HEALTH INSURANCE CROSS-SELL PREDICTION

Hariom Bhardwaj, Mayank Kumar,
Shivam Mishra, Saifuddin Raja
Sarvesh Kumar Yadav
Data science trainees,
AlmaBetter, Bangalore

Abstract:

Cross-selling is a great way to make more money for any insurance agency without starting from scratch.

We can build from a business from the book you already have with your current customer relationships. It is not only benign for the company but withal for the clients. It is not only the matter of making more profit. But It is withal about integrating value and bringing solutions to the indemnification-cognate challenges your customers face.

Just like medical insurance, there is vehicle insurance where every year the customer needs to pay a premium of certain amount to insurance provider company so that in case of unfortunate accident by the vehicle, the insurance provider company will provide a compensation (called 'sum assured') to the customer.

Our dataset is based on Health insurance Customers database.

This Experiment can help to understand what can be affecting factors for cross-selling of Insurance Plans for the already existing customers.

The model can be used for any insurance plan dataset to predict cross-selling.

Keywords: machine learning, Supervised machine learning, Cross-selling, Predictive Model building, KNN, XG Boost, Random Forest

1. Problem Statement

Our client is an insurance company that has provided Health insurance to its customers now they require our avail in building a model to prognosticate whether the policyholders from the past year will additionally be intrigued with the Vehicle insurance provided by the company.

An insurance policy is an arrangement by which a company undertakes to provide an assurance of emolument for a designated loss, damage, illness, or death in reciprocation for the payment of a designated premium. A premium is a sum of profit that the customer needs to pay customarily to an insurance company for this assurance.

The dataset consists of these important features: -

1. ID : Unique ID for the customer

2. Gender : Gender of the customer

3. Age : Age of the customer

4. Driving_Licence : whether Customer has DL

5. Region_Code : Unique code for the region of the customer

6. Previously_Insured : Whether Customer already has Vehicle Insurance

7. Vehicle_Age : Age of the Vehicle

8. Vehicle_Damage : Whether Customer got his/her vehicle damaged in the past.

9. Annual_Premium : The amount customer pay yearly for Insurance.

2. Introduction

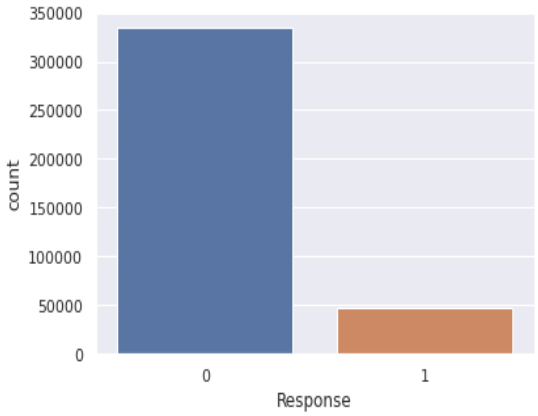
cross-selling is a new marketing strategy based on data analysis, showing that different needs also exist, people can become customers and fulfill their needs through selling services or products. various related products.

We are utilizing the KNN algorithm, XG Boost, Random Forest for building different prediction models, XGBoost is very good for unbalanced datasets because it takes care of it automatically. The Random Forest engenders decision trees on randomly selected data samples, gets predictions from each tree, and selects the best solution by means of voting. Here we have features like age, driving license, and vehicle_age which is going to give a relationship with the response variable.

3. Response type:

1. Interested
2. Not Interested

Segregation of customers according to their interest is first step towards building of prediction model, we have features like age, vehicle age, annual premium of the policy taken. These features have correlation with response variable.



4. Need for cross-selling Model Building:

The need for Cross-Selling Models is

1. To better understand the customers, need.
2. To drive profit maximization from already existing customers instead of launching new marketing campaigns.
3. To provide a personal feel to the customers.

5. Steps involved:

Exploratory Data Analysis

After loading the dataset, we performed this method by comparing our target variable that is Response with other independent variables. This process helped us to figure out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

We have done EDA on our DataSet to analyse, investigate and summarize their main characteristics, often employing data visualization methods.

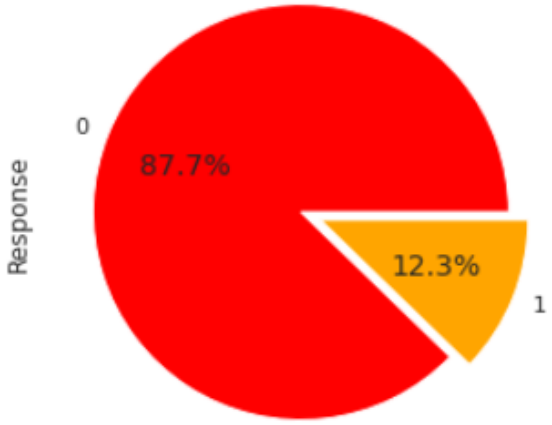


Fig: Distribution of data on Customer Response

Our dataset is unbalanced, as we can see in the above figure.

Feature Engineering:

Feature engineering is the process that takes raw data and transforms it into features that can be used to create a predictive model using machine learning or statistical modelling, such as deep learning.

We used the following techniques;

1) **Null values Treatment**

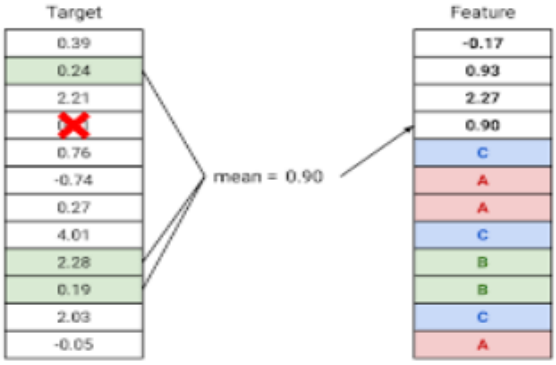
   Our dataset contains no null values.
   Hence, no treatment is needed for this feature data.

2) **Label Encoding:**

   In machine learning, we usually deal with datasets that contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human-readable form, the training data is often labelled in words.
   We replaced values of features vehicle age, Gender, Vehicle damage with numerical values.

3) **Target mean encoding:**

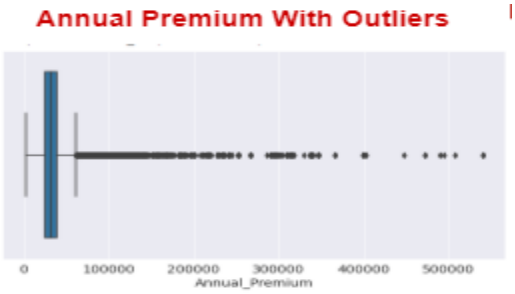
   Target encoding is the process of replacing a categorical value with the mean of the target variable. Any non-categorical columns are automatically dropped by the target encoder model.
   We applied Target Mean Encoding with Policy Sales Channel and Region Code.

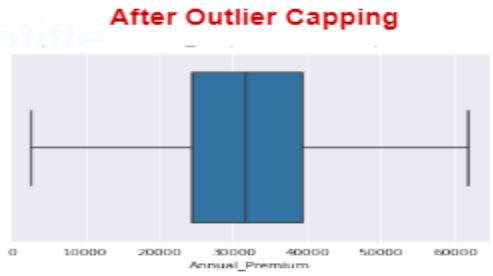


4) **Capping Outliers:**

   We can replace the extreme values by values closer to other values in the variable, by determining the maximum and minimum boundaries with the mean plus or minus the standard deviation, or the interquartile range proximity rule. This procedure is also called bottom and top coding, censoring, or capping.

   An only annual premium has outliers. We are applying capping for annual premium data.





5) **Standardization:**

Standardization refers to shifting the distribution of each attribute to have a mean of zero and a standard deviation of one (unit variance).

Here, we have used Robust scaling, which uses the median for scaling purposes in place of the mean.

For KNN model deployment, we used robust scaling.

- **Tuning the hyperparameters for better accuracy**

  Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in the case of tree-based models
  Like Random Forest Classifier and XGBoost classifier.

- **SHAP Values for features**

  We have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn't put much weight on the performance of our model.

7.1. Algorithms:

1. K- Nearest Neighbour(KNN):

K-Nearest neighbour model can be use for both regression and classification model. However, it is widely used for classification problem.
KNN represents a supervised classification algorithm that will give new data points accordingly to the k number or the closest data points, while k-means clustering is an unsupervised clustering algorithm that gathers and groups data into k number of clusters.

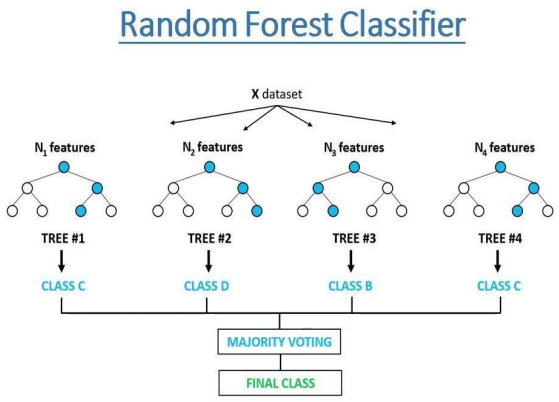
Fitting different models:

For modelling, we tried various classification algorithms like:

1. **KNN classifier**
2. **Logistic Regression**
3. **Random Forest Classifier**
4. **XGBoost classifier**

2. Random Forest Classifier:

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets, and then averages the final prediction depending on the most number of times a label has been predicted out of all.

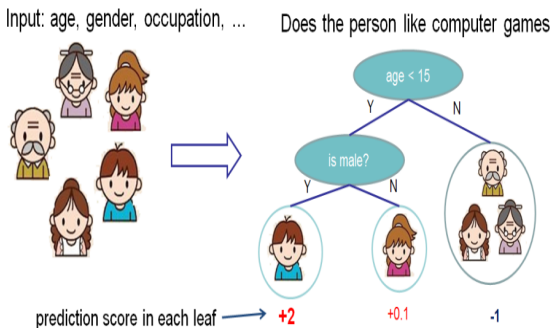


3. XGBoost-

To understand XGBoost we have to know gradient boosting beforehand.

Gradient Boosting-

Gradient boosted trees consider the special case where the simple model is a decision tree



In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).
When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A 'greedy' way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

XGBoost is one of the fastest implementations of gradient boosting. Trees it does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

7.2. Model performance:

The model can be evaluated by various metrics such as:

Confusion Matrix-

The confusion matrix is a table that summarizes how successful the classification models at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

1. **Precision/Recall-**
   Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP
   Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

2. **Accuracy-**
   Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

3. **Area under ROC Curve(AUC)-**
   ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

4. **F1 Score-**
   The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary

classification systems, which classify examples into ‘positive’ or ‘negative’.

7.3. Hyperparameter tuning:

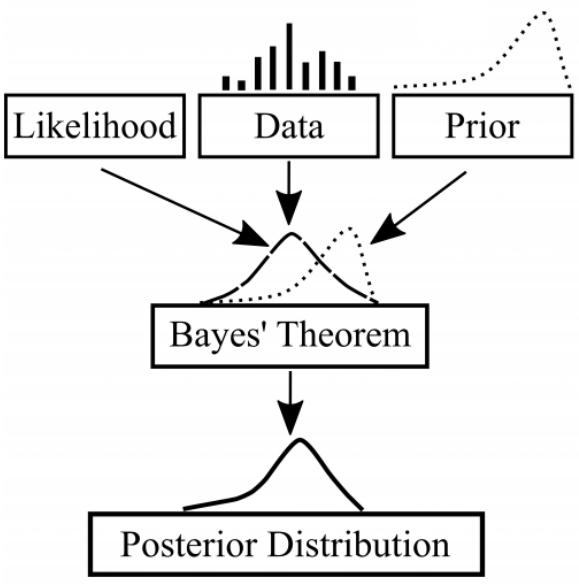
Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyperparameter tuning. This also results in cross validation, and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

2. **Randomized Search CV-** In Random Search, the hyperparameters are chosen at random within a range of values that it can Assume the advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist’s control.

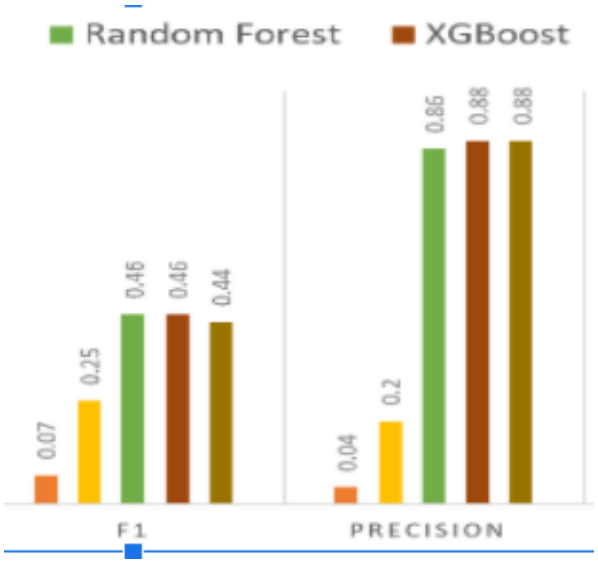
3. **Bayesian Optimization-** Bayesian Hyperparameter optimization is a very efficient and interesting way to find good hyperparameters. In this approach, in naive interpretation way is to use a support model to find the best hyperparameters. A hyperparameter optimization process based on a probabilistic model, often Gaussian Process, will be used to find data from data observed in the later distribution of the performance of the given models or set of tested hyperparameters.

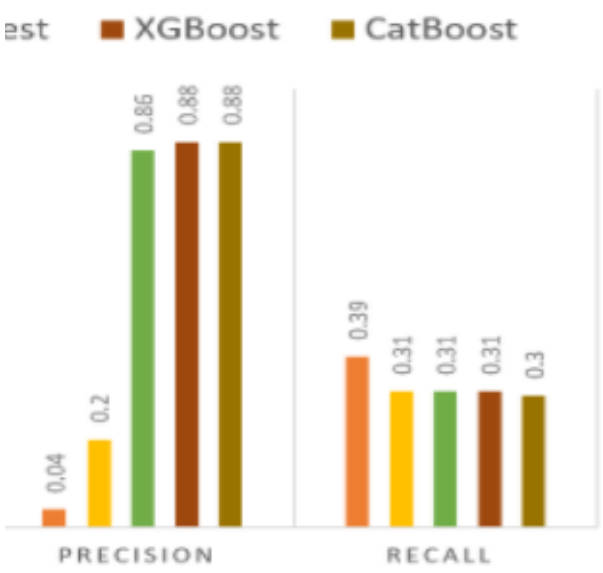


As it is a Bayesian process, at each iteration, the distribution of the model’s

performance in relation to the hyperparameters used is evaluated and a new probability distribution is generated. With this distribution, it is possible to make a more appropriate choice of the set of values that we will use so that our algorithm learns in the best possible way.

8. Result and Report:





9. Conclusion:

We Started with loading the data, also analyzed the data using EDA and different Visualization Plots. Later on, we did feature engineering(Fixed outliers and Encoding of different Categorical Features) to fix the data and remove the unwanted information.
And finally, we created the baseline model and high-performance model to compare our F1 score and AUC-ROC value.
In all of these models, our accuracy revolves in the range of 70 to 74%.
And there is no such improvement in accuracy score even after hyperparameter tuning.
So the accuracy of our best model is 73%, which can be said to be good for this large dataset. This performance could be due to various reasons like no proper pattern of data, very few features that are related to vehicle insurance. And as a conclusion basis, we can say that: Customers of age between 30 and 70 are more likely to buy insurance, Customers with Vehicle_Damage are more likely to buy insurance, Customers with Vehicle age between 1 and 2 years are more likely to be interested as compared to the other two categories.

and customers who have previously been insured are tended not to be interested. By improving on the above point we can improve our business

The variables: Age, Previously_insured, Annual_premium are more affecting the target variable. Comparing the ROC curve, we can see that the XGBoost model performs better.

-

References-

1. Machine Learning Mastery
2. GeeksforGeeks
3. Analytics Vidhya