ISSN 2563-7568

RESEARCH ARTICLE

Optimizing Travel Insurance Purchase Detection using Predictive Models

Benjamin Borketey[1*], Ernest F Aboagye[2], Kwasi Danquah[3]

[1]*Department of Economics, The university of Akron, USA*
[2]*J Mack Robinson College of Business, Georgia State University, USA*
[3]*Department of Mathematics, Youngstown State University, USA*

Abstract

What traveler features should be considered when designing airline travel insurance policies, and can predictive modeling enhance the accuracy of purchase predictions? Motivated by the increased need to safeguard investments due to frequent flight interruptions and cancellations during the COVID-19 pandemic and its travel restrictions, we investigate the uptake of flight travel insurance using predictive models. This study applies various machine learning techniques to a dataset consisting of 1,987 travelers, examining whether they purchased travel insurance (a binary classification problem). Performance metrics such as misclassification rate, precision, recall, F-score, and the area under the receiver operating characteristic curve (AUC) are used to assess model effectiveness. The models were optimized using cross-validation on the training data. Among the models tested, eXtreme Gradient Boosting Machine (XGBoost) achieved the highest accuracy rate of 86%, along with the best AUC, precision, recall, and specificity, indicating a 98% accuracy in predicting who will purchase travel insurance. Other robust models, such as ensemble methods and neural networks, also demonstrated strong performance, with similar AUC and precision scores. Features such as annual income, age, travel history, and education history were found to be the most significant predictors, while chronic disease history had little impact. Parsimonious predictive models, using only the most important variables, yielded better performance. Our findings highlight the critical role of predictive accuracy in helping insurers mitigate the financial risk due to travel interruptions.

Key Words: *Travel; Machine learning, AUC; Ensemble; Error rate; Insurance; Logistic; Neural networks; Precision; Recall; Specificity; XGBoost*

***Corresponding Author**: *Benjamin Borketey, Department of Economics, The university of Akron, USA; E-mail: bbortey9@gmail.com*

Received Date: *November 22, 2024,* ***Accepted Date***: *December 15, 2024,* ***Published Date***: *December 26, 2024*

Citation: *Borketey B, Aboagye EF, Danquah K. Optimizing Travel Insurance Purchase Detection using Predictive Models. Int J Auto AI Mach Learn. 2024;4(2):173-207.*

1. Introduction

As air travel continues to be a cornerstone of modern connectivity for individuals, businesses, and governments, ensuring travelers' safety and financial security has become increasingly vital. Travel insurance serves as a critical safeguard against unexpected financial losses during domestic or international trips, covering events such as trip cancellations, lost luggage, and emergency medical evacuations [1]. This demand for travel insurance has grown significantly in recent years. According to the United States Travel Insurance Association, Americans spent approximately $4 billion on travel protection in 2018, reflecting a 41% increase since 2016, with the number of people covered increasing by more than 10.7% from 2021. The increasing number of insured travelers underscores a growing awareness of the benefits of such policies.

The COVID-19 pandemic highlighted the indispensable role of travel insurance. The widespread travel restrictions disrupted plans globally, leaving many uninsured travelers financially vulnerable. Forbes (2020) reported that flight cancellations impacted 62% of American adults, with many losing money due to a lack of insurance. While some passengers received vouchers or refunds, these cases highlighted the financial risks faced by uninsured travelers. Post-pandemic, intermittent disruptions—including weather-related cancellations and strikes—have reinforced the necessity of travel insurance as a fundamental component of trip planning. Consequently, policies offering trip cancellation and interruption benefits accounted for nearly 90% of travel protection products purchased in 2018.

Despite the increasing adoption of travel insurance, research exploring factors influencing its purchase behavior remains limited. Existing studies primarily emphasize qualitative analyses or focus on the medical aspects of insurance [2,3]. While recent works by Karl and Kerr and Kelly [4,5] delve into risk perception and uncertainty in travel decisions, they do not address predictive modeling of insurance purchases. This gap calls for a deeper understanding of the demographic, social, and economic factors driving consumer behavior. Designing targeted insurance programs necessitates insights derived from predictive analytics, offering insurers actionable strategies to optimize product design and marketing.

Advances in machine learning (ML) have revolutionized predictive analytics in the insurance sector. Predictive modeling has been employed for risk mitigation in various insurance sectors for streamlining the underwriting and claims processes, computing pure premiums, detecting fraud in the insurance industry and in health analytics [6-10]. Among the consistently used ML techniques, XGBoost, random forests, and neural networks provide superior predictive accuracy compared to traditional statistical models such as logistic regression [11]. Feature selection approaches, including SHAP values, enhance model interpretability and ensure actionable insights for insurers [12]. Additionally, addressing challenges like data imbalance using methods such as SMOTE has improved the fairness and reliability of models, particularly for predicting rare events [13].

This research makes significant contributions to both theory and practice by integrating machine learning approaches to predict travel insurance purchases. It compares traditional statistical methods with cutting-edge models to evaluate predictive performance and economic implications. By focusing on feature selection, model robustness, and practical applications, the study provides a comprehensive framework for designing personalized insurance products and effective marketing strategies. Additionally, it examines the broader economic implications of predictive modeling, highlighting its potential to optimize risk

assessment, reduce operational costs, and improve customer retention. By situating predictive modeling within the context of the travel insurance sector, this research addresses critical gaps in existing literature while offering actionable insights for insurers to navigate an increasingly dynamic market. Insights from predictive models allow insurers to target potential customers with tailored offers, increasing conversion rates [14,15]. Consequently, predictive models in travel insurance can lead to automation in customer segmentation and pricing strategies that enables insurers to set premiums that reflect individual risk, ensuring profitability and reducing underwriting costs while remaining competitive. Adaptive pricing driven by real-time data enhances customer satisfaction by offering more affordable options to low-risk travelers, making insurance more accessible. Subsequently, affordable and reliable travel insurance encourages more individuals to travel, indirectly supporting tourism and associated industries.

The remainder of this paper is structured as follows: Section 2 reviews relevant literature on machine learning applications in insurance; Section 3 describes the methodology, including data sources, performance metrics, and cross-validation techniques. Section 4 presents the results and discusses their implications. Finally, Section 5 concludes with key findings and directions for future research.

2. Related Studies in Insurance and Predictive Models

Predictive modeling, also known as data mining, has been defined as abstracting useful information from available data using statistical and machine learning techniques [16]. Despite the host of data mining techniques and applications at present, studies into insurance, finance and economics are very few. One possible reason for this is the dearth of data for research. Employed six data mining techniques to examine the predictive accuracy of default of credit card clients using the sorting smoothing technique as a basis to select the artificial neural network as the best performing model [17]. Popular and some recent finance literature looking into credit fraud detection has had these classification techniques chosen as the best: artificial neural networks, support vector machines, discriminant analysis, k-nearest neighbors, logistic regression, Bayesian learning, and random forests [17-28]

For insurance industries, using machine learning technology has the most important advantage of the data set facility. Every type of data whether it is structured, unstructured or semi-structured can be modified using machine learning. The use of machine learning is dependent on the worth chain, through cutting-edge accuracy, feature engineering, and, qualitatively, client conduct.

Classification machine learning predictions in insurance have largely been spent analyzing data either in the auto insurance industry or data from insurance claims or fraud detection in auto insurance. Using artificial neural network, multinomial logistic regression, and decision trees, predicted auto insurance claims and concluded that the neural network performed better than the other two with a test accuracy of about 62% [29]. Wüthrich analyzed claims in liability insurance using the non-parametric classification and regression tree (CART) techniques to study feature information [30]. Ensemble models have been used multiple times, especially in credit scoring algorithms, as they are more stable and accurate in predicting the results and liabilities. In their research, Tsai C-F conclude that variance and bias are also known to be reduced by using ensemble models [31].

Since the invention of the eXtreme Gradient Boosting (XGBoost) technique in 2014, the machine learning literature has seen an uptick in its use and importance for classification

model prediction. Many data scientists have used this technique to achieve state-of-the-art results on a lot of challenges in predictive modeling. According to Chen and Guestrin an impressive 17 of 29 winning solutions of the Kaggle data science competitions used the XGBoost model [32]. The authors contributed to its literature by comparing the predictive accuracy of claims data from insurers such as Allstate Insurance using XGBoost, AdaBoost and neural networks. XGBoost performed best. It is also worthwhile to note that deep neural network was the second most used among the winning models for the Kaggle competitions. XGBoost is usually extremely good for tabular problems, and deep learning is the best for unstructured data problems. With both speed and performance provided by this algorithm, the implementation, acceptance, and love of XGBoost have grown exponentially in the last 5 years. It is a highly optimized algorithm with the approach of parallel processing and tree-pruning with the ability to handle missing values and regularization to avoid situations of bias and overfitting.

2.1. Travel insurance literature

Both risk perception and population characteristics are the key deciding factors in the purchasing willingness of travel insurance [32]. In fact, the higher the risk perception, the higher the demand for insurance. Lo, Cheung & Law found that the purchase of insurance, bringing extra cash, and searching for the latest information about the destination were the three most adopted risk-reduction strategies when planning future travel. Experienced travelers were more likely to use these three strategies than inexperienced travelers, who chose to seek advice from family or friends, seek advice from their travel agent, and not travel independently but travel in a tour group. In exploring the relationship between risk perception and the decision to travel to China, found the inclination to purchase travel insurance resulted from risk perception, willingness to pay a premium, medium length of stay, and higher monthly income. Al Mamun et al explored the willingness and purchase of travel insurance for international travel among working adults during the COVID-19 pandemic using partial least squares modeling and concluded factors that influence attitude toward travel insurance include insurance literacy about the product, perceived health risk of travel, and one's health consciousness [1]. They suggest an emphasis on travel insurance literacy and perceived health risk education among working adults to encourage them to purchase travel insurance policies for traveling abroad.

2.2. Predictive models

Classification models are commonly employed for categorical response variables, making them suitable for this study’s focus on a two-class binary classification problem. Recent advancements in machine learning for insurance modeling have introduced innovative approaches to improve classification accuracy and interpretability. Notable contributions include Althati, which explores ensemble methods for underwriting optimization, who examines neural network architectures for predicting claim probabilities, and utilize explainable AI to enhance decision-making in policy pricing [6,32]. These studies stress the importance of leveraging cutting-edge classification tools to address specific challenges in the insurance domain. In this paper, we explore the predictive accuracy of the classification models below, adapting and comparing their performance for travel insurance purchase prediction.

2.2.1 Logistic regression

Logistic regression measures the relationship between a categorical response variable and some independent variables by estimating probabilities using a logistic function, which is a cumulative logistic distribution. The conditional distribution of y given x is a Bernoulli distribution because the dependent variable is binary. It is an alternative to Fisher's 1936 method, linear discriminant analysis, but does not require the multivariate normal assumption of the latter. It is well-understood, easy to use, remains one of the most used for data mining in practice, and therefore provides a useful baseline for comparing the performance of newer methods, and can produce a simple probabilistic formula of classification [17,20]. The major advantage of this approach is that it can produce a simple probabilistic formula of classification.

2.2.2. K-Nearest Neighbors (KNN)

The k-Nearest Neighbors algorithm is a non-parametric lazy learning method for supervised learning, where an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. In learning systems, generalization performance is affected by a trade-off between the number of training examples and the capacity (e.g., the number of parameters) of the learning machine. The major advantage is that it is not required to establish a predictive model before classification. KNN does not perform any training when you supply the training data but rather stores the data during the training time and does not perform any calculations. Hence its lazy-learner attribute. It does not build a model until a query is performed on the dataset. This makes KNN ideal for data mining.

2.2.3. Discriminant analysis

Discriminant Analysis, also known as Fisher's rule, is a classification technique which projects onto a line an n-dimensional data by maximizing between-class mean and minimizing within-class variance and performs classification in this one-dimensional space. We have two common forms of Discriminant Analysis used in data mining: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). According to James, Witten, Hastie and Tibshirani, the LDA assumes the predictor is drawn from a multivariate Gaussian distribution with a class-specific mean vector and a common covariance matrix [33]. We need this model because when the classes are well-separated, the parameter estimates for the logistic regression model, the most used traditional classification technique, are surprisingly unstable. LDA does not suffer this problem and is even more popular with more than two response classes. Like the LDA, the QDA classifier results from assuming the observations from each class are drawn from a Gaussian distribution. QDA assumes a covariance matrix for each class, unlike the LDA. More recent methods include the Shrinkage Discriminant Analysis (SDA) and the Penalized Discriminant Analysis (PDA) which produced output identical to the logistic classifier in this research.

2.2.4. Partial Least Squares (PLS)

Partial Least Squares (PLS) is a well-known dimension reduction method which has been adapted for high dimensional classification problems in biology. It is a versatile method that can be used to predict either continuous or discrete/categorical variables. Classification with PLS is termed PLS-DA, where the DA stands for discriminant analysis. The PLS-DA algorithm has many favorable properties for dealing with multivariate data; one of the most important

of which is how variable collinearity is dealt with, and the model's ability to rank variables' predictive capacities within a multivariate context. We can say that PLS-DA regularizes in a way that behaves like squeezing the pooled covariance matrix of the LDA into a spherical shape the more the fewer latent variables are used.

2.2.5. Naïve Bayes Classifier (NB)

The naive Bayes classifiers are a form of simple "probabilistic classifier" that uses Bayes' theorem with strong (naive) independence assumptions. Naive Bayes classifiers are highly scalable, requiring a set of parameters that is proportional to the number of attributes. The difference between QDA and NB is that NB assumes independence of the features, which means the covariance matrices are diagonal matrices. This feature often tends to be an explanation for the poorer efficiency of NB when compared to QDA.

2.2.6. Regularization methods

Sometimes we need to choose between low variance and low bias. There is an approach that prefers some bias over high variance, this approach is called Regularization. It works well for most of the classification problems. Regularization methods involve fitting a model containing all p predictors but the estimated coefficients are shrunken toward zero relative to their least squares estimates. Hence, shrinkage methods can also perform variable selection. The shrinkage has the effect of reducing variance and may offset increased bias. Ridge regression and Lasso regression are the two most common forms of regularization, which seek to control variance by adding a tuning parameter λ. A third form called Elastic Net regression is the middle ground between ridge and lasso. The mathematical representation below will help to explain the three. The regular least-squares criterion minimizes the least-squares of the error plus a regularization term that is a product of a constant (α) and a sum of coefficients (beta)

$$min\{\sum(y - y_i)^2 + [\alpha\,\lambda \sum|\beta| + (1-\alpha)\lambda \sum|\beta|^2]\} \quad (1)$$

For lasso, α, which is the penalty coefficient, is set to 1 which makes the regularization term, a sum of absolute deviations. For ridge, the objective function is the residual sum of squares plus the product of the penalty, α which is 1, and the sum of squares of the coefficients. Both use a regularization parameter, λ, to control for variables and to balance minimizing the RSS versus minimizing the coefficients. When λ = 0, results are the same as regular linear regression: you have removed the penalty from the ridge regression. Coefficients get closer to zero as λ increases to infinity. Model complexity increases as λ increases. λ is just a scalar that should be learned from the data using cross-validation (tuning), always between 0 and 1. As opposed to the ridge, the lasso will always perform variable selection on those coefficients being shrunk toward 0 – they will be eliminated with a lasso but not with ridge regression. Elastic Net regression uses a weighted combination of both ridge and lasso regressions. From equation 1, α is always between 0 and 1. Elastic net has the power to do variable selection and shrink some coefficients to 0 as does lasso.

2.2.7. Decision trees and ensemble methods

The popularity of decision tree models in data mining arises from their ease of use, flexibility in terms of handling various data attribute types, and interpretability. Single tree models, however, can be unstable and overly sensitive to specific training data. Ensemble methods seek to address this problem by developing a set of models and aggregating their predictions in determining the class label for a data point. While there are numerous implementations of

decision trees, one of the most well-known is the C5.0 algorithm. The C5.0 algorithm has become the industry standard for producing decision trees because it does well for most types of problems directly out of the box. Compared to more advanced and sophisticated machine learning models, the decision trees under the C5.0 algorithm generally perform nearly as well but are much easier to understand and deploy. An advantage of the C5.0 algorithm is that it takes care of many of the decisions automatically using reasonable defaults using pruning. Its overall strategy is to post-prune the tree. It does this by first growing a large tree that overfits the training data. Afterward, nodes and branches that have little effect on the classification errors are removed.

Random decision forests (RF) are an ensemble learning method of classification (or regression) trees operated by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct the training overfitting problem of decision trees. The training algorithm for random forests applies to the general technique of bootstrap aggregating, or bagging to tree learners, by using a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. Random forests are computationally efficient since each tree is built independently of the others. With a large number of trees in the ensemble, they are also noted to be robust to overfitting and noise in the data. The number of attributes, p, used at a node and the total number of trees, ntree, in the ensemble are user-defined parameters. The error rate for a random forest has been noted to depend on the correlation between trees and the strength of each tree in the ensemble, with lower correlation and higher strength giving lower error. The number of random variables randomly sampled as candidates at each node split for classification is $\sqrt{p}$. Attribute selection at a node is based on the Gini index, though other selection measures may also be used. Breiman proved random forests have comparable performance to the other modern sophisticated techniques like support vector machines, boosting, and artificial neural networks [34]. Random forest models are much more flexible than linear models and can model complicated nonlinear effects as well as automatically capture interactions between variables. They tend to give very good results on real-world data.

Boosting is a machine learning meta-algorithm for regression and classification problems that produces a prediction model in the form of an ensemble of weak prediction decision-tree models. Most boosting algorithms iteratively learn the distribution of weak classifiers and add them to a final strong classifier through a weighting and re-weighting mechanism related to the weak learners' accuracy. Thus, future weak learners focus more on the examples that previous weak learners misclassified. In this research, two types of boosting called gradient boosting machine (GBM) and eXtreme gradient boosting (XGBoost) machines are employed and have been proven to improve performance and are very popular as noted in the literature review section. Computational capacity-wise, GBM is many times faster than XGBoost and is a much better approach when dealing with large datasets. GBM is boosting with errors minimized by gradient descent. XGBoost executes gradient boosted decision trees designed for speed and performance.

2.2.8. Support Vector Machines (SVM)

A support vector machine model is the representation of the points in space mapped so that examples of the separate categories are divided by a wide clear gap. It then maps new examples into that same space and the predicted to belong to a category based on which side of the gap they fall on. It is a non-probabilistic binary linear classifier. Bhattacharyya states that SVMs are statistical learning techniques that are very successful in a variety of

classification tasks and that several unique features of these algorithms make them especially suitable for binary classification problems like predicting insurance purchase choices [20]. SVMs are linear classifiers that work in a high dimensional feature space that is a non-linear mapping of the input space of the problem at hand. An advantage of working in a high-dimensional feature space is that in many problems the non-linear classification task in the original input space becomes a linear classification task in the high-dimensional feature space. SVMs work in the high dimensional feature space without incorporating any additional computational complexity. The simplicity of a linear classifier and the capability to work in a feature-rich space make SVMs attractive for insurance purchase detection tasks where the highly unbalanced nature of the data (purchase or do not purchase cases) makes extraction of meaningful features critical to the detection of non-purchase transactions is difficult to achieve. The strength of SVMs comes from two important properties they possess kernel representation and margin optimization. In SVMs, mapping to a high-dimensional feature space and learning the classification task in that space without any additional computational complexity is achieved using a kernel function. We consider three kernels: linear, radial, and polynomial. Polynomial kernels are computationally slow but can give better predictions than radial or linear kernels. The second property of SVMs is the way the best classification function is arrived at. SVMs minimize the risk of overfitting the training data by determining the classification function (a hyper-plane) with a maximal margin of separation between the two classes. This property provides SVMs with very powerful generalization capability in classification.

2.2.9. Neural Networks (ANN, MLP)

The goal of the neural network is to solve problems in the same way that the human brain would, although several neural networks are more abstract. Theoretically, artificial neural networks (ANN) are highly robust in data distribution and can handle incomplete, noisy, and ambiguous data. They are well suited for modeling complex, nonlinear phenomena ranging from financial management, and hydrological modeling to natural hazard prediction. For any neural computing, training time is always the biggest bottleneck and thus, every effort is needed to make training effective and affordable. Training time is a function of the complexity of the network topology which is ultimately determined by the combination of hidden layers and neurons. A trade-off is needed to balance the processing purpose of the hidden layers and the training time needed. One of the major developments in neural networks over the last decade is the model combining or ensemble modeling. A network without a hidden layer is only able to solve a linear problem. To tackle a nonlinear problem, a reasonable number of hidden layers is needed. The ANN plot for the complete data with two hidden layers is found in Appendix C2. The plot shows the network interconnectedness and how the model activity functions to predict the probability of travel insurance purchase in this case.

The application of artificial neural networks using modern hardware is called deep learning. It allows the development and training, and the utilization of much larger neural networks, thus including more layers, than what was previously thought possible. Three cases: MLPs, CNNs, and RNNs. These three classes of deep learning networks provide a lot of flexibility and have proven themselves over decades to be useful and reliable in a wide range of problems. They are multilayer perceptron (MLP), convoluted neural network (CNN), and the recurrent neural network (RNN). MLPs are suitable for classification prediction problems where inputs are assigned a class or label and so are considered in this literature. A multilayer perceptron consists of at least one of an input layers, a hidden layer, and an output layer making up its nodes. Each node as a neuron uses a nonlinear activation function, with the exception of the input node. It uses a supervised learning technique called backpropagation

for training. Its multiple layers and non-linear activation What distinguish the MLP from a linear perceptron are its number of layers and non-linear activation. It can distinguish data that is not linearly separable. Preparing the data for a neural network is very important as all the covariates and responses need to be numeric. In our case, we have all of them categorical. The caret package in R software allows us to quickly create dummy variables as our input features.

3. Methodology

3.1. Data and data pre-processing

We use a publicly available dataset extracted from the Kaggle Data Science community website. This data has 1,987 observations of travelers and contains 9 variables including a binary response variable, Travel Insurance, having entries "Yes" or "No" representing the purchase choice of the observation. Table 1 presents a numerical summary of each variable in the data and their level of coding. To improve the data quality and produce more reliable results we need a thorough preparation of the data. This helps eliminate some of the noise in the data which may be caused by incomplete or missing data [35]. We re-coded all two-level categorical variables such as Travel Insurance (Yes for 1, No for 2), Employment Type (1 for Government Sector, 2 for Private Sector/Self Employed), Chronic Disease (1 for Yes, 0 for No), Graduate Or Not (1 for Yes, 0 for No), Frequent Flyer (1 for Yes, 0 for No), and Ever Traveled Abroad (1 for Yes, 0 for No). There were no missing values in the dataset. The table shows that of the 1987 observations, 36% of the travelers did purchase insurance, 85% are graduates, 28% have a history of chronic illness, 21% are Frequent Flyer passengers, and 19% indicated that they have traveled abroad in the past. Each flyer has approximately 5 family members, most likely works in the private sector or is self-employed and has an annual income of about 933,000 on average.

Figure 1 gives a graphical representation of the distribution of some of the variables. The average age of the observations is around 30 years with a minimum flyer age of 25 years and a maximum of 35 years. The modal age of the data is 28 years having a frequency of 506 observations. The pie chart shows the distribution of the response variable, Travel Insurance, discussed above: ~36% bought travel insurance while the rest did not. The distribution of the variable, Family Members, is skewed to the right. The plot on the bottom right of Figure 1 represents the histogram of frequency density of the annual income of the travelers. It has three peaks meaning annual income is tri-modal: one peak occurring at 300,000 to 700,000, the second one from 700,000 to 900,000, and the last one from 900,000 to 1,500,000. Graphical distributions of the other predictor variables are in Appendix A, corroborating the summary in Table 1.

Table 1: *Summary statistics of data.*

S. no	Variable	Levels	Mean	Median	Mode	Min	Max	N
0	Travel insurance	Yes (1), No (0)	0.36	0	0	0	1	1987
1	Age	25-35	29.65	29	28	25	35	1987
2	Employment type	GS (1), PS/SE (2)	1.71	2	2	1	2	1987
3	Graduate or not	Yes (1), No (0)	0.85	1	1	0	1	1987
4	Annual income	300K – 1.8M	933k	900K	800K	300K	1800K	1987
5	Family members	2, 3, 4, …, 9	4.75	5	4	2	9	1987
6	Chronic diseases	Yes (1), No (0)	0.28	0	0	0	1	1987
7	Frequent flyer	Yes (1), No (0)	0.21	0	0	0	1	1987
8	Ever traveled aboard	Yes (1), No (0)	0.19	0	0	0	1	1987

GS*= Government sector, PS = Private Sector, SE= Self-employed ***300k= 300,000

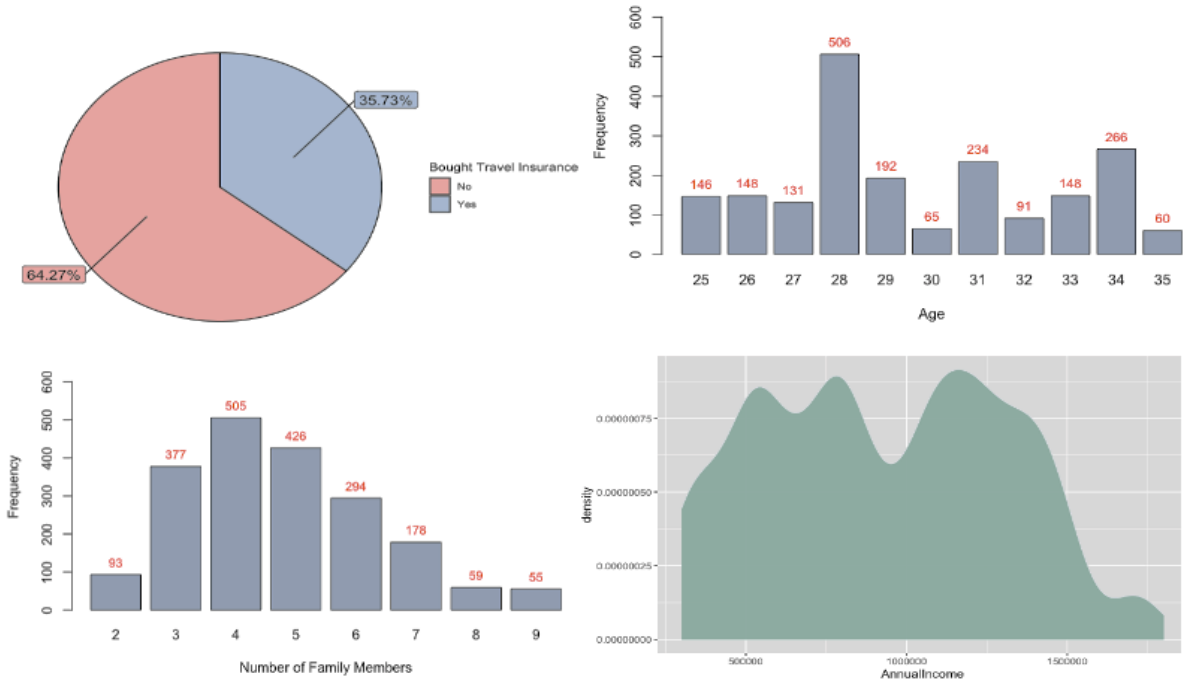


Figure 1: *Distribution of variables.*

The data is analyzed using Microsoft Excel and the open-source R Studio (version 2022.03.2+492) software due to its manipulability, reproducibility, and depth of packages for predictive analysis. Since the gold standard of model validation is out-of-sample evaluation, we split the dataset randomly for training and testing. In some preliminary analysis, we used the approach by Leach and Thayasivam to find the optimal splitting for the predictive models used with our dataset [18]. A 25%/75% train/test split was the most optimal for our research and so we used that for our analyses. 25% of the data is held out for testing the accuracy of our models against unseen data, while the remaining 75% is used for training and cross-validating our models. Thus 1491 observations are used for training the model and 496 for testing the model. The 25%/75% random sampling split of the data into training/testing ensures the same percentage of "Yes" (~36%) and "No" (~64%) responses to variable Travel Insurance existing in both splits as in the full data, thus removing any biases in the ordering of the dataset. Before each code is run, we set a seed for the same results to be reproduced for a model if run multiple times with no change in code.

3.2. Classification performance metrics

3.2.1. Model evaluation-misclassification rate, sensitivity, specificity

An evaluation metric measures model performance after training. The basic idea of model-wide evaluation is that performance measures are calculated by multiple threshold values. Most machine learning models produce some kind of scores in addition to predicted labels. These scores can be discriminant values, and posterior probabilities, among others. Model-wide evaluation measures are calculated by moving threshold values across the scores and selecting a single threshold value results in determining predicted labels. The scores predicted by our models fall in the range of 0.0 to 1.0. We used the default 0.5 probability threshold cutoff. A label is predicted as "positive" if the corresponding score is larger than a certain threshold value (0.5 in this case) or predicted as "negative", otherwise. Positive class for our response variable, Travel Insurance, is a "No" and negative class is a "Yes".

The two main measures of classification performance commonly noted in machine learning literature are considered as performance measures in this research: the misclassification rate and the area under the receiver operating characteristic curve (AUC). The misclassification (error) rate is the most common and intuitive measure derived from the confusion matrix. It is calculated as the number of all incorrect predictions divided by the total number of the dataset used for the prediction (TP+TN)/(TP+TN+FP+FN). Antonymous to the error rate is the accuracy rate which is calculated as (1–error rate) %. We report the accuracy values for consistency with the other performance measures considered.

The confusion matrix is a two-by-two table that contains four outcomes produced by a binary classifier. Various measures, such as error rate (1 – accuracy), specificity, sensitivity, precision, and F1 score are derived from the confusion matrix. Figure 2 is a sample confusion matrix derived as an output from our logistic regression model on the data. Results for the other models are captured in Table 2. Also called the True Positive Rate (TPR) or Recall, Sensitivity tells us what proportion of the positive class ("No") got correctly classified. In our case, it helps answer the question "What percentage of the people who traveled actually did not buy travel insurance?". It is calculated as TP/(TP+FN). Similarly, Specificity (or True Negative Rate) is the proportion of the negative class ("Yes") classified correctly after prediction. It answers the question, "What percentage of the people who traveled actually did buy travel insurance?" It is calculated as TN/(TN+FP). The 0.5 threshold can be adjusted depending on

whether we want to improve either sensitivity or specificity. Since we would like to know what impacts people's inability to purchase travel insurance, incorrectly classifying the "No" scores will not be in our best interest. Hence, improving sensitivity is ideal in our research: the higher the better. Precision score deals with only the positive predictions ("No" in our case). It is calculated as TP/(TP+FP). F-score is the harmonic mean of sensitivity and precision. Since both sensitivity and precision deal with the positive class, the F-score helps to summarize their predictive performance in one output. In this research, we used the F1 score and so assumed a β =1. It is calculated as (2*TPR*Precision)/(TPR+Precision). The F1 score does not care about how many true negatives are being classified. When working on an imbalanced dataset that demands attention on the negatives, Balanced Accuracy (the weighted average of train accuracy and test accuracy) does better than F1. Balanced accuracy is a better metric for this than F1 in cases when our focus is on both positives and negatives and not just one. When working on problems with heavily imbalanced datasets and you care more about detecting positives than detecting negatives, as is the case in this research, then you would prefer the F1 score more.

Tables 2 and 3 report the results for all these metrics. The best error rate is 0%, whereas the worst is 100%. For all the other metrics reported, the best is 100% and the worst is 0%. Except for the accuracy of the training data, the values reported in Tables 2 and 3 are from the predictions based on the test data. Balanced Accuracy values are reported in Appendix B. In the figure below, out of the 496 observations held out for testing the trained models, 311 and 111 were correctly classified as positives ("No") and negatives ("Yes") which gives an accuracy of 85.1% (= (311+111)/496). This implies a misclassification rate of 14.9%. The statistics for the other performance measures can be seen in the confusion matrix exhibit.

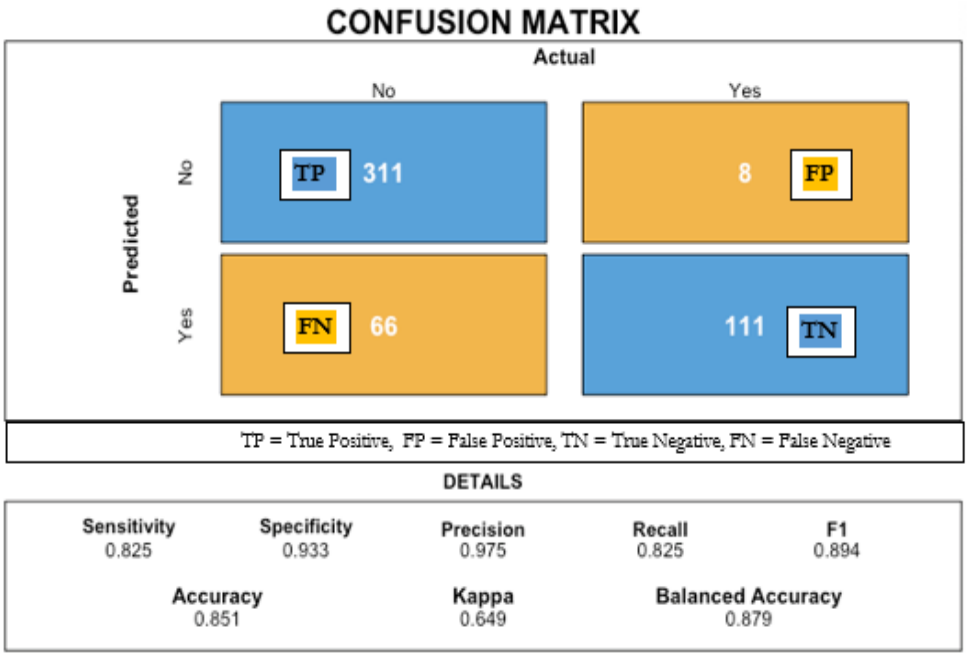


Figure 2: *Sample confusion matrix showing the four possible outcomes for a two-class classification problem.*

3.2.2. Area under the Receiver Operating Characteristics (ROC) curve (AUC)

Overall accuracy (minimum error rate) is not a sufficient performance indicator when there is significant class imbalance in the data since a default prediction of all cases into the majority class will show a high-performance value (Bhattacharyya et al, 2011). To supplement this, the

area under the receiver operating characteristic (ROC) curve (AUC), calculated as the space occupied by the ROC curve, is also used as a classification performance metric. Ling, Huang and Zhang (2003) argue that the AUC is a better measure than accuracy in comparing learning algorithms because the curve evaluates all possible thresholds for splitting predicted probabilities into predicted classes without worrying about threshold probability calibration [36]. ROC has been used in a wide range of fields, and the characteristics of the plot are also well studied. The ROC plot shows the trade-offs between specificity and sensitivity. It helps to visualize how well a machine learning classifier is performing. The AUC, reported as a percentage value, is the numerical measure of the ability of a classifier to distinguish between classes. The AUC score of a random classifier is 0.5 and should lie on a 45-degree baseline. Meaningful classifiers are greater than 0.5 and a high AUC means a better model which can help separate positive and negative classes. Figure 3 and Appendix E show the ROC plot of all the models reported in this research and their corresponding AUC values.

3.3. Optimization for the ML techniques

3.3.1. Hyper parametrizing

Cross-validation (CV) is a sampling method used to compare and select parameters for a machine learner on unseen data. This results in estimates with lower bias and modest variance than other traditional methods. Data is split into folds. For this research, we used 10-fold repeated cross-validation [ISRL, 2013]. This gave a better accuracy compared to either a 5-fold or 10-fold cross-validation without repetition. Repeated k-fold cross-validation helps to improve the estimated performance of a predictive model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs. This means the result is expected to be a more accurate estimate of the true unknown underlying mean performance of the model on the dataset, as calculated using the standard error. Using the 10-fold repeated cross-validation, we tuned our training models. Most of the models require a tuning parameter tune Length which helps explore more potential models for the best fit. Those that have larger values than the model could take are automatically truncated leading to better run times.

Logistic regression, LDA and QDA do not tune any parameter. The cross-validated k in our k-NN model is 64 after looping from 1 through 100 nearest neighbors. This means that our model predictors used 64 nearest neighbors to optimally predict what the next predicted class of Travel Insurance will be. The regularization models optimize the λ for computing the best accuracy. To achieve the optimum model, as can be seen in Table 2 and Appendix B2, ridge regression uses 0 α and an approximate 0.03 λ, lasso regression uses 1 α and 0.01 λ while the elastic net regression uses an α of 0.17 and λ of 0.38 – implying a mixture of 38% lasso regression and 62% ridge regression.

The primary tuning parameter of random forest models is mtry. This tuning parameter controls the number of features that go into each split. The default mtry is the square root of p, where p is the number of predictors used. Another important parameter is the number of trees at each split, ntree = 200 gave the least training error. Appendix G shows that the highest accuracy achieved from repeated cross-validation of the RF model occurred when 2 predictors are used.

With SVM, the best accuracy is produced by the polynomial kernel of degree 3 with cost 2 and scale=0.1 located in Appendix C1. With artificial neural network (ANN), a size of 2 and a decay of 0.29 gave the best prediction. The neural interconnectedness of the ANN for our model is shown in Appendix C2. Similarly, testing different values of the three layers of MLP,

we used 10 layers for all three to get the optimum metric values. The best-tuned hyper-parameters for all the models are reported in Appendix B. A summary of these is presented in the last columns of Tables 2 and 4.

3.3.2. Data balance

The class imbalance problem arises when the data have a proportion of one class significantly higher than another. This could be alleviated by various techniques including oversampling of the minority class or under sampling of the majority class to balance both classes. It is expected that when there is data imbalance, the prediction model will be more accurate for skewed classes (positive class). Since the sensitivity and specificity for most of our models considered have identical values, we are confident not treating for data balance. In almost all our models, the specificity numbers are greater than the sensitivity numbers. Even though the negative class ("Yes") is the minority class and since detecting the class who do not purchase travel insurance ("No") is of superior importance to us, we do think our model inherently solves the issue of data imbalance based on the results in Table 3. Nevertheless, neither under sampling nor oversampling provided better outputs for our final models. Also, from the output in each of our final predictive models, the confusion matrix additionally spits out a p-value. This p-value is used to evaluate a one-sided test to see if the test accuracy is better than the No Information Rate (NIR). The NIR is the largest class percentage in the data. This will be the positive ("Yes") class in our case since 64% of the travelers bought travel insurance. So NIR is about 64%. If you have a class imbalance, you might want to know if your model's accuracy is better than this proportion. This test answers the question, "Is test accuracy any better than the NIR of 64%? For all our models considered, the answer is yes, the accuracy is better since their corresponding p-values are all <0.000 and so we will reject the null hypothesis.

4. Results and Discussion

4.1. The full models

Tables 2 and 3 present the output results of 17 predictive models out of over 20 models tested. These supervised learning models were selected based on what has been used in the literature in the past, based on new models that performed better than known ones in the literature, or some models not making it to this paper had near-same values for at least one of the models considered here. From Table 2, we see that Elastic Net had the highest prediction for true positives ("Yes") with 312 observations, followed closely with 311 by the XGBoost and RF classifiers. ANN had the largest correct classification of negatives with 112 observations followed by GBM, XGBoost and DT classifiers.

Consequently, XGBoost produced the least test error of 14.9%, the highest accuracy of 85.1% on a 95% confidence interval of 81.6% to 88.1%. XGBoost had the best sensitivity of 82.5% and best specificity value of 93.3% which means that even though it is able to correctly classifier both positive and negative classes better than any of the predictive models considered using the 0.5 probability threshold cutoff, the XGBoost classifier can help predict the proportion of travelers who will purchase travel insurance in future more effectively. XGBoost also had the highest F1 score of 89.4% and the second highest Precision score of 97.5% only trailing that of Elastic Net by 0.3 percentage points. Because the Elastic Net model can correctly predict the positive classes, having the highest true positives (312) and the lowest false positives (7), it gave the highest precision rate 97.8% (since precision only cares about the positive class). Despite that statistic, the elastic net was one of the worst performing models with a test error

rate of 23% better than only LDA (23.4% – the worst) and ridge regression (23.2% – second worst).

GBM, DT and RF models also produced higher accuracies of about 84.3%, comparable to the XGBoost test accuracy. The artificial neural network model (ANN) performed better than deep learning neural network model (MLP) in all the statistics but the AUC % of the test data. Logistic, discriminant analyses or regularizers had accuracies and sensitivities less than 80%. The Naïve Bayes, Elastic Net and RF classifiers had the biggest difference between sensitivity and specificity predictions with a difference of at least +10 percentage points. This means either of these models can best help predict one class (the "Yes" in our case) far better than the others. All train and test accuracies exceed 75%.

Figure 3 graphs the ROC curves for all the predictive models. Any model on the 45-degree line is no more than a random classifier, and thus will have an AUC of 50%. Models with plots below that baseline are worse off. All our predictive models have their ROC curves above the 45-degree line. Since the ROC curve is a plot that considers the predicting probability at all threshold levels, not just 0.5, the higher AUC values can be used as a complement to the accuracy parameters considered here when choosing a model. The best performing model should have a curve that hangs toward the top left of the ROC space. Such a plot will have the lowest bias and minimum variance. The Elastic Net model had the least area of 68.7% bettered slightly by the LDA at 70%. Our best performing models based on the AUC criteria are GB, XGBoost and DT, all with AUC = 79.5% followed by the SVM with 79.4%. MLP had an AUC of 77.2% which is better than that of the ANN of 74.5% despite the ANN model having the better accuracy.

Ensemble methods have shown to be better predictive models for our Travel Insurance dataset. We can conclude that since XGBoost had the least misclassification rate, highest train and test accuracies, highest sensitivity, specificity, F1-score, AUC value and the second highest Precision rate, it is our best performing machine learning model for our data set.

Table 2: *Confusion matrix output results for full model.*

Model	True +	False +	False –	True –	Test total	Hyper-parameters tuned
Logistic	294	25	84	93	496	
k-NN	293	26	78	99	496	k=18
Naïve Bayes	307	12	91	86	496	laplace = 0.5, usekernel = TRUE, adjust = 0.5
LDA	297	22	94	83	496	
QDA	287	32	78	99	496	
SDA	295	24	83	94	496	$\Lambda = 0.5$
PLS	282	37	72	105	496	ncomp=2
Ridge	295	24	91	86	496	$\alpha = 0, \lambda = 0.02782559$
Lasso	296	23	91	86	496	$\alpha = 1, \lambda = 0.01$
Elastic Net	312	7	107	70	496	$\alpha = 0.1668101, \lambda = 0.3764936$

GBM	307	12	66	111	496	n.trees=50, interaction.depth=3, shrink=3, eta=0.3
XGBoost	311	8	66	111	496	gamma=0, nrounds=100, max_depth=2, eta=0.3
SVM	308	11	73	104	496	kernel=polynomial, degree=3, scale=0.1, 694 support vectors, cost=2
DT	307	12	66	111	496	
RF	311	8	70	107	496	ntree=200, mtry = 2
ANN	302	17	65	112	496	size=2, decay=0.29
MLP	301	18	68	109	496	layer1 = 10, layer2 =10, layer3 = 10

*Positive Class is "No". Negative Class is "Yes"

Table 3: *Performance results of full model.*

Model	**Accuracy %**		**95% CI* %**	**Sensitivity %**	**Specificity %**	**Precision %**	**F Score %**	**AUC %**
	Train	**Test**	**Test**	**Test**	**Test**	**Test**	**Test**	**Test**
Logistic	76.4	78.0	(74.1, 81.6)	77.8	78.8	92.2	84.4	72.1
k-NN	78.6	79.0	(75.2, 82.5)	79.0	79.2	91.8	84.9	72.5
Naïve Bayes	78.6	79.2	(75.4, 82.7)	77.1	87.8	96.2	85.6	72.4
LDA	76.0	76.6	(72.6, 80.3)	76.0	79.0	93.1	83.7	70.0
QDA	76.6	77.8	(73.9, 81.4)	78.6	75.6	90.0	83.9	73.0
SDA	76.4	78.4	(74.5, 82.0)	78.0	79.7	92.5	84.6	72.8
PLS	75.2	78.0	(74.1, 81.6)	79.7	73.9	88.4	83.8	73.9
Ridge	76.6	76.8	(72.9, 80.5)	76.4	78.2	92.5	83.7	70.5
Lasso	76.1	77.0	(73.1, 80.7)	76.5	78.9	92.8	83.9	70.7
Elastic Net	76.7	77.0	(73.1, 80,7)	74.5	90.9	97.8	84.6	68.7
GBM	82.7	84.3	(80.8, 87.4)	82.3	90.2	96.2	88.7	79.5
XGBoost	82.8	85.1	(81.6, 88.1)	82.5	93.3	97.5	89.4	79.5
SVM	81.6	83.1	(79.5, 86.3)	80.8	90.4	96.6	88.0	79.4
DT	82.7	84.3	(80.8, 87.4)	82.3	90.2	96.2	88.7	79.5
RF	82.6	84.3	(80.8, 87.4)	81.6	93.0	97.5	88.9	79.0
ANN	80.6	83.5	(79.9, 86.6)	82.3	86.8	94.7	88.0	74.5
MLP	80.8	82.7	[redacted]	81.6	85.8	94.4	87.5	77.1

*95% confidence interval of test data accuracy. Sensitivity= TPR. No values for blacked-out box.

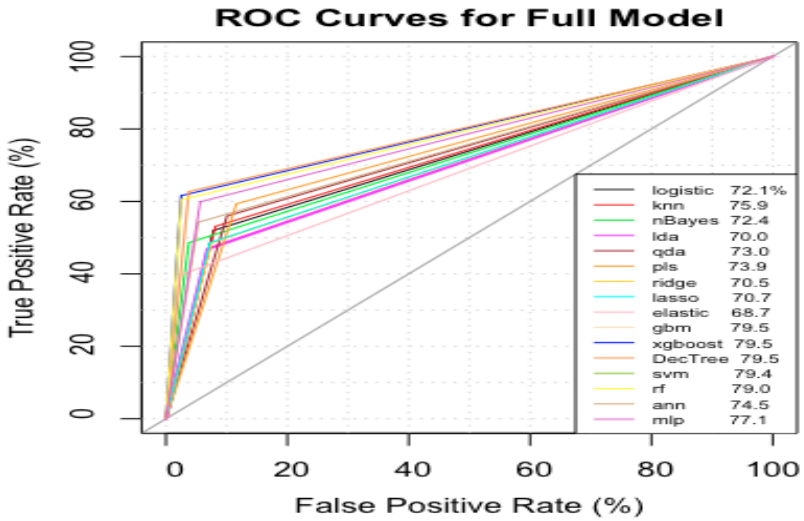


Figure 3: *ROC curves for full models.*

4.2. Feature selection and variable importance

During the analysis of our predictive models on the full dataset, we noticed that certain variables were very vital for prediction, while others were not so needed. This was first evident to us when we run the classification using Lasso and Elastic Net classifiers. Both automatically performed variable selection on our predictors before producing the accuracy outputs presented earlier. Lasso eliminated variables Chronic Disease, Graduate Or Not, Employment Type, and Family Members. The Ridge classifier shrunk the coefficients of Chronic Disease, Employment Type, and Graduate Or Not toward zero, portraying their less significance for that model. Elastic Net removed variables Age and Family Members in addition to Chronic Disease, Employment Type, and Graduate Or Not. A table of the coefficients of the regressions can be found in Appendix D2. Lasso and Elastic Net produced slightly better accuracy, specificity, precision, and F1 score than Ridge after variable selection.

Based on this and what had been done in previous literature [37-42]. We decided to consider variable importance for all our models. Aside from boosting the accuracy of our models, feature selection helped reduce the run time of some of our predictive models considerably and boosted the accuracy of some of the models considered, particularly the most important ones from the original model. Figure 4 shows variable importance plots for selected models. These are the most and least important predictors: Logistic (best – Ever Traveled Abroad, Annual Income, Age; worst – Chronic Disease, Graduate Or Not, Employment Type), RF and XGBoost (best–Annual Income, Age, Ever Traveled Abroad; worst–Chronic Disease, Graduate Or Not, Employment Type, Frequent Flyer), ANN (best – Annual Income, Age, Ever Traveled Abroad, Family Members), Ridge, Lasso and Elastic Net has been discussed. Conclusively, the best model is one that includes a combination of Ever Traveled Abroad, Annual Income, and Age while excluding either any or a combination of Chronic Disease, Graduate Or Not, or Employment Type.

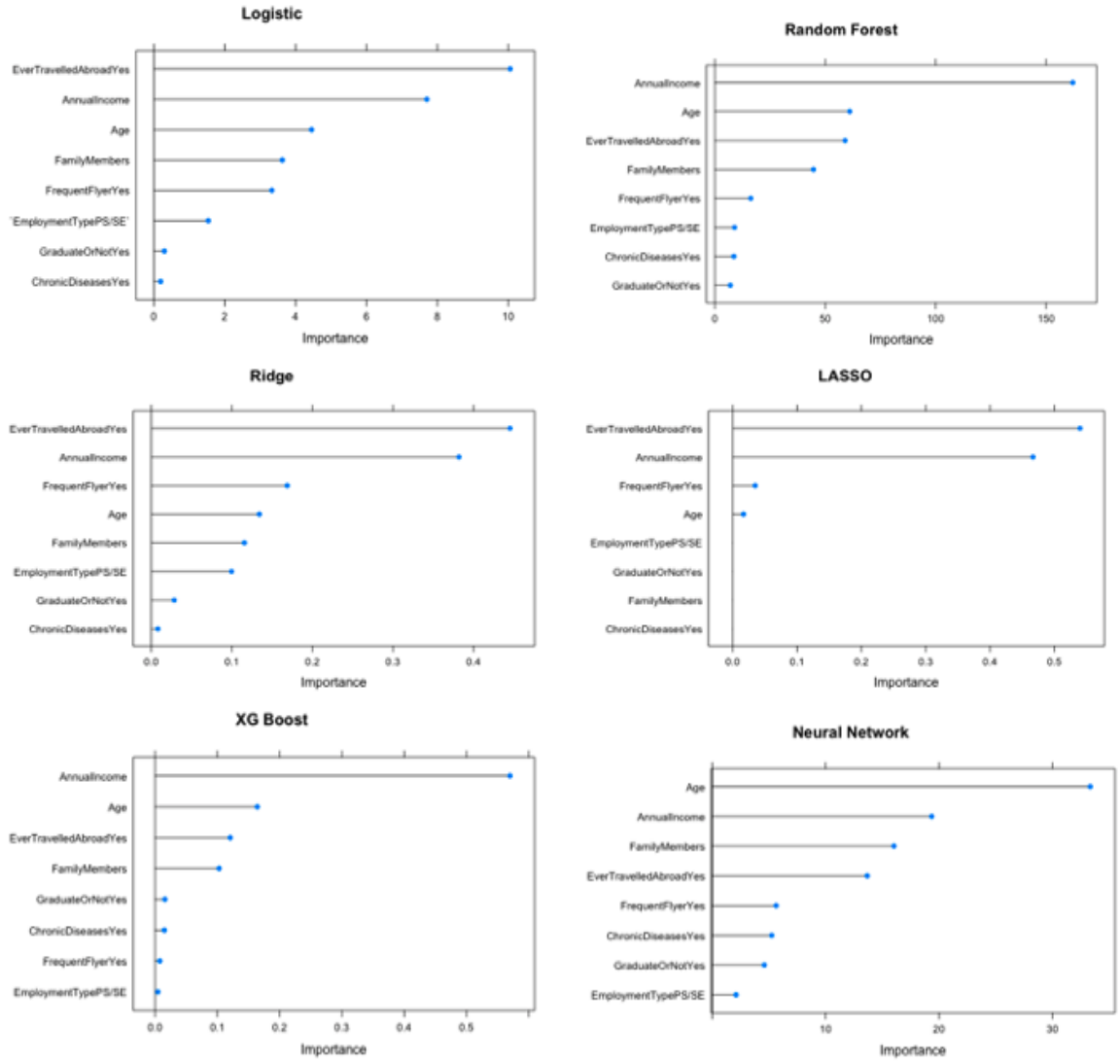


Figure 4: *Variable importance plots.*

4.2.1. Subset selection algorithm

The three widely used subset selection techniques, forward and backward stepwise and best subset selection, were also considered to aid in selecting the most important variables for a reduced model analysis. Forward stepwise selection starts with a model containing only the intercept and then adds predictors to the model, one at a time until all the predictors are in the model. The model adds the variable that gives the greatest additional improvement to the fit at each step and stops the iteration when it has sufficient predictors that give the lowest AIC value or the highest adjusted R-squared. The backward stepwise selection algorithm begins with the full least squares model containing all predictors (p=8), and then iteratively removes the least useful predictor one at a time until the optimal AIC is reached for the sufficient variables. Under the best subset selection algorithm, we fit a separate least squares regression for each possible combination of the 8 predictors and select the best model from among all the 2^p combinations, in our case 2^8 =256 possibilities.

Results of our subset selection algorithm are presented in Appendix F. All three methods chose all the variables to enter the model at the same time. Ever Traveled Abroad is seen as the most important contributory variable and so chosen first, followed by Annual Income and

then Family Members followed by Age. Variables Frequent Flyer and Graduate Or Not are the 5th and 6th chosen to enter the model. Chronic Disease and Employment Type are the final two (7th and 8th respectively) variables chosen to enter the model. This implies that these variables contribute less information to our model than the others. Using the adjusted R-squared criteria, all three algorithms conclusively chose only the first 6 variables to be used for the model. This means we can do without the variables Chronic Disease and Employment Type and the model will be very well explained by the remaining variables. Statistically, these variables added minimal incremental explanatory power and negligible improvement in adjusted R-squared. From a practical standpoint, Chronic Disease may have limited relevance in predicting travel insurance purchases, as it could influence decisions related to health insurance more directly than travel-specific policies. Similarly, Employment Type likely introduces noise rather than meaningful variation, given its indirect relationship with travel behavior compared to variables like income and travel history.

In order not to overfit the model any further and for the sake of parsimony, we used these six remaining variables to test the performance metrics of all our predictive models considered in Table 3. Note that variable importance plots on the full model of some of the machine learners also confirm this. We will call these new models Reduced Models.

Although the dataset exhibits class imbalance, the models evaluated, including the reduced models, showed improved predictive performance over the No Information Rate (NIR) of 64%, confirming their robustness despite the imbalance. We did not see improvements by using either oversampling or under sampling techniques, suggesting that the models, as developed, inherently account for the imbalance. The accuracy of all models considered exceeded the NIR threshold, with significant improvements in test accuracy, as evidenced by p-values < 0.000 in our hypothesis tests.

4.2.2. The reduced models

The optimal parameters for each predictive model changed. KNN now needed only 16 nearest neighbors to help with response predictor detection. RF still used mtry of 2 but with an increased ntree of 500 to achieve the optimum accuracy. ANN size increased from 2 to 5 and sacrificed some decay from 0.29 to 0.05. MLP, Naïve Bayes and SDA still used the same parameters to get their optimal results.

The XGBoost model still performs the best on the full data with fewer predictors, even though there was no change in values from test accuracy, through sensitivity to AUC. The only improvement achieved by the XGBoost technique was on the train accuracy, increasing by +0.2% to 83.0%. KNN had the biggest increase in test accuracy by +1.8% followed by ANN and Ridge with +1.1% and +1.0% respectively. In addition to XGBoost, the test accuracies of GBM, DT, PLS, Elastic Net, and Naïve Bayes did not change. SDA, QDA and Logistic were the only predictive models that declined in test accuracies by -0.6, -0.2 and -0.2 respectively. In general, the test accuracy of the reduced models was slightly better than that of the full models with all the predictors. On average, both sensitivity and specificity increased. ANN had the highest increase for sensitivity (+2.8%) and the highest decrease for specificity (-5.0%). All train error rates decreased or remained the same but never increased, after variable selection.

The AUC for most of the models did not change, especially the ensemble methods. No model had a decrease in AUC value, but notable increases are associated with KNN (+3.4%), ANN (+1.7%), LDA (+1.0%), and RF (+0.7%).

We managed to decrease the runtime of our full model and managed a very slight increase in out-of-sample accuracy. It has been shown that variable selection on the most important variables leads to marginal increases in accuracy of predictions for models, of which predictive models are no exception [37-39].

Table 4: *Confusion matrix output results for reduced model.*

Model	True +	False +	False –	True –	Test Total	Hyper- parameters tuned
Logistic	294	25	85	92	496	
k-NN	292	27	68	109	496	k=16
Naïve Bayes	307	12	91	86	496	Laplace = 0.5, usekernel = TRUE, adjust = 0.5
LDA	296	23	90	87	496	
QDA	288	31	80	97	496	
SDA	294	25	85	92	496	$\Lambda = 0.5$
PLS	282	37	72	105	496	ncomp=2
Ridge	299	20	90	87	496	$A = 0, \lambda = 0.08497534$
Lasso	297	22	91	86	496	$A = 1, \lambda = 0.01$
Elastic Net	312	7	107	70	496	$A = 0.1668101, \lambda = 0.3764936$
GBM	307	12	66	111	496	n.trees=25, interaction.depth=4, shrink=0.15, eta=0.3
XGBoost	311	8	66	111	496	gamma=0, nrounds=50, max_depth=2, eta=0.3
SVM	308	11	67	110	496	kernel=polynomial, degree=3, scale=1, 82 support vectors, cost=0.25
DT	307	12	66	111	496	
RF	310	9	67	110	496	ntree=500,mtry = 2
ANN	291	28	65	112	496	size=5, decay=0.05
MLP	301	18	68	109	496	layer1 = 10 layer2 =10, layer3 = 10

*Positive Class is "No".

Table 5: *Performance results of reduced model.*

Model	Accuracy %		95% CI* %	Sensitivity %	Specificity %	Precision %	F Score %	AUC %
	Train (Delta)	Test (Delta)	Test (Delta)	Test (Delta)	Test (Delta)	Test (Delta)	Test (Delta)	Test (Delta)
Logistic	76.4 (0.0)	77.8 (-0.2)	(73.9, 81.4)	77.6 (-0.2)	78.6 (-0.2)	92.2 (0.0)	84.2 (-0.2)	72.1 (0.0)
k-NN	80.5 (+1.9)	80.8 (+1.8)	(77.1, 84.2)	81.1 (+2.1)	80.1 (+0.9)	91.5 (-0.3)	86.0 (+1.1)	75.9 (+3.4)

Naïve Bayes	78.9 (+0.3)	79.2 (0.0)	(75.4, 82.7)	77.1 (0.0)	87.8 (0.0)	96.2 (0.0)	85.6 (0.0)	72.4 (0.0)
LDA	76.1 (+0.1)	77.2 (+0.6)	(73.3, 80.8)	76.7 (+0.7)	79.1 (+0.1)	92.8 (-0.3)	84.0 (+0.3)	71.0 (+1.0)
QDA	76.6 (0.0)	77.6 (-0.2)	(73.7, 81.2)	78.3 (-0.3)	75.8 (+0.2)	90.3 (+0.3)	83.8 (-0.1)	72.5 (-0.5)
SDA	76.6 (+0.2)	77.8 (-0.6)	(73.9, 81.4)	77.6 (-0.4)	78.6 (-1.1)	92.2 (-0.3)	84.2 (-0.4)	72.1 (-0.7)
PLS	75.2 (0.0)	78.0 (0.0)	(74.1, 81.6)	79.7 (0.0)	73.9 (0.0)	88.4 (0.0)	83.8 (0.0)	73.9 (0.0)
Ridge	76.7 (+0.1)	77.8 (+1.0)	(73.9, 81.4)	76.9 (+0.5)	81.3 (+3.1)	93.7 (+1.2)	84.5 (+0.3)	71.4 (+0.2)
Lasso	76.2 (+0.1)	77.2 (+0.2)	(73.3, 80.8)	76.5 (0.0)	79.6 (+0.7)	93.1 (+0.3)	84.0 (+0.1)	70.9 (0.0)
Elastic Net	76.7 (0.0)	77.0 (0.0)	(73.1, 80.7)	74.5 (0.0)	90.9 (0.0)	97.8 (0.0)	84.6 (0.0)	68.7 (0.0)
GBM	82.8 (0.0)	84.3 (0.0)	(80.8, 87.4)	82.3 (0.0)	90.2 (0.0)	96.2 (0.0)	88.7 (0.0)	79.5 (0.0)
XGBoost	83.0 (+0.2)	85.1 (0.0)	(81.8, 88.1)	82.5 (0.0)	93.3 (0.0)	97.5 (0.0)	89.4 (0.0)	79.5 (0.0)
SVM	82.2 (+0.6)	84.3 (+1.2)	(80.8, 87.4)	82.1 (+1.3)	90.9 (+0.5)	96.6 (0.0)	88.8 (+0.8)	79.4 (0.0)
DT	82.7 (0.0)	84.3 (0.0)	(80.8, 87.4)	82.3 (0.0)	90.2 (0.0)	96.2 (0.0)	88.7 (0.0)	79.5 (0.0)
RF	82.9 (+0.3)	84.7 (+0.4)	(81.2, 87.7)	82.2 (+0.6)	92.4 (-0.6)	97.2 (-0.3)	89.1 (+0.2)	79.7 (+0.7)
ANN	80.2 (-0.4)	81.3 (+1.1)	(76.9, 84.0)	81.7 (+2.8)	80.0 (-5.0)	91.2 (-3.5)	86.2 (+0.2)	76.2 (+1.7)
MLP	81.5 (+0.7)	82.7 (+0.6)	[redacted]	81.6 (+0.7)	85.8 (+0.3)	94.4 (0.0)	87.5 (+0.4)	78.0 (+0.9)

*95% confidence interval of test data accuracy. Sensitivity = Recall = TPR. Parenthesis values shows the delta from comparing parameter values to the full model.

4.3. Marginal effect of variables

Controlling for all other features, how does one feature, say Age, affect the final predictive probability of purchasing travel insurance? To check the consistency of the directional impact of the important variables per model, we employed the partial dependent (PD) plot analysis. In addition to depicting the marginal effect of the features on the predicted model outcomes, the PD plot can detect whether a variable is linearly or non-linearly related to the response [43]. The PD plots are exhibited in Appendix H for the XGBoost and Random Forest models. As expected, the behavior of each feature to the response variable, Travel Insurance, is non-linear just like the supervised models which generated these relationships. For both models and the others not shown, the probability of purchasing travel insurance increases with age,

peaks at age 28, and decreases to age 30 before increasing one year later after which it decreases thereafter. The Family Members feature interpretation follows similarly: the more members one has in the family up to about 5 members, the more likely one will buy travel insurance. The probability decreases thereafter as the number increases and only experiences a slight rise as the family count increases to 7 or 8. A disadvantage of using the PD plot analysis is that it assumes the features are independent, which is inconsistent with reality.

5. Conclusion

This study explored the application of machine learning algorithms to predict travel insurance purchases, offering significant contributions to the insurance and aviation sectors. By identifying key predictors such as age, income, travel history, and graduate status, the study demonstrated how robust classifiers like XGBoost, random forests, and gradient boosting outperform traditional models like logistic regression in predictive accuracy and reliability. These findings provide actionable insights for insurers to develop personalized insurance policies, enhance customer targeting, and optimize marketing strategies. Moreover, airline companies can leverage these insights to design tailored travel insurance advertisements, increasing customer conversion rates. Travelers, in turn, benefit from affordable and need-specific policies, fostering a more inclusive insurance market.

Beyond practical applications, this research underscores the importance of variable selection in predictive modeling. By focusing on parsimonious models that prioritize the most relevant variables, it reduces overfitting and enhances interpretability, ensuring robust predictions across diverse datasets.

While this study provides valuable insights, it acknowledges limitations in data recency and scope. The dataset, although sufficient, would benefit from more recent, comprehensive data that incorporates additional intuitive variables such as flight costs, insurance premiums, domestic versus international travel, and socio-economic factors. Future studies could address these gaps by analyzing data spanning pre- and post-COVID-19 periods, enabling better comparisons of travel insurance behavior over time. Expanding the scope to include insurance for other transportation modes, such as car rentals or cruises, presents another promising direction for future research.

This research highlights the transformative potential of machine learning in insurance modeling, offering a robust framework for optimizing risk assessment and operational efficiency. By bridging gaps in existing literature, it paves the way for more inclusive, data-driven approaches in the travel insurance industry.

References

1. Al Mamun A, Rahman MK, Yang Q, et al. Predicting the willingness and purchase of travel insurance during the COVID-19 pandemic. Front Public Health. 2022;10:907005.

2. Pickup L, Bowater S, Thorne S, et al. Travel insurance in adult congenital heart disease—do they declare their condition? Int J Cardiol. 2016;223:316-7.

3. Hung KK, Lin AK, Cheng CK, et al. Travel health risk perceptions and preparations among travelers at Hong Kong International Airport. J Travel Med. 2014;21:288-91.

4. Karl M. Risk and uncertainty in travel decision-making: tourist and destination perspective. J Travel Res. 2018;57:129-46.

5. Kerr G, Kelly L. Travel insurance: the attributes, consequences, and values of using travel insurance as a risk-reduction strategy. J Travel Tour Mark. 2019;36:191-203.

6. Althati C, Perumalsamy J, Konidena BK. Enhancing life insurance risk models with ai: predictive analytics, data integration, and real-world applications. J Artif Intell Res Appli. 2023;3:448-86.

7. Billa MM, Nagpal T. Medical insurance price prediction using machine learning. J Electr Syst. 2024;20:2270-9.

8. Hosein P. A data science approach to risk assessment for automobile insurance policies. Int J Data Sci Anal. 2024;17:127-38.

9. Jones KI, Sah S. The implementation of machine learning in the insurance industry with big data analytics. Int J Data Informatics Intell Computing. 2023;2:21-38.

10. Terekhov MA, Demirezen EM, Aytug H. Business analytics: emerging practice and research issues in the health insurance industry. Prod Oper Manag. 2024;33:432-55.

11. Boodhun N, Jayabalan M. Risk prediction in life insurance industry using supervised learning algorithms. Complex Intell Syst. 2018;4:145-54.

12. Hanafy M, Ming R. Machine learning approaches for auto insurance big data. Risks. 2021;9:42.

13. Mehrabi N, Goyal P, Verma A, et al. Resolving ambiguities in text-to-image generative models. The 61st Annual Meeting of the Association for Computational Linguistics, Toronto, Canada. 2023:pp.14367-88.

14. Perumalsamy AAJ, Krothapalli B, Althati C. Machine learning algorithms for customer segmentation and personalized marketing in life insurance: a comprehensive analysis. J Artif Intell Res. 2022;2:83-123.

15. Nimmagadda VS. Artificial intelligence for customer behavior analysis in insurance: advanced models, techniques, and real-world applications. J AI Healthcare Med. 2022;2:227-63.

16. Turban E, Sharda R, Delen D. Decision support and business intelligence system (9thedn). Pearson, New Jersey, United State of America. 2011;23:2020.

17. Yeh IC, Lien CH. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert syst Appl. 2009;36:2473-80.

18. Leach J, Thayasivam U. Optimizing data evaluation metrics for fraud detection using machine learning. Int J Math Comput Sci. 2022;1:52-9.

19. Albashrawi M. Detecting financial fraud using data mining techniques: a decade review from 2004 to 2015. J Data Sci. 2016;14:553-69.

20. Bhattacharyya S, Jha S, Tharakunnel K, et al. Data mining for credit card fraud: a comparative study. Decis Support syst. 2011;50:602-13.

21. Ţitan E, Tudor AI. Conceptual and statistical issues regarding the probability of default and modeling default risk. Database Syst J. 2011;2:13-22.

22. Wu J, Xiong H, Chen J. COG: local decomposition for rare class analysis. Data Min Knowl Discov. 2010;20:191-220.

23. Whitrow C, Hand DJ, Juszczak P, et al. Transaction aggregation as a strategy for credit card fraud detection. Data Min Knowl Discov. 2009;18:30-55.

24. Panigrahi S, Kundu A, Sural S, et al. Credit card fraud detection: a fusion approach using Dempster–Shafer theory and Bayesian learning. Inform Fusion. 2009;10:354-63.

25. Statnikov A, Wang L, Aliferis CF. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC bioinformatics. 2008;9:1-10.

26. Chen RC, Chen TS, Lin CC. A new binary support vector system for increasing detection rate of credit card fraud. Int J Pattern Recognit Artif Intell. 2006;20:227-39.

27. Kou Y, Lu CT, Sirwongwattana S, et al. Survey of fraud detection techniques. IEEE international conference on networking, sensing and control. Marseille, France. 2004.

28. Brause R, Langsdorf T, Hepp M. Neural data mining for credit card fraud detection. 11th International Conference on Tools with Artificial Intelligence. Kyoto, Japan. 2025.

29. Weerasinghe KP, Wijegunasekara MC. A comparative study of data mining algorithms in the prediction of auto insurance claims. Eur Int J Sci Tech. 2016;5:47-54.

30. Wüthrich MV. Machine learning in individual claims reserving. Scand Actuar J. 2018;2018:465-80.

31. Tsai CF. Feature selection in bankruptcy prediction. Knowl-Based Syst. 2009;22:120-7.

32. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. 22nd acm sigkdd international conference on knowledge discovery and data mining, New York, USA. 2016.

33. James G, Witten D, Hastie T. An introduction to statistical learning. Springer Publishing, New York, USA. 2021.

34. Breiman L. Random forests. Mach Learn. 2001;45:5-32.

35. Bedia C, Tauler R, Jaumot J. Introduction to the data analysis relevance in the omic era. Compr Anal Chem. 2018;82:1-12.

36. Ling CX, Huang J, Zhang H. AUC: a better measure than accuracy in comparing learning algorithms. Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI. Halifax, Canada. 2003.

37. Li Y, Li C, Li M, et al. Influence of variable selection and forest type on forest aboveground biomass estimation using machine learning algorithms. Forests. 2019;10:1073.

38. Speiser JL, Miller ME, Tooze J, et al. A comparison of random forest variable selection methods for classification prediction modeling. Expert Syst Appl. 2019;134:93-101.

39. Kim HH, Swanson NR. Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. Int J Forecast. 2018;34:339-54.

40. Sanchez-Pinto LN, Venable LR, Fahrenbach J, et al. Comparison of variable selection methods for clinical predictive modeling. Int J Med Inform. 2018;116:10-7.

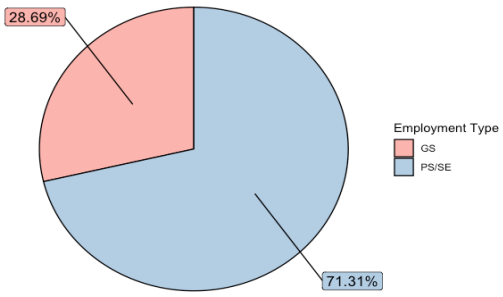
41. Tyralis H, Papacharalampous G. Variable selection in time series forecasting using random forests. Algorithms. 2017;10:114.

42. Rakotomamonjy A. Variable selection using SVM-based criteria. J Mach Learn Res. 2003;3:1357-70.

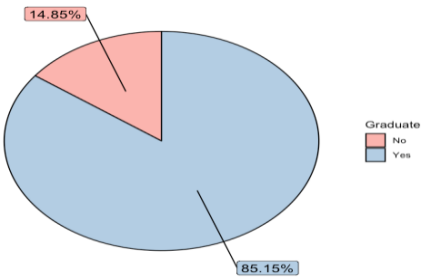
43. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001:1189-232.

APPENDIX

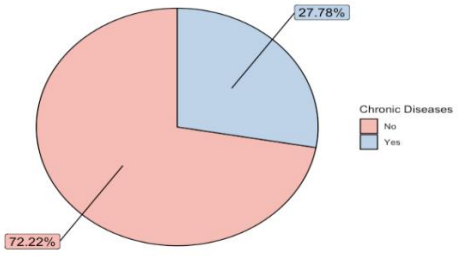
A1. Employment Type



A2. Graduate Status



A3. Chronic Diseases



A4. Frequent Flyer

A5. Ever Traveled Abroad?

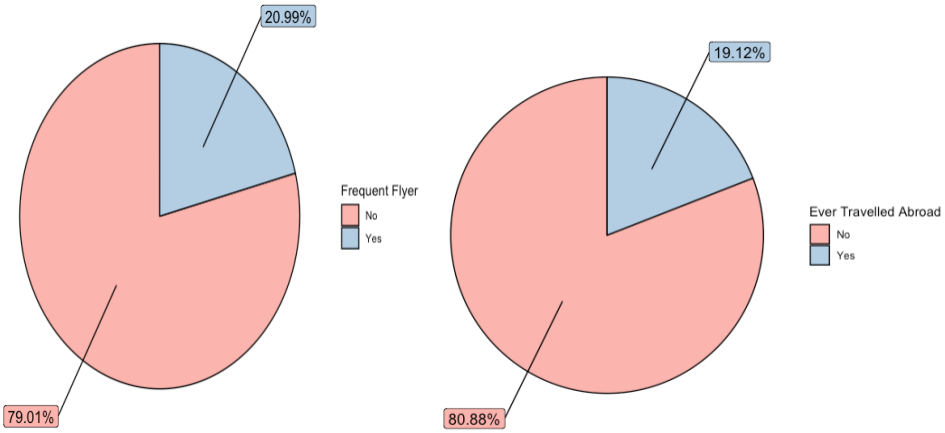


Figure A: *Plots of the distribution of other data variables.*

Table B1: *Performance results of predictive models of full model.*

Model	Accuracy %			95% CI*	Sensitivity %	Specificity %	Precision %	Recall %	F %	AUC %
	Train	Test	Balanced	Test	Test	Test	Test	Test	Test	Test
Logistic	76.4	78.0	78.3	(74.1, 81.6)	77.8	78.8	92.2	77.8	84.4	72.1
k-NN	78.6	79.0	79.1	(75.2, 82.5)	79.0	79.2	91.8	79.0	84.9	72.5
Naïve Bayes	78.6	79.2	82.5	(75.4, 82.7)	77.1	87.8	96.2	77.1	85.6	72.4
LDA	76.0	76.6	76.6	(72.6, 80.3)	76.0	79.0	93.1	76.0	83.7	70.0
QDA	76.6	77.8	77.1	(73.9, 81.4)	78.6	75.6	90.0	78.6	83.9	73.0
PDA	76.0	76.6	77.5	(72.6, 80.3)	76.0	79.0	93.1	76.0	83.7	70.0
SDA	76.4	78.4	78.9	(74.5, 82.0)	78.0	79.7	92.5	78.0	84.6	72.8
PLS	75.2	78.0	76.8	(74.1, 81.6)	79.7	73.9	88.4	79.7	83.8	73.9
Bayes GLM	76.4	78.0	78.3	(74.1, 81.6)	77.8	78.8	92.2	77.8	84.4	72.4
Ridge	76.6	76.8	77.3	(72.9, 80.5)	76.4	78.2	92.5	76.4	83.7	70.5
Lasso	76.1	77.0	77.7	(73.1, 80.7)	76.5	78.9	92.8	76.5	83.9	70.7
Elastic Net	76.7	77.0	82.7	(73.1, 80,7)	74.5	90.9	97.8	74.5	84.6	68.7

GBM	82.7	84.3	86.3	(80.8, 87.4)	82.3	90.2	96.2	82.3	88.7	79.5
XGBoost	82.8	85.1	87.9	(81.6, 88.1)	82.5	93.3	97.5	82.5	89.4	79.5
SVM	81.6	83.1	85.6	(79.5, 86.3)	80.8	90.4	96.6	80.8	88.0	79.4
DT	82.7	84.3	86.3	(80.8, 87.4)	82.3	90.2	96.2	82.3	88.7	79.5
RF	82.6	84.3	87.3	(80.8, 87.4)	81.6	93.0	97.5	81.6	88.9	79.0
ANN	80.6	83.5	84.6	(79.9, 86.6)	82.3	86.8	94.7	82.3	88.0	74.5
MLP	80.8	82.1	81.4		80.9	85.5	94.4	80.9	87.1	77.1

*95% confidence interval of test data accuracy. Sensitivity = Recall = TPR. No values for blacked-out square.

Table B2: Confusion *matrix output and tuned parameters of full model.*

Model	True +	False +	False –	True –	Test total	Hyper- parameters tuned
Logistic	294	25	84	93	496	
k-NN	293	26	78	99	496	k=18
Naïve Bayes	307	12	91	86	496	Laplace = 0.5, usekernel = TRUE, adjust = 0.5
LDA	297	22	94	83	496	
QDA	287	32	78	99	496	
PDA	297	22	94	83	496	$\Lambda = 0$
SDA	295	24	83	94	496	$\Lambda = 0.5$
PLS	282	37	72	105	496	ncomp=2
Bayes GLM	294	25	84	93	496	
Ridge	295	24	91	86	496	$A = 0, \lambda = 0.02782559$
Lasso	296	23	91	86	496	$A = 1, \lambda = 0.01$
Elastic Net	312	7	107	70	496	$A = 0.1668101, \lambda = 0.3764936$
GBM	307	12	66	111	496	n.trees=50, interaction.depth=3, shrinkage=3, eta=0.3, n.minobs=10
XGBoost	311	8	66	111	496	Gamma=0, nrounds=100, max_depth=2, eta=0.3, colsample_bytree=0.6
SVM	308	11	73	104	496	kernel=polynomial, degree=3, scale=0.1, 694 support vectors, cost=2
DT	307	12	66	111	496	
RF	311	8	70	107	496	ntree=200, mtry = 2
ANN	302	17	65	112	496	size=2, decay=0.29

MLP	301	18	71	106	496	layer1 = 10 layer2 =10, layer3 = 10
Positive Class is "No"						

Table B3: *Performance results of predictive models of reduced model.*

Model	Accuracy %			95% CI*	Sensitivity %	Specificity %	Precision %	Recall %	F %	AUC %
	Train	Test	Balanced	Test	Test	Test	Test	Test	Test	Test
Logistic	76.4	77.8	78.1	(73.9, 81.4)	77.6	78.6	92.2	77.6	84.2	72.1
k-NN	80.5	80.8	80.6	(77.1, 84.2)	81.1	80.1	91.5	81.1	86	75.9
Naïve Bayes	78.9	79.2	82.4	(75.4, 82.7)	77.1	87.8	96.2	77.1	85.6	72.4
LDA	76.1	77.2	77.9	(73.3, 80.8)	76.7	79.1	92.8	76.7	84	71.0
QDA	76.6	77.6	77.0	(73.7, 81.2)	78.3	75.8	90.3	78.3	83.8	72.5
PDA	76.1	77.2	77.9	(73.3, 80.8)	76.7	79.1	92.8	76.7	84	71.0
SDA	76.6	77.8	78.1	(73.9, 81.4)	77.6	78.6	92.2	77.6	84.2	72.1
PLS	75.2	78.0	76.8	(74.1, 81.6)	79.7	73.9	88.4	79.7	83.8	73.9
Bayes GLM	76.4	77.8	78.1	(73.9, 81.4)	77.6	78.6	92.2	77.6	84.2	72.1
Ridge	76.7	77.8	79.1	(73.9, 81.4)	76.9	81.3	93.7	76.9	84.5	71.4
Lasso	76.2	77.2	78.1	(73.3, 80.8)	76.5	79.6	93.1	76.5	84.0	70.9
Elastic Net	76.7	77.0	82.7	(73.1, 80.7)	74.5	90.9	97.8	74.5	84.6	68.7
GBM	82.8	84.3	86.3	(80.8, 87.4)	82.3	90.2	96.2	82.3	88.7	79.5
XGBoost	83.0	85.1	87.9	(81.8, 88.1)	82.5	93.3	97.5	82.5	89.4	79.5
SVM	82.2	84.3	86.5	(80.8, 87.4)	82.1	90.9	96.6	90.9	88.8	79.4
DT	82.7	84.3	86.3	(80.8, 87.4)	82.3	90.2	96.2	82.3	88.7	79.5
RF	82.9	84.7	87.3	(81.2, 87.7)	82.2	92.4	97.2	82.2	89.1	79.7
ANN	80.2	81.3	80.9	(77.5, 84.6)	81.7	80.0	91.2	81.7	86.2	76.2
MLP	81.5	82.7	82.0		81.6	85.8	94.4	81.6	87.5	78.0

*95% confidence interval of test data accuracy. Sensitivity = Recall = TPR. No values for blacked-out square.

Table B4: *Confusion matrix output and tuned parameters of reduced model.*

Model	True +	False +	False –	True –	Test Total	Optimal model parameters tuned
Logistic	294	25	85	92	496	
k-NN	292	27	68	109	496	k=16
Naïve Bayes	307	12	91	86	496	Laplace = 0.5, usekernel = TRUE, adjust = 0.5
LDA	296	23	90	87	496	
QDA	288	31	80	97	496	

PDA	296	23	90	87	496	$\Lambda = 0$
SDA	294	25	85	92	496	$\Lambda = 0.5$
PLS	282	37	72	105	496	ncomp=2
Bayes GLM	294	25	85	92	496	
Ridge	299	20	90	87	496	$\Lambda = 0$, $\lambda = 0.08497534$
Lasso	297	22	91	86	496	$\Lambda = 1$, $\lambda = 0.01$
Elastic Net	312	7	107	70	496	$\Lambda = 0.1668101$, $\lambda = 0.3764936$
GBM	307	12	66	111	496	n.trees=25, interaction.depth=4, shrinkage=0.15, eta=0.3, n.minobs=10
XGBoost	311	8	66	111	496	gamma=0, nrounds=50, max_depth=2, eta=0.3, colsample_bytree=0.8
SVM	308	11	67	110	496	kernel=polynomial, degree=3, scale=1, 82 support vectors, cost=0.25
DT	307	12	66	111	496	
RF	310	9	67	110	496	ntree=500,mtry = 2
ANN	291	28	65	112	496	size=5, decay=0.05
MLP	301	18	68	109	496	layer1 = 10 layer2 =10, layer3 = 10

Positive Class is "No"

Table B5: *Delta of model results.*

Model	Accuracy Delta			Sensitivity Delta	Specificity Delta	Precision Delta	Recall Delta	F Delta	AUC Delta
	Train	Test	Balanced	Test	Test	Test	Test	Test	Test
Logistic	0.0	-0.2	-0.2	-0.2	-0.2	0.0	-0.2	-0.2	0.0
k-NN	1.9	1.8	1.5	2.1	0.9	-0.3	2.1	1.1	3.4
Naïve Bayes	0.3	0.0	-0.1	0.0	0.0	0.0	0.0	0.0	0.0
LDA	0.1	0.6	1.3	0.7	0.1	-0.3	0.7	0.3	1.0
QDA	0.0	-0.2	-0.1	-0.3	0.2	0.3	-0.3	-0.1	-0.5
PDA	0.1	0.6	0.4	0.7	0.1	-0.3	0.7	0.3	1.0
SDA	0.2	-0.6	-0.8	-0.4	-1.1	-0.3	-0.4	-0.4	-0.7
PLS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bayes GLM	0.0	-0.2	-0.2	-0.2	-0.2	0.0	-0.2	-0.2	-0.3
Ridge	0.1	1.0	1.8	0.5	3.1	1.2	0.5	0.3	0.9
Lasso	0.1	0.2	0.4	0.0	0.7	0.3	0.0	0.1	0.2
Elastic Net	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GBM	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XGBoost	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
SVM	0.6	1.2	2.9	1.3	0.5	0.0	1.3	0.8	0.0
DT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RF	0.3	0.4	0.0	0.6	-0.6	-0.3	0.6	0.2	0.7
ANN	-0.4	1.1	-3.7	2.8	-5.0	-3.5	2.8	0.2	1.7
MLP	0.7	0.6	0.0	0.7	0.3	0.0	0.7	0.4	0.9

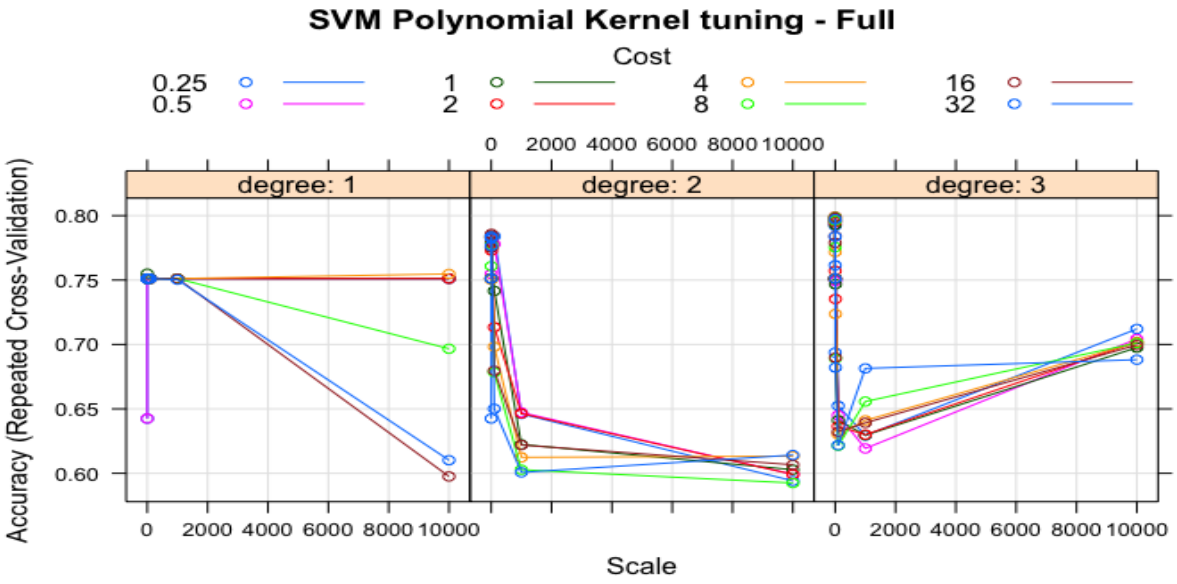


Figure C1: *Tuning for SVM model: Full Model uses C=62, degree=3, scale=0.1, 694 support vectors.*

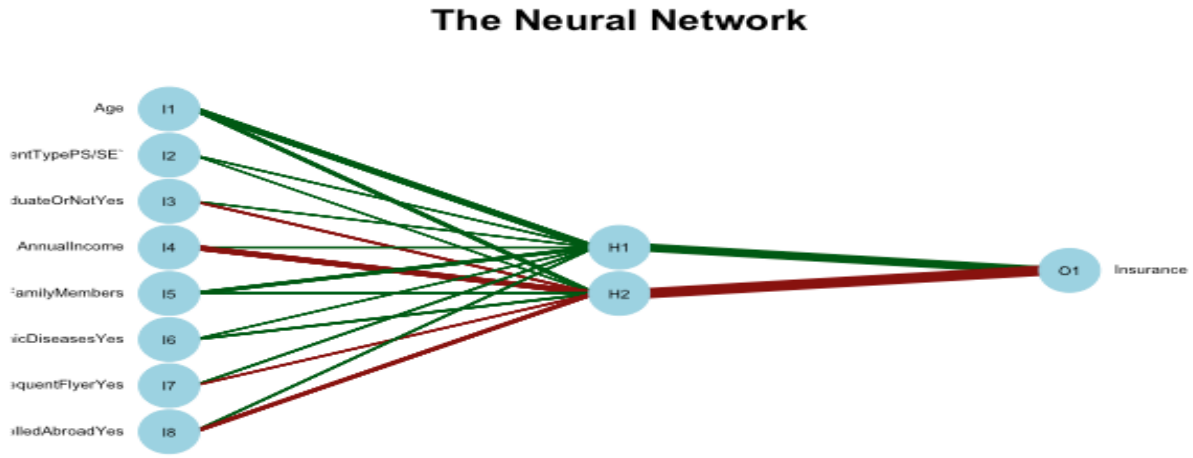


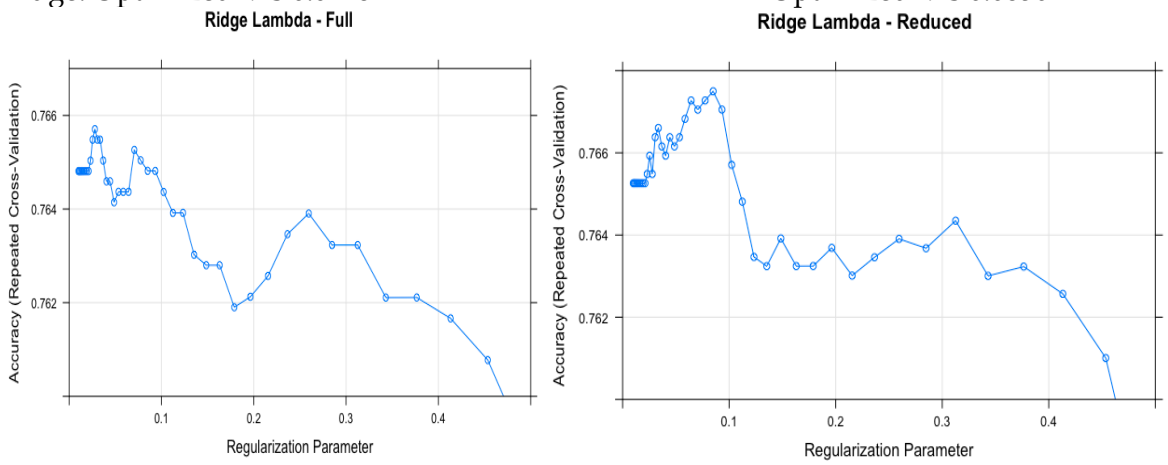
Figure. C2: *Graphical representation of the neural network.*

Full Model

Reduced Model

Ridge: Optimized λ is 0.0278

Optimized λ is 0.0850



Lasso: Optimized λ is 0.01

Optimized λ is 0.01

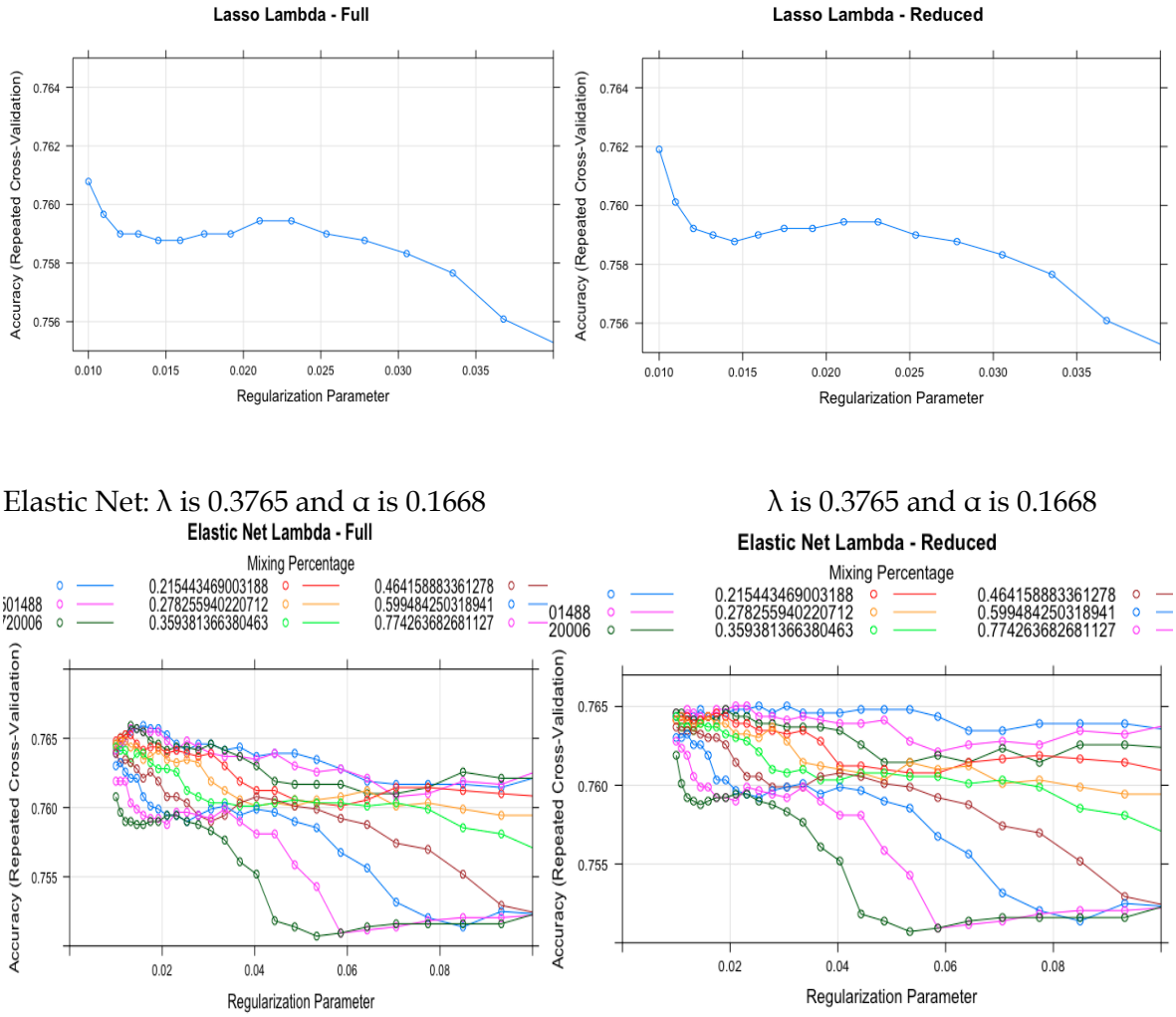


Figure. D2: *Coefficients of regularization models.*

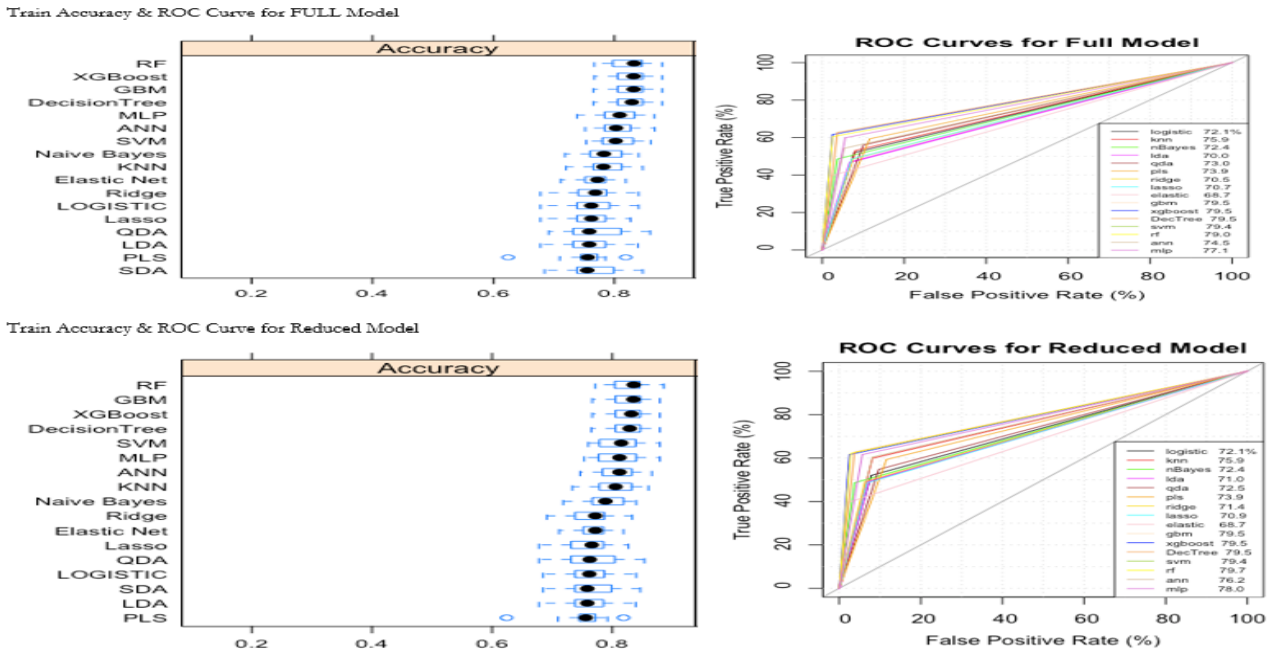


Figure E: *Box and whisker plot of train accuracy & ROC curves.*

All three methods (forward stepwise, backward stepwise and best subset selection) choose all the variables to enter the model at exactly the same time. Variable "Ever Traveled Abroad" is seen as the most important contributory variable and so chosen first, followed by "Annual Income" and then "Family Members" followed by "Age". Variables "Frequent Flyer" and "Graduate Or Not" are the 5th and 6th chosen to enter the model. "Chronic Disease" and "Employment Type" are the final two (7th and 8th respectively) variables chosen to enter the model. This implies that these variables contribute less information to our model than the others. Using the adjusted R-squared criteria, all three algorithms conclusively chose only the first 6 variables to be used for the model. This means we can do without the variables "Chronic Disease" and "Employment Type" and the model will be very well explained by the remaining variables. We used these six remaining variables in our reduced model to confirm this. Note that, variable importance plots on the full model of some of the machine learners also confirm this.

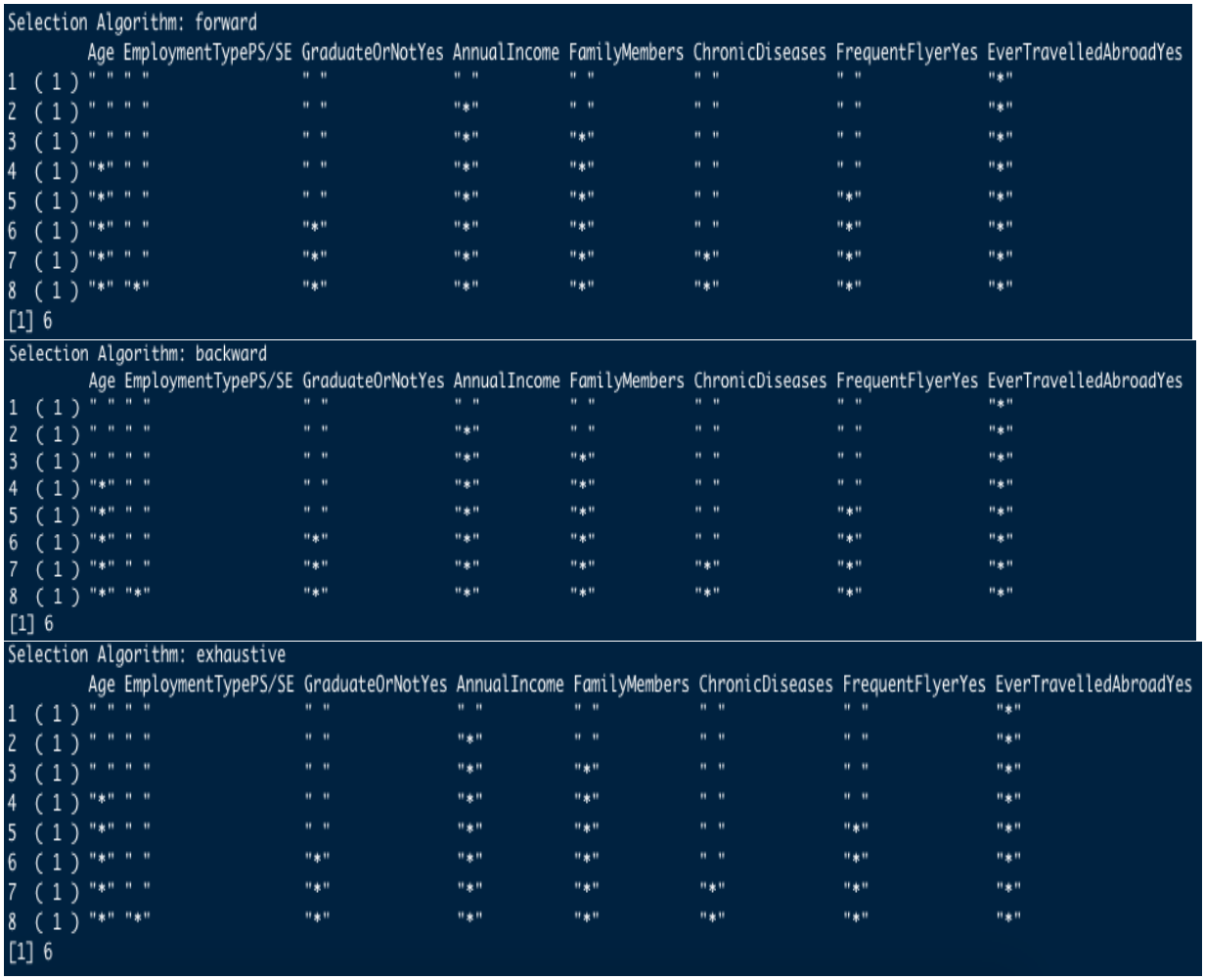
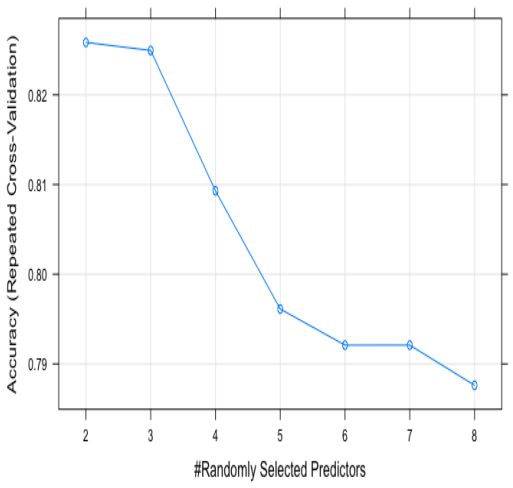


Figure F: *Subset selection algorithm results.*

Note: The higher the number of "stars" below a specific variable, the higher the importance to the model. For all three procedures, the best one-variable model is the model that contains only the Ever Traveled Abroad variable and the intercept. The best two-variable model includes Annual Income to the one-variable model, in that order based on the total number of stars a variable.

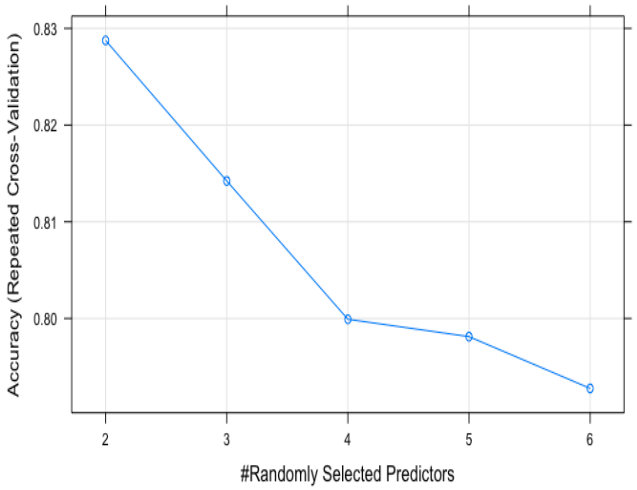
Full Model

Number of Predictors selected

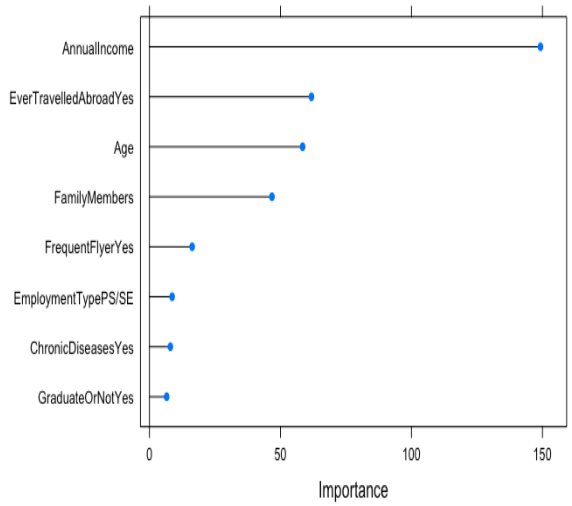


Reduced Model

Number of Predictors selected



Variable Importance Plot for Full Model



Variable Importance Plot for Reduced Model

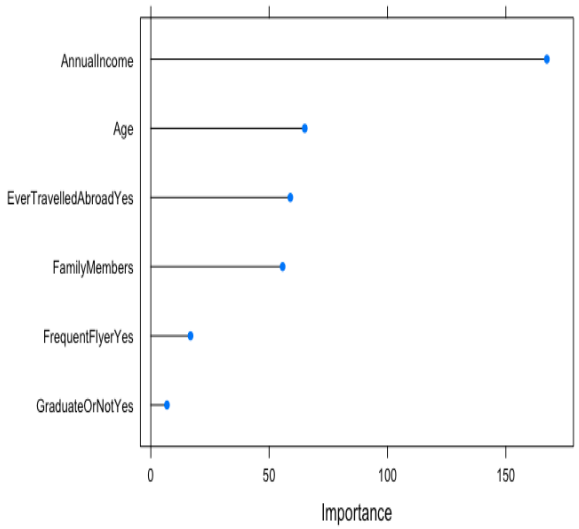
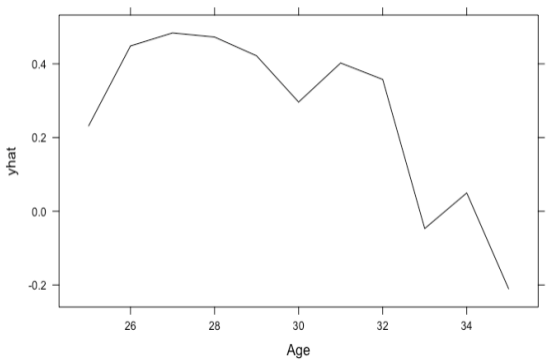
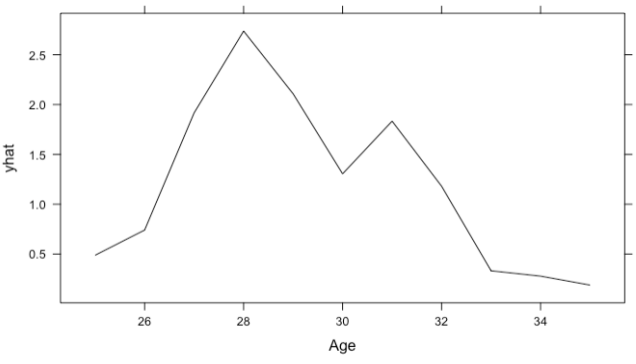


Figure G. *Random forest variable importance plots & predictor select.*

XGBoost



Random Forest



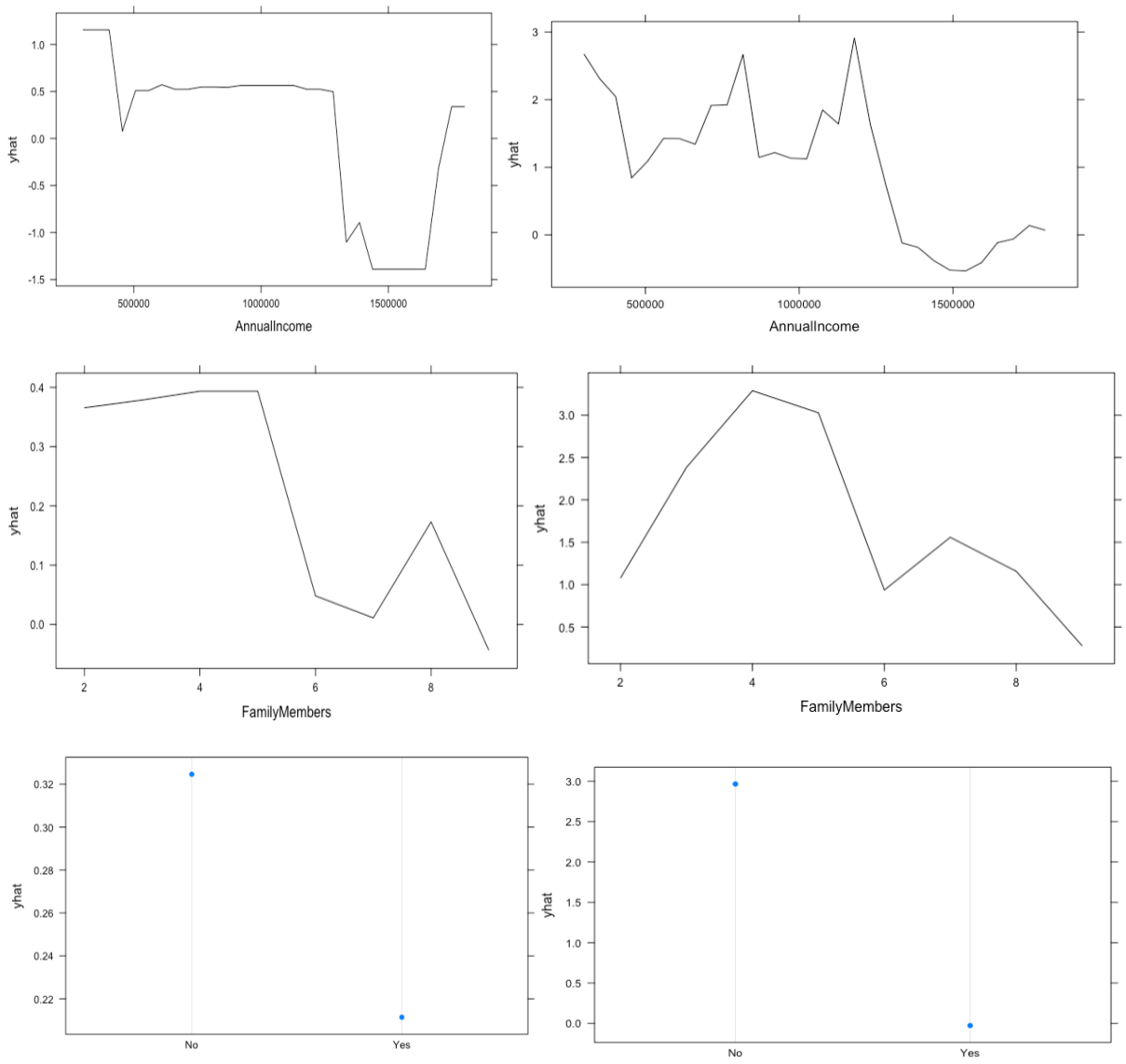


Figure H. *Partial dependence plots.*

Definitions

Acronym	Meaning	Acronym	Meaning
AIC	Akaike Information Criteria	NB	Naïve Bayes
ANN	Artificial Neural Network	NIR	No Information rate
AUC	Area under the Curve	NN	Neural Network
CART	Classification and Regression Trees	PDA	Penalized Discriminant Analysis
CNN	Convoluted Neural Network	PLS	Partial Least Squares
CV	Cross Validation	PS/SE	Private Sector / Self-Employed

DA	Discriminant Analysis	QDA	Quadratic Discriminant Analysis
FN	False Negative	RF	Random Forest
FP	False Positive	RNN	Recurrent Neural Network
FPR	False Positive Rate	ROC	Receiver Operating Characteristic
GLM	Generalized Linear Model	SDA	Shrinkage Discriminant Analysis
GS	Government Sector	SVM	Support Vector Machines
KNN	k-Nearest Neighbors	TN	True Negative
LASSO	Least Absolute Shrinkage and Selection Operator	TP	True Positive
LDA	Linear Discriminant Analysis	TPR	True Positive Rate
MLP	Multi-layer Perceptron	XGBoost	eXtreme Gradient Boosting
PD Plot	Partial Dependence Plot		