SOCIETY OF ACTUARIES

Considerations for Predictive Modeling in Insurance Applications

May 2019

Considerations for Predictive Modeling in Insurance Applications

AUTHORS

Eileen Burns, FSA, MAAA
Gene Dan, FCAS, MAAA, CSPA
Anders Larson, FSA, MAAA
Bob Meyer, FCAS, MAAA
Zohair Motiwalla, FSA, MAAA
Guy Yollin
Milliman

SPONSORS

Modeling Section
Predictive Analytics and Futurism Section
Committee on Life Insurance Research
Product Development Section
Reinsurance Section

CONTENTS

Acknowledgments

The authors wish to thank the Society of Actuaries (SOA) for sponsoring this paper and the members of the Project Oversight Group for their assistance in preparing this paper. The Project Oversight Group is comprised of Jim Wright, Carlos Brioso, Dorothy Andrews, Vikas Sharan, and Ronora Stryker. In addition, the authors gratefully acknowledge the assistance of Tony Huang, Steve Walsh, James Dodge, Hyunsu Kim, Christian Verdin, Loretta Agyemang, and Helen Wang for their assistance with various aspects of this project. We thank Al Klein, Michael Niemerg, Shea Parkes, Peggy Brinkmann, and Matthias Kullowatz for their thoughtful feedback and review. We also wish to thank the numerous members of the global Milliman community who have helped shape this document through their contributions to discussions within our predictive analytics and data science community of practice and other means. Last, the authors wish to thank the actuaries of the case study firm for volunteering information about their predictive modeling processes and the survey participants for providing valuable insight for this report.

Considerations for Predictive Modeling in Insurance Applications

Section 1: Introduction

In a world where data and analytics are quickly making over many industries, the Predictive Analytics and Futurism and the Modeling sections of the Society of Actuaries (SOA), along with other SOA sections, are interested in educating actuaries on how best to implement predictive modeling into relevant areas of actuarial practice. The SOA engaged Milliman to study this topic and develop a research report that includes a review of existing literature and current industry practices, as well as a comprehensive set of considerations for predictive modeling in insurance applications.

Insurance companies are at varying degrees of adopting predictive modeling into their standard practices, making it a good time to pull together experiences of some who are further on that journey. It is also important to provide lessons learned in other industries and applications and to identify areas where actuaries can improve their methods. Because data science and predictive analytics are rapidly advancing, there will no doubt be continued opportunities to improve on current leading methodologies, so we include a discussion to address the concern of staying current.

For purposes of this paper, we refer to Max Kuhn's definition of predictive modeling: "the process of developing a mathematical tool or model that generates an accurate prediction."[1]

This is broad enough to include any modeling process that generates a prediction about a future or unknown event. This does include many classical actuarial models, such as a mortality table or a system of multiplicative rating factors for developing health insurance premiums, as well as relatively basic algorithms like ordinary least squares linear regression. However, the prevalence of modern, more complex algorithms and techniques has increased the need for robust validation and model governance and has introduced new considerations for actuaries.

Stanford's Andrew Ng has summarized the process broadly as:

- Begin with a question or an idea.
- Translate the idea into code.
- Experiment to determine the initial idea's success or failure.[2]

We identify the components of an advanced predictive modeling function that, based on our experience, a company must address to have the best chance of gaining buy-in from stakeholders. Those components include:

1. Definition of project objective.

[1] Kuhn, Max. 2014. *Applied Predictive Modeling*. New York: Springer.

[2] Clark, Dan. 7 Useful Suggestions From Andrew Ng "Machine Learning Yearning." *KDNuggets.com,* 2018, *https://www.kdnuggets.com/2018/05/7-useful-suggestions-machine-learning-yearning.html* (accessed January 23, 2019).

2. Data collection and preparation.
3. Algorithm selection.
4. Feature engineering and selection.
5. Model evaluation and measurement.
6. Model deployment.
7. Model governance.
8. Software selection.
9. Staying current.

We then approached the question of defining leading practices in each area from a few angles. We completed a literature review to identify references that document how experienced practitioners approach questions about how they should structure their work. Where these were not from an actuarial application, we highlight any implications for pulling them into an actuarial context. In the section on model evaluation and measurement, we note a lack of specific, instructive examples in the literature for actuarial applications and suggest this as an area for new research. We conducted a survey of SOA members to assess what methods practitioners currently use for each component, to share how that may differ across disciplines, and to share any additional insights or suggestions actuarial practitioners thought were important. We compiled two case studies with a single life insurer to document examples of predictive modeling in development and in production within a familiar context. We used the case study approach to showcase how one company has made many of the decisions required to adopt predictive modeling and to highlight success and struggles along the way.

Ultimately, predictive modeling has emerged as a field that requires judgment at nearly every step. For any question, there may be many reasonable answers. We have, therefore, delivered our guidance as a compilation of concerns practitioners should address, with suggestions offered for how one might make the necessary decisions in a given application.

This report begins with the literature review to give appropriate context for the guidance and considerations presented in the second section. We then share two case studies as an application of the guidance. The first case study covers the key pieces of predictive modeling leading up to model evaluation, via an application related to flexible premium payment patterns. The second case study covers the pieces later in the process, the model deployment and model governance, via an application related to post-level-term lapse patterns. Appendix 1 presents the full findings of our SOA member survey, including the list of questions and possible responses in the survey. Where relevant, we reference those findings in earlier sections as well. Finally, Appendix 2 includes a glossary of predictive modeling and statistical terms used in this paper.

Section 2: Literature Review

2.1 BACKGROUND

The world is now awash with data. Today's data look different from that of the past in terms of quantity, structure, and sources. While actuaries are mathematical and statistical experts in using large data sets for deriving specific valuations, there is far more data available for analysis than can be neatly constrained to Excel. A more comprehensive software solution is required. The data are also varied: structured, semi-structured, and unstructured. Furthermore, data can be collected from a multitude of sources that are outside of the usual informational intake sources of insurance companies.

Ultimately, insurance is a big data business. The machine learning revolution has provided tools and techniques for leveraging data on this new scale. The insurance industry should therefore try to leverage what it can from the machine learning revolution. Surveys of life insurance companies that the SOA conducted in 2017[3] and 2018[4] indicated that predictive analytics were already being used for a wide range of programs within marketing, underwriting and post-issue management.

It's no surprise that, per a PwC report issued in March 2018, "insurers are increasingly looking to integrate data scientists into their organizations."[5] While incorporating data scientists into the actuarial realm is, indeed, one option, actuaries possess the fundamental requirements for data science and the domain knowledge to ensure it is applied properly. However, not many actuaries currently have training in machine learning, deep learning and the programming languages required for building advanced predictive models from the ground up.[6] By adding a predictive analytics component to its exam series, the SOA has signaled its recognition of the value-added benefit of incorporating predictive analytics into the actuarial skill set.[7] With a foundation in the fundamentals, ongoing education in methodology and experience related to operational aspects of analytics, more and more actuaries can begin to merge their expertise with both the data science process and the model governance objectives discussed below.

As we describe the current leading practices in each operational area of predictive modeling, we must address the similarities and differences between predictive models and other actuarial models. Predictive models that actuaries use are a subset of the models they use, and as such, we can rely on an abundance of literature related to best practices in modeling and model governance. These include items such as actuarial standards of practice (ASOPs), practice notes and the like. Predictive models, however, are not the focus of these documents, and as such, there remain important details to be pulled from other sources. We note that many of the best sources are from outside of the actuarial realm. We expect that while the actuarial literature will continue to expand, the number of data scientists far exceeds the number of actuaries nationally and globally, so it will benefit actuaries to continue to watch these sources for new ideas and inspiration.

---

3 Society of Actuaries. Predictive Analytics and Accelerated Underwriting Survey Report. *SOA,* May 2017, *https://www.soa.org/experience-studies/2017/predictive-analytics-underwriting* (January 23, 2019).
4 Society of Actuaries. Predictive Analytics and Accelerated Underwriting Follow-up Survey Report. *SOA,* March 2018, *https://www.soa.org/resources/experience-studies/2018/predictive-analytics-accelerated-follow-up* (accessed January 23, 2019).
5 PWC. How Do Actuarial and Data Science Skills Converge at P&C (Re)Insurers? March 2018, *https://www.pwc.com/us/en/insurance/publications/assets/pwc-pc-actuaries-data-scientists-role.pdf*, (accessed January 23, 2019).
6 Li, Xiaochuan (Mark). 2017. Learning Data Science from an Actuary's Perspective. *Society of Actuaries, Actuary of the Future Newsletter*, Issue 40, Page 10.
7 Society of Actuaries. Exam PA: Predictive Analytics. 2018, *https://www.soa.org/Education/Exam-Req/edu-exam-pa-detail.aspx* (accessed January 23, 2019).

We pulled from many sources for this review. For the technical pieces related to model selection and evaluation, we relied heavily upon several key textbooks known to our team. For operational or project management topics, we found that a widely cited document published by SAS gave reasonably comprehensive coverage. For model governance, we relied mainly on a few key documents from the insurance and banking industries. We supplemented these sources with articles, blogs, and whitepapers as needed to bring the content forward to current practices. For example, a classic textbook will cover imputation of missing values but hasn't been updated to include the latest algorithms available to accomplish this task. We identified such relevant resources by recalling past use of certain references, searching known web resources for specific content, and by reaching out to the broader Milliman community to gather additional ideas. It is easy to find multiple references on most of the topics covered, so we referenced the most complete and credible sources in case our readers wish to delve deeper.

This literature review is intended to document processes currently used by practitioners in predictive modeling. We recognize areas in which actuaries may be moving forward on a spectrum from beginner to advanced practitioner. We hope this provides a pathway for actuaries to continually advance.

2.2 PROJECT OBJECTIVE

Defining the project objective includes describing the modeling objective as well as the constraints within which modeling occurs. For our purposes, we are focusing on the "project" of building and implementing the predictive model, which may be part of a larger "project" for the company with a broader set of objectives. The modeling objective means the question that is under consideration and the criteria that will be used to determine success. Modeling will occur within an organization and is, therefore, subject to constraints of data availability, human resource, computing resources, stakeholder knowledge, budget and time. A well-defined project objective addresses all of these components and includes contingencies for risks that may arise in each area. It may also include expectations for how the model will move forward after it is built. The goal of this project definition is to ultimately allow for a cost-benefit analysis of the modeling effort.

A thorough plan addresses each of the following areas. These are described in more detail in IBM's CRISP-DM guide[8] and in studies from the Casualty Actuary Society,[9] Dorothy Andrews[10] and Hal Kalechofsky.[11]

Business problem. There must be a reason the project has been proposed as one that can be solved with a predictive modeling solution. There is a business problem to be answered with data. The business problem in relation to the data and the algorithmic approach will be refined as business requirements and stakeholder input are gathered, including requirements on time.

8 IBM. IBM SPSS Modeler CRISP-DM Guide. *ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf* (accessed January 23, 2019).
9 Goldburd, Mark, Anand Khare, and Dan Tevet. Generalized Linear Models for Insurance Rating. *Casualty Actuary Society*, 2016, *https://www.casact.org/pubs/monographs/papers/05-Goldburd-Khare-Tevet.pdf* (accessed January 23, 2019).
10 Andrews, Dorothy L. 2017. Predictive Model Building 101. *Society of Actuaries, Predictive Analytics and Futurism Newsletter*, Issue 15, Page 10.
11 Kalechofsky, Hal. A Simple Framework for Building Predictive Models: A Little Data Science Business Guide," MSquared Consulting, September 2016, *http://www.msquared.com/wp-content/uploads/2017/01/A-Simple-Framework-for-Building-Predictive-Models.pdf* (accessed January 23, 2019).

Business goals. There is a way to determine success. Note that success may be defined subjectively or objectively, and objective definitions will be much easier to determine completed. "Remember to keep goals realistic."[12]

Modeling objective. As stated in the SOA PA Exam learning objectives, delineating the business problem and translating "a vague question into one that can be analyzed with statistics and predictive analytics"[13] is a crucial step in designing and deploying a predictive model. In data science, question construction is both an art and a science for deriving the specific and measurable business problem that the stakeholders intend to solve.[14]

Organizational context. Every organization has its own culture related to analytics. Some organizations are just beginning on the journey of analytics, and for some, it is well established as a tool for making business decisions. This spectrum implies corresponding differences in knowledge. Staff members performing the work may have deep or surface knowledge of modeling; stakeholders responsible for determining success of a modeling effort may be well versed or may need educating. Other questions to consider include: What is the regulatory environment? What ASOPs may apply? What information security concerns exist? Who are all the players who will be affected? Where will the model live later?

Resources. People, computers, software, subject matter experts, data and other resources will all play into the ultimate success of the project and should be well understood and documented at the outset.

Contingencies. Taking a predictive model from the drawing board to deployment involves many steps, each one offering risks to the project's completion. Take the time to identify and communicate key risks and plan for workarounds.

2.3 DATA ACQUISITION AND PREPARATION

Data collection and preparation are components of exploratory data analysis (EDA). Sifting through the who, what, where, when, why and how of the data will further refine both the quantity and quality of what's available. This process is the next step in understanding the data more precisely. Data can be acquired from several different sources, which further depend on business requirements, e.g., a limit on the types of data that can be used in the analysis, anonymizing the data, the period under analysis and other business requirements.

Raw data can be a messy affair. Approximately 80% of data are unstructured,[15, 16] and they need to be extracted from external and/or internal sources and cleaned prior to the model-building phase.

Internal data sources that are already stored in the company's database management system, e.g., transactional data regarding policy changes or existing policyholder information, are generally easier to access if the appropriate data pipelines have been built. The concept of a data pipeline has been proposed

---

[12] See footnote #8.
[13] See footnote #7.
[14] See footnote #10.
[15] Rizkallah, Juliette. The Big (Unstructured) Data Problem. *Forbes.com*, June 5, 2017, *https://www.forbes.com/sites/forbestechcouncil/2017/06/05/the-big-unstructured-data-problem/#42ec344a493a* (accessed January 23, 2019).
[16] Shacklett, Mary. Unstructured Data: A Cheat Sheet. *TechRepublic*, July 14 2017, *https://www.techrepublic.com/article/unstructured-data-the-smart-persons-guide* (accessed January 23, 2019).

as an "ideal mix of software technologies that automate the management, analysis and visualization of data from multiple sources, making it available for strategic use."[17] In practice, its construction can be a difficult task. In most cases, the information technology (IT) department—more specifically the data engineer or database administrator—constructs the data pipelines based on the data scientist's needs.

External sources may or may not pose a data collection challenge. For example, extracting certain census data from the U.S. government can be as simple as downloading a CSV file. Alternatively, data may be embedded in the JSON or XML formats. Python and R have libraries or packages that can be used for extracting data from a wide range format.[18]

Data scientists can spend anywhere between 50% and 80% of their time on data preparation.[19, 20] Missing and null values, transposition and spelling errors, duplicate values and outliers disrupt the data quality and consistency, which will skew the model's predictive accuracy. Data preparation is an iterative process that can be performed manually, automatically or in some combination of both. For massive data sets, manual data cleansing greatly increases the lead time for the actual model construction, testing and validation processes.

Scripts written in Python or R can automate the data cleaning process.[21, 22, 23, 24] Automated and semi-automated methods are also available, which use an if/then rules-based set of algorithms and a violation graph that focuses on "either a functional dependency or a conditional rule" that will "detect and repair" the data.[25] Furthermore, enterprise data-cleaning applications are available via IBM,[26] OpenRefine,[27] RapidMiner,[28] Alteryx[29] and Trifacta.[30]

A final data preparation task is often to split the data into training, validation and test data sets. We mention it here because it is a data preparation task, though its benefits are leveraged in feature selection and model evaluation. The predictive algorithm learns from the training data set. The validation data set is

---

[17] Bordei, Alex. The Data Pipeline – Analytics at the Speed of Business. *Dataconomy*, April 10, 2017, *https://dataconomy.com/2017/04/data-pipeline-analytics-business* (accessed January 23, 2019).
[18] PypiPython Software Foundation. Json_to_CSV 1.2.9 Description, Project Description. March 13, 2016, *https://pypi.org/project/json_to_csv* (accessed January 23, 2019).
[19] SAS. 5 Data Management for Analytics Practices, Best Practices. 2018. *https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/data-management-for-analytics-best-practices-107769.pdf* (accessed January 23, 2019).
[20] Ruiz Gabernet, Armand and Jay Limburn. Breaking the 80/20 Rule: How Data Catalogs Transform Data Scientists' Productivity. IBM, August 23, 2017, *https://www.ibm.com/blogs/bluemix/2017/08/ibm-data-catalog-data-scientists-productivity* (accessed January 23, 2019).
[21] Aliya Small, Naeemah. Clean Your Data in Seconds With This R function. *R-bloggers*, July 17, 2018, *https://www.r-bloggers.com/clean-your-data-in-seconds-with-this-r-function* (accessed January 23, 2019).
[22] Surles, William. Cleaning Data in R. *Rpubs.com*, July 11, 2017, *https://rpubs.com/williamsurles/291107* (accessed January 23, 2019).
[23] Agarwal, Malay. Pythonic Data Cleaning With NumPy and Pandas. *Real Python*, March 26, 2018, *https://realpython.com/python-data-cleaning-numpy-pandas* (accessed January 23, 2019).
[24] Tutorialspoint.com. Python – Data Cleansing. 2018, *https://www.tutorialspoint.com/python/python_data_cleansing.htm* (accessed January 23, 2019).
[25] Tian, Yongchao, Pietro Michiardi, and Marko Vukolić. "Bleach: A Distributed Stream Data Cleaning System." In 2017 IEEE International Congress on Big Data (BigData Congress), pp. 113-120. IEEE, 2017.
[26] IBM. IBM InfoSphere DataStage and QualityStage. 2018, *https://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.nav.doc/containers/cont_iisinfsrv_ds_and_qs.html* (accessed January 23, 2019).
[27] OpenRefine. Introduction to OpenRefine. *http://openrefine.org* (accessed January 23, 2019).
[28] RapidMiner Platform. 2018, *https://rapidminer.com/products* (accessed January 23, 2019).
[29] Alteryx. Data Preparation. 2018, *https://www.alteryx.com/solutions/analytics-need/data-preparation* (accessed January 23, 2019).
[30] Trifacta. *https://www.trifacta.com* (accessed January 23, 2019).

"the sample of data used to provide an unbiased evaluation of a model fit on the training data set while tuning model hyper-parameters."[31] The final unbiased evaluation of the predictive model is done on the test data set.[32] The term holdout data set is sometimes used to refer to the test data set.[33] Splitting the data as described is referred to as holdout-validation; an alternative is cross-validation.[34] If holdout-validation is used, standard practice is to randomly assign records to the training and holdout data sets, typically setting a random seed so that the split is repeatable.[35]

2.4 ALGORITHM SELECTION

In general, machine learning or predictive algorithms can be defined as Y = f(X), where f is the target function for learning that maps X to Y. Said differently, these algorithms attempt to find an optimal function f such that f(X) is as close to Y as possible within imposed constraints. The goal here is algorithmic generalizability and continued accuracy because new data are ingested and analyzed. Mapping of the function f occurs through three main processes: supervised, semi-supervised and unsupervised. [36]

For supervised algorithm testing and training, where the algorithm is initially learning the relationships between X and Y through labeled training and testing data, there are two primary algorithmic subtypes: classification and regression. Classification algorithms are used with categorical target (Y) variables and include logistic regression, naive Bayes classifier, decision trees and K-nearest neighbor (KNN).[37] For continuous target variables, regression algorithms are used, including linear regression (of which there are many varieties) and decision trees.[38, 39] Many algorithms can be used for both classification and regression, including generalized linear models,[40] random forests, support vector machines and neural networks. Prime use cases for supervised learning within an actuarial framework include product or contract pricing (and repricing), developing mortality and lapse rate assumptions, pension valuations, projecting individual health care claim costs, and determining a customer's risk of developing a particular disease.

Using semi-supervised learning algorithms reduces the time required for labeling a large data set, because the objective is for the algorithm to learn the data structure without every single target variable or event having a label.[41] The algorithms above also can be used within the context of semi-supervised learning.

---

[31] Shah, Tarang. Train, Validation and Test Sets. *Tarang Shah Blog,* December 2, 2017, *https://tarangshah.com/blog/2017-12-03/train-validation-and-test-sets* (accessed January 23, 2019).
[32] Brownlee, Jason. What Is the Difference Between Test and Validation Datasets? Machine Learning Mastery, July 14, 2017, *https://machinelearningmastery.com/difference-test-validation-datasets* (accessed February 21, 2019).
[33] Artificial Intelligence Wiki (n.d.). Training Sets, Validation Sets, and Holdout Sets. Data Robot, *https://www.datarobot.com/wiki/training-validation-holdout* (accessed February 21, 2019).
[34] Zheng, Alice. (October 16, 2015). Evaluating Machine Learning Models. *O'Reilly*, October 16, 2015, *https://www.oreilly.com/ideas/evaluating-machine-learning-models/page/4/offline-evaluation-mechanisms-hold-out-validation-cross-validation-and-bootstrapping* (accessed February 21, 2019).
[35] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2017. *An Introduction to Statistical Learning: With Applications in R*. New York: Springer.
[36] *Ibid.*
[37] Valencia-Zapata, Gustavo A., Daniel Mejia, Gerhard Klimeck, Michael Zentner, and Okan Ersoy. "A Statistical Approach to Increase Classification Accuracy in Supervised Learning Algorithms." arXiv preprint arXiv:1709.01439 (2017).
[38] Seif, George. Selecting the Best Machine Learning Learning Algorithm for Your Regression Problem. *Towards Data Science*, March 4, 2018, *https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef* (accessed January 23, 2019).
[39] Qui, Junfei, et al. A Survey of Machine Learning for Big Data Processing. *EURASIP Journal on Advances in Signal Processing*, May 28, 2016, *https://link.springer.com/article/10.1186/s13634-016-0355-x* (accessed January 23, 2019).
[40] Society of Actuaries. Predictive Modeling: A Modeler's Introspection – A Paper Describing How to Model and How to Think Like a Modeler. June 2015, *https://www.soa.org/research-reports/2015/2015-predictive-modeling* (accessed March 21, 2019).
[41] Oliver, Avital, Augustus Odena, Colin A. Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. "Realistic evaluation of deep semi-supervised learning algorithms." In Advances in Neural Information Processing Systems, pp. 3235-3246. 2018.

Accordingly, the semi-supervised approach provides the same actuarial use case scenarios, with the main difference being that the semi-supervised approach is less time-consuming because target variables do not need to be named or labeled. [42]

With unlabeled training and test data sets, unsupervised learning techniques reveal patterns within the data. There is no predefined Y output for unsupervised learning methods, because this is a discovery process. The most commonly used algorithmic applications are K-means clustering, hierarchical clustering, support vector clustering and neural networks. In reference to actuarial objectives, cohort creation and customer segmentation are possible applications for unsupervised predictive algorithms. [43]

As a final note on applications, the financial sector has made tremendous strides in applied research for other predictive models that are not listed above, e.g., parameter learning, sparsity-aware learning, Bayesian methods and Monte Carlo methods. [44] These tools may become more common among actuaries in the future.

As with all statistical algorithms, predictive algorithms need to match the objective of the prediction and the available data. Choosing the best-fit algorithm relies upon the size, type and quality of the data; the computational resources available; the need for interpretability; and the time constraints. [45]

Data. "Fundamentally, classification is about predicting a label and regression is about predicting a quantity." [46] Hence, predicting a continuous target variable is essentially a regression problem, whereas predicting a categorical target variable is a classification problem. Supervised learning algorithms are used when working with labelled data (Data for which you already know the target answer) and unsupervised learning algorithms are used when the data are not labelled. [47, 48] Giant data sets will bring up questions about computational resources and time to completion.

Computational resources. Certain algorithms, e.g., deep learning, are very computationally intensive and may require distributed computing and/or cloud resources.

Interpretability. "While the primary interest of predictive modeling is to generate accurate predictions, a secondary interest may be to interpret the model and understand why it works. The unfortunate reality is

[42] DeepAI. Semi-Supervised Learning. https://deepai.org/machine-learning-glossary-and-terms/semi-supervised-learning (accessed January 23, 2019).
[43] Panlilio, Alex et al. Practical Application of Machine Learning Within Actuarial Work. Institute and Faculty of Actuaries, January 30, 2018, *https://www.actuaries.org.uk/documents/practical-application-machine-learning-within-actuarial-work* (accessed May 5, 2019).
[44] Accenture Consulting. Model Behavior. Nothing Artificial: Emerging Trends in the Validation of Machine Learning and Artificial Intelligence Models. 2017, *https://www.accenture.com/t20180427T082714Z__w__/us-en/_acnmedia/Accenture/Conversion-Assets/MainPages/Documents/Global/Accenture-Emerging-Trends-in-the-Validation-of-ML-and-AI-Models.pdf#zoom=50* (accessed January 23, 2019).
[45] Li, Hui. Which Machine Learning Algorithm Should I Use? *KDNuggets.com*, 2017, *https://www.kdnuggets.com/2017/06/which-machine-learning-algorithm.html* (accessed January 23, 2019).
[46] Brownlee, Jason. Difference Between Classification and Regression in Machine Learning. *Machine Learning Mastery*, December 11, 2017, *https://machinelearningmastery.com/classification-versus-regression-in-machine-learning* (accessed January 23, 2019).
[47] Khatun, Amnah. Let's Know Supervised and Unsupervised in an Easy Way. *Chatbots Magazine*, July 10, 2018, from https://chatbotsmagazine.com/lets-know-supervised-and-unsupervised-in-an-easy-way-9168363e06ab (accessed February 21, 2019).
[48] Amazon Machine Learning n.d. Collecting Labeled Data. *Amazon Web Service,* https://docs.aws.amazon.com/machine-learning/latest/dg/collecting-labeled-data.html (accessed February 21, 2019).

that as we push toward higher accuracy, models become more complex and their interpretability becomes more difficult. This is almost always the trade-off we make when prediction accuracy is the primary goal."[49]

Time. If there's time for research, new algorithms may be investigated. If not, use old standards. Time to fit algorithms goes along with computational resources, and time to predict from algorithms does too. When experimenting with various algorithms, if speed is an issue, common practice would be to subsample the data at this exploratory phase of the process.

Implementability. A complex predictive model, while robust and accurate in and of itself, may not easily be incorporated into a traditional actuarial projection model. Consider a predictive model for lapses that leverages macroeconomic explanatory variables, including a specific Treasury interest rate and the unemployment rate. While most company cash flow testing models at present would be well suited to reflecting simulated future Treasury interest rates, modeling simulated unemployment rates might pose more of a challenge.

Across the model governance literature (discussed in more detail in a later section),[50, 51, 52, 53] common recommendations regarding the algorithm selection include:

- Document the reasoning for selecting the algorithm.
- Document the assumptions implied by that algorithm (e.g., independence of features, linear relationships).
- Document any data manipulations needed to meet the requirements of the selected algorithm.
- Ensure that the structure and design of the chosen algorithm are appropriate for its intended use.

2.5 FEATURE ENGINEERING AND SELECTION

The set of features used is not always the same as the set of variables available in the raw data; some additional features will be created by combining, bucketing or transforming the original variables (feature engineering), and some variables will be excluded from the modeling process (feature selection). Feature engineering "attempts to create additional relevant features from the existing raw features in the data, and to increase the predictive power of the learning algorithm." Feature selection "selects the key subset of original data features in an attempt to reduce the dimensionality of the training problem."[54] In his widely cited paper A Few Useful Things to Know About Machine Learning, University of Washington Professor Pedro Domingos observes: "At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used."[55]

---

[49] See footnote #1.
[50] See footnote #8.
[51] AAA. PBR Model Governance Checklist: Some Considerations for Practicing Actuaries. *PBR Boot Camp*, June 2016, *http://actuary.org/files/imce/AcademyPBRBootCamp-PBRChecklist_06.06.2016.pdf* (accessed January 23, 2019).
[52] AAA. Model Governance Practice Note: Some Considerations for Practicing Life Actuaries. *American Academy of Actuaries*, December 2016, *https://www.actuary.org/files/publications/ModelGovernancePracticeNote_FinalDraft_10.30.2016.pdf* (accessed January 23, 2019).
[53] Actuarial Standards Board. Modeling (Fourth Exposure Draft). June 2016, *http://www.actuarialstandardsboard.org/asops/modeling-fourth-exposure-draft* (accessed January 23, 2019).
[54] Microsoft Azure. Feature Engineering in Data Science. November 20, 2017, *https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features* (accessed January 23, 2019).
[55] Domingos, Pedro. 2012. A Few Useful Things to Know About Machine Learning. *Commun. ACM*. 55, 78–87. 10.1145/2347736.2347755, *https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf* (accessed January 23, 2019).

Additional goals of feature engineering include increasing the model's interpretability, reducing redundancy (e.g., multicollinearity or correlated variables) and dimensionality, revealing additional relationships between the variables and standardizing or normalizing the data. Specific actions may include binning (translating a continuous variable into discrete), rescaling, applying principal components analysis (PCA), or using machine learning to identify interdependencies that may be captured through new features. Feature engineering may occur during initial data preparation, after algorithm selection, during feature selection or all of the above.[56]

Feature selection is used to further reduce dimensionality (i.e., the number of features in your model) and hone in on the features that directly apply to the business problem,[57] with the focal question being, "What is the best representation of the sample data that optimally allows the algorithm to learn a solution to your problem?" [58] As eloquently stated by The Little Prince author Antoine de Saint-Exupery, "Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away." The principle of parsimony is an overarching guide.

Feature engineering and selection can be laborious tasks when done manually, and choosing the correct approach is dependent upon all of the prior steps in the model building process, e.g., selected model performance measures, the predictive model or models, and how well the data has been prepped.[59] There are automated tools that assist with this phase of model development. At the library level, there is the Python library Featuretools;[60] the R package featuretoolsR provides an interface to this library from R.[61] The following machine learning systems also support feature selection and feature engineering: DataRobot,[62] Google Cloud Machine Learning Engine, Machine Learning on AWS[63] and Azure Machine Learning Studio.[64]

Finally, some algorithms may implicitly perform feature engineering and/or feature selection. Algorithms that utilize decision trees, such as random forests and gradient boosting machines (GBMs), effectively select important features by testing their impact on a specific error metric in real time. Splits in each tree are made across whichever feature most greatly reduces the error metric. Neural networks connect features to the target through "hidden layers" of generalized linear model (GLM) functions. The coefficients of these mini-GLMs are often called weights, and weights are adjusted through back-propagation to optimize the relationship between the features and the target. In this way, the relationship between a particular feature and the target can be zeroed out through the weights, which is basically feature selection. Penalized regression modifies the error metric by adding a constraint to how big the

---

[56] Harlalka, Rajat. Choosing the Right Machine Learning Algorithm. *Hacker Noon*, June 18, 2018, *http://hackernoon.com/choosing-the-right-machine-learning-algorithm-68126944ce1f* (accessed January 23, 2019).
[57] See footnote #54.
[58] Brownlee, Jason. Discover Feature Engineering, How to Engineer Features and How to Get Good at It. *Machine Learning Mastery*, September 26, 2014, *http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it* (accessed January 23, 2019).
[59] *Ibid.*
[60] Featuretools. *https://www.featuretools.com* (accessed January 23, 2019).
[61] Magnusfurugard. An R interface to the Python module Featuretools. 2019, *https://github.com/magnusfurugard/featuretoolsR* (accessed January 23, 2019).
[62] DataRobot. What Is Automated Machine Learning? 2008, *http://www.datarobot.com/automated-machine-learning* (accessed January 23, 2019).
[63] Amazon. Machine Learning on AWS. 2008, *https://aws.amazon.com/machine-learning* (accessed January 23, 2019).
[64] Microsoft Azure. Azure Machine Learning Studio. 2008, *https://azure.microsoft.com/en-us/services/machine-learning-studio* (accessed January 23, 2019).

coefficients can be in absolute value. This process can effectively drop features that are not predictive enough to make the cut.

2.6 MODEL EVALUATION AND MEASURES OF SUCCESS

The goal of any predictive modeling effort is to achieve the best predictive capability within the established constraints. Said differently, we are managing the bias-variance trade-off[65] to achieve the desired measures of success while adhering to the known business requirements. Model evaluation indicates whether we have done this.

Machine learning engineers at Google offer a mantra that underscores the importance of model evaluation and success measures: measure first, optimize second.[66] This implies addressing the following categories.

Business measures of success. Per the IBM CRISP-DM guide,[67] this will be set at the outset of the project, based on the project's and stakeholders' full business context. These measures can be objective, such as the model impact on company profitability, or subjective, such as ensuring the model is implemented seamlessly in the existing business process.

Performance metrics for predictive models. There are many sources for statistical learning that give helpful advice for how to approach performance metrics. Kuhn,[68] James,[69] Hastie[70] and Goldburd[71] are a few. The following are a representative sample of metrics described in those sources and in common use in data science.

- Confusion matrix: A tabular summary of predictions versus actual events in holdout data to compare rates of true and false positives and true and false negatives for a classification problem. Derivatives include specificity, sensitivity/recall, accuracy, receiver operating characteristic (ROC), area under the ROC curve (AUC) and F_1 score.
- Mean squared error: The average of squared errors in predictions for a regression problem. In linear regression algorithms, parameters are often selected by minimizing this metric. Similar metrics include root mean square error, mean absolute error, R-squared and adjusted R-squared.
- Logarithmic loss: A measurement of error for classification problems where the prediction is a probability between 0 and 1.
- Log-likelihood: A measure of the plausibility of the model parameters, given the observed data. Related metrics include the Akaike information criterion (AIC) and Bayesian information criterion (BIC). These metrics are only applicable for algorithms that assume a statistical distribution.
- Lift charts: A visual technique for showing the effectiveness of a predictive model. Different varieties of lift charts allow for comparison to a random guess or to another model, etc.

---

[65] Foreman-Roe, Scott. Understanding the Bias-Variance Tradeoff. June, 2012, *http://scott.fortmann-roe.com/docs/BiasVariance.html* (accessed February 14, 2019).
[66] Zinkevich, Martin. Rules of Machine Learning: Best Practices for ML Engineering. *Google.* October 24, 2018, *http://developers.google.com/machine-learning/guides/rules-of-ml* (accessed May 5, 2019).
[67] See footnote #8.
[68] See footnote #1.
[69] See footnote #35.
[70] Hastie, Trevor, Robert, Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. *Springer,* January 13, 2017, *https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf* (accessed January 23, 2019).
[71] See footnote #9.

- Actual versus expected analysis: A visual or tabular technique for showing the fit of a predictive model on holdout data, for example plotting the model prediction relative to the actual experience in the holdout at each policy duration. The analysis may be done across the range of each model feature and other values as well such as calendar year. Depending on the scale of the prediction, it may be helpful to consider the actual and predicted values, their ratios or their differences.

Performance metrics are typically selected based on the specific model type and application. The Basel Committee on Banking Supervision from the banking industry drafted an excellent example of how to select metrics for default models.[72] At the time of its publication, there were already many published references from which the authors were able to pull examples. For instance, they were able to say that a typical accuracy ratio for default models falls in the range of 50–80%, thus giving practitioners an idea of whether their model is within reasonably acceptable standards. Some lessons are more general, such as certain metrics that are useful for comparison but can never give a positive conclusion about the appropriateness of a model (e.g., AUC), or other metrics are tied to the number of observations in such a way as to prevent them from giving a positive conclusion (e.g., p-value). The authors proceeded to discuss testing the model's calibration (e.g., via the binomial and Chi-square tests), in essence assessing whether the deviation in experience can be attributed to random chance or whether there is bias in the model fit.

Unfortunately, there is currently little published discussing appropriate performance metrics for specific actuarial applications. Many actuarial applications will have parallel model structures from other industries and can leverage analyses done in those applications. For example, modeling defaults in banking is a binary classification problem, so many of the lessons learned from banking can be applied directly (with care) to a variety of actuarial binary classification problems such as mortality, morbidity, lapse (alternately retention), policy conversion, rider utilization, annuitization, fraud, etc.

This is an area where further research is needed to provide explicit examples for actuarial applications. Such a research endeavor should cover binary classification problems as well as other actuarial applications, such as risk classification, health plan selection, IBNR completion rates, agent performance, premium patterns, accelerated underwriting, pricing and claim amounts (all reported as current applications in our survey). As in the Basel paper, it should address, for a variety of model forms from contingency table to machine learning algorithm, as many of the following as possible.

- What tests are recommended?
- When is each test appropriate?
- What are the tests' limitations?
- Under what circumstances will a test yield inconclusive results?
- What is a "good" result for that application?
- How much data are required?
- How does the quantity of data influence each test?

[72] Basel Committee on Banking Supervision. Studies on the Validation of Internal Rating Systems. May 2005, *https://www.bis.org/publ/bcbs_wp14.pdf* (accessed February 14, 2019).

Credibility. In an actuarial context, an assessment of credibility may be required. ASOP 25 on Credibility[73] provides some guidance, with some wiggle room for predictive models.

Independent review. Many sources suggest an independent review of the modeling process, results, implementation and appropriateness of the approach. "A guiding principle throughout the guidance is that managing model risk involves 'effective challenge' of models: critical analysis by objective, informed parties that can identify model limitations and produce appropriate changes."[74]

Thus far, we have used the term "model evaluation" to encompass the quantitative steps taken to show a model's appropriateness. In contrast, the term "model validation" frequently occurs in model governance literature, where it is used to describe a broader effort. That said, the items we have discussed in this section include much of the content that should be covered in a model validation (as used in model governance). In 2005, the Federal Deposit Insurance Corporation (FDIC) issued a paper on model governance for financial institutions that offers criteria for model validation, which is summarized as follows:

- Clear and detailed documentation regarding the quantitative methods used, the reasoning behind the methods, how the model was tested, and the risk assessments used.
- Description of the entire backend process including data inputs, how the model operates and interacts with the data, what quality controls were used, how the code was assessed for accuracy, and how the model's usefulness was determined.
- An analysis of the expected outputs, the level of the model's predictive precision, any results of back-testing, and a comparison or benchmarking of the model with other similar models or qualitative research sources.[75]

2.7 MODEL DEPLOYMENT

"Predictive model deployment provides the option to deploy the analytical results in to every day decision-making process, for automating the decision making process."[76] Ideally, the model's final home was established at the beginning of the project and included the IT team (or other implementation team), which will ultimately be integrating the model into the larger technological infrastructure, if that is the objective. For actuaries, adopting a predictive model into production may mean a purely manual process or a fully automated process. For example, there may not be available solutions to integrate a predictive model automatically into an actuarial projection system, so its adoption will be largely manual. Alternatively, a real-time scoring application might simply need a new model to be uploaded after it is approved.

---

[73] Actuarial Standards Board. Actuarial Standard of Practice No. 25: Credibility Procedures. December 2013, *http://www.actuarialstandardsboard.org/asops/credibility-procedures* (accessed January 23, 2019).
[74] Federal Reserve Board of Governors. Guidance on Model Risk Management. April 4, 2011, *https://www.federalreserve.gov/supervisionreg/srletters/sr1107.pdf* (accessed January 23, 2019).
[75] FDIC. Model Governance. *Supervisory Insights*, 2005, *https://www.fdic.gov/regulations/examinations/supervisory/insights/siwin05/siwinter05-article1.pdf* (accessed January 23, 2019).
[76] PAT Research. What Is Deployment of Predictive Models? *https://www.predictiveanalyticstoday.com/deployment-predictive-models* (accessed January 23, 2019).

Questions to consider before the launch include the following:[77, 78, 79, 80]

- What are the requirements for the model in producing predictions? (e.g., Are you expecting to need predictions in a stream or batch, what latency is required, will it be built into a projection system, will it need to be able to run offline?)
- Who is receiving predictions from the model?
- How will the model's performance be evaluated in production and how often? Is there a backup plan if the model does not perform to expectations? Is there a plan to be able to quickly find the source of performance issues? Will resources be available postproduction for this?
- How frequently do I plan on retraining, refreshing or rebuilding my model?
- How will I collect, store and use additional data?
- Will the format of the production input data differ from the model training data? (e.g., Is it possible for predictor variables to take on values in the production environment that are outside of their range in the training data?)
- What are the data preprocessing needs for the incoming data?
- Is the approved model deployed directly from its development language? If not, is the code for the model in a replicable format for the software engineers, IT team or implementation team? How will I test that it was implemented as intended? Who owns which parts of the process? How are changes integrated into the process?
- How are software updates integrated into the process?

The model may have performed exceedingly well with the testing and training data. But when it is time to ingest real-time data from various sources and of various types, the likelihood of failure will increase significantly. A risk mitigation plan should be in place in case the model fails or quickly degrades under the increased load. In short, it is a necessity to have risk mitigation or contingency plans in place that account for both success and failure scenarios.

Models should be routinely monitored and adjusted periodically, because models do degrade and eventually fall below acceptable thresholds of accuracy, specificity, etc. Monitoring visualizations such as a model degradation lift chart can be used for this purpose.[81] Business objectives change, and new problems arise that may render the current model obsolete.

2.8 MODEL GOVERNANCE

Model governance ensures that the model works as intended, by enforcing documentation standards for each stage of the process, establishing version and access controls, and checking that the processes were followed as drafted. Even if model development, implementation, use and validation are satisfactory, a weak governance function will reduce the effectiveness of overall model risk management. The model governance function is formally established by the board and senior management on an enterprise-wide

---

[77] Godfried, Isaac. Deploying Deep Learning Models: Part 1 an Overview. *Towards Data Science*, June 19, 2018, *http://towardsdatascience.com/deploying-deep-learning-models-part-1-an-overview-77b4d01dd6f7* (accessed January 23, 2019).
[78] Brownlee, J. Deploy Your Predictive Model to Production. *Machine Learning Mastery*, September 30, 2016, *https://machinelearningmastery.com/deploy-machine-learning-model-to-production* (accessed January 23, 2019).
[79] See footnote #77.
[80] Open Data. How to Deploy Analytic Models into Production Successfully. January 30, 2018, *https://www.opendatagroup.com/blog/how-to-deploy-analytic-models-into-production-successfully* (accessed January 23, 2019).
[81] Best Practices for Managing Predictive Models in a Production Environment. *SAS*, 2007, *https://support.sas.com/resources/papers/proceedings/proceedings/forum2007/076-2007.pdf* (accessed February 13, 2019).

basis. [82] A key aspect is the "three lines of defense" corporate governance framework that a variety of financial institutions widely use to ensure that business operations, enterprise risk management functions and compliance functions coordinate model governance activities both seamlessly and effectively.[83]

- Business operations ("First Line"): Consists of the model owners, who develop, review and use the models on a regular basis.
- Control function ("Second Line"): Consists of the enterprise risk-management function, which provides a secondary layer of independent model review, validation, controls and risk monitoring.
- Compliance function ("Third Line"): Provided by the internal audit function, which assesses the overall effectiveness of the model governance framework and provides direction on potential improvements that can be implemented.

While each line operates separately, there are several common themes: effective challenge, enterprise consistency, and delineation of roles and responsibilities. These are discussed in the Federal Reserve's Guidance on Model Risk Management,[84] which is a valuable reference for practitioners working in financial institutions, including (but by no means limited to) insurance companies and banks.

This guidance states that there should be effective challenge of models, ideally by independent individuals that are subject matter experts with respect to the models that are under consideration. Enterprise consistency in the standards used throughout the entire process of developing and deploying a predictive model is an important goal. The guidance also indicates that delineation of roles and responsibilities should be formalized in the model governance framework. As the first line of defense, business operations are ultimately responsible for the risks inherent in the models that they use. Business operations provides support to the other lines for model validation activities, such as responding to requests for technical audits or clarity on assumptions and/or methodology. The monitoring and control of these risks is the responsibility of the control function (the second line), and this includes the role of independent model validation. It is important to point out that the role of compliance function (the third line) is not to duplicate the model risk-management activities performed by the first and second lines. Instead, its role is to provide assurance to the board and senior management by evaluating whether said model risk management is comprehensive, rigorous and effective.[85]

These principles apply generally to any model used for actuarial, statistical and/or financial applications. Specific to predictive modeling, SAS has developed the model governance system that employs an approach that is very similar to the three lines of defense approach outlined above.[86] Each component of the SAS Model Governance system is comprised of separate departments and carry clear-cut responsibilities to ensure the model's validity both in terms of statistical robustness and regulatory compliance.

---

[82] See footnote #74.
[83] Chartered Institute of Internal Auditors. Governance of Risk: Three line of defense. March 21, 2019, *https://www.iia.org.uk/resources/audit-committees/governance-of-risk-three-lines-of-defence* (accessed May 5, 2019).
[84] See footnote #74.
[85] *Ibid.*
[86] Maltesoz, Spyros. 8 Key Elements for a Solid Model Governance Framework. *SAS*, November 16, 2017, *http://blogs.sas.com/content/hiddeninsights/2017/11/16/model-governance-framework-mrm* (accessed January 23, 2019).

Last, a critical component of model governance is robust and readily available model documentation. Such documentation should describe all aspects of the model. Key considerations include clearly articulating the following. [87]

- Model purpose.
- Development of key assumptions and methodology.
- Sources of internal and external data.
- Evidence of technical peer review.
- Regression testing.
- Model limitations and disclosure of the domain over which the model should apply.
- Independent reviewer and management sign-off.

As a best practice, a model validation report should be sufficiently detailed that it can stand on its own without the need to access the underlying models themselves. Some companies often engage the third line or external parties to perform a gap analysis on the model validation report to identify any potential limitations of the work and where there is room for improvement in the next iteration of model validation. In addition, a model governance report would be more process-oriented with the intent to show that the company's model governance framework is being followed appropriately.

Model governance has been an active topic in actuarial circles in the past few years. In addition to the sources described earlier in this section, key resources include the latest exposure draft of the ASOP on Modeling,[88] the American Academy of Actuaries (AAA) Model Governance Checklist,[89] the AAA Model Governance Practice Note,[90] and the Federal Reserve.[91] They have had a lot to say, as have actuaries presenting at industry conferences. All of these resources are of great relevance, and maybe even more so for predictive models, because there is generally less understanding of predictive models, and they are a growing subset of models that actuaries use.

2.9 SOFTWARE SELECTION

The choice of software underlies all predictive modeling efforts. We have saved this for last, because the choice should not affect the validity of any predictive model. It is important to address, because it will impact nearly every stage of model development. There are generally two classes of software in regard to predictive analytics. There are data science and machine learning platforms like RapidMiner and Alteryx. They are software applications that would allow a user to perform data prep, modeling and visualization all from within the software product. Alternatively, there are programming languages such as R and Python, which support the development of software code that performs data prep, modeling and visualization.

[87] See footnote #51
[88] See footnote #53.
[89] See footnote #51.
[90] See footnote #52.
[91] See footnote #74.

Both Python and R are free,[92] open-source[93] programming languages. Additionally, both are continuously updated, and widespread community support and ample documentation are also free of charge.[94] SPSS[95] and SAS[96] do provide machine learning capabilities, but there are higher costs associated with these proprietary software programs.

The SOA's PA Exam specifically requires knowledge of R as deployed in the RStudio environment. According to the exam syllabus, candidates must be able to "write and execute basic commands in R using RStudio."[97] R is an open-source programming language based on the S language created by John Chambers and his cohort from Bell Labs in 1976.[98] After a 17-year evolutionary period and retooling by Ross Ihaka and Robert Gentleman from the University of Auckland, New Zealand, R was released to the public in 1993.[99, 100] Since then, it has gained tremendous momentum as one of the most popular statistical programming languages for advanced statistical analysis, including machine learning.

Through the last few decades, an active open-source community has contributed more than 13,000 add-on packages to enhance R's utility and machine learning functionality, e.g., caret package, neural networks and deep learning, recursive partitioning, boosting and gradient descent, and so forth.[101, 102] Through RStudio, an integrated development environment for R, these packages can quickly be downloaded via the Comprehensive R Archive Network.[103]

While the SOA exam focuses on R, Python is the No. 1 programming language for employers seeking data science candidates, according to a 2018 review of job postings.[104] Python is a general-purpose programming language frequently used in web and internet development, database access, graphical user interface development, network programming, and software and game development.[105] However, an active open-source community has contributed a large number of libraries to support data science and predictive analytics—NumPy, Pandas, Matplotlib, Scikit-learn, Keras, Tensorflow and NLTK are a few of the main libraries that are commonly used.[106]

---

[92] The Free Software Foundation, *https://www.fsf.org* (accessed May 5, 2019).
[93] The Open Source Initiative website, *https://opensource.org* (accessed May 5, 2019).
[94] Kappelman, Erik. 5 reasons you should learn R programming language now. IBM. February 20, 2018, *https://developer.ibm.com/dwblog/2018/why-r-programming-language-data-analytics* (accessed January 23, 2019).
[95] IBM. Create and Train a Machine Learning Model Without Coding. 2018, *https://www.ibm.com/cloud/garage/tutorials/ibm-spss-modeler/create-and-train-a-machine-learning-model-without-coding* (accessed January 23, 2019).
[96] SAS. SAS Visual Data Mining and Machine Learning. 2018, *https://www.sas.com/en_us/software/visual-data-mining-machine-learning.html* (accessed January 23, 2019).
[97] See footnote #7.
[98] R. What is R? *https://www.r-project.org/about.html* (accessed January 23, 2019).
[99] Theime, Nick. 2018. The R generation. *Significance Magazine*, 2018.
[100] Ihaka, Ross. R: Past and Future History. *The University of Auckland*, 1998, *https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf* (accessed January 23, 2019).
[101] Hothorn, Torstern. CRAN Task View: Machine Learning and Statistical Learning. *CRAN*, August 5, 2018, *https://cran.r-project.org/web/views/MachineLearning.html* (accessed January 23, 2019).
[102] Shitut, Neha. Know the Best Machine Learning Packages in R. *Analyticstraining.com*, November 30, 2018, *https://analyticstraining.com/machine-learning-packages-in-r* (accessed January 23, 2019).
[103] Datazar. How to Install and Include an R package. *R-bloggers.com*, February 8, 2018, *https://www.r-bloggers.com/how-to-install-and-include-an-r-package* (accessed January 23, 2019).
[104] Hale, Jeff. The Most in Demand Skills for Data Scientists. *Towards Data Science*, October 12, 2018, *https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db* (accessed February 13, 2019).
[105] Python. About Python. 2018, *https://www.python.org/about* (accessed May 5, 2019).
[106] Altexsoft. Best Machine Learning Languages, Data Visualization Tools, DL Frameworks, and Big Data Tools. *KDNuggets.com*, 2018, *https://www.kdnuggets.com/2018/12/machine-learning-data-visualization-deep-learning-tools.html* (accessed January 23, 2019).

In addition to the predictive modeling capabilities in R and Python, there are also mechanisms for these languages to interoperate. For example, the reticulate package provides an interface from R to Python modules, classes and functions. As mentioned earlier, the R package featuretoolsR provides an interface to the Python module Featuretools; the R package Keras[107] provides an interface to the Python module Keras while the R package Tensorflow[108] provides an interface to the TensorFlow library. Additionally, the Python package rpy2 facilitates calling R functions from Python.

There are many factors to consider when choosing which solution to use. A few of the biggest include how you want to structure analytics within your company, your budget and the functionality required.

- Core competency: A fundamental decision is whether to invest in developing a predictive analytics system or to buy a predictive analytics platform. The former requires staff with programming abilities.
- Cost: This includes the cost of the software, as well as the associated costs in human resources, computing resources, support and ongoing costs in each area. For commercial packages like SPSS and SAS, the software cost will be the base product and the various add-ons needed to achieve the functionality desired. As already mentioned, R and Python are free and open-source languages.
- Analytics: A full assessment of the functionality available in each commercial solution is out of the scope of this review but can be understood by referring to the websites for each. For example, SPSS and SAS document available add-on packages for different functionalities.[109, 110] The Gartner Magic Quadrant for Data Science and Machine Learning Platforms[111] is a helpful resource for understanding the relative capabilities of each vendor. R and Python, because they are not commercial, are not compared in Gartner directly. They provide a wide array of built-in data science and machine learning features and allow the user to download packages and libraries with specific functionalities (at no cost).[112, 113, 114]
- Visualization: The Gartner Magic Quadrant for Analytics and Business Intelligence[115] gives a parallel snapshot of commercial solutions for visualization. Python and R have excellent support for publication-quality graphics, e.g., ggplot2 for R and Matplotlib in Python.[116] Additionally, both can take advantage of popular JavaScript libraries including: HyCharts, leaflet, plotly, D3, WebGL, billboard, and dygraphs.[117, 118] Dashboard creation should be included in discussion of visualization

[107] CRAN. keras: R Interface to "Keras." April 5, 2019, *https://cran.r-project.org/web/packages/keras/index.html* (accessed January 23, 2019).
[108] CRAN. tensorflow: R Interface to "TensorFlow." April 5, 2019, *https://cran.r-project.org/web/packages/tensorflow/index.html* (accessed January 23, 2019).
[109] SAS. Advanced Analytics. *https://www.sas.com/en_us/solutions/analytics.html#view-all-products* (accessed January 23, 2019).
[110] IBM. SPSS Software. *https://www.ibm.com/analytics/spss-statistics-software* (accessed January 23, 2019).
[111] Gartner. Magic Quadrant for Data Science and Machine-Learning Platforms. February 22, 2018, *https://www.gartner.com/doc/3860063/magic-quadrant-data-science-machinelearning* (accessed January 23, 2019).
[112] CRAN. CRAN Task Views. *https://cran.r-project.org/web/views* (accessed January 23, 2019).
[113] See footnote #101.
[114] Bobriakov, I. Top 20 Python Libraries for Data Science in 2018. June 11, 2018, *https://medium.com/activewizards-machine-learning-company/top-20-python-libraries-for-data-science-in-2018-2ae7d1db8049* (accessed January 23, 2019).
[115] See footnote #111.
[116] Kan, C.E. Data Science 101: Is Python better than R? *Towards Data Science*, August 21, 2018, *https://towardsdatascience.com/data-science-101-is-python-better-than-r-b8f258f57b0f* (accessed January 23, 2019).
[117] Hill, Asha. 9 Useful R Data Visualization Packages for Any Discipline. *Mode Analytics*, May 18, 2018, *https://blog.modeanalytics.com/r-data-visualization-packages* (accessed January 23, 2019).
[118] Grolemund, Garrett. Quick List of Useful R Packages. *Rstudio Support*, January 8, 2019, *https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages* (accessed January 23, 2019).

capabilities as well. In R, Shiny[119] facilitates interactive web applications and dashboards. In Python, similar capabilities are available through Dash[120] and Bokeh.[121] Tableau[122] and PowerBI[123] each have the ability to create dashboards for end users unfamiliar with working in a programming language.

There are many possible options for software. We have focused on R and Python as the two most likely choices for companies that opt to develop a predictive analytics system, because they give good examples of how a solution can span the breadth of features desired for most predictive modeling workflows. Alternative options for data science programming languages include Julia[124] and MatLab. Vendors of data science platforms—including SAS, Alteryx, KNIME, RapidMiner, and H20.ai—are among the leaders, according to Gartner. Vendors leading in the visualization space include Tableau, Microsoft and Qlik.

[119] The Shiny From R Studio, *https://shiny.rstudio.com* (accessed May 5, 2019).
[120] The Dash by plotly, https://plot.ly/products/dash (accessed May 5, 2019).
[121] The Boheh, *https://bokeh.pydata.org/en/latest* (accessed May 5, 2019).
[122] Tableau, https://www.tableau.com (accessed May 5, 2019).
[123] The Power BI, *https://powerbi.microsoft.com/en-us* (accessed May 5, 2019).
[124] Kopf, D. Should Data Scientists Using R and Python Switch Over to Julia? *Quartz*, August 17, 2018, *https://qz.com/1360318/is-julia-a-good-alternative-to-r-and-python-for-programmers* (accessed January 23, 2019).

Section 3: Predictive Analytics Considerations

In this section, we have compiled our learnings from throughout the report development, including a review of existing literature, the case study and our survey of actuaries (highlighted in blue text), as well as the report authors' collective experience. We have organized the considerations by topic, which are roughly in the chronological order for a typical predictive modeling project. Although we include recommendations, they will not necessarily apply to every situation, and so readers should exercise professional judgment as appropriate. We have tried to focus on considerations and questions that can guide actuaries managing a predictive modeling project. We also encourage readers to reference the literature review, case study and appendix of survey results for more detail on specific topics.

There are so many ways to perform predictive modeling exercises that it would not be possible to provide a prescriptive list of instructions that would be reasonable for every application. Rather, we have organized this section around lists of considerations to be addressed for each distinct phase of predictive modeling. Our recommendation is that practitioners consider each item and document the reasons behind their decisions. Reasons that are acceptable for a first project may become dated over time, so having a documented record will enable a smoother transition as a group becomes more sophisticated or adds resources with different skills.

3.1 PROJECT OBJECTIVE

The most important step in any predictive modeling project is to carefully define the project objective. For our purposes, we are focusing on the "project" of building and implementing the predictive model, which may be part of a larger "project" for the company with a broader set of objectives. Defining the project objective can be a straightforward step, but it should not be overlooked. The main purpose is to ensure that all stakeholders are identified and recorded along with their unique requirements for the project's outcome. Objectives may be strategic (how can I decrease claims?), monetary (how much can I decrease claims?), or knowledge-based (what can I learn about patterns in claims?). Beware that knowledge-based objectives don't necessarily lead to actionable results. It will be easiest to claim success when the objective is carefully recorded from the outset and when the project's completion results in an identifiable and positive monetary impact. Not every project that is proposed will be feasible once all requirements are understood.

Actuaries use predictive modeling for a wide variety of applications. In our survey, there were nine different business areas where at least 10% of respondents indicated they had applied predictive modeling.[125] Even within the same area of work, each project is unique.

Below is a list of questions that company management and the predictive modeling team leaders should address before embarking on a new project. They should document the answers and make them available to all parties involved in the project at any stage. This is not an exhaustive list but should be a reasonable starting point for most projects. It is also worth noting that the project objective may need to be revisited throughout the project as more stakeholder input is gathered, team members gain a better understanding of the data, and strategic goals evolve.

1. What are the primary business objectives?

[125] See question 7 in Appendix 1.

2. How will this provide value to the business and customers?

3. What are the specific modeling objectives?

4. What is the organizational context? Who will be impacted? How much education will need to be delivered with the results of the project? What is the regulatory environment? Which ASOPs may apply (e.g., Risk Classification, Data Quality, Credibility, Actuarial Communications, Modeling Draft, Assumption Setting Draft, etc.)?

5. What data will be needed? What sources, timeframe, granularity, data fields, etc. are needed?

6. Are there existing limitations to accessing and using the data?

7. What resources are available (computing, modeling staff, data, subject matter experts, etc.)?

8. What is the contingency plan for unexpected delays during each of the model-building steps (e.g., acquiring data, preparing data, constructing the model, testing the model, and deploying the model into a production environment)?

9. How do we define and measure the model's success?

10. What is the end state? How will the model be used? Who will maintain the model in production and how?

3.2 DATA ACQUISITION AND PREPARATION

Acquiring, organizing and preparing data are critical steps in any predictive modeling project, and the time and cost for these steps can be quite high. The literature indicates that data scientists spend 50% to 80% of their time on data preparation. Projects often use both internal and external data sources, and even data that is internal to the company is often pulled from a different department. In our survey, 41% of respondents said they pulled data directly themselves, 25% said they requested it from a different area, and 34% indicated they did both.[126] Nearly 75% of respondents use external data. In our case study, the firm used internal data only but had to extract it from three separate data warehouses.

Once acquired and organized, data need to be reviewed thoroughly and often cleansed before they are ready for use in predictive modeling. Data cleaning includes adjustments to handle missing values, outliers and potentially erroneous records. According to our survey, removing incomplete observations and imputation were both common approaches for handling missing data, although nearly one-third of respondents indicated they had retrieved missing values from other data sources.[127] Similarly, actuaries reported using a combination of removal and capping for outlier values.[128] More than 15% stated they did

[126] See question 11 in Appendix 1.
[127] See question 21 in Appendix 1. Although not shown there, the results also indicated that 47% of respondents used *both* imputation and removal (depending on the situation).
[128] See question 22 in Appendix 1. Although not shown there, the results also indicated that 37% of respondents used *both* removal and capping (depending on the situation).

not adjust for outlier values. Practitioners should be sure to understand the reason outliers occur (misreported data, isolated events with a known cause, random variability) before determining the best approach for handling them. In our case study, the firm removed a small amount of data from its universal life (UL) model that could not be reconciled across data sources. On an ongoing basis, data should generally be reconciled not only to control totals but also to external sources, if possible. For instance, medical claims used in a predictive model should be compared to comparable totals from company financial statements.

This is an iterative process for which many steps can be automated, although more than 40% of respondents in our survey indicated they still have a largely or completely manual data preparation process, and less than 5% had fully automated the process.

At the start of this phase of the project, it can be useful to lay out key questions and considerations related to the specific project. Because this phase is iterative, the answers to these questions may evolve throughout the project.

1. What data are available internally and externally?
    a. Do internal data need to be enriched by external data to meet the project objectives? If so, how will that be done, and how will match rates affect the project?
    b. Is one data source considered the "gold standard" or "source of truth" in situations where the two sources do not agree?
2. Who will gather the data? Depending on the breadth of data sources, this may be one for several people.
3. Where will the data be stored once acquired and how will they be accessed?
4. What issues and challenges are anticipated, and what is the plan for addressing them?
    a. Were the data received the same as what was expected? Do record totals tie to source totals? Do formats match what is expected?
    b. Are there any duplicate records?
    c. Are the values received in each field reasonable, and are changes over time reasonable (with special attention to the target variables)? Are the key relationships among predictors as expected, and will their relationships affect algorithm selection (e.g., how will you handle correlated variables)?
    d. Are the relationships between the target variables and the potential predictors reasonable?
    e. Will all potential predictive variables be available at the time of prediction once the model is in production?
    f. How will you handle missing data? Will you impute values from relationships with other variables, impute them from other sources, remove records with missing values, or search for more data? How will you assess whether the missing data are themselves informative and/or suggestive of some insight?
    g. How will you handle outliers? Will you cap or floor extreme values, remove records with extreme values, impute new values from relationships with other variables or from other sources, or recognize the values as valid?
    h. How will you handle levels of categorical variables with little exposure? How will you handle any severe class imbalance issues?
    i. How will you recognize and incorporate new values in a variable (if in a repeated modeling process)?
5. What checks can be automated? How can the performance of automated checks be tested?
6. In subsequent iterations:
    a. Have any assumptions changed the way you will address future challenges?

   b. Can you automate more checks? Note that it may be helpful to revisit this again after feature engineering and selection. If you determine that you need more data than originally planned, how will you go about getting these, incorporating them, and adjusting the project objective?
   c. If you have identified data deficiencies or issues, how will you address these for future data consumption or by new models?
7. What will happen if the person familiar with the data leaves the company or the team? Are all steps and judgment calls made in the process documented?
8. What will happen if the data warehouse or underlying data source changes?
9. How will the team handle data security, HIPAA compliance, General Data Protection Regulation and other confidentiality requirements?

3.3 ALGORITHM SELECTION

There are myriad algorithms available within all the most common predictive modeling software, and selecting an appropriate algorithm for a given project is of utmost importance. Predictive algorithms should match both the prediction objective and the data available. There may be instances where an algorithm theoretically generates the correct type of prediction but is not suitable for the volume of data to be used; many machine learning algorithms can be quite slow with large data sets. The algorithm should also be practical; simplicity is preferred where possible. In our survey, the most common type of algorithm that respondents use was the GLM. Fewer than 15% of respondents reported using a deep neural network (DNN), despite the fact that a DNN is able to provide additional predictive power for some complex problems. In our case study, the firm chose a GLM for its UL model because of simplicity and transparency compared to the alternatives, including a random forest and neural network.

The prior Literature Review section includes more detail about matching the type of prediction to the appropriate algorithm, including discussion of supervised versus unsupervised learning and continuous versus categorical outcomes. Rather than address those type of technical considerations here, we instead suggest several recommended questions to address in the algorithm selection process.

1. What is your methodology for selecting an algorithm? Will you start with a small data set to test out multiple options expediently? How will you account for expected changes on a larger data set (for instance, longer run times or hardware limitations)?
2. What are the pros and cons of your candidate and selected algorithms? For example, what are their relative speeds for training and predicting? What are their underlying assumptions for the data? Will your data meet these assumptions, and if not, is it reasonable to proceed?
3. Does your selection fit within stakeholders' requirements?
4. Will your chosen algorithm allow you to maximize predictive accuracy relative to requirements for interpretability? If your selected algorithm is very complex, will a model explainer help to ease its adoption by other stakeholders? Can you achieve the same thing with a simpler or more interpretable algorithm?
5. How will you identify and document limitations (ranges of applicability, small sample sizes, etc.)? How will you effectively communicate them to other stakeholders and model users?

3.4 FEATURE ENGINEERING AND SELECTION

Even after cleaning, it is not appropriate to simply feed the available data into a predictive algorithm. In many cases, you should not use some variables to train the model or make predictions, and often there are additional variables that can be generated by combining, bucketing or transforming the original variables (feature engineering). The goal of this process is to increase the model's interpretability, reduce

redundancy and dimensionality, reveal additional relationships between the variables, and standardize as well as normalize the data. This can be an iterative process in that features are engineered initially, a subset of all available features is selected to be used in modeling, and then the model is tuned and calibrated, which may reveal the need for further feature engineering to improve performance. In our experience, extra time spent on feature engineering can yield better "return on investment" than any other aspect of the modeling process.

While there are automated tools that you can use for feature engineering, this process is still largely manual for many actuaries. Approximately 42% of survey respondents indicated they used machine learning algorithms for feature engineering, but 56% to 70% relied on either business/domain knowledge or previous experience. In the case study, the firm engineered eight new features for its UL model for a variety of reasons, many of which were a result of its understanding of policyholder payment timing.

Feature selection must be done with care and with attention to the strengths and limitations of the chosen algorithm. Many algorithms control multicollinearity, and techniques such as penalization or regularization can minimize adverse effects on accuracy; however, even in these cases, multicollinearity can still make it harder to interpret results. At least 40% of respondents to our survey used each stepwise feature selection, machine learning algorithms such as GBMs or random forests, and least absolute shrinkage and selection operator (LASSO) and/or Ridge techniques, although many experts advise against stepwise feature selection.[129, 130] Actuaries must also be cognizant of the regulatory environment, which often prevents the use of certain features for specific purposes (for instance, individual and small group health care premiums often cannot vary by health status). Approximately 37% of respondents indicated that they have excluded significant features from a predictive model because it was required by law.

Below are key considerations for the feature engineering and selection stages of a predictive modeling project.

1. What is your plan for feature engineering? How will you check that the features created are as expected? What parts can you automate, if any? How will your algorithm choice affect the type of data you create?
2. What steps can you take before feature selection to ensure a smooth process? Will you bin continuous variables to help identify the ideal shape of the relationship? Will you standardize all variables to help the interpretation of relative importance? What is your plan for capping outliers, handling missing values, etc.?
3. What is your plan for addressing collinearity of variables? For example, will you use clustering methods to identify potential problems? Will you use specific tests such as correlation matrices, scatter plots, variance inflation factors and principal components analysis? Will you choose algorithms that are robust to collinearity (GBM, random forest, regularized regression)?
4. What method will you use for feature selection? If the breadth (number of variables) of your data set is much larger than what you could potentially use as a final model, will you limit the number of variables you consider in your main selection process, and if so, how? Will you use LASSO and/or Ridge Regression, clustering, machine learning or another method? Will you rely on past models and experience? What is your reasoning?

---

[129] STATA. What Are Some of the Problems With Stepwise Regression? *https://www.stata.com/support/faqs/statistics/stepwise-regression-problems* (accessed January 23, 2019).
[130] Flom, Peter. Stopping Stepwise: Why Stepwise Selection Is Bad and What You Should Use Instead. *Towards Data Science,* September 22, 2018, *https://towardsdatascience.com/stopping-stepwise-why-stepwise-selection-is-bad-and-what-you-should-use-instead-90818b3f52df* (accessed January 23, 2019).

5. What limitations are implied by regulatory, legal or privacy considerations?
6. How will your model evaluation plans affect the preparation of your modeling data? Will you reserve a holdout data set and/or an out-of-time holdout data set for later testing? If so, how will you ensure and document that all feature selection decisions are made on the training data set only? If you plan to use *k*-fold cross-validation or another method that allows all data to be used for feature selection, have you documented your reasoning?
7. How will you verify that the data support your assumptions about the underlying relationships between features and the response, e.g., linearity? If this process involves creating new functional forms of variables via splines or piecewise linear splits of continuous variables or combining previously created bins, how will you incorporate the derived variables into your data preparation process going forward?
8. How will you identify and document limitations (ranges of variables present in training data set, credibility of model predictions, etc.)? How will you effectively communicate them to other stakeholders and model users?

3.5 MODEL EVALUATION AND MEASURES OF SUCCESS

Evaluating the predictive model's performance is a crucial step in any predictive modeling project but not always a straightforward one. Appropriate performance metrics will vary based on a number of factors, including the nature and distribution of the target variable, risks associated with inaccurate predictions, and broader business objectives for the predictive model.

You should select performance metrics and establish a robust process for measuring and quantifying the model's performance before attempting to optimize that model. Thus, because the guidance here is intended to aid in establishing a modeling plan, in practice you should address many of the concerns identified in this section should be addressed before feature engineering and selection. We have kept the sections in this order because the actual model evaluation will be done after the model has been fit.

With any predictive model, navigating the bias-variance trade-off is critical to avoid overfitting or underfitting. Models should be optimized on data that was not used to train the model; this can be done by holding out data from the training data set or applying other machine learning techniques such as cross-validation. Holdout data should be selected in such a way as to ensure its independence from the training data set. In our survey, the most common approach was to use a random seed to split training and holdout sets (72% of respondents), although *k*-fold cross-validation (52%) and using a time threshold to split training and holdout sets (45%) were also common. In our case study, the firm tested its UL model by comparing performance metrics on a holdout set and comparing to results produced by the firm's existing model, a classical, tabular-based model. It calculated the performance metrics on aggregated predictions and also at a more granular level.

There are a wide variety of metrics available for evaluating model performance, and some number of survey respondents used each of the options we provided. The most commonly used metrics among respondents were adjusted R-squared, actual to expected (A/E) ratio, AIC, R-squared, AUC, lift charts, and log-likelihood. We note that some metrics, such as AIC and log-likelihood, are only appropriate to models with an assumed statistical distribution.

Model evaluation techniques may be very similar from project to project, which can help stakeholders and the model's users to more easily adopt new models.

1. How will you ensure you do not overfit the model, balancing the bias-variance trade-off? What are the reasons you selected your methodology? Under what circumstances would you make a

different choice? Will you use holdout validation, cross validation, or an alternate method? If your choices resulted from having a small data set (e.g., using cross validation, or working without a test data set), how could future iterations differ if you are able to gather more data?

2. What metrics will you use to evaluate the relative performance among your candidate models? How are your chosen metrics preferable to others, and what are their limitations? Is there an existing model that will serve as a baseline from which to judge improvement? Is there an absolute level of your chosen metric that will qualify as success? If you change the algorithm or number of variables selected relative to a prior model, how will your chosen metrics compare between models?
3. What visualizations will you use to evaluate the relative performance among your candidate models? Is there an existing model that will serve as a baseline from which to judge improvement?
4. Will you address credibility of the model predictions, and if so, what measures of credibility will you use? Are you required to address credibility by regulation or by stakeholders? Which options under ASOP 25 on Credibility are relevant for your particular model?
5. Do the training and testing data sets reflect a diversity of scenarios? Does synthetic data need to be created to test the model against these scenarios?
6. What other considerations will help you choose between models? Prediction time, ease of implementation, maintainability, etc., may be relevant items to consider.
7. How will you measure the model's performance in production to attribute deviations to random variation or bias?
8. What are your business measures of success? How will you evaluate the model in terms of benefit to the business? How will this model help you reach your goals?
9. How will you communicate the model's value and limitations to all stakeholders? How could a narrative of the process and findings assist? How will you communicate the impact of any model changes?

3.6 MODEL DEPLOYMENT

In most cases, developing a predictive model is not a one-time project. After development, the model-building process will be documented and saved. The model will be deployed into the company's larger technological infrastructure, which may be called "productionalizing" or "operationalizing" the model. Depending on the model's use and availability of resources, models will be tested against emerging experience and refit as needed.

Actuaries responding to our survey reported most commonly that they checked their models against actual experience on an annual basis, with only 17% checking less frequently than once per year. Similarly, more than 50% of respondents indicated they refreshed their models (including parameter adjustments) once per year.[131]

In our case study, the firm's post-level term lapse model was implemented on a widely used asset/liability cash flow projection platform. A group of actuaries with the requisite technology experience headed this effort. In other situations, if actuaries who developed the model do not have sufficient programming expertise to deploy the model, software engineers from outside the actuarial department may be responsible for putting the model into production. Communication between teams is crucial to ensure the model is implemented as intended.

[131] See questions 35 and 36 in Appendix 1.

Below are key considerations for the major steps in the model deployment process.

1. Implementation
    a. How will you document the model and associated assumptions to communicate to users?
    b. How will you archive the model development process for retrieval at a later date, if needed?
    c. Will all the data that the model requires be available once it is deployed?
    d. How will the model be operationalized? Where will the model go, and who will be responsible for implementing it? How familiar are the recipients with the process for creating the model? How much disruption will the model create in existing processes?
    e. How often will you require predictions from the model? If you will access scores intermittently, do you want to pull them through an application programming interface (API)? Are there other ways to get batch predictions? If you will generate predictions on the fly (e.g., within an actuarial projection system), do the actuaries who will implement the model understand the inputs and parameters you have used to describe the model?
    f. How will you handle extrapolation? For example, if policy duration extends beyond what was in the training data, will you assume the same relationship extends to the new range?
2. Validation
    a. How will you check that the model is performing as expected? How will you monitor scores and performance against actual experience? How frequently will you do these checks? Who will perform the validation (and provide both effective and independent challenge)?
3. Updates
    a. How often or under what circumstances will you retrain the model on updated experience? Who will be responsible for retraining the model? Will it be the model's recipient, or will it be the team that originally built the model? How will changes be communicated and implemented?
    b. How will you recognize and handle new data that imply the conditions under which you originally fit the model are changing? How will that affect your plans for retraining the model?
    c. If your data size increases significantly, how will your software and hardware scale to handle that?
    d. If you find an error in your modeling process, how will you implement a fix to the model in production? How will you assess its impact locally and down the line?

3.7 MODEL GOVERNANCE

Successfully putting a model into production is an important milestone for a predictive modeling project, but it does not represent the end of the process. Model governance assures that business operations, risk management functions, and independent governance and assurance are collaborating throughout the model design, development, and deployment procedures.

Typical model governance procedures that other actuarial functions follow can apply equally well to predictive modeling functions. In the literature review, we highlighted resources that thoroughly describe model governance for actuarial models. For predictive models in particular, there is a need for additional attention to data governance, version control, model evaluation and model deployment.

Responses to our survey[132] indicate that there is considerable room for improvement in ways actuaries govern predictive models.

- Many predictive models that actuaries use exist outside of a formal model governance framework. Only 31% of respondents said their company has a modeling governance framework that applies to its predictive modeling work. Of the respondents whose companies did have a model governance committee, nearly half said that predictive modeling was not represented.
- Half of respondents said their company has no model inventory, and in companies that do have a model inventory, predictive models are currently reflected only some of the time. Among respondents, fewer than half reported that they have identified a risk level for their company's predictive models. Roughly 30% of respondents indicated there is no independent review mechanism for models.
- Version control is often a challenge. About one-third of respondents use version control software (such as Git), and most use filenames or file structure to handle versioning, while approximately 20% do not use version control. Only half of respondents said that predictive models are able to be retrieved easily from archive.

While this may sound dire, it should be understood within the context that the majority of respondents have fewer than five years of experience working with predictive models, and thus their organizations are likely in the early phases of adopting predictive models into their business practices.

The firm in our case study had a model governance framework with oversight by three separate committees. Its model governance included maintaining an inventory of models, version control practices, independent review of models and documentation of assumptions. However, based on the responses to the survey, this is not necessarily the norm across the industry at the present day.

Below are recommended questions to address model governance for predictive models.

1. Will your predictive models be recorded in your company's model inventory? How will you assess your predictive model's risk levels?
2. How does your predictive model fit into your company's model governance policy? What additions do you need to make to capture the unique aspects of predictive models related to data, version control, model evaluation and model implementation? Will you include a representative of your predictive modeling function or someone with predictive modeling experience on relevant governance groups?
3. How will you work with IT on data governance and establish a balance of responsibility and information sharing? Who will maintain data dictionaries, metadata and data quality assessments?
4. How will you approach version control to ensure that no unintended changes make their way to the end user? How will you control access to models and data? For those using R or Python (or another programming language), storing code in version control is widely considered standard practice.
5. If you are just getting started, what is a minimum requirement for model governance based on the model's risk level, and what is your plan for improving governance in the future?
6. Who will be responsible for each portion of model governance? How will business operations, risk management functions and independent governance areas be involved?

[132] See questions 37–46 in Appendix 1.

7. How will you audit compliance with your model governance policy and procedures?
8. How will you communicate the model's risks and/or limitations to relevant stakeholders?

3.8 SOFTWARE SELECTION

Software and programming language selection must be done at the beginning of any predictive modeling effort, such as when the predictive modeling team is established. If care is taken in selection and reasons are documented, the question can be revisited infrequently thereafter, and it should become clear more quickly if a change is desirable. In many cases, the programming language will dictate a smaller universe of potential software options. For some pieces of predictive modeling work, including many visualization tools, there are options that can be used across a variety of languages and platforms. In the considerations below, we refer to the suite of decisions that must be made as "the software."

Among survey respondents, R is the most prevalent choice in programming language, but Python and SAS are well represented, and there are others in use beyond those we asked about. There are some differences in prevalence when viewed by discipline; in particular, SAS is more commonly used in health than in other disciplines. Using a combination of languages for a single project is not uncommon; in our case study, the firm relied on Python for data collection and R for modeling in its universal life premium model.

There are many variables to consider when determining which software is most appropriate for a given predictive modeling project.

1. Core competency: Do you want to build your software solution (e.g., programming in R or Python), or do you prefer to purchase an application or framework that generates solutions that do not require programming?
2. Cost: What is your budget, and what is the project's scope? A proof of concept on a small data set may be done with no software cost, while establishing a data science center of excellence may require an enterprise system at great cost.
3. Analytics: What predictive modeling capabilities will you need (e.g., algorithms, distributed computing)?
4. Visualization: How do you want to deliver your data, models, and results? Who will want to view them? How interactive does the delivery mechanism need to be (e.g., a static report versus an interactive web site or dashboard)?

Other items with limited literature, but with ample anecdotal evidence, are important to include in your assessment.

1. Processing speed and distributed computing: How much data and what complexity of models do you plan to work with? What is the expected turnaround time for projects? How many iterations do you wish to complete within expected timeframes? If all projects are expected to be on small data sets that can be loaded into a local computer's memory, with room to spare, this may not be a concern. If certain projects are expected to reach into terabytes of data, you will need software that is capable of running on servers and in parallel.
2. Cross-functionality: What issues could there be at the stage of putting your model in production? With what other platforms or systems within your company will you wish to integrate your process? Do you have a preference for all functionality available in a single platform versus specific functionalities available only in distinct platforms? In many cases, it can be appropriate to use a

combination of software to accomplish different tasks, but using multiple tools may introduce an additional learning and maintenance burden.

3. Client or stakeholder standards: Do your clients or stakeholders require that you use or not use a particular software? For example, some companies may not be comfortable with open-source software, or some clients may request that a specific language be used because they have more comfort with it. Will a commercially supported version of open-source language be acceptable if the open-source community version is not (e.g., a Microsoft version of R)?
4. Availability of support for software: Do you require software that is commercially supported, or will your team be able to self-support by using online communities? Availability of paid support and depth of online community support vary between software options.
5. Existing knowledge within company or modeling team: Does your team have existing knowledge with one software option? Education can be costly to get an entire team onto a new software option, so this should be weighted accordingly.
6. Ability to hire additional qualified practitioners: Is there a more common software skill found among your typical hiring pool? Many colleges have a preferred software so that aspiring actuaries in that location will be more readily found with that skill than another option. The actuarial syllabus is expanding into predictive analytics, with a leaning toward R for now. This means experienced actuarial hires will be more likely than in the past to have knowledge of R. Outside of the actuarial community, Python is the most frequently taught language in universities.[133]
7. Are you comfortable with your ability to assess the quality of freely available packages for open-source languages such as R and Python? Do you have a process established to track the version changes for packages that are used (to ensure your team is always using a consistent version that operates as you originally intended)?

3.9 STAYING CURRENT

Data science is an ever-evolving field. There is no substitute for a solid foundation in statistical methods. We have referenced several key textbook sources in this document such as *Applied Predictive Modeling*, *An Introduction to Statistical Learning: With Applications in R*, and *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. To keep pace with advancements in data science, a plethora of resources is available for continuing education. The following list, while not exhaustive, highlights key references in a variety of flavors for predictive modeling groups to follow:

- Web resources run the gamut from educational material to providing programming support to supporting data science competitions to blogs that track new ideas.
    - **Coursera:** Coursera is an online learning platform founded by Stanford professors Andrew Ng and Daphne Koller that works with universities and other organizations to offer online courses in a variety of subjects, including mathematics, computer science, data science and others.[134] See www.coursera.org.
    - **DataCamp:** DataCamp is an interactive website offering both free and paid courses in importing data, data visualization, machine learning, etc., in several programming languages.[135] See www.datacamp.com.
    - **StackOverflow:** This is the preeminent programming Q&A site. See stackoverflow.com.

[133] TIOBE. TIOBE Index for January 2019: January Headline: Python Is TIOBE's Programming Language of the Year 2018! 2019, *https://www.tiobe.com/tiobe-index* (accessed January 23, 2019).
[134] Coursera, *https://en.wikipedia.org/wiki/Coursera*.
[135] DataCamp, *https://www.linkedin.com/company/datamind-org/about*.

    - **Kaggle:** Well-known for its data science and machine learning competitions, Kaggle has a multitude of different data sets for practice, in addition to revealing how competitors approached finding a solution for the proposed business problems and intended outcomes (Kaggle Kernels). There's no need to enter a Kaggle competition to peruse and learn from its wide selection of blogs and predictive analytics techniques. See www.kaggle.com.
    - **R-bloggers:** For all things R-related, R-bloggers offers news and tutorials that will keep R users updated with the most recent release of packages and how to use them in a variety of contexts. See www.r-bloggers.com.
    - **KDNuggets:** Almost every aspect of predictive analytics and data science is covered via KDNuggets. This site has a comprehensive list of data sets, tutorials and blogs dedicated to honing data science and machine learning expertise. See www.kdnuggets.com.
- Community: Involvement in the data science community is a great resource for keeping up to speed. This can be through attending local Meetup groups, such as UseR, or by simply making friends in the industry.
- RSS feeds: As an alternative to visiting websites every so often, you can set up an RSS feed via a browser extension that will capture website updates and additional data science-related information sources such as Machine Learning Mastery,[136] The Morning Paper,[137] DataTau[138] and Elite Data Science.[139] *Data Science Weekly*[140] is another option for weekly data science news and articles that will be delivered directly to a preferred email address.
- SOA Predictive Analytics & Futurism section: Actuaries involved in predictive modeling share knowledge through newsletters, webcasts, podcasts, sessions at industry meetings and LinkedIn. See www.soa.org/sections/pred-analytics-futurism/pred-analytics-futurism-landing.
- Social media: LinkedIn and other social media sites allow users to follow topic areas of interest.

[136] Machine Learning Mastery, *https://machinelearningmastery.com*.
[137] *The Morning Paper*, *https://blog.acolyer.org*.
[138] DataTau, *https://www.datatau.com*.
[139] Elite Data Science, *https://elitedatascience.com.*
[140] *Data Science Weekly, https://www.datascienceweekly.org*.

Section 4: Case Study

4.1 CASE STUDY BACKGROUND

The considerations offered in this paper represent best practices. We understand, however, that organizations operating within the insurance industry have analytics functions that vary by maturity and face a diverse array of business challenges that may lead them to be industry leaders in some areas (hence, inspiring the considerations presented in this report), while falling behind in other areas compared to their peers.

As part of this research project, Milliman worked with an insurer (henceforth referred to as "the firm") that was willing to participate as a case study. This firm is a life insurance company that started building predictive models in 2013. The firm has requested that this report respect the confidentiality of its identity.

Our goal for this case study is to illustrate one company's experiences in developing predictive models and establishing a model governance framework in comparison to best practices and to provide recommendations on how it could improve in areas where it falls short. We will conclude each section with commentary on how the firm's modeling practices relate to the considerations presented in this report.

The firm agreed to discuss the details of two of its predictive models—one model in development and another in production:

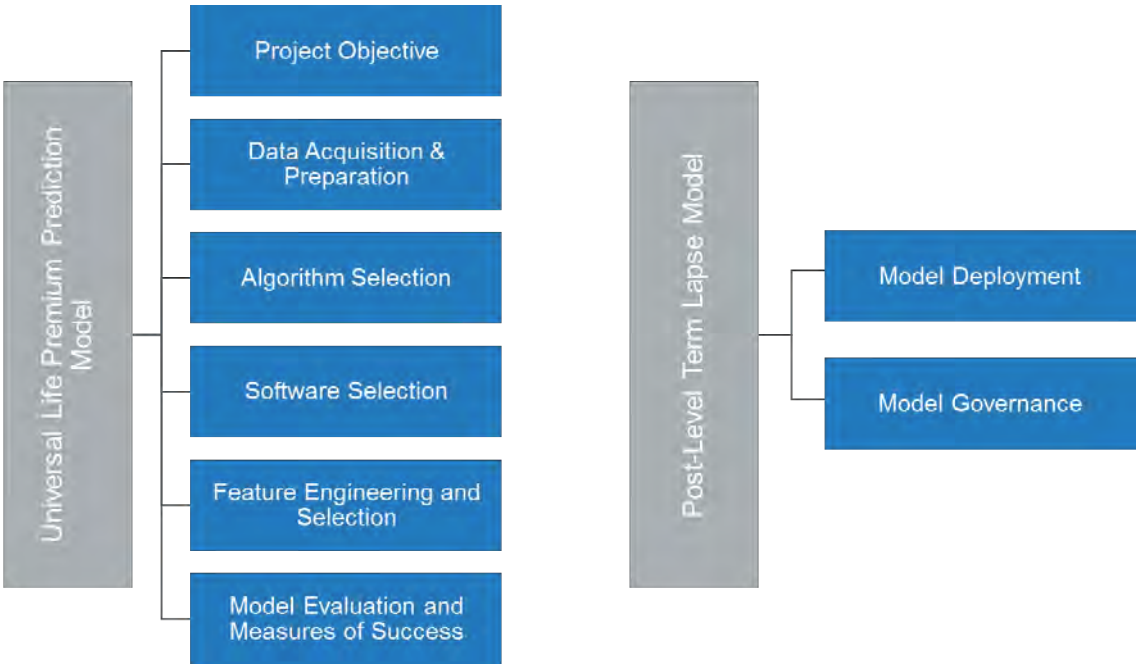
1. Universal life premium model (currently in development): A model designed to forecast premium payments on a book of UL policies.
2. Post-level term lapse model (currently in production): A logistic regression model designed to predict the annual lapse rates after the level-premium paying period on the firm's level-term business. Large lapse rates typically happen at the end of the level premium period for term life insurance policies due to large increases in premium.

The firm agreed to share the model development stages of the universal life premium model, because the development staff members were available to participate in the study, and they believed it would be beneficial to have a third party review the model. However, because this research project covers the entirety of the predictive modeling process, a second model was needed to cover the post-implementation stages of predictive modeling. We, therefore, came to an agreement to use the term lapse model for this purpose.

By discussing two models, we believed the case study would offer insights on model governance—that is, how the firm manages different model types and how the firm structures its governance framework to maintain existing models while offering the flexibility to incorporate new models in the future.

Thus, the following sections of the case study are structured as shown in Figure 1.

Figure 1: Structure of Case Study



Note that these sections are analogous to the sections offered in this report. The UL premium model will cover the following stages of building a predictive model: project objective, data acquisition and preparation, algorithm selection, software selection, feature engineering and selection, and model evaluation and measures of success. The term lapse model will cover the model deployment and model governance phases.

4.2 PROJECT OBJECTIVE

The UL premium model's project objective was to forecast premium payments on a book of UL policies. Premium payments are uncertain because policyholders may choose, at any given time, to either alter their typical payment amounts or to not contribute a payment amount at all.

These predictions are important because they can impact the company's profitability.

- Minimum premium levels are expected to fund the policy. A policyholder who does not pay at least the minimum premium will be more likely to lapse based simply on the policy's structure.
- Higher lapses can indicate higher mortality on the product overall because it is usually the healthier lives that lapse, because they can acquire other coverage.
- The level of premiums will also impact the amount of money to be invested and potentially the total return a company can achieve.
- They may also impact management and marketing decisions. It is possible that poor decisions could be made to the extent that the premium projections are inaccurate.

After setting this objective, the modelers had to create a plan to achieve it. Because they desired to forecast premium payments, a logical next step would be to hypothesize what policyholder attributes and

behaviors could be predictive of premium payments and, from there, to determine what data sources would best represent those attributes.

4.2.1 COMMENTARY

The firm defined a clear business objective pertaining to its UL modeling project—that is, to predict premium payments on a book of UL policies. It wants the model to be interpretable, to perform better than the current model, and to help the firm discover and understand previously unknown policyholder behaviors. As the considerations offered in this report suggest, the identification of available data is important and addressed in the next section. The firm also established a plan to measure the model's success, discussed in the model evaluation section. The project was well-understood by the modeling team and the members of the governance framework overseeing the project. As we will discuss later, this governance framework consists of a diverse group of professionals within the company, including stakeholders and executive management, who were responsible for approving the project. Having buy-in from all relevant parties is crucial to preventing operational failure during the project's latter stages. In the firm's case, the modelers met with managers of the modeling platform, authors of experience studies, their reinsurance management team and the chief actuary throughout the course of the project to discuss business needs and progress on the project.

4.3 DATA ACQUISITION AND PREPARATION

4.3.1 DATA SOURCES

The firm used three internal data sources for its analytical projects:

1. Legacy data store (LDS): The firm's IT department produced this data store specifically for the use of premium studies. It consists of large, delimited text files. This data store contains premium payments, account values, and loan values for all UL policies between the years 2000 and 2011.
2. Current data warehouse (CDW): The firm's IT department produced this data warehouse to support a web application that company underwriters use to view policy information. This data warehouse is a SQL database that contains the premium payment for UL policies beginning in 2010. Because this database supports the business's live operations, the actuaries conduct analytical work using a backup database that is refreshed daily.
3. Reserving data warehouse (RDW): The firm maintains this data warehouse to support its reserving activities; the warehouse contains UL policy data from 2004 and onward, including planned premium.

The actuaries join the data from the three systems by using a combination of policy number and date. This composite identifier of policy month becomes the unique identifier of the aggregated modeling data set.

The firm explored the possibility of incorporating external data to complement its internal data. It was hypothesized that macroeconomic factors such as interest rate spreads, market indexes, and unemployment rates, along with customer propensity-to-pay data and other socioeconomic variables, could potentially be significant in predicting future premium payments. However, the firm opted not to use external data due to the expense involved in acquiring the data, as well as the labor time and complexity involved in joining third-party data sets to internal data.

Although external data were excluded from the analysis, the firm was open to the idea of incorporating such data in the future as a possible enhancement to the model.

4.3.2 DATA RECONCILIATION

Operational risks that may arise when joining data from multiple systems are inconsistencies from different fields, missing values and time horizons. To reconcile the data, the actuaries used the policy month to join the LDS and RDW tables together and made a comparison between the amounts paid, account value and loan amount. For these three fields, there was essentially no conflicting data, and very rarely were there missing data. A consistency check discovered that approximately 0.2% of the policies needed further investigation. Because this consisted of such a small portion of the data set, the firm removed them from the study.

The actuaries used LDS as the primary data set; they used RDW to validate that data; and they used CDW to expand the study window forward to more recent calendar years.

In addition to the above data sources, the firm considered a policy administration system—an IBM mainframe containing the official policy data. However, these data were not conducive to statistical modeling because the firm could only pull records one at a time. Therefore, the firm's actuaries determined the mainframe to be impractical to use as a modeling data set. While current efforts are in place to make the administration system more suitable for analytical work, it will likely take a few years to accomplish. The modelers did use the administration system to perform spot checks against the modeling data set. The data were reconciled for the policies examined.

4.3.3 COMMENTARY

The firm faced the challenge of merging three heterogeneous data sources, each of which spanned a different timeframe and were originally intended for analyses other than building predictive models. The firm's data warehouses are external to the actuarial and predictive modeling area, so the team worked with the data owners to acquire the data needed. The firm noted only minor discrepancies in the reconciliation process and removed the questionable data. The use of only internal data also reduces the risk of quality and consistency issues that external data presented. We believe that the modelers made a reasonable effort, given the resources available to them, to ensure that the merged data set was accurate, complete and free from material error. This is a critical step, because data quality is foundational to the success of any predictive modeling effort.

The firm explored the possibility of incorporating external data but ultimately chose not to. The decision to first build a model from internal data and then determine whether external data are needed is easily justified; external data may not be necessary or worth the cost if a good enough model can be built using only internal data.

4.4 ALGORITHM SELECTION

4.4.1 CANDIDATE ALGORITHMS

The firm considered the following options for its UL premium model: generalized linear models, mixed models, neural networks and random forests. When selecting a particular algorithm, the firm wanted to balance three criteria:

1. Predictive performance

Ideally, the model in development should make better predictions than the model that is currently in production. The better predictions support business decisions that upper management makes, as well as inform them of previously unknown policyholder behaviors.

2. Model interpretability

   Senior management will not accept models that are not explainable to internal and external stakeholders. Stakeholders include investors, regulators, auditors and internal business leaders who must implement the models. While complex models such as neural networks and random forests may offer increased predictive performance, they are more difficult to explain and thus stakeholders and regulators are less likely to approve.

3. Model practicality

   For a model to be successful, it must be practical to implement. Software engineers (or in this case, actuaries) must be able to faithfully replicate it in the firm's production systems. Furthermore, the variables used in training the model must be available in a live environment when the production system is ready to make a decision. In the firm's case, the models are constrained by their ability to be implemented in the modeling platform.

With these considerations, the firm opted to use generalized linear models, mixed models and mixture models as potential candidates for production and senior leadership's acceptance. The firm deemed more sophisticated models, such as the neural networks and random forests, to be too complex and opaque for adoption. However, the firm is using these models for experimental purposes to test the potential for improved performance.

4.4.2 CONCEPTUAL DESIGN AND ALGORITHM SELECTION

To build the UL premium model, the actuaries had to understand the process by which premium payments were made. A simple hypothesis would be that because the response variable (premium) is a continuous quantity, a single regression algorithm could model this. However, another way of interpreting the process is that a series of events or decisions could occur, leading up to the target event (premium payment). Each of these events may differ substantially in nature, and, therefore, a prediction based on a combination of models (with one model for each event or decision) may capture more information than a single model.

Because UL premiums involve a chain of events, the modelers determined that multiple models, one for each event, were required. To determine what models were appropriate, the team outlined the premium payment process with a tree diagram, with each node representing an event and corresponding model. The team combined these models to make the final premium prediction.

The following paragraphs describe this process within the context of the UL product and the resulting models.

UL is a permanent life insurance product that combines death benefit coverage with funds in something similar to a savings account to which interest is credited. Policyholders pay premiums that earn interest on a tax-deferred basis to provide cash value. The policy remains in force so long as the cash value in the account is sufficient to cover the cost of insurance charges. In addition, some product designs may have guarantees that ensure the policy will not lapse (even if the policyholder stops making payments) so long as certain contractual conditions are met. Although the mortality margin is the primary contributor to profitability (i.e., the cost of insurance charges that the firm collects exceed the death benefits the firm

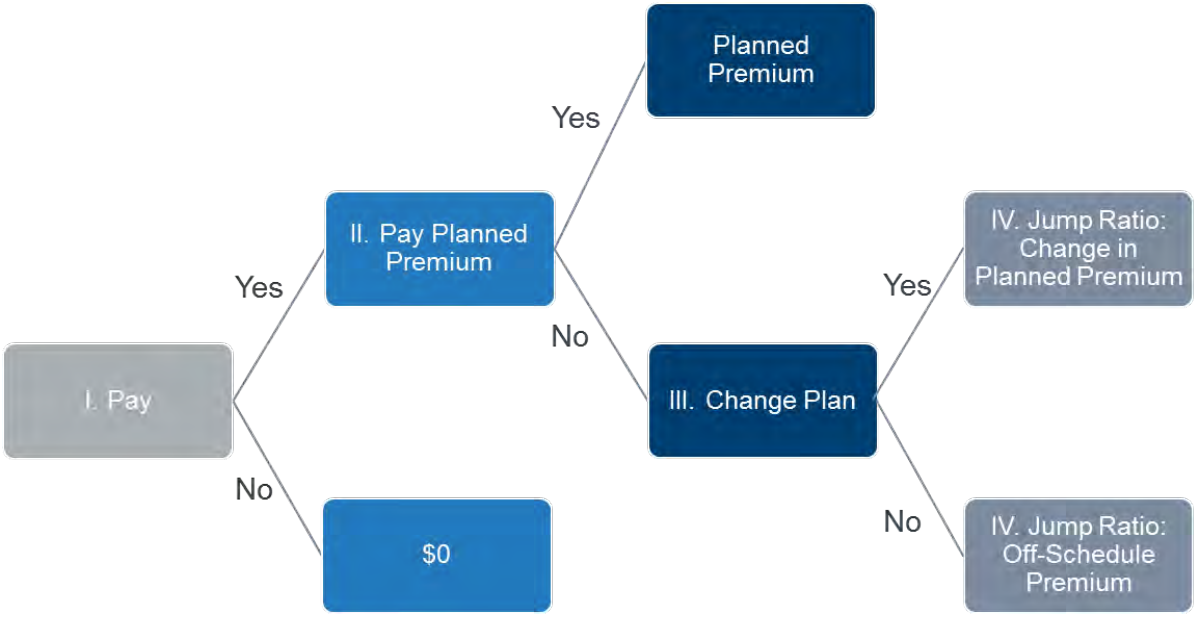
expects to pay), lapse rates also heavily influence profitability. When a policy lapses, the firm no longer earns profits on that policy. Furthermore, the firm's actuaries have observed that small adjustments to their premium payment assumptions have historically had large impacts on modeled profitability. Therefore, the firm has an incentive to build a model to more accurately model premium payments.

The modeling team determined three possible outcomes for the premium payment process:

1. No premium payment.
2. A premium payment equal to the planned periodic payment.
3. A payment not equal to the planned periodic payment.

These outcomes are represented in Figure 2.

Figure 2: Outcomes to Premium Payment Process



I. Pay: The probability of payment, conditional on the policy being in force through the month.
II. Pay planned premium: The probability that the premium deposit is the planned modal premium, conditioned that a payment is made.
III. Change plan: The probability that a change is made to the planned premium, given that a payment is made that is not the previous planned premium.
IV. Jump ratio: The ratio of the premium paid to the planned premium, given that the premium paid is positive and not equal to the planned premium. When Change Plan = 1, this corresponds to a change in the planned premium. Otherwise, this is an off-schedule premium amount.

The final predicted premium is calculated via the sum of a probability-discounted premium from each path in the decision tree. The firm selected generalized linear models for each step. The specific variations are withheld, per the firm's request. Although this model only predicts premium payments one month in

advance, the firm's goal is to eventually use the model as a basis for predicting premium payments for months or even years further out into the future.

4.4.3 COMMENTARY

Generalized linear models give the firm a good balance between predictive power and interpretability. Two of the variations the firm considered—mixed models and mixture models—add to model complexity and can decrease interpretability but can offer important benefits. For example, the modeler at the firm noted, "It is a good idea to test [policy ID] as a random effect when repeatedly observing the same subjects (panel/longitudinal data). I've also tested calendar month and other variables. If the estimate (of variance) of the random effect is sufficiently small, one can exclude it from the model."

The firm did a thorough job of documenting the premium payment process, as evidenced by the observation that nonlife actuaries among the authors were able to understand the process despite not having a background in life insurance. By using a series of models to effectively break down the premium payment into three sizes ($0, planned, other), the modelers not only achieved their goal of accurate premium predictions but also achieved a deeper understanding of consumer premium payment behavior.

A disadvantage of this method is the added layers of complexity and analysis that the four submodels bring. However, looking at two alternatives indicates the trade-off may be well warranted. A potentially more expedient alternative would be to combine the first three models into a single multinomial model, rather than a series of binomials, because each premium payment necessarily falls into one of four categories ($0, planned, other with change in plan, other without change in plan). However, it can be more difficult to interpret how a multinomial model works than to understand three layered binomial models. Another option is to model the premium amount directly with a single model; but this is unlikely to have a familiar distribution if there are higher densities at $0 and at the planned premium.

The remaining complexity is in implementation, which will be addressed as the firm moves forward with this modeling effort.

4.5 SOFTWARE SELECTION

For the project, the firm modelers used a combination of Python and R, two open-source languages widely used in the predictive analytics community. Python was used for data collection efforts, whereas R was used for modeling. Although the team had the option of using a single language to script the entire workflow, it chose to use two separate languages because the person responsible for collecting data was more comfortable, based on experience using Python, whereas the person responsible for modeling preferred R.

The team opted to use several R packages to facilitate its modeling effort: data.table, DBI, RODBC, rSQLite and the tidyverse packages were used for data import, data manipulation and data visualization; rtensorflow was used to run experimental neural networks; rmarkdown was used for model documentation and facilitated the team's goal of implementing reproducible research. We encourage readers who are interested in these packages' capabilities to read their documentation, which you can find online.

To implement version control, the team used Git, an open-source version control system (VCS) developed by Linus Torvalds, who also pioneered the Linux operating system. Git stores all the project's files in a folder known as a repository and enables a team to keep all committed changes that have been made to the project, known as commits. The VCS allows the team to track changes throughout the entirety of the

project lifecycle and to rollback or perform experimental coding (known as branching), if necessary. In addition to versioning, Git facilitates distributed development (especially for geographically dispersed teams), bug tracking, and conflict management. Due to the small nature of the team, the team primarily used Git for versioning, and the modelers found it unnecessary to make use of its branching and merging features.

It is important to balance the benefits of using certain tools with considerations around intellectual property, security, support and maintainability. The firm is comfortable that these languages are mainstream in the predictive analytics area and that these considerations have been sufficiently addressed through documentation and governance.

4.5.1 PROJECT STRUCTURE

The project file structure consists of a Git repository with six subdirectories, as summarized in Figure 3 below:

1. Sources

   This folder contains data external to the project (but internal to the firm). These data are typically Excel files, delimited text, etc. Note that this folder excludes large data sets such as those that could be stored on databases hosted on internal company servers (or on those of the cloud hosting provider, if the company uses cloud computing).

2. Data

   This folder contains the intermediate results of data manipulation scripts applied to internal data sources. While a project can, in theory, be conducted without storing copies of data pulls on disks, they can sometimes be helpful—because they can speed up development time by having to avoid multiple reruns of early stage queries (especially when experimenting with query parameter changes). For example, in a sequence of five queries, you can experiment with making changes to the fifth query without having to rerun the first four every time a parameter is changed.

   Another reason the team maintains a cache of intermediate data frames is facilitated debugging. This is especially important during the testing phase prior to model implementation. During this stage, practical errors such as missing data, the emergence of new fields, or structural changes to the underlying data sources can cause models to break down. Being able to examine the results of intermediate data transformations may make it easier to identify the precise cause of the issue than a cryptic error message.

3. R

   This folder contains the R scripts. The R code is broken up by task—for example, there are separate scripts for data importation, modeling and validation. Separating scripts by task as opposed to having one long script makes the code easier to read and maintain. Commonly used functions are included with R packages—which have the advantages of being reusable across projects and being distributable between modelers.

4. Python

This folder contains the Python scripts that were used to conduct the data pulls from internal company databases.

5. Excel

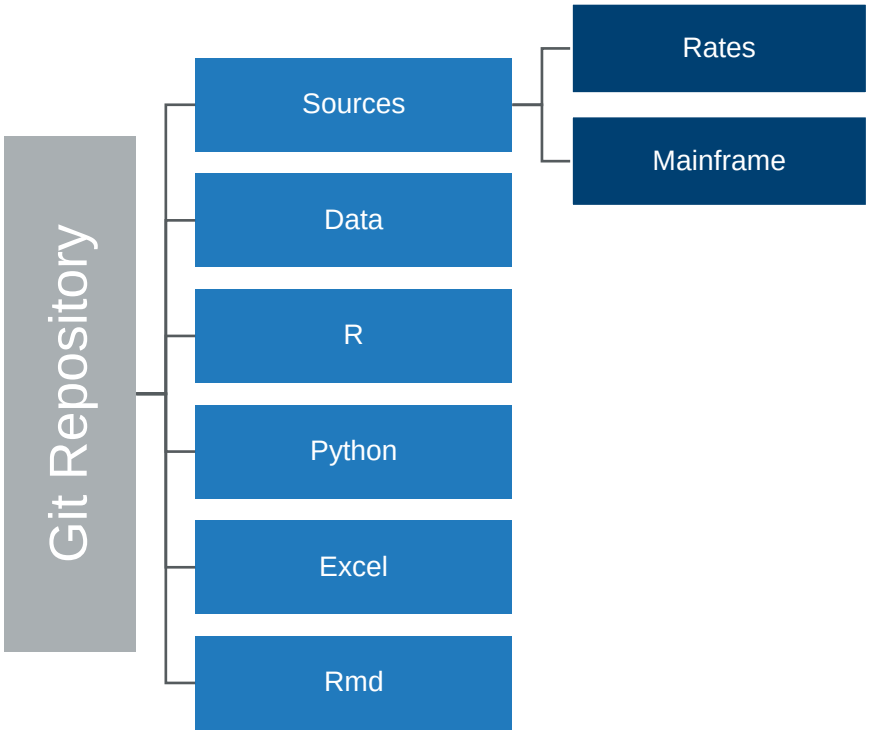
   This folder contains Excel files used for ad hoc analyses.

6. Rmd

   This folder contains R Markdown scripts. An R Markdown script is a type of file (suffixed .Rmd) that combines documentation using the Markdown markup language and statistical computation using the R programming language. R code is specified within the document either via inline scripts or code chunks. To transform an R Markdown script into a document, the programmer compiles the script via RStudio, which renders the Markdown code and executes the R code. The R code embedded in the script may have the ability to pull straight from internal databases, with the calculated output printed onto the document. The final output of the script can be a PDF document or web page.

   R Markdown is an elegant implementation of reproducible research. This project is reproducible because the code used to document the project also contains the code used to conduct the analysis. This level of integration reduces the amount of inference and guesswork that an independent reviewer may otherwise have to make when trying to reproduce a peer's work.

Figure 3: Project Structure



4.5.2 COMMENTARY

The modelers have opted to build their solution with open-source tools rather than purchase software. Open-source tools, by nature, allow anyone who is interested to inspect the underlying code understand how calculations are being made, which aids the firm in its goals for transparency and interpretability. Furthermore, the firm is ahead of the curve in comparison with survey respondents with respect to versioning (see survey question 44 in Appendix 1) and documentation (survey question 46). Not only does the firm use open-source tools to build its models but also to render the documentation describing those models, enabling reproducible research of the entire workflow. The firm has not found a need to go as far as building dashboards to present documentation in an interactive fashion, though this capability is available in either Python or R, should the firm see the need in the future.

The firm has elected to use a combination of Python and R, with specific reasons for doing so. We simply recognize that there is a trade-off between having a workflow in a single programming language versus having team members work in the language in which they are most comfortable. If they incorporate actuaries (who have gone through the new SOA exam) in the modeling team in the future, those actuaries may join with some experience with R and therefore be able to get up to speed more quickly.

Looking forward, we note that because the modelers are working with more than 100 million observations, they may wish to explore the use of parallel computing functionality in their chosen programming languages to enable faster iteration.

4.6 FEATURE ENGINEERING AND SELECTION

The source data included financial variables related to the account, variables related to customer payment behavior, details on the product, and customer details. Once these data were acquired, the modelers' next step was to perform feature engineering—that is, to create new variables from combinations of existing variables. As illustrated below, certain engineered features may have stronger predictive performance, as well as a more intuitive explanation behind the response variable, than its individual component variables.

- Monthly_Charges: Monthly charges include administration fee, cost of insurance and per thousand expense loads. They exclude riders. Earlier versions of the model included percentage of premium charges. The model has been retrained excluding these charges. While the model in development is only designed to forecast the next month's premium payment, the firm desires to eventually forecast premiums multiple months in advance. Excluding the percentage of premium makes forecasting more than one month much more tractable and does not materially impact fit or precision.
- Funded_Ratio: This is a metric calculated from the account value and monthly charges, and it is used to determine how well funded an account is.
- State: Either regular-paying (RP), intermittent-paying (IP) or not-paying (NP). If there are 13 or fewer observations, the default state is RP.
    RP: The policyholder paid the premium in the last two months and the premium collected in the prior year is at least half of the planned premium amount.
    IP: The state is not RP, the planned premium amount is positive, and the premium collected in the prior year is at least half of the planned premium amount.
    NP: The state is not RP or IP.

The full training data set was large (approximately 100 million observations); the modelers initially modeled only policies without loans with a monthly planned payment frequency. This data were split into 15 batches using stratified sampling, each of which contained about 3.6 million observations.

For the purposes of keeping this paper at a manageable length, we illustrate the firm's feature selection procedure by focusing on the first node of the premium payment process: the probability of payment, which is modeled by logistic regression with random effects. The firm's procedure is iterative and parallels the scientific method:

- Hypothesis: The modeler makes a hypothesis.
- Prediction: The hypothesis is translated into a model.
- Testing: A regression is performed. Success of the prediction is determined by model fit, measured by negative log-likelihood (NLL) for this model.

The modelers began with the base (null) model and subsequently added variables that they hypothesized would be predictive of payment behavior.

Modelers first made the hypothesis that policies that are well funded are less likely to pay premium, where the policy's funding status is measured by Funded_Ratio. A senior actuary with expert judgment suggested this variable.

Base Model: P(Pay) = 1/(1+exp(-η)), η = 1.710445. NLL: 1,540,701.

Model 1: P(Pay) = 1/(1+exp(-η)), η = 1.66682267 + 0.00038814 * Funded_Ratio. NLL = 1,539,993.

Coefficients	Estimate	Standard Error	P-Value
(Intercept)	1.66682267	0.00186506	< 2e-16
Funded_Ratio	0.00038814	0.00001054	< 2e-16

Model 1 shows a small yet statistically significant positive correlation between Funded_Ratio and the probability of payment. The improvement to NLL is quite small. The size of the coefficient estimate is also small. The modelers conducted further analysis to find that there are a (relatively) small number of customers with large funding ratios. The modelers thus adapted their hypothesis to use a log-transform of the funded ratio:

Model 2: P(Pay) = 1/(1+exp(-η)), η = 1.3307445 + 0.0977192 * log(1+Funded_Ratio). NLL = 1,535,759.

Coefficients	Estimate	Standard Error	P-Value
(Intercept)	1.3307445	0.0039985	< 2e-16
log(1+Funded_Ratio)	0.0977192	0.0009758	< 2e-16

This model is superior to Model 1 by likelihood.

However, the results are contrary to the original hypothesis with respect to a relationship between the probability of pay and the funding status. The modelers predicted that there may be an omitted variable. They tested a new hypothesis that included the effect of varying the parameter for funding ratio in Model 2 by State:

Model 3: P(Pay) = 1/(1+exp(-η)),

η = 1.331047 + (-0.266503 – 0.993250 $I_{NP} + 0.923363 I_{RP}$)* log(1+Funded_Ratio). NLL = 556,227

Coefficients	Estimate	Standard Error	P-Value
(Intercept)	1.331047	0.005407	< 2e-16

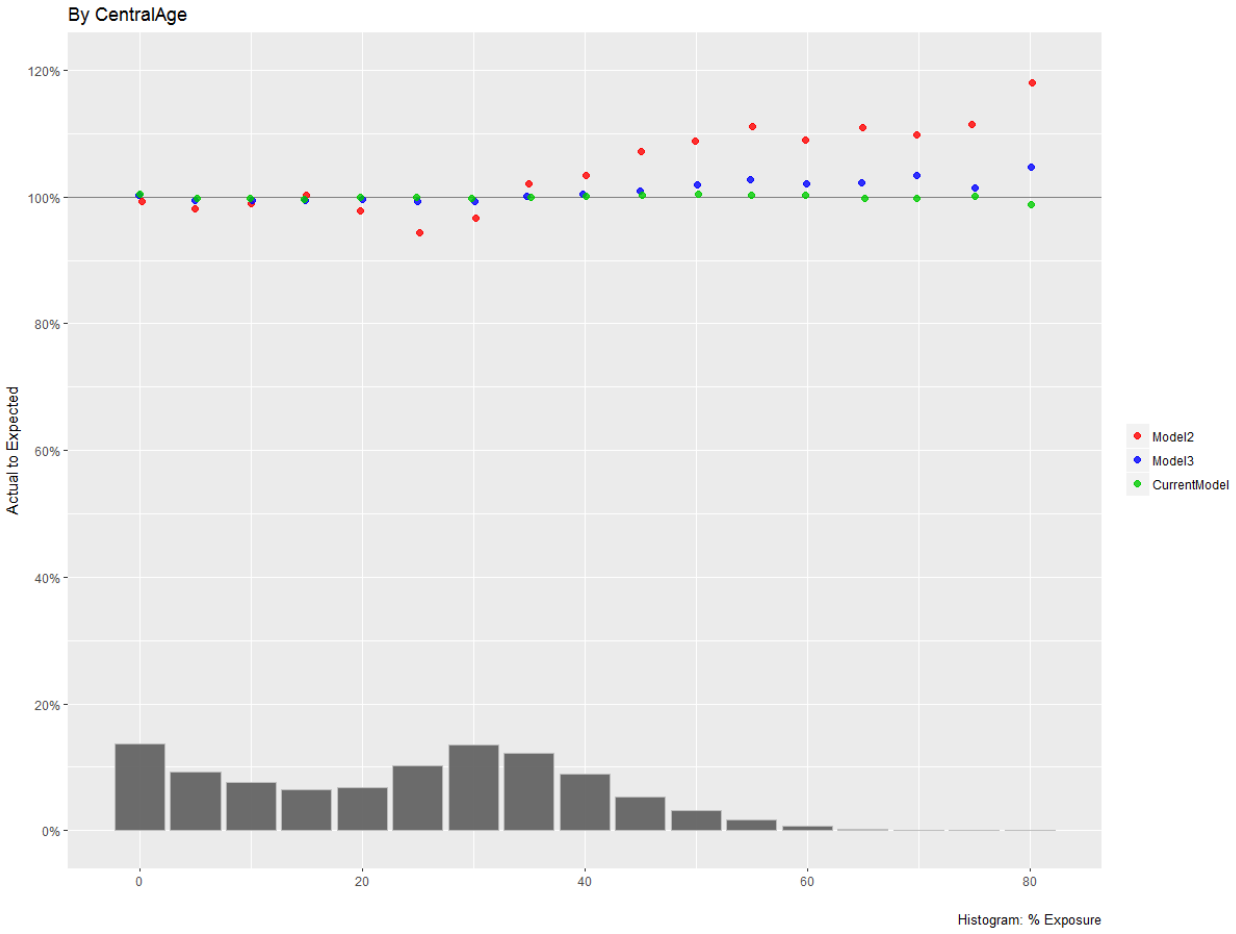
log(1+Funded_Ratio)	-0.266503	0.001765	< 2e-16
StateNP:log(1 + Funded_Ratio)	-0.993250	0.002858	< 2e-16
StateRP:log(1 + Funded_Ratio)	0.923363	0.001721	< 2e-16

There is a negative correlation between the funding ratio and probability of paying for policyholders that are not in State RP. The likelihood ratio improved by about 1 million. (This is an unusually large improvement.)

The modelers continued testing additional hypotheses, including that state has other effects that are not related to the funding ratio.

In addition to log-likelihood as a performance metric, the modelers use data visualizations, which can sometimes indicate where the model may be improved. For example, Figure 4 plots an actual versus expected ratio on the y-axis against grouped issue ages on the x-axis. The actual versus expected ratio is calculated by dividing the sum of the binary paid variable by the sum of the probability of paid for each group of issue ages. The three colors of dots each represent different models. Red dots correspond to Model 2 (discussed previously), whereas blue dots correspond to Model 3. The green dots (indicated as "CurrentModel" in the legend) represent the final model the team selected (to avoid confusion, it is not the existing tabular production model).

Figure 4: Actual vs. Expected Ratio



The histogram at the bottom of Figure 4 plots the distribution of issue ages. We can see from the plot that there is a positive trend to the actual versus expected ratios from Model 2 and Model 3, along with a corresponding lack of fit due to the lack of data on the right-hand side. This is indicated by the upward movement of dots compared to the 100% actual versus expected ratio and the lower height of bars beyond age 40. Although the incorporation of State from Model 2 to Model 3 did much to remove the effect of issue age (as seen by the blue dots being much closer to the 100% actual versus expected ratio than the red dots), there is still a positive correlation between actual versus expected and issue age for Model 3. This visualization led the team to hypothesize that issue age would be a good predictor for the probability of payment. Incorporating the issue age led to a significant improvement in the log-likelihood, so the team decided to use the variable in the current model.

Using this procedure, the team iteratively added more variables until it achieved a satisfactory model. For reference, the GLM representing current model has NLL = 455,080 on this batch, a substantial improvement (including 13 additional parameters).

4.6.1 COMMENTARY

With respect to feature engineering, the modelers relied on expert judgment from experienced stakeholders to create variables for the model. This is an appropriate practice, because the variables that end up in the model ought to be explainable and understood by business stakeholders, as well as consumers affected by the model, or even regulators.

With respect to feature selection, the modelers did not face the dimensionality challenges that may be found in a wider data set—that is, a set containing more variables than a human (or small group of humans) can feasibly examine within a reasonable timeframe. The modelers iteratively applied judgment to determine whether each variable ought to be included in the model, based on the magnitude of improvement as captured by the NLL. Without endorsing or critiquing this practice, we note that it leaves open the question of how much improvement is enough to justify an additional feature and that this metric in particular does not penalize for additional parameters, as AIC or BIC does.

As shown in the example of first adding the funded ratio as a feature, the firm tested various functional forms for each predictor along the way. A possible addition to this step could be binning the funded ratio, or introducing a spline, to investigate whether further nonlinearities in the relationship exist. The case study does not detail how the firm's concerns of collinearity were addressed, but we note that coefficients of generalized linear models are sensitive to collinearity, and it is wise to monitor the correlation between features.

4.7 MODEL EVALUATION AND MEASURES OF SUCCESS

In addition to using log likelihood as a metric for feature selection, the modelers tested the predictive model's efficacy by comparing it to the firm's current production model, a classical, tabular-based model. Two criteria were selected to compare the models:

1. Actual versus expected analysis—tested using the actual versus expected ratio

   This ratio equals the sum of next month's actual premium for each observation, divided by the sum of next month's predicted premium. Thus, a ratio greater than 1 indicates underprediction (actual premium is greater than expected), and a ratio less than 1 indicates overprediction (actual premium is less than expected). Ideally, the ratio should be as close to unity as possible. This metric may be calculated at different levels of granularity and on a variable-by-variable basis.

2. Precision-tested using the mean absolute percentage error (MAPE)

   The MAPE equals the mean of the absolute value of the ratio between the difference of each observation's actual and predicted premium to the actual premium.

4.7.1 ACTUAL VERSUS EXPECTED ANALYSIS

The following actual versus expected ratios were calculated by restricting the test data set to policies with at least one year of experience that were incepted during calendar year 2007 and beyond. This was done because the production model is explicitly intended only for these policies.

For the test data, the predictive model resulted in a slight improvement to the actual versus expected ratio compared to the production model when calculated on the holdout data set (see Figure 5).

Figure 5: A/E Ratios for Predictive and Production Models

Model	Actual vs. Expected Ratio
Predictive	1.01
Production	1.02

On an aggregate, expected-value basis, the current production model is quite accurate, because the data set was large enough to produce accurate predictions.

The modeling team further evaluated the actual versus expected ratio on a variable-by-variable basis, with the variables chosen by expert judgment via interaction with actuaries and underwriters. This is to ensure that pricing is done accurately not only at an aggregate level, but also by important subdivisions of the business that leadership cares about. Doing so will prevent issues such as adverse selection and will ensure that organizational resources (such as marketing) are used efficiently.

Figure 6 plots the actual versus expected ratio on the y-axis against policy duration on the x-axis. The horizontal line at 100% actual versus expected represents the set of perfect average predictions. The red dots represent predictions made from the predictive (development) model, whereas the blue dots represent predictions made from the classical (production) model. Ideally, the dots should be as close to this line as possible. The histogram at the bottom of Figure 6 indicates the volume of data expressed as a percentage of the book for each level of duration.

Figure 6: A/E Ratio by Policy Duration

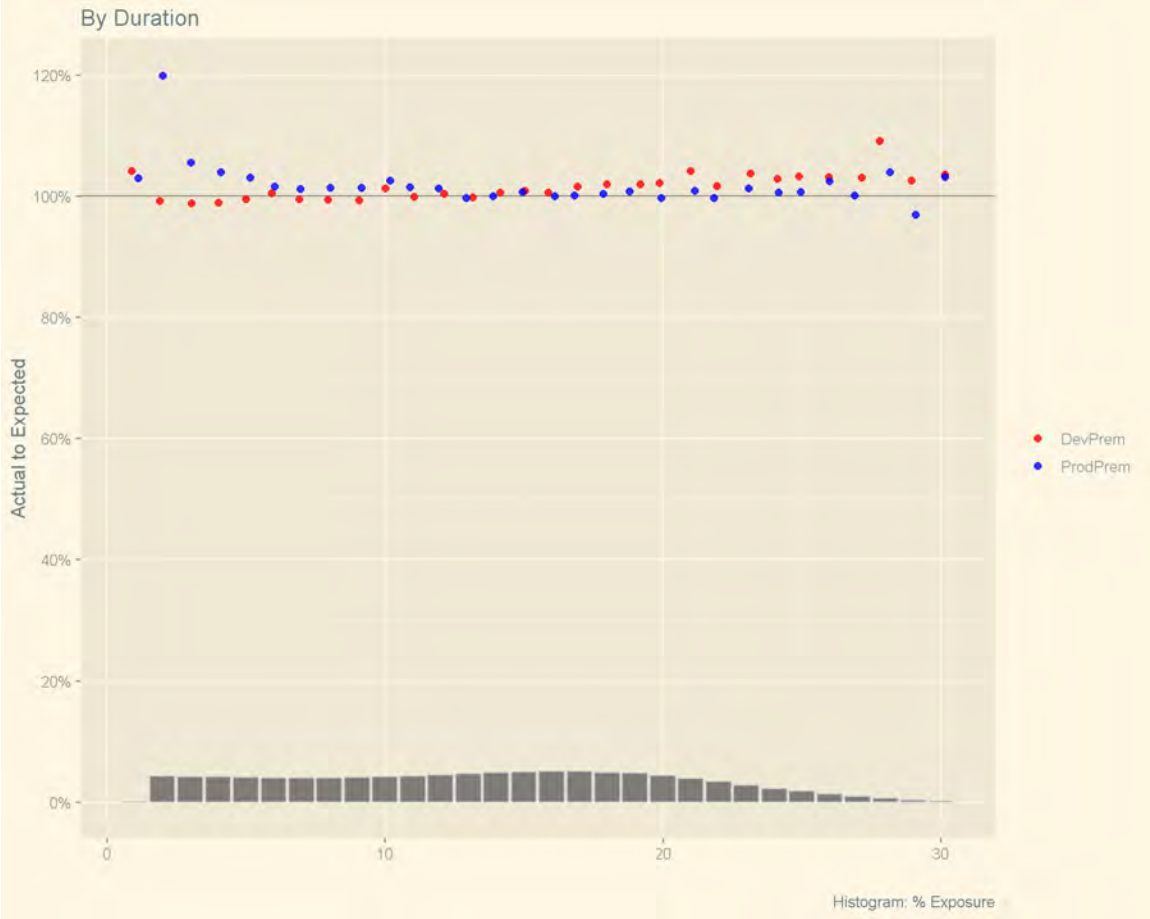
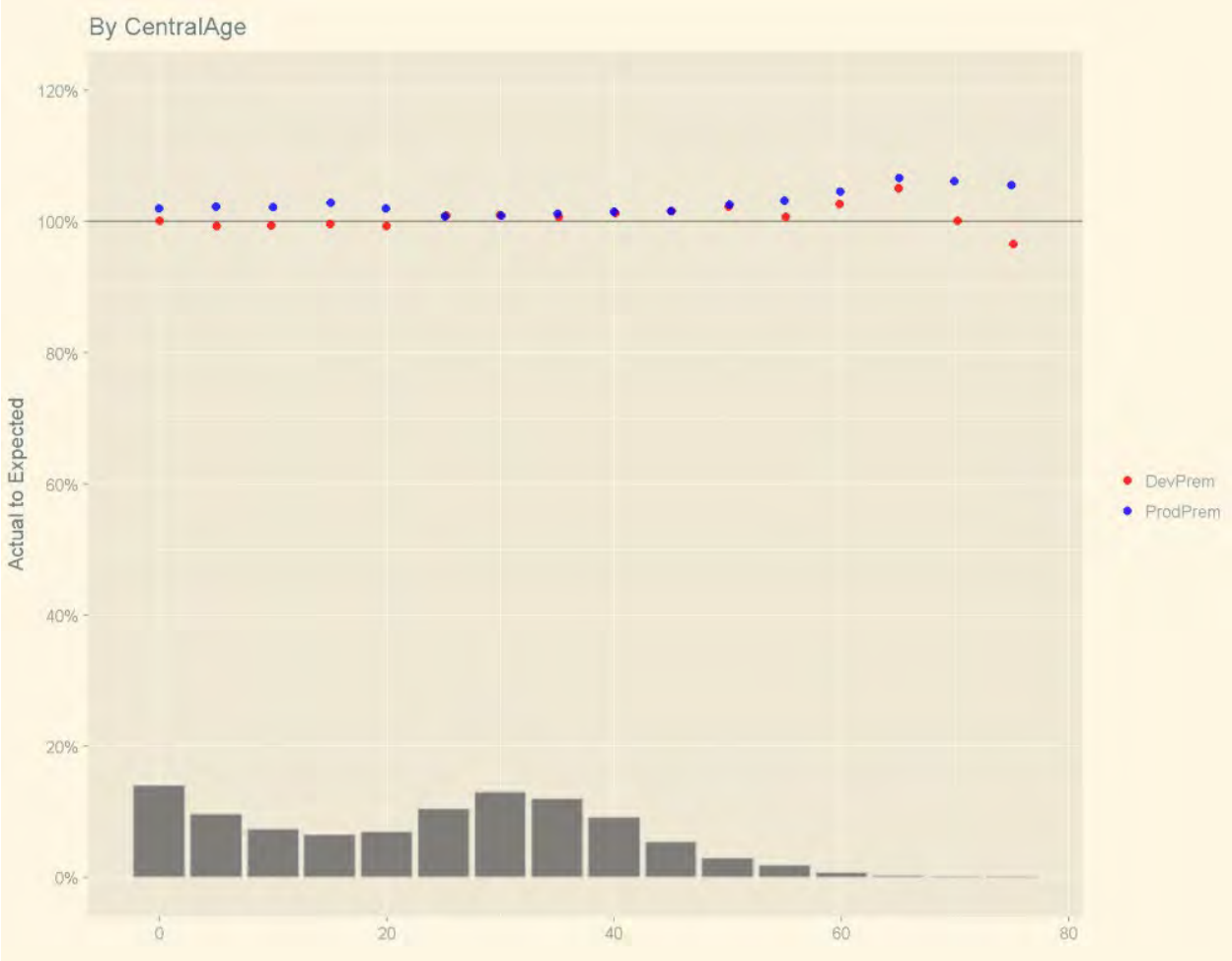


Figure 6 indicates that both models make their strongest predictions at the levels of duration with the most amount of data. Both models make their weakest predictions at the lowest and highest extremes of duration, indicated by the large distance of the blue and red dots from the 100% horizontal line. In this case, the discrepancy between actual and expected at the extremes may stem from insufficient data to calibrate actual rates. The development model generally made better predictions than the production model through the first 12 durations and generally worse predictions beyond the 17th duration.

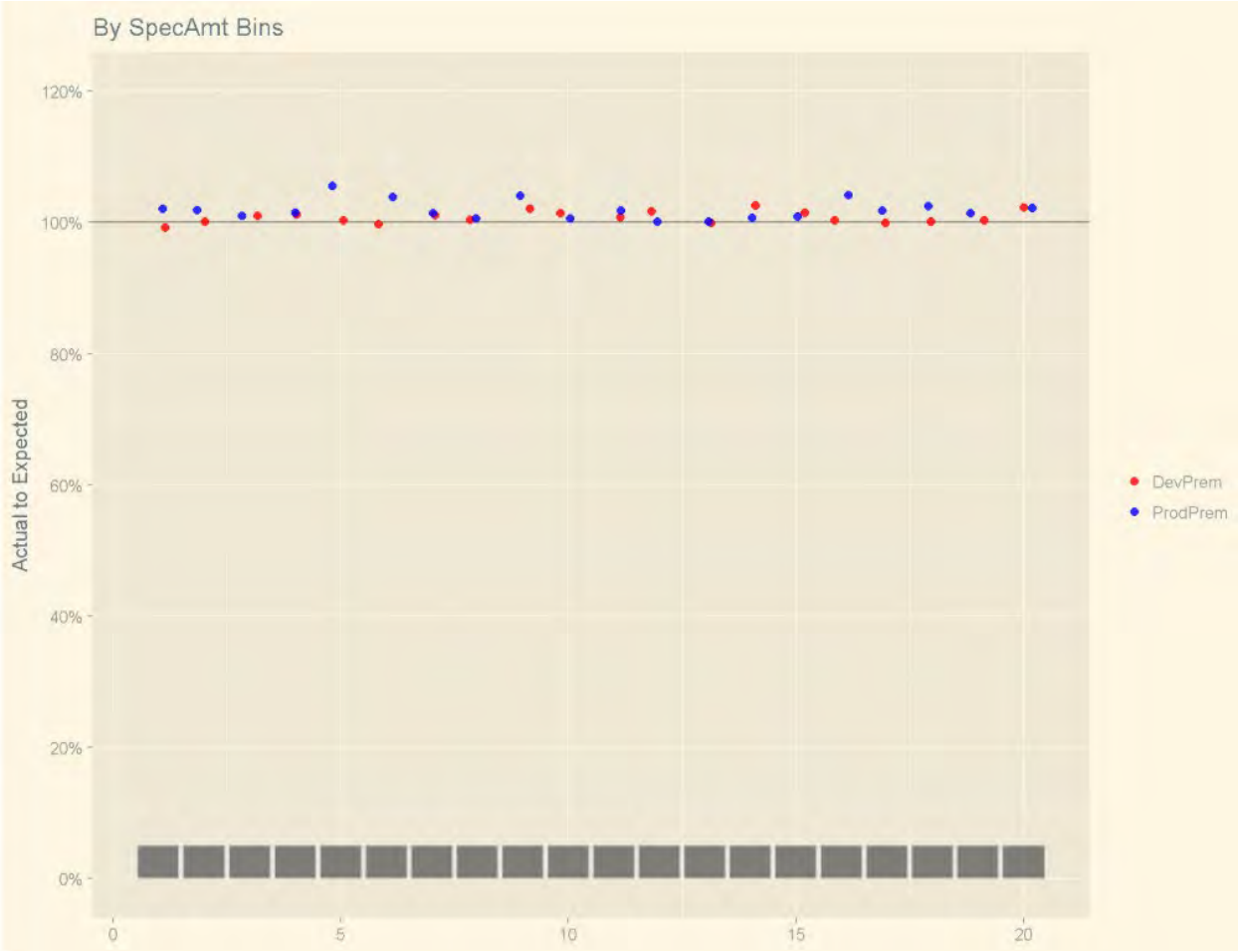
Figures 7 to 9 follow a similar format but plot the actual versus expected ratio against other variables: Central Age, SpecAmt (binned) and Cash Amount (binned).

Figure 7: A/E Ratio by Central Age



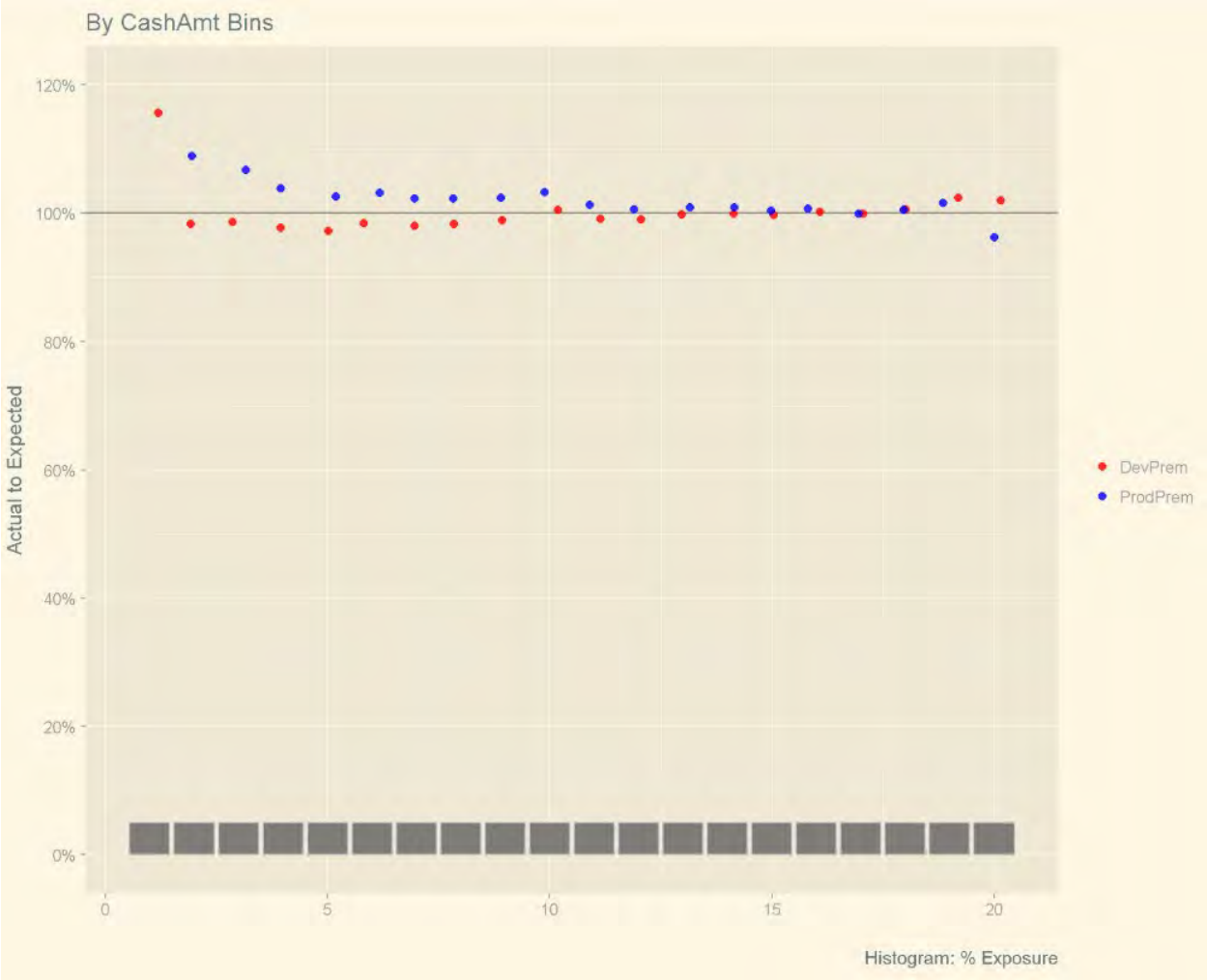
By Central Age, the development model appears to make better predictions than the production model through all ages.

Figure 8: A/E Ratio by SpecAmt Bins



By SpecAmt bins, the development model generally makes better predictions than the production model, except for bins 10–15.

Figure 9: A/E Ratio by Cash Amount Bins



By account value (CashAmt), neither model is very predictive at the low levels, but the development model is typically better than the production model at most levels.

Reliance on a single metric, however, can be deceiving. If we were to solely rely on the actual versus expected ratio to compare models, we may well conclude that there is very little to be gained with the introduction of GLMs. The next section demonstrates that there are insights to be gained when we introduce precision as a metric.

4.7.2 PRECISON

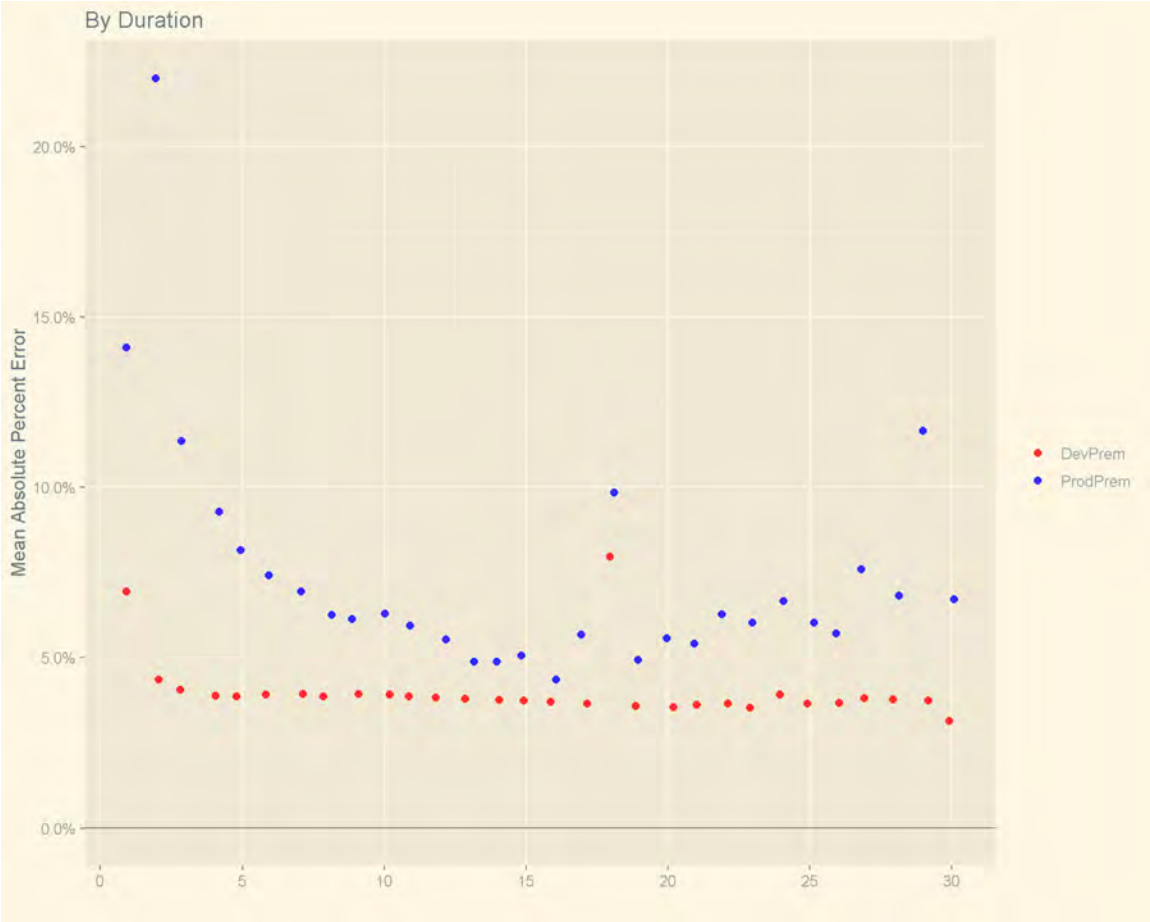
Although the predictive model only yields slight improvements when considering the actual versus expected ratio, it demonstrates significant added value when considering precision.

For this model, precision is defined as the mean absolute percentage error (observations with zero actual premium are removed). Like the actual versus expected ratio, this metric can be calculated at different levels of granularity, as well as from the perspective of different variables.

On the test data, the predictive model yielded a mean absolute percentage error that was half of that of the production model. Roughly 15% of the observations were removed because the actual premium was $0 for those observations. To account for them, modelers examined the mean premium amounts predicted by each model for the observations with $0 actual premium, which were $2.45 for the predictive model and $2.51 for the production model. The predictive model produced an amount that was closer to $0 than the production model, and this was deemed to be reasonably close to $0.

Figure 10 plots the mean absolute percentage error against the duration. The blue dots represent predictions made by the production model, whereas the red dots represent predictions made by the predictive model. The horizontal line represents a 0% MAPE and ideally the points should be as close to this line as possible.

Figure 10: MAPE by Policy Duration



In this case, the predictive model represents a clear improvement over the production model with respect to precision, because the red dots are much closer than the blue dots to the horizontal 0% line.

Figures 11 to 13 follow a similar format but plot the MAPE against other variables: Central Age, SpecAmt (binned) and Cash Amount (binned).

Figure 11: MAPE by Central Age

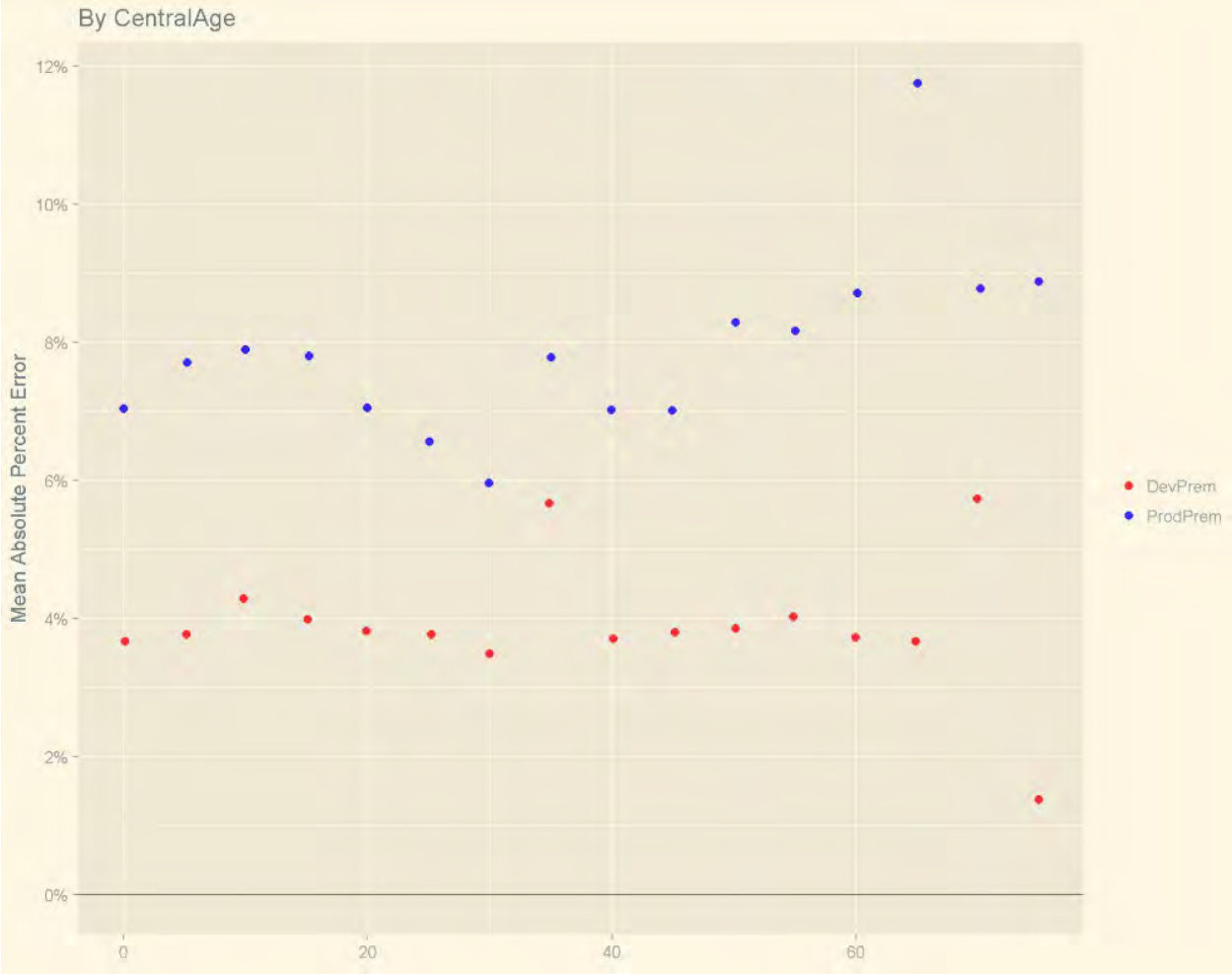


Figure 12: MAPE by SpecAmt Bins

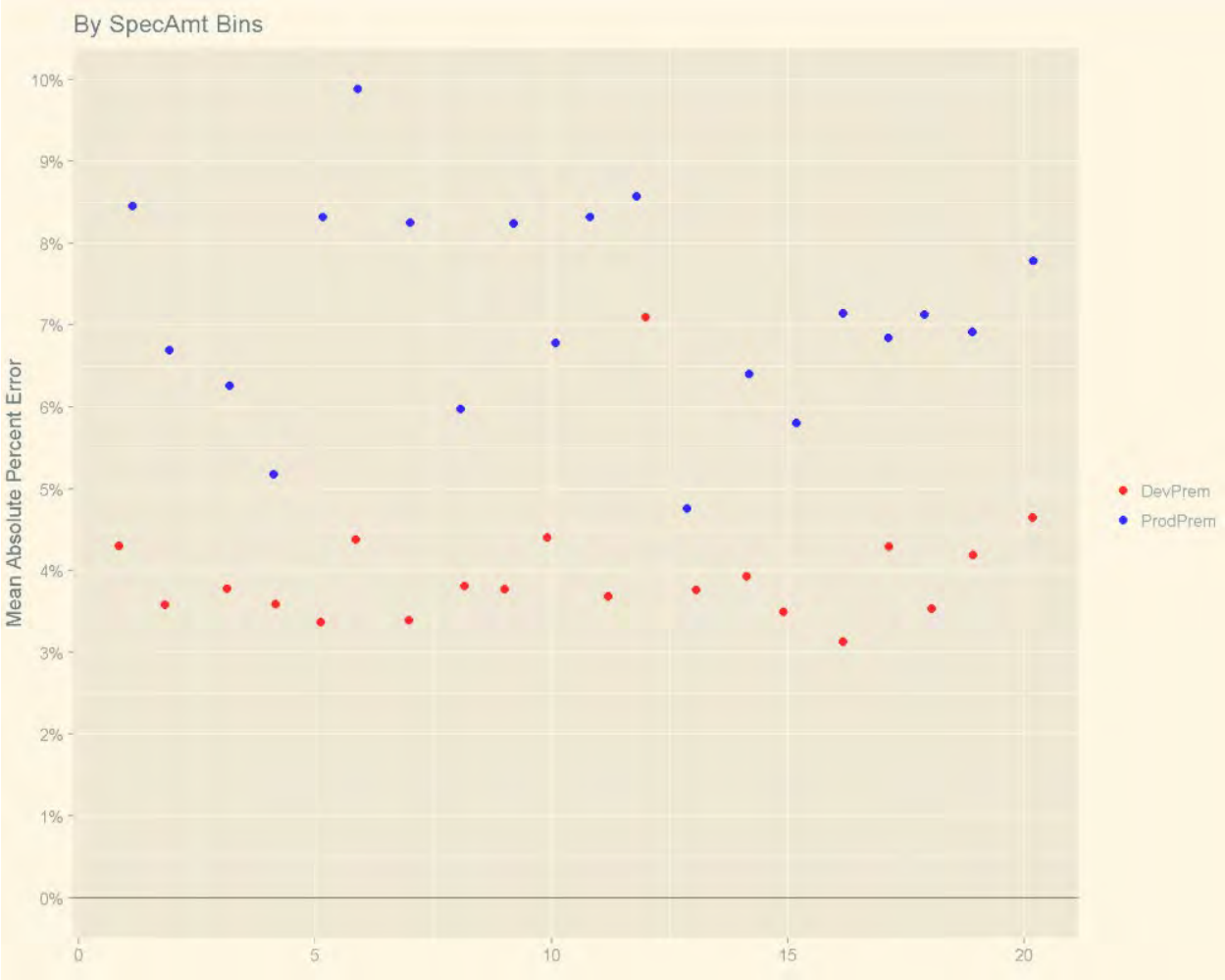
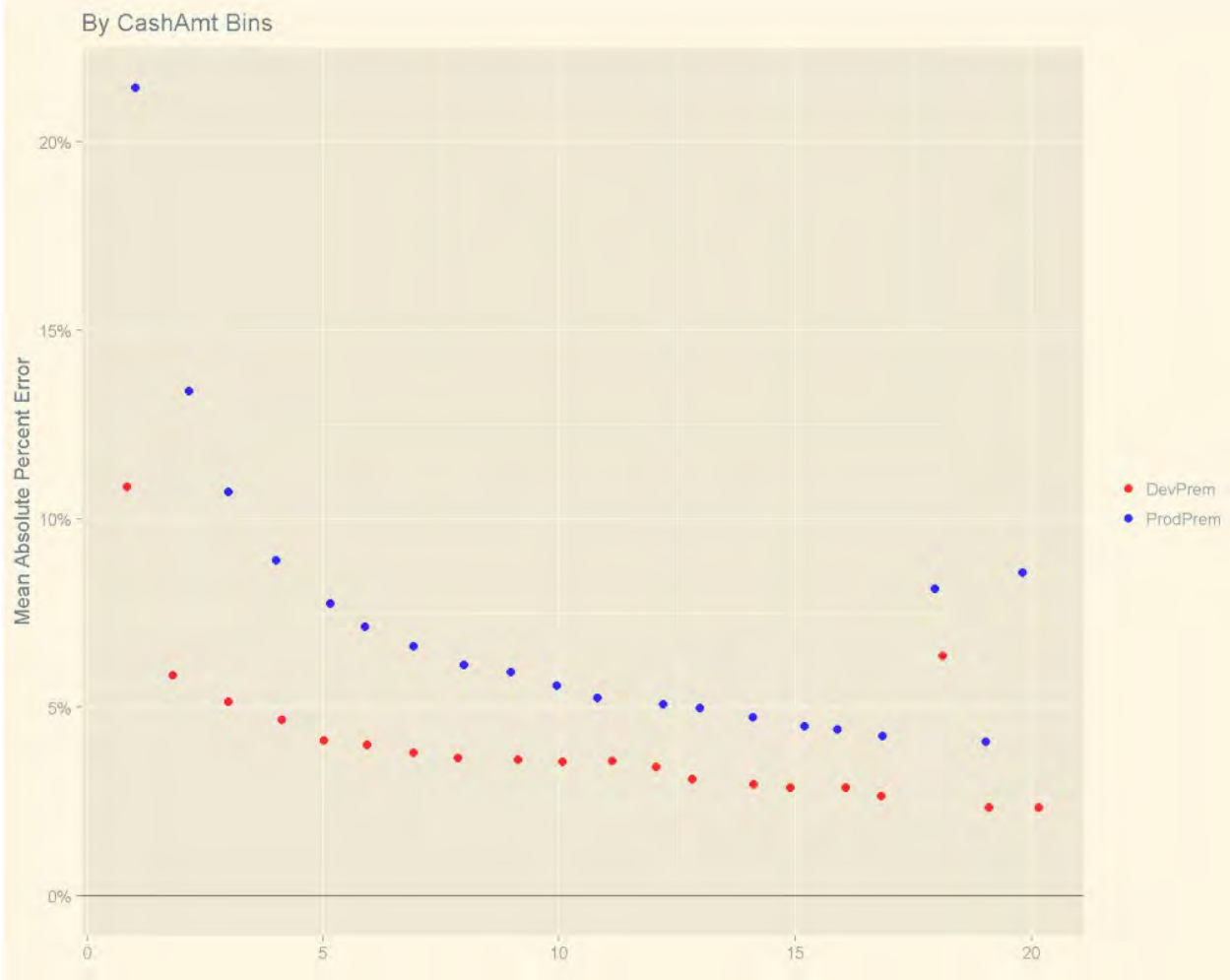


Figure 13: MAPE by Cash Amount Bins



Using precision in addition to the actual versus expected ratio as a performance metric emphasizes the difference between traditional and predictive modeling. Developing a more precise model may lead to insights on policyholder behavior that inform company leadership. Even if the model does not get implemented in production, it may still be used to assess the efficacy of the existing production model (such as testing the efficacy of cell-level groupings). The firm has indicated that other departments, such as pricing and in-force management, may be interested in policy-level models, for such purposes as customer retention through targeted customer communication.

4.7.3 COMMENTARY

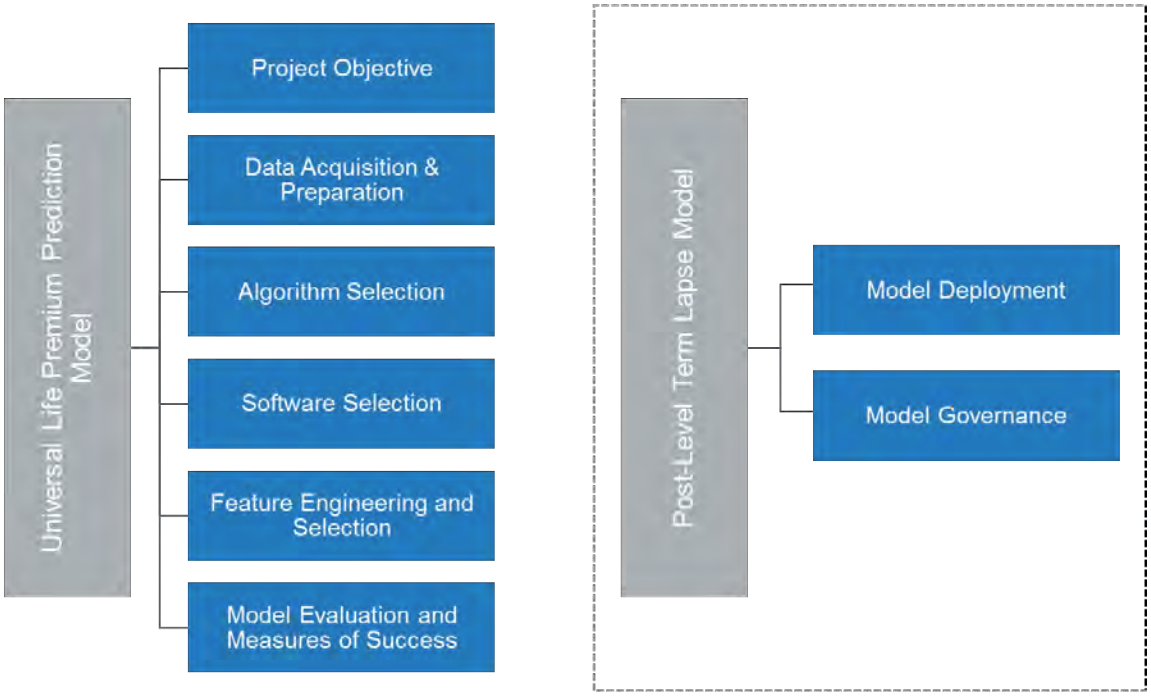
The firm made the appropriate choice by choosing the current production model—the classical, tabular-based model implemented in projection platform as a basis for comparison. This was a straightforward decision, because the project's goal was to improve upon the predictions that the firm was already making with respect to policyholder premiums. Had the production model not already existed, the team could have built a "challenger" model to ensure the chosen model performed better than alternatives. Examples of challenger models can be seen in the feature engineering phase, where candidate models were compared against the final development model on an actual versus expected and log-likelihood basis. A less ideal approach in this situation would have been to compare against the baseline, which is the average

premiums over all observations. In this scenario, the exhibits above would have compared the actual versus expected ratios and precision metrics between the development model and baseline predictions. By performing this test on data withheld from the model fitting process, the firm has helped to prevent overfitting. By including the exposure histogram in the A/E ratio figures, it has helped reviewers to understand how much information is supporting model predictions, which may help address questions of credibility.

4.8 MODEL DEPLOYMENT

For the following sections, we will now change the focus from the UL premium prediction model to the term lapse model, indicated by the dotted line box in Figure 14.

Figure 14: Structure of Case Studies



The firm's modeling group Implemented the term lapse model in the projection platform. Conceptually, this group was required to translate a predictive model that made predictions at a policy level to a production model that made predictions at a cell level. In the production environment, all policies within a cell are treated identically. The modeling team was able to accomplish this without a loss of precision because the production cells are completely homogeneous with respect to this particular model.

This consideration will, however, be an issue when it comes time to implement the UL premium model, since it makes predictions on a policy-level and not cell-level basis. Current plans to address this include defining each cell in production to be equivalent to a policy and comparing the projection results. The firm is willing to sacrifice a small amount of precision in exchange for improved model run time. While the firm expects aggregate accuracy to be preserved, it plans to conduct sensitivity tests via cell-clustering adjustments.

The firm's modeling group is led by a dedicated model steward who oversees the implementation process. Their team consists of a group of actuaries who have considerable experience in programming in a variety of languages, including Perl, SAS, R and Python. By hiring actuaries who are also capable of programming, the firm has avoided the communications issues that often arise when the actuaries who create the models must interface with software engineers who may not have deep insurance or statistical knowledge. This is an operational risk that often plagues insurance companies' analytical initiatives.

While having actuaries be responsible for coding up the production system softens the communication barrier between modelers and implementers, cross-team communication is still required because those who develop the models will have more intimate knowledge of their product, and conversely, those who are responsible for implementing the models are likely to have more knowledge of the production environment.

The model specifications were delivered to the modeling group via an R Markdown document. This document contains a comparison between the prior and new models and highlights the changes between them. It defines all the relevant covariates and includes the most recent estimates of the parameters.

4.8.1 COMMENTARY

Because the implementation of the firm's predictive models remain largely within the confines of its actuarial function, the firm has avoided many of the operational challenges that arise when models need to be implemented in areas outside of the actuarial department (such as models implemented within the company underwriting system, or implemented in the marketing department). As the company expands the reach of its predictive analytics to other areas of the firm, it will have to consider cross-department communication and technical challenges, but that does not appear to be a current concern.

4.9 MODEL GOVERNANCE

4.9.1 ORGANIZATIONAL STRUCTURE

The firm's model governance framework consists of three committees:

1. Model Assumptions Committee (MAC)
2. Executive Finance Committee (EFC)
3. Model Oversight Committee (MOC)

The MAC is made up of a subset of the actuarial leadership. It includes the chief actuary, the appointed actuary, the managing actuaries responsible for financial reporting, and the model steward. These people are responsible for providing technical oversight of the assumption-setting process on a yearly basis. Proposed changes are submitted to the committee, whose voting members are responsible for approving the proposals. Examples of such proposals include whether to adopt a new logistic function to project policy loan utilization rates or whether to adopt new lapse rates.

Upon the determination of yearly assumptions, the MAC submits a memorandum to the EFC detailing the rationale for the assumptions as well as the financial impact of making those assumptions. An example of a financial impact would be how the present value of future profits (PVFP) would change with the introduction of new assumptions. The PVFP is calculated using the new and old assumptions, with the difference quantified.

The EFC consists of senior company leadership, including the chief financial officer, controller, appointed actuary and chief actuary. Their responsibilities are to ensure that the assumptions and their likely financial impacts are understood. The rationales behind the assumptions that the MAC submitted are discussed and finalized.

The MOC oversees the implementation of the assumptions in production reporting processes, particularly with regard to projection platform. The MOC consists of experienced, management-level actuaries from various areas of the firm. These actuaries have voting authority and are organized by a nonvoting member who is also an actuary. The MOC ensures that the assumptions are implemented as they are approved. Furthermore, the MOC ensures that the models are protected from unwarranted changes and that model integrity is enforced.

4.9.2 MODEL INVENTORY

The firm has a formalized model inventory that it maintains in a spreadsheet. This spreadsheet is called the *Model Assumption Update Inventory* and lists all assumptions that are the firm uses in the model. Each year, the firm uses the spreadsheet to plan which assumptions will be updated (most assumptions are updated annually, but there are a few that are updated every other year).

The MAC has to approve any action taken, including the decision not to update an assumption. It is worth noting that this spreadsheet does not specify complete details of each assumption. For example, if the MAC decided to replace a simple assumption (e.g., a rate that varies by product and policy year) with a predictive model, the spreadsheet would not change. There would be separate documentation that would describe the model.

The models contained within the model inventory include two predictive models—the term lapse model, and an anti-selective mortality model. In addition to predictive models, the firm maintains numerous classical experience models, which include persistency and mortality assumptions for all business in-force. Persistency includes policy retention, agent persistency, and, for flexible premium products, premium persistency. Mortality modeling is generally done in relation to appropriate industry tables, adjusted using limited fluctuation credibility to the firm's experience and reflecting anticipated mortality improvement.

Furthermore, the firm maintains data-driven assumptions regarding cash flow models, including policy loan utilization, inflation rates and expense attribution.

4.9.3 VERSION CONTROL PRACTICES

The actuarial department uses a combination version control systems (VCS) for various components of its modeling system. A diverse array of tools is necessary due to evolving technology and the particular combination of proprietary and open-source software that the firm uses. The firm must manage legacy systems that are not completely compatible with modern VCS practices while simultaneously adopting these approaches for the newer models that it seeks to develop.

- The actuarial department uses Subversion, a VCS sponsored by the Apache Software Foundation, for version-controlling Perl, SAS and Python scripts that impact production.
- Predictive modeling, consisting of R scripts, is controlled via Git.
- Models developed are version-controlled with a proprietary system that the firm developed.
- Excel-based models are not subject to a VCS. While it is technically possible to use a VCS such as Git to version control binary Excel files, it is not practical, because each new version of each Excel file is saved as a standalone binary (rather than just saving the differences), which leads to heavy

storage requirements. Such models, however, are saved in read-only files in protected network locations.

4.9.4 INDEPENDENT REVIEW OF MODELS

The firm's external auditor began reviewing the methodology development and implementation of principle-based reserves in 2016. The audit firm reviewed the mortality and lapse best estimate development and margin setting, asset assumption development, and important management decisions, including the treatment of post-level term expense allowances and profits. The audit firm conducted a full-process walk-through meeting, which reviewed the following:

1. Data sources (including key points).
2. Experience studies.
3. Assumption development and documentation.
4. Models of both assets and liabilities.
5. Results of the analysis, including aggregation or consolidation and any out-of-model adjustments.
6. Recording to the general ledger.
7. Reporting and sign-offs, including governance and monitoring.
8. Any handoffs in the process.
9. Key controls covering the process.
10. IT considerations (e.g., systems, data reports, etc.).

There were no unresolved issues identified during the review process. In addition, the firm engaged a consulting firm to review the 2017 version of the post-level term lapse model.

4.9.5 DOCUMENTATION PRACTICES

Documentation for predictive models has been produced in R Markdown with an effort to conform to standards of reproducible research. The full corporate assumptions are documented in an Excel spreadsheet that is intended to be human-readable. Translation of those assumptions into files fit for the modeling platforms use has been a manual process that the firm's systems group is currently automating.

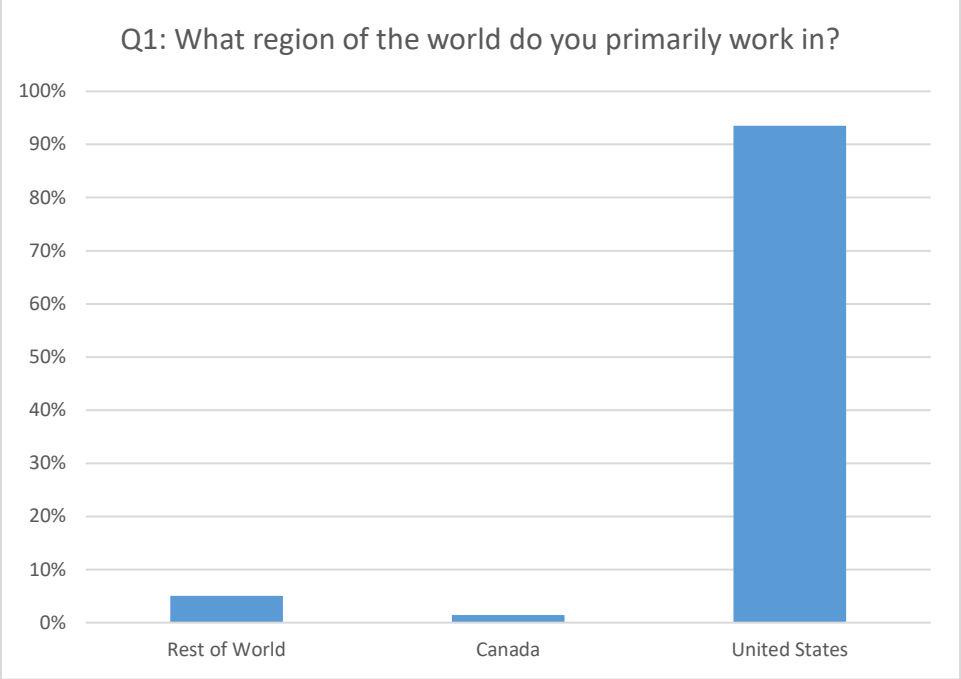
4.9.6 COMMENTARY

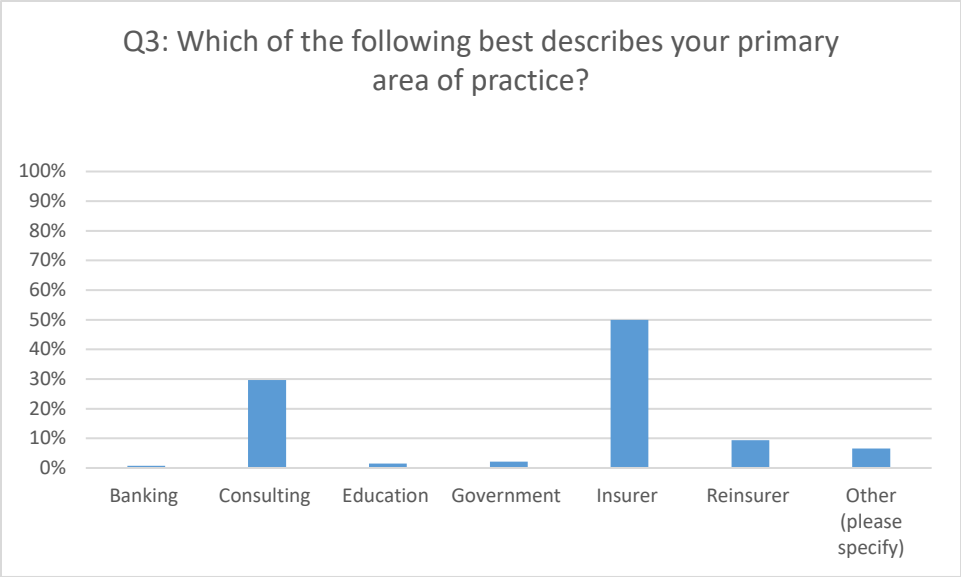
The firm's status in this area is mature, having established a governance framework that includes professionals from a diverse array of functions, as well as executive management, and established Git as its version control system. We consider these to be good practices, regardless of the firm's size. Executives' involvement helps prevent disputes that may scuttle predictive analytics efforts even after the expenditure of a large amount of time and resources.

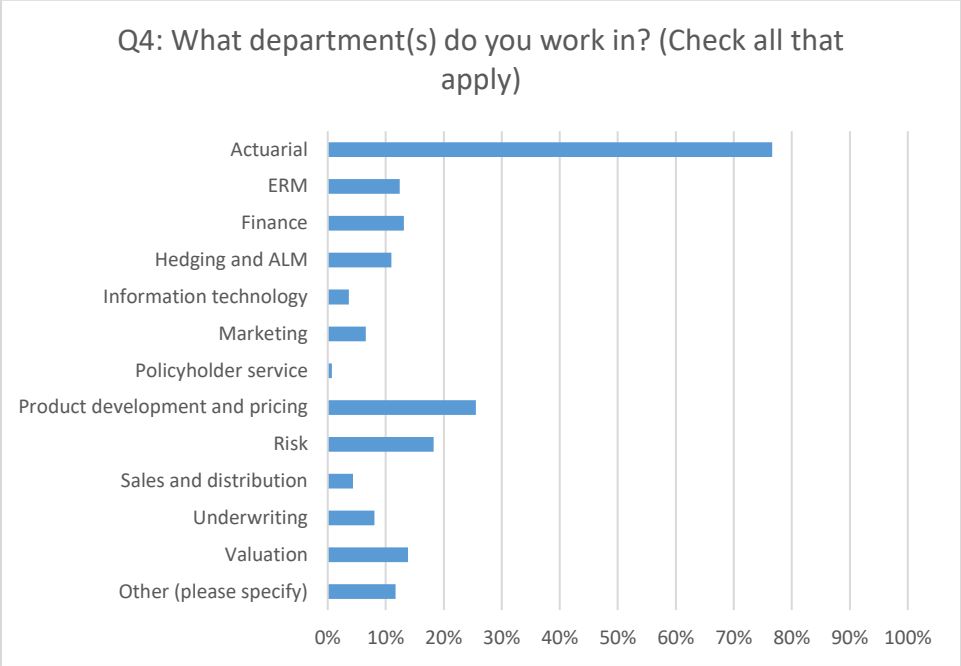
Appendix 1: Survey Results

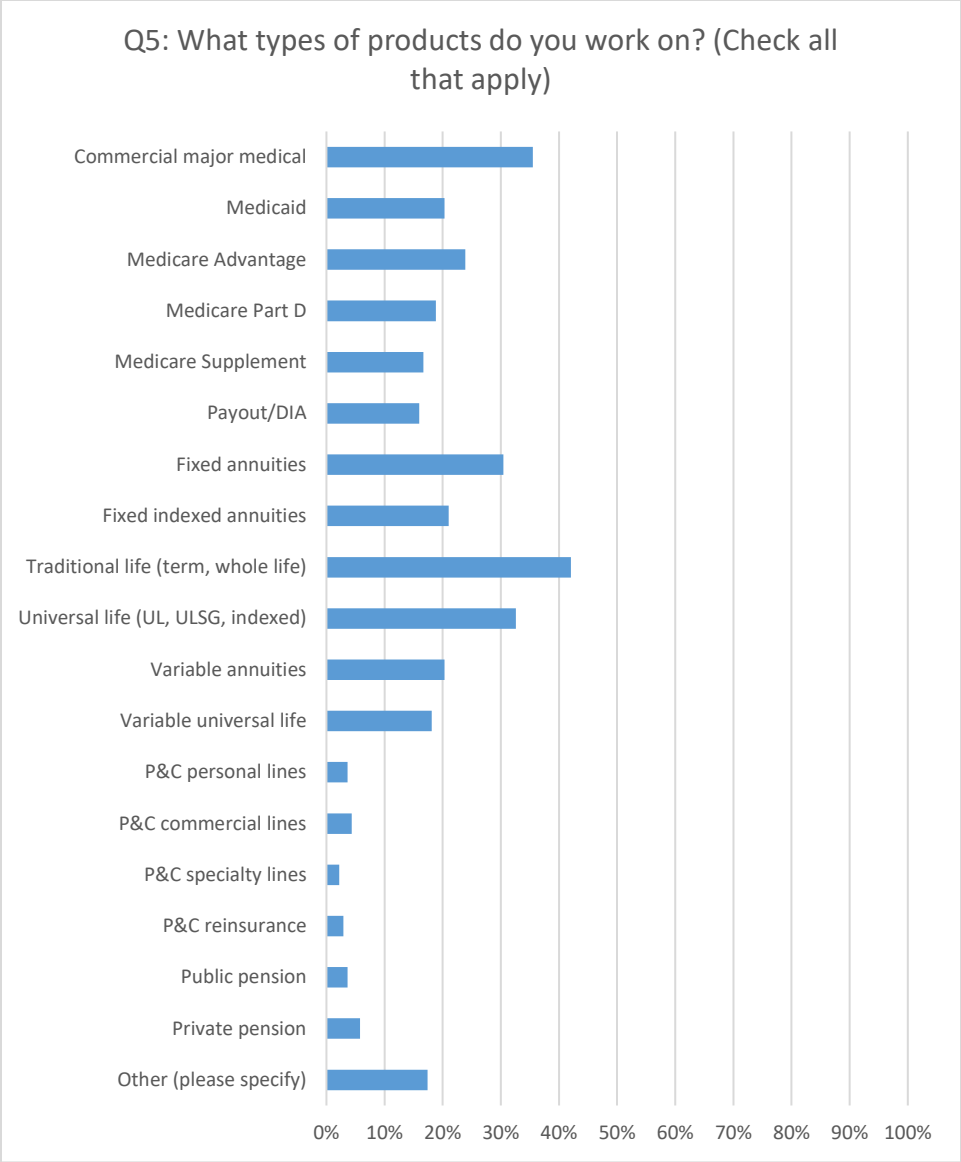
As part of this research effort, we conducted a survey of actuaries from the SOA regarding their current practices in predictive modeling. The survey included 46 questions and was administered in fall 2018 via SurveyMonkey. The respondents were primarily members of the Predictive Analytics and Futurism and the Modeling sections. We received a total of 143 responses. This appendix presents summarized results for each question, organized by topic.

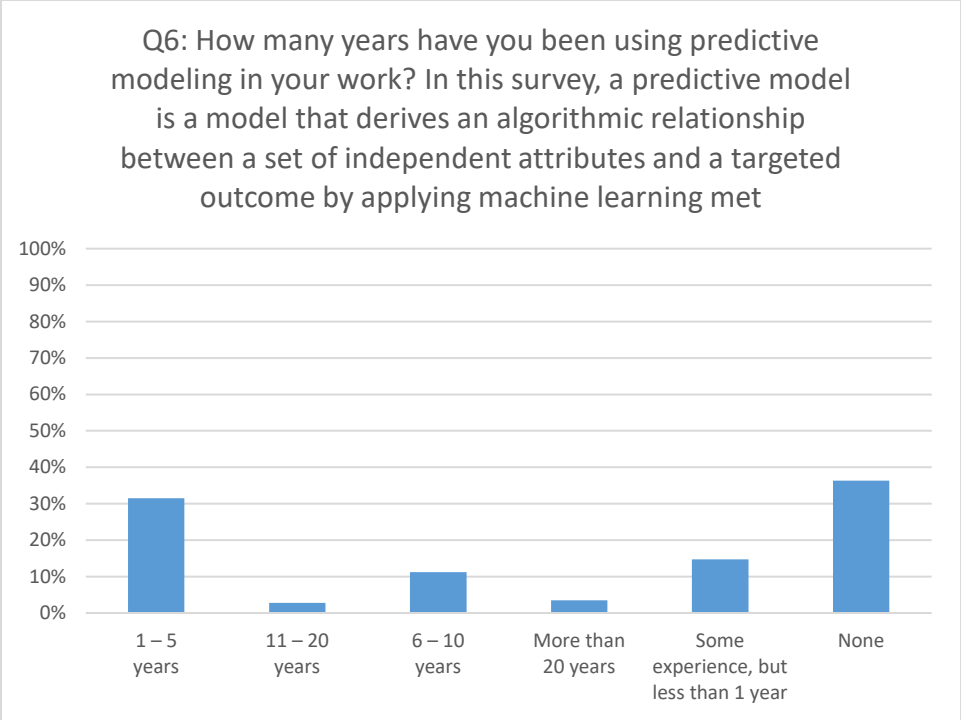
DEMOGRAPHICS



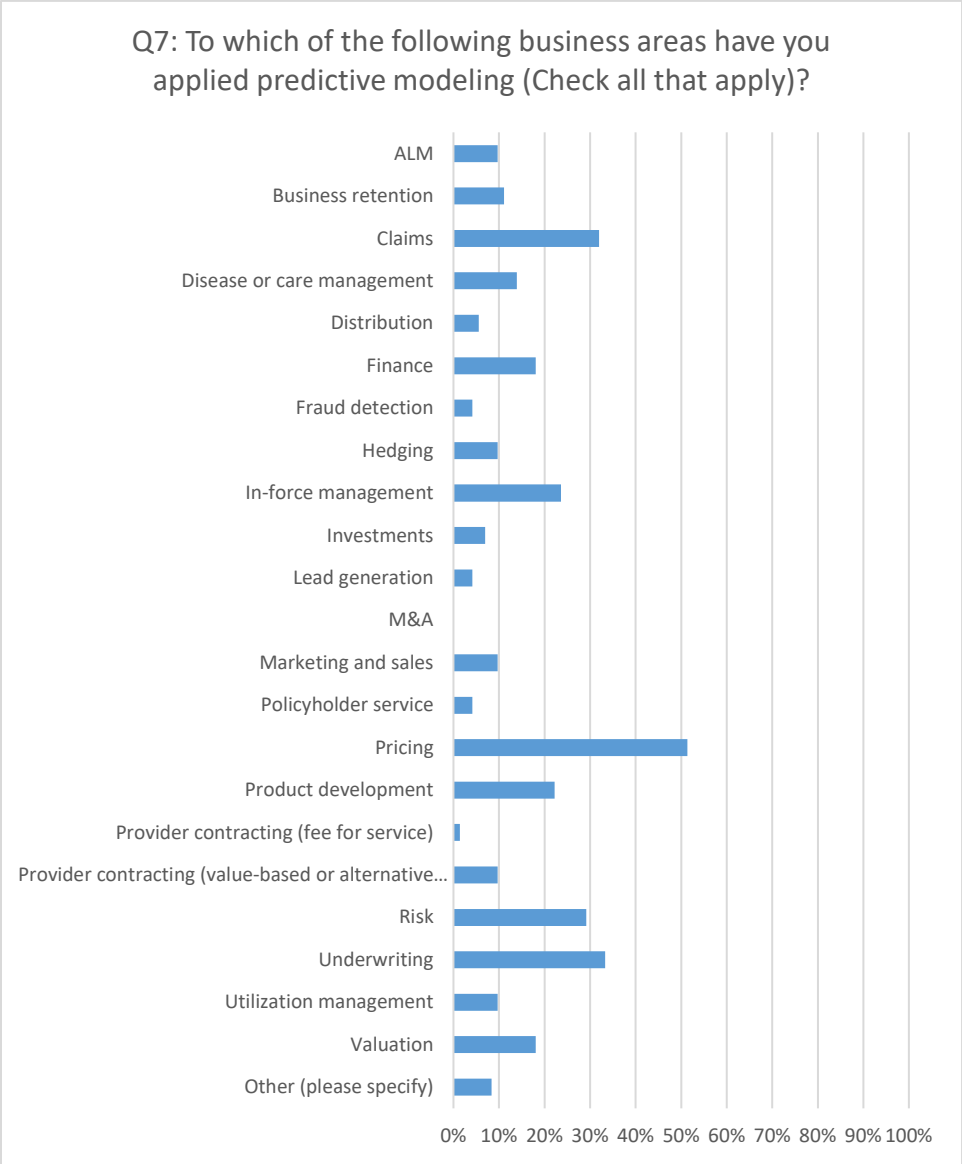


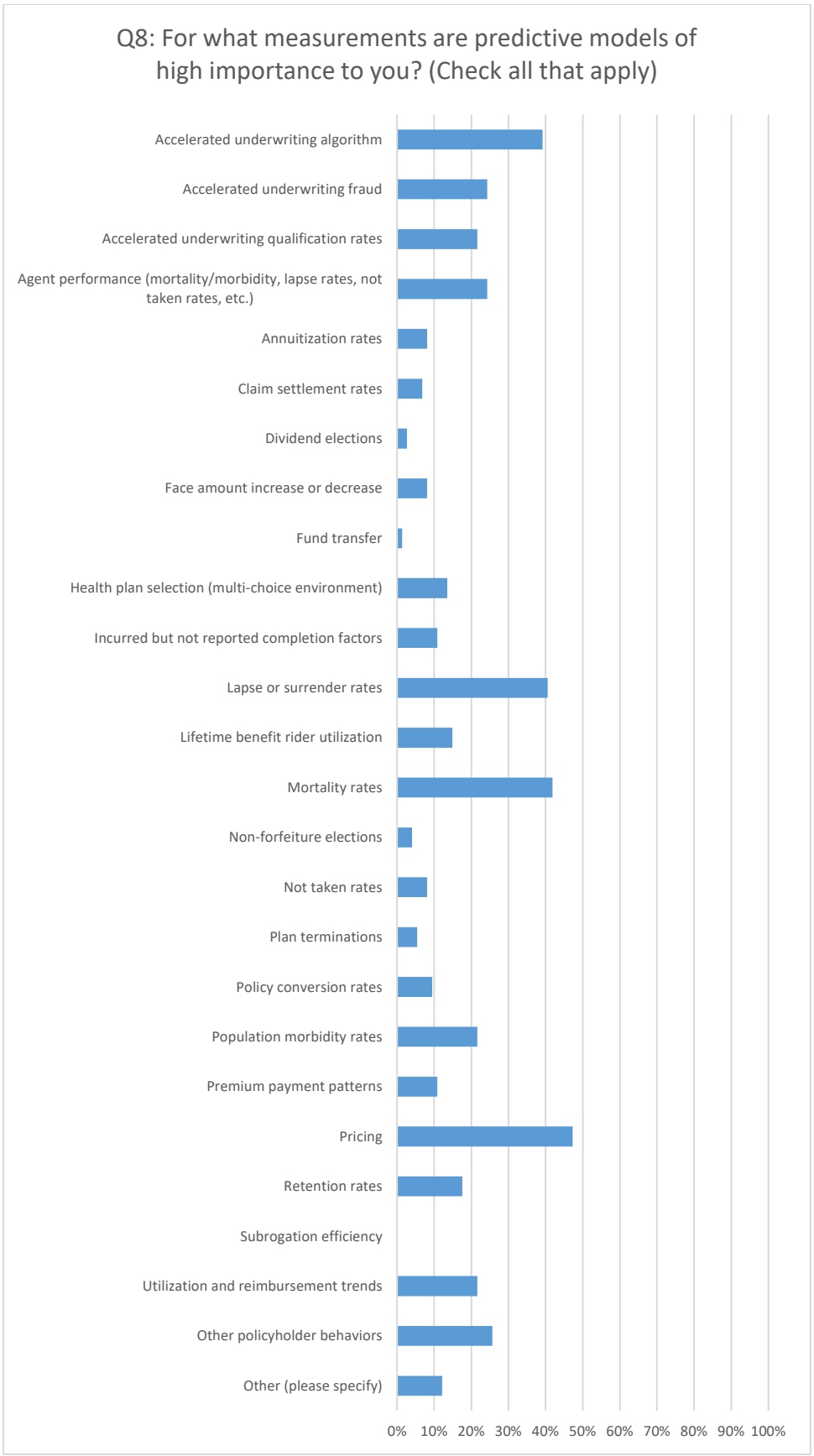


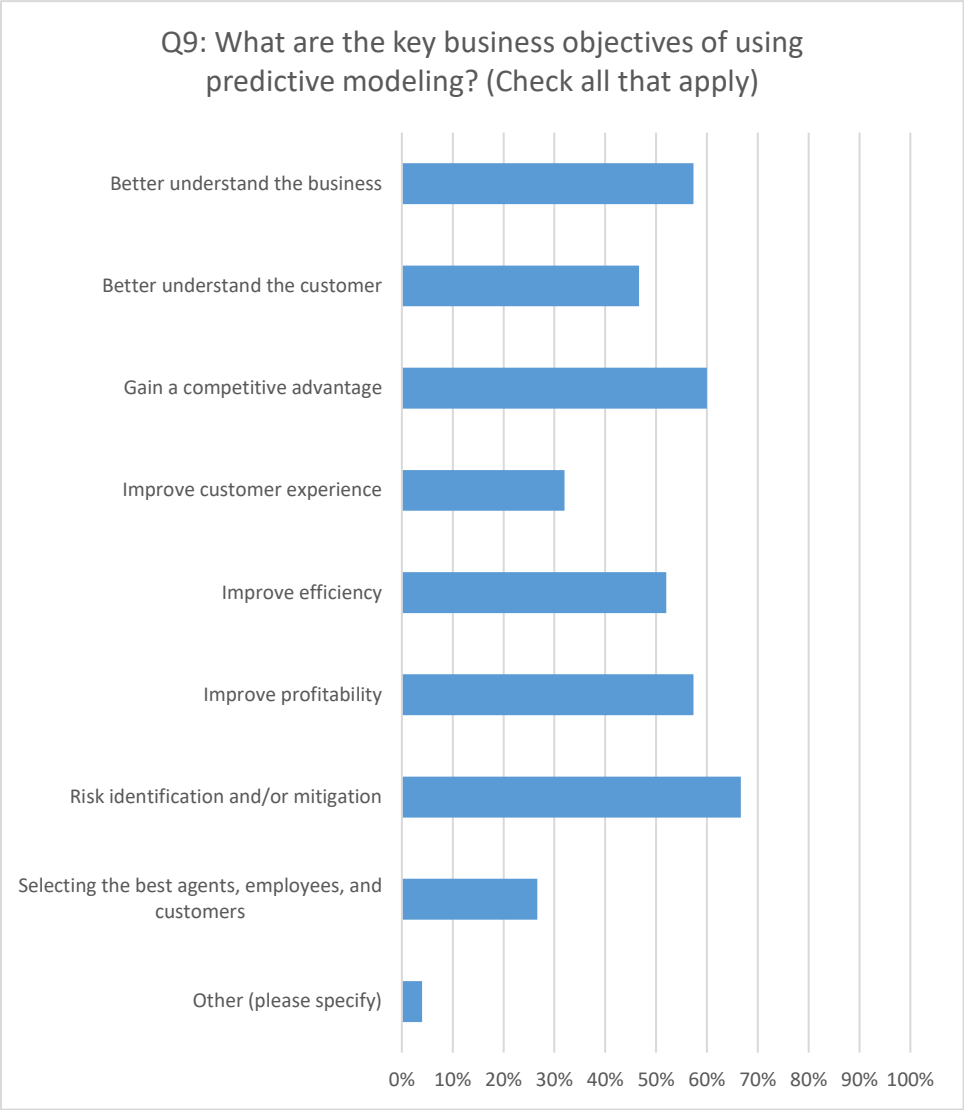




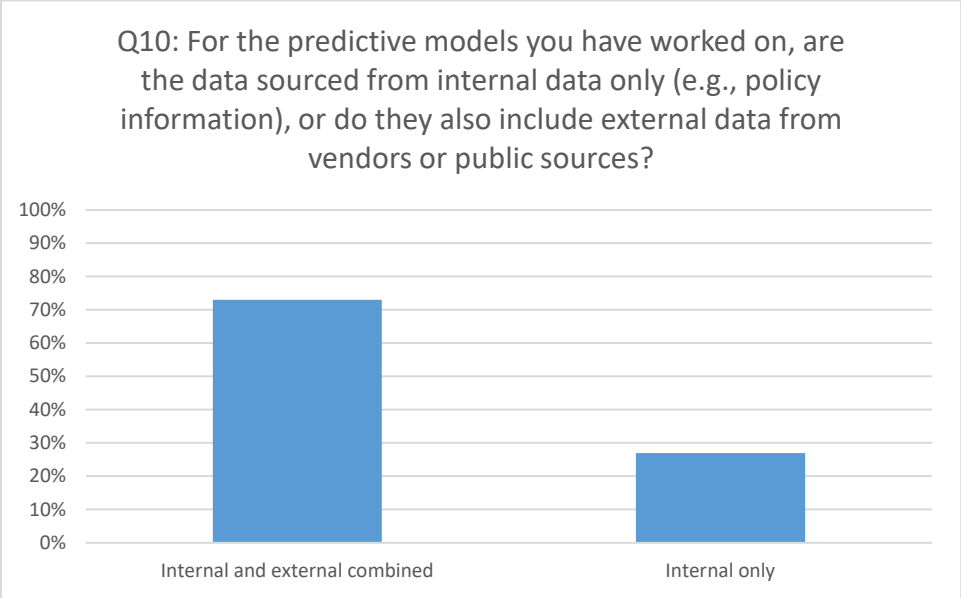
BUSINESS PURPOSE

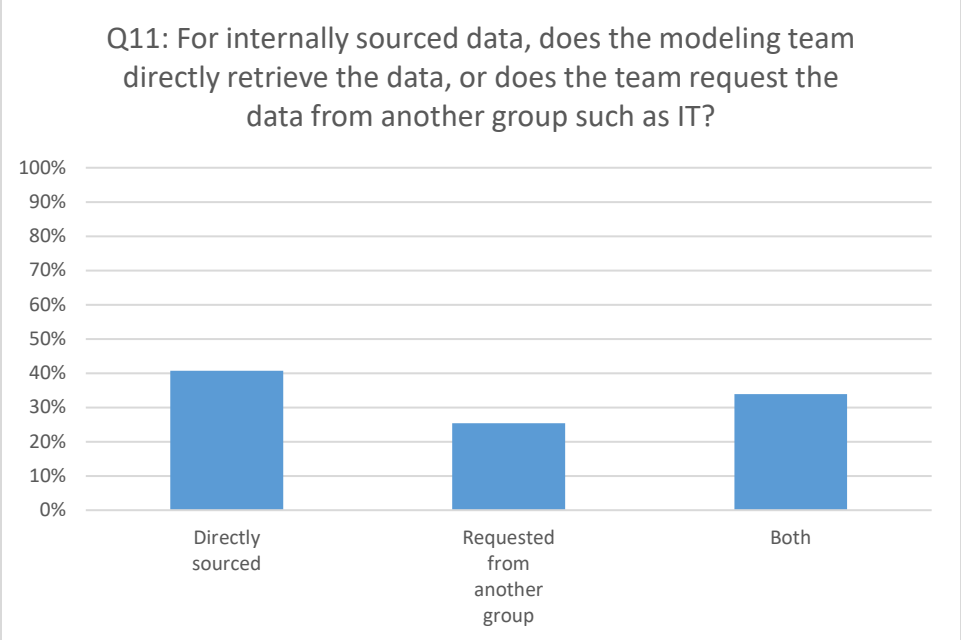


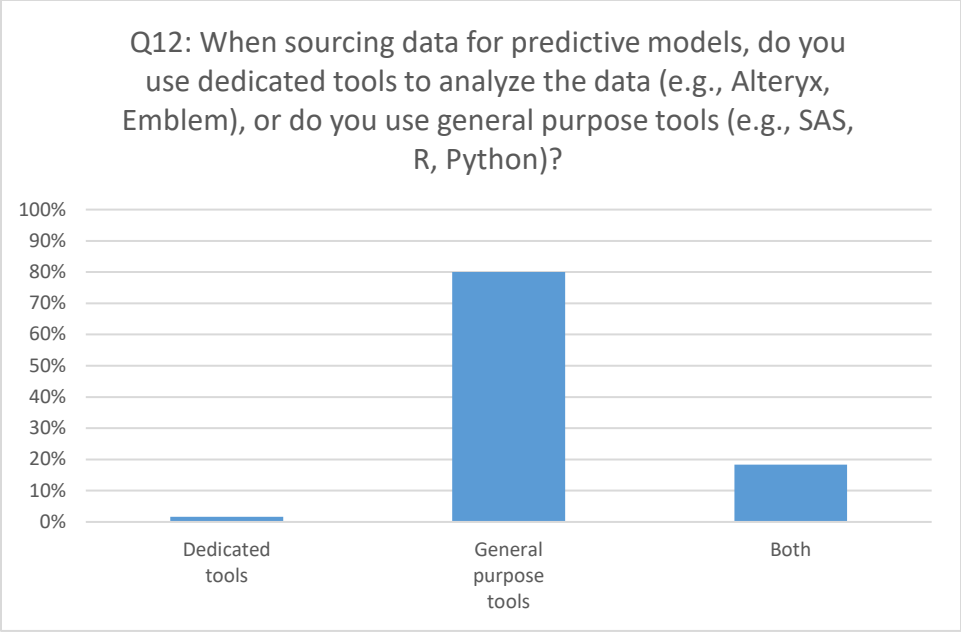


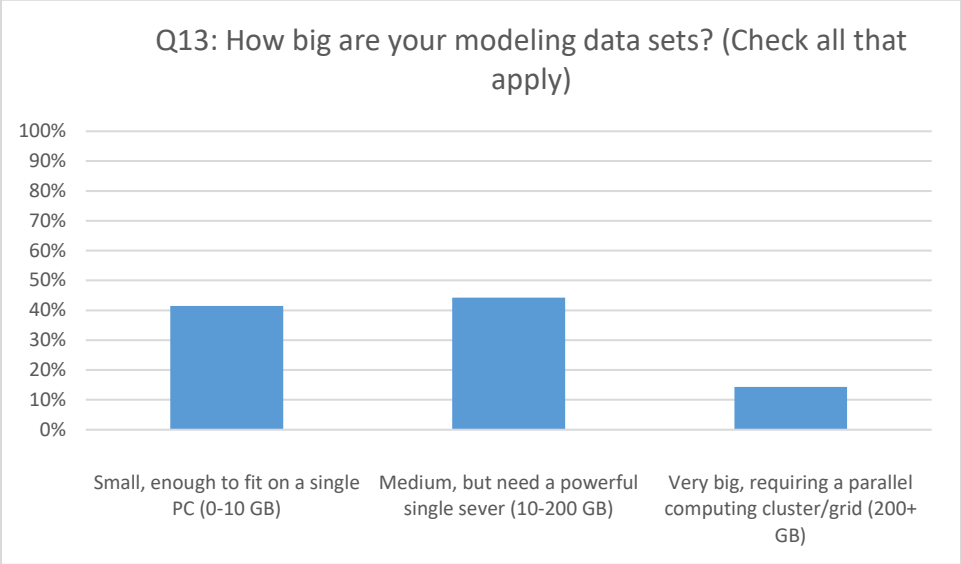


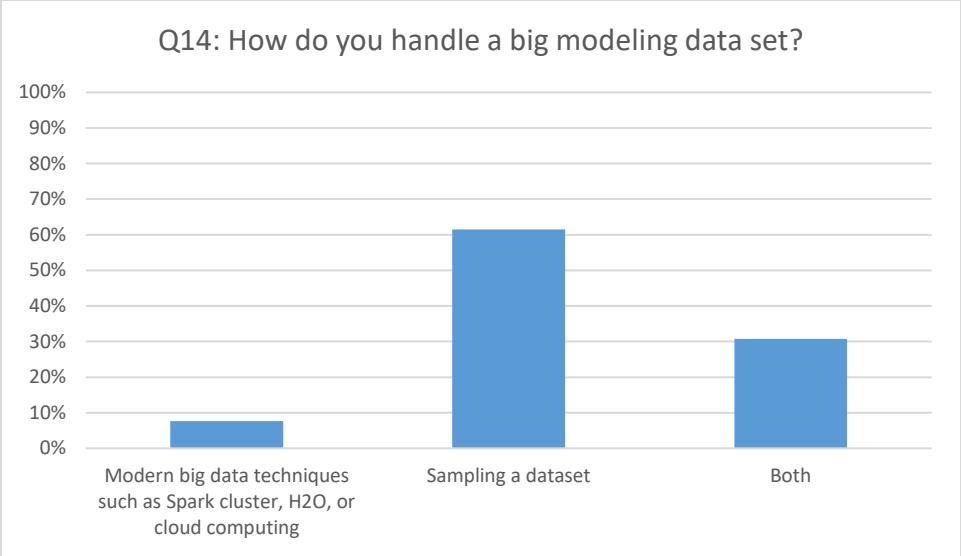
DATA ACQUISITION AND PREPARATION

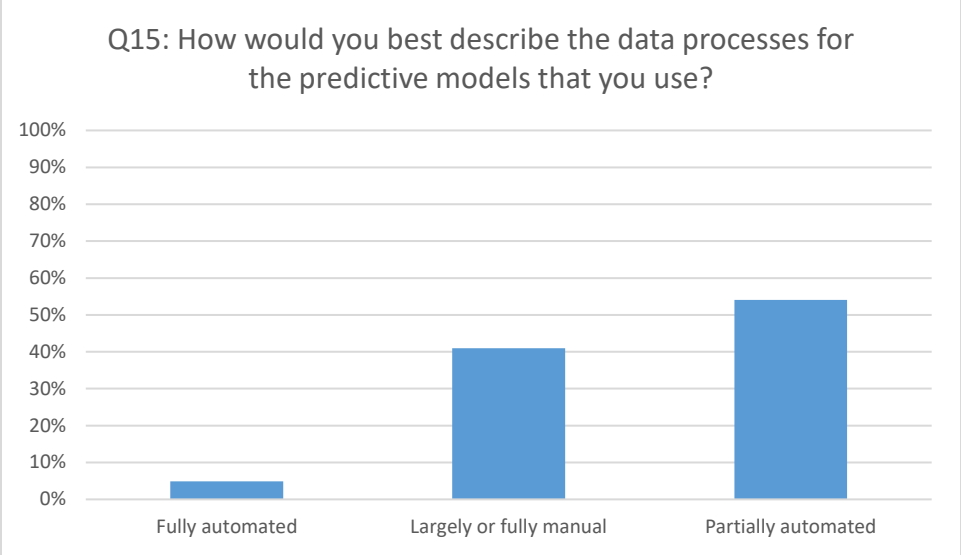




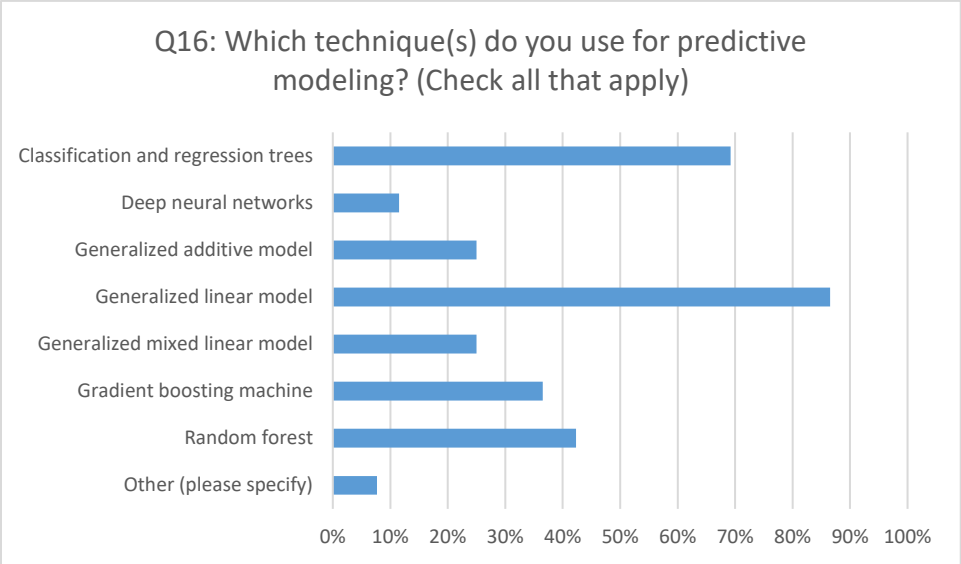


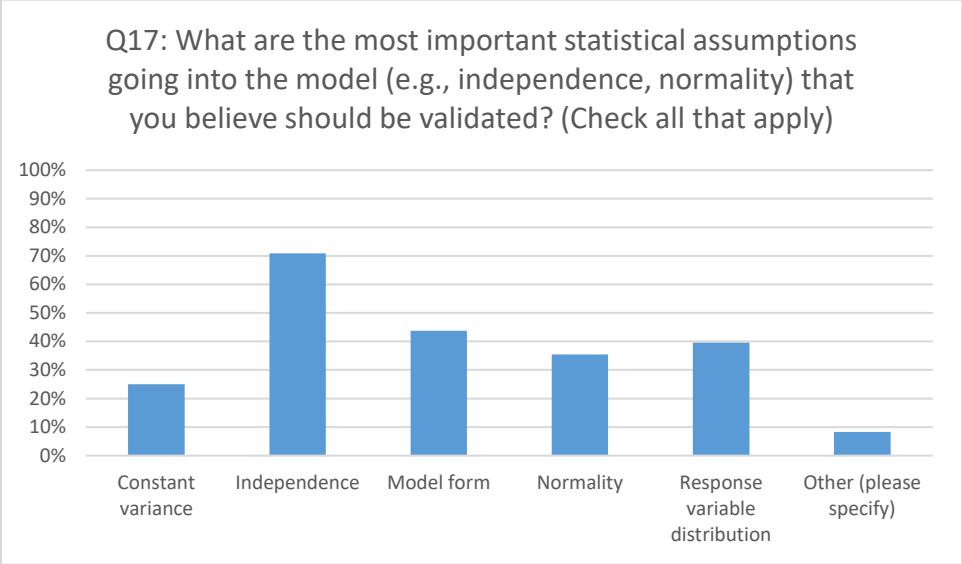


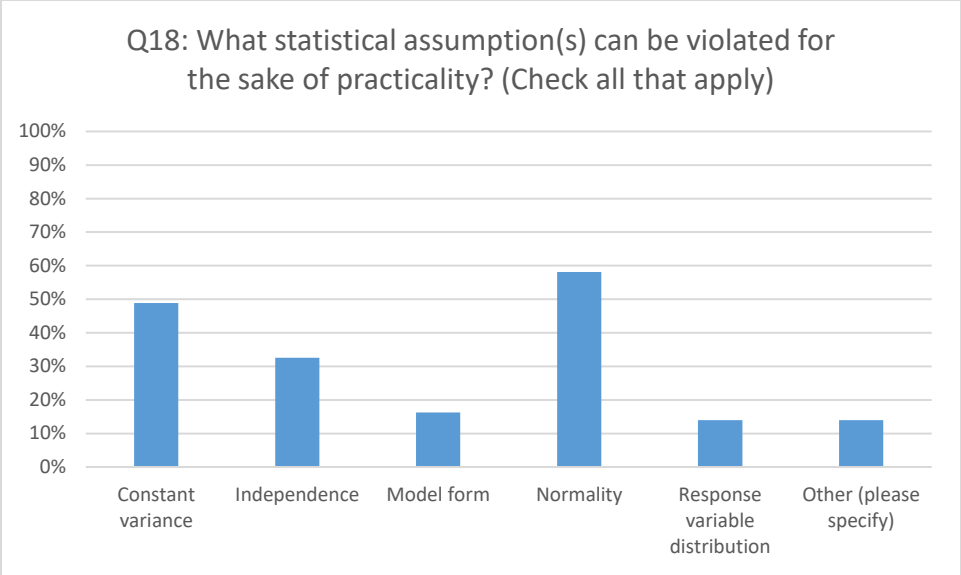


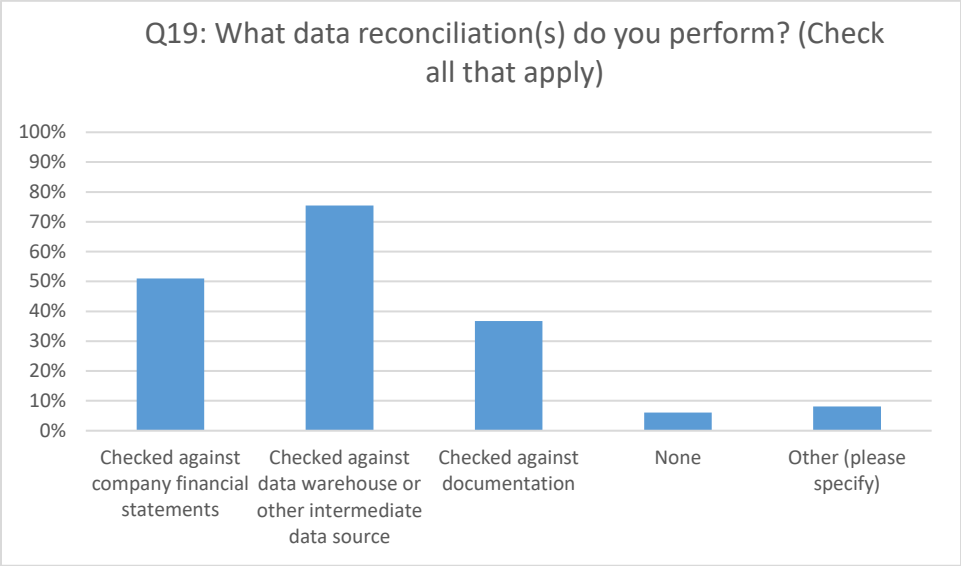


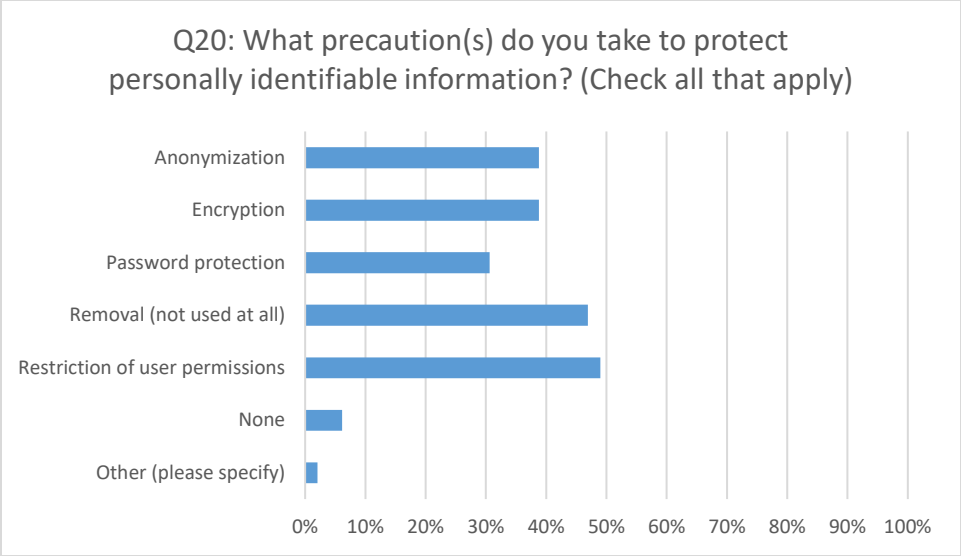
ALGORITHM SELECTION

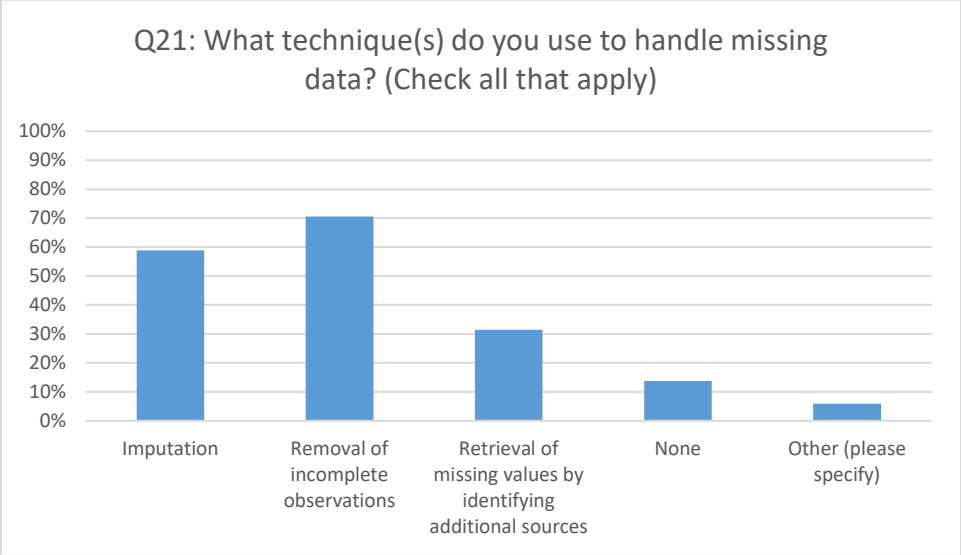


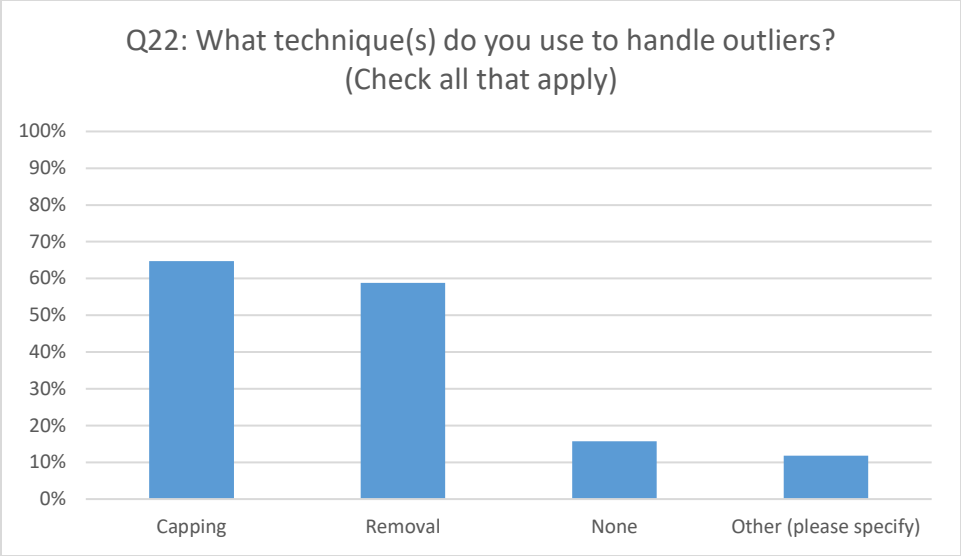


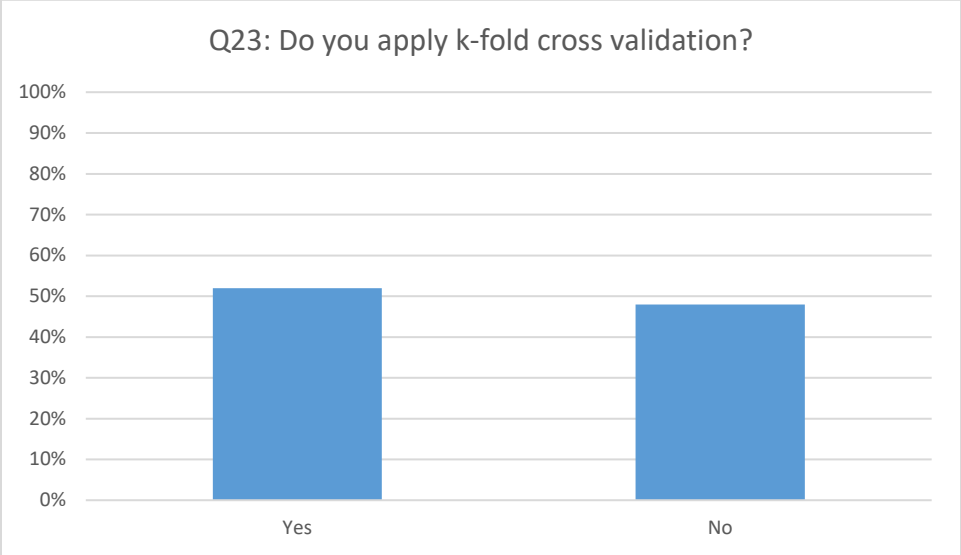


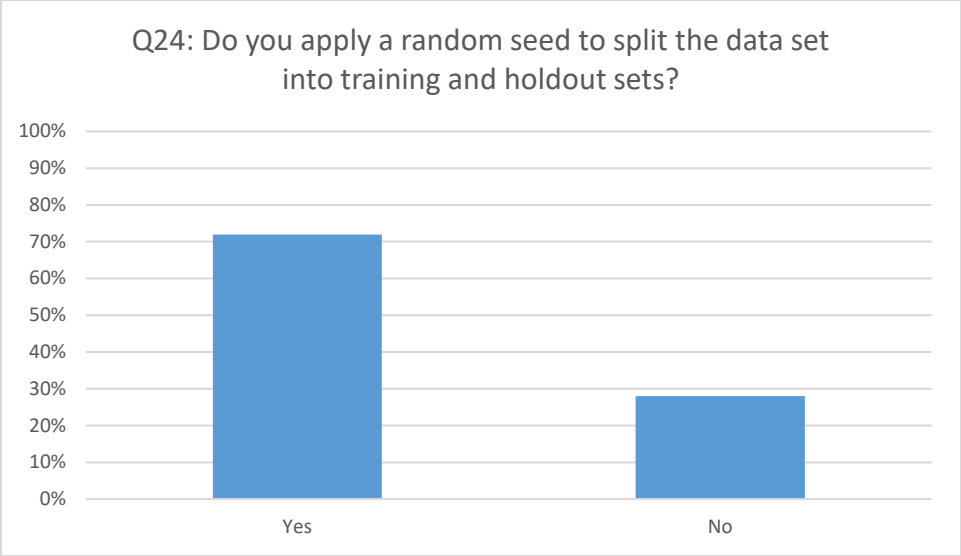


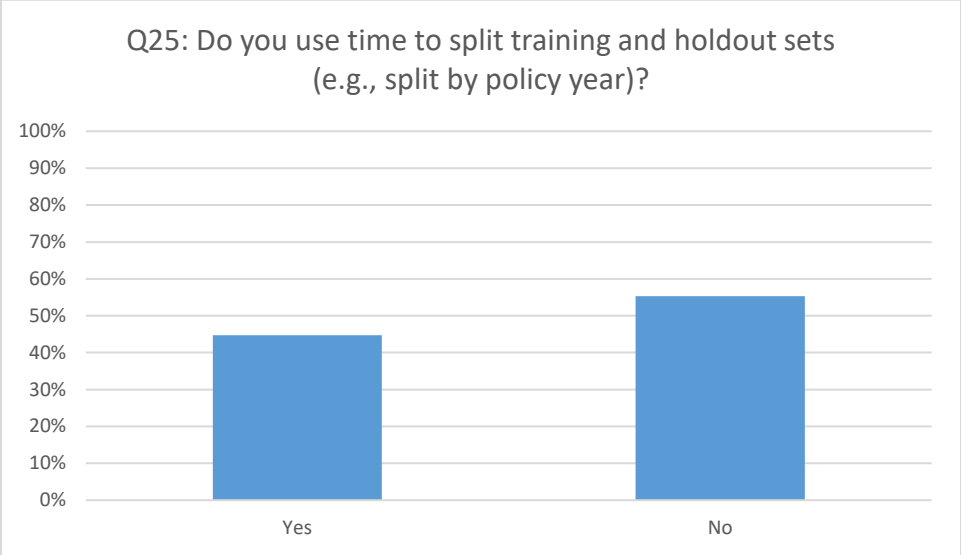


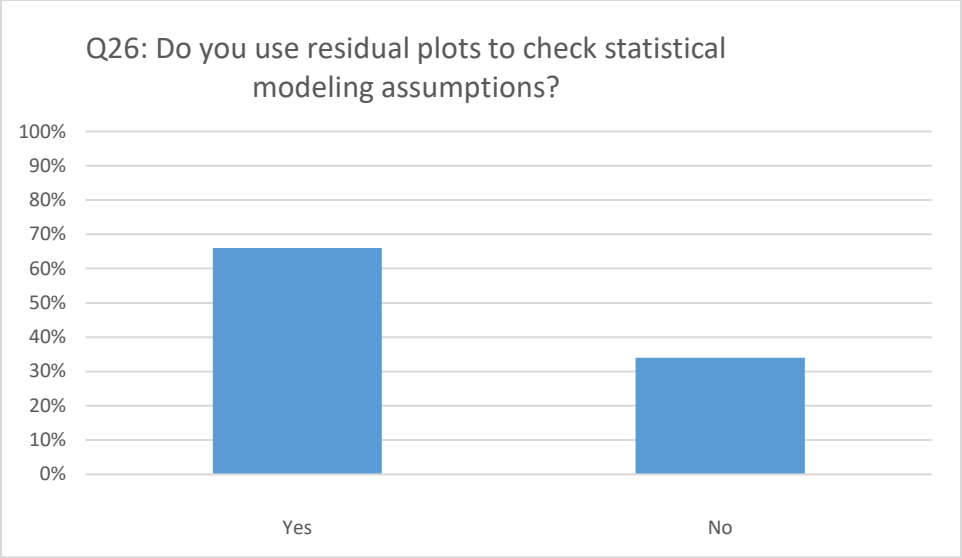


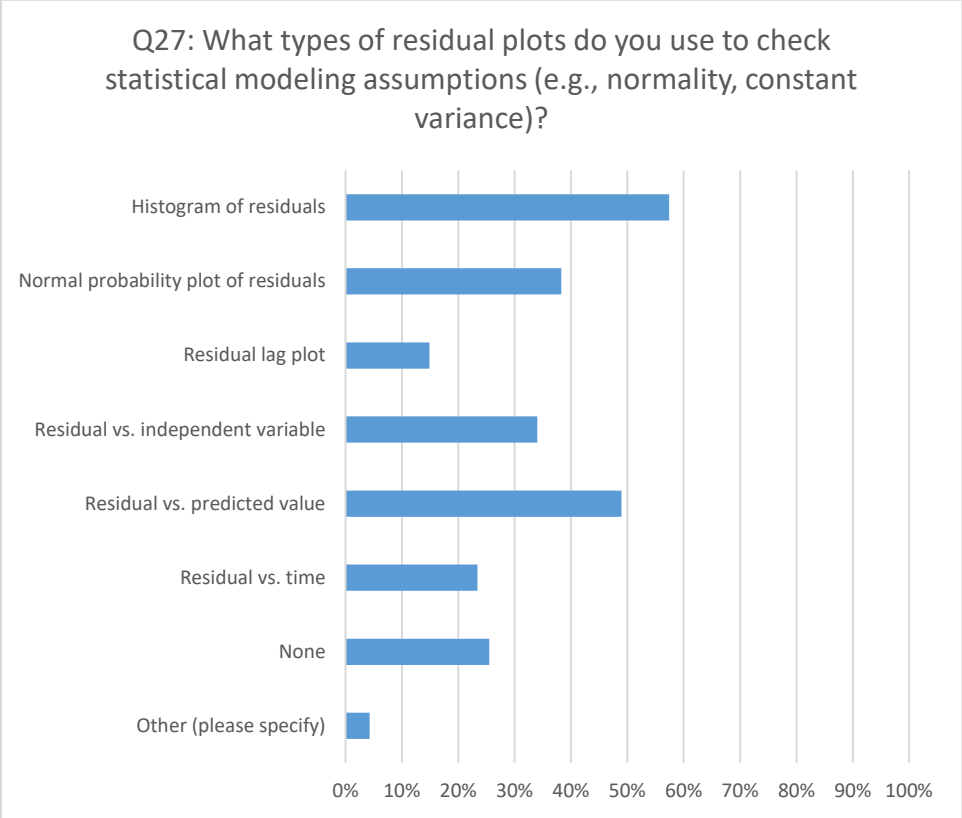




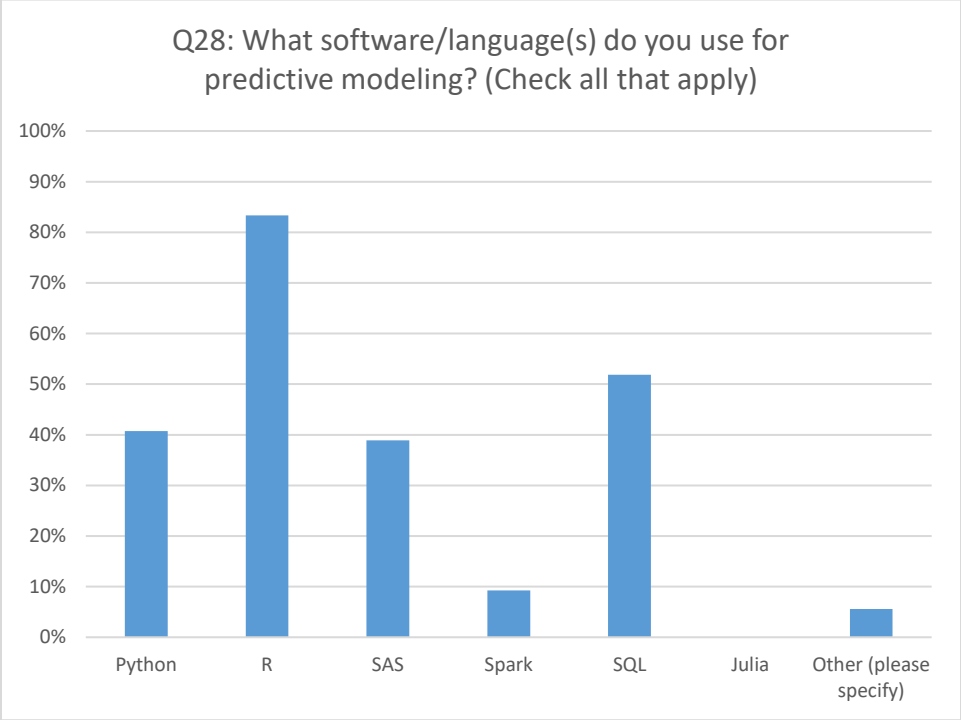




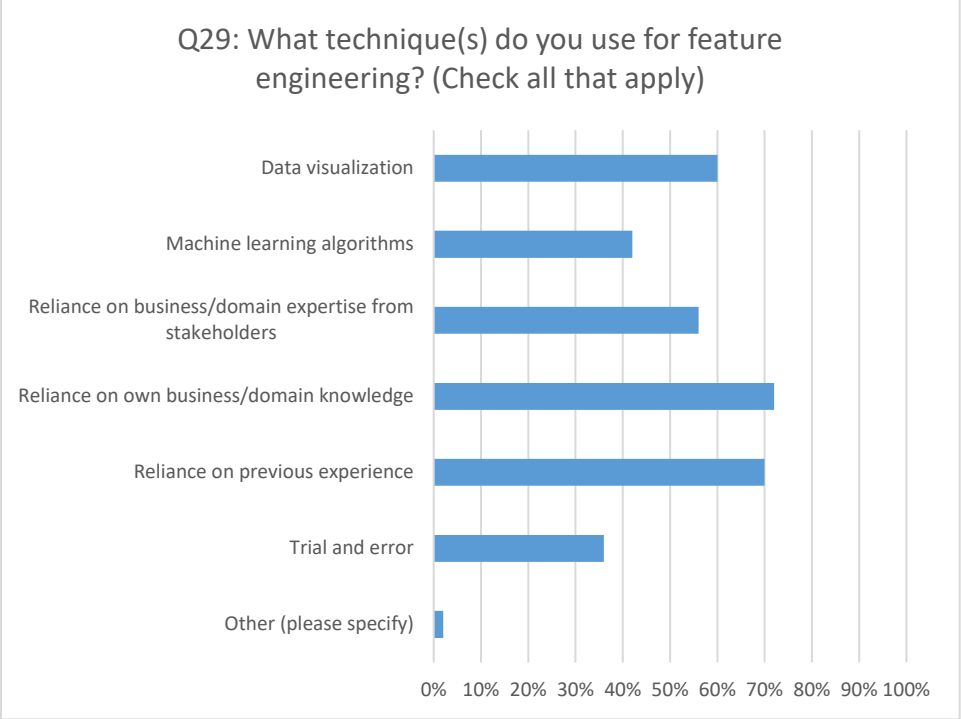


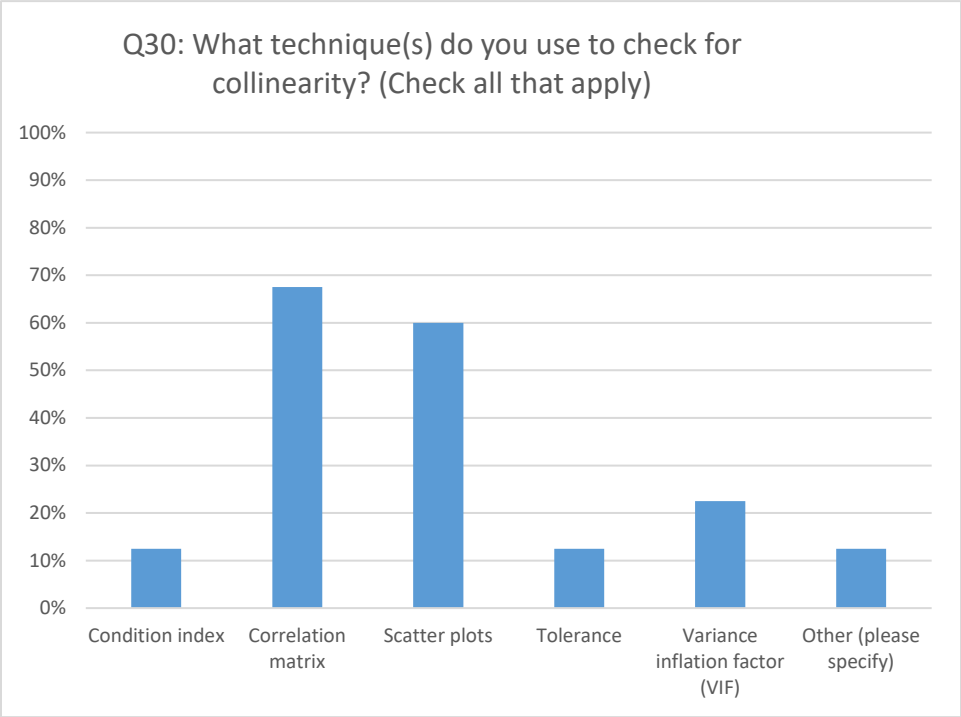


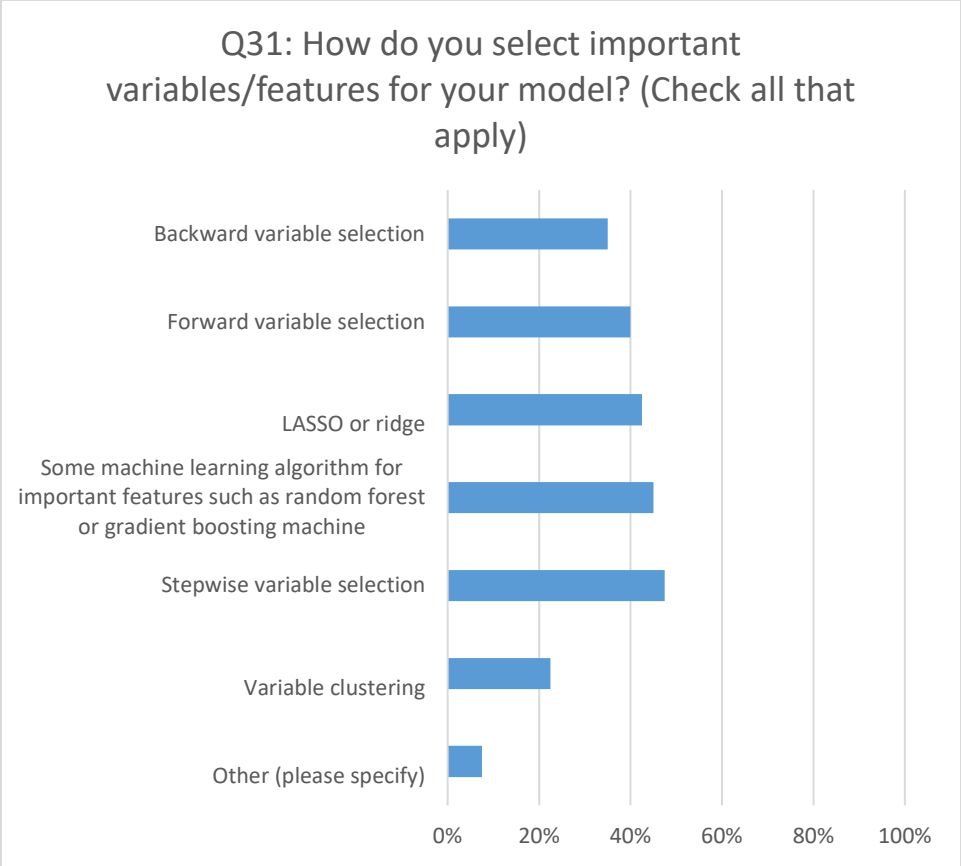
SOFTWARE SELECTION

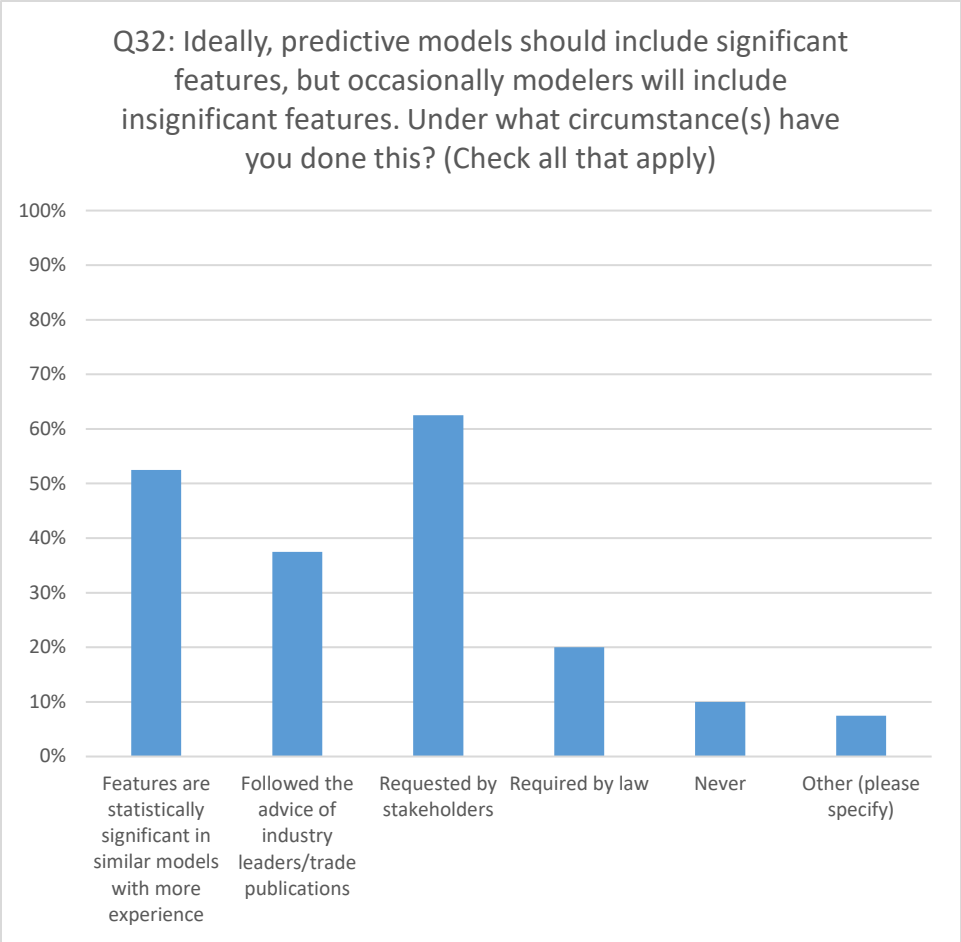


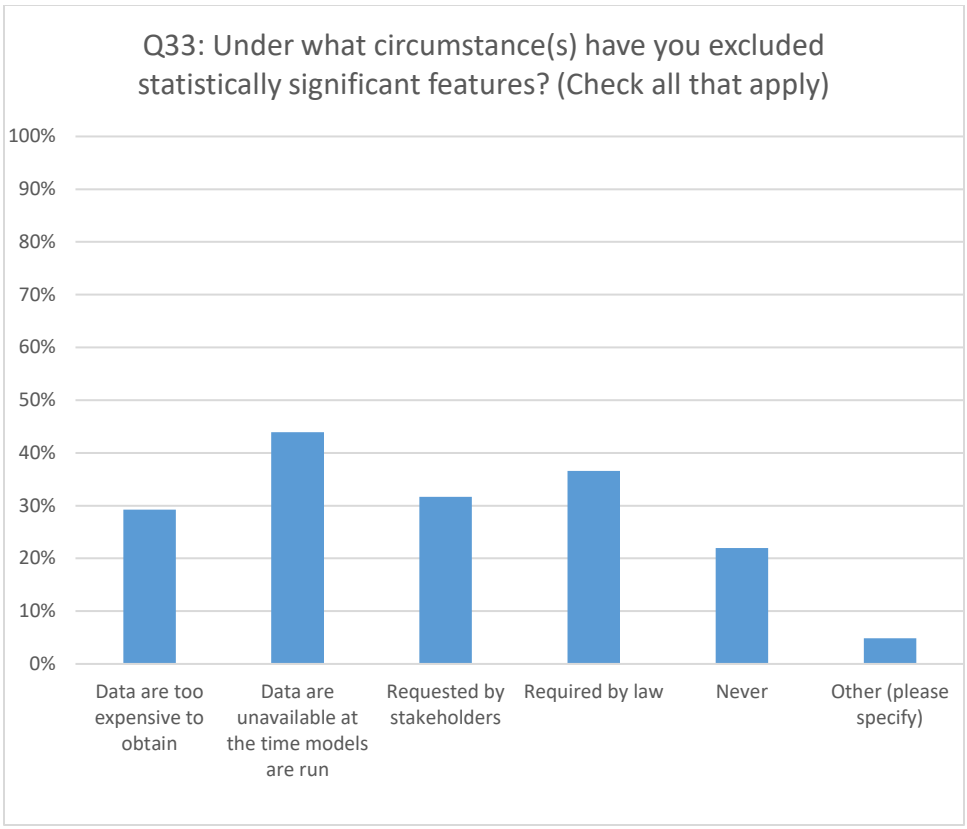
FEATURE ENGINEERING AND SELECTION



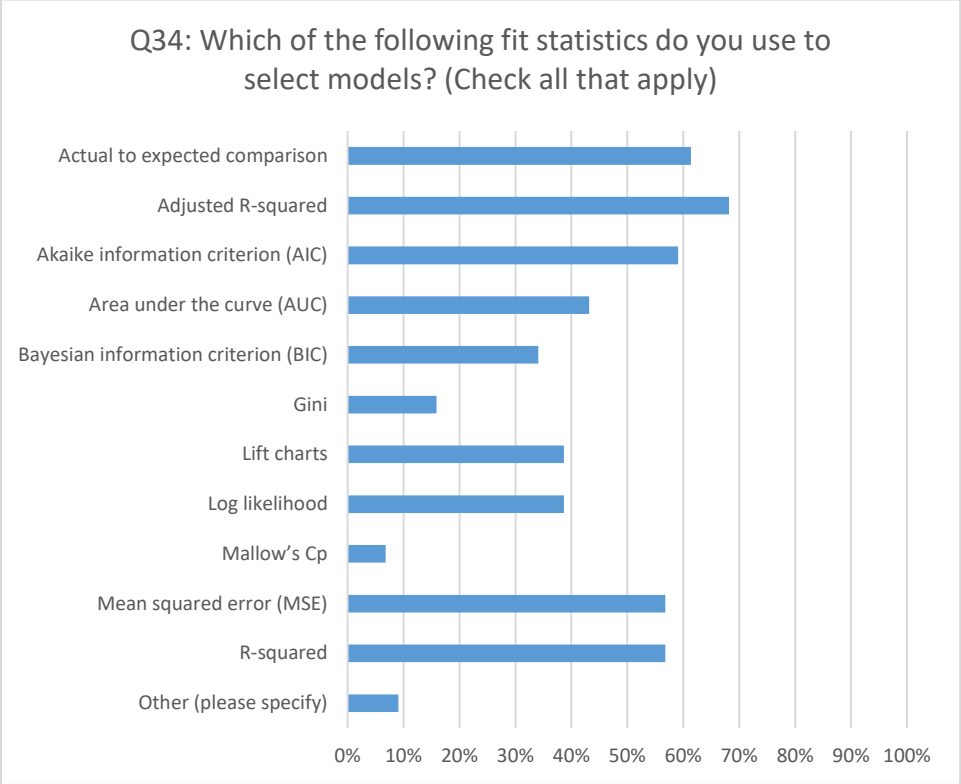




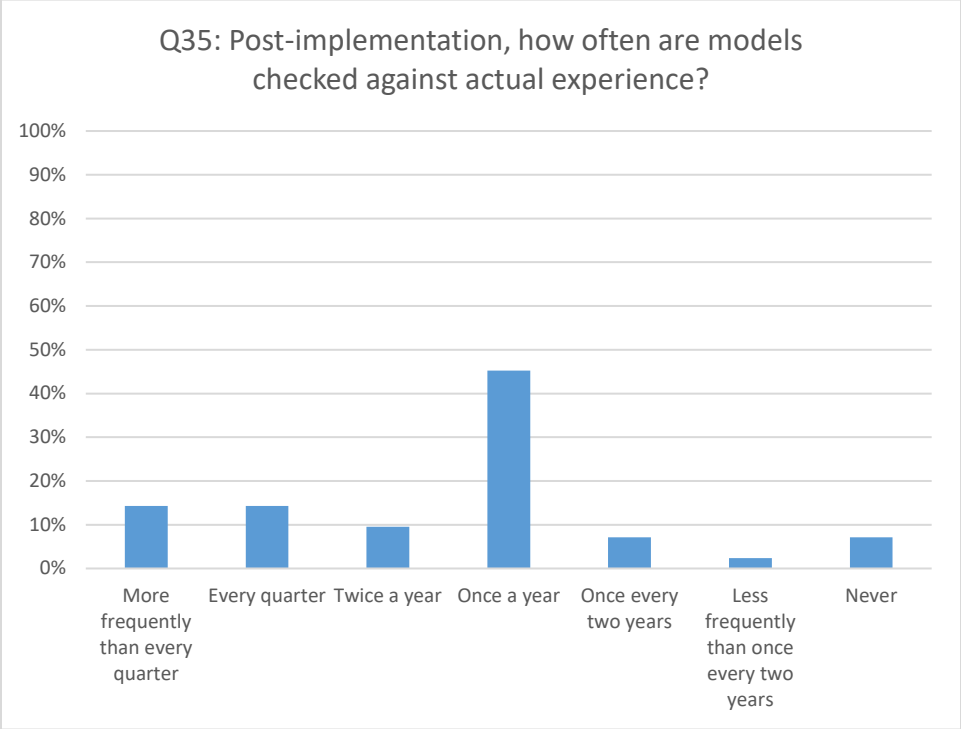


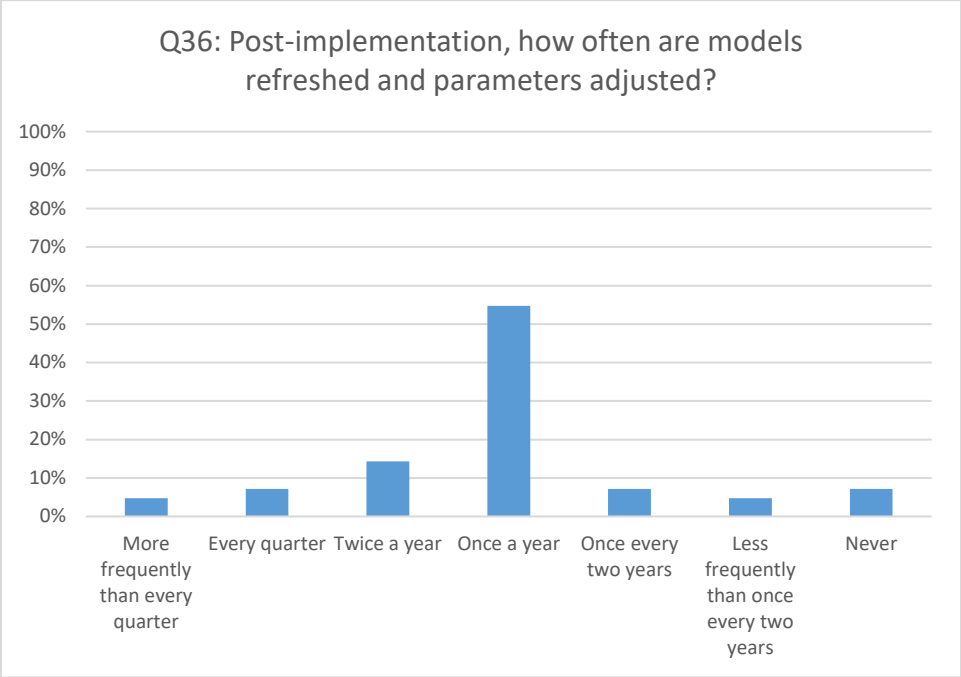


MODEL EVALUATION AND MEASUREMENT

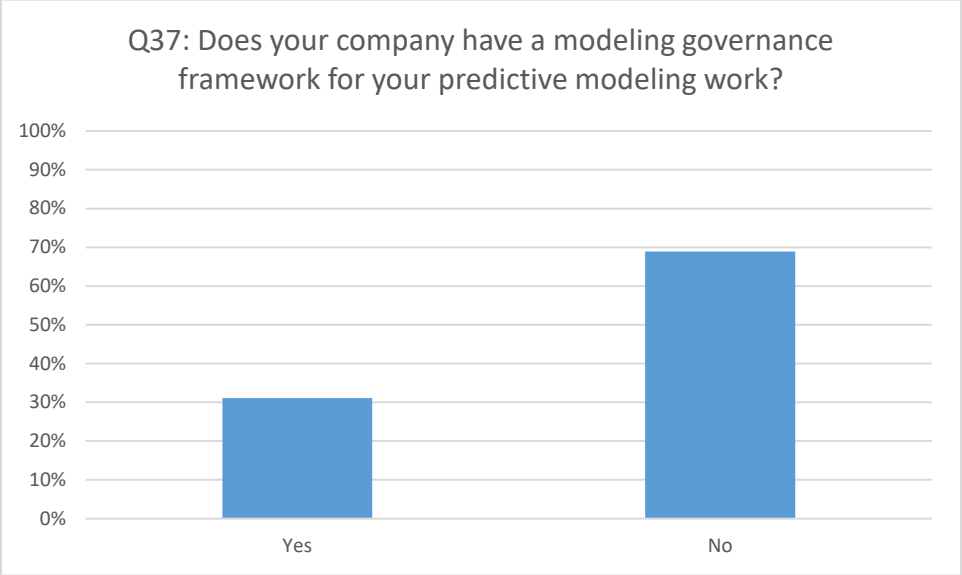


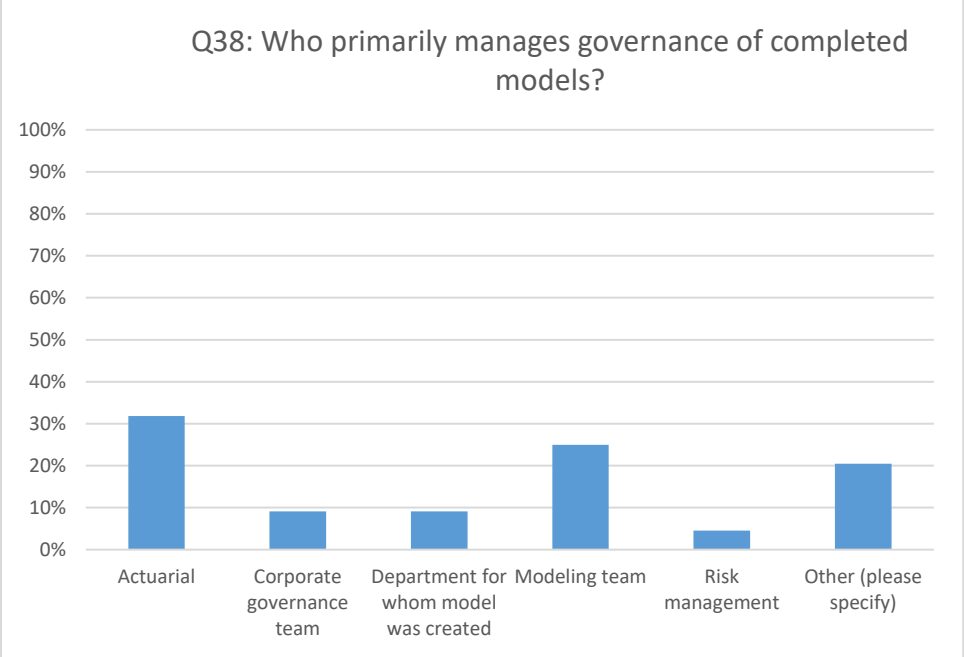
MODEL DEPLOYMENT

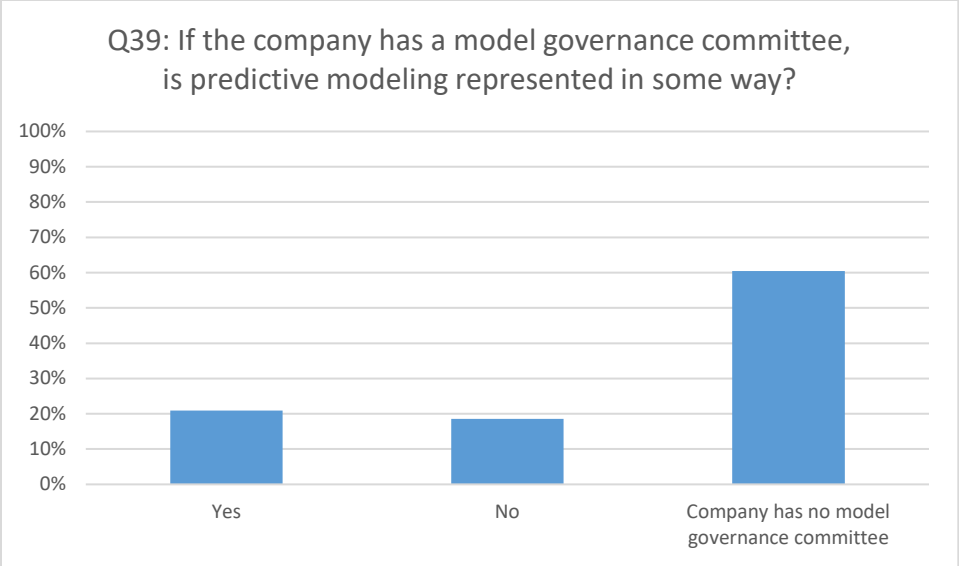


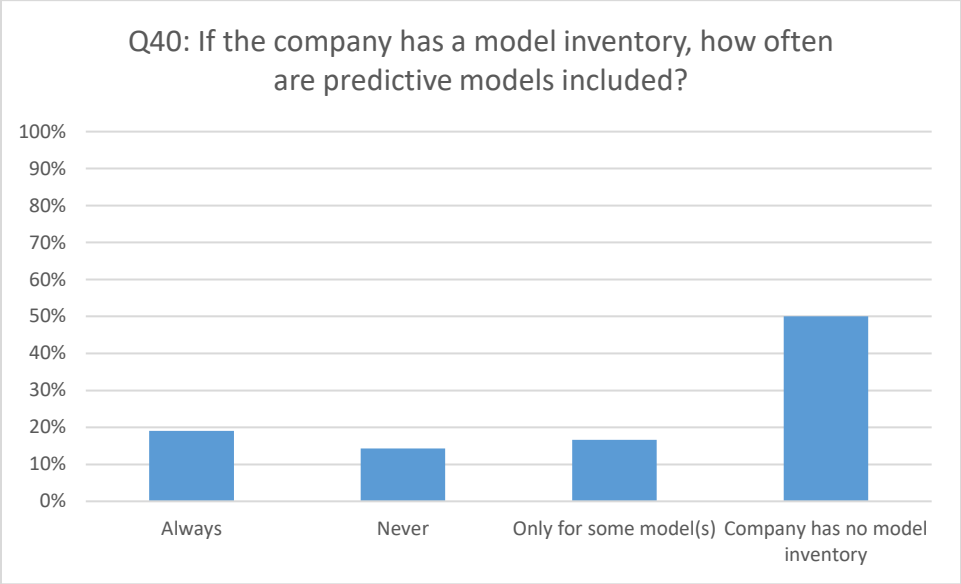


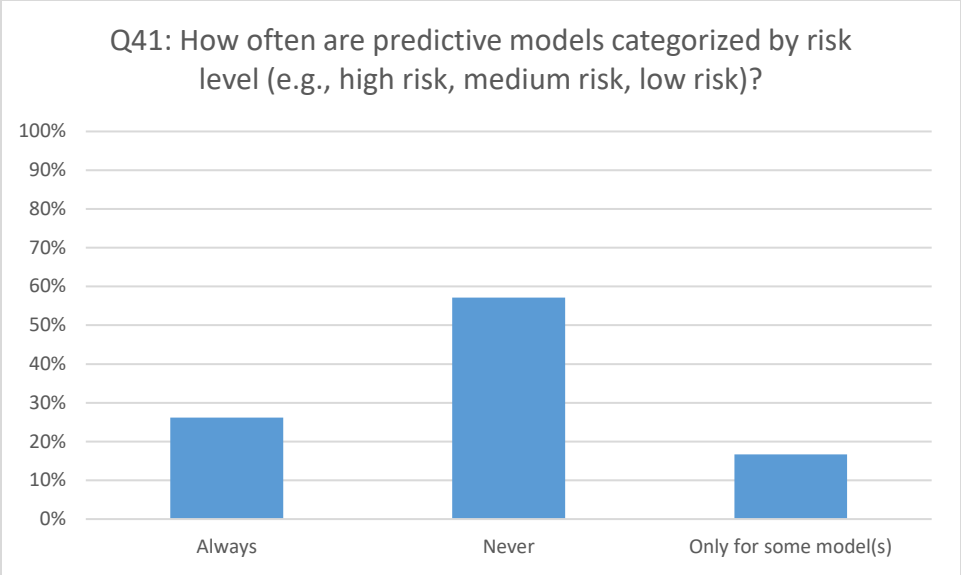
MODEL GOVERNANCE

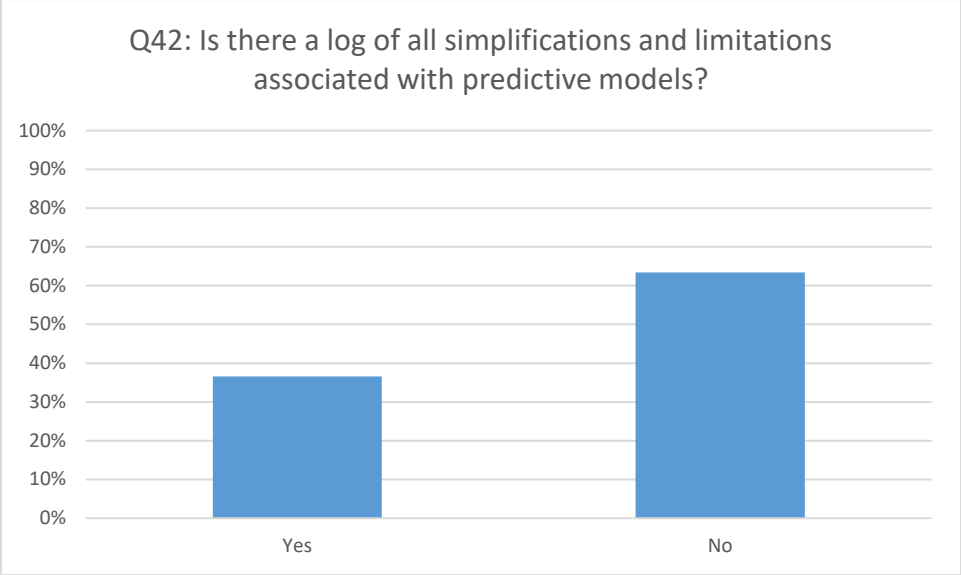


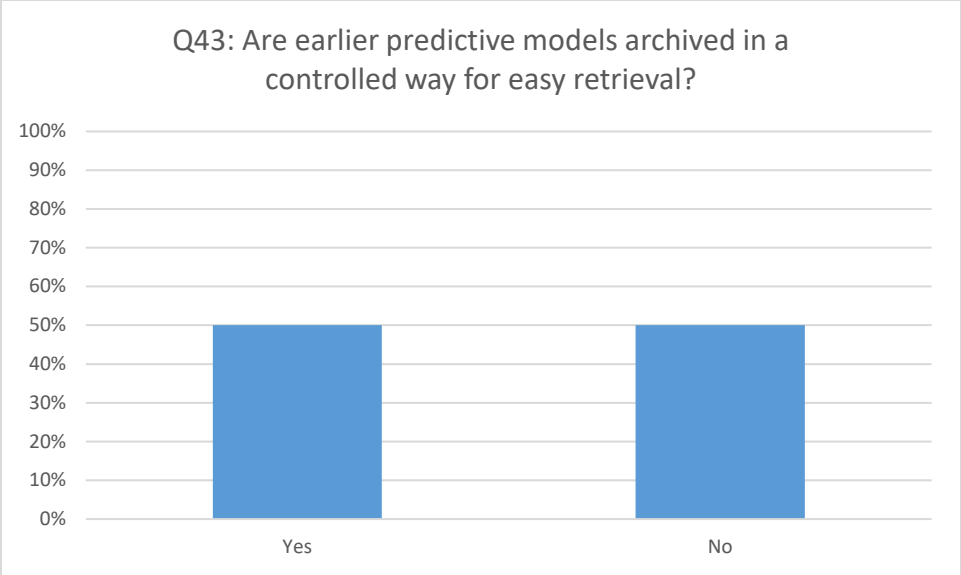


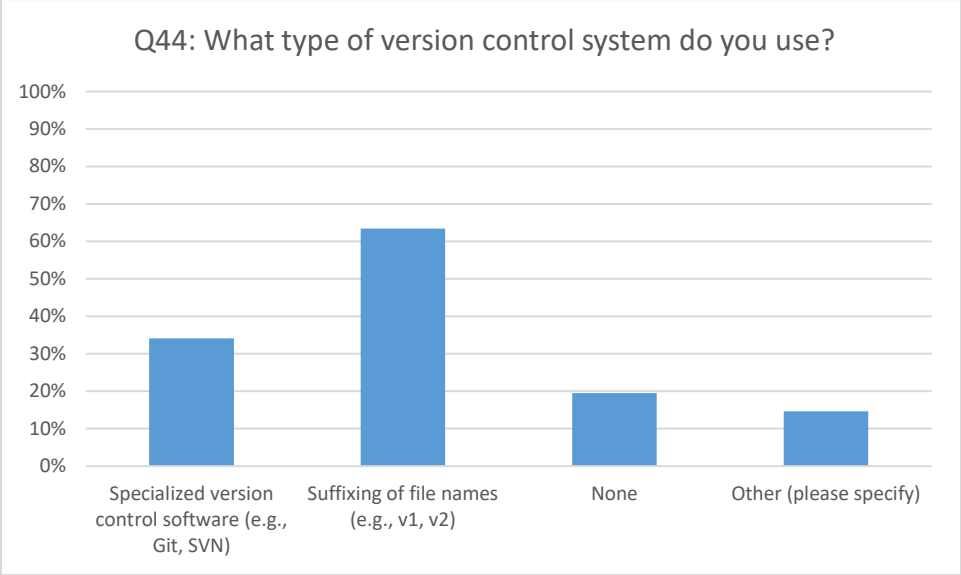


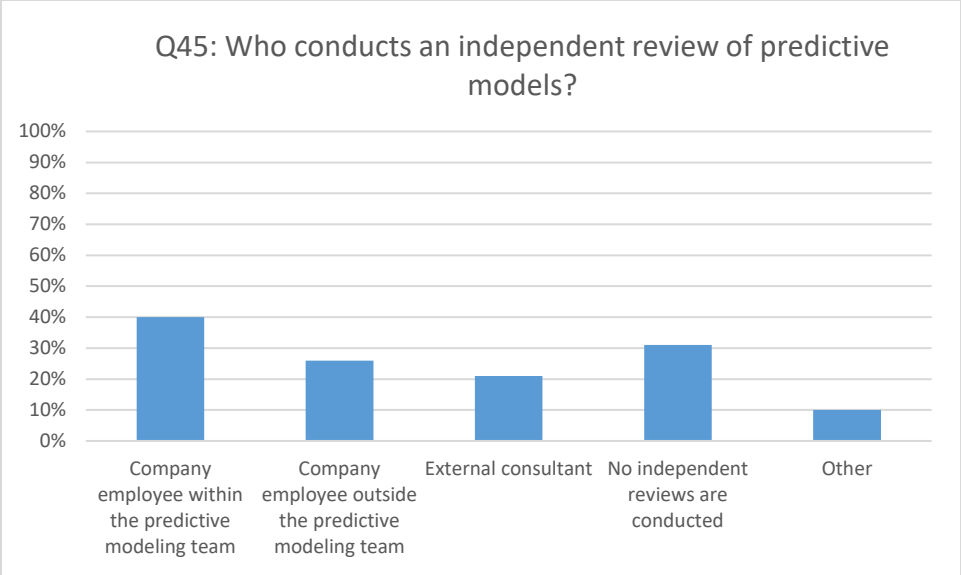


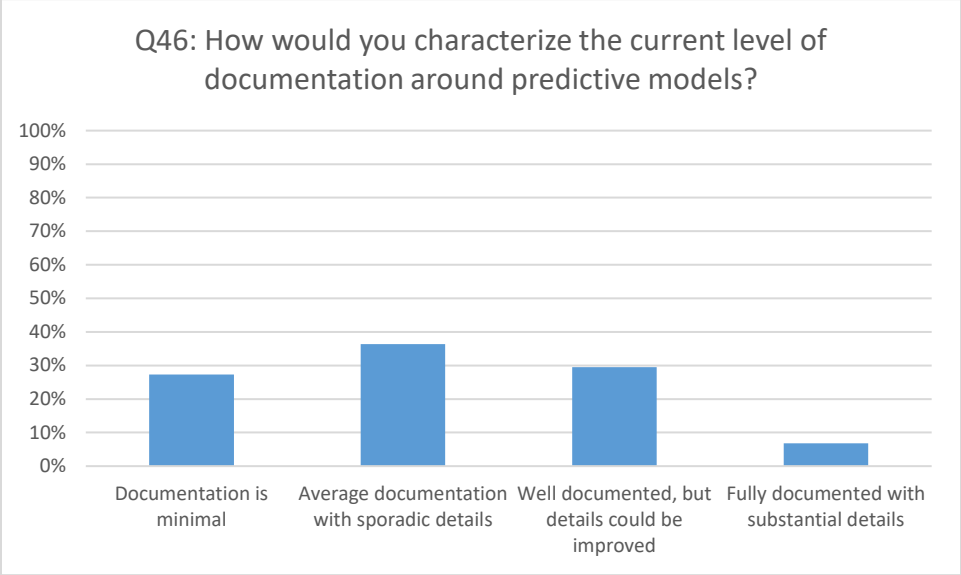












Appendix 2: Glossary

A/E ratio	A validation method performed by dividing actual (observed) values by expected (predicted) values within subintervals across a chosen dimension.
Adjusted R-squared	A modified version of R-squared that penalizes the R-squared metric for the addition of parameters to avoid overfitting.
Akaike information criterion (AIC)	A validation metric that estimates the goodness of fit for each model in a set of models while penalizing additional parameters to avoid overfitting. The model with the lowest AIC is preferred.
Application programming interface (API)	A set of rules that allows two software applications to communicate with each other.
Back-propagation	A technique used to calibrate the weight of each neuron in a neural network.
Bias-variance tradeoff	The tendency whereby predictive models with a low bias tend to have a high variance and vice versa. It is desirable to minimize both bias and variance but difficult to do so due to their inverse relationship. Data scientists seek to maximize predictive performance while balancing bias and variance.
Bayesian information criterion (BIC)	A model selection criterion that penalizes goodness of fit for additional parameters and observations; the model with the lowest BIC is preferred.
Clustering	A machine learning technique used to segment a data set into groups based on the similarities of observations within those groups.
Collinearity	A strong linear relationship between two independent variables in a regression model.
Correlation matrix	A matrix in which each cell contains the correlation coefficient between two variables.
Cross validation	A class of validation techniques that tests the performance of predictive models via the partitioning of a data set.
Data dictionary	A catalog of definitions of data elements, data types, relationships to other data, origin, usage and format.
Data governance	The oversight and management of data quality throughout an enterprise, focusing on integrity, availability, security and usability.
Data pipeline	A computational framework that coordinates the flow of data throughout an organization from the source of the data to the end users of that data.

Data warehouse	A centralized repository of data collected from sources internal and external to the enterprise, used to facilitate analytical processing of that data.
Decision tree	A flowchart-like schema represented with a tree structure that is used to build classification and regression models by grouping categorical or numerical data sets into smaller subsets and organizing data points into decision nodes, leaf nodes and root nodes.[141]
Deep neural network	A complex set of algorithms modeled loosely off biological neurons that map inputs to outputs and are able to find correlations between data sets to assist in classification, clustering and predictive analytics.[142]
Dimensionality	Refers to the number of attributes in a data set.
Exploratory data analysis (EDA)	An initial stage of the data modeling process where an analyst discovers patterns amongst data using graphing techniques like histograms and scatterplots.
Extrapolation	The process of estimating a phenomenon beyond the range of values observed in a data set.
Features	A variable used as a predictor or candidate predictor for a model. A measurement of an observed phenomenon.
Generalized linear model (GLM)	A generalization of ordinary linear regression that enables the prediction of a response variable that has a distribution other than a normal distribution. This generalization is enabled via a link function that relates the response variable to a linear predictor.[143]
Gini index	A coefficient that measures the magnitude of inequality among a set of observed values.
Git	An open-source distributed version control system used to coordinate project management.
Gradient boosting machines (GBM)	A machine learning technique for classification and regression problems that produces a model by building upon a successive ensemble of weak trees, with each successive tree learning from a previous tree.[144]

[141] Sayad, Saed. n.d. Decision Tree – Classification. An Introduction to Data Science. *https://www.saedsayad.com/data_mining_map.htm* (accessed January 23, 2019).
[142] Aggarwal, Charu. 2018. *Neural Networks and Deep Learning: A Textbook.* New York: Springer.
[143] Goldburd, Mark, Anand, Khare, Dan, Tevet. Generalized Linear Models for Insurance Ratemaking. Casualty Actuarial Society, 2016, *https://www.casact.org/pubs/monographs/papers/05-Goldburd-Khare-Tevet.pdf* (accessed February 21, 2019).
[144] U.C. Business Analytics. n.d. Gradient Boosting Machines. *http://uc-r.github.io/gbm_regression* (accessed February 21, 2019).

HIPAA	The Health Insurance Portability and Accountability Act is a regulation that aims to protect people from the misuse of their personally identifiable health care information.[145]
Imputation	The process of replacing missing data with substituted values.
k-fold cross validation	An iterative cross validation technique that partitions a data set into *k* equal-sized subsets. Each subset is used as a holdout set against which a model fitted to the rest of the data is tested.
Lasso	A regression analysis method that performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the statistical model it produces.[146]
Lift chart	Measures the effectiveness of models by calculating the ratio between the result obtained with a model and the result obtained from a random prediction or another model.
Log likelihood	The natural logarithm of the likelihood. The likelihood is the function of parameters, given a set of observations, that we seek to maximize to obtain the optimal parameters for a model. In practice, maximizing the log-likelihood is computationally more efficient and equivalent to maximizing the likelihood.[147]
Machine learning	A branch of artificial intelligence and a method of data analysis that automates analytical model building based on the concept that systems learn from data, search for better data representations, and make decisions without using explicit instructions.
Mallows's Cp	A measure of fit used in ordinary least squares that seeks to avoid overfitting by penalizing for the number of parameters. Given a set of models, the value with the smallest Mallow's Cp is desirable.[148]
Mean absolute error (MAE)	The average of the absolute values of all the errors found when comparing predicted values with observed values. Taking the absolute value focuses on the magnitude of the error, ignoring the direction.
Metadata	Data describe other data. Metadata provide basic information on data, which make them easier to understand, find and organize with particular instances of data.

[145] U.S. Department of Health and Human Services. n.d. Health Information Privacy. *https://www.hhs.gov/hipaa/index.html* (accessed February 21, 2019).
[146] Tambe, Milind, and Rice, Eric (Eds.). Artificial Intelligence to Predict Intimate Partner Violence Perpetration. 2018. Artificial Intelligence and Social Work in *Artificial Intelligence for Social Good*, 195-210. Cambridge: Cambridge University Press. doi:10.1017/9781108669016.013.
[147] Taboga, Marco n.d. Log-likelihood. *StatLect. https://www.statlect.com/glossary/log-likelihood* (accessed January 23, 2019).
[148] Mallows, Cohn. (1973). Some Comments on CP. *Technometrics*, 15(4), 661-675. doi:10.2307/1267380.

Missing values	Absent data for the variable in an observation. Missing values are a common occurrence, and inaccurate inference can be drawn from the data if missing values are not handled properly.
Mean squared error (MSE)	The average of the squares of all the errors found when comparing predicted values with observed values. It tells how close the fitted line is to a set of points. Squaring removes negative signs and gives more weight to bigger errors.
Multicollinearity	Two or more variables in regression or linear model that are moderately or highly correlated.
Neural network	A series of algorithms used to recognize underlying relationships in a set of data through a process that is inspired by biological neurons.
Outlier	An observation point that is far from the value that the model predicts. Outliers can arise for a variety of reasons, such as incorrect entry of an observation during data collection or variability in the measurement.
Overfitting	Poor predictive performance that occurs when a model is fit too closely to characteristics or random noise that do not extrapolate to unseen data.
Principal components analysis	A dimensionality reduction technique that transforms high-dimensional and possibly correlated variables into a set of linearly uncorrelated variables while retaining data trends and patterns.
Random forest	A method used to build predictive models by training groups of decision trees on random subsets of data set features and then averaging out their predictions
Real-time scoring application	A program designed to generate real-time values by supplying predictive models with input data.
Regression testing	The process of testing changes to software to ensure that previously developed software still works with the new changes.[149]
Regularization	In the context of regression, regularization is a method that seeks to balance fit against the complexity of the model. It does this by penalizing the likelihood function, or loss function. This helps to solve the overfitting problem. Some examples of this methodology include LASSO and Ridge regression.
Ridge regression	A method of regularization, ridge regression is a regression analysis method that penalizes the loss function by constraining the sum of the squared value of the model coefficients, known as the L2-

[149] TechTarget Network. n.d. Regression testing. *https://searchsoftwarequality.techtarget.com/definition/regression-testing* (accessed February 21, 2019).

	Norm. Ridge regression is a technique that can be used to analyze regression data that suffer from multicollinearity.
R-squared	A measurement of how close the data are to the fitted line. It is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
Scatter plot	A visualization that shows the correlation between two variables by plotting data points on a horizontal and a vertical axis. Each member of the data set gets plotted as a point whose x-y coordinates depends on its values for the two variables.
Semi-structured data	Types of data that contain semantic tags but do not conform to structure associated with typical relational databases. Examples include data stored in email, XML and other markup languages.
Semi-supervised machine learning	Learning algorithms trained on a combination of labeled and unlabeled data, where labels created by a human user are used as a starting point from which the machine is able to be trained on unlabeled data.[150]
Spearman's correlation	The Pearson correlation coefficient between ranked variables. Spearman's correlation coefficient measures the strength and direction of association between two ranked variables.
Structured query language (SQL)	A database language designed for managing data held in a relational database management system.
Standardization	The process of rescaling data to zero mean and unit variance.
Structured data	Cleary defined data types with a high degree of organization and that often have a maximum or expected size defined, making them easy to organize and query using a relational database or basic algorithms.
Supervised machine learning	A commonly used machine learning approach requiring a human user to identify and label the algorithm's outputs beforehand so that the machine can be taught or trained to produce a correct outcome from labeled data.[151]
Support vector machines	Generalizations of a simple and intuitive classifier called the maximal margin classifier. The goal is to have the largest possible margin between the decision boundary that separates the two classes and the training instances.[152]
Test data set	The sample of data not used for model fitting. The test data set is used to provide an unbiased evaluation of a final model fit on the training data set.

[150] Zhu, Xiaojin. n.d. Semi-Supervised Learning. Retrieved February 25, 2019, from http://pages.cs.wisc.edu/~jerryzhu/pub/SSL_EoML.pdf
[151] See footnote 70.
[152] *Ibid.*

Training data set	The sample of data used to fit the model.
Underfitting	A modeling error that occurs when the model is too simple to learn the underlying trend of the data.
Unstructured data	Data that are not organized in any discernable manner with no associated data model, making them more challenging to search using traditional relational database models. Examples include formats like audio, video and social media posts.
Unsupervised machine learning	A branch of machine learning that learns from test data that has not been labeled, classified, or categorized. Instead of responding to feedback, unsupervised learning identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data.[153]
Variable selection	The process of selecting a subset of relevant features (variables, predictors) to train on among existing features.
Variance inflation factor	A metric used to detect multicollinearity, based on the extent to which one variable can be predicted from the other variables in a model. [154]
Version control software	A software utility that tracks and manages changes to a file or a set of files to allow people to recall a specific version later.

[153] See footnote #70.
[154] See footnote #35.

Limitations

DISTRIBUTION

In conducting our analysis, Milliman relied upon survey data submitted by SOA members, an extensive literature search, and an insurer case study. Milliman did not audit or independently verify any of the information furnished, except that we did review the data for reasonableness and consistency. To the extent that any of the data or other information supplied to us was incorrect or inaccurate, the results of our analysis could be materially affected. This report is intended for the benefit of the SOA. Although the authors understand that this report will be made widely available to third parties, Milliman does not assume any duty or liability to such third parties with its work. In particular, the results in this report are technical in nature and are dependent on certain assumptions and methods. No party should rely upon these results without a thorough understanding of those assumptions and methods. Such an understanding may require consultation with qualified professionals. This report should be distributed and reviewed only in its entirety. This report is subject to the agreement between Milliman and the SOA dated June 12, 2018. We—Eileen Burns, Gene Dan, Anders Larson, Bob Meyer and Zohair Motiwalla—are consulting actuaries for Milliman. We are members of the American Academy of Actuaries and meet the qualification standards of the Academy to render the actuarial opinions contained herein.

About The Society of Actuaries

The Society of Actuaries (SOA), formed in 1949, is one of the largest actuarial professional organizations in the world dedicated to serving more than 32,000 actuarial members and the public in the United States, Canada and worldwide. In line with the SOA Vision Statement, actuaries act as business leaders who develop and use mathematical models to measure and manage risk in support of financial security for individuals, organizations and the public.

The SOA supports actuaries and advances knowledge through research and education. As part of its work, the SOA seeks to inform public policy development and public understanding through research. The SOA aspires to be a trusted source of objective, data-driven research and analysis with an actuarial perspective for its members, industry, policymakers and the public. This distinct perspective comes from the SOA as an association of actuaries, who have a rigorous formal education and direct experience as practitioners as they perform applied research. The SOA also welcomes the opportunity to partner with other organizations in our work where appropriate.

The SOA has a history of working with public policymakers and regulators in developing historical experience studies and projection techniques as well as individual reports on health care, retirement and other topics. The SOA's research is intended to aid the work of policymakers and regulators and follow certain core principles:

Objectivity: The SOA's research informs and provides analysis that can be relied upon by other individuals or organizations involved in public policy discussions. The SOA does not take advocacy positions or lobby specific policy proposals.

Quality: The SOA aspires to the highest ethical and quality standards in all of its research and analysis. Our research process is overseen by experienced actuaries and nonactuaries from a range of industry sectors and organizations. A rigorous peer-review process ensures the quality and integrity of our work.

Relevance: The SOA provides timely research on public policy issues. Our research advances actuarial knowledge while providing critical insights on key policy issues, and thereby provides value to stakeholders and decision makers.

Quantification: The SOA leverages the diverse skill sets of actuaries to provide research and findings that are driven by the best available data and methods. Actuaries use detailed modeling to analyze financial risk and provide distinct insight and quantification. Further, actuarial standards require transparency and the disclosure of the assumptions and analytic approach underlying the work.

Society of Actuaries
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org