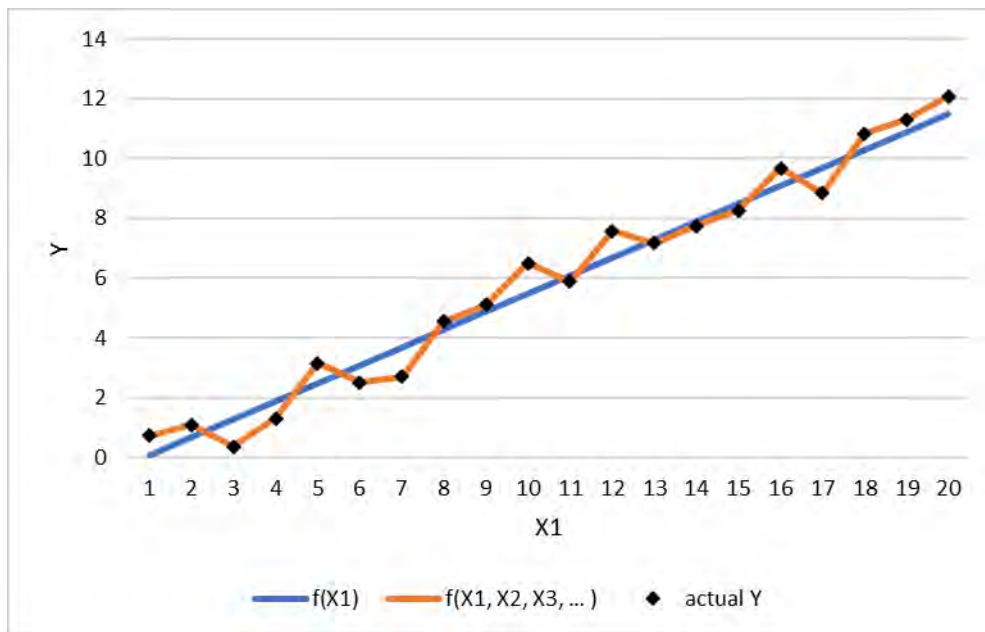
#### SAMPLE ERROR FUNCTIONS



### A.4.2 OVERFITTING

Another important issue to consider for model training is overfitting. When too many variables are unintentionally used to explain the random noises rather than the relationships, the model overfits the data and shows a very high accuracy of prediction with training data. However, a much lower prediction accuracy is usually observed using the validation data. Figure A.22 shows that a linear regression model with only one explanatory variable  $X_1$  can capture the main relationship even though the accuracy is lower than a perfect matching nonlinear model with much more explanatory variables. If we use the overfit model to make predictions it is likely going to underperform the less overfit linear regression model.

**Figure A.22**  
OVERFITTING EXAMPLE



Many methods can be used to address the issue of overfitting.

- **Dimensionality reduction.** As discussed in [Section A.2.4](#), dimensionality reduction helps encapsulate explanatory variables into a handful of principal components that capture the majority of the volatility in the data. With less inputs as explanatory variables, the chance of overfitting is smaller.
- **Variable selection.** Similar to dimensionality reduction, the number of explanatory variables can be reduced by selecting only important ones. As explained in Shang (2017), a few approaches can be used to select important variables by running models multiple times, as shown in Figure A.23. The forward approach starts from an empty model and adds one variable at a time. At each step, the variable with the biggest accuracy improvement is chosen. The forward process ends when the model accuracy stops improving or the improvement is trivial. The backward approach starts from a full model with all variables and removes one variable at a time. At each step, the variable with the biggest negative impact or the smallest positive impact is removed, until the model accuracy stops improving or reaches the desired level. For both the forward and backward approach, the sequence of the explanatory variables matters. The stepwise approach addresses this issue by combining the forward approach and the backward approach. At each step, an additional variable is added, and then the new model works backward to remove any existing variables that have a negative or trivial impact on model accuracy. Another more comprehensive yet costly approach is to iterate through all possible combinations of explanatory variables and choose the subset with the smallest set of variables given that the model accuracy meets the target.

**Figure A.23**  
VARIABLE SELECTION METHOD



- **Regularization.** By adding a penalty for model complexity into the error function, regularization can be used in many predictive models to mitigate the risk of overfitting. For example, ridge regression is a version of linear regression with regularization. Normal regularization includes L1 regularization, which uses the sum of the absolute value of parameters, as in the LASSO model, and L2 regularization, which uses the sum of the squared value of parameters, as in ridge regression.

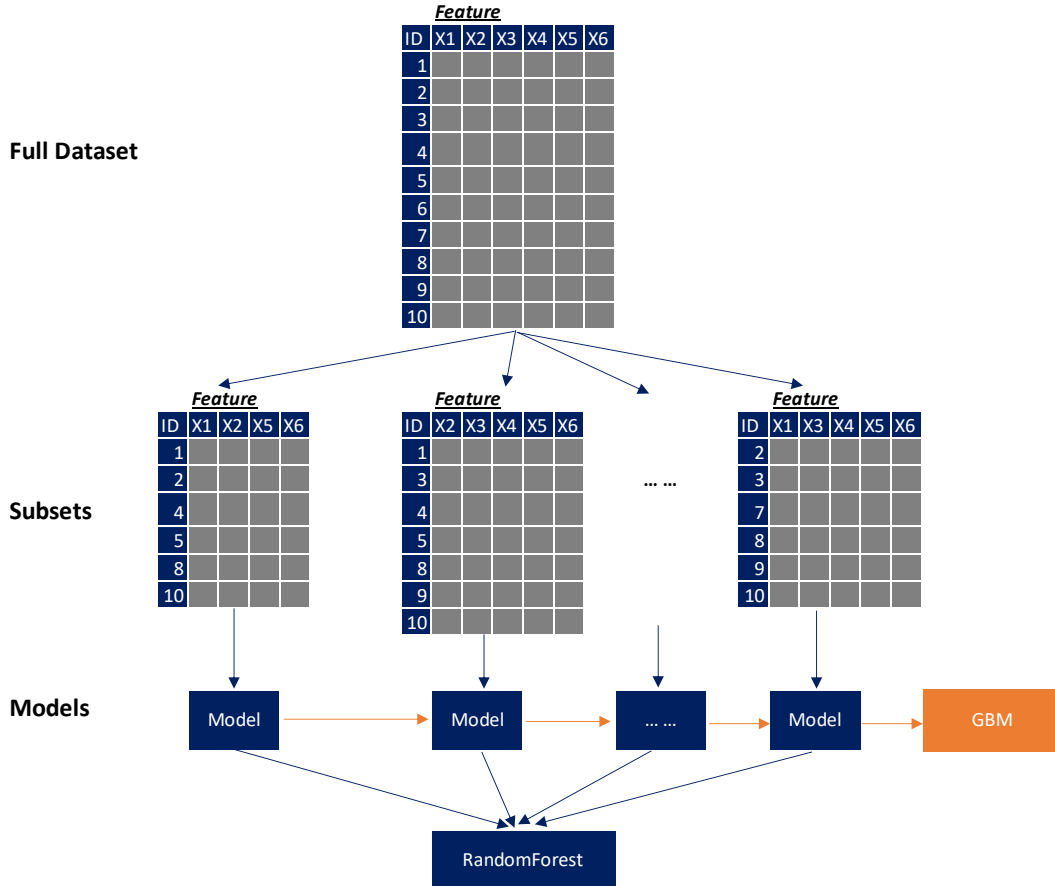
$$\text{Lasso Regression: } \min_{\beta} \sum_{j=1}^m (Y_j - \sum_{i=1}^n x_i^j \beta_i)^2 + \lambda \sum_{i=1}^n |\beta_i|$$

$$\text{Ridge Regression: } \min_{\beta} \sum_{j=1}^m (Y_j - \sum_{i=1}^n x_i^j \beta_i)^2 + \lambda \sum_{i=1}^n \beta_i^2$$

Parameter  $\lambda$  controls the weight of the penalty. Other models, such as GLM and ANN, can also include a regularization term in the optimization goal to address overfitting. For tree-based models, controlling the depth of the tree is also a way of applying regularization.

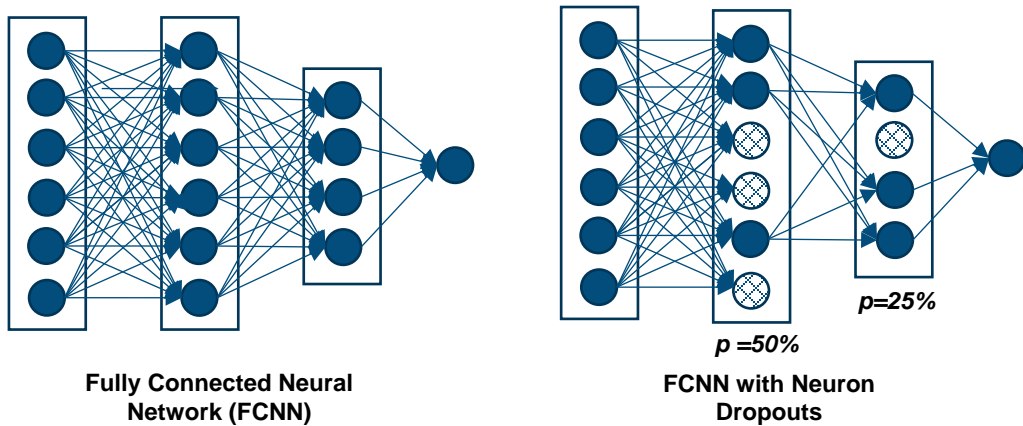
- **Random data subset.** Instead of using the entire training dataset at one time, random data subsets can be used several times during model training. By using different subsets, the calibrated model is unlikely to have perfect prediction for all training data but captures the patterns in the data. This is utilized in the Random Forests and GBM models but can be applied to other models as well.
- **Random feature subset.** Similar to using random data subsets, choosing random subsets of explanatory variables is also helpful for addressing overfitting. Random Forests and GBM models use both random data subsets and random feature subsets, as shown in Figure A.24.

**Figure A.24**  
RANDOM SUBSETS IN RANDOM FORESTS AND GBM



For ANNs, Srivastava et al. (2014) introduced the method of dropping a percentage of neurons in the network during model training. The dropped-out neurons are randomly chosen and their values are set to zero. The remaining neurons are scaled up by  $\frac{1}{1-p}$  where  $p$  is the percentage of neurons dropped out. Figure A.25 shows an example of neuron dropout with different percentages at different hidden layers.

Figure A.25  
NEURON DROPOUT



### A.4.3 OPTIMIZATION ALGORITHM

For linear regression using the ordinary least squares method, model parameters can be derived using a closed-form solution. However, many predictive models are nonlinear models and closed-form solutions are not available. Even for linear models, new optimization algorithms may be needed to estimate model parameters given much bigger data volume and the need to add regularization terms as discussed in the previous section. The gradient descent method, or its variants, is the most popular optimization algorithm in model training.

We will use a logistic model to illustrate the concept. Logistic model  $Y = \frac{1}{1+e^{-(b+w_1x_1+w_2x_2+w_3x_3+w_4x_4)}}$  and loss function (MSE) with L2 regularization can be defined as follows:

$$L = \frac{1}{m} \sum_{j=1}^m [Y_j - f(X^j; W, b)]^2 + \lambda (b^2 + \sum_{j=1}^4 w_j^2)$$

A model parameter  $p$  is updated gradually in the optimization process until the loss function stops decreasing.

$$p = p - \alpha \frac{\partial L}{\partial p} \quad p \in [W, b]$$

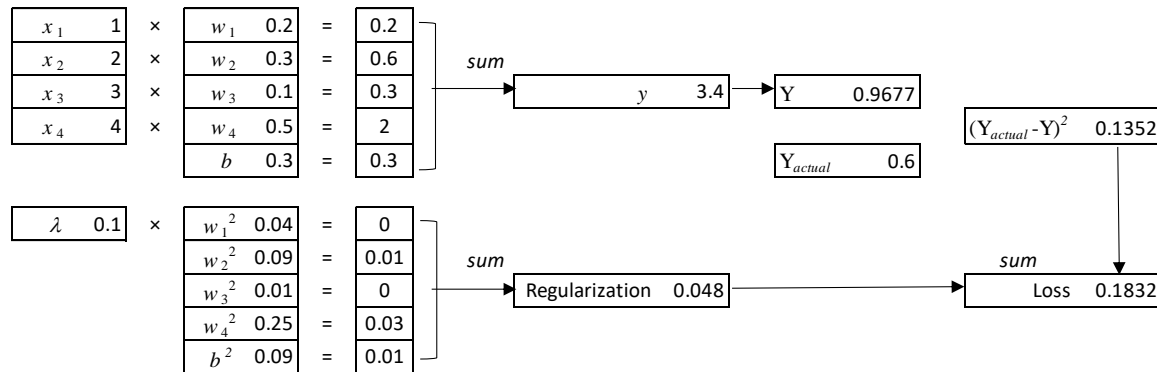
Where

$\alpha$ : learning rate that controls the updating speed.

For illustration, assuming only one training data is available ( $m=1$ ) and the initial model parameters are randomly generated. Figure A.26 shows the initial pass from the input to the loss function.

Figure A.26

#### LOSS FUNCTION AT INITIAL STEP



Note: some cells are showing a value of 0 due to rounding in Figure A.26 to Figure A.28.

With all the values in the forward pass, the gradient  $\frac{\partial L}{\partial p}$  can be calculated backward. It is the ratio of the change in the loss function  $L$  given a small change in the value of parameter  $p$ . For example, the calculation of  $\frac{\partial L}{\partial w_1}$  can be done in the following steps:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial y} \frac{\partial y}{\partial w_1} + 2\lambda w_1$$

$$\frac{\partial L}{\partial Y} = \frac{\partial (Y_{real} - Y)^2}{\partial Y} = -2(Y_{actual} - Y) = -2(0.6 - 0.9677) = 0.7354$$

$$\frac{\partial Y}{\partial y} = \frac{\partial}{\partial y} \frac{1}{1 + e^{-y}} = \frac{e^{-y}}{(1 + e^{-y})^2} = Y(1 - Y) = 0.9677(1 - 0.9677) = 0.0313$$

$$\frac{\partial y}{\partial w_1} = \frac{\partial(b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4)}{\partial w_1} = x_1 = 1$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial y} \frac{\partial y}{\partial w_1} + 2\lambda w_1 = 0.7354 \times 0.0313 \times 1 + 2 \times 0.1 \times 0.2 = 0.0630$$

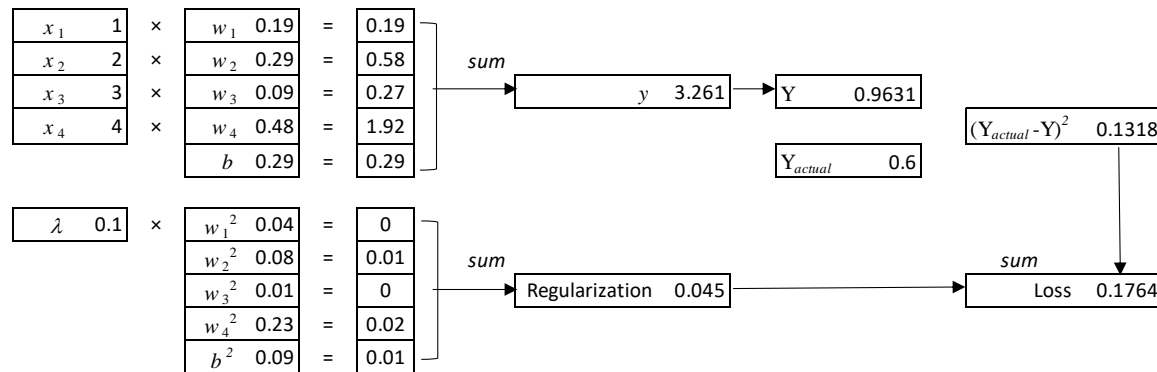
For the next iteration,  $w_1$  can be updated as

$$w_1 = w_1 - \alpha \frac{\partial L}{\partial w_1} = 0.2 - 0.1 \times 0.0630 = 0.1937$$

After updating all the parameters, the loss reduces from 0.1832 to 0.1764, as shown in Figure A.27.

Figure A.27

#### LOSS FUNCTION AT STEP 2



The loss reduces to 0.004 after 28 steps, as shown in Figure A.28. The remaining loss is purely from the regularization term with a perfect match between actual and predicted  $Y$ . In this illustration, five model parameters are used to fit the model to one data record and overfitting is expected. In practice, the process can be easily expanded with more training data where the average of the prediction errors is used in the loss function.

Figure A.28

#### LOSS FUNCTION AT STEP 28

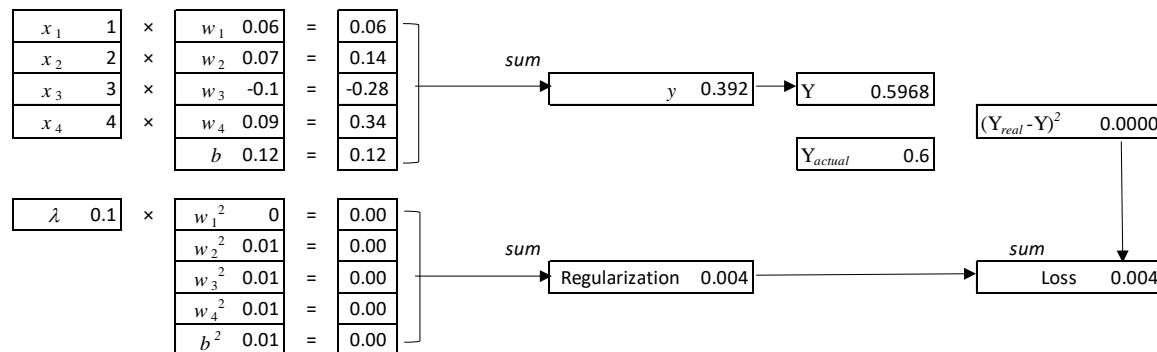
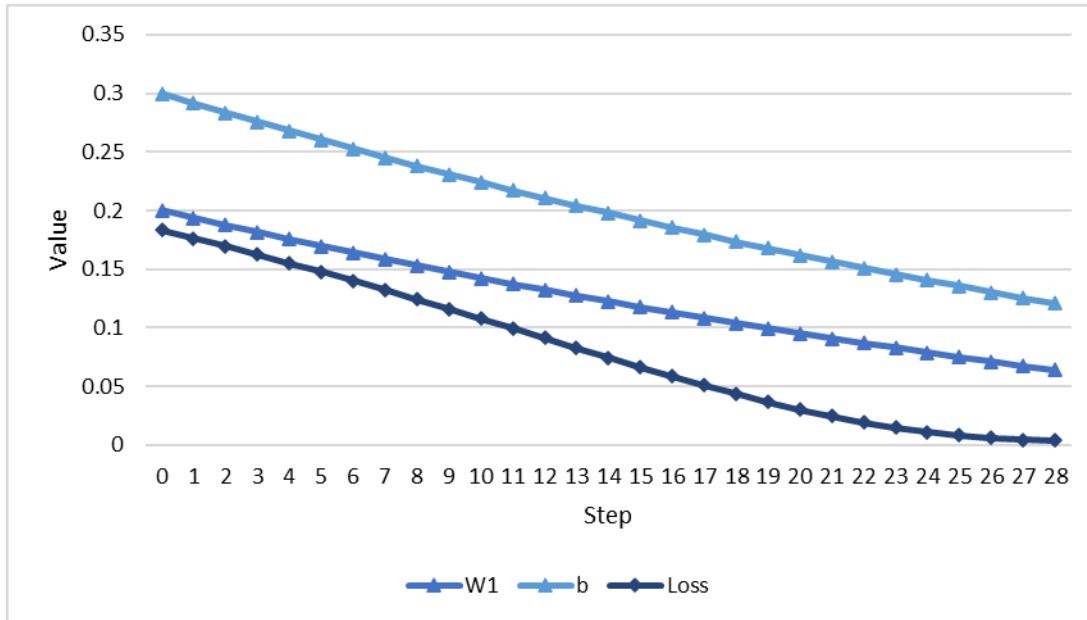


Figure A.29 shows the trajectory of the optimization process including a couple of model parameters and the loss function.

**Figure A.29**  
**MODEL CONVERGENCE USING GRADIENT DESCENT**



The gradient descent method explained above has many different versions. The backpropagation algorithm used to train ANNs is based on the gradient descent method. It uses the chain rule from the loss function backward through the layers to calculate the gradients and update all the parameters. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, Levenberg-Marquardt (LM) algorithm, and stochastic gradient descent (SGD) algorithm use gradient descent as well. While some algorithms can only guarantee reaching local minima, the SGD algorithm is less prone to be limited to local minima and more promising to find the global minimum loss. It is also faster than some other algorithms as it does not run through all training data records before updating the model parameters in the optimization process.

In addition to choosing a specific optimization algorithm such as the SGD, testing using different values of initial model parameters can help reduce the chance of falling into a non-optimal local minimum. Using a relatively large learning rate at the early stage of model training may also be helpful as a wider value range of model parameters can be tried.

#### A.4.4 HYPERPARAMETERS

Another aspect of model training is fine tuning model hyperparameters. Compared to model parameters that are usually part of a model formula, hyperparameters are fitting configuration parameters that cannot be derived from the data but help control the process of estimating model parameters. The loss function, regularization, optimization algorithm, and learning rate discussed above are all hyperparameters. In addition, model format, model complexity, and training stopping criteria also belong to model hyperparameters.

For an efficient model training, the learning rate is usually non-constant. It may follow a specific schedule that usually starts with a large value but gradually decreases with the optimization step. This allows a wide range of search at the initial stage and gradual improvement at later stages. On the other hand, many adaptive learning rate methods are available where the learning rate is adjusted based on real-time calibration results. For example, the RMSprop method is an adaptive learning rate method that uses the moving average of squared gradients to adjust parameter updating.



$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\text{MeanSquare}_t + \epsilon}} g_t$$

$$\text{MeanSquare}_t = 0.9\text{MeanSquare}_{t-1} + 0.1g_t^2$$

Where

$\theta$ : model parameter to update.

$\eta$ : learning rate.

$g_t$ : gradient of the loss function w.r.t. parameter  $\theta$  at step t.

$\text{MeanSquare}_t$ : moving average of squared gradients.

$\epsilon$ : a small constant.

Here the learning rate  $\frac{\eta}{\sqrt{\text{MeanSquare}_t + \epsilon}}$  is not a constant  $\alpha$  but inversely correlated with squared gradients. Other adaptive learning rate methods such as Adagrad and Adam are available as well. The Adagrad method adjusts the learning rate by parameter. Lower learning rates are used for parameters associated with explanatory variables that are frequently non-zero while higher learning rates are used for those associated with explanatory variables that are frequently zero. It is efficient for a dataset that contains sparse data. The Adam method is built on RMSProp with the addition of momentum that considers not only the impact of mean and squared gradients during the latest optimization step but also during previous steps. Although many adaptive learning rate methods are available, RMSprop is a good starting method that is effective in most cases.

Model format is another category of model hyperparameters. What explanatory variables should be included in the model? What new variables created through feature engineering should be used? How many hidden layers and neurons and what activation functions should be used in an ANN?

Model complexity hyperparameters are usually model specific. In a Random Forests model, the number of trees, the depth of trees and the minimum data points for a leaf are examples of hyperparameters that control the complexity of the model. Optimal values of these hyperparameters depend on the dataset and often times require fine tuning.

The stopping criteria of the optimization process to minimize the loss function needs to be set as well. The training process can be terminated if it passes the required number of steps, generates a loss value no greater than a threshold, or has not experienced improvement in the latest steps.

When model accuracy is not satisfactory, model hyperparameters can be fine-tuned to improve model performance.

## A.5 MODEL VALIDATION

After model training, calibrated models need to be assessed and compared using standard validation methods. It is important to know that validation data (out-of-sample data) needs to be used for a meaningful comparison so that the issue of overfitting can be identified. As described in [Section A.4](#), the data needs to be split into a training dataset and validation dataset before model training. Table A.9 lists a few methods of data splitting.

Table A.9  
DATA SPLITTING METHOD

Method	Definition	Usage
<b>Splitting by data record</b>	Randomly select validation data from all available data records, based on a fixed percentage.	Most popular approach with standard 80/20 split.
<b>Splitting by the value of variable(s)</b>	Randomly select validation data based on one or more variables so that the values of the variable(s) selected by the training dataset and the validation dataset are different.	It is useful for testing whether the model is useful for predicting new cases without past experience. For example, when using customers' transaction records to estimate credit score or life events, it is important to split the data by customer ID. Otherwise, a customer's transaction records may exist in both the training dataset and the validation dataset. This is sometimes called data leakage where training dataset contains the actual value of response variable used by the validation dataset.
<b>K-fold cross validation</b>	The dataset is split into $K$ parts evenly. Model training will be conducted $K$ times. Each time one part will be used as the validation data and the rest $K-1$ parts will be used as the training dataset.	It requires more computing time but is helpful for testing the robustness of model training.
<b>Leave one out cross validation (LOOCV)</b>	Similar to K-fold cross validation, LOOCV splits the dataset ( $N$ records) into individual data records and performs the model training $N$ times. Each time, only one data record is used for validation and the rest $N-1$ records are used for training.	It is very computationally intensive but can be used on small datasets where each individual estimation is important.

With the validation data ready, different methods can be used for different model types. The rest of this section discusses model validation for supervised learning (classification and regression), unsupervised learning and reinforcement learning, with the focus on supervised learning.

#### A.5.1 REGRESSION MODEL VALIDATION

To assess the goodness-of-fit of regression models, a common measure is coefficient of determination, also known as  $R^2$ .

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} = 1 - \frac{\sum_i (y_{pred,i} - y_{actual,i})^2}{\sum_i (y_{actual,i} - \bar{y}_{actual})^2}$$

This is applicable to not only linear regression but also other regression models. To offset the positive impact of an increasing number of parameters on  $R^2$ , adjusted  $R^2$  penalizes the goodness-of-fit measure based on the number of model parameters.

$$\text{Adjusted } R^2 = 1 - (1 - R^2) \frac{n - 1}{n - p - 1}$$

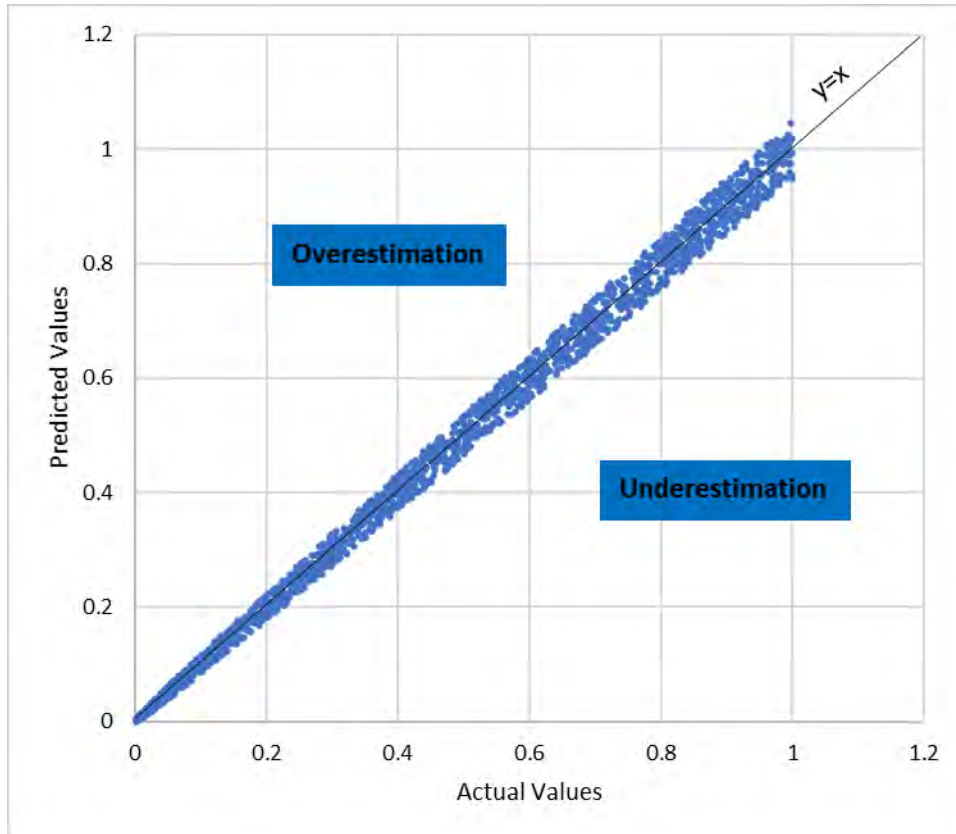
Where

$n$ : number of data records.

$p$ : number of model parameters.

Models can be ranked based on goodness-of-fit measures at a high level. However, further analysis is usually desired to look at the actual predictions. Scatter plots of the actual values and predicted values are a good way to identify outliers and get comfortable with model accuracy. Figure A.30 shows an example of a scatter plot to evaluate regression model accuracy. Dots lying on line  $y=x$  represent perfect estimation. Even if a model has a high  $R^2$ , scatter plots may help identify outliers which may be too important to ignore and may lead to a different model choice. As we described in the EDA section, plotting and visualizing the data is very useful.

**Figure A.30**  
**SCATTER PLOT: REGRESSION MODEL VALIDATION**



In addition to scatter plots, feature importance analysis is also helpful for model validation and will be discussed in [Section A.5.3](#).

## A.5.2 CLASSIFICATION MODEL VALIDATION

When assessing whether a model is good for solving a classification problem, different measures from those used with regression are used. Precision, recall and the F-measure are popular measures based on the confusion matrix, as shown in Table A.10.

**Table A.10**  
**SAMPLE CONFUSION MATRIX**

	Predicted: True	Predicted: False
Actual: True	True Positive	False Negative
Actual: False	False Positive	True Negative

Precision measures the Type I error<sup>22</sup> and recall measures the Type II error. F-measure (or F-score) is the harmonic average of precision and recall and may be used as a high-level measure to rank the performance of different models.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall (True Positive Rate)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$F - \text{measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Another widely used measurement in classification problems is the Receiver Operating Characteristic (ROC) curve and the area under the curve (AUC). The ROC curve helps understand the tradeoff between the true positive rate and the false positive rate by varying the threshold that is used to determine whether a prediction is positive or negative.

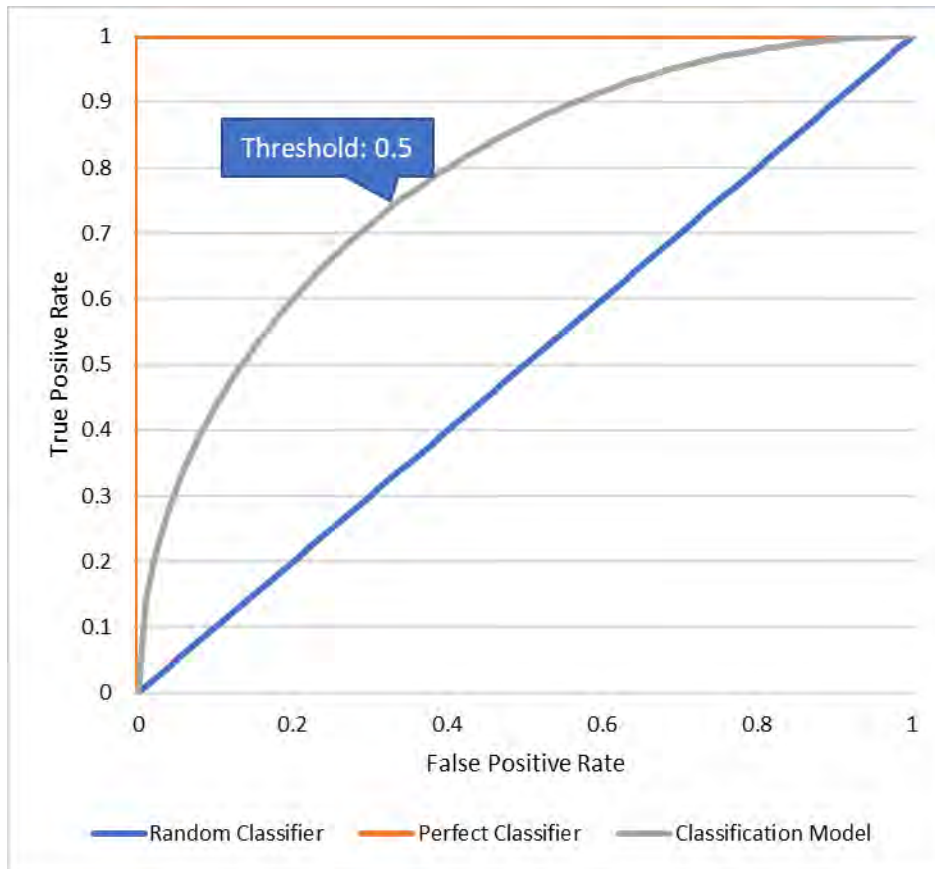
$$\text{fall - out (False Positive Rate)} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}}$$

The AUC tells the capability of the classification model to distinguish between two classes. Figure A.31 shows the ROC curves of a perfect classifier, a random classifier, and a sample model classifier. A perfect classifier has an AUC of 1 and a random classifier has an AUC of 0.5. AUC can be used to rank classification models.

---

<sup>22</sup> Recall from classical statistics, a Type 1 error is a false positive where you reject a true hypothesis. A Type II error is a false negative and occurs when you fail to reject a false hypothesis.

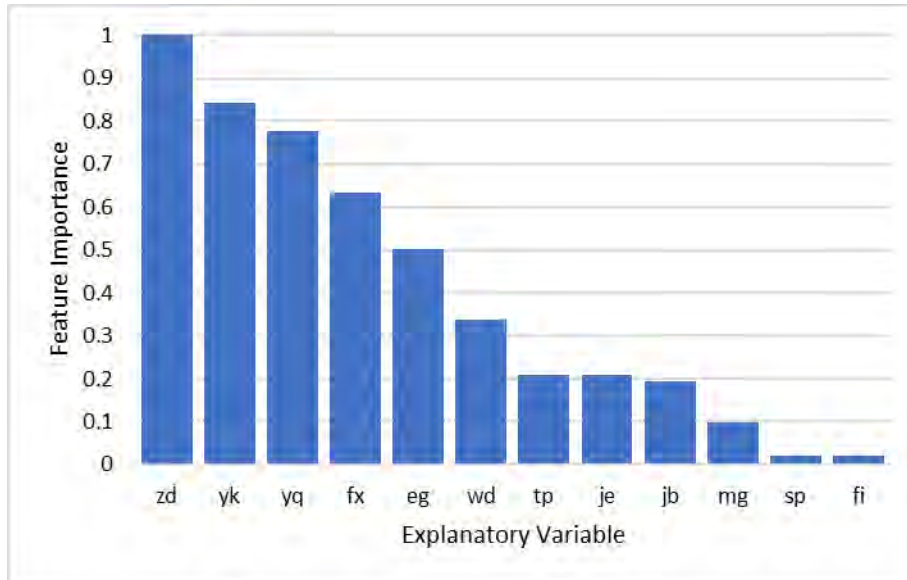
Figure A.31  
SAMPLE ROC



### A.5.3 FEATURE IMPORTANCE

For both classification and regression model validation, it is helpful to understand what explanatory variables are driving the prediction. Figure A.32 illustrates a typical feature importance analysis.

Figure A.32  
FEATURE IMPORTANCE ILLUSTRATION



It is beneficial in three ways.

- If some unexpected variables show in the list of important features, it helps identify potential issues with the model and data and requires further investigation before implementing the model.
- Important features can be used to set up key risk indicators and be frequently monitored for material changes.
- In the presence of overfitting, unimportant features may be removed.

The method of determining feature importance varies by model.

- **Linear regression** and **GLM**: The explanatory variables are normalized to the range of [0,1] before model fitting. A variable's importance is measured by the absolute value of the coefficient of that variable.
- **CART**: A variable's importance is measured by the increase of data purity because of a split based on that variable. For a regression problem, the importance of variable  $x_i$  can be calculated as follows.

$$Imp_{regression}(x_i) = \frac{1}{T} \sum_{t=1}^T \left( \sum_{s=1}^S \frac{N_L \cdot N_R}{N_L + N_R} (\bar{Y}_L - \bar{Y}_R)^2 \cdot Ind(split = x_i) \right)$$

Where

$x_i$ : the  $i$ th input variable.

T: total number of CART models in the RF model.

S: total number of splits in a CART model.

$\bar{Y}_L$ : the mean of Y in the left node after the split.

$\bar{Y}_R$ : the mean of Y in the right node after the split.

$N_L$ : the number of records in the left node after the split.

$N_R$ : the number of records in the right node after the split.

$Ind(split = x_i)$ : indicator function with a value of 1 if the split is based on variable  $x_i$  and a value of 0 otherwise.

For a classification problem, the measure  $\frac{N_L \cdot N_R}{N_L + N_R} (\bar{Y}_L - \bar{Y}_R)^2$  needs to be replaced. A possible measure is the improvement of the Gini impurity index  $G(N)$ , as defined in [Section A.3.1](#). For each split based on variable  $x_i$ , the Gini importance can be measured as the reduction in the Gini index:

$$Gini Imp(x_i) = (N_L + N_R)G(N) - N_L G(N_L) - N_R G(N_R)$$

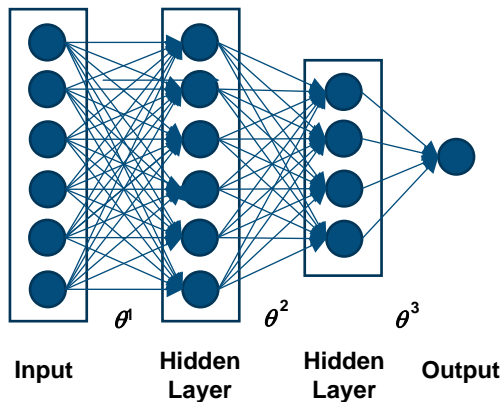
If the variable is used in multiple splits, the Gini importance is aggregated for the variable.

$$Imp_{classification}(x_i) = \frac{1}{T} \sum_{t=1}^T \left( \sum_{s=1}^S Gini Imp(x_i) \cdot Ind(split = x_i) \right)$$

- **Random Forests:** A variable's importance can be measured as the average importance level in each individual CART in the Random Forests model.
- **ANN:** For complicated models like ANN, it is not always obvious how feature importance should be calculated. As an example, one possible way to measure feature importance of an ANN model with two hidden layers and three sets of parameters  $\theta^1$ ,  $\theta^2$  and  $\theta^3$  is given below. Figure A.33 shows the architect of the ANN model.

Figure A.33

#### ANN MODEL ILLUSTRATION



A possible measure is to consider the impact of the explanatory variable through three layers using a chain rule including the two hidden layers and the output layer, based on the ratio of the absolute value of the parameter and the sum of all parameters in that layer:

$$Imp_{ANN}(x_i) = \sum_{j=1}^{n1} \sum_{k=1}^{n2} \frac{|\theta_{ij}^0|}{\sum_{r=1}^n |\theta_{rj}^0|} \cdot \frac{|\theta_{jk}^1|}{\sum_{s=1}^{n1} |\theta_{sk}^1|} \cdot \frac{|\theta_{kY}^2|}{\sum_{t=1}^{n2} |\theta_{tY}^2|}$$

Where

$x_i$ : the  $i$ th input variable.

$n1$ : the number of neurons in the first hidden layer.

$n_2$ : the number of neurons in the second hidden layer.

$n$ : the number of explanatory variables.

$\theta_{ij}^0$ : the parameter that determines the weight of the  $i$ th input variable applied to the  $j$ th neuron in the first hidden layer.

$\theta_{jk}^1$ : the parameter that determines the weight of the  $j$ th neuron in the first hidden layer applied to the  $k$ th neuron in the second hidden layer.

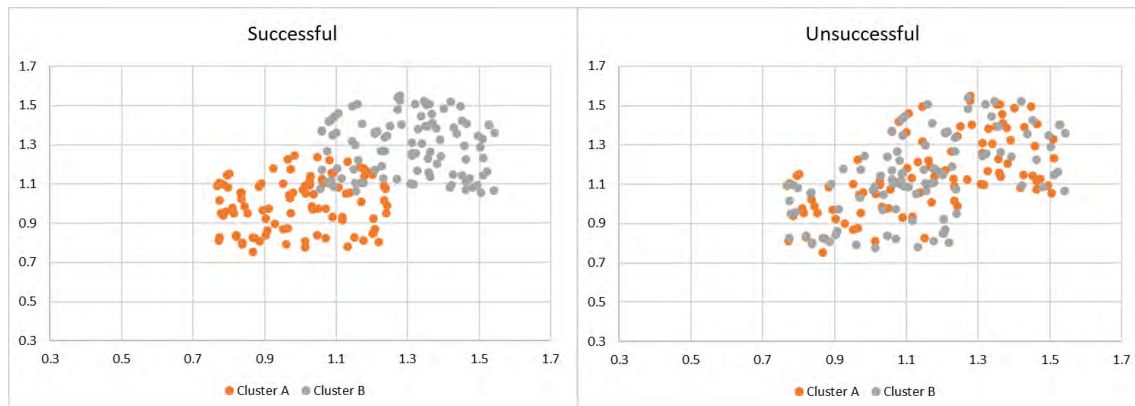
$\theta_{ky}^2$ : the parameter that determines the weight of the  $k$ th neuron in the second hidden layer applied to the output variable  $Y$ .

#### A.5.4 UNSUPERVISED LEARNING MODEL VALIDATION

Validating unsupervised learning models is less structured than supervised learning as normally the response variable is not involved in unsupervised learning. For methods such as PCA and clustering that follow standard algorithms, validation focuses more on reviewing and interpreting the results, and investigating any data or model issues in the presence of unexpected results. Figure A.34 shows results of two sample clustering exercises. By visualization, it is clear that the graph on the left represents a successful model while the graph on the right represents an unsuccessful one.

**Figure A.34**

#### CLUSTERING RESULT VALIDATION



On the other hand, for methods such as autoencoder that has an error function defined as the difference between recovered data and the original data to minimize, the same validation approaches used by regression analysis can also be applied here.

#### A.5.5 REINFORCEMENT LEARNING MODEL VALIDATION

Unlike traditional supervised learning problems that have actual values and predicted values, it is not straight forward to tell if a reinforcement learnt strategy is optimal or not. To assess when the reinforcement learning is effective, a few checks can be made.

- Using a reward function where supervised learning can be applied to check if the deep learning models are effective for estimating the actual reward. It then becomes a supervised learning validation issue and the methods described in [Section A.5.1](#) can be used.
- Even though the reward function is learned by a deep learning model, which means the true reward value is unknown, it is still possible to check the convergence of the trained model by comparing the error



function to the estimated reward function. For good model convergence, it is expected that prediction error as a percentage of estimated reward decreases to the expected level.

- The reinforcement learnt strategy can be compared to strategies from other optimization methods such as grid searching and dynamic programming. Often times these other optimization methods can solve a simple version of the problem. If we can show reinforcement learning produces similar results as these methods under the simplified construct, we have more confidence the model can produce reliable results under more complicated circumstances. The latter cases often times cannot be solved by the other optimization methods and that is why reinforcement learning is used.

## A.6 RESULT COMMUNICATION

Predictive analytics contain many technical concepts that can be difficult to explain, especially with the growing number of models and their complexity. Material efforts are needed to be able to effectively communicate with the final decision-maker the results of predictive analytics. Although a separate paper may be needed on this topic, a few key recommendations are given below:

- As in any effective communication, knowing your audience is the key. With an understanding of your audience's background, prior knowledge of and experience with predictive analytics, the communicator can carefully weigh on the content to be presented and the way they are presented.
- Relevance is important to attract interest. By linking the predictive analytics with something that the audience cares about, the chance of success will be higher. For example, when discussing a predictive model that estimates mortality, in addition to talking about model accuracy, a more relevant topic would be how the model can improve decision-making and the financial impact of switching to the model compared to maintaining the status quo. Actionable suggestions need to be embedded in predictive analytics results communication.
- No matter what the background of the audience is, it is always easier and more fun to explain difficult concepts using graphs and/or tables. Result visualization is a powerful tool to deliver messages. Also being able to model on the fly in a dynamic way can really allow decision makers to gain intuition on what the models are doing and their impact on metrics of concern.
- Rather than communicating everything at one time, it is easier to present the results gradually and sometimes repeatedly throughout a predictive analytics project. Even in one presentation, if a difficult technical detail is necessary to get the buy-in from the audience, it may be better to split the discussion into smaller pieces to explain.
- In addition to one-way communication, active involvement of stakeholders is very important to get them on board. Stakeholders should be encouraged to provide inputs to the process and should be consulted about their interests, concerns, expectations and opinions.
- Actuaries should also consult with appropriate ASOPS. For example, ASOP 41 on actuarial communications and ASOP 56 on modeling, are of particular relevance.
- To the extent end decision makers do not need to know the details of the model, communicating sufficiently in an Appendix or separate technical document is also recommended.

Once sufficient communication is made among stakeholders, a decision needs to be made regarding the best model to be used, if at all. Compared to existing decision-making rules, the financial impact of using the new model can also be quantified. By subtracting the cost of implementing the new model, the net impact can be used as a selection criterion. Costs of implementation include computing resources, database, program maintenance, training cost, and so on. Validation datasets can be used to evaluate the impact of adopting the model. In addition to model accuracy and financial impact, model complexity and model risk are also important factors to consider. Given two models both of which have satisfactory prediction accuracy and financial impact, the model that is easier to

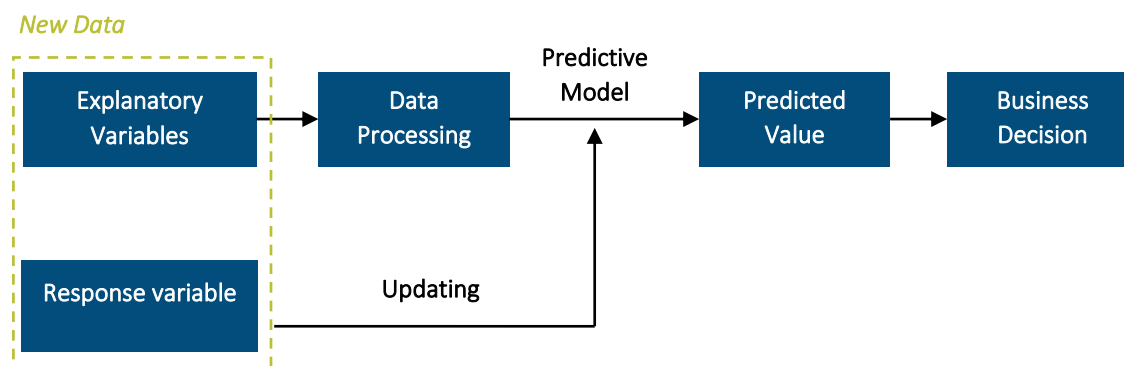
understand, communicate and validate is likely to be chosen even though it has a lower accuracy. Simplicity also often times trumps complexity in designing a more robust, less potentially overfit model to deploy in real time.

## A.7 MODEL IMPLEMENTATION

Once it is decided that a model will be used for a real prediction task, a prediction and model updating process needs to be set up, as shown in Figure A.35.

Figure A.35

### SAMPLE PREDICTION PROCESS



When new data of explanatory variables arrives, it will be fed into the same data cleaning process used in model training. With the clean data, the selected predictive model can be used to estimate the value of the explained value. The estimated value can then be used to make business decisions. When the actual value of the response variable is available, new data records containing both explanatory and response variables can be added to the training dataset and used to update the predictive model when appropriate.

The updating process depends on many factors such as the volume of new data, the type of new changes, and the impact on decision-making.

- If the new data exhibits similar distributions and relationships to the existing data, model updating is not necessary. EDA can be used as preliminary analysis to evaluate whether a full-scale updating is needed.
- A threshold of new data volume may be set to trigger the updating process. However, the determination of the threshold can be arbitrary. A rule of thumb is that the new data is more than 10% of existing data. The threshold may be determined using  $k$ -fold cross validation, as described in [Section A.5](#). By testing different values of  $k$ , the smallest  $k$  when an undesired difference is spotted among the  $k$  sets of training results can be found.  $1/k$  can be used as the threshold so that if the new data volume is  $1/k$  of existing data, an updating is needed.
- When changes in a variable are spotted, if the volatility of that variable has been fully reflected in existing training datasets, an updating may not be necessary. This is because we already expect the variable to exhibit some amount of natural fluctuation. Alternatively, when the new data brings in values that are beyond what could be expected or there are outliers compared to existing values, an updating is needed. For example, if interest rates are positive in existing data and new data contains negative interest rates, a model updating is likely to be desired, or at least considered.
- The usage of the predictive modeling can also affect the updating cycle. If the model is used for pricing and the repricing follows a quarterly cycle, a quarterly model updating seems to be a reasonable choice.
- The required efforts to update the model can also play a role. If an automated process is set up for model updating and computing resources are available, more frequent model updating can be implemented.

Model risk is also an important area to focus on during implementation. Although model risk can be vastly mitigated during model validation and model selection, efforts still need to be made to make sure the model has been applied correctly for prediction. If a complicated model such as an ANN model is used, it may be a good idea to use a simple model as a benchmark to make sure the predictions are not too far off. It is difficult to check an ANN model given the large number of parameters, but it is easy to check a linear model without programming.

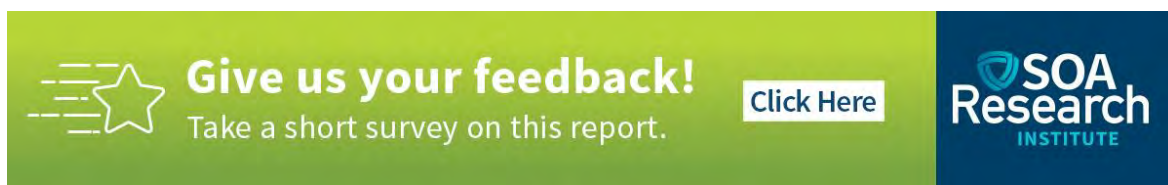
## Appendix B: Open-Source Python Program


Python codes are created for education purpose and hosted at [GitHub - Society-of-actuaries-research-institute/AR135-Predictive-Analytics-for-Retirement](https://github.com/Society-of-actuaries-research-institute/AR135-Predictive-Analytics-for-Retirement).


The codes are presented in the format of Jupyter notebook to be interactive. Two notebooks are available:

- `mort_cs.ipynb`: Python program used for the relative mortality case study in [Section 2](#). It includes EDA, data processing, model training and model validation.
- `derisking_cs.ipynb`: Python program used for the de-risking activity prediction in [Section 4](#). It includes all five classification models and model validation.

Each notebook is self-explained with input data available in the same GitHub repository.



 **Give us your feedback!**  
Take a short survey on this report. [Click Here](#)



## About The Society of Actuaries Research Institute

Serving as the research arm of the Society of Actuaries (SOA), the SOA Research Institute provides objective, data-driven research bringing together tried and true practices and future-focused approaches to address societal challenges and your business needs. The Institute provides trusted knowledge, extensive experience and new technologies to help effectively identify, predict and manage risks.

Representing the thousands of actuaries who help conduct critical research, the SOA Research Institute provides clarity and solutions on risks and societal challenges. The Institute connects actuaries, academics, employers, the insurance industry, regulators, research partners, foundations and research institutions, sponsors and non-governmental organizations, building an effective network which provides support, knowledge and expertise regarding the management of risk to benefit the industry and the public.

Managed by experienced actuaries and research experts from a broad range of industries, the SOA Research Institute creates, funds, develops and distributes research to elevate actuaries as leaders in measuring and managing risk. These efforts include studies, essay collections, webcasts, research papers, survey reports, and original research on topics impacting society.

Harnessing its peer-reviewed research, leading-edge technologies, new data tools and innovative practices, the Institute seeks to understand the underlying causes of risk and the possible outcomes. The Institute develops objective research spanning a variety of topics with its [strategic research programs](#): aging and retirement; actuarial innovation and technology; mortality and longevity; diversity, equity and inclusion; health care cost trends; and catastrophe and climate risk. The Institute has a large volume of [topical research available](#), including an expanding collection of international and market-specific research, experience studies, models and timely research.

Society of Actuaries Research Institute  
475 N. Martingale Road, Suite 600  
Schaumburg, Illinois 60173  
[www.SOA.org](http://www.SOA.org)