PREDICTION OF SECONDARY INSURANCE PURCHASE

Rohan Venkatesh Sirigeri, Narasimha Gaonkar, Suryatej Hittala Mathada, Aishwarya Halvagal Veeresh

Group 7

Abstract

Coming from a country with the population close to 1.5 billion and the insurance penetration of just 3.17%, we feel that by choosing the insurance sector and providing some actionable insights, we can not only make the major insurance providers more profitable, but also save countless lives and families even if we insure just the sole/primary bread earner of every family in the country. Here in this paper, we are building a model to predict the secondary insurance purchase. The approach involves using a supervised learning probabilistic classification to determine the decision of customers which will help insurance provider company to plan accordingly.

Introduction

An insurance business that has offered health insurance to its clients is looking for your assistance in developing a model to determine whether policyholders (clients) from the previous year will also be interested in the vehicle insurance offered by the firm.

An insurance policy is a contract whereby a business agrees to guarantee compensation in the event of a certain loss, damage, disease, or death in exchange for the payment of a predetermined premium. The amount of money that the client must consistently pay to an insurance provider in exchange for this assurance is known as a premium.

Like medical insurance, vehicle insurance requires the client to pay an annual premium to the insurance provider business for them to be compensated in the event that an unfortunate accident involving the vehicle occurs.

Building a model to forecast a customer's interest in vehicle insurance is incredibly beneficial for the business since it allows it to design its marketing strategy to reach out to those clients and maximize its business model and revenue.

Data Collection and Cleaning

We discovered a file with information from a study of an insurance firm that we may use for our project. The corporation used that poll to see whether it's customers would be willing to buy additional insurance from them. The file consisted of fundamental columns such **as** id**,** Gender**,** Age**,** Driving License**,** Region_Code**,** Previously_Insured **,**Vehicle_Age**,** Vehicle_Damage 1 **,**Annual_Premium**,** PolicySalesChannel **,**Vintage**,** Response**.**

Data cleaning is the process of ensuring that data is accurate, consistent, and usable. To prevent making the same mistakes again, you can clean data by searching for errors or corruptions, fixing or removing them, or manually processing data as necessary.

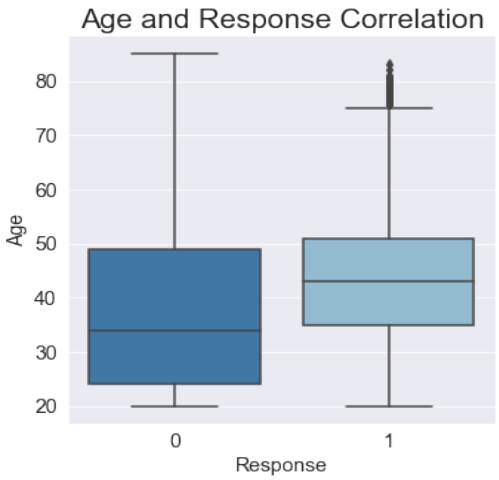
Most data cleansing operations can be assisted by software; however, some must be carried out by hand. Although it may seem like a difficult task, data cleansing is a crucial part of maintaining corporate data. Data cleansing is essential for businesses that use a lot of data. By eliminating unwanted data, more space is made available for the data that is still being collected. Keeping just relevant data simplifies data analysis as well. With appropriately cleaned data, it is simpler to produce significant business insights and actions.

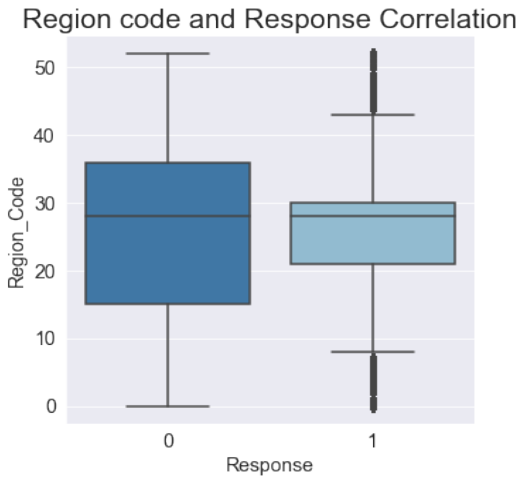
When many data sources are integrated into a single dataset, it is impossible to avoid all the significant differences and errors that might occur. By quickly extracting the information you want from the data you already have, using data cleaning technologies will increase team productivity. There will be fewer errors, which will make customers happier and staff less irritated. You may use it to map different data functions, which will help you understand what your data is intended to do and where it came from.

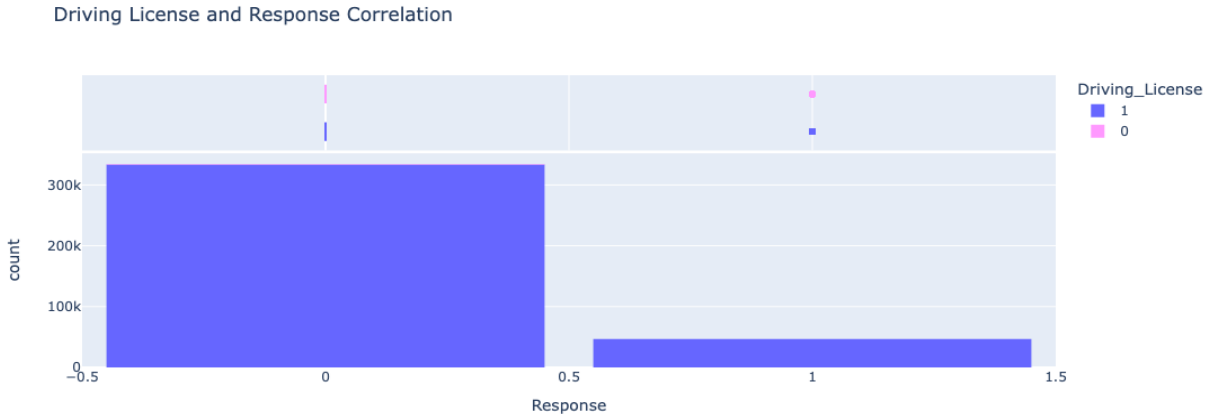
The data cleaning process for our data is done by checking the null and NA values and treat them. The next is changing the data types of the attributes to make prediction easier. For example, for the gender attribute we previously had string data type for which it had values as 'female' and 'male' which was converted into integer data type of values '0' and '1'. The same is done for vehicle_age, vehicle_damage attributes as well.

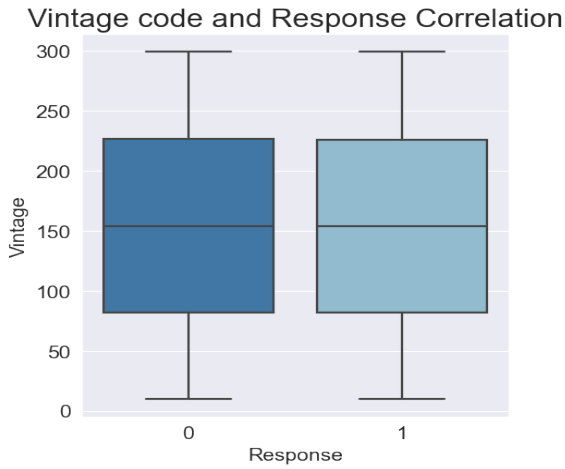
Analysis & Visualization

We visualized a few columns that we felt had a stronger correlation with the response column. These are listed below.

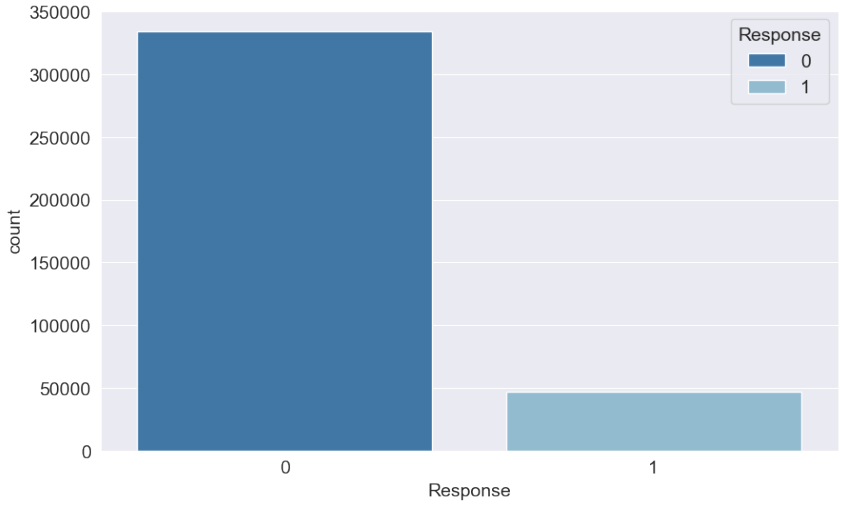








We may infer from the image above that reaction is more closely tied to vintage code. This suggests that owners of classic automobiles are more likely to purchase vehicle insurance to safeguard their asset.



We discovered that we had less favorable answers after checking the data's responses. If we make a prediction based on this data, it will lean more toward the negative than the positive responses. To solve this issue, a plan must be developed.

Feature Selection

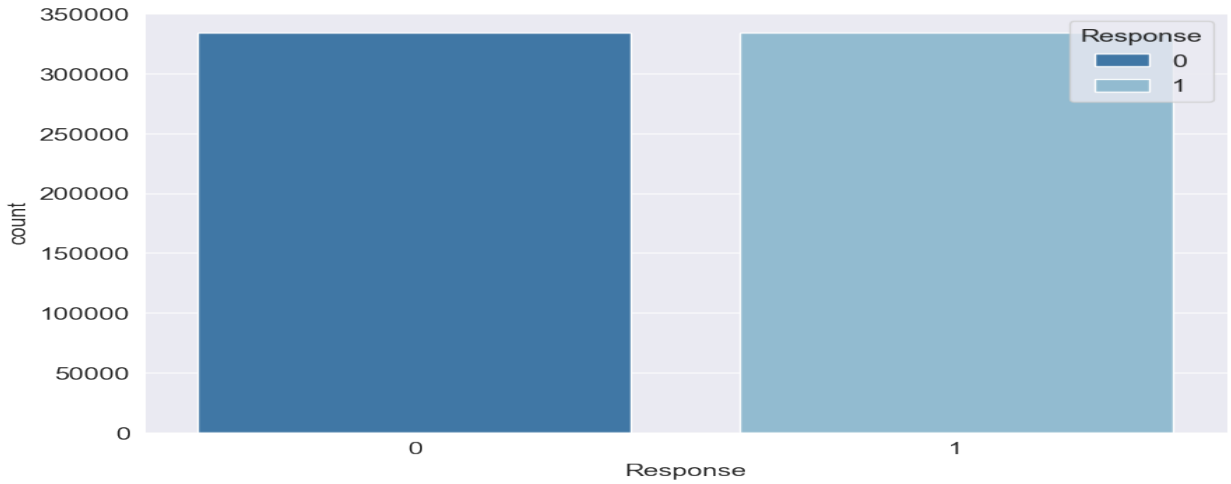
Before selecting the features, we must make sure that the data is not skewed. So, to achieve that we use the technique called Synthetic Minority Oversampling Technique (SMOTE).

Working with unbalanced datasets presents several difficulties, including the fact that most machine learning algorithms will underperform on the minority class even though this class's performance is often of utmost importance.

The minority class can be oversampled as one method of rectifying unbalanced datasets. Duplicating instances from the minority class is the easiest method, but these examples don't provide the model any new insights. As an alternative, fresh instances might be created by

synthesizing the old ones. The Synthetic Minority Oversampling Technique, often known as SMOTE, is a method of data augmentation for the minority class.

Following the use of the SMOTE approach, our response graph appears as follows:



The correlation between all the variables is calculated in the correlation matrix, and the values are shown as a matrix. We determine the correlation between variables and express it as a matrix in the correlation matrix.

Correlation analysis is used to determine how dependent different variables are on one another. Since the data we have are the data points of a bivariate function, the covariance of the variables is determined by considering them as random variables. If these variables rely linearly on one another, there is a strong connection between them. Bivariate data always have a correlation between 1 and -1; if the correlation is close to 1 or -1, the two variables are closely connected. This is possible because we treat all the data's columns as Xn and Y random variables. Given that the values for X range from 0 to 1 and the values for Y range from -1 to 1, the correlation between the independent and dependent variables now varies from -1 to 1. So, the correlation values tend to be larger if the variables meet the requirement of probabilistic independence.

When two variables are dependent on one another, the expected value condition and covariances are used to determine the correlation between the two variables. When the

condition is applied, if the predicted values are positive, the correlation between the variables rises. As a result, a correlation matrix is a n*n matrix with Xn variables and correlation values where the diagonal of the matrix is 1. Since our data has a bell-shaped distribution and is normally distributed, the correlation matrix has values between -1 and 1. This is since conditional expected value, marginal mean, and variance for variables are linearly dependent when random variables have a normal distribution.

The correlation matrix's formula is provided by

Model Selection and Fitting:

While trying to find the probabilistic classification model that would work accurately and match our dataset the best. As a result, we thought of using the following probabilistic models:

1. Logistic Regression:

   Contrary to its name, logistic regression is a classification model rather than a regression model. For binary and linear classification issues, logistic regression provides a quicker and more effective solution. This classification approach performs very well with linearly separable classes and is relatively simple to implement. It is a widely used algorithm for categorization in business.

2. Decision Tree Classifier:

   An internal node represents a feature (or property), a branch represents a decision rule, and each leaf node indicates the conclusion in a decision tree, which resembles a flowchart. The root node in a decision tree is the first node from the top. It gains the ability to divide data according to attribute values. Recursive partitioning is the process of repeatedly dividing a tree. This framework, which resembles a flowchart, aids in decision-making. It is a flowchart-like representation that perfectly replicates how people think. Decision trees are simple to grasp and interpret.

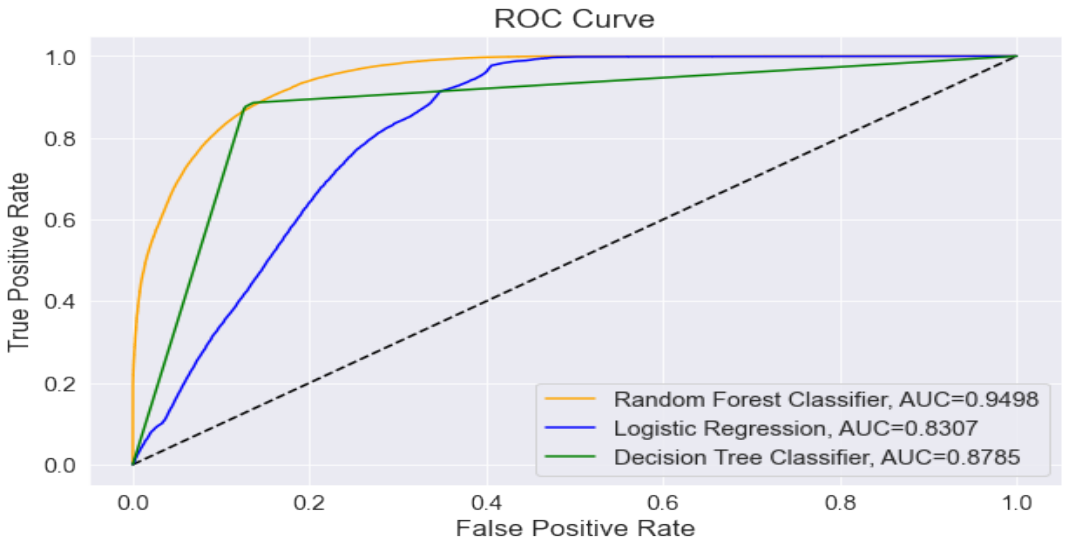
   Decision tree for our data is as follows:

3. Random Forest Classifier:

The machine learning technique known as random forest is adaptable and simple to use, and it typically yields excellent results even without hyper-parameter adjustment. It creates a "forest" out of a collection of decision trees that were typically trained using the "bagging" technique. The main principle behind the bagging approach is that combining learning models improves the end outcome. The ability of random forest to be applied to both classification and regression problems, which make up most contemporary machine learning systems, is a significant benefit. A decision tree or a bagging classifier have virtually identical hyperparameters to random forest. The hyperparameters of a random forest are quite like those of a decision tree or a bagging classifier. Fortunately, using the classifier-class of random forest eliminates the requirement to combine a decision tree with a bagging classifier. Using the algorithm's regressor, you may use random forest to handle regression problems as well.

We divided our data into train and test data ratios of 8:2 each. After that, we used the train dataset to run the three probabilistic classifier models and made predictions using the test dataset. We wrote Python code to fit our model using the Scikit-Learn module.

For each of the three models, we displayed the ROC (Receiver Operating Characteristic) curve and calculated the AUC (Area Under the Curve) for each model. The following is the ROC plot:



The models' F-1 Score, and Accuracy are as follows:

Model	F1 Score	Accuracy
Logistic Regression	81.99%	79.0%
Decision Tree Classifier	87.46%	87.0%
Random Forest Classification	87.62%	87.0%

By considering the AUC values, F-1 scores, and Accuracy, we can conclude that the Random Forest Classifier model outperformed other models.

Conclusion: In this project, we tried to predict whether the customers of a particular insurance company we interested in purchasing a secondary insurance offered by the same company. The Best-Fit model which we obtained is the Random Forest Classifier Model with highest AUC in the ROC curve of 0.9498, F-1 score of 87.62% and Test Accuracy of 87.0%. So, using the actionable insights obtained by using the Random Forest model, several major insurance providers across India can collectively achieve the insurance penetration of more than the current 3.17% by targeted marketing and in-turn save countless lives and families.

References:

1. https://www.kaggle.com/

2. https://scikit-learn.org/stable/

3.machine-learning/crash-course