Machine Learning
Life Insurance Application Risk Prediction

Submission Date: 21 May 2021

Word count: 1943 (excluding references)

ABSTRACT

A vital area of the life insurance business is classifying its applicants by assessing their risk. Referring to Legal & General's guideline as a reference [1], the underwriting process of accepting an insurance application accounts for all the information provided by the applicant. The decision of whether or not to accept the application is a result of the assessed risk based on a combination of factors sourced from the provided information. As the use of automated decision making models have become increasingly prevalent in the industry, the underwriting process has benefitted from the decrease in application processing times. Automated risk scoring is capable of saving time and decreasing subjectivity or unintentional biases, thus simplifying the task of manually processing applications [2]. This paper will explore useful machine learning algorithms that can be used to accurately predict an applicant's risk when applying for a life insurance policy. This paper will also briefly discuss the ethics surrounding automated decision making models.

1. INTRODUCTION

Life insurance or life assurance is a policy that pay out a cash sum upon the applicant's demise or diagnosis of a terminal illness. The need for life insurance is relevant to the applicant's personal circumstances – whether or not they have dependants that require support. A life insurance policy's premium and approval are dependant on the applicant's age, health, medical history and lifestyle.

In the past, a policy's approval is reliant on manual work where individual applications will be processed for approval. This manual approach is known as a scorecard [2] where attributes and rules are manually selected based on the personal experience of the analysts. This manual risk scoring extends the application process and this in turn, may have played a factor in the decreased uptake of life insurance over the years.

In the recent light of increased artificial intelligence implementation, insurance companies have taken on automated decision-making models that can speed up the application process. This automated risk scoring not only decreased processing times but also unintentional biases.

This paper explores the use of three machine learning algorithms: K-Nearest Neighbors, Logistic Regression and XGBoost models to classify risk levels of a set of life insurance applicants based on the criteria provided in the application.

However, the proposed models are what is considered to be black box models. These models limit the user's access to understanding how the results are obtained from the models. The ethical obligations surrounding the transparency of these black box will be briefly discussed in the conclusion.

The paper is organised as follows: Section 2 discusses the recent literature on life insurance and credit scoring. Section 3 explores the data and describes its analysis for the use of the machine learning algorithms and Sections 4 and 5 will discuss the findings and conclude the study.

2. LITERATURE REVIEW

Machine learning algorithms are becoming increasingly common across various industries, aside from the life insurance industry. Boodhun and Jayabalan (2018) [4] had studied suitable machine learning algorithms (ie. Multiple Linear Regression and Random Tree Classifiers) to predict risk level of life insurance applicants. They concluded that the use of data analytical solutions promotes faster and better results in comparison to the traditionally complex actuarial formulas. Customer satisfaction and loyalty increases when customers are able to experience faster, more accurate services.

Another study referred to for this paper is written by Bussmann et al. (2020) [5] that explored explainable artificial intelligence in the field of fintech risk management. Their study proposed the XGBoost model to predict peer to peer lending approval. Shapley values were then utilised as post-processing to explain the model's results. This was a step towards the growing need for transparency within the automated decision making process regardless of its industry. Their study looked to explore such an implementation in an industry aside from credit risk management.

3. METHODOLOGY

3.1. DATA

The dataset explored was provided by Prudential Life Insurance on Kaggle on their competition titled "Prudential Life Insurance Assessment" [3]. The dataset contains 59,381 entries with a total of 128 features including the "Id" and "Response" variables. This data was split into training, cross validation and testing data.

The table below contains a description of the features of the dataset:

Data.csv	The dataset containing the response values
Id	A unique identifier associated with an application
Product_Info_1-7	A set of normalised variables relating to the product applied for
Ins_Age	The normalised age of the applicant
Ht	The normalised height of the applicant
Wt	The normalised weight of the applicant
BMI	The normalised BMI of the applicant
Employment_Info_1-6	A set of normalised variables relating to the employment history of the applicant
InsuredInfo_1-7	A set of normalised variables providing information about the applicant
Insurance_History_1-9	A set of normalised variables relating to the insurance history of the applicant
Family_Hist_1-5	A set of normalised variables relating to the family history of the applicant
Medical_History_1-41	A set of normalised variables relating to the medical history of the applicant
Medical_Keyword_1-48	A set of dummy variables relating to the presence/absence of a medical keyword being associated with the application
Response	This is the target variables, an ordinal variable relating to the final decision associated with an application

Table 1. List of feature descriptions sourced from Prudential's Kaggle Competition

The aim of the paper is to build a predictive model that will accurately classify the risk of an applicant using an automated approach to allow the streamlining of the approval process. The end result is to predict the "Response" variable for each "Id" in the test set.

3.2. DATA PREPROCESSING

The dataset was checked for missing values by defining a function that sifted through each column in the dataset and calculated the percentage of missing data in each feature. There were 13 columns that contained some form of missing values.

	Total	Percentage
Medical_History_10	58824	99.060
Medical_History_32	58274	98.140
Medical_History_24	55580	93.600
Medical_History_15	44596	75.100
Family_Hist_5	41811	70.410
Family_Hist_3	34241	57.660
Family_Hist_2	28656	48.260
Insurance_History_5	25396	42.770
Family_Hist_4	19184	32.310
Employment_Info_6	10854	18.280
Medical_History_1	8889	14.970
Employment_Info_4	6779	11.420
Employment_Info_1	19	0.030

Figure 1. Summary table of missing values

Out of the 13 columns, the top nine had at least 30% of their data missing and were dropped from the dataset. The deleted features are as follows:

Medical_History_10	Medical_History_32
Medical_History_24	Medical_History_15
Family_Hist_5	Family_Hist_3
Family_Hist_2	Insurance_History_5
Family_Hist_4	

Table 2. List of features removed from the dataset

Three of the remaining four categories (Employment_Info_6, Employment_Info_4 and Employment_Info_1) were continuous and Med_History_1 was discrete. The Little's Test was performed using R Studio to classify whether they were one of the following three categories:

1. Missing Completely at Random (MCAR)
2. Missing at Random (MAR)
3. Missing Not at Random (MNAR)

The result is as below:

```
> # Continuous Variable
> mcar_test(train["Employment_Info_6"])
# A tibble: 1 x 4
  statistic    df p.value missing.patterns
      <dbl> <dbl>   <dbl>            <int>
1  3.97e-25     0       0                2
> mcar_test(train["Employment_Info_4"])
# A tibble: 1 x 4
  statistic    df p.value missing.patterns
      <dbl> <dbl>   <dbl>            <int>
1  1.11e-25     0       0                2
> mcar_test(train["Employment_Info_1"])
# A tibble: 1 x 4
  statistic    df p.value missing.patterns
      <dbl> <dbl>   <dbl>            <int>
1         0     0       1                2
>
> # Discrete Variable
> mcar_test(train["Medical_History_1"])
# A tibble: 1 x 4
  statistic    df p.value missing.patterns
      <dbl> <dbl>   <dbl>            <int>
1  8.45e-27     0       0                2
```

Figure 2. Little's Test results from R

As above, three of the variable's p-value was less than 0.05 which implies that the data is not MCAR so it can alternatively be MAR or MNAR. For the feature "Employment_Info_1" however, the p-value is 1 which is reason to believe that the feature may be MCAR. Thus, the analysis remains unbiased because the missingness of the data does not result in a bias in the parameters.

The missing data from the four features were imputed using the KNNImputer module from Sklearn. The resulting dataset contained no missing values.

3.3. EXPLORATORY DATA ANALYSIS

The Seaborn package was used to visualise the features of the dataset.

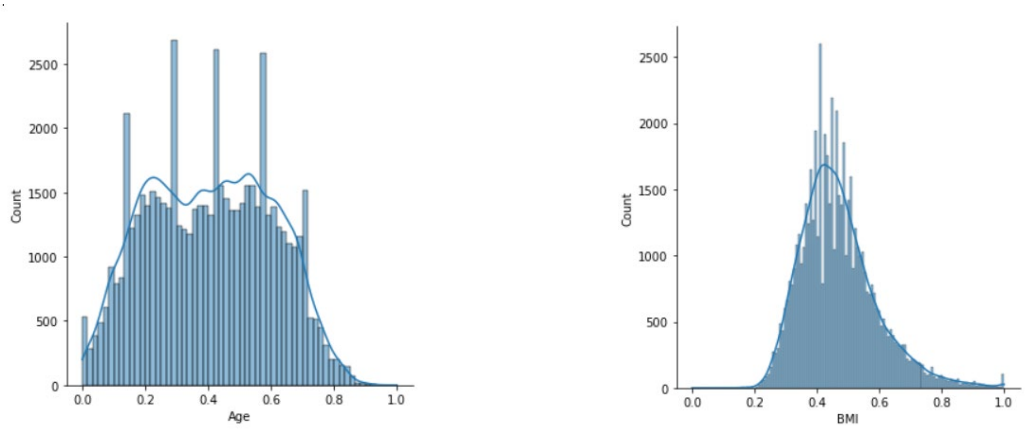

Figure 3. The distribution of the Age and BMI from the dataset

The BMI was skewed to the right though the Age has a broad spread throughout the middle. This was expected to be seen as people around the ages of late 20s to 50 are more likely to undertake a life insurance policy.

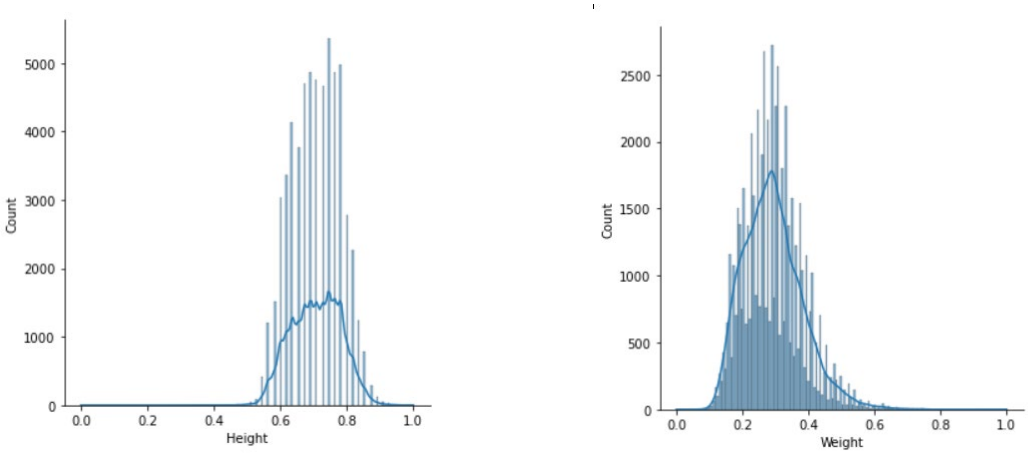

Figure 4. The distribution of the Heigh and Weight from the dataset

The height was seen to be skewed to the left and the weight was skewed to the right.
Some categorical variables like “Product_Info_6” have only two levels whereas some like “Employment_Info_2” have more than 10 levels (Figure 5). This indicated the potential need for dummy variables for their use in supervised learning algorithms. The get_dummies() function from the Pandas package was used to convert the categorical variables into indicator variables.

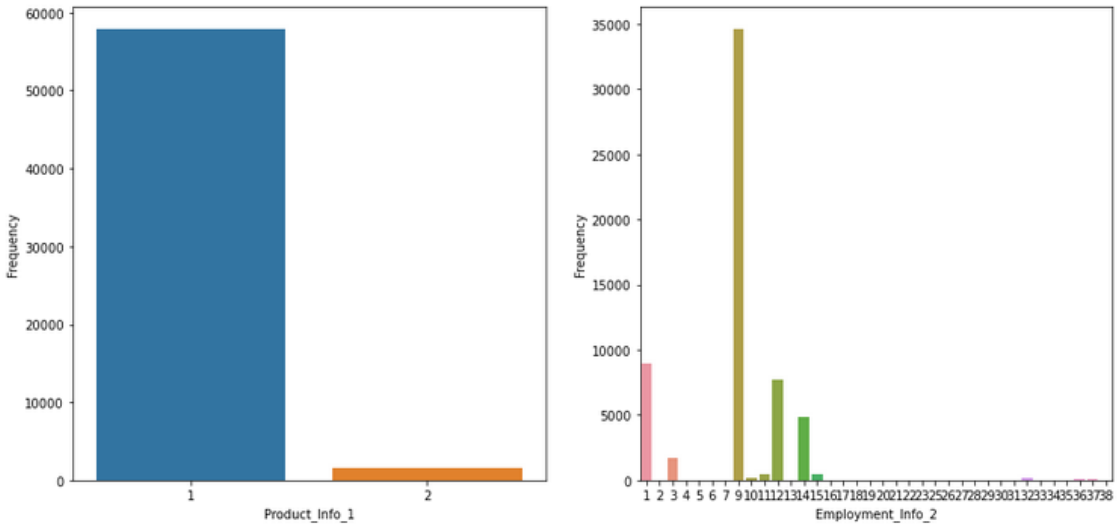

Figure 5. Sample of levels from two features in the dataset (Product_Info_1, Employment_Info_2)

Visualisation was carried out for the spread of risk classification across the “Response” variable, as shown in Figure 6. As can be seen below, there was a high percentage of Level 8 risk.

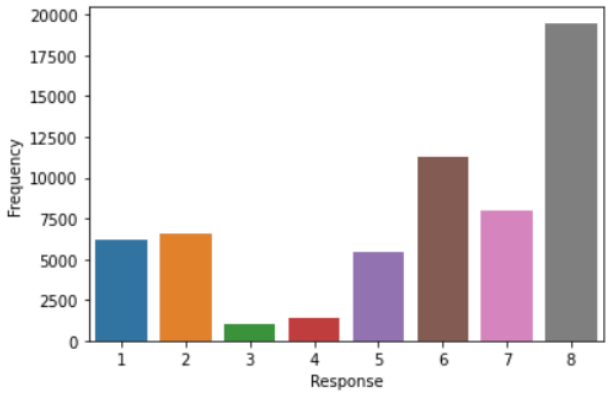

Figure 6. Spread of risk classification across the Response variable

The data was then normalised (excluding the “Id” and “Response” variables) to set the values to a common scale without distorting the differences in the range of values.

3.4. MACHINE LEARNING ALGORITHMS

3.4.1. K-NEAREST NEIGHBOR CLASSIFICATION

The model will assume that entries that are similar are in close proximity of others and thus belong to similar classes. The hyperparameters of the KNN algorithm was selected using GridSearch. The suggested n_neighbors = 8 was not suitable as it provided a lower accuracy score than the n_neighbors = 9 that was used in the final model. The KNN was selected as a base model to be used to compare with the logistic regression and XGBoost models.

3.4.2 LOGISTIC REGRESSION

The logistic regression algorithm was selected as the task at hand is a multi-class classification. The hyperparameters of the logistic regression was tuned using GridSearch. The hyperparameters that were considered were first the C value (an inverse of lambda, the regularization factor). The values ranged from 0.01 to 100 in the logarithmic scale. The resulting best value of C was 0.1 (Figure 7). The solver utilised was ‘newton-cg’ as it was appropriate for a multi-class classification.

```
grid_search.best_estimator_
```

```
LogisticRegression(C=0.1, class_weight='balanced', random_state=0,
                   solver='newton-cg')
```

Figure 7. Resulting best estimator for the logistic regression algorithm

3.4.3. XGBOOST

XGBoost (or Extreme Gradient Tree Boosting) was another algorithm best suited for supervised multi-class classification. Based on the study by Chen and Guestrin (2016) [8] and Bussmann et al. [5] (2020), it was expected to improve the predictive performance of the model.

4. RESULTS

The data was split into a training set of 80% and a test set of 20%. The K-Nearest Neighbor model was first estimated on the training set and the obtained model is applied to the test set. This was repeated with the Logistic Regression and XGBoost models. The graphs of the mean absolute error and mean squared error are as per Figure 8 below.

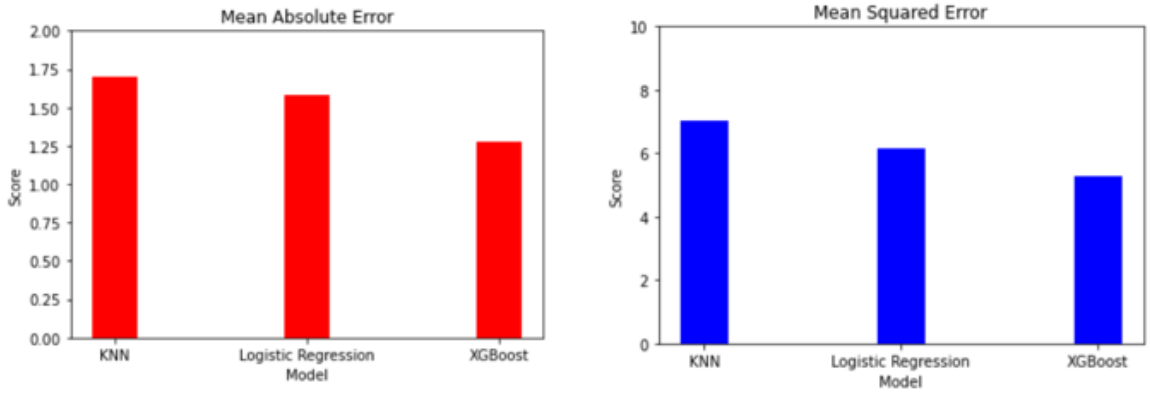

Figure 8. Summary of MAE and MSE across all three models

The use of both the Logistic Regression and XGBoost model greatly improves the predictive accuracy, as seen in the reduction in both the Mean Absolute Error and Mean Squared Error. The table below also shows a comparison of accuracy scores across the three models, confirming that the XGBoost does improve the predictive performance of the model.

	KNN	**Logistic Regression**	**XGBoost**
Accuracy Score	40.12%	43.05%	53.99%

Table 3. Accuracy scores across the three algorithms

5. CONCLUSION

The increasing use of artificial intelligence in automated decision making in the insurance industry has proved beneficial with respect to decreased processing time and reduction in human error. There is however, room for improvement for the proposed models in deciding the appropriate combinations of features that increase the predictive accuracy of the proposed models.

In light of the recent discussions surrounding Artificial Intelligence Ethics, future work that is proposed in extension of this paper is an exploration into the model explainability and transparency. As per the literature review, Bussmann et al. (2020) [5] had proposed the use of Shapley values alongside the XGBoost model in predicting peer to peer lending approval. This will contribute to the focus on fairness in AI decision making and in the context of life insurance approval, allowing the customers to understand why their application was accepted or rejected. On the other hand, this will also benefit model developers by highlighting features that have a higher contribution towards a model's results which in turn can improve feature selection.

6. REFERENCES

[1] Legal & General Assurance Society Limited, *Underwriting Explained.*
Accessed: 26 April 2021 [Online]
Available: https://www.legalandgeneral.com/_resources/pdfs/life-cover/Underwriting-Explained-W10188-11-11-Web.pdf

[2] Chen, M., Dautais, Y., Huang, L., Ge, J. (2017) *Data driven credit risk management process: a machine learning approach.* 2017 International Conference on Software and System Process. Association for Computing Machinery, 109-113

[3] Prudential Life Insurance Assessment (2016) Kaggle
Accessed: 19 April 2021 [Online]
Available: https://www.kaggle.com/c/prudential-life-insurance-assessment/overview

[4] Boodhun, N. and Jayabalan, M. (2018) *Risk Prediction in Life Insurance Industry using Supervised Learning Algorithms*, Complex & Intelligent Systems

[5] Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J. (2020) *Explainable AI in Fintech Risk Management*, Frontiers in Artificial Intelligence

[6] Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J. (2020) *Explainable Machine Learning in Credit Risk Management*, Computation Economics

[7] Zahi, S., Achchab, B. (2019) *Clustering of the population benefiting from health insurance using K-menas*, 4th International Conference on Smart City Applications 2019

[8] Chen, T., and Guestrin, C. (2016). *Xgboost: a scalable tree boosting system*, inProceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

[9] European Commission High Level Expert Group on AI on their Ethics Guidelines for Trustworthy AI. Accessed: 5 April 2021 [Online]
Available: https://ec.europa.eu/futurium/en/ai-alliance-consultation

[10] Gunning, D. (2018) *Explainable artificial intelligence (XAI)*, Defense Advanced Research Projects Agency (DARPA). Accessed: 18 April 2021 [Online]
Available: https://www.darpa.mil/program/explainable-artificial-intelligence

[11] Towards Data Science. Accessed: April-May 2021 [Online]
Available: https://towardsdatascience.com/