Classifying Risk in Life Insurance Underwriting While Addressing Human Needs

by

Jacob C. Perez

Dr. Szczurek, Advisor

Masters project submitted in partial fulfillment of the

requirements for the Master of Science in Data Science degree

in the College of Arts and Sciences of

Lewis University

Classifying Risk in Life Insurance Underwriting While Addressing Human Needs

Jacob C. Perez, *Graduate Student, IEEE*

***Abstract*—Risk assessment is a very important process in getting insurance, specifically for life insurance products. The majority of life insurance companies currently perform traditional underwriting, while some have explored algorithmic underwriting, there isn't substantial research on the subject. With traditional underwriting, underwriters attempt to make decisions on the risk level associated to each applicant by using actuarial tables, and then they price policies based on risk. With machine learning and predictive analytics on the rise, there is hope in improving the overall experience of life insurance, with huge benefits for the business (i.e. faster processing and less expenses on medical testing). A real world dataset provided by Prudential was used for this research. This dataset consists of over a hundred anonymized attributes, and over 59,000 applicants, or instances. Dimensionality reduction techniques were used to identify important features to improve the prediction of the classification models through Principal Components Analysis (PCA), and Random Forest techniques. The supervised machine learning algorithms Logistic Regression, Random Forests, and Multilayer Perceptron, were used to predict the risk level of each applicant, since these algorithms perform on multiple classes. The results showed that Random Forests using Random Forests as a feature selector performed the best with a Cohen Kappa Score of 0.393, Precision of 0.508, Recall of 0.534, and an F1 score of 0.496.**

I. INTRODUCTION

Risk assessment is an important step in classifying applicants for life insurance (LI). Companies use underwriting to make decisions on applications, and to price policies accordingly. Individual life insurance organizations still rely on conventional actuarial formulas to predict mortality rates and premiums of policies. LI companies have recently started carrying out predictive analytics to improve their business efficacy, but there is still a lack of extensive research on how predictive analytics can enrich the LI domain [1]. With advances in the amount and type of data available, along with similar advances in predictive analytics, underwriting can be supplemented for faster processing of applications as well as possibly more accurate classifications.

Manulife insurance company in Canada was the first to offer insurance to HIV suffering applicants through analyzing survival rates [2]. This is an example of how LI is changing to a more

customized approach. By using the results from medical research advances, they are able to adjust their underwriting indicators of survival to incorporate more people who were previously denied insurance. The increasing survival rates of those with HIV, for example, meant it was time to price them into existing products. Manulife uses publicly available data, like medical research or statistics, to make better assumptions about the population. As they gather this knowledge, they are less likely to rely on traditional underwriting. Their overall goal is to make the life insurance process less annoying by using information that they gather to benefit their consumers.

Analytics can also help in the underwriting process to provide the right premiums for the right risk to avoid adverse selection. Adverse selection refers to a situation where the insurers do not have all information on the applicant, and they end up giving policies to customers with a high-risk profile [3]. Insurance firms with business-savvy underwriting teams stress on making the least possible losses for the business. In other words, the insurers strive to avoid adverse selection, as it can have powerful impacts on the life insurance business. Adverse selection can be avoided by correctly classifying the risk levels of individual applications through predictive analytics, which is a goal of this research. Currently, this is a struggle for many organizations.

II. Motivation

When talking directly to people who are shopping for LI, the design team I work with was able to uncover qualitative insights around what people are thinking and feeling as they go through this process. These insights will inspire the goals of the research.

The first insight is that when people receive out-of-reach LI quotes, it makes them feel like they've failed their families. Price is a very important factor in purchasing LI, and it is our intuition that people want more accurate quotes so they can actually afford a policy. Shoppers usually blame themselves for high rates, when in reality, it could be the way their risk is classified by the insurance company. The second insight is that people are wary of being cheated by LI companies if they don't understand what's going on. This means that the interpretability and explainability is important when classifying risk, so we will need to incorporate models that are not black boxes in order to understand what the data is telling us. The third insight is focused on how LI companies use health questions to determine policy price. This makes people feel defined by a perceived health condition, which is important because people who have a medical history don't like taking medical exams or having

companies determine their risk by things that are out of their control. Are there other ways to classify risk without sharing private health information or taking a medical exam that is fair? Lastly, people expect quotes to work the same way, regardless of what they're shopping for, which leads to a feeling of a bait-and-switch when their expectation isn't met. What they really want is to know is 'how much is this is going to cost?', with the least effort possible. This means it is crucial to surface the costs of life insurance upfront, without having to wait for traditional underwriting to give an answer.

These insights lead into my goals for this project by reframing the insights as machine learning questions, or objectives. First, I wanted to understand the current state of accuracy around risk classification through supervised machine learning classifiers. By understanding which factors are contributing to the classification error, I'll be able to discover how it can be improved. I'd like to be able to explain how and why the risk was classified the way it was by analyzing which features contribute most to an accurate classification. I also wanted to learn what the minimal amount of data is needed to provide an accurate life insurance rate, and if the process is explainable to shoppers (will people trust and use this process)? Lastly, I'd like to know how much of an impact does medical data have on risk classification, and what can be used in its place.

III. Literature review

From here, it would be useful to gain an understanding around how LI companies are currently classifying risk. Risk profiles of individual applicants are thoroughly analyzed by underwriters. The job of underwriters is to ensure that the risks are evaluated, and premiums are as accurate as possible. Risk classification is a common term used among insurance companies, which refers to grouping customers according to their estimated level of risks, determined by their historical data. For decades, LI firms have been relying on the traditional mortality tables and actuarial formulas to estimate life expectancy and devise underwriting rules. However, the conventional techniques are time-consuming, usually taking over a month, and are costly to the business. It is essential to find ways to make the underwriting process faster and more economical. Predictive analytics have proven to be useful in streamlining the underwriting process and improve decision-making [4][5]. However, extensive research has not been conducted in this area.

Underwriting involves gathering extensive information about the applicant [6]. This means that applicants usually undergo at least one medical test and need to submit all of the relevant documents to

the insurance company. Then, the underwriter assesses the risk profile of the customer and determines if the application can be accepted. Once that step is complete, the premiums are calculated [7]. On average, it takes at least 30 days for an application to be processed, which is too long for people who need coverage today, or within a few days, to protect their families. Due to the underwriting process being lengthy and time-consuming, customers are more likely to go to a competitor, or they might just abandon getting LI overall. With a lack of efficient underwriting methods, shoppers will most likely put it off until they can afford it, which sometimes is never, reinforcing the industry-wide decline of policy sales. Similarly, underwriting processes and the medical procedures required by the insurance company to profile the risks of the applicants can be expensive [8]. Usually, all the costs to perform the medical examinations are initially taken on by the company. Underwriting costs are usually fully paid from the contract and can last 10-30 years. In the cases where there is a policy lapse, the insurer incurs great losses [9]. All of this evidence supports the idea that the conventional underwriting processes can be supplemented by automation through predictive analytics. Predicting the significant factors impacting the risk assessment process can help to streamline procedures, making it more efficient and economical for both consumers and businesses.

IV. Methodology

A. Description of the dataset

The dataset for this project, see Fig. 1, was provided for a kaggle competition, hosted by Prudential, one of the largest issuers of LI in the U.S. Prudential's goal of making this data publicly available was to figure out how a predictive model can accurately classify risk by using an automated approach, which also aligns with my goal. Prudential's motive for the challenge echos what we heard before, which is that the LI application process is antiquated and people are turned off by it. This real-world dataset consists of 59,381 applications with 128 anonymized attributes, which describe the characteristics of the life insurance applicants. This dataset has a mix of nominal, continuous, and discrete variables.

Attributes	**Type**	**Description**
Product_Info_1-7	Categorical	7 normalized attributes for life insurance product applied for
Ins_Age	Numeric	Normalized age of applicant
Ht	Numeric	Normalized height of applicant
Wt	Numeric	Normalized weight of applicant
BMI (Body Mass Index)	Numeric	Normalized BMI of applicant
Employment_Info_1-6	Numeric	6 normalized attributes for employment history of applicant
InsuredInfo_1-6	Numeric	6 normalized attributes for info about applicant
Insurance_History_1-9	Numeric	9 normalized attributes for insurance history of applicant
Family_Hist_1-5	Numeric	5 normalized attributes for applicant's family history
Medical_History_1-41	Numeric	41 normalized variables for medical info of applicant
Medical_Keyword_1-48	Numeric	48 dummy variables for medical keyword associated with application
Response	Categorical	Target variable, an ordinal measure for risk, with values 1-8

Fig. 1, Description of Dataset

B. Exploratory Data Analysis

I began this project with Exploratory Data Analysis. This step helps identify the distributions of the features in order to select the best prediction models for the dataset. I used Tableau to get visual analytics on the data. After exploring a few attributes' distributions, I realized that the most useful were

the continuous attributes like age, BMI, height, weight, and lastly, the categorical attribute response. This was because of the nature of the data provided. Since all of the attributes were anonymized, it was hard to gain any insight from the other attributes since the descriptions were minimal on purpose. Meaning, the rest of the attributes don't provide enough data to dig deeper into posing hypotheses around how they affect the risk response at face value. The biggest insight here was that the response attribute, i.e. the risk class assigned, did not follow a normal distribution. I would expect the risk to be normally distributed across populations where most of the population carries average risk, and the lower and higher risk applicants are less common, although this is not the case for this dataset. This tells me that the actuaries that assign the risk class seem to be biased in some way, so the results of predictive algorithms will most likely also be biased towards the response classes. This could be because actuaries classify risk based on the traits of the individual, and don't take into consideration the individual's risks compared to the population. I'd like to dig deeper and see if there are any patterns in the data that lead to the response attribute to appear biased, since that is just a hypothesis at this point. Age, height, weight, and BMI, on the other hand, were skewed normal distributions. In Fig. 2, we see the age distribution with two different peaks, but otherwise a normal distribution. This graph showcases how LI applications usually take place around two different times in someone's life, but generally not too young or too old.

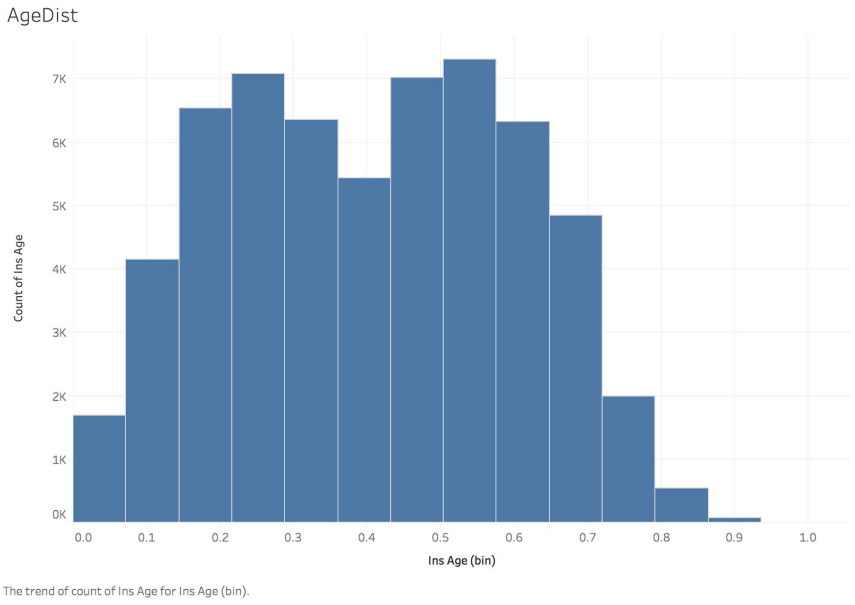

Fig. 2, Age Distribution

The next graph, Fig. 3, shows a slightly skewed normal distribution of the BMI index of LI applicants, speaking to the fact that weight, which is a factor in calculating BMI, is usually skewed positively.

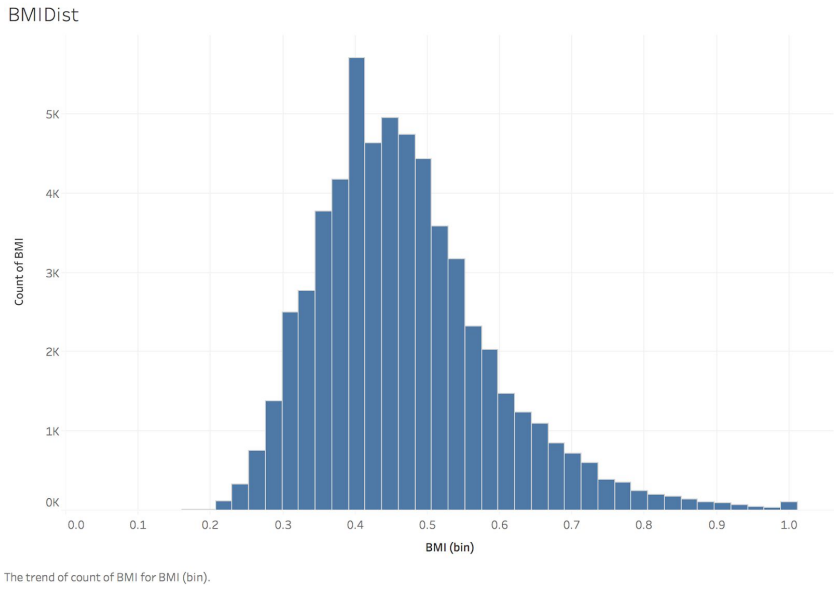

Fig. 3, BMI Distribution

In Fig. 4, we see an expected narrow distribution of height.

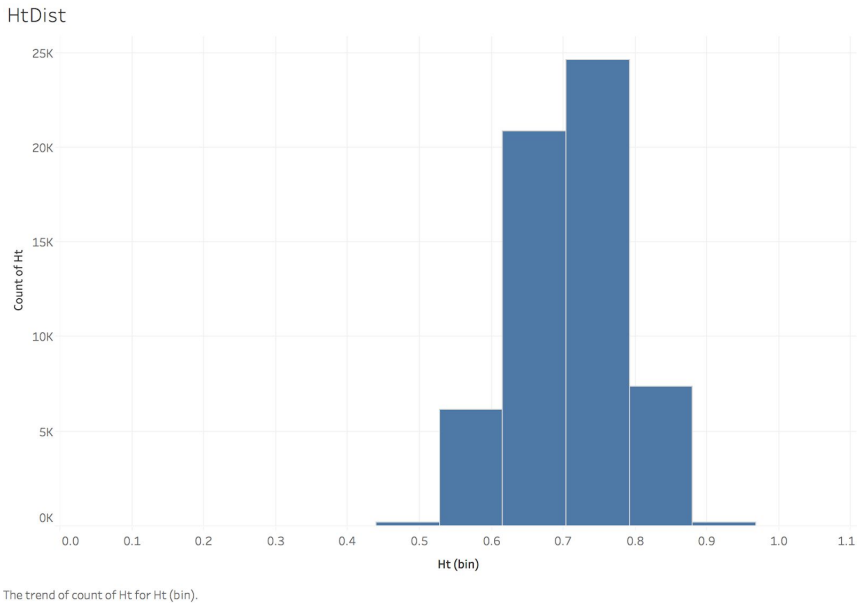

Fig. 4, Height Distribution

And lastly, in Fig. 5, we see that weight is positively skewed.

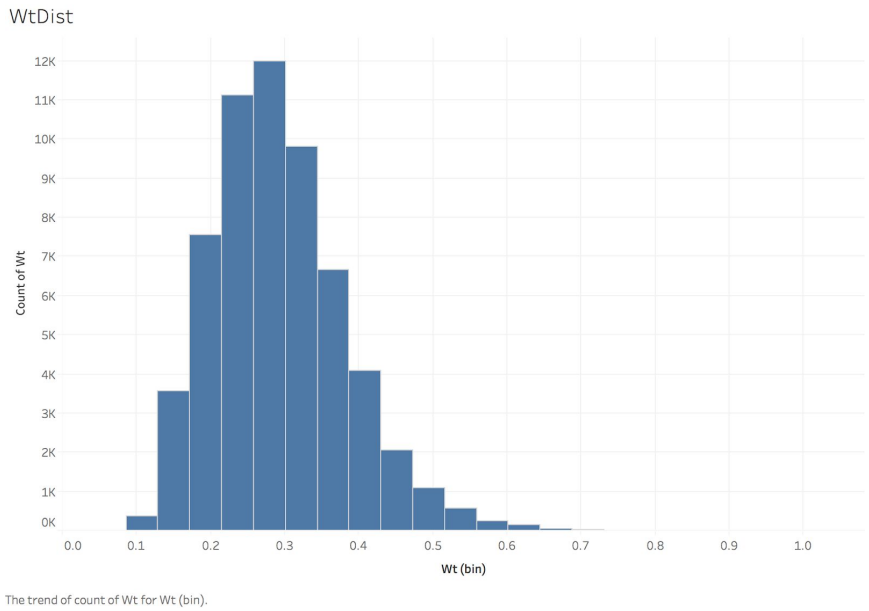

Fig. 5, Weight Distribution

Lastly, in Fig. 6, we see an overrepresentation of classes 5 and 7, with an underrepresentation of 2 and 4. After seeing the distributions of the other variables, it's unexpected to see risk distributed this way.

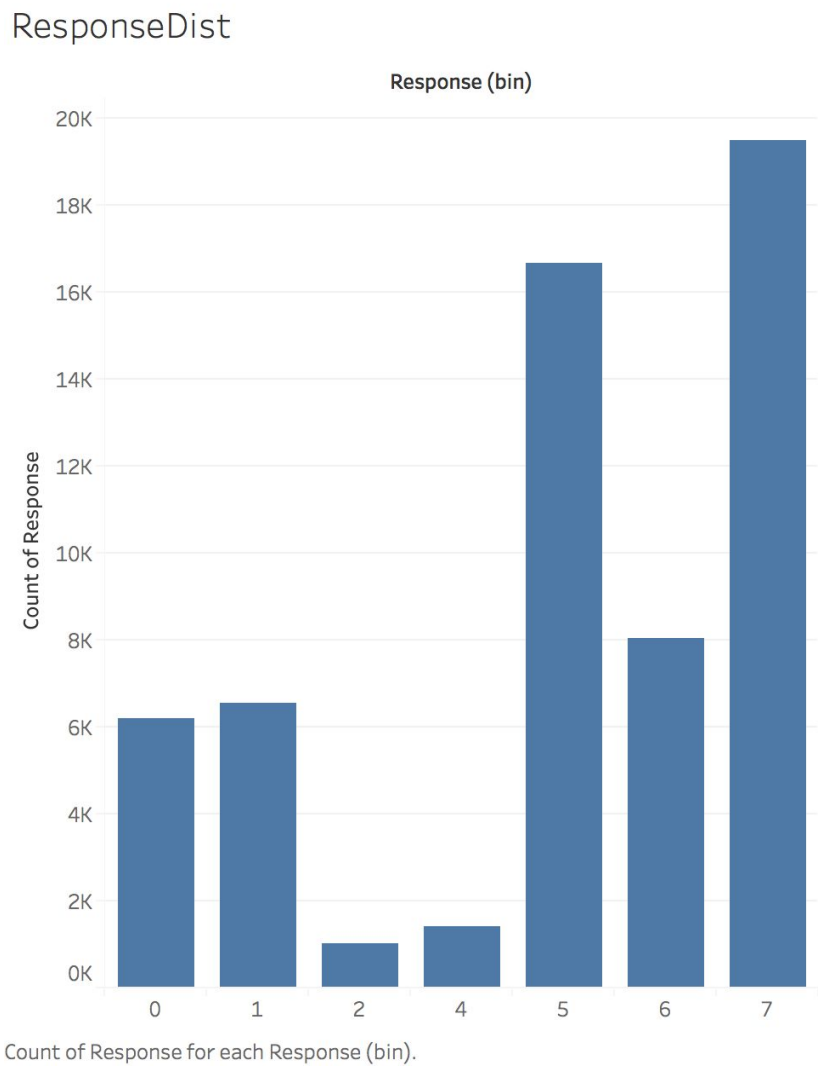

Fig. 6, Response Distribution

C. *Data pre-processing*

1) *Dealing with missing data:* The first step I took for pre-processing was to deal with the missing values in the data. The Prudential dataset has five attributes that have a large amount of missing data (over 30%), which were then dropped completely. The rest of the seven columns have less than 30% of their data missing, which make them ideal for imputation methods, but first, these attributes need to be analyzed to determine which imputation method is appropriate. Usually, there are three mechanisms of missing data, which are Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR) [10].

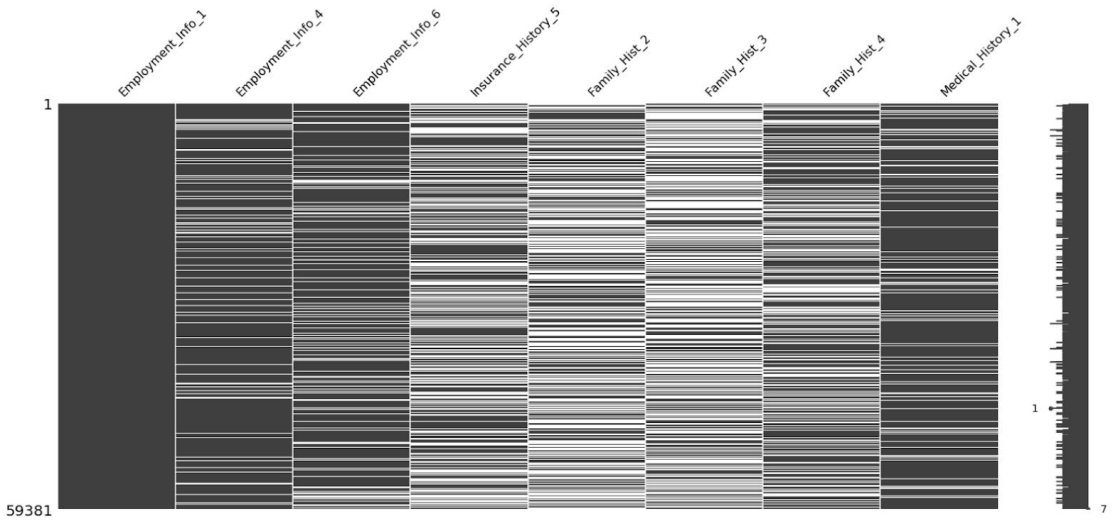

Fig. 7, Distribution of Missing Data

After plotting the attributes in Fig. 7, it is clear that there is no pattern of missing values, so they will be treated as MAR. I decided to use median as the imputation method to account for the skewed distributions in the data.

2) *Transforming the data*: Many machine learning libraries require that class labels be encoded as integer values, so it is considered good practice to convert class labels into integer arrays to avoid trouble down the road. By using the get_dummies feature from pandas, I transformed the categorical feature, Product_Info_2, to numeric so it can be better analyzed by a variety of machine learning algorithms. This method turned the Product_Info_2 feature into 19 features.

3) *Partitioning and scaling the data*: Since most machine learning algorithms run best on standardized data sets, aside from random forests and decision trees, I partitioned the dataset into training and test data (80:20 split) and followed that up by standardizing the features [11]. This is because the majority of machine learning algorithms perform better when features are on the same scale.

4) *Dimensionality reduction*: Dimensionality reduction is the process of reducing the number of variables to be used for more efficient modeling. There are two main methods for achieving this, feature selection and feature extraction. Feature selection involves selecting a subset of the prominent variables from the data, where feature extraction is used to transform high dimensional data onto fewer dimensions while also eliminating redundant and irrelevant features. These

methods are used to train machine learning algorithms faster and attempt to increase model accuracy by reducing overfitting [12].

By using a random forest as a feature selector, we can measure feature importance as the averaged impurity decreases from all decision trees in the forest without making assumptions about linear separability [11]. This method produces a plot that ranks the different features in the dataset by their relative importance (where all the features sum up to 1.0).

Principal components analysis (PCA) is an unsupervised linear feature extraction technique that is used across multiple fields, but is mostly used for dimensionality reduction. PCA aims at reducing the size of the data by extracting features that have the most information (read: variance). In other words, PCA helps identify patterns in the data based on the correlations between the features.

D. Modeling

Based on the nature of the data, I decided to use supervised machine learning algorithms to predict multi-class responses using the labeled training data. The algorithms of choice are Logistic Regression, Random Forest, and Multilayer Perceptron.

1) *Logistic Regression*: Logistic Regression is a classification model that is very simple to implement, but as a caveat, it performs well on linearly separable classes. Because it is one of the most widely used algorithms for classification in the industry, I chose to use Logistic Regression as the initial base classifier because of its robustness with machine learning problems, and the ability to easily handle multi-class classification.
2) *Random Forests*: Random Forests is typically regarded as the holy grail, out-of-the-box, classifier for machine learning problems. It has gained this popularity because of its good classification performance, scalability, and again, ease of use. As you read before, this algorithm can be used for feature selection and classification. As an ensemble classifier, Random Forests is able to combine weak learners to build a more robust model, or strong learners, which lead to better generalization error and less likely to be prone to overfitting.
3) *Multilayer Perceptron:* The simplified concept behind an artificial neural network was built upon hypotheses and models of how the human brain functions to solve complex tasks. In general, a multilayer perceptron is comprised of three layers: one input layer, a

hidden layer, and an output layer. The units in the hidden layer are fully connected to the input layer, and the output layer is fully connected to the input layer [11]. These layers harness the power of feedforward propagation to minimize the cost function, which make this algorithm ideal for really complex data like LI risk classification.

E. Experiments and results

1) *Evaluation*: After much experimentation, Fig. 8 shows the best combinations of methods and techniques for risk classification. While these scores aren't the best, they're a good starting point towards building on the analysis. I'll explain how I got to these iterations below and what I learned along the way.

Classifier	**Accuracy** (training, testing)	**Cohen Kappa Score**	**Precision**	**Recall**	**F1**
Logistic Regression via PCA	0.449, 0.451	0.267	0.398	0.446	0.397
Random Forests via Random Forests	0.999, 0.534	0.393	0.508	0.534	0.496
Multilayer Perceptron via Random Forests	0.480,0.479	0.313	0.421	0.479	0.421

Fig.8, Final Classification Metrics Evaluation

2) *Initial baseline accuracy*: When I started modeling, I wanted to understand how each classifier performed on the data out of-the-box to get baseline scores to tell if I am improving the model. Logistic Regression was initialized using solver = 'lbfgs', and multi_class = 'multinomial'. Random Forest was set with criterion = 'entropy', and n_estimators = '1000'. Lastly, the Multilayer Perceptron Classifier was set using activation = 'relu', solver = 'adam', alpha = 0.0001, and hidden_layer_sizes = (5,2).

Classifier	**Accuracy** (training, testing)
Logistic Regression	0.501, 0.497
Random Forest	1.0, 0.538
Multilayer Perceptron	0.453, 0.444

Fig. 9, Initial Baseline Classification Accuracy

3) *Dimensionality reduction*: After using Random Forests as a classifier, I simply reused it to perform feature selection. This involved mapping the feature importances to a graph in order to gain insight into which features were most important for analysis.

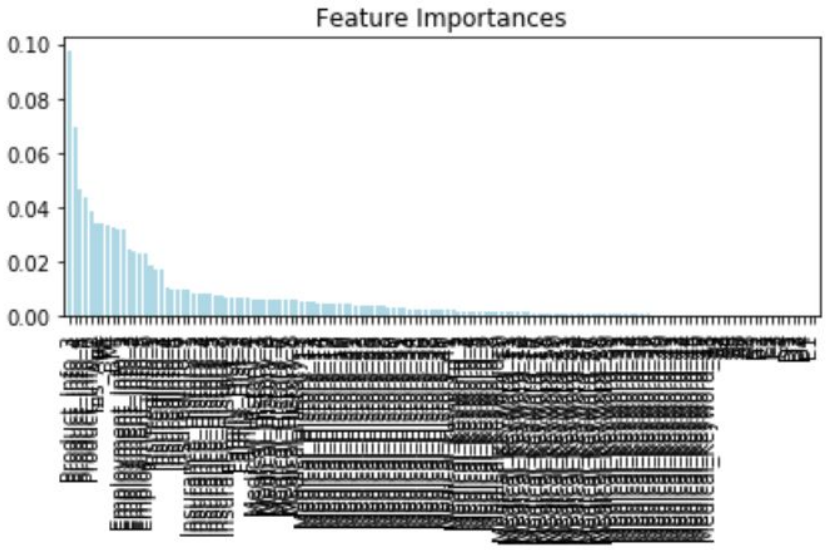

Fig. 11, Feature Importances

After looking at the feature importances graph, Fig. 11, it's clear that most of the features don't provide much value as predictors. By using a feature list index, I was able to see what the top features were. To my surprise, the features containing medical data were ranked after the 17th feature, meaning the more important features were actually not

related to medical data, which provides evidence that there are more important features that should be considered before asking for a medical exam (i.e., Product_Info 5, Ht, Wt, BMI, Employment_Info_1-3,5, InsuredInfo_1, 4, 7, Insurance_History_5,7,9, and lastly, Family_Hist_2,3, and 4). I then used the SelectFromModel library to feed the output from initial Random Forest to train the classifiers again, using only a subset of the features by setting the threshold of SelectFromModel equal to 0.008 (i.e. 27 features).

4) *Principal Component Analysis (PCA):* To counter the Random Forest feature selection method, I decided to use 27 features via PCA.

Classifier	**PCA** (training accuracy, testing accuracy)	**Random Forest** (training accuracy, testing accuracy)
Logistic Regression	0.449, 0.451	0.466, 0.190
Random Forest	1.0, 0.451	0.999, 0.534
Multilayer Perceptron	0.439, 0.442	0.480, 0.479

Fig. 12, Classification Metrics using Dimensionality Reduction Techniques

Compared to the initial baseline classifiers from Fig. 9, before dimensionality reduction techniques, Logistic Regression performs the best without either dimensionality reduction technique, but performs slightly lower than the baseline by using PCA. Random Forests performs the best by using Random Forests as a feature selector. The Multilayer Perceptron model also performed better by using Random Forests as a feature selector. While these changes in accuracy aren't substantial, they are impressive, given that we reduce the dataset from 139 to 27 features. In order to move forward, I decided to take a look at the learning and validation curves to diagnose the data, since accuracy is still pretty low (see Fig. 12).

5) *Debugging using learning and validation curves:* To figure out how to proceed with analysis, I used a Linear Regression model to plot the learning and validation curves to

get an idea of what's going on, i.e. if an algorithm is overfitting (high variance) or underfitting (high bias).

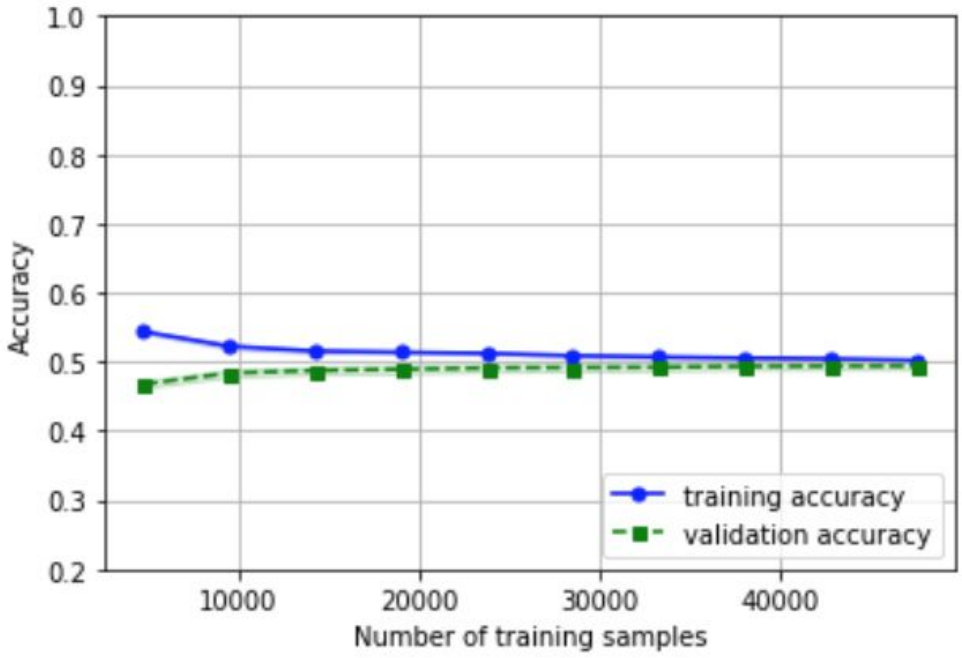

Fig. 13, Learning Curve

According to Raschka, the Linear Regression model is experiencing high bias in Fig. 13, which means the model has both low training and cross-validation accuracy, which indicates that it is underfitting the training data [11]. Common ways to address this are to increase the number of parameters of the model, either through collecting or constructing additional features. This is validated since the Linear Regression model performed better without dimensionality reduction, or by decreasing the degree of regularization. To address this, I re-ran the learning curve using a lower regularization.

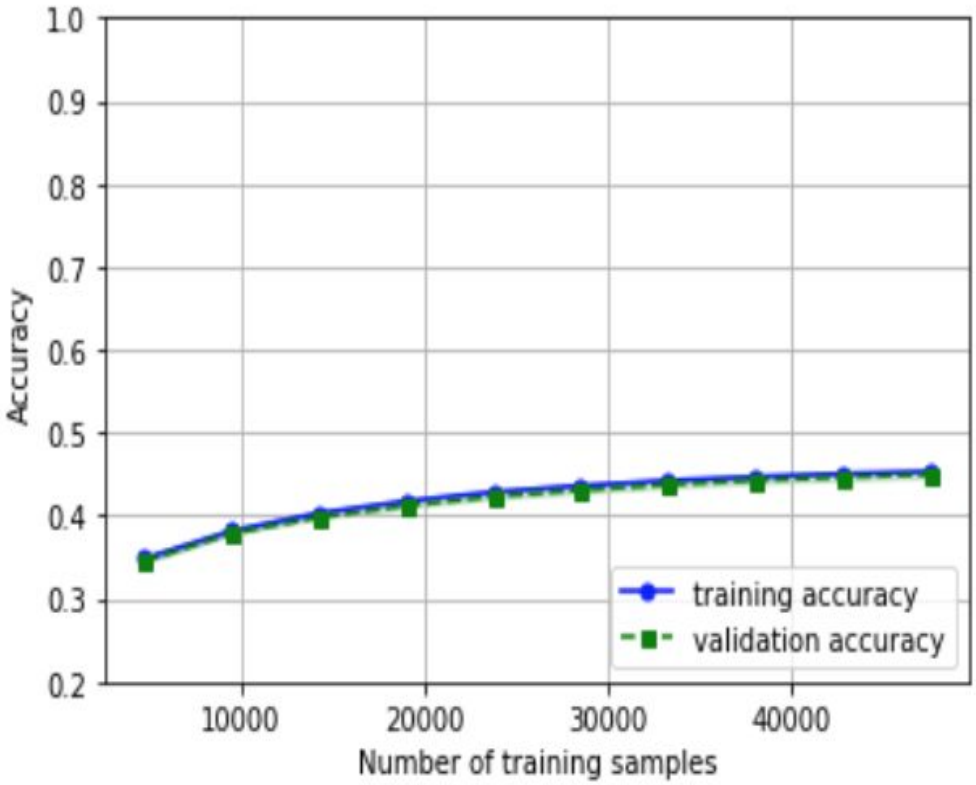

Fig. 14, Learning Curve with Low Regularization

Since I used low regularization for this model (Fig. 14), with the lower accuracy output, I can only assume that the next best thing to do in this scenario is to add more features (although this is out of scope for this project and my current skill set). Since I can't add more training data to decrease the chance of overfitting, I can attempt to reduce the noise of the dataset.

Next, I used the validation curve seen in Fig. 15 to address the modeling issues.

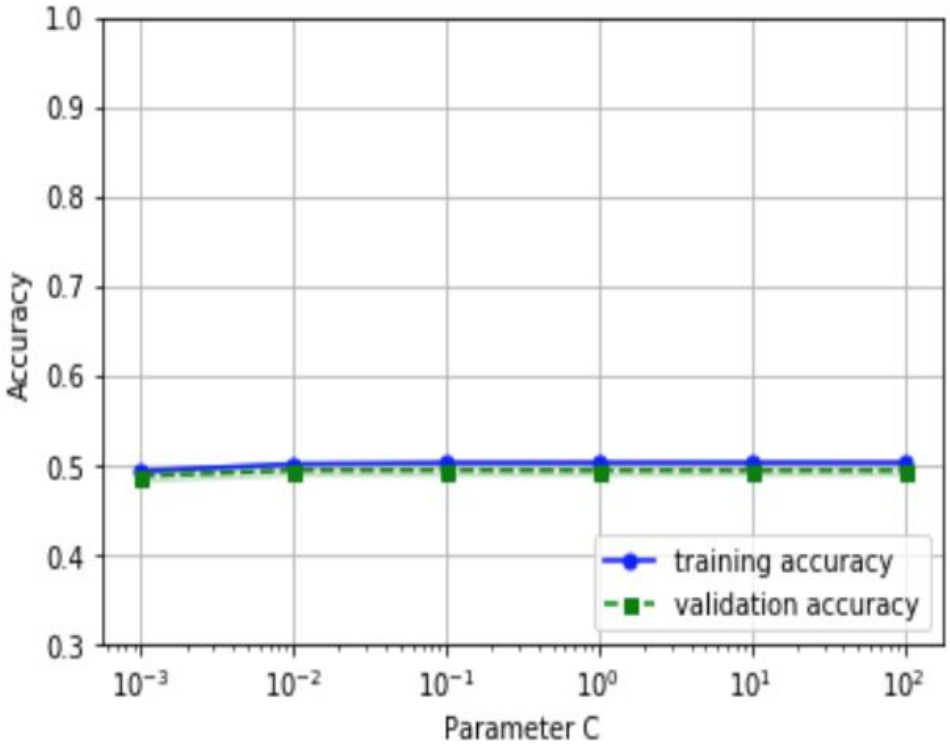

Fig. 15, Validation Curve

This curve tells us that the model doesn't really overfit or underfit the data. The only concern is that accuracy is still low, so at this point I decided to look at different performance evaluation metrics to evaluate the models.

6) *Confusion matrix and Cohen Kappa Score:* Once I changed the evaluation metrics, I was able to see how each model predicts each class (8 total). This moment was important because I realized a factor that has been affecting the models. When looking at each confusion matrix, I realized that classes 5 and 8 were extremely over represented in predictions, which makes sense because the distributions of the classes are heaviest on class 5 and 8 (majority classes), with the rest underrepresented (minority classes). This leads us to another concept called the "Accuracy Paradox", which explains how accuracy can be misleading when dealing with imbalanced classification problems. In the next section, I'll go over how to combat imbalanced data.

Fig. 16 is an example of one of the confusion matrices using Random Forests with Random Forests as a feature selector.

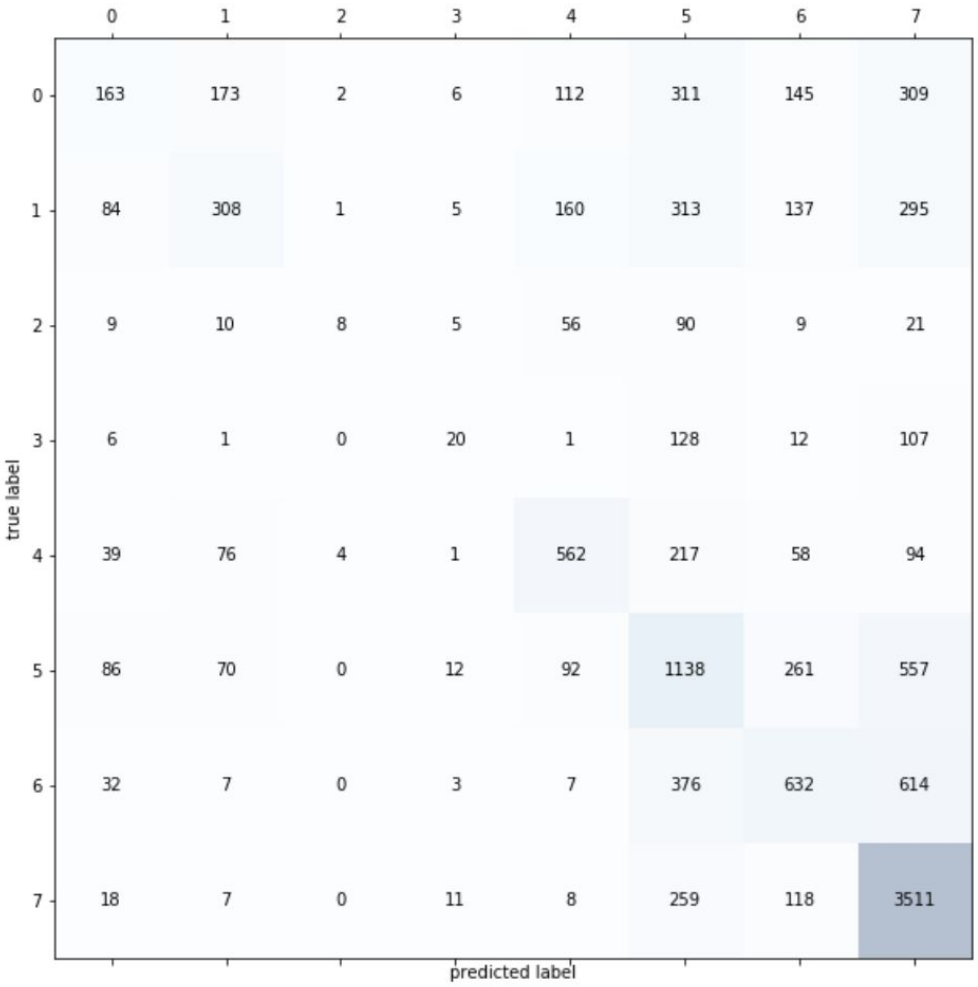

Fig. 16, Confusion Matrix

After applying all the new evaluation metrics in Fig. 17, the Random Forests with Random Forests feature selection performed the best, followed by the Multilayer Perceptron with Random Forests, with Logistic Regression via PCA as last. One explanation for Random Forests performing the best is that it performs well on imbalanced datasets. The splitting rules that look at the class variables can force all classes to be addressed.

Classifiers	**Cohen Kappa Score**	**Precision**	**Recall**	**F1**
Logistic Regression via PCA	0.267	0.398	0.446	0.397
Random Forests via Random Forests	0.393	0.508	0.534	0.496
Multilayer Perceptron using Random Forests	0.313	0.421	0.479	0.421

Fig. 17, New Evaluation Metrics

7) *Resampling methods:* Since we found out earlier that our dataset is highly imbalanced, I took steps to address this via various re-sampling methods, Random Under Sampler (RUS), Balanced Bagging Classifier (BBC), and Random Over Sampler (ROS). By using the Random Forests classifier with Random Forests feature selection as the base classifier. RUS attempts to balance class distribution by randomly eliminating majority class examples, whereas ROS increases the number of instances in the minority class by randomly replicating them to increase the representation of the minority class. The Balanced Bagging Classifier chains a RUS and a given classifier, while the Bagging Classifier is using the imbalanced data directly. It helps the ensemble avoid focusing on the majority class.

Classifier	**Cohen Kappa Score**	**Precision**	**Recall**	**F1**
Random Forests w/ RUS	0.318	0.479	0.438	0.436
Random Forests w/ BBC	0.342	0.499	0.456	0.459
Random Forests w/ROS	0.393	0.497	0.528	0.500

Fig. 18, Evaluation of Re-sampling Methods

After a few experiments seen in Fig. 18, the clear winner of the re-sampling methods was ROS. This is understandable because Random Forests typically focus on the majority classes for classification. Now, I'll apply ROS to the most successful models from the previous section.

Classifier	**Cohen Kappa Score**	**Precision**	**Recall**	**F1**
Logistic Regression using ROS and PCA	0.251	0.421	0.378	0.386
Random Forests using ROS and Random Forests	0.393	0.497	0.528	0.500
Multilayer Perceptron using ROS and Random Forests	0.268	0.422	0.394	0.386

Fig. 19, Evaluation of ROS Performance Against Classifiers

After comparing the results from the previous section, it's clear that this resampling method only increased the scores for the Random Forests classifier, marginally. For the other classifiers, it looks like the only metric that increased was Precision, while Recall and F1 decreased. What this tells me is that Logistic Regression and the Multilayer Perceptron do an adequate job of predicting from both majority and minority classes. Since I didn't try every re-sampling technique with Logistic Regression or Multilayer Perceptron, I'd like to explore that later in the future by applying each technique to separate instances of the minority and majority classes. Ensemble methods would also be an alternative choice to handle these issues, since they modify the existing classification algorithms to make them appropriate for imbalanced data.

V. Conclusion

Risk classification for LI is a complex problem. Having gone through the process, I can say that this dataset requires a lot of pre-processing to get it to perform better when being modeled. The main issue that was blocking my progress was dealing with imbalanced data. If you look back to the distribution of the response, or risk classes, you'll see a huge over representation of two classes, with low representation of the other classes. For future research, I'd recommend attempting to improve the predictions by exploring different ensemble methods for classification, since they optimize towards imbalanced data. A more advanced approach to feature engineering would also help identify which features are relevant and which are noise. This ties back to my objectives as I intended to explore the accuracy of classification methods. While I marginally improved accuracy, I was able to increase other metrics like the Cohen Kappa Score, Precision, Recall, and F1. As I said earlier, the classifiers with the best scores can be fed into ensemble methods, like Majority Voting, to improve the overall scores. My second objective was to discover which features contributed the most to classification. By using Random Forests as a feature selector, I was able to identify the top features. To my surprise, given my research focus, medical data was not at the top of the list. This reinforces the idea that companies should move away from expensive medical exams which only lengthen the process. Instead, companies can increase their focus on other attributes like applicant height and weight, employment info, insured info, insurance history, and family history (all of which can be determined at the time of application). Lastly, by using various dimensionality reduction techniques, we can see that we can improve accuracy and other

classification metrics by using less features (from 139 to 27 features). This is important because by using less data to get the same scores, if not better, we can attempt to drastically improve the user experience of applying for LI by asking far less about the applicants than what the industry does currently (i.e. medical data).

VI. References

[1] Boodhun, N., & Jayabalan, M. (2018). Risk prediction in life insurance industry using supervised learning algorithms. Complex & Intelligent Systems, 1-10.

[2] Bell M (2016) Is analytics the underwriting we know? [Online]. https://insurance-journal.ca/article/is-analytics-changing-the-underwriting-we-know/

[3] Harris, T., & Yelowitz, A. (2014). Is there adverse selection in the life insurance market? Evidence from a representative sample of purchasers. Economics Letters, 124(3), 520-522.

[4] Bhalla A (2012) Enhancement in predictive model for insurance underwriting. Int J Comput Sci Eng Technol 3:160–165

[5] Umamaheswari K, Janakiraman D (2014) Role of data mining in Insurance Industry. Int J Adv Comput Technol 3:961–966

[6] Wuppermann A (2016) Private information in life insurance, annuity and health insurance markets. Scand J Econ 119:1–45

[7] Prince, A. E. (2016). Tantamount to Fraud: Exploring Non-Disclosure of Genetic Information in Life Insurance Applications as Grounds for Policy Rescission. Health Matrix, 26, 255.

[8] Huang, Y., Kamiya, S., & Schmit, J. T. (2016). A Model of Underwriting and Post-Loss Test without Commitment in Competitive Insurance Market.

[9] Carson J, Ellis CM, Hoyt RE, Ostaszewski K (2017) Sunk costs and screening: two-part tariffs in life insurance. SSRN Electron J 1–26

[10] Nicholson, J. S., Deboeck, P. R., & Howard, W. (2017). Attrition in developmental psychology: A review of modern missing data reporting and practices. International Journal of Behavioral Development, 41(1), 143-153.

[11] Raschka, S. (2015). Python machine learning. Packt Publishing Ltd.

[12] Hoque, N., Singh, M., & Bhattacharyya, D. K. (2018). EFS-MI: an ensemble feature selection method for classification. Complex & Intelligent Systems, 4(2), 105-118.