PREDICTING RISK FOR LIFE INSURANCE APPLICATIONS

FINAL REPORT:

SPRINGBOARD DATA SCIENCE INDEPENDENT CAPSTONE 2

BY: BRIANNA ABDALLA

6/9/2023

OVERVIEW

Traditionally, life insurance applications involve extensive paperwork and time-consuming procedures, resulting in a lengthy processing time of up to 30 days. This outdated approach hinders customer satisfaction and limits the acquisition of new customers. To overcome this, Prudential seeks to leverage data analytics to develop a model that efficiently classifies applications based on health information and predicts the associated risk score.

By successfully building an effective model, Prudential can reduce application processing times, enhance customer experience, and improve new customer acquisition. The automated risk assessment tool will provide faster and more informed decision-making capabilities, benefiting individuals seeking life insurance coverage and their families. The project aligns with Prudential's goal of modernizing the application process and utilizing data-driven insights to drive efficiency and customer-centricity in the life insurance industry.

PROBLEM STATEMENT

What opportunities exist for Prudential Life Insurance to identify the attributes associated with risk, shorten the application process and assess the risk of new applicants within 95% confidence?

THE DATA

The dataset used for this project was obtained from a Kaggle competition launched in 2016 and consists of over 58,000 completed life insurance applications. Each application contains more than 120 features and is associated with a risk value represented by the Response column. The risk values range from 1 to 8 and serve as an ordinal measure of risk. Due to the pre-normalization and encoding of the data, interpreting the meaning of individual features can be challenging. Therefore, it is crucial to develop a model that can accurately classify each risk value, ensuring appropriate acceptance or denial of life insurance applications.

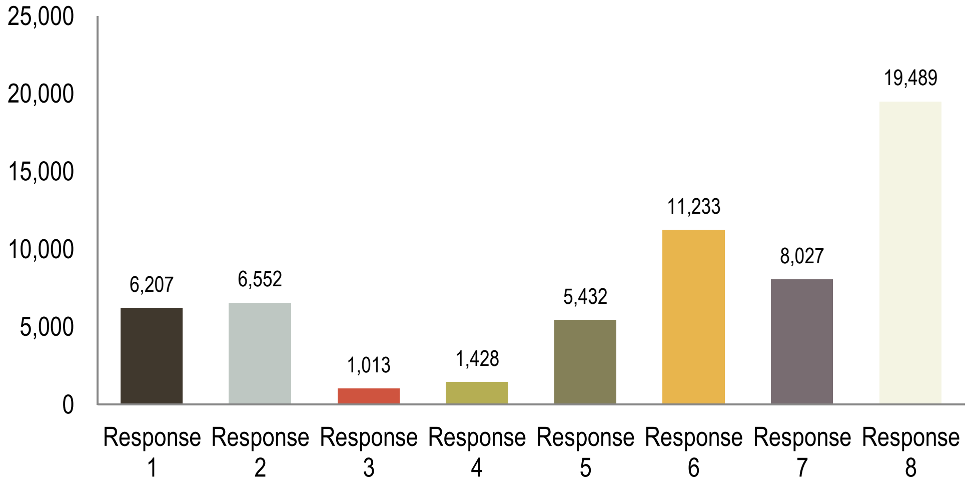
In addition to achieving accurate risk classification, it is essential to consider the severity of misclassifications in risk prediction. While any misclassification is undesirable, the impact may vary depending on the proximity between the predicted risk label and its true value. For example, misclassifying a true risk label of 1 as 2 might have less severe consequences compared to incorrectly predicting a risk label of 1 as 8. This aspect introduces a challenge, given the highly imbalanced distribution of risk values in the training dataset.

Addressing these challenges and developing an effective risk prediction model is of utmost importance to Prudential Life Insurance. It will not only improve the efficiency of their application process but also ensure appropriate risk assessment for new applicants.

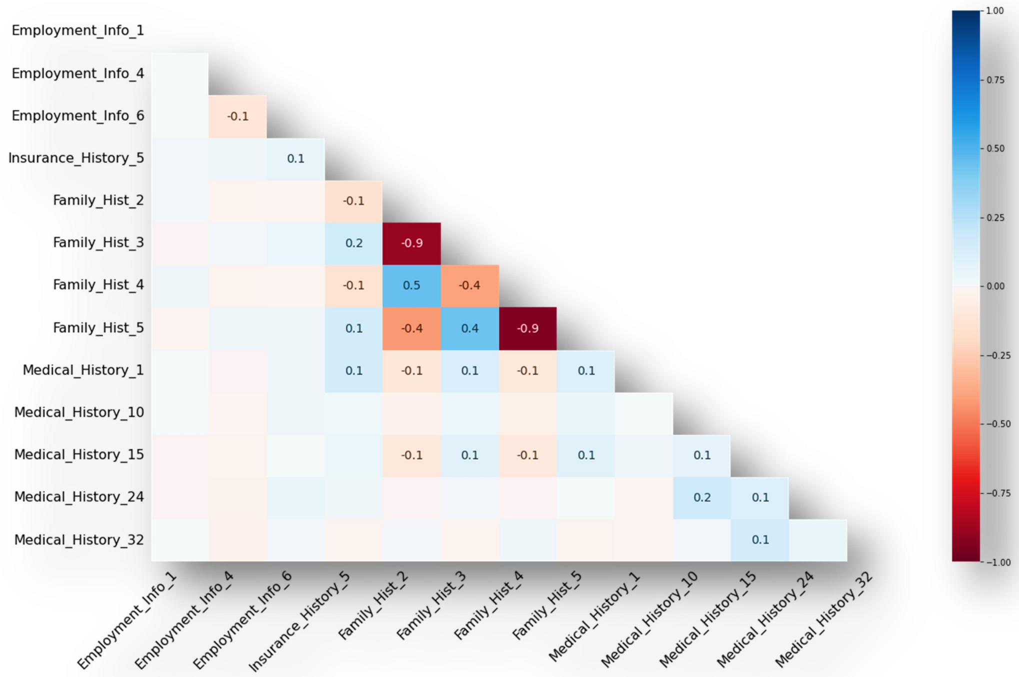
Link: https://www.kaggle.com/competitions/prudential-life-insurance-assessment/overview

DATA CLEANING

The initial phase of the project involved cleaning and organizing the data. A crucial step was to ensure the absence of duplicate Id values within the dataset, which was successfully verified. The focus then shifted to examining the target variable, Response. As previously mentioned Response comprises eight distinct values and is characterized by a highly imbalanced distribution.



The subsequent task was to identify missing values within the dataset. Nine variables were found to have missing data ranging from 30% to 99%. A heatmap revealed correlations among missing values for certain pairs of features. Recognizing the potential significance of these missing observations in relation to the target variable, it was determined that these variables would require distinct treatment.



EXPLORATORY DATA ANALYSIS

During the exploratory data analysis (EDA) phase, the primary objective was to uncover relationships among the features and their connection to the target variable. Fortunately, Prudential provided valuable information regarding the organization of the data, including categorical, continuous, discrete, and dummy variable lists. This eased navigation through the EDA process.

The conclusions drawn from the EDA are as follows:

- The categorical features consist of binary, trinary, and multiple categories.
- A categorical feature, Product_Info_2, contained alphanumeric data that required encoding.
- Outliers were observed throughout the dataset, indicating the need for models capable of handling outliers.
- A continuous variable, Family_hist_4, exhibited incorrect normalization and necessitated adjustments.
- Several correlations were identified among different variables, with BMI and Weight being the most intuitive.

Regarding the correlations with the target variable, the following observations were made:

- BMI showed a correlation of -0.382.
- Wt demonstrated a correlation of -0.351.
- Ins_Age displayed a correlation of -0.210.
- Product_Info_4 exhibited a correlation of 0.202.
- Medical_History_39 revealed a correlation of 0.220.
- Medical_History_4 showed a correlation of 0.240.
- Medical_History_15 displayed a correlation of 0.277.
- Medical_History_23 exhibited a correlation of 0.287.
- Medical_Keyword_3 demonstrated a correlation of -0.258.
- Medical_Keyword_15 revealed a correlation of -0.259.

An interesting observation is that Medical_History_15 had 75% of its values missing, yet it displayed a stronger correlation with risk values. This highlights the importance of considering incomplete data when assessing the relationship between variables and the target variable.

Overall, the EDA phase provided insights into the relationships among features, the need for data preprocessing , and the significant correlations between certain variables and the target variable.

PRE-PROCESSING

The preprocessing phase aimed to prepare the data for modeling purposes. The initial plan involved utilizing a Random Forest Classifier as the baseline model, followed by the selection of XGBoost and LightGBM models for the final modeling stage. Two different approaches were employed for the data, depending on the specific model requirements.

For the Random Forest Classifier and XGBoost models, several modifications were implemented to suit the model's needs. In this dataset, all categorical variables were subjected to one-hot encoding, and binary data were cleaned to have only 0 and 1 values, simplifying the input for the models. Discrete variables were scaled using the Min-Max scaler, and three features with over 90% missing values were dropped from the dataset. Mean imputation was applied to variables with missing values comprising 18% or less of the data. Any remaining missing values were replaced with -999 to denote null observations.

The resulting dataset for the Random Forest and XGBoost models contained over 879 features. To reduce dimensionality and improve efficiency, a feature selection technique was employed using a Random Forest Classifier. The top 100 features were then selected and used as input for the models.

In contrast, the LightGBM model utilized the original dataset without applying one-hot encoding to categorical variables. Instead, a list of categorical features was created and passed into the model's "Categorical_feature" parameter for appropriate handling. In this project, no additional feature engineering techniques were employed since the dimensionality of the data was already large, and further modifications did not appear to be necessary at the time.

By adapting the data according to the specific requirements of each model, the preprocessing phase ensured that the dataset was appropriately formatted and ready for subsequent modeling and analysis.

MODELING

The models were evaluated using ROC AUC, a suitable scoring metric for assessing performance in the presence of class imbalances. This metric ranks positive instances higher than negative instances, which is important given the imbalanced classes.

The baseline model, RandomForestClassifier, utilized a randomized search grid to optimize hyperparameters and underwent 5-fold cross-validation. It achieved an AUC score of 0.85 and an accuracy of 0.57.

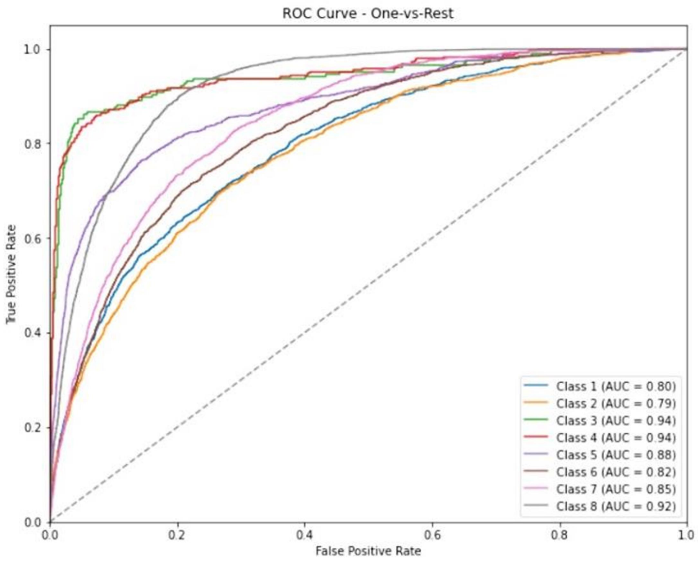
Similarly, XGBoost employed a randomized search grid and 5-fold cross-validation. It yielded an AUC score of 0.86 and an accuracy of 0.58.

The third model, LightGBM, utilized a modified dataset as described in the preprocessing section. It outperformed both RandomForestClassifier and XGBoost with an AUC score of 0.86 and an accuracy of 0.58.

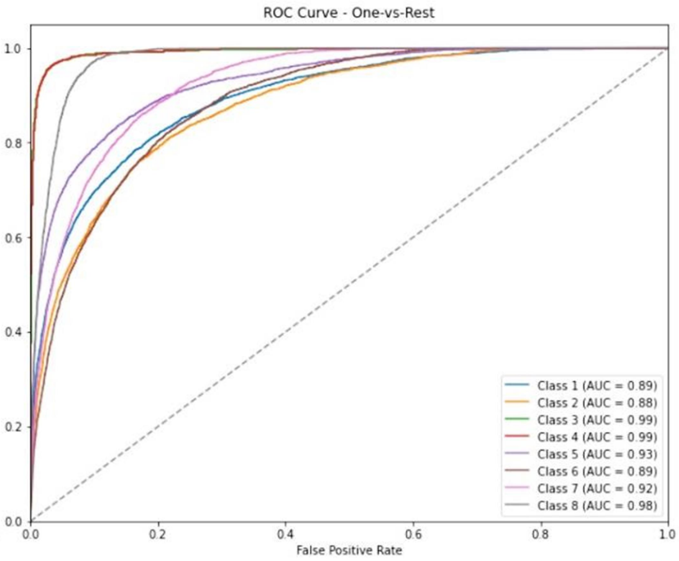
Despite achieving a decent AUC score, additional model manipulation was undertaken to enhance the classification accuracy. The ultimate and most optimal model is a LightGBM model that incorporates the SMOTE technique to oversample the minority classes during training. For this particular model, Class 8 was undersampled and limited to 9000 samples. The best-performing model, named "best_model_GBM_SMOTE," was saved to a pickle file and demonstrated an impressive AUC score of 0.93, along with an improved accuracy of 0.69.

Here are the side by side comparisons from the first LightGBM model and the final model:

First LightGBM



Final LightGBM with SMOTE



It can be seen in the ROC-AUC graphs above that both models classify 3, 4 and 8 very well while the other classes preform decently. The final model has a high probability of ranking a randomly chosen positive classification higher than a randomly chosen negative classification. This implies that the model is effective at distinguishing between positive and negative samples with very few misclassifications.

In summary, the study demonstrated the process of model selection, optimization, and manipulation to achieve the most accurate classification results. The final "best_model_GBM_SMOTE" incorporating SMOTE oversampling technique delivered impressive performance, opening avenues for practical implementation in real-world scenarios.

THE KAGGLE COMPETITION

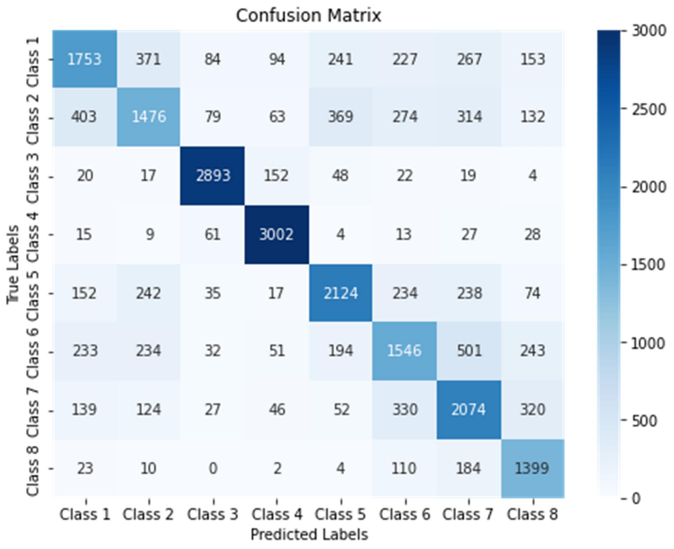
In the Kaggle competition, the submissions were evaluated using a quadratic weighted kappa metric. The highest public score achieved in the competition was 0.68325. However, for this project, only the train CSV file was utilized. The train-test split was performed solely on this data, which differs from the competition setup where a separate test set is typically used.

In our analysis, the quadratic weighted kappa was calculated on the final LightGBM model and resulted in a score of 0.64615. If this model were to be evaluated in the competition, it would have been ranked at 1385 out of 2563 among the final 40,000 plus submissions. This indicates a solid performance, although slightly lower than the top-ranking submissions.

It is important to note the differences in data usage and evaluation methodology between this project and the Kaggle competition, as it impacts the comparison of results. Nonetheless, the achieved quadratic weighted kappa score showcases the effectiveness of the final LightGBM model in predicting and classifying the target variable within a reasonable range compared to the competition's submissions.

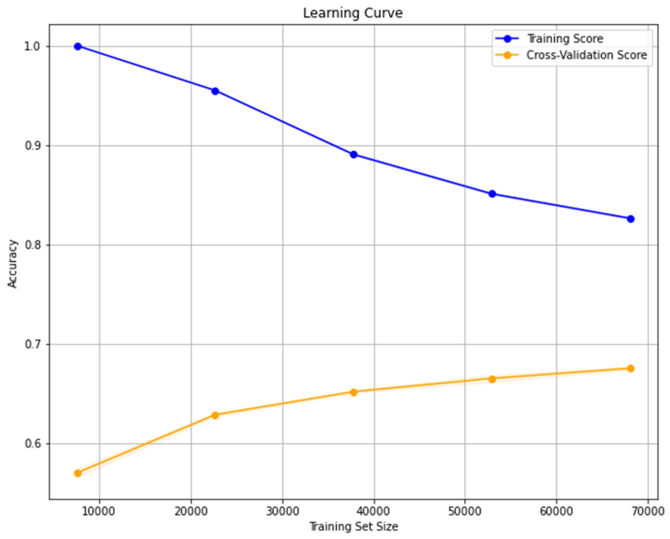
MODEL PREFORMANCE

In general, the final LightGBM model demonstrates good performance in ranking class probabilities and achieving a reasonable accuracy. However, it does exhibit some misclassifications, which could potentially pose challenges for Prudential. To provide a comprehensive summary of the model's performance and identify areas of concern, a confusion matrix is presented below. This matrix compares the true class labels against the predicted class labels.



Once again, the analysis reveals that classes 3 and 4 exhibit excellent classification performance, demonstrating minimal errors. However, other classes exhibit higher misclassification rates. Of particular concern are classes 1 and 2, as larger class labels tend to be misclassified as them more frequently. This suggests that the model may have difficulty distinguishing between classes 1 and 2, potentially leading to false predictions or misclassifications.

A learning curve was generated to investigate the impact of increasing sample sizes on model performance.



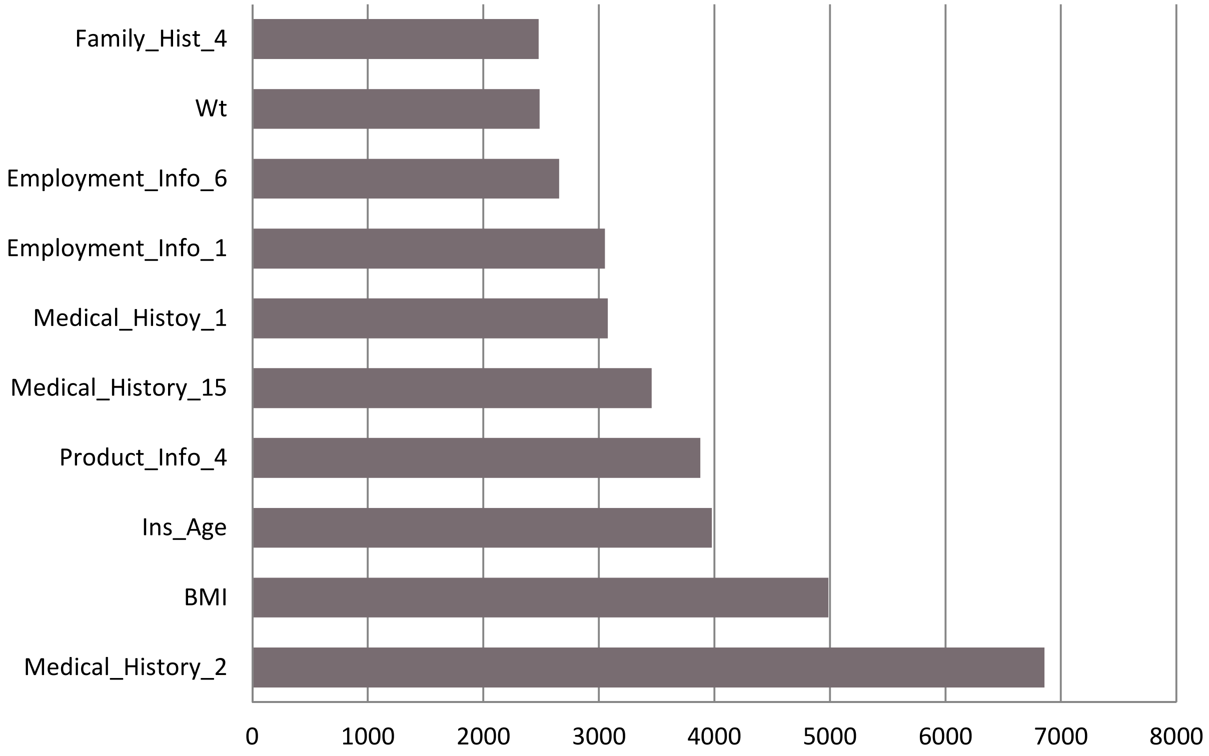
Based on the observed graph, it can be inferred that the model initially overfits the training data but becomes more balanced and better generalizes as the training set size increases. Collecting more data or applying regularization techniques could further improve the model's performance and reduce the gap between the training and validation curves.

RECOMMENDATIONS TO PRUDENTIAL

The LightGBM model with SMOTE resampling emerges as a strong candidate for evaluating life insurance applications. It achieves an AUC score of 0.93, and this performance is consistent within a 95% confidence range during cross-validation. Overall, the model demonstrates robust classification capabilities and effectively predicts the rank of different classes.

In light of these results, the recommendation for Prudential would be to identify the specific class or classes associated with the lowest risk values. By focusing on these classes, Prudential can fine-tune the model to predict them with near-perfect accuracy. This strategic adjustment would enable Prudential to efficiently approve low-risk applications, attract new customers, and separate riskier applications for further analysis.

In addition to this recommendation, provided below is the model output for feature importance in risk classification for the top 10 variables:



FUTURE IMPROVEMENTS

Future improvements for this project include utilizing the Kaggle test set and maximizing the quadratic weighted kappa score.

To achieve the stated goals and improve the project further, several ideas can be considered:

1. Explore feature engineering: Further investigation into feature engineering techniques, particularly with the inclusion of medical keywords, can potentially uncover additional meaningful patterns or relationships in the data. This could lead to improved model performance and a higher quadratic weighted kappa score.
2. Address classes 1 and 2: As classes 1 and 2 are of particular concern due to misclassifications, one approach is to assign higher weight or importance to these classes during model training. This can help prioritize their accurate prediction and reduce errors.
3. Class-specific models or iterations: Developing separate models or iterations so each individual class can be explored to improve the model's ability to distinguish between classes with similar risk values. This approach allows for more targeted and specialized modeling for each class, potentially leading to enhanced performance.

In conclusion, this project has been highly valuable for learning and has provided insights into approaching multi-class classification problems with highly unbalanced data.

CREDITS

Image Credit: Getty Images/iStockphoto

Special Thanks to my Springboard Mentor: Ajith Patnaik