Life Insurance Data Analysis

## Objective

Analysis goal is to identify features that influence risk score associated to an insurance application and develop an automated approach to generate risk scores given a set of insurance/applicant features.

## Background

In a one-click shopping world with on-demand everything, life insurance application process is antiquated. Customers provide extensive information to identify risk classification and eligibility, including scheduling medical exams, a process that takes an average of 30 days. Due to this, most people often only buy life insurance coverage through employment-sponsored products and never purchase externally. This user experience is in sharp contrast of purchasing other insurance products externally where the process is much smooth. Hence, only 40% of U.S. households own individual life insurance. Ignoring external factors that influence processing timelines of an insurance application, we choose to focus on the data collected by applicants and try to identify factors that influence the risk score given to applicants. The reasoning is that by identifying factors, we can simplify the application process. Next, insurance companies can build automated models that could help make quicker decisions on applications making the overall process quicker and less labor intensive.

## Method

Dataset is provided by Prudential Life Insurance as part of a hiring challenge hosted on Kaggle. Dataset consists of 59, 381 observations of 128 attributes extracted from the life insurance application process. A closer inspection of the dataset shows that most parameter names are grouped under categories and anonymized. This is done to protect applicant privacy and comply with Prudential’s internal security requirements. User inputs from application can be divided into the following categories:

* Product Information
* Age of the applicant
* Height
* Weight
* BMI
* Employment Information
* Insured Information
* Insurance History
* Family History
* Medical History
* Medical Keyword

During EDA, dataset was grouped into different variable types (continuous, categorical) and distribution of variables with risk score (response) generated. Additionally, dataset was verified for missing values, duplicated rows. From the analysis, missing data was found to be at 5%. Certain physical attributes (such as Age, Ht, Wt and BMI) have missing data and zero values. Considering that the dataset deals with applicant information related to life insurance process, zero values for physical attributes was found to be surprising and may highlight gaps or issue with data collection process at Prudential. For data modeling, missing values were imputed by elimination, as the missing percentage was relatively low. Finally, assuming that text variables are of categorical type we replace them with numeric id’s.

To identify top features that influence applicant risk scores and help generate future scores, we decide to build predictive models. For our predictive model we decide to use caret package available in R. Caret (Classification and Regression Training) package provides a toolkit for building classification and regression models. The big advantage with caret package is the ease of use due to the standardization of how people build models and view results. For our analysis, we start by building a XGBoost model. Extreme Gradient Boosting (XGBoost) is an implementation of gradient boosting machines available for data modeling through Caret package. The algorithm is designed to be efficient in computational and memory requirements with support for parallel tree construction, automatic handling of missing data (sparse aware), distributed computing. As part of data preprocessing before fitting the model, the training data was divided into 75:25 ratio using createDataPartition method with 25 percent reserved for validating model performance. We use the trainControl method in caret package to perform 5 fold cross validation. The following tuning parameters are used for data modeling:

* gamma (Minimum Loss Reduction) - Is the minimum loss reduction required to make a further partition on a leaf node of the tree. The larger value will create more conservative model.
* min\_child\_weight (Minimum Sum of Instance Weight) - You can try to begin with thinking of min bucket size in decision tree( rpart).It is like number of observations a terminal node.If the tree partition step results in a leaf node with the sum of instance.
* colsample\_bytree (Subsample Ratio of Columns) – Helps to randomly choose the number of columns out of all columns or variables at a time while tree building process.
* nrounds (no. of boosting Iterations) – Represents the number of iterations the model runs before stopping. With higher value of nrounds model will take more time and vice-versa. Eta (Shrinkage) – Represents step size shrinkage, which actually shrinks feature weights. With high value of eta, model will run fast and vice versa. With higher eta and lesser nrounds, model will take lesser time to run. With lower eta and higher nrounds model will take more time.
* max\_depth (Maximum Tree Depth) - Higher value of max\_depth will create more deeper trees and more complex model. Higher value of max\_depth may create overfitting and lower value of max\_depth may create underfitting.

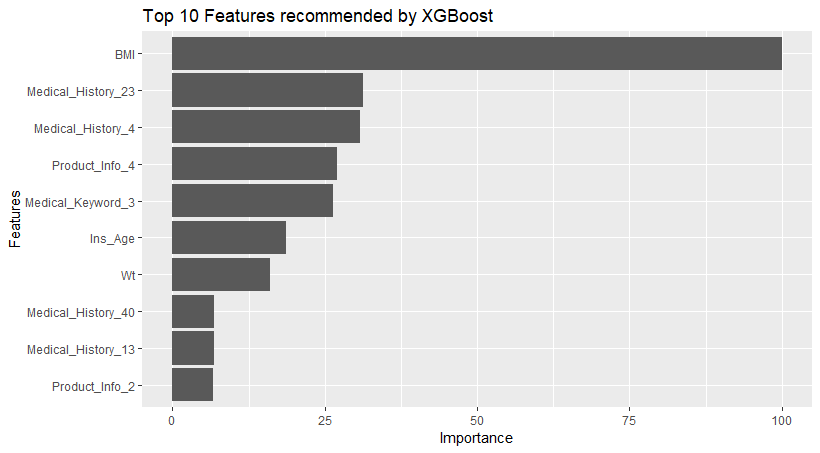
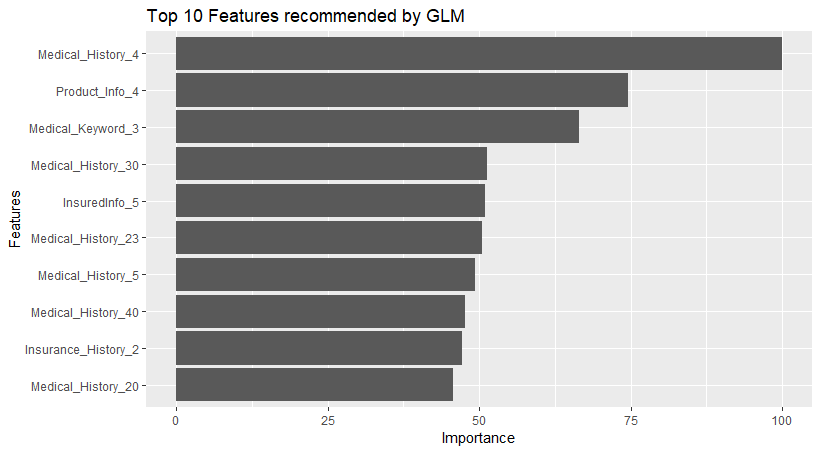
For our model development, we wanted to develop a decent sized model while trying to avoid the curse of overfitting. Hence, the following values for tuning parameters was chosen:

eta = 0.1, colsample\_bytree = c (0.5,0.7), max\_depth = c(3, 6), nrounds = 100, gamma = 1, min\_child\_weight = 2. After model development, importance matrix of top feature was generated.

Next, to compare the results of XGBoost model, a simple predictive model was fit using generalized linear model (GLM) was fit using caret package. The cross validation technique used for XGBoost model was reused for the GLM technique. No other tuning parameters are necessary to build the model. Once the model was trained, top important features were identified. Finally, predictions followed by confusion matrix was generated for XGBoost and GLM model to calculate model accuracy.

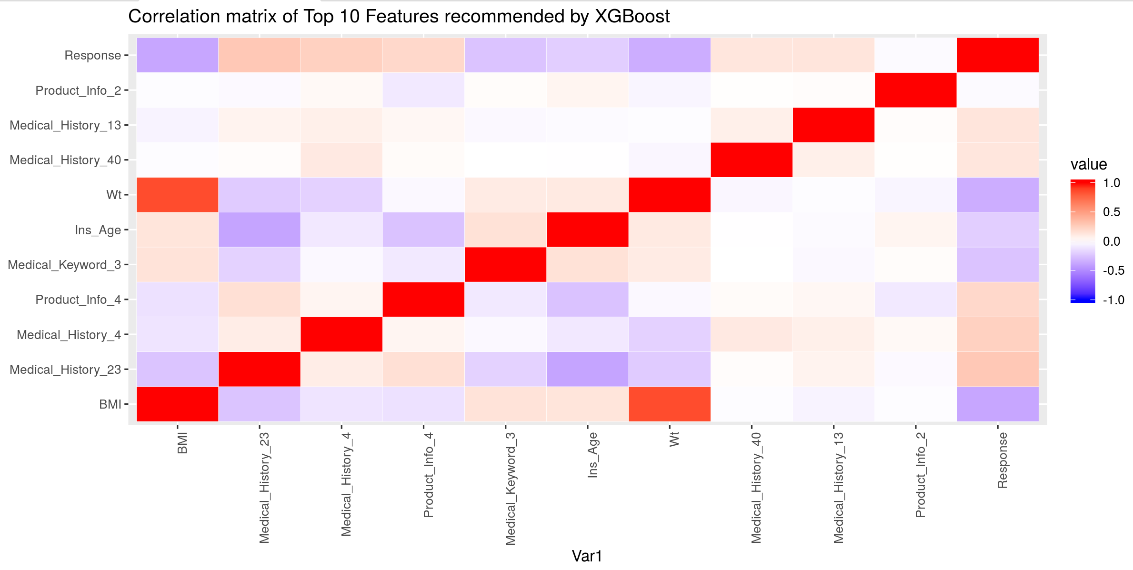
## Results

Once predictive models were fit on the training data, we identify the top 10 features recommended by both models. Looking at the results, the recommendation that features such as BMI, applicant age, medical history have high influence on risk score of life insurance applicant make sense.

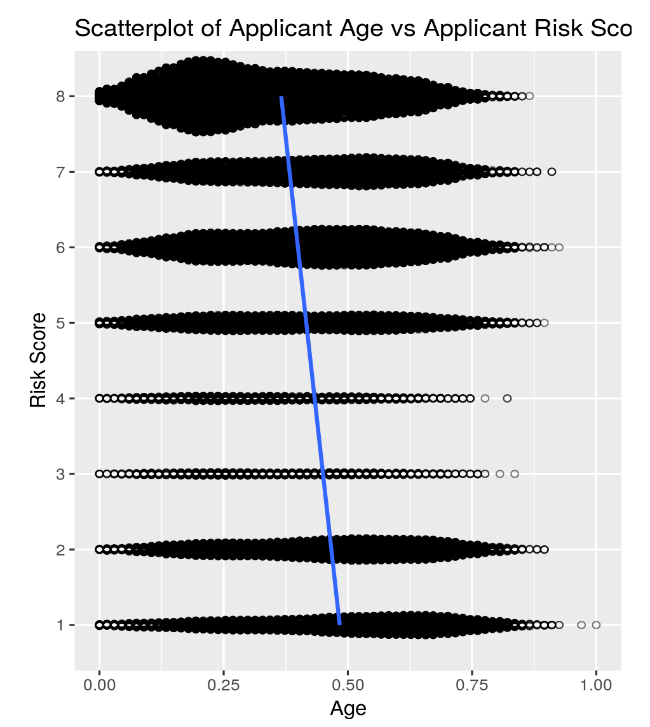
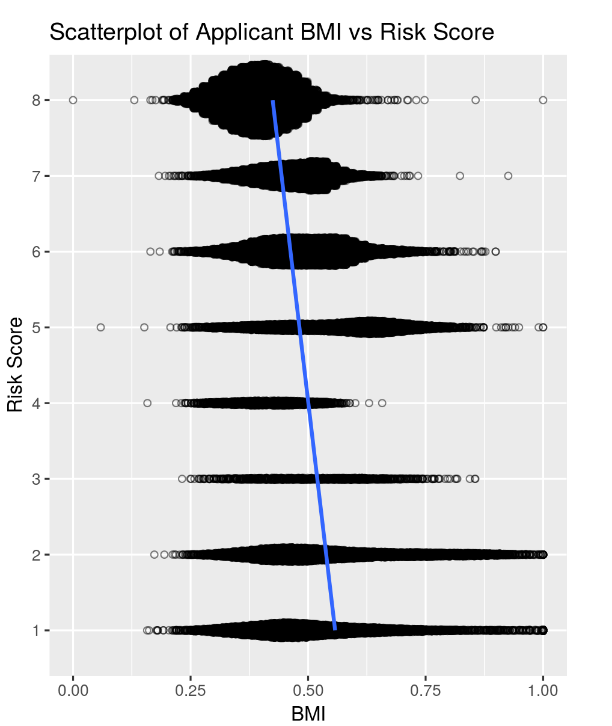
Picture 1: Plot of top 10 recommended features by XGBoost and GLM Models

Generating correlation plots to visualize the relationship between the top 10 features and risk score we see few interesting relationships. Physical attributes such as BMI and Age have a negative correlation with respect to the risk score. This finding also makes sense in that applicants with higher Body Mass Index (BMI) or older age group are categorized as higher risk for life insurance purposes.



Picture 2: Correlation Plot of features recommended by XGBoost

Comparing the results from the two models, we see lot of same features recommended in the top 10 list. However, when inspected closely the top 10-feature list of the GLM model reveals an interesting observation. The top 10 recommended features do not consist of a single continuous variable, which is surprising because from a life insurance perspective applicant health and physical attributes should carry a lot of weight in determining the risk score. To identify the reason behind this we decide to look at the distribution of these variables with respect to risk score.



Picture 3: Scatterplot of BMI and Age vs Risk Score

Looking at the above scatter plots of BMI with respect to different risk scores we see that both these parameters exhibit a non-linear distribution wrt response. Without any variable transformations, modeling such relationship in a simple model like GLM is challenging. Another important aspect to discuss here is that when developing both models, the response variable risk score is modeled as a continuous variable type with root mean square error rate as the evaluation metric. This assumption raise some concern especially since risk score is of nominal type. However, statistically speaking there is no best method and there are plenty of examples where models are developed where nominal response type are treated as continuous type and continuous type response are discretized for model fitting purpose. Due to constraints on time and resources, we could not contrast the above two models with a predictive model of classification type and would love to do this in the future.

## Conclusion

From the analysis and model development, we were able to identify top features that influence applicant risk score. Using both models, we generated predictions for our test data and then created a confusion matrix to compare the predicted and actual values. Ignoring the results of predictions for a second and just looking at the top 10 features recommended by both models one can see that the XGBoost model is a better fit for the data and is better at identifying features that influence risk score. In the previous section, we saw that the accuracy of GLM model was compromised by the non-linear relationship of few features with response confirming that XGBoost model is better. Looking at the predictions, it makes sense that applicant physical attributes and medical history has greater influence on risk score compared to parameters such as employment history/status.

## Application

When it comes to the actual application of the model in real world, there are pros and cons with having an automated model making decisions. Looking at the positives, an automated model will help life insurance application process rate and help cut down on the significant delay faced by customers. By identification of important features from the application, insurance companies such as Prudential can trim their application to only ask questions of importance upfront instead of the current longer application questionnaire. However, there are concerns with respect to generalizability of this model. Both predictive models were developed using dataset of approx. 60,000 observations. This may sound significant but in the real world, insurance companies deal with more life insurance application than the above number. Without information on how the data was sourced and whether all class of applicants truly reflected in an unbiased manner, it’s hard to conclude that predictability of a model. Next important factor to consider is the impact of predictions in the real world. Considering that the model is generating risk scores for applicants that is used to make decisions (yes/no) it is important to recognize that a denial on the application means that an individual may not be able to seek coverage for health which puts the said individual at risk exposure to financial burden short/long term. To account for such scenarios, the predictive model should have a feedback mechanism where impact of denial/approval is considered as an input parameter in addition to application features.

## References

[1] Kaggle. (2015, November 11). Prudential Life Insurance Assessment. Can you make buying life insurance easier? [Information on a page online]. Retrieved from <https://www.kaggle.com/c/prudential-life-insurance-assessment/>

[2] XGBoost. (2016). XGBoost R Tutorial [Documentation]. Retrieved from <https://xgboost.readthedocs.io/en/latest/R-package/xgboostPresentation.html>

[3] Kuhn, Max. (2018, May 26). The caret Package. Retrieved from https://topepo.github.io/caret/

[4] Gupta, Arpan. (2017, January 25). Model tuning in Crossvalidation using caret in R [Documentation]. Retrieved from https://analyticsdataexploration.com/xgboost-model-tuning-in-crossvalidation-using-caret-in-r/