Implementation of Risk Prediction in Life Insurance Industry using Supervised Learning Algorithms

Xiaorui Zhang, Yuxin Cheng, Zhanfei Gu
SYED 522 – Group 9, Option 2
Systems Design Department
University of Waterloo, Canada
{x2228zha, y2339cheng,z52gu}@uwaterloo

***Abstract*—Accurate risk prediction is pivotal in the life insurance industry for effectively classifying applicants, determining appropriate premiums, and setting policy terms. Traditional underwriting methods often rely on manual assessments and conventional statistical models, which may fall short in handling the complexity and high dimensionality of modern applicant data. This study aims to implement and compare multiple supervised learning algorithms, including Logistic Regression, Support Vector Machines (SVM), Decision Trees, and Neural Networks, to predict the risk levels of life insurance applicants. Utilizing the Prudential Life Insurance dataset, which comprises 59,381 instances and 128 attributes, we conducted a comprehensive empirical evaluation. The methodology involved data preprocessing steps such as Principal Component Analysis (PCA) for dimensionality reduction, handling missing values through iterative imputation, and encoding categorical variables via one-hot encoding. Each model underwent hyperparameter tuning using techniques like RandomizedSearchCV to optimize performance. Performance metrics, including accuracy, mean absolute error (MAE), and ROC/AUC scores, were employed to assess and compare the effectiveness of each algorithm. Our results indicate that Neural Networks achieved the highest testing accuracy, demonstrating superior capability in capturing complex, nonlinear relationships within the data. Decision Trees also performed competitively, offering a balance between interpretability and predictive power. In contrast, Logistic Regression and SVM provided reasonable performance but were outperformed by the more complex models. The findings underscore the potential of advanced machine learning techniques in enhancing risk prediction models in the life insurance sector. This study contributes to the ongoing efforts to leverage data-driven approaches for improving underwriting processes, offering insights into the strengths and limitations of various algorithms in real-world applications. Future research directions include exploring ensemble methods and alternative dimensionality reduction techniques to further enhance predictive accuracy and model robustness.**

***Index Terms*—Risk Prediction, Logistic Regression, Neural Networks, Support Vector Machines(SVM), Decision Tree**

I. Introduction

Accurate risk prediction is a cornerstone of the life insurance industry, essential for classifying applicants, determining premiums, and setting policy terms. Traditional underwriting processes often rely on manual assessments and conventional statistical models, which may not effectively handle the complex, high-dimensional data associated with modern applicant profiles. This paper focuses on implementing and comparing multiple supervised learning algorithms, including logistic regression, support vector machines, decision tree, and neural networks, to predict the risk levels of life insurance applicants.

The relevance of improving risk prediction models is underscored by several limitations in existing methodologies. Traditional statistical approaches, such as logistic and linear regression, often struggle with capturing non-linear relationships and interactions within high-dimensional datasets[1]. These models typically assume linearity and independence among predictors, assumptions that are rarely met in real-world insurance data. Furthermore, manual underwriting processes are time-consuming and subject to human error, which can lead to inconsistent risk assessments[2]. Machine learning techniques have been proposed to address these challenges by automatically learning complex patterns from data, but there is still a lack of consensus on which algorithms are most effective for this specific application[3].

Existing studies have explored various machine learning methods for life insurance risk prediction but often focus on a limited set of algorithms or lack comprehensive comparisons. Boodhun and Jayabalan implemented Multiple Linear Regression, Neural Networks, and Random Forests but did not extensively compare their performance under consistent evaluation metrics[1]. Perumalsamy et al. examined logistic regression and SVMs but did not consider ensemble methods or deep learning techniques[2]. Kasaraneni emphasized the potential of Gradient Boosting Machines but noted challenges related to overfitting and interpretability[4]. These limitations highlight a gap in the literature for a systematic evaluation of multiple supervised learning algorithms on a common dataset to identify the most effective models for risk prediction in the life insurance sector.

To address this gap, we aim to implement and compare several supervised learning algorithms on the Prudential Life Insurance dataset[5], which contains 59,381 instances and 128 attributes related to applicant information and risk levels. By conducting a thorough empirical evaluation, including data preprocessing using Principal Component Analysis (PCA), model tuning, and performance assessment using accuracy metrics, we seek to determine which algorithms offer the best balance of accuracy, efficiency, and practical applicability to improve risk assessment in life insurance underwriting.

This paper is organized as follows. Section 2 provides a background review of previous work on life insurance risk prediction, categorizing existing techniques and summarizing their contributions. Section 3 details the methodologies of the selected algorithms, including their theoretical frameworks and implementation considerations. Section 4 describes the experimental setup, including data preprocessing steps, model tuning procedures, and the evaluation metrics used for performance assessment. In Section 5, we present and analyze the results of our empirical evaluation, comparing the performance of each model and discussing their limitations. Finally, Section 6 concludes the study with insights on the best-performing approach and offers recommendations for future research directions.

II. Background

Over the years, life insurance companies have worked to streamline the sale of their products through a process known as underwriting. This process typically involves two key steps: selection and classification. The selection process determines whether an applicant qualifies for coverage, while the classification process assigns each accepted applicant to a specific risk category[6][7]. To explore advances in underwriting, our literature review will be structured into three categories: traditional statistical models, interpretable machine learning methods, and neural networks-based algorithms.

A. Traditional Statistical Models

The most commonly used models in the life insurance industry include multiple linear regression and logistic regression. As multiple linear regression is mostly used for predicting continuous outcomes, various life insurance metrics can be predicted by using it, such as expected mortality rates, premium rates, and total claims amount[8]. Moreover, Cerchiara et al. investigated the use of Generalized Linear Models to examine underwriting risks in life insurance, focusing on the variability of decrement rates which referred to the rate at which policyholders exited or left an insurance policy over time[9]. It highlighted the sensitivity of lapse rates (terminated rates) to factors such as policy duration, calendar year, and policyholder age which contributed to the development of dynamic insurance models. Further studies on life insurance lapse were also conducted using a dataset covering more than 1 million contracts. The authors then concluded that both the types of product and the characteristics of the policyholders were important drivers of the lapse rates[10]. As for logistic regression, it is a good choice for binary classification tasks such as predicting whether a customer will renew a policy, or file an insurance claim. Zhu et al. selected a data subset in The Human Mortality Database from the US and utilized logistic regression models to predict the mortality outcomes and concluded that gender, smoker status, policy duration, etc. were all significant predictors of insured mortality[11]. A study reliability analysis was also conducted to evaluate the predictive performance of logistic regression models using the AUC (Area Under the Curve). It ranged from 0.679 to 0.753 which indicated a fair to good predictive ability.

B. Interpretable Machine Learning Methods

Machine learning algorithms have been increasingly adopted in the life insurance industry, transforming traditional processes by offering more efficient and interpretable solutions to complex challenges. These advanced techniques, such as decision trees, gradient boosting machines (GBM), and support vector machines offer the advantage of handling diverse and dynamic factors affecting insurance risk. Boodhun and Jayabalan compared the prediction accuracies of multiple machine learning algorithms to enhance risk assessment in the life insurance industry and concluded that REPTree provided the best performance, achieving the lowest mean absolute error (MAE) of 1.5285 and root mean square error (RMSE) of 2.027 when using Correlation-Based Feature Selections (CFS)[1]. Sahai et al. by using the same dataset named Prudential Life Insurance available at Kaggle.com further focused on the comparative analysis between tree-based classifiers such as Decision Tree, Random Forest and XGBoost, in the context of underwriting decisions. Their findings showed that the XGBoost classifier demonstrated the best performance by achieving an AUC value of 0.86 and F1-score above 0.56 on the validation set[12]. It is also essential to note that when dealing with insurance applicants' historical data to predict risk levels, missing data is a common issue that should be carefully handled. Imputations were performed in both studies mentioned. However, Rusdah and Murfi, using the same dataset, concluded that the XGBoost model without imputation pre-processing could achieve comparable accuracy to XGBoost models with imputation[13]. Hutagaol and Mauritsius also analyzed prediction accuracies by evaluating three machine learning models: Support Vector Machine (SVM) with kernel functions, Random Forest and Naive Bayes. Their results highlighted that the Random Forest algorithm achieved the highest precision of 0.85, demonstrating its strength in correctly identifying relevant positive predictions. Meanwhile, the SVM with a linear kernel achieved a precision of 0.72, indicating it was less effective in reducing false positives compared to Random Forest[3].

C. Neural Networks

Neural networks (NNs) have revolutionized the life insurance industry by effectively modeling complex, non-linear relationships in large, high-dimensional datasets. Unlike traditional methods, NNs capture intricate interactions among factors like age, health metrics, and payment history, improving predictions for tasks such as risk classification and mortality assessment. A typical NN consists of an input layer, hidden layers with activation functions (e.g., ReLU), and an output layer, enabling it to learn meaningful patterns and relationships. For example, Boodhun and Jayabalan applied Multilayer Perceptrons (MLPs) to classify risk levels in the Prudential Life Insurance dataset, achieving higher accuracy compared to traditional methods[1]. Fernandez-Arjona demonstrated that combining feature engineering with neural networks enhanced prediction accuracy, showing the importance of selecting and transforming relevant features for improved model performance[14]. The training process of NNs relies on optimization techniques such as backpropagation and gradient descent, which iteratively update connection weights to minimize prediction error. Regularization methods like dropout and L2 regularization are commonly applied to improve the model's generalizability by reducing overfitting. Srivastava et al. showed that dropout significantly boosted the performance of neural networks in underwriting tasks, enabling the models to better adapt to new data[15]. Bolancé et al. found that neural networks excel at predicting lapse rates and mortality risks, effectively capturing subtle, non-linear dependencies in policyholder behavior. Their study demonstrated the strength of neural networks in processing diverse datasets and delivering precise predictions essential for life insurance risk assessment[16]. Mahbobi et al. also emphasized the utility of neural networks in accurately classifying applicants into risk categories, which is critical for setting fair premiums and improving the underwriting process[17][18]. Neural networks have thus proven to be a highly effective solution for analyzing large-scale, diverse insurance datasets, providing insurers with actionable insights to enhance decision-making and operational efficiency.

III. Method

This section provided a detailed explanation of the algorithms used to construct predictive models. The methods implemented include Logistic Regression, Support Vector Machine (SVM), Decision Tree, and Neural Networks. These algorithms were selected as representative and state-of-art techniques within their respective categories, as introduced in the background section.

A. Logistic Regression

Logistic Regression is a classical supervised machine learning algorithm widely used for binary and multiclass classification tasks. Despite its name including "regression," its primary goal is to predict categorical outcomes rather than continuous values. Logistic Regression builds upon a linear regression model by applying a logistic (sigmoid) function to the linear combination of features, constraining the output to the range [0, 1] to represent probabilities[19]. This makes it particularly suitable for classification problems, such as risk prediction and medical diagnosis. The main strengths of Logistic Regression lie in its simplicity and interpretability. The model's coefficients directly indicate the contribution of each feature to the classification outcome. Logistic Regression can also be extended to multiclass problems, for example, using the One-vs-Rest or Softmax classification approaches. Additionally, regularization techniques like L1 (Lasso) and L2 (Ridge) can be applied to control model complexity and prevent overfitting[20]. While Logistic Regression assumes linear separability of the data, it can handle some nonlinearity through feature transformations and interaction terms. Logistic Regression performs well in high-dimensional datasets, particularly when the number of features exceeds the number of samples. Its robustness and reliability make it a common choice in applications such as risk classification.

The foundation of Logistic Regression lies in the logistic (sigmoid) function, which maps the output of a linear model to a probability. The classification decision is made by predicting whether the probability is above or below a threshold (typically 0.5). To estimate the model parameters, Logistic Regression maximizes the log-likelihood function, which measures how well the model's predicted probabilities align with the actual labels. The process involves using optimization techniques like gradient descent to find the optimal parameters, ultimately leading to a reliable classification model.

B. Support Vector Machine(SVM)

Support Vector Machine (SVM) is an interpretable supervised machine learning algorithm primarily utilized for classification tasks. Developed by Vladimir Vapnik and his colleagues in the 1990s, SVM is designed to identify the optimal hyperplane that separates data points from different classes in a high-dimensional space[21]. By focusing on the points closest to the decision boundary, known as support vectors, SVM effectively maximizes the margin between classes. This maximization minimizes the risk of overfitting, making SVM effective even with limited training data. Additionally, SVM can be implemented with various kernel types, such as linear, polynomial, and radial basis function (RBF) kernels, allowing it to handle both linear and non-linear classification problems, thereby enhancing its generalization capabilities[22]. Furthermore, SVM performs well in scenarios where the number of features exceeds the number of observations, making it particularly effective in risk classification tasks involving numerous risk factors. All these strengths make SVM a popular choice in many fields, further solidifying its status as a state-of-the-art algorithm. The effectiveness of (linear) SVM is grounded in its mathematical foundation, which focuses on finding the optimal hyperplane that separates data points from different classes.

C. Decision Tree

Decision Trees are one of the state-of-the-art approaches to risk prediction due to their ability to provide interpretable and effective models for structured datasets, such as those in the life insurance industry. These models have demonstrated strong predictive performance in insurance applications, particularly in classifying risk levels, as evidenced by their widespread use in related studies. For example, Boodhun and Jayabalan applied Decision Trees to the Prudential Life Insurance data set and found them to be highly effective when combined with feature selection techniques [1], achieving competitive mean absolute error (MAE). In addition, their ability to handle both categorical and numerical data without requiring extensive preprocessing makes them a versatile tool in underwriting tasks. In this study, a Decision Tree Classifier was implemented to predict the risk categories of life insurance applicants. The model was trained on the PCA-reduced dataset, retaining 80% of the variance, which simplified the dataset while preserving its essential information. The Decision Tree used the entropy criterion to measure information gain at each split, ensuring optimal partitioning of the data based on its features. The maximum depth of the tree was capped at 5 to prevent overfitting and maintain interpretability.

This approach was selected for its interpretability and robustness in identifying key features that influence risk classification. Decision Trees inherently rank feature importance, enabling insurers to understand the relative contribution of variables such as age, BMI, and employment history in assessing risk. Hyperparameter tuning via RandomizedSearchCV further optimized the model, identifying parameters such as minimum samples per split and leaf size for enhanced performance. By providing a clear and explainable decision-making framework, this method bridges the gap between complex machine learning algorithms and practical application in insurance underwriting.

D. Neural Networks

Neural Networks are powerful supervised learning algorithms inspired by the structure and functioning of the human brain. They consist of interconnected layers of nodes (neurons), where each layer transforms the data through weighted connections and activation functions. Neural Networks are highly flexible and can model complex, non-linear relationships, making them widely used in classification and regression tasks. [23] The architecture of a Neural Network typically includes an input layer, one or more hidden layers, and an output layer. Each neuron processes the input it receives using a weighted sum, applies an activation function, and passes the result to the next layer. Common activation functions include ReLU (Rectified Linear Unit) for hidden layers and sigmoid or softmax for the output layer. [24] Neural Networks excel in tasks with large datasets and high-dimensional input spaces. They can learn intricate patterns that are challenging for traditional algorithms, making them particularly effective in risk prediction scenarios where multiple features and complex dependencies are present.

The foundation of Neural Networks lies in the forward and backward propagation processes. In forward propagation, each neuron computes a weighted sum of its inputs and applies an activation function to determine its output. The network's output is then compared to the true labels using a loss function, such as binary cross-entropy, to assess how well the model is performing. Backward propagation involves calculating the gradients of the loss function with respect to the weights using the chain rule. This information is used to update the weights in order to minimize the loss and improve the model's accuracy. Neural Networks' ability to approximate non-linear functions and their scalability make them a powerful tool for predictive modeling in the life insurance industry.

IV. Results & Analysis

In this section, we evaluated the predictive performance of four machine learning models on the Prudential Life Insurance dataset. We examined the impact of dimensionality reduction and the effect of standardization on model performance as well. Key performance metrics were analyzed to assess the trade-offs between model complexity and predictive performance. These findings provide insights into the effectiveness of the suitability of different machine learning approaches for life insurance risk assessment.

A. Description of Dataset

The chosen dataset to conduct our experiment was the Prudential Life Insurance dataset [5]. It consists of 59,381 applications, and in total 128 attributes containing information about life insurance applicants, including their demographic, health, and lifestyle attributes. Table I describes the variables present in the dataset.

Variable	**Description**
Id	Unique identifier for each observation.
Product_Info_1 to 7	Information regarding product type.
Int_Age, Ht, Wt, BMI	Normalized age, height, weight, BMI.
Employment_Info_1 to 6	Employment-related information.
InsuredInfo_1 to 3	Personal information.
Insurance_History_1 to 3	History of insurance claims.
Family_Hist_1	Family medical history.
Medical_History_1 to 2	Medical history information.
Medical_Keyword_1	Presence of medical keyword.
Response	Response variable: Level 1 to 8.

TABLE I: Dataset Variable Description

B. Data Pre-processing

In this study, the response variable was changed to binary to simplify our experiment. Levels 1 to 7 were assumed to represent applications being rejected, while level 8 was assumed to indicate an approved application. (Appendix: Fig 1) This simplification also aligned with the typical decision-making process in the life insurance industry.

C. Data Cleaning

The data cleaning step involved four key actions. Firstly, features with more than 30% missing data were removed, leaving a total of 119 features along with one response variable. Secondly, we assumed the remaining missing data were MAR (Missing at Random) so IterativeImputer from the scikit-learn library was used to impute the missing values [3]. Thirdly, categorical variables were encoded using one-hot encoding to make them suitable for machine learning algorithms. Lastly, all independent variables were standardized to bring them to a uniform scale, and the dataset was split into training and testing sets for model development and evaluation.

D. Dimensionality Reduction

Principal Component Analysis (PCA) was applied for dimensionality reduction to simplify the dataset while retaining essential information. We experimented with retaining 40% and 80% of the explainable variance, as well as using the dataset without applying PCA, to compare the impact of different levels of dimensionality reduction on model performance. All datasets were standardized prior to analysis to ensure consistency and enhance model performance.

E. Modeling

1) Logistic Regression: For Logistic Regression, we performed hyperparameter tuning using RandomizedSearchCV. The hyperparameter grid included different solvers (liblinear, lbfgs), L2 regularization as the penalty, and regularization strengths (C) with values of 1.0, 0.1, and 0.01. Cross-validation was conducted using RepeatedStratifiedKFold with 5 splits and 1 repeat to ensure robust evaluation. The optimization process selected the best parameters by maximizing accuracy on the validation sets. The final model was trained using the optimized hyperparameters and evaluated on both the training and testing datasets. Table II below summarized the best hyperparameters obtained across three data setups: the standardized original dataset, PCA retained 80% of the variance, and PCA retained 40% of the variance. Table III presented the corresponding best performance metrics including accuracy, MAE, and ROC/AUC score, across the same setups.

The results demonstrated that the standardized original dataset provided the best predictive performance, achieving the highest testing accuracy. This indicated that dimensionality reduction through PCA might not always enhance model effectiveness and suggested that the richness of the original dataset's features contributed significantly to the Logistic Regression model's success.

TABLE II: LR Part 1: Hyperparameter Tuning

Data Setup	Solver	Penalty	C	Training Accuracy
Original Data	lbfgs	L2	0.1	0.8162
PCA (80%)	lbfgs	L2	0.01	0.7954
PCA (40%)	lbfgs	L2	1.0	0.7862

TABLE III: LR Part 2: Performance Metrics

Data Setup	Testing Accuracy	MAE	ROC/AUC Score
Original Data	0.8128	0.1872	0.8852
PCA (80%)	0.7927	0.2073	0.8647
PCA (40%)	0.7830	0.2170	0.8530

2) Support Vector Machine (SVM): For the Linear Support Vector Machine (SVM), we conducted hyperparameter tuning to optimize its performance using Stochastic Gradient Descent (SGD) as the optimizer. Regularization strength α was varied across 0.1, 1, and 10 to balance the trade-off between margin maximization and classification error. Both L1 and L2 penalties were explored to assess the impact of different regularization techniques, while the hinge loss function was used as the loss function. We tested constant and decay learning rate schedules, with the initial learning rate (η_0) sampled from a uniform range between 0.01 and 0.1. Additionally, the decay rate of the learning rate was tuned with values of 0.5 and 0.8. Finally, the maximum number of iterations was set to 5000 to allow sufficient convergence during training. Table IV and V presented our results of hyperparameter tuning and performance metrics for SVM. Similar to Logistic Regression, SVM performed the best with the standardized original dataset as well.

TABLE IV: SVM Part 1: Hyperparameter Tuning

Data Setup	α	Penalty	η_0	Power t	Training Accuracy
Original Data	0.1	L2	0.07	0.5	0.8122
PCA (80%)	0.1	L2	0.07	0.5	0.7897
PCA (40%)	0.1	L2	0.07	0.5	0.7815

TABLE V: SVM Part 2: Performance Metrics

Data Setup	Testing Accuracy	MAE	ROC/AUC Score
Original Data	0.8089	0.1911	0.8787
PCA (80%)	0.7899	0.2101	0.8595
PCA (40%)	0.7814	0.2186	0.8486

3) Decision Tree: For the Decision Tree Classifier, hyperparameter tuning was conducted to optimize its performance on the Prudential Life Insurance dataset. The key parameters tuned were the maximum depth of the tree ("Max Depth") and the minimum number of samples required in a leaf node ("Min Leaf"). Using RandomizedSearchCV, the optimal parameters were identified as a maximum depth of 7 and a minimum leaf size of 2 across all data setups. This ensured a balance between model complexity and generalizability, mitigating overfitting while preserving predictive power. Entropy was chosen as the splitting criterion to maximize information gain at each node. The Decision Tree Classifier used the "entropy" criterion for optimization, effectively splitting data based on information gain. Performance was evaluated using the same metrics mentioned above, offering assessment of classification effectiveness, model error, and discrimination ability.

As shown in Tables VI and VII, the Decision Tree achieved the highest testing accuracy (0.8177) and ROC AUC (0.8785) on the original dataset, outperforming the PCA-reduced datasets (80% and 40% variance). The PCA-transformed datasets exhibited slightly lower testing accuracy (0.7599 for 80% and 0.7592 for 40%), indicating that dimensionality reduction removed potentially relevant features. Additionally, the original dataset achieved the lowest MAE (0.1823), suggesting better predictive precision. The superior performance of the Decision Tree on the original dataset can be attributed to its ability to leverage the complete set of features, which retained all relevant information. PCA, while reducing dimensionality, may have excluded subtle but important predictors. Moreover, standardization did not significantly impact performance since Decision Trees are insensitive to feature scaling.

TABLE VI: Decision Tree Part 1: Hyperparameter Tuning

Data Setup	Max Depth	Min Leaf	Training Accuracy
Original Data	7	2	0.8162
PCA (80%)	7	2	0.7758
PCA (40%)	7	2	0.7579

TABLE VII: Decision Tree Part 2: Performance Metrics

Data Setup	Testing Accuracy	MAE	ROC AUC
Original Data	0.8177	0.1823	0.8785
PCA (80%)	0.7599	0.2401	0.8123
PCA (40%)	0.7592	0.2408	0.8111

Initially, the Decision Tree experienced overfitting, particularly on the original dataset, where it achieved an unreasonable training accuracy of 1. This was then mitigated by limiting the tree depth to 7 and requiring at least 2 samples per leaf node. One-hot encoding of categorical variables, while useful for compatibility, may have introduced sparsity in the data, reducing model efficiency. Additionally, without PCA, the Decision Tree retained its interpretability and feature importance rankings, whereas PCA-transformed features were less interpretable. Finally, while standardization typically benefits other models, its negligible effect on Decision Tree performance highlights the algorithm's inherent robustness to feature scaling. Overall, the Decision Tree without PCA demonstrated superior performance, underscoring the value of retaining full feature sets for risk prediction in life insurance.

4) Neural Network: For the Neural Network, we utilized a model with two hidden layers and a binary classification output layer, reflecting our output, which could either be 0 or 1. We kept the number of layers constant to manage runtime. The network was optimized using a grid search approach. Key hyperparameters included the number of neurons in each layer, learning rate, dropout rate, activation function chosen between Relu and Sigmoid, batch size, and number of epochs. The optimization process was conducted with a 5-fold cross-validation setup, aiming to maximize accuracy. The Adam optimizer was used to minimize binary cross-entropy loss. Additionally, early stopping and learning rate reduction were incorporated to enhance training efficiency and model performance. The results were summarized in the following two tables. Similarly, NN performed the best with the standardized original dataset.

TABLE VIII: Neural Network Part 1: Hyperparameter Tuning

Data Setup	Activation	Dropouot	Training Accuracy
Original Data	Sigmoid	0.2	0.8546
PCA (80%)	Sigmoid	0.2	0.8316
PCA (40%)	Relu	0.2	0.8265

TABLE IX: Neural Network Part 2: Performance Metrics

Data Setup	Testing Accuracy	MAE	ROC AUC
Original Data	0.8244	0.1756	0.8972
PCA (80%)	0.8156	0.1844	0.8913
PCA (40%)	0.8063	0.1937	0.8808

F. Model Evaluation

Table X presented a comparison of testing accuracy across the algorithms discussed, as well as an evaluation of the original dataset both with standardization (original) and without standardization (original unstd). Notably, the standardized original dataset outperformed the PCA-transformed datasets for all four algorithms. This could be attributed to the relatively small number of features, 128 features in the original dataset, which might limit PCA's ability to capture sufficient variance for enhancing model performance. Overall, Neural Network emerged as the top-performing algorithm in the analysis.

The study faced several limitations and challenges, including the use of a relatively small dataset, whereas in reality, risk factors could be more complex and correlated. Besides the implementation of one-hoc encoding, while necessary for categorical variables, might have negatively impacted the performance of Decision Tree models. Additionally, potential issues with data quality, such as the method chosen to input missing values, could have influenced the accuracy of the results. Furthermore, due to runtime constraints, the hyperparameter search was limited, which might have affected the overall performance.

V. Conclusions

This study implemented and evaluated four supervised learning algorithms—Logistic Regression, Support Vector Machines (SVM), Decision Tree, and Neural Networks—for predicting risk levels in the life insurance industry. Through comprehensive hyperparameter tuning and performance analysis, the results highlighted the advantages and limitations of each approach in handling high-dimensional datasets and capturing complex relationships.

Among the algorithms tested, Neural Networks emerged as the top-performing approach, achieving the highest testing accuracy (82.44%) and ROC AUC (0.8972) on the original standardized dataset. Its ability to model non-linear relationships and capture intricate dependencies among features makes it a robust choice for life insurance risk prediction. However, this algorithm is computationally intensive and requires careful hyperparameter tuning to prevent overfitting, as well as significant domain expertise for interpretability. The Decision Tree classifier also performed well, achieving competitive testing accuracy (81.77%) and ROC AUC (0.8785) on the original dataset. Its simplicity, interpretability, and capability to rank feature importance make it highly suitable for practical applications in insurance underwriting. However, it is prone to overfitting without proper parameter constraints, as evidenced in the initial stages of our study. Logistic Regression and SVM provided reasonable performance but fell short of Neural Networks and Decision Trees. Logistic Regression demonstrated robustness and interpretability, making it suitable for datasets with linear separability, while SVM excelled in handling high-dimensional data. However, both algorithms struggled to capture non-linear relationships inherent in the dataset.

Despite these results, none of the techniques tested can definitively solve the problem of risk prediction in the life insurance industry. The complexity and variability of applicant profiles necessitate ongoing improvements to address issues like overfitting, missing data, and imbalanced datasets. Future study under this area could explore alternative dimensionality reduction techniques beyond PCA to identify critical features more effectively. Expanding the hyperparameter search space for the algorithms may further enhance their predictive performance by optimizing configurations more comprehensively. Additionally, implementing stacking models, which combine the strengths of multiple algorithms, might further improve the accuracy of risk predictions.

VI. References

[1] Boodhun, N., & Jayabalan, M. (2018). Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems*, *4*(2), 145–154. doi: 10.1007/s40747-018-0072-1.

[2] Perumalsamy, J., Konidena, B. K., & Krothapalli, B. (2023). AI-driven risk modeling in life insurance: Advanced techniques for mortality and longevity prediction. *Journal of Artificial Intelligence Research and Applications*, Sep. 2023. [Online]. Available: https://aimlstudies.co.uk/index.php/jaira/article/view/157.

[3] Hutagaol, B. J., & Mauritsius, T. (2020). Risk level prediction of life insurance applicants using Machine Learning. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(9), 299–306. doi: 10.30534/ijatcse/2020/199922020.

[4] Kasaraneni, B. P. (2019). Advanced artificial intelligence techniques for predictive analytics in life insurance: Enhancing risk assessment and pricing accuracy. *Distributed Learning and Broad Applications in Scientific Research*, Dec. 2019. [Online]. Available: https://dlabi.org/index.php/journal/article/view/121.

[5] Prudential Life Insurance Assessment, Kaggle. (n.d.). [Online]. Available: https://www.kaggle.com/competitions/prudential-life-insurance-assessment/data.

[6] Mishra, K. (2016). *Fundamentals of life insurance theories and applications* (2nd ed.). PHI Learning Pvt. Ltd.

[7] Wang, Y. P. (2021). Predictive machine learning for underwriting life and health insurance. Presented at the *Life Insurance Conference*, Oct. 2021, pp. 19–22.

[8] Rawat, S., Rawat, A., Kumar, D., & Sabitha, A. S. (2021). Application of machine learning and data visualization techniques for decision support in the insurance sector. *International Journal of Information Management and Data Insights*, *12*, 100012. doi: 10.1016/j.jjimei.2021.100012.

[9] Cerchiara, R. R., Edwards, M., & Gambini, A. (2009). Generalized linear models in life insurance: Decrements and risk factor analysis under Solvency II. Presented at the *AFIR Colloquium*, Rome, 2009.

[10] Eling, M., & Kiesenbauer, D. (2014). What policy features determine life insurance lapse? An analysis of the German market. *The Journal of Risk and Insurance*, *81*(2), 241–269. doi: 10.1111/j.1539-6975.2012.01504.x.

[11] Zhu, Z., Li, Z., Wylde, D., Failor, M., & Hrischenko, G. (2015). Logistic regression for insured mortality experience studies. *North American Actuarial Journal*, *19*(4), 241–255. doi: 10.1080/10920277.2015.1039135.

[12] Sahai, R., et al. (2023). Insurance risk prediction using machine learning. In *Data Science and Emerging Technologies*, vol. 165, Springer Singapore Pte. Limited, pp. 419–433. doi: 10.1007/978-981-99-0741-0_30.

[13] Rusdah, D. A., & Murfi, H. (2020). XGBoost in handling missing values for life insurance risk prediction. *SN Applied Sciences*, *2*(8), 1336. doi: 10.1007/s42452-020-3128-y.

[14] Fernandez-Arjona, L. (2021). A neural network model for solvency calculations in life insurance. *Annals of Actuarial Science*, *15*(2), 259–275.

[15] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

[16] Bolancé, C., Guillen, M., & Padilla-Barreto, A. E. (2016). Predicting probability of customer churn in insurance. In *Modeling and Simulation in Engineering, Economics and Management*, vol. 135, Springer International Publishing, pp. 82–91.

[17] Mahbobi, M., Kimiagari, S., & Vasudevan, M. (2023). Credit risk classification: An integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*, *330*(1), 609–637.

[18] J. Perumalsamy, C. Althati, and M. Muthusubramanian, "Leveraging AI for mortality risk prediction in life insurance: Techniques, models, and real-world applications," *Journal of Artificial Intelligence Research*, Jan. 2023. [Online]. Available: https://www.thesciencebrigade.com/JAIR/article/view/266.

[19] Hosmer, D. W., Lemeshow, R. X., & Sturdivant, R. (2013). *Applied logistic regression*. John Wiley & Sons.

[20] Pedregosa, F. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

[21] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. doi: 10.1007/BF00994018.

[22] Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

[23] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: The MIT Press.

[24] Nielsen, A. (2015). *Neural networks and deep learning*.

VII. Appendix

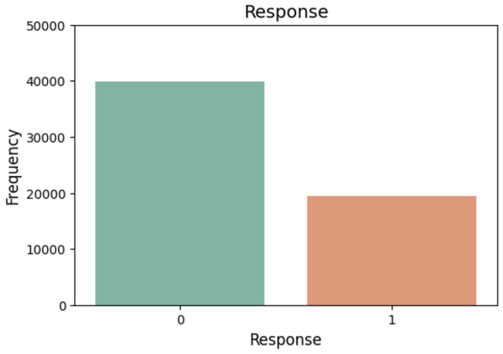

Fig. 1: Distribution of Responses: Frequency of Binary Outcomes (0 and 1)