MSc in Data Analytics

Life Assurance Application Conversion Prediction using Supervised Machine Learning

Greg Langella

Student Number: sba22194

# **Research Problem**

This research project aims to implement a supervised machine learning model that can predict the likelihood of a life assurance application being converted into an active policy. This prediction can help the Life Assurance Company allocate resources effectively and improve their overall business efficiency.

The project will involve analysing various data points, including product selections and application details to identify the factors that affect the conversion of life assurance applications. By leveraging machine learning algorithms, the model will be able to provide a probability score for each application, indicating the likelihood of the application being converted.

The project is pertinent to Data Analytics because it involves the use of supervised machine learning techniques to tackle a real-world problem in the life assurance sector. To build an accurate and efficient machine learning model, the project will necessitate the usage of several data analysis techniques such as data cleansing, feature engineering, and model selection. Furthermore, the project will involve the application of numerous best practices methodologies in data analytics to assure the model's validity and reliability.

# **Research Objectives**

Based on the research topic, four research objectives that could be addressed using the Problem Definition model have been identified:

1. Evaluate the impact of the independent features’ correlation with the dependent variable and the impact of feature importance score to determine the variables that have the greatest impact on the conversion of life assurance applications, and develop a supervised machine learning model using the features that have the greatest impact.
2. Design and execute experiments to test the impact of features that are most highly correlated with application conversion or have the highest feature importance scores. The findings of these experiments should help to identify the features that are most strongly associated with application conversion, validate the importance of specific features in predicting application conversion, and refine the predictive model to improve its performance.
3. Critically evaluate and examine the effectiveness of multiple supervised machine learning models and their hyperparameters in predicting the likelihood of life assurance application conversion accurately and how the hyperparameters contribute to the performance of the machine learning models.
4. Validate the performance of the developed machine learning models for predicting the likelihood of life assurance application conversion accurately to evaluate their generalisability, by testing them on unseen data, and comparing this performance to the performance achieved in training of the models. The findings of this can provide valuable insights into the performance and reliability of the machine learning models, and be used to indicate areas for additional model improvement and modification to improve performance and applicability.

These objectives align with the research topic, as they aim to develop and implement a supervised machine learning model that can accurately predict the likelihood of a life assurance application being converted into an active policy. The objectives also address the key challenges and requirements of the project, such as data analysis techniques, model selection, and evaluation metrics.

# **Literature Review**

Life assurance companies can receive vast volumes applications every day, and it is vital to determine which applications are most likely to be converted into a policy. Predictive analytics, and more specifically, supervised machine learning models have proven to be effective tools for such tasks. This literature review aims to provide an overview of the current research related to the development and implementation of supervised machine learning models for insurance application conversion prediction. In the context of life assurance application conversion prediction, supervised machine learning models can be trained on application data to identify the factors that impact conversion and predict the likelihood of a new application being converted.

# **Machine Learning in Insurance**

Machine Learning techniques have been used within the insurance industry for some time to address a range of classification problems and feature importance identification, including prediction of customer purchase behaviour.

van der Putten et al. (2000) conducted a predictive modelling competition, the CoIL Challenge 2000, with the goal of identifying potential customers for insurance policies and explaining the factors that influence their purchasing decisions. The results showed that machine learning algorithms can be used effectively to correctly predict insurance policy purchasing, with accuracy rates as high as 88%. The best predicting features of policy ownership included demographic variables as well as variables related to the insurance product benefits. However, the effectiveness of the models depended on the specific characteristics of the sample population. This study provided a yardstick for evaluating the performance of machine learning models for predicting insurance policy purchasing, and informed the feature selection and engineering process.

Chang & Lai (2021) adopted a neural network-based approach to predict customer intentions of purchasing insurance policies. The researchers utilized the dataset from the CoIL Challenge 2000 (van der Putten et al.,2000) and employed three data pre-processing approaches to address the issue of imbalanced class distributions. The results obtained were found to be comparable with the top performing entries of the CoIL Challenge 2000, indicating the efficiency of the proposed model in predicting customer intentions. The study also aimed to identify factors that impact probability to purchase insurance policies via feature selection. Neighbourhood component analysis (NCA), sequential forward selection (SFS), and sequential backward selection (SBS) were used. The results of implementing SFS and eliminating socio-demographic features were found to be comparable with other submissions of the CoIL Challenge 2000. The study highlighted the importance of using machine learning approaches, such as artificial neural networks, in predicting intention of purchasing insurance policies. The utilisation of various data pre-processing techniques, including feature selection, feature construction, and under-sampling, proved useful in addressing the issue of imbalanced datasets.

Rubi et al. (2022) evaluated ten classification algorithms to select a model with the highest accuracy in predicting whether a customer would purchase insurance or not. Random Forest, Decision Tree Classifier, and Stochastic Gradient Descent models provided the highest levels of accuracy. The research provides important insights into identifying the features that have impact on decisions of customers when considering whether or not to by an insurance policy. The use of machine learning techniques to predict insurance purchasing behaviour is specifically relevant as it provides a data-driven approach to understanding customers behaviour.

Ampt (2017) aimed to investigate the potential of machine learning techniques in predicting customer interest in insurance products. The study utilised ten classification algorithms and conducted six experiments to determine which machine learning technique had the highest potential for predicting insurance product interest. It was found that the Decision Tree and Logistic Regression algorithms showed the highest potential for predicting insurance product interest. The accuracy achieved by machine learning techniques was up to 94%, allowing for confident predictions of customer insurance product interest. Moreover, machine learning showed ability in the handling of irrelevant features, eliminating the need for data scientists to comb through data to pick relevant features. Overall, the study suggests that machine learning can be an effective tool for predicting customer interest in insurance products, and recommends the use of Decision Tree and Logistic Regression algorithms for this purpose.

An et al. (2021) proposed a predictive model to predict whether existing health insurance customers were likely to purchase car insurance. Using logistic regression and boosted decision tree algorithms, they were able to develop a model with a high level of accuracy. They further concluded that this model could be useful for predicting customer behaviour, particularly insurance policy purchase intention.

Mau et al. (2018) aimed to accurately forecast the likelihood of purchasing life insurance using digital consumer data. The study discovered that customer data fed into a random forest model, yielded a prediction accuracy of over 90%. Overall, the study demonstrated that using consumer reaction data might significantly improve the accuracy of predicting purchase behaviour in the insurance industry. Through the research of Mau et al. (2018), Rubi et al. (2022), Ampt (2017), and Mau et al. (2018), it is clear that there is high level of support for the use of Decision Tree algorithm and their extensions, such as Random Forests, for accurately predicting customer purchasing intention in the insurance industry. There is also an element of support for Logistic Regression for prediction purposes.

Jaiswal (2022) explored the use of big data and machine learning in predicting the intention of a customer to purchase an insurance policy. Among six machine learning models, it was concluded that LightGBM was the most suitable for predicting purchase intention. Jaiswal further recommends the use of personal, geographical, and regional factors to predict the likelihood of a customer completing the policy purchase.

Mai et al. (2020) found that purchase intention, attitudes, financial awareness, and product accessibility all influence life insurance purchasing behaviour. Also, Nomi & Sabbir (2020) investigated the characteristics that influence consumer purchasing intentions for life insurance. According to the findings, attitude, subjective norms, risk aversion motives, saving motives, and financial literacy all have a significant favourable impact on customer purchase intention for life insurance. Saving motives have been recognized as a mediator in the association between risk aversion motives and purchase intention, as well as the relationship between financial literacy and purchase intention.

A range of studies have been carried out on the use of machine learning for other classification purposes within the non-life insurance industries, mostly churn prediction, claim prediction, and fraud detection.

Random Forest and AdaBoost have proven to be effective for classification purposes within the insurance industry, particularly in predicting customer churn (Stucki, 2019). It was found that that machine learning in general was a better and more feasible method of predicting customer churn than methods traditionally used within the insurance industry. Groll et al. (2022) explored the use of machine learning to predict policy cancellation likelihood. They found no significant difference observed between the performances of tree-based and logistic regression approaches to classify the life insurance policies. Mauritsius et al. (2020) evaluated the ability of each of Decision Tree, Naïve Bayes, and Artificial Neural Network as classification methods for a customer churn problem of an insurance company. In this case Decision Tree was found to be the most suitable approach for the creation of a customer churn model. Zhang et al. (2017) proposed a combined Deep & Shallow model for a classification task related to customer churn prediction within the insurance industry. They argue that this type of model has advantages such as generalisation and memorisation being present in one model. It was also concluded that the combined model outperformed both the deep-only and shallow-only methods in the classification task.

Pesantez-Narvaez et al. (2019) compared the use of logistic regression and XGBoost algorithms for classification purposes in the prediction of claims. Better predictive capacity and interpretability meant that logistic regression was the more suitable approach for problem presented in this study. Among a range of machine learning methods, Random Forest was found to have the best accuracy and generalisablity when it came to the classification task based on the insurance policy and customer data (Hanafy & Ming, 2021). The use of decision trees, and their subsequent extensions, such as gradient boosting and random forests was explored by Quan & Valdez (2018) as potential predictive models for insurance claim prediction. They found that multivariate tree-based models generally outperform univariate tree-based models. Frempong et al. (2017) developed a decision tree predictive model to predict the likelihood of a claim being made based on a number of risk factors within the insurance industry. While developing the model, they discovered that certain features had a greater impact on the likelihood of an insurance claim being made. Differences have been observed between how traditional machine learning methods deal with classification tasks in the insurance industry, and how deep learning approaches deal with the same (McDonnell et al., 2023). In claim prediction tasks, a deep learning architecture called TabNet outperformed more traditional machine learning models such as GLMs and XGBoost in terms of interpretability and accuracy. However, it was noted that the time to run TabNet and effort needed for hyperparameter tuning are possible limitations that must be considered.

Severino & Peng (2021) found that ensemble methods were most effective for fraud detection within property insurance, outperforming a range of other models, including logistic regression. The best suited model, used along with feature selection techniques, can be adapted for a probabilistic approach and improved with spatial analysis and other machine learning algorithms. However, their study did not use imbalanced classification methods or hyperparameter tuning, resulting in a gap in their research. Xia et al. (2022) determined that deep learning models combining CNN, LSTM, and DNN can perform better than traditional machine learning models in classification task, such as fraud detection, within insurance. They explain that deep learning is better suited to deal with the high dimensionality and large amounts of data that are often present within insurance dataset. Deep learning classification models have had issues with classification tasks within the insurance industry when the dataset is imbalanced. Muranda et al. (2021) addressed this problem by using sampling techniques while pre-processing the data in order to give balance to the dataset. In their experiments, the deep learning classification models performed well in detecting fraudulent claims when the dataset was balanced, but performed less well on imbalanced datasets.

Taha et al. (2022) considered the importance of feature selection when applying machine learning in the insurance industry. One particular challenge is the amount of noise often present within insurance datasets, and the subsequent negative impact this can have on performance of machine learning models. Taha et al. propose that this can be dealt with by using a selected set of features over the use of an entire dataset without feature selection applied. The most powerful variables on the intention to purchase life insurance are financial knowledge and attitude toward the purchase of life insurance. The influence of product accessibility, risk perception, and subjective norms on insurance intention is quantified and explored. Li (2019) highlighted the difficulties that insurance firms confront in staying competitive and discovering worthwhile consumers. The article discussed the use of customer data for descriptive statistical analysis and data cleaning in order to improve data quality. The paper then discussed how to estimate client preferences for life insurance products using logistic regression models, decision trees, and random forests. The results suggest that the combination model of random forest and logistic regression predicts customer behaviour the best. Dragos et al. (2020) presented an empirical study designed to better understand the impact of behavioural and socio-demographic characteristics on purchasing. The study discovered that specific behavioural characteristics and insurance knowledge are significantly significant for the purchase choice but not for the purchase intention. The study also discovered that financial education, as measured by a self-constructed Index of Insurance Knowledge, has a significant impact in explaining financial decisions. Life insurance is strongly encouraged by marital status, high levels of education, and income.

There are a number of ethical issues to consider when using data analytics and machine learning within the insurance industry, particularly in relation to issues around discrimination and fairness (Barry & Charpentier, 2022). Biases such as the use of irrelevant features and correlated but not causal features can be seen within machine learning approaches. It was concluded that contestability and transparency should be adopted when using machine learning within insurance in order to ensure fairness within in insurance. According to Anagol et al. (2017), instead of focusing on the coverage customers require, agents overwhelmingly propose unsuitable, high commission products and cater to the assumptions of misinformed consumers. Poor advice is motivated by commission incentives and agents' insufficient product expertise. The study also argues that financial product disclosure standards should be similar across the menu of substitutable items, as concealing information may be a significant component of agents' sales approach. The study asks how emerging markets with new investors might get excellent information on making financial decisions.

Due to the sheer amount of data now being produced within the insurance industry, there is an awareness that this has create a requirement for adequate technologies to effectively leverage this data for business benefit (Paruchuri, 2020). Machine learning can have a number of uses within the insurance industry including underwriting, fraud detection, entitlements management, and client capability.

# **Correlation Analysis**

Gogtay & Thatte (2017) list a number of considerations that should be taken into account when using correlation analysis. These include the limitations of correlation with repeated measures, the impact of outliers, the presence of non-linear relationships, potential for false correlations, and the importance of sample size.

In explaining the usefulness of linear correlation coefficient between two variables in order to find the multicollinearity of variables in a model, Senthilnathan (2019) also warns of how the interpretation of correlation must not be conflated to state that it incorrectly represents as causation effect. It is insisted that correlation only explores and indicates the type and degree association between variables, but does not explicitly explain the relationship between them or causal effect.

When conducting correlation analysis, appropriate consideration should be given to the size of the sample being used in order to ensure that the results of the correlation analysis are able to achieve the required minimum correlation coefficient value with adequate power and type I error or p-value (Bujang & Baharum, 2016).

# **Feature Importance**

Feature importance has been described as one of the most prevalent methods of explaining the way in which machine learning models behave (Saarela & Jauhiainen, 2021). Simple classification itself is not always the desired outcome, but rather knowing the importance of how specific features in a model can in some ways be explained, and furthermore how certain actions can either prevent or increase the likelihood of a certain classified outcome.

Gopagoni et al. (2020) evaluated important features and factors for better insurance sale conversion rates. The logistic regression model achieved a predictive accuracy of 84% and a cross-validation score of 81%. The SVM algorithm achieved a predictive accuracy of 80% accuracy. This provides valuable insights into the use of machine learning algorithms for selling insurance and the importance of selecting the right algorithm to improve the success rate of selling campaigns in insurance.

More recently, Merikanto (2022) focused on developing machine learning models to predict which customer attributes affected purchase decisions. Separate machine learning models were created for each product using LightGBM. Merikanto found that the models had high accuracy, with one product model having an almost 80% accuracy for predicting whether a customer would purchase the product or not.

RemOve And Retrain (ROAR) was developed as a benchmark to provide a way to evaluate the accuracy of feature importance identification methods in deep learning networks (Hooker et al., 2018). One interesting finding in this piece of research was that there were some instances a number of feature importance estimators were found to be less accurate at identifying feature importance than randomly assigning feature importance values the features in a dataset. This only further highlights the importance of using the correct and most suitable feature importance detection method.

Relative Feature Importance (RFI) has been said to grant a more nuanced approach to calculating feature importance (Konig et al., 2021). RFI has the ability to calculate the importance of a given feature relative to any other subset possible of features present in the dataset. There is, however, more research and development needed in relation to the challenges presented in sampling from unknown continuous variables and in using RFI on datasets with high dimensionality.

Wojtas & Chen (2020) introduced the concept of a dual-net architecture, where an operator and a selector work collectively in order to identify an optimal feature importance ranking and feature subset for feature importance ranking and subset selection for the purposes of deep learning. This method was used to develop an algorithm that learns by training both nets concurrently, resulting in an approach out-performs many of the best-in-class methods of ranking feature importance and feature selection.

Another proposed method of feature selection is the use of a feature selection algorithm called Dynamic Feature Importance based Feature Selection (DFIFS) (Wei et al., 2020). In addition to this, DFIFS can be used along with a traditional filter to create an algorithm known as Modified-Dynamic Feature Importance based Feature Selection (M-DFIFS). After applying it to 14 different high dimensional datasets, Wei et al. concluded that M-DFIFS performed better in relation to computational time and accuracy in comparison to a range of other feature selection algorithms.

When adopting feature selection in classification problems, classifier specific and classifier agnostic methods should be considered in order to calculate feature importance ranks (Rajbahadur et al., 2022). However, the limitations of using these should be kept in mind, as the results of using these measures do not always produce results that agree with each other. For example, where classification dataset consists of a high number of features that interact with each other, this can have an impact on how classifier specific feature importance methods performs, but not on the classifier agnostic method.

Zhou & Hooker (2021) discussed how bias has an impact in relation to split-improvement variable importance measures in tree-based methods, particularly Random Forest. This bias can be seen in the way split-improvement variable importance measures often give too much weight to features with more potential splits, leading to a skew in the derived rankings of feature importance. However, it was shown how this issue can be addressed by incorporating the split-improvement measured on out-of-sample data in order to correct the bias.

Using the Random Forest algorithm along with correlated predictors as a method of feature selection was evaluated by Gregorutti et al. (2016). They highlighted that high dimensionality in a dataset can be seen as a limitation for this type of approach in both classification and regression frameworks. However, they concluded that the use of Recursive Feature Elimination (RFE) can aid the performance of Random Forest when used for feature selection purposes.

Greenwell et al. (2018) proposed the development of a model-based approach to deriving feature importance that could be used with any supervised machine learning model. This approach would attempt to identify the level of interaction between variables by taking the effect of all features into account, and would then result in the same interpretation regardless of the selected supervised machine learning algorithm. However, there could be limitations to this approach, such as the impact of outliers and computational resources required to deal with large datasets.

# **Feature Selection**

Feature selection has been found to be an effective way to reduce data dimensionality when pre-processing a dataset for machine learning (Li et al., 2017). When done correctly, feature selection can result in more comprehensive, simpler machine learning models. Li et al. state one area of concern to be that most feature selection models require the number of selected features to be specified prior to actually knowing the optimal number of features for the given problem. But the reward for getting it right is the reduction of noise within the dataset.

Chandrashekar & Sahin (2014) noted that comparisons between different types of feature selection methods can only be done when the methods are applied to the same dataset, and then based on the when comparing to baseline classification performance metrics, the most suitable feature selection approach can be chosen. They further conclude that a number of factors should be considered when making selecting a feature selection method, including simplicity, stability and classification accuracy. When applied correctly, feature selection can result in improved classification, enhanced generalisation and identification of noise.

Another benefit of feature selection and dimensionality reduction can be seen in how the these approaches help deal with the ‘curse of dimensionality’ by reducing noise and therefore helps models to avoid overfitting (Venkatesh & Anuradha, 2019). When reviewing feature selection methods, Venkatesh & Anuradha inferred that while wrapper methods are computationally more costly than filter-based methods, they are generally more accurate.

Huang et al. (2019) discuss how dimensionality reduction can be successful in identifying the essential characteristics needed in a dataset for machine learning purposes, particularly classification and clustering. However, it is warned that current dimensionality reduction techniques can lack efficiency due to their complex nature, particularly as the dimensionality of the data grows.

In a review of feature selection methods among datasets with a high level of dimensionality, it was observed that selecting the optimal feature selection method can be of great importance to improve the performance of machine learning algorithms, reduce the time needed for the model to learn, and increase the accuracy of the learning (Asir et al., 2016). While evaluating feature selection methods, it was found that subset-based methods were computationally inefficient, and therefore not suitable for high-dimensional data, while ranking methods showed improved generalisability as well more efficient computational performance. However, it was concluded that filter methods are ultimately the optimal choice for dealing with high-dimensional data as they require less computational power, and can perform better across classification algorithms.

To address high-dimensionality difficulties, Ke et al. (2017) proposed two novel techniques, Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which reduce the data size and number of features, respectively. They implemented these techniques in a new GBDT algorithm called LightGBM, and demonstrated its efficiency and scalability compared to other Gradient Boosted Decision Trees (GBDT) algorithms, such as XGBoost. The experimental results showed that LightGBM can achieve up to 20 times faster training time than conventional GBDT algorithms, while maintaining almost the same level of accuracy.

In a review of dimensionality reduction techniques, it was concluded that adopting Principal Component Analysis (PCA) along with machine learning algorithms generally produce in better results than machine learning alone when a dataset has a high level of dimensionality (Reddy et al., 2020). However, the review also determined that dimensionality reduction techniques should only be used where appropriate, such as datasets with high dimensionality, as applying dimensionality reduction techniques to datasets with low dimensionality resulted in poorer performance than using machine learning alone.

While there is much support for the benefits associated with feature selection, there are a number of limitations to be considered (Heinze & Dunkler, 2016). It can sometimes cause an unnecessary level of complication to analysis, while also invalidating statistical tools such as P-values and confidence intervals. Heinze & Dunkler further argue that expert knowledge is more valuable than over-complicated feature selection techniques.

# **Hyperparameter Optimisation**

Andonie (2019) highlighted the significance of hyperparameter optimization in machine learning models, as well as the need to utilise a combination of optimization, search space, and training time reduction strategies to identify the optimum hyperparameters. According to Andonie, there is no quantitative procedure for selecting the right hyperparameters for a specific dataset, and the selection is based on trial and error.

The advantages of adopting automated hyperparameter optimization (HPO) in model-based reinforcement learning (MBRL) was discussed by Zhang et al. (2021). Because MBRL algorithms are sophisticated and have many hyperparameters and architectural options, they are difficult to apply to new problems without significant human input. Zhang et al. demonstrated that automated HPO can greatly outperform human tuning, and that dynamically tweaking hyperparameters during training can further increase performance. The trials shed light on the influence of various hyperparameters on training stability and the subsequent rewards.

Franceschi et al. (2017) investigated two methods (reverse-mode and forward-mode) for calculating the gradient of the validation error with regard to the hyperparameters of any iterative learning algorithm. The reverse-mode technique is related to past work but does not require reversible dynamics, whereas the forward-mode procedure is appropriate for real-time hyperparameter updates, which can accelerate hyperparameter optimization on big datasets. Franceschi et al. referred to research on data cleaning and learning task interactions and demonstrated that if the number of hyperparameters is minimal, forward-mode computing may be preferred to reverse-mode computation.

Yang & Shami, (2020) explored the significance of hyperparameter optimization in machine learning and proposed several cutting-edge optimisation approaches for common machine learning models. It also examined the performance of various optimisation approaches using benchmark datasets. According to Yang & Shami, if randomly selected subsets are highly representative of the given dataset, BOHB were the best choice for optimising a machine learning model, while BO models were recommended for small hyperparameter configuration space and PSO was the best choice for large configuration space.

Using Bayesian optimization, Joy et al. (2016) presented a novel paradigm for hyperparameter tuning on big data. The method separates large amounts of data into smaller chunks and uses typical Bayesian optimisation to build hyperparameter configurations for each chunk in parallel. Using a transfer learning configuration, the knowledge collected from the chunks is then used to tune the hyperparameters for the entire big dataset. The suggested method outperforms state-of-the-art hyperparameter tuning methods with less computing time when tested on two machine learning algorithms and two real-world datasets.

The importance of hyperparameters in machine learning algorithms and the difficulties in optimising them were discussed by Wu et al. (2019). To characterise the problem as an optimisation problem, the Wu et al. offered a hyperparameter tuning approach based on Bayesian optimisation and Gaussian processes. The approach proved to be effective in discovering the appropriate hyperparameters for frequently used machine learning models such as random forest and neural networks while drastically lowering runtime when compared to manual search.

Using 94 classification datasets from OpenML, Mantovani et al. (2018) analysed the effects of hyperparameter adjustment on three Decision Tree induction algorithims (CART, C4.5, and CTree). The goal was to assess the importance of hyperparameters and to identify the best optimisation approaches for hyperparameter tuning. According to the study, hyperparameter adjustment yielded statistically significant gains for C4.5 and CTree in only one-third of the datasets, and for CART in the majority of the datasets. The Irace approach was the best for all algorithms, and tweaking a specific small group of hyperparameters contributed the majority of the achievable ideal predictive performance.

The Random Forest (RF) algorithm is affected by a number of hyperparameters. While the default values are adequate, tweaking hyperparameters can increase performance, and the package tuneRanger automates this process using model-based optimisation (Probst et al., 2019). The number of trees should be set to a high value, and mtry is the most important hyperparameter. Sample and node sizes have a minimal impact but are worth adjusting. Large-scale comparison studies on hyperparameters and their impact on variable significance measures are lacking in the literature. Comparison studies are vital for evaluating and comparing the behaviours and performances of RF variations and hyperparameter choices. Although RF tuning can boost performance, the effect is smaller than that of other machine learning approaches. tuneRanger outperformed standard RF and other software that implements RF tuning.

Deep neural network (DNN) performance depends on hyperparameter optimization, and manual tuning can be time-consuming and inconvenient. Bayesian Optimization (BO)-based automated methods have been established, and Cho et al. (2020) analysed four strategies to improve BO for DNN hyperparameter optimization: diversification, early termination, parallelization, and cost function transformation. DEEP-BO, a simple yet resilient technique, outperformed well-known solutions on six DNN benchmarks. Research of the four techniques showed that diversity, conservative early termination, using partial training performance while parallelising, and heuristic cost function modification can all increase BO's performance. DEEP-BO performed at or near the top of all benchmarks examined.

# **Experimental Design**

The automatic tweaking of design flow parameters was presented by Xie et al. (2020) as a machine learning-based solution. The suggested approach makes use of approximation sampling and clustering approaches to boost tuning effectiveness and reuses feature extraction information from earlier designs. The method makes use of a XGBoost model and suggests a novel dynamic tree methodology to get around overfitting. When compared to earlier techniques, experimental results on benchmark circuits and two industrial designs reveal a considerable gain in design quality or decrease in sampling cost.

The usefulness of label-specific justifications for digits in a convolutional neural network representation was assessed by Ahern (2019). A 5-layer convolutional network was used in the experiment, which was run on MNIST, and it attained a test accuracy of 99.04%. Instead of evaluating a specific prediction on a specific image, the study assessed how well the explanations conveyed the critical properties for each digit in the dataset.

The effectiveness feature selection algorithms (FSAs) was evaluated by Molina et al. (2002). The experimental methodology was described in depth, and many experiment parameters were quantified. To evaluate the effectiveness of the FSAs, twelve families of data sets were created and examined. The FSAs were modified, and a filtering standard was developed to reduce their output to a subset of features. It was argued that all FSAs should have roughly the same possibilities to compete in terms of the computational resources in order to select which algorithm to utilise in specific circumstances.

Miao & Niu (2016 ) examined the most recent feature selection algorithms. They tested feature selection techniques on 12 publicly accessible datasets and evaluated the results using normalized mutual information and clustering accuracy. MaxVar, Laplacian Score, SPEC, SPFS-SFS, MCFS, UDFS, NDFS, and EUFS are a few of the algorithms. The experiment employed the K-means algorithm with numerous random initializations, and it presented the mean findings together with the standard deviation. The findings demonstrated that feature selection strategies are advantageous for machine learning tasks and enhance clustering performance.

In order to discover causal relationships between word characteristics and class labels in document classification, Paul (2017) suggested a matching strategy. The method seeks to find more significant and broadly applicable features than only correlational approaches. The study made use of datasets of reviews from the medical, film, and product industries. Results revealed that the suggested strategy, especially when used with non-domain data, significantly improves classification performance and identifies interpretable word connections with sentiment. Propensity score matching outperforms McNemar's test in two out of three datasets where feature selection is concerned, according to comparison of the two methods.

Imai et al. (2012) discussed the limitations of experiments in identifying causal mechanisms and proposed alternative experimental designs to overcome these limitations. The proposed designs involved manipulating the mediator variable and assuming that the manipulation does not directly affect the outcome. They emphasised the importance of identifying assumptions directly linked to experimental design and highlighted recent social science experiments to illustrate the proposed designs. It was expected that the designs would open up possibilities for identifying causal mechanisms through clever manipulations and future technological developments in various scientific disciplines, including social and medical sciences.

# **Validation**

While prediction models can be vital for decision-making and measuring performance, external validation is required in order to confirm the predictive accuracy of the model (Hickey & Blackstone, 2016). To carry out a suitably rigorous external validation study, a number of elements must be present, including appropriate study design, correct statistical methods, and clear and transparent reporting. Internal validation may not be sufficient to demonstrate predictive accuracy, and overfitting can lead to poor performance in external validation.

Adibi et al. (2020) gave warning on the importance of model validation, such that a lack of external validation among large amounts of data leads to many tests being untested and unvalidated, meaning that there could be a challenge to identify and select the most useful models. Fragmented efforts that assess only one model at a time do not allow for a reliable ranking of comparative performance.

Steyerberg & Harrell (2016) explored the significance of validation in predictive modelling, particularly highlighting the fact that model development studies are often not large enough, and that internal validation is of utmost importance, even more so than random split sample methods. They argue in favour of internal-external validation and direct tests for heterogeneity in predictor effects, concluding that fully independent external validation with data not available at the time of prediction model development is important.

Rahman et al. (2017) reviewed and evaluated a number of performance measures for external validation of prediction models. They recommended using Uno's concordance measure or Gönen and Heller's measure for quantifying concordance, Royston's D for assessing discrimination, and the calibration slope for assessing calibration. Also, investigating the characteristics of the validation data before choosing performance measures was recommended as a validation approach.

Accounting for competing events when developing and validating prediction models is also of great importance in model development (van Geloven et al., 2022). Failing to account for competing events can lead to overestimation of the cumulative incidence of an event of interest and distorted model performance. van Geloven et al. suggest methods of calculating and interpreting performance measures relating to the full risk distribution and a decision analytic perspective, consistent with TRIPOD guidelines for reporting prediction models. It is also noted that large sample sizes would generally be needed for a reliable performance assessment.

Debray et al. (2015) presented a framework for examining and improving the interpretation of prediction model external validation findings. By analysing their respective case-mix differences, the proposed methodological approach quantifies the degree of relatedness between development and validation samples on a scale spanning from reproducibility to transportability. The model's performance in the validation sample is evaluated and interpreted in light of case-mix changes, and the model is changed to the validation setting if necessary. The suggested framework improves the comprehension of results obtained during external validation of prediction models.

Validation Experiment Design Optimization (VEDO) method for prediction model design was proposed by Ao et al., (2017). This method was developed to maximise he information gain for model validation within the available testing constraints. In order to improve the robustness of the validation experiment design, a number of sources of uncertainty are included during the optimisation process.

Chen et al. (2007) provided a model validation strategy based on design, with the goal of increasing confidence in design decisions using a Bayesian prediction model. This method uses data from physical experiments and computer models to provide a framework for making predictions in the intended design domain. The proposal gives a fresh and enhanced perspective on model validation by connecting its definition to a specific design choice related to a specific design purpose, as well as direct estimations of the global influence of uncertainty sources on confidence in a design decision.

In the context of predictive modeling, Morrison et al. (2013) presented a systematic technique for splitting legacy data into calibration and validation sets, adopted from cross-validation. The approach is illustrated through an example using generated experiments of a nonlinear one degree-of-freedom oscillator. The proposed framework is broad in scope and can be used to a variety of challenges. The method is computationally intensive and needs to be improved.

Azpurua et al. (2014) discussed the significance of validation techniques in simulation tools for complicated situations, as well as the shortcomings of the standardized feature selected validation (FSV) method. By evaluating its shortcomings and complexities, it was hoped to uncover improvement opportunities to make FSV a more robust tool for data validation.

Parvandeh et al. (2020) explored how to utilise feature selection to increase machine learning model accuracy while avoiding overfitting. A consensus nested cross-validation (cnCV), a new approach that combines feature stability from differential privacy and nested cross-validation (nCV) were presented. The cnCV approach picks fewer features than nCV and has comparable accuracy to other methods such as private evaporative cooling (pEC). Parvandeh et al. compared these methods using simulated and real data and come to the conclusion that cnCV is an excellent and efficient way for combining feature selection and classification. The cnCV methodology can be combined with other feature selection and classification approaches, and it can handle overfitting by adjusting the threshold in the inner folds.

Misra & Yadav (2020) addressed how to use feature selection to increase the predictive accuracy of machine learning models. The Recursive Feature Elimination with Cross-Validation (RFECV) method was suggested and tested on a dataset using five distinct machine learning methods. According to the results, the simplest model, Logistic Regression, had the best accuracy. The study also implies that simpler models can outperform sophisticated models if the problem nature and appropriate feature selection strategies are thoroughly investigated. The research suggests that while feature selection is vital in enhancing the accuracy of ML models, the nature of the data, its quality, and volume should also be taken into account.

Demircioğlu (2021) examined how skewed results in radiomics datasets can be caused by poor feature selection prior to cross-validation. The researchers ran two experiments on ten publicly accessible radiomics datasets to assess the amount of bias introduced by feature selection prior to cross-validation. The findings revealed a significant positive bias, with higher dimensionality datasets more prone to overfitting. The study emphasized the necessity of avoiding data leakage and using feature selection correctly. The paper also analyses the effect of feature selection on classifier selection and compares the bias of various feature selection algorithms.

The need of validation in building robust multivariate models was discussed by Shi et al. (2018), as is the requirement for algorithms that can choose both minimal-optimal and all-relevant variables while effectively cross-validating. The MUVR algorithm used recursive variable elimination in a repeated double cross-validation procedure to uncover both minimal-optimal and all-relevant variables for regression, classification, and multilevel analysis. When compared to other methods, MUVR supported partial least squares and random forest modelling and has been found to provide prudent models with low overfitting and enhanced performance.

Cabitza et al. (2021) presented a meta-validation method for evaluating the reliability of external validation procedures for machine learning models. To inform the dependability of a validation approach, the suggested method takes dataset cardinality and similarity between training and validation sets into account. The methodology is demonstrated by validating a COVID-19 diagnostic model on 8 external validation sets. The validation datasets were determined to be adequate in terms of dataset cardinality and similarity, and the validated model reported good discrimination, usefulness, and calibration, implying that the results were sound. The research emphasizes the need of adequate external validation and presents a qualitative guideline for evaluating the reliability of validation techniques.

The bias and variance of model validation procedures has been investigated in the context of defect prediction models used by software quality assurance teams (Tantithamthavorn et al., 2017). The study discovered, through a case study of 18 systems, that single-repetition holdout validation produces estimates with greater bias and variation than the top-ranked model validation procedures, and advised out-of-sample bootstrap validation instead. The relevance of adopting an effective model validation technique as a major experimental design decision for accurate and reliable defect prediction was also emphasised.

The relevance of verifying predictive models was discussed by Ivanescu et al. (2015). It discussed why predictive validity decreases and presents metrics that are routinely used to estimate predictive validity. The research emphasises the need of reporting a model's projected loss of predictive power in new samples and gives methods for measuring and reporting validity shrinkage and predicted predictive validity. According to Ivanescu et al., future predictive modelling research should always report the projected decrease in predictive power of a model in new samples.

Steyerberg & Vergouwe (2014) presented a methodology for constructing and evaluating prediction models, with seven critical processes and four model performance measures: calibration-in-the-large, calibration slope, discrimination, and clinical applicability. They also explored model validation issues such as miscalibration and minor improvements in discrimination with additional markers, emphasising the significance of involving statistical specialists. The suggested approach aims to increase the methodological rigour and predictive model quality.

Ali & Gravin (2021) analysed various model validation methods for datasets containing software development effort estimation (SDEE) and software fault prediction (SFP). The study analysed estimate strategies' prediction accuracy and stability using ten different cross-validation (CV) and bootstrap validation methods. The results demonstrated that the model validation procedures that yield the best prediction accuracy are repeated 10-fold CV with SDEE data and optimistic boot with SFP data. The most stable model validation method for both SDEE and SFP datasets is repeated 5-fold CV. The study recommended employing model-agnostic methodologies to identify essential variables and instance-level interpretations to explain whether software systems are clean or flawed.

Adler & Painsky (2022) described a weakness in the commonly used Gradient Boosting Machines (GBM) technique that causes bias in its feature importance (FI) estimates due to the usage of decision trees that are biased towards categorical variables with large cardinalities. A cross-validated unbiased base learner framework (CVB) that addresses this issue and is effective in a variety of synthetic and real-world settings is proposed. According to the study, GBM FI is unique to each implementation, but CVB provides impartial FI without sacrificing generalization capabilities.

Altmann et al. (2010) highlighted the significance of interpretability in machine learning models and how linear models are frequently employed to evaluate feature relevance. However, it has been discovered that more complicated models, such as support vector machines and Random Forest (RF) models, have biased feature importance measurements. Altmann et al. offer a solution for normalizing feature significance measures in a non-informative context by using repeated permutations of the outcome vector to estimate the distribution of measured importance for each variable. This updated measure of feature importance enhanced model interpretability and is applicable to different learning methods.

# **Model Evaluation Metrics**

Bylinskii et al. (2019) analysed and suggested 8 distinct evaluation measures and their properties under specified assumptions and for specific applications. The research stated that the choice of metric is determined by the qualities of the inputs, and that multiple metrics may be required for different tasks and applications.

Zhang et al. (2020) discussed the significance of assessment metrics in batch evaluations of information retrieval (IR) systems. The findings provided suggestions for fine-tuning assessment metric parameters and promote the consistency of user behaviour modelling and satisfaction measurement.

The area under the receiver operating characteristic curve (AUC) is a typical measure of discrimination for binary outcome prediction models, but it has been criticized for its shortcomings. Under the assumption of multivariate normality, Pencina et al. (2012) analysed this claim by linking the AUC to clinical performance indicators based on sensitivity and specificity. They found that, unless where good specificity is required, the change in the AUC is an appropriate predictor of the change in clinical performance indicators. In such circumstances, the discrimination slope may be a more accurate predictor of model improvement than AUC. However, if the baseline model performs well, increasing the AUC may be more difficult. There are some limitations to the study, such as the assumption of multivariate normality, linear discriminant analysis, and the restricted number of clinical measurements and risk thresholds considered. Nonetheless, the study implies that reporting the AUC increment is fair because changes in the AUC are proportionate to changes in clinical measures of prediction performance. If clinically meaningful metrics can be discovered, they should also be reported.

Marcot (2012) presented a variety of existing and new metrics for evaluating the performance and uncertainty of Bayesian Network (BN) models, including metrics for conducting model sensitivity analysis, evaluating scenarios, depicting model complexity, assessing prediction performance, and evaluating model posterior probability distributions' uncertainty. Marcot emphasised the value of metrics in enhancing model credibility, acceptance, and suitable application. The research emphasised the significance of balancing model performance and prudence. In addition, the study advises that metrics be chosen early in the model-building process to avoid post-hoc selection bias, and that metrics of performance and uncertainty can be used to assist select the best model from a group of competing models in a multi-model approach.

Tian et al. (2016) suggested that a correct error model should be used instead of metrics to evaluate models. Traditional metrics are interdependent, imperfect, and incapable of accurately assessing uncertainty because they are based on linear, additive, Gaussian errors. A accurate error model, on the other hand, contains the entire error information, conveys the error structure more naturally, and explicitly quantifies uncertainty. The error modelling methodology applies to both linear and nonlinear errors, however the metrics only apply to linear errors. The error model contains all of the information needed to evaluate the prediction model and can be used to build the conditional distribution between the data and the reference.

# **Conclusions**

While supervised machine learning models have been used within the life insurance industry to predict purchase intention behaviour of customers, there has been little research into the effectiveness of supervised machine learning models to predict whether an existing life insurance application will eventually become an active policy. Also, while there has been some research into the features that contribute to the likelihood of a customer purchasing non-life insurances, the same level of knowledge and research is not present in relation to life insurance. By satisfying the proposed research objective, this study can address these research gaps, while also taking into account previously researched best practices in machine learning in insurance, correlation analysis, feature importance, feature selection, hyperparameter optimisation, experimental design, and model evaluation metrics.

This literature review provided useful insights into the significance of several elements connected to the usage of machine learning techniques in the insurance industry. The research examined have shown that machine learning algorithms are useful at accurately forecasting client purchase behaviour and handling classification challenges in the insurance industry. The CoIL Challenge 2000, in particular, demonstrated the high accuracy rates attained by machine learning algorithms in forecasting insurance policy purchase while taking demographic data and insurance product characteristics into account.

The importance of feature selection and feature engineering has emerged as a significant subject in the literature. Techniques like NCA, SFS, SBS, and different data pre-processing approaches have proved helpful in resolving issues like uneven class distributions and finding characteristics that impact the likelihood of purchasing insurance policies. Furthermore, the Decision Tree algorithm and its variants, such as Random Forest, have shown continuous effectiveness in forecasting customer purchase intention, offering interpretable insights into the elements driving customer behaviour.

Furthermore, the literature review demonstrated the efficacy of machine learning approaches in various classification tasks in the insurance industry, such as churn prediction, claim prediction, and fraud detection. Ensemble approaches, such as Random Forest and Deep Learning models, have showed promise in terms of accuracy and interpretability, driving predictive modelling in insurance forward. However, the research has stressed the significance of ethical issues and fairness in machine learning applications, notably in the insurance industry, to ensure that predictive models do not perpetuate prejudices or discriminate against certain groups.

Moreover, the literature review emphasised the need of validation in predictive modelling and machine learning. External validation, in addition to internal validation, is required to check the predicted accuracy and resilience of models. For trustworthy comparative performance assessment and assuring the generalisability and repeatability of predictive models, rigorous validation techniques and transparent reporting are essential. To improve the reliability of model validation, many validation strategies such as heterogeneity testing, performance metrics, and innovative frameworks have been investigated.

Finally, the literature review stressed the significance of adopting proper evaluation criteria for model evaluation. Different assessment measures have different qualities and may be better suited to different tasks and applications. The assessment metrics used are determined by the features of the inputs, the model's aims, and the environment in which the model will be implemented. To guarantee a complete assessment of model performance and to make informed judgments in model creation and selection, researchers must carefully evaluate these elements.

In conclusion, the literature research highlighted some critical elements of the usage of machine learning techniques in the insurance business. The results of the research examined give a basis for selecting relevant algorithms, feature selection strategies, and data pre-processing procedures, eventually improving the accuracy and efficiency of predictive models. Furthermore, the research emphasises the need of ethical issues, fairness, validation procedures, and proper assessment criteria in assuring predictive model reliability and generalisability in insurance applications. Future research attempts can use these insights to expand the area of data analytics in insurance and contribute to improved decision-making processes and consumer satisfaction.

# **Methodology**

**Data Collection**

The study methodology includes both descriptive and analytical components. Descriptive statistics are used to describe and illustrate the features of the life assurance applications, while analytical approaches are used to look for patterns, correlations, and trends in the dataset.

The data was provided by a life insurance business and covers all applications submitted between 2017 and 2022. The researcher worked with the life insurance company to extract pertinent data from their records. This information included product details, application dates, application statuses, and any other pertinent elements.

Because the full information over a six-year period is available, it gives a comprehensive perspective of all applications and enables for in-depth study. With such comprehensive data, there is less worry about sampling error or misrepresentation of the data.

The dataset consists of diverse features related to insurance policies, including Product, ProductGroup, ProductType, Agency, WorkflowStatus, Indexation, NoOfLives, CommDateProvided, PaymentFreq, UWDecision, ComissionSacrifice, CommissionSacrificeType, RenewalSacrificeType, CommissionTerms, Discount, BonusCommission, FreeCover, SeriousIllnessType, and SignedDecReceived. PolicyIssued is the variable of interest.

**Data Cleansing**

This section describes the methods used for data cleansing, with a particular emphasis on finding and handling outliers within the dataset. Outliers are data points that differ greatly from the overall trend of the data, causing statistical analysis and model performance to be distorted. The data cleaning procedure is critical for ensuring the accuracy and dependability of the following analysis.

The first stage requires preparing the dataset. For the sake of this research, columns having the data type 'int64' are considered categorical characteristics. Following that, these categorical columns are transformed to the ‘category’ data type. This conversion not only saves memory but also allows for more efficient categorical data handling.

Then, by choosing columns with numeric data types, numerical properties are segregated. For each numeric parameter, the Median Absolute Deviation (MAD), a robust measure of data variability, is determined. MAD gives a more trustworthy assessment of data dispersion than traditional metrics such as standard deviation and is less susceptible to outliers.

A threshold multiplier (denoted as ‘k’) is chosen to identify probable outliers. In this investigation, a multiplier of three is used. The threshold for each characteristic is calculated by multiplying the MAD by the multiplier of choice, establishing a standard for finding data points that differ considerably from the norm.

A comparison method is used to identify outliers. Individual data points’ absolute variances from their respective attribute medians are compared to a predetermined threshold. The result is a binary matrix that highlights the existence (True) or absence (False) of outliers for each data point.

Certain characteristics, most notably CommissionSacrificePercentage and BonusCommissionPercentage, have been identified as potentially outliers. The np.log1p function is used to perform a logarithmic transformation on these properties. This change reduces the influence of extreme values and brings them closer to the middle of the distribution.

It is critical to quantify the amount of outliers within each characteristic. The total number of outliers for each characteristic is calculated by adding the binary outlier matrix along the rows. This provides a thorough view of the distribution of outlier occurrences in the dataset.

By systematically applying this data cleansing methodology, the research guarantees a robust and accurate foundation for the subsequent stages of data analysis and modelling

**Data Exploration**

This section explains the process used for data exploration, which includes strategies for discovering patterns, correlations, and insights within the dataset. Data exploration is the first stage in understanding the intrinsic structure of the data, identifying trends, and informing future studies and decision-making.

Descriptive statistics are produced using the numeric\_data.describe() method to acquire a basic overview of the dataset. This produces important statistical measures including mean, median, standard deviation, and quartiles, which provide insight into the central tendency and dispersion of numerical variables. In addition, preliminary observations on the data’s features are made. The preponderance of zero values in the CommissionSacrificePercentage field, for example, indicates that the majority of records do not include commission sacrifice.

Understanding data patterns requires visualizing the distribution and change of important variables. To show the distribution of CommissionSacrificePercentage and BonusCommissionPercentage side by side, box plots and violin plots are constructed. The box plot shows the quartiles and outliers, but the violin plot shows the distribution's form in greater detail. These plots graphically represent the variability and range of the variables, assisting in the detection and comprehension of potential outliers.

Density plots and histograms can reveal information about the distribution of numerical data. Density plots provide the data’s estimated probability density function, whereas histograms show data frequency in bins. For each numeric variable, a density map is created to visually analyse the underlying distribution and potential multimodality. Histograms augment this evaluation by displaying the frequency of data points inside predetermined bins.

Quantile-Quantile (Q-Q) plots are used to determine if data follows a given theoretical distribution, most often the normal distribution. Deviations from the predicted distribution are shown by comparing the actual data quantiles to those of a theoretical distribution. The Q-Q plot of each numeric variable is constructed to examine its deviation from normalcy.

Temporal patterns are critical for understanding data dynamics throughout time. The dataset’s temporal variable PropDate is examined in terms of several dimensions such as year, month, and day of the week. Line charts, bar charts, and other suitable approaches are used to depict aggregated data. This investigation reveals probable seasonality, patterns, or variations in data behaviour across various time intervals.

Categorical variables provide crucial information into the dataset's properties. The process comprises of a number of steps. The steps include taking relevant elements from PropDate, such as the day of the week, day of the month, month, and year, and changing categorical columns to suitable data types (categories or integers). To understand the composition of categorical variables, calculate and illustrate their frequency distribution.

There are cross-tabulations between category variables and the target variable PolicyIssued. These tables provide insights into the relationship between variables and aid in the identification of patterns. The chi-square test evaluates the independence of categorical variables and the target variable, assessing if actual and predicted frequencies differ considerably. Bar plots and heatmaps are used to show correlations and patterns in cross-tabulations.

The strength of correlations between pairs of categorical variables is determined by Cramer's V, a measure of association for categorical variables. This demonstrates the extent to which variables are dependent on one another beyond the reported frequencies. To show correlations among categorical variables, a matrix of Cramer's V values is produced and presented as a heatmap.

Time series analysis investigates patterns and trends in temporal data. Techniques include categorising data by time periods (e.g., monthly, quarterly), generating aggregated statistics, and showing patterns using line charts, bar charts, or other appropriate ways. This study aids in the discovery of insights connected to cyclic activity or long-term trends in data.

Correlation analysis investigates the connections between numerical variables. To understand the strength and direction of relationships, correlation matrices are constructed, shown via heatmaps, and analysed. The emphasis is on identifying variables that are highly correlated with the target variable and with one another, indicating possible predictive power or multicollinearity problems.

Patterns within categorical variables are investigated to learn more about their distribution and relevance to the target variable. Techniques include making stacked bar charts, calculating proportions by category, and investigating how various circumstances influence the chance of a desired outcome.

**Feature Correlation**

A cross-sectional research design is used in this study in order to evaluate the impact of the independent features’ correlation with the dependent variable and the impact of feature importance score to determine the variables that have the greatest impact on the conversion of life assurance applications. Data is gathered on a variety of categorical factors and the relationships between these variables and the outcome variable PolicyIssued is explored.

The methodology of the study is to look at the associations between categorical factors and the binary outcome variable PolicyIssued, which signifies whether or not a policy was issued. The purpose of this is to understand how various categorical factors are correlated to the likelihood of policy issuance.

Chi-square tests are used to determine the relationship between categorical variables. The observed frequencies of data in a contingency table are compared to the predicted frequencies if the variables were independent. The Chi-square tests are performed using the PolicyIssued variable for each categorical variable being analysed (e.g., Product, ProductGroup, etc.) to evaluate if there is a significant correlation. For each test, chi-square statistics and p-values are provided. The Chi-square statistic quantifies the strength of the relationship, whilst the p-value indicates its statistical significance.

The analysis of the study focuses on understanding Chi-square statistics and p-values. The Chi-square statistic values reveal the degree of association and correlation between categorical factors and policy issuance. Higher Chi-square values indicate stronger relationships. The p-values represent the possibility that the observed link may have happened by chance. Low p-values, close to zero, indicate a significant correlation.

**Feature Importance**

This section describes the methods used to identify and evaluate feature relevance in the context of policy issuance prediction. Using machine learning techniques and statistical testing, the investigation tries to improve knowledge of the influence of various factors on policy issuance prediction.

To measure initial feature relevance, a Random Forest classifier with 100 estimators is trained on the training data. The Gini impurity measure is used to calculate feature significance, which is then sorted in descending order. The top-N characteristics are chosen for further investigation based on their significance rankings.

A number of experiments are carried out to better understand the impact of different feature combinations on model performance. The Random Forest classifier is trained and evaluated in these trials with varied numbers of chosen features (ranging from 1 to N). To evaluate model performance, classification accuracy and Receiver Operating Characteristic Area Under the Curve (AUC-ROC) are evaluated for each experiment, and a thorough classification report is created.

Experiments are carried out to evaluate the influence of hyperparameters on feature significance evaluation by altering the test size and the number of estimators in the Random Forest classifier. There are test sizes of 20% and 30% as well as n\_estimators values of 50, 100, 150, and 200. For each configuration, feature importance, classification accuracy and AUC-ROC are recorded.

A chi-square test is used to assess the independence of feature pairs for categorical characteristics. This test examines if the presence of one categorical variable is contingent on the presence of another. Cramer's V statistic is used to determine the degree of association between categorical characteristics and the target variable.

To resolve class imbalance, two resampling strategies, BorderlineSMOTE and SMOTE, are used to produce synthetic minority class samples. To undersample the majority class, RandomUnderSampler is utilised. The resampled data is used to train Random Forest, Gradient Boosting, and LightGBM classifiers. To identify an appropriate collection of features for each classifier, Recursive Feature Elimination with Cross-Validation (RFECV) is used.

The methodology described above is a systematic approach to assessing feature importance and its impact for policy issuance prediction. This work intends to give important insights into the relevance of individual variables in the context of insurance policy issuance prediction by employing a combination of machine learning models, resampling approaches, statistical tests, and visualisation tools.

**Feature Engineering**

This section describes the methodology used to develop and modify features for improving machine learning models' prediction capabilities. Feature engineering is critical in converting raw data into useful representations that allow for efficient model learning. A systematic strategy is used in this work to preprocess categorical data, build composite features, select important qualities, and evaluate their influence on model performance.

Categorical features in the dataset are transformed to string data type before to feature engineering. This guarantees that categorical data is treated consistently and minimises any errors during further processing. Using the .astype(str) method, the categorical columns WorkflowStatus, UWDecision, Product, ProductGroup, ProductType, CommissionSacrificeType, RenewalSacrificeType, CommDateProvided, and FreeCover are converted into strings.

Concatenating particular columns within the dataset yields composite features. These combined variables, which include names like Combined\_Status1, Combined\_Status2, Combined\_Status3, and Combined\_Status4, are designed to record subtle interactions between qualities. The concatenation of relevant features aims to generate more useful categorical variables that encompass varied data dimensions.

Label encoding is used to make it easier for machine learning algorithms to use categorical data. To transform the composite category features into numerical labels, the LabelEncoder module is used. This transformation ensures that algorithms that require numerical input are compatible. The newly created features, WorkflowStatus\_UWDecision, Product\_ProductGroup\_ProductType, CommissionSacrificeType\_RenewalSacrificeType, and CommDateProvided\_FreeCover, will be label encoded for further analysis.

The original categorical columns and composite features are removed from the dataset as part of the feature engineering process. Because the altered features include the necessary information collected from the original attributes, this procedure is performed after label encoding.

The StandardScaler is used to normalise continuous numerical features before implementing machine learning methods. This normalisation ensures that all characteristics contribute equally to the learning process and avoids variables with greater magnitudes from dominating.

Model-specific feature selection approaches are used to discover the most significant characteristics for predictive modelling. This study employs three well-known classifiers: Random Forest, Gradient Boosting, and LightGBM. Each classifier is trained using a preprocessed and scaled dataset with different hyperparameter settings, such as test sizes and estimator counts.

RFECV (Recursive Feature Elimination with Cross-Validation) is used for each classifier to repeatedly pick features based on their influence on model performance. This technique assists in identifying the ideal selection of qualities that significantly contribute to correct forecasts. The RFECV approach reveals the most important characteristics, improving model interpretability and efficiency.

Visualisation and statistical analysis are used to assess the efficiency of engineered features and feature selection procedures. Heatmaps are used to show the effect of changing hyperparameters on model accuracy, allowing optimal configurations to be identified. Additionally, feature importance statistics such as minimum, maximum, mean, and median importance scores are produced. These statistics give a well-informed view of the consistency and relevance of properties across multiple classifiers.

This section’s methodology explains a systematic approach to feature engineering, from preprocessing and composite feature development through label encoding, model-specific selection, and assessment. This comprehenive technique aims to improve the performance of machine learning models by identifying and utilising significant and influential features.

**Machine Learning on Original Imbalanced Dataset**

This section describes the methodology used to conduct the experimental evaluation of several machine learning algorithms for predicting the PolicyIssued outcome. The section outlines the methods taken to assure the study's integrity and validity.

This study's research design entails a thorough examination of machine learning algorithms utilising real-world insurance data. The primary goal is to assess the predictive power of the Logistic Regression, Stochastic Gradient Descent (SGD) Classifier, Decision Tree, Random Forest, LightGBM and Neural Networks algorithms.

The dataset has several characteristics that are related to policy application data. For the predictive modelling task, the dataset is separated into features (X) and the target variable (y). The relevance of machine learning algorithms to the research objective and their prevalence in insurance-related prediction tasks are used to choose them. Because of their interpretability, efficiency, and variety, the Logistic Regression, SGD Classifier, Decision Tree, Random Forest, LightGBM and Neural Networks were chosen.

The dataset is divided into training and test sets using various proportions (0.2 and 0.3) of the data to provide a rigorous assessment of model generalisation. To preserve the distribution of the target variable across splits, stratified sampling is used. The training dataset is used to train the relevant model for each method. Models for Logistic Regression and SGD Classifier are fitted to data using gradient-based optimization approaches. Ensemble learning methods are used to capture complicated interactions in Decision Tree, Random Forest, and LightGBM. The models are assessed on both the training and test sets after training. Accuracy and Area Under the Receiver Operating Characteristic Curve (AUC) are performance measures computed. Confusion matrices and classification reports are also created to acquire a better understanding of the models' strengths and weaknesses. All of the Decision Tree, Random Forest and LightGBM methods use k-fold cross-validation to improve the robustness of the findings. The dataset is divided into several subsets, and the models are trained and assessed iteratively on each subset, with the average performance metrics derived over folds. The results are thoroughly examined and reported. To show how algorithm performance fluctuates over different test sizes and cross-validation folds, bar graphs and heatmaps are used. This provides a more complete picture of the algorithms' behaviour.

To predict policy issuance, a neural network model built with the Keras framework is used. The target variable and related attributes are extracted. Using a test size of 0.2, the data is divided into training and test sets. The neural network is made up of two layers: a densely linked hidden layer with ReLU activation and an output layer with sigmoid activation. To analyse the neural network's convergence behaviour, it is trained over a number of epochs (10, 20, and 30). Following each training, the model's predictions on both the training and test datasets are assessed in terms of AUC and accuracy scores.

**Machine Learning on Rebalanced Undersampled Dataset**

The methods used to perform a detailed examination of predictive modelling strategies for the PolicyIssued classification problem is presented in this section. The dataset was preprocessed, separated into training and testing sets, and exposed to a variety of classification techniques such as Logistic Regression, Stochastic Gradient Descent (SGD) Classifier, Decision Tree, Random Forest, and LightGBM. Furthermore, k-fold cross-validation was used to evaluate the models' generalisation capability.

Initially, the dataset was divided into the feature matrix (X) and the target variable (y). Except for the target variable, PolicyIssued, the feature matrix comprised all variables. The data was then separated into two subsets: an 80-20 split (X\_train\_80, X\_test\_80, y\_train\_80, y\_test\_80) and a 70-30 split (X\_train\_70, X\_test\_70, y\_train\_70, y\_test\_70) using the scikit-learn library's "train\_test\_split" function. The RandomUnderSampler approach was applied to the training data to construct balanced subsets (X\_resampled\_80, y\_resampled\_80) and (X\_resampled\_70, y\_resampled\_70).

The first classification algorithm was Logistic Regression. The model was trained on resampled training data (X\_train, y\_train) and tested on training and testing datasets. AUC, accuracy, confusion matrix, and classification report were all computed. These parameters were used to evaluate the model's performance on both training and testing datasets. The Stochastic Gradient Descent (SGD) Classifier, like Logistic Regression, was trained and assessed using resampled data. AUC and accuracy metrics were generated for both the training and testing datasets. The Decision Tree and Random Forest algorithms were trained on resampled training data and their performance was assessed using the same set of criteria as previously described. The LightGBM Classifier, a gradient boosting framework, was similarly trained and assessed like the other methods. The predictive performance of the model was evaluated using AUC and accuracy measures on both training and testing data.

K-fold cross-validation was used to evaluate the models' robustness and generalisability. For training and validation, the dataset was partitioned into subsets (folds). Across tests, the number of folds varied (5, 10, and 20 folds). The models were trained and assessed within each fold, and the average AUC and accuracy were obtained across all folds. Based on the performance metrics acquired from individual model training, testing, and k-fold cross-validation, a comparison analysis was carried out. The measurements revealed information about each algorithm's capacity to distinguish between classes and generalise to new data.

A neural network model is used to detect underlying patterns in the data. To predict binary outcomes, the architecture consists of an input layer with a ReLU activation function, a hidden layer with a ReLU activation function, and an output layer with a sigmoid activation function. The model is built and trained for each resampled dataset using a different number of epochs. The Adam optimizer and binary cross-entropy loss function are used to build the model. Backpropagation is used to iteratively update the model's weights based on training data throughout the training phase. To assess the model's performance during training and minimise overfitting, the training data is further separated into training and validation subsets.

The model's performance is evaluated after each training session using test data that is kept separate from the training process. The accuracy and loss of the model on the test set are computed and printed. In addition, the area under the AUC-ROC is calculated to assess the model's prediction abilities. The evaluation process is repeated for a set number of epochs, namely 10, 20, and 30. This enables for a thorough examination of how the model's performance varies as the number of training iterations increases. For each epoch, the metrics of accuracy, loss, and AUC-ROC are recorded and displayed in the results. The results of several model assessments are examined to identify the best number of epochs for each resampled dataset. To determine the model's stability and generalisation capabilities, the research considers changes in accuracy, loss, and AUC-ROC measures over multiple epochs. Furthermore, the two resampled datasets are compared to determine any potential trends or changes in model performance due to the differing dataset compositions.

**Machine Learning on Rebalanced Oversampled Dataset**

The original dataset is divided into characteristics (X) and the target variable (y), PolicyIssued. The data is separated into two sets: one with 80% training and 20% testing and another with 70% training and 30% testing. This is accomplished through the use of the train\_test\_split function. Furthermore, the features are scaled using the StandardScaler to assure feature scaling consistency. To address the uneven class distribution in the dataset, the Synthetic Minority Over-sampling Technique (SMOTE) is used. To achieve a balanced class distribution, SMOTE is used on the training data to oversample the minority class. The 80% and 70% training datasets are resampled separately.

On the resampled 80% training data, a Logistic Regression model is trained. The performance of the model is assessed on both the training and test datasets. The area under the ROC curve (AUC) and accuracy are calculated. For both the training and test datasets, the confusion matrix and classification report are created. The Stochastic Gradient Descent (SGD) Classifier, like Logistic Regression, was trained and assessed using resampled data. AUC and accuracy metrics were generated for both the training and testing datasets. The Decision Tree and Random Forest algorithms were trained on resampled training data and their performance was assessed using the same set of criteria as previously described. The LightGBM Classifier, a gradient boosting framework, was similarly trained and assessed like the other methods. The predictive performance of the model was evaluated using AUC and accuracy measures on both training and testing data.

A neural network model is used to detect underlying patterns in the data. To predict binary outcomes, the architecture consists of an input layer with a ReLU activation function, a hidden layer with a ReLU activation function, and an output layer with a sigmoid activation function. The model is built and trained for each resampled dataset using a different number of epochs. The Adam optimizer and binary cross-entropy loss function are used to build the model. Backpropagation is used to iteratively update the model's weights based on training data throughout the training phase. To assess the model's performance during training and minimise overfitting, the training data is further separated into training and validation subsets.

**Machine Learning using only combinations Top-N features**

The dataset contains numerous characteristics associated with insurance applications. The columns that will be used for each iteration are predefined as feature\_sets. These feature sets are chosen to investigate various predictor combinations from the already identified most important features. The dataset is divided into feature variables X and the target variable y, which denotes whether a policy was issued, for each iteration.

The initial step in each cycle is to train a Logistic Regression model. StandardScaler is used to scale the feature variables. Due to the imbalance in the dataset, SMOTE is used to oversample the minority class in the training data. After that, the resampled data is utilised to train the Logistic Regression model. The model is tested using evaluation measures such as AUC and accuracy on both the training and test datasets. The same process is carried out using SGD Classifier, Decision Tree, Random Forest, and LightGBM classifiers.

k-Fold Cross-Validation is used to evaluate the models' generalisation performances. To assess the robustness of the models, several values of k (5, 10, and 20) are investigated. The data is partitioned into k subsets for each iteration for each k, and training and validation are conducted repeatedly. To offer a more representative performance measure for each model configuration, the evaluation metrics, including AUC and accuracy, are computed for each fold and averaged.

A neural network model is used to detect underlying patterns in the data. To predict binary outcomes, the architecture consists of an input layer with a ReLU activation function, a hidden layer with a ReLU activation function, and an output layer with a sigmoid activation function. The model is built and trained for each resampled dataset using a different number of epochs. The Adam optimizer and binary cross-entropy loss function are used to build the model. Backpropagation is used to iteratively update the model's weights based on training data throughout the training phase. To assess the model's performance during training and minimise overfitting, the training data is further separated into training and validation subsets.

The approach given in this part enables a thorough evaluation of multiple machine learning models utilizing various feature sets. The use of Logistic Regression, SGD Classifier, and Decision Tree Classifier, Random Forest, LightGBM, and Neural Networks in conjunction with k-Fold Cross-Validation, allows for a complete evaluation of the models' prediction capabilities. The assessment metrics that arise provide insights into the models' efficacy in predicting policy issuance and provide suggestions on which model configurations and features enable the greatest performance. The same experimentation using the combinations of the top-N features is also carried out on the undersampled datasets.

**Machine Learning using Engineered features**

Feature Engineering is carried out as outlined above. To iteratively evaluate model performance with different feature sets, specific columns are selected for each iteration. Multiple feature sets are constructed, starting with a minimal set of features and progressively incorporating additional ones. The feature sets include combinations of Agency, WorkflowStatus\_UWDecision, CommDateProvided\_FreeCover, Product\_ProductGroup\_ProductType, CommissionSacrificePercentage, BonusCommissionPercentage, and SignedDecReceived.

Logistic regression models are trained and assessed for each feature set. The dataset is divided into training and testing subsets, and the features are scaled using the StandardScaler. On the training data, SMOTE is used to correct class imbalance. The logistic regression model is then trained on the resampled data and assessed on both the training and test sets using measures such as AUC and accuracy. The same process is carried out using SGD Classifier, Decision Tree, Random Forest, and LightGBM classifiers.

k-Fold Cross-Validation is used to evaluate the models' generalisation performances. To assess the robustness of the models, several values of k (5, 10, and 20) are investigated. The data is partitioned into k subsets for each iteration for each k, and training and validation are conducted repeatedly. To offer a more representative performance measure for each model configuration, the evaluation metrics, including AUC and accuracy, are computed for each fold and averaged.

A neural network model is used to detect underlying patterns in the data. To predict binary outcomes, the architecture consists of an input layer with a ReLU activation function, a hidden layer with a ReLU activation function, and an output layer with a sigmoid activation function. The model is built and trained for each resampled dataset using a different number of epochs. The Adam optimizer and binary cross-entropy loss function are used to build the model. Backpropagation is used to iteratively update the model's weights based on training data throughout the training phase. To assess the model's performance during training and minimise overfitting, the training data is further separated into training and validation subsets.

**Hyperparameter tuning**

To ensure uniform scaling, the features are standardised using the StandardScaler prior to model training. Because the dataset is imbalanced, SMOTE is used to balance the class distribution in the training data. As a consequence, an expanded training dataset with balanced class proportions is produced.

The performance of the Decision Tree Classifier is assessed first in the evaluation process. The feature sets are iterated, with each holding a unique collection of features. Stratified sampling is used to divide the dataset into training and testing sets for each feature set. GridSearchCV is used for hyperparameter tuning, which investigates various combinations of hyperparameters such as max\_depth, min\_samples\_split, min\_samples\_leaf, max\_features, and criterion. The assessment employs k-fold cross-validation, including scoring criteria such as ROC AUC and accuracy.

Following that, the Random Forest Classifier is evaluated in a similar manner. Iterating over feature sets, dividing the data, scaling features, applying SMOTE, and performing hyperparameter tweaking are all part of the assessment. Tuning hyperparameters include n\_estimators and bootstrap.

The optimum hyperparameters are found for each combination of feature set, scoring metric, and number of folds, and the model with these hyperparameters is trained on resampled training data. The performance of the trained model is then tested on the test dataset. Metrics for evaluation include ROC AUC, accuracy, and the highest scores acquired during hyperparameter tuning.

The evaluation results are displayed for each feature set, scoring metric, and number of folds combination, offering insights into the models' performance. The methodology evaluates decision tree and random forest classifiers across different circumstances, making it easier to choose the best model configuration for predicting life insurance policy issuance.

# **Results**

This section presents the findings from the evaluation of the supervised machine learning classifiers on the life insurance policy issuance dataset. Tables, figures, and graphs are used to show the findings, which give a visual depiction of the performance indicators across a range of experiments.

**Original Imbalanced Dataset**

**Correlation & Feature Importance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Cramer's V** | **Chi2 Statistic** | **P-value** |
| PolicyIssued vs. Product | 0.12 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. ProductGroup | 0.0963 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. ProductType | 0.0953 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. Agency | 0.2151 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. WorkflowStatus | 0.4079 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. Indexation | 0.0381 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. NoOfLives | 0.07 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. CommDateProvided | 0.2193 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. PaymentFreq | 0.0077 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. UWDecision | 0.3466 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. ComissionSacrifice | 0.0199 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. CommissionSacrificeType | 0.0202 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. RenewalSacrificeType | 0.0151 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. CommissionTerms | 0.001 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. Discount | 0.0063 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. BonusCommission | 0.0108 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. FreeCover | 0.0454 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. SeriousIllnessType | 0.016 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. SignedDecReceived | 0.0924 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. day\_of\_week | 0.0094 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. day\_of\_month | 0.0048 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. month | 0.0079 | 1123.3155 | 4.34E-199 |
| PolicyIssued vs. year | 0.0333 | 1123.3155 | 4.34E-199 |

|  |  |
| --- | --- |
| **Feature** | **Importance Score** |
| Agency | 0.55978 |
| WorkflowStatus | 0.173468 |
| UWDecision | 0.118432 |
| CommDateProvided | 0.033788 |
| SignedDecReceived | 0.013722 |
| NoOfLives | 0.013316 |
| ProductGroup | 0.011579 |
| SeriousIllnessType | 0.011124 |
| BonusCommission | 0.011008 |
| Product | 0.010811 |
| CommissionTerms | 0.010571 |
| Indexation | 0.009274 |
| ProductType | 0.006298 |
| RenewalSacrificeType | 0.004066 |
| CommissionSacrificeType | 0.003736 |
| ComissionSacrifice | 0.003247 |
| Discount | 0.002656 |
| PaymentFreq | 0.002633 |
| FreeCover | 0.00049 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experiment Features** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Support** |
| 1 feature: ['Agency'] | 0.7796 | 0.73 | 0.78 | 0.69 | 31453 |
| 2 features: ['Agency', 'WorkflowStatus'] | 0.7996 | 0.78 | 0.8 | 0.78 | 31453 |
| 3 features: ['Agency', 'WorkflowStatus', 'UWDecision'] | 0.8021 | 0.78 | 0.8 | 0.78 | 31453 |
| 4 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided'] | 0.8024 | 0.78 | 0.8 | 0.78 | 31453 |
| 5 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived'] | 0.7976 | 0.78 | 0.8 | 0.78 | 31453 |
| 6 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives'] | 0.7954 | 0.78 | 0.8 | 0.78 | 31453 |
| 7 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup'] | 0.7927 | 0.77 | 0.79 | 0.78 | 31453 |
| 8 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType'] | 0.7882 | 0.77 | 0.79 | 0.78 | 31453 |
| 9 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission'] | 0.7854 | 0.77 | 0.79 | 0.78 | 31453 |
| 10 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product'] | 0.7862 | 0.77 | 0.79 | 0.78 | 31453 |
| 11 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms'] | 0.7847 | 0.77 | 0.78 | 0.77 | 31453 |
| 12 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation'] | 0.7839 | 0.77 | 0.78 | 0.77 | 31453 |
| 13 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType'] | 0.7834 | 0.77 | 0.78 | 0.77 | 31453 |
| 14 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType'] | 0.7831 | 0.77 | 0.78 | 0.77 | 31453 |
| 15 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType', 'CommissionSacrificeType'] | 0.7829 | 0.77 | 0.78 | 0.77 | 31453 |
| 16 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType', 'CommissionSacrificeType', 'ComissionSacrifice'] | 0.7838 | 0.77 | 0.78 | 0.77 | 31453 |
| 17 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType', 'CommissionSacrificeType', 'ComissionSacrifice', 'Discount'] | 0.7835 | 0.77 | 0.78 | 0.77 | 31453 |
| 18 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType', 'CommissionSacrificeType', 'ComissionSacrifice', 'Discount', 'PaymentFreq'] | 0.7831 | 0.77 | 0.78 | 0.77 | 31453 |
| 19 features: ['Agency', 'WorkflowStatus', 'UWDecision', 'CommDateProvided', 'SignedDecReceived', 'NoOfLives', 'ProductGroup', 'SeriousIllnessType', 'BonusCommission', 'Product', 'CommissionTerms', 'Indexation', 'ProductType', 'RenewalSacrificeType', 'CommissionSacrificeType', 'ComissionSacrifice', 'Discount', 'PaymentFreq', 'FreeCover'] | 0.7829 | 0.77 | 0.78 | 0.77 | 31453 |

**IMBALANCED DATASET**

**A chart with many different colored squares

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**A close-up of a chart

Description automatically generated**



**OVERSAMPLED DATASET – ALL FEATURES**



**OVERSAMPLED DATASET – TOP-N FEATURES**





**OVERSAMPLED DATASET – WITHOUT SELECTED FEATURES**



**OVERSAMPLED DATASET – ENGINEERED FEATURES**





**UNDERSAMPLED DATASET – ALL FEATURES**



**UNDERSAMPLED DATASET – TOP-N FEATURES**





**HYPERPARAMETER TUNING & OPTIMISATION – DECISION TREE**



**HYPERPARAMETER TUNING & OPTIMISATION – RANDOM FOREST**

**Accuracy**



**ROC AUC**



# **References**

Adibi, A., Sadatsafavi, M. and Ioannidis, J.P. (2020) ‘Validation and utility testing of clinical prediction models’, JAMA, 324(3), p. 235. doi:10.1001/jama.2020.1230.

Adler, A.I. and Painsky, A. (2022) ‘Feature importance in gradient boosting trees with cross-validation feature selection’, Entropy, 24(5), p. 687. doi:10.3390/e24050687.

Andonie, R. (2019) ‘Hyperparameter optimization in Learning Systems’, Journal of Membrane Computing, 1(4), pp. 279–291. doi:10.1007/s41965-019-00023-0.

Ahern, I., Noack, A., Guzman-Nateras, L., Dou, D., Li, B. and Huan, J., 2019. NormLime: A new feature importance metric for explaining deep neural networks. arXiv preprint arXiv:1909.04200.

Ali, A. and Gravino, C. (2021) ‘An empirical comparison of validation methods for software prediction models’, Journal of Software: Evolution and Process, 33(8). doi:10.1002/smr.2367.

Altmann, A. et al. (2010) ‘Permutation importance: A corrected feature importance measure’, Bioinformatics, 26(10), pp. 1340–1347. doi:10.1093/bioinformatics/btq134.

Ampt, A.B.F. (2017) *On the potential for machine learning in prediction of insurance policy sales: Helping Insurance Intermediaries Get Insights in their clients' insurance needs*. thesis.

AN, S.H., YEO, S.H. and KANG, M., 2021. A Study on a car Insurance purchase Prediction Using Two-Class Logistic Regression and Two-Class Boosted Decision Tree. Korea Journal of Artificial Intelligence, 9(1), pp.9-14.

Anagol, S., Cole, S. and Sarkar, S. (2017) ‘Understanding the advice of commissions-motivated agents: Evidence from the Indian Life Insurance Market’, Review of Economics and Statistics, 99(1), pp. 1–15. doi:10.1162/rest\_a\_00625.

Ao, D., Hu, Z. and Mahadevan, S. (2017) ‘Design of validation experiments for Life prediction models’, Reliability Engineering & System Safety, 165, pp. 22–33. doi:10.1016/j.ress.2017.03.030.

Asir, D., Appavu, S. and Jebamalar, E. (2016) ‘Literature review on feature selection methods for high-dimensional data’, International Journal of Computer Applications, 136(1), pp. 9–17. doi:10.5120/ijca2016908317.

Azpurua, M.A. et al. (2014) ‘A review on the drawbacks and enhancement opportunities of the feature selective validation’, IEEE Transactions on Electromagnetic Compatibility, 56(4), pp. 800–807. doi:10.1109/temc.2014.2304622.

Barry, L. and Charpentier, A., 2022. The Fairness of Machine Learning in Insurance: New Rags for an Old Man?. arXiv preprint arXiv:2205.08112.

Bujang, M.A. and Baharum, N. (2016) ‘Sample size guideline for Correlation Analysis’, World Journal of Social Science Research, 3(1), p. 37. doi:10.22158/wjssr.v3n1p37.

Cabitza, F. et al. (2021) ‘The importance of being external. methodological insights for the external validation of machine learning models in medicine’, Computer Methods and Programs in Biomedicine, 208, p. 106288. doi:10.1016/j.cmpb.2021.106288.

Chandrashekar, G. and Sahin, F. (2014) ‘A survey on feature selection methods’, Computers & Electrical Engineering, 40(1), pp. 16–28. doi:10.1016/j.compeleceng.2013.11.024.

Chang, W.T. and Lai, K.H. (2021) “A neural network-based approach in predicting consumers' intentions of purchasing insurance policies,” Acta Informatica Pragensia, 10(2), pp. 138–154. Available at: <https://doi.org/10.18267/j.aip.152>.

Chen, W. et al. (2007) ‘A design-driven validation approach using Bayesian prediction models’, Journal of Mechanical Design, 130(2). doi:10.1115/1.2809439.

Cho, H. et al. (2020) ‘Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks’, IEEE Access, 8, pp. 52588–52608. doi:10.1109/access.2020.2981072.

Debray, T.P.A. et al. (2015) ‘A new framework to enhance the interpretation of external validation studies of Clinical Prediction Models’, Journal of Clinical Epidemiology, 68(3), pp. 279–289. doi:10.1016/j.jclinepi.2014.06.018.

Demircioğlu, A. (2021) ‘Measuring the bias of incorrect application of feature selection when using cross-validation in radiomics’, Insights into Imaging, 12(1). doi:10.1186/s13244-021-01115-1.

Dragos, S.L., Dragos, C.M. and Muresan, G.M. (2020) ‘From intention to decision in Purchasing Life Insurance and private pensions: Different effects of knowledge and behavioural factors’, Journal of Behavioral and Experimental Economics, 87, p. 101555. doi:10.1016/j.socec.2020.101555.

Franceschi, L., Donini, M., Frasconi, P. and Pontil, M., 2017, July. Forward and reverse gradient- based hyperparameter optimization. In International Conference on Machine Learning (pp. 1165-1173). PMLR.

Frempong, N.K., Nicholas, N. and Boateng, M.A., 2017. Decision tree as a predictive modeling tool for auto insurance claims. International Journal of Statistics and Applications, 7(2), pp.117-120.

Greenwell, B.M., Boehmke, B.C. and McCarthy, A.J., 2018. A simple and effective model-based variable importance measure. arXiv preprint arXiv:1805.04755.

Gregorutti, B., Michel, B. and Saint-Pierre, P. (2016) ‘Correlation and variable importance in random forests’, Statistics and Computing, 27(3), pp. 659–678. doi:10.1007/s11222-016-9646-1.

Groll, A., Wasserfuhr, C. and Zeldin, L., 2022. Churn modeling of life insurance policies via statistical and machine learning methods--Analysis of important features. arXiv e-prints, pp.arXiv-2202.

Gogtay, N.J. and Thatte, U.M. (2017) Statistics for Researcher: Principles of Correlation Analysis. Journal of the Association of Physicians of India, 65, 78- 80.

Gopagoni, D.R., Lakshmi, P.V. and Siripurapu, P. (2020) “Predicting the sales conversion rate of car insurance promotional calls,” Rising Threats in Expert Applications and Solutions, pp. 321–329. Available at: https://doi.org/10.1007/978-981-15-6014-9\_37.

Hanafy, M. and Ming, R. (2021) ‘Machine learning approaches for auto insurance big data’, Risks, 9(2), p. 42. doi:10.3390/risks9020042.

Heinze, G. and Dunkler, D. (2016) ‘Five myths about variable selection’, Transplant International, 30(1), pp. 6–10. doi:10.1111/tri.12895.

Hickey, G.L. and Blackstone, E.H. (2016) ‘External model validation of binary clinical risk prediction models in cardiovascular and Thoracic Surgery’, The Journal of Thoracic and Cardiovascular Surgery, 152(2), pp. 351–355. doi:10.1016/j.jtcvs.2016.04.023.

Hooker, S. et al. (2018) Evaluating feature importance estimates, Google Research. Available at: https://research.google/pubs/pub47088/ (Accessed: 11 May 2023).

Huang, X., Wu, L. and Ye, Y. (2019) ‘A review on dimensionality reduction techniques’, International Journal of Pattern Recognition and Artificial Intelligence, 33(10), p. 1950017. doi:10.1142/s0218001419500174.

Imai, K., Tingley, D. and Yamamoto, T. (2012) ‘Experimental designs for identifying causal mechanisms’, Journal of the Royal Statistical Society Series A: Statistics in Society, 176(1), pp. 5–51. doi:10.1111/j.1467-985x.2012.01032.x.

Ivanescu, A.E. et al. (2015) ‘The importance of prediction model validation and assessment in obesity and Nutrition Research’, International Journal of Obesity, 40(6), pp. 887–894. doi:10.1038/ijo.2015.214.

Jaiswal, R., 2022. PROGNOSTICATING CUSTOMERS’INTENTION TO PURCHASE AN INSURANCE PLAN WITH MACHINE LEARNING. Fostering Resilient Business Ecosystems and Economic Growth: Towards the Next Normal, p.292.

Joy, T.T., Rana, S., Gupta, S. and Venkatesh, S., 2016, December. Hyperparameter tuning for big data using Bayesian optimisation. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2574-2579). IEEE.

Ke, G. et al. (2017) Lightgbm: Proceedings of the 31st International Conference on Neural Information Processing Systems, Guide Proceedings. Available at: https://dl.acm.org/doi/10.5555/3294996.3295074 (Accessed: April 1, 2023).

Konig, G. et al. (2021) ‘Relative feature importance’, 2020 25th International Conference on Pattern Recognition (ICPR) [Preprint]. doi:10.1109/icpr48806.2021.9413090.

Li, J. et al. (2017) ‘Feature selection’, ACM Computing Surveys, 50(6), pp. 1–45. doi:10.1145/3136625.

Li, S. (2019) ‘Insurance customer purchase prediction based on data optimization’, Statistics and Application, 08(05), pp. 784–796. doi:10.12677/sa.2019.85089.

Mai, T.H. et al. (2020) ‘A study on behaviors of purchasing life insurance in Vietnam’, Management Science Letters, pp. 1693–1700. doi:10.5267/j.msl.2020.1.011.

Mantovani, R.G., Horváth, T., Cerri, R., Junior, S.B., Vanschoren, J. and de Carvalho, A.C.P.D.L.F., 2018. An empirical study on hyperparameter tuning of decision trees. arXiv preprint arXiv:1812.02207.

Mau, S., Pletikosa, I. and Wagner, J. (2018) ‘Forecasting the next likely purchase events of insurance customers’, International Journal of Bank Marketing, 36(6), pp. 1125–1144. doi:10.1108/ijbm-11-2016-0180.

Mauritsius, T. et al. (2020) ‘Customer churn prediction models for PT. XYZ Insurance’, 2020 8th International Conference on Orange Technology (ICOT) [Preprint]. doi:10.1109/icot51877.2020.9468788.

McDonnell, K. et al. (2023) ‘Deep learning in insurance: Accuracy and model interpretability using TabNet’, Expert Systems with Applications, 217, p. 119543. doi:10.1016/j.eswa.2023.119543.

Merikanto, K (2022) Using Machine Learning To Predict Purchase Potential From Customer Data. Thesis.

Miao, J. and Niu, L. (2016) ‘A survey on feature selection’, Procedia Computer Science, 91, pp. 919–926. doi:10.1016/j.procs.2016.07.111.

Misra, P. and Yadav, A.S., 2020. Improving the classification accuracy using recursive feature elimination with cross-validation. Int. J. Emerg. Technol, 11(3), pp.659-665.

Molina, L.C., Belanche, L. and Nebot, A. (2002) ‘Feature selection algorithms: A survey and experimental evaluation’, 2002 IEEE International Conference on Data Mining, 2002. Proceedings. [Preprint]. doi:10.1109/icdm.2002.1183917.

Morrison, R.E. et al. (2013) ‘Data Partition Methodology for validation of Predictive Models’, Computers & Mathematics with Applications, 66(10), pp. 2114–2125. doi:10.1016/j.camwa.2013.09.006.

Muranda, C., Ali, A. and Shongwe, T. (2021) ‘Deep learning method for detecting fraudulent motor insurance claims using unbalanced data’, 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS) [Preprint]. doi:10.1109/itms52826.2021.9615264.

Nomi, M. and Sabbir, Md.M. (2020) ‘Investigating the factors of consumers’ purchase intention towards Life Insurance in Bangladesh: An application of the theory of reasoned action’, Asian Academy of Management Journal, 25(2). doi:10.21315/aamj2020.25.2.6.

Paruchuri, H. (2020) ‘The impact of machine learning on the future of insurance industry’, American Journal of Trade and Policy, 7(3), pp. 85–90. doi:10.18034/ajtp.v7i3.537.

Parvandeh, S. et al. (2020) ‘Consensus features nested cross-validation’, Bioinformatics, 36(10), pp. 3093–3098. doi:10.1093/bioinformatics/btaa046.

Paul, M.J. (2017) ‘Feature selection as causal inference: Experiments with text classification’, Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017) [Preprint]. doi:10.18653/v1/k17-1018.

Pesantez-Narvaez, J., Guillen, M. and Alcañiz, M. (2019) ‘Predicting motor insurance claims using telematics data—XGBoost versus logistic regression’, Risks, 7(2), p. 70. doi:10.3390/risks7020070.

Probst, P., Wright, M.N. and Boulesteix, A. (2019) ‘Hyperparameters and tuning strategies for Random Forest’, WIREs Data Mining and Knowledge Discovery, 9(3). doi:10.1002/widm.1301.

Quan, Z. and Valdez, E.A. (2018) ‘Predictive analytics of insurance claims using multivariate decision trees’, Dependence Modeling, 6(1), pp. 377–407. doi:10.1515/demo-2018-0022.

Rahman, M.S. et al. (2017) ‘Review and evaluation of performance measures for survival prediction models in external validation settings’, BMC Medical Research Methodology, 17(1). doi:10.1186/s12874-017-0336-2

Rajbahadur, G.K. et al. (2022) ‘The impact of feature importance methods on the interpretation of defect classifiers’, IEEE Transactions on Software Engineering, 48(7), pp. 2245–2261. doi:10.1109/tse.2021.3056941.

Rubi, M.A. et al. (2022) “Machine learning prediction of consumer travel insurance purchase behavior,” 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT) [Preprint]. Available at: https://doi.org/10.1109/icccnt54827.2022.9984470.

Reddy, G.T. et al. (2020) ‘Analysis of dimensionality reduction techniques on Big Data’, IEEE Access, 8, pp. 54776–54788. doi:10.1109/access.2020.2980942.

Saarela, M. and Jauhiainen, S. (2021) ‘Comparison of feature importance measures as explanations for classification models’, SN Applied Sciences, 3(2). doi:10.1007/s42452-021-04148-9.

Senthilnathan, S. (2019) ‘Usefulness of correlation analysis’, SSRN Electronic Journal [Preprint]. doi:10.2139/ssrn.3416918.

Severino, M.K. and Peng, Y. (2021) ‘Machine learning algorithms for fraud prediction in Property Insurance: Empirical Evidence Using Real-World Microdata’, Machine Learning with Applications, 5, p. 100074. doi:10.1016/j.mlwa.2021.100074.

Shi, L. et al. (2018) ‘Variable selection and validation in multivariate modelling’, Bioinformatics, 35(6), pp. 972–980. doi:10.1093/bioinformatics/bty710.

Steyerberg, E.W. and Harrell, F.E. (2016) ‘Prediction models need appropriate internal, internal–external, and external validation’, Journal of Clinical Epidemiology, 69, pp. 245–247. doi:10.1016/j.jclinepi.2015.04.005.

Steyerberg, E.W. and Vergouwe, Y. (2014) ‘Towards better clinical prediction models: Seven steps for development and an ABCD for validation’, European Heart Journal, 35(29), pp. 1925–1931. doi:10.1093/eurheartj/ehu207.

Stucki, O. (2019) Predicting the customer churn with machine learning methods - CASE: private insurance customer data. thesis.

Taha, A., Cosgrave, B. and Mckeever, S. (2022) ‘Using feature selection with machine learning for generation of insurance insights’, Applied Sciences, 12(6), p. 3209. doi:10.3390/app12063209.

Tantithamthavorn, C. et al. (2017) ‘An empirical comparison of model validation techniques for defect prediction models’, IEEE Transactions on Software Engineering, 43(1), pp. 1–18. doi:10.1109/tse.2016.2584050.

van der Putten, P., de Ruiter, M. and van Someren, M. (2000) CoIL Challenge 2000 Tasks and Results: Predicting and Explaining Caravan Policy Ownership, Coil Challenge 2000 Report. Available at: https://liacs.leidenuniv.nl/~puttenpwhvander/library/cc2000/ (Accessed: April 1, 2023).

van Geloven, N. et al. (2022) ‘Validation of prediction models in the presence of competing risks: A guide through modern methods’, BMJ [Preprint]. doi:10.1136/bmj-2021-069249.

Venkatesh, B. and Anuradha, J. (2019) ‘A review of feature selection and its methods’, Cybernetics and Information Technologies, 19(1), pp. 3–26. doi:10.2478/cait-2019-0001.

Wei, G. et al. (2020) ‘A novel hybrid feature selection method based on dynamic feature importance’, Applied Soft Computing, 93, p. 106337. doi:10.1016/j.asoc.2020.106337.

Wojtas, M. and Chen, K., 2020. Feature importance ranking for deep learning. Advances in Neural Information Processing Systems, 33, pp.5105-5114.

Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H. and Deng, S.H., 2019. Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1), pp.26-40.

Xia, H., Zhou, Y. and Zhang, Z. (2022) ‘Auto Insurance Fraud Identification based on a CNN-LSTM fusion deep learning model’, International Journal of Ad Hoc and Ubiquitous Computing, 39(1/2), p. 37. doi:10.1504/ijahuc.2022.120943.

Xie, Z. et al. (2020) ‘Fist: A feature-importance sampling and tree-based method for automatic design flow parameter tuning’, 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC) [Preprint]. doi:10.1109/asp-dac47756.2020.9045201.

Yang, L. and Shami, A. (2020) ‘On hyperparameter optimization of Machine Learning Algorithms: Theory and practice’, Neurocomputing, 415, pp. 295–316. doi:10.1016/j.neucom.2020.07.061.

Zhang, B., Rajan, R., Pineda, L., Lambert, N., Biedenkapp, A., Chua, K., Hutter, F. and Calandra, R., 2021, March. On the importance of hyperparameter optimization for model-based reinforcement learning. In International Conference on Artificial Intelligence and Statistics (pp. 4015-4023). PMLR.

Zhang, R. et al. (2017) ‘Deep and shallow model for insurance churn prediction service’, 2017 IEEE International Conference on Services Computing (SCC) [Preprint]. doi:10.1109/scc.2017.51.

Zhou, Z. and Hooker, G. (2021) ‘Unbiased measurement of feature importance in tree-based methods’, ACM Transactions on Knowledge Discovery from Data, 15(2), pp. 1–21. doi:10.1145/3429445.