**Health Insurance Cross Sell Prediction**

**Shubham chandrakar**

**Data science trainee,**

**AlmaBetter, Bangalore**

**Abstract:**

Our client is an Insurance company that has provided Health Insurance to its customers. Now they need the help of the data science team in building a model to predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company.

In this project, at first data wrangling is performed followed by exploratory data analysis (EDA) is performed to understand the market and behavior of the customers. After that several machine learning models were built to predict whether the past customer who was interested in health insurance policy will also be interested in vehicle insurance policy.

***Keywords:machine learning, vehicle insurance ,classified labels, supervised machine learning, imbalanced dataset.***

**1.Problem Statement**

Our client is an insurance company which is involved in selling health insurance. An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee.Now the insurance company needs the help of a data science team in building a model to predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company.

Our main objective is to explore, analyze and draw insights from the data as well as predict very efficiently where policyholders (customers) from past year will also be interested or not.

**2. Introduction**

### The Insurance company is involved in the health insurance business. They have data of customers related to health insurance. The data consist of following features:

1. id : Unique ID for the customer
2. Gender : Gender of the customer
3. Age : Age of the customer
4. Driving\_License 0 : Customer does not have DL, 1 : Customer already has DL
5. Region\_Code : Unique code for the region of the customer
6. Previously\_Insured : 1 : Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
7. Vehicle\_Age : Age of the Vehicle
8. Vehicle\_Damage :1 : Customer got his/her vehicle damaged in the past. 0 : Customer didn't get his/her vehicle damaged in the past.
9. Annual\_Premium : The amount customer needs to pay as premium in the year
10. PolicySalesChannel : Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
11. Vintage : Number of Days, Customer has been associated with the company
12. Response : 1 : Customer is interested, 0 : Customer is not interested

Response feature is our dependent variable which is binary in nature. It is a categorical variable, the value will be “ 1 ” if the customer was interested in a health insurance policy or the value will be “ 0 ” if the customer was not in health insurance.It is found out that data was heavily imbalanced.

Our goal here is to build a predictive model, which could help the insurance company in predicting whether the customer will be interested in purchasing a vehicle insurance policy.

## **3. Factor which health insurance affects cross-sell**

The following are factors which affect health insurance cross-sell with respect to vehicle insurance:

* Past vehicle damage.
* Age group of customers.
* Age of the vehicles.
* Region of operation .
* Policy sales channel.
* Previously insured or not (insurance status)

# **4. How some factors affect sale of vehicle insurance**

## **Most of the customer belong to young age group**

When we divided the age of customers into different age groups i.e. youth ( age less than 35 ), middle-aged (age between 35 and 50), senior ( age between 50 and 62 ) and super senior ( age greater than or equal to 62 ). It is found that most of the customers belong to the youth age group followed by middle age.

It is found out that middle aged people are most responsive followed by seniors. Least responsive age group is youth.

The client must focus more on youth as the positive responsiveness is low for the youth age group. They must also focus towards senior and super seniors.

## **Most of the vehicle age is between 1-2 year**

It is seen that most of the customers have vehicles with age between 1 - 2 years followed by customers with vehicles with age less than 1 year.

## **Sales of vehicle insurance policy depends upon policy sales channel**

It is found that some policy sales channels have very good performance .There are 36 policy sales channels which generated no response. There are 27 out 155 sales channels that generated more than 20% positive response. There are 78 out 155 sales channel generated more than 10% positive response

So it can be seen that the sales depend on the performance of the sales channel.

Clients must provide proper resources and training to the 36 policy sales channels with zero response. They must channel experience of 78 sales channels with 10% conversion rate to improve customers.

## **Sales of vehicle insurance policy depends upon region of operation**

All regions are generating more than 4% conversion . 15 out of 53 regions under operation generated 12% conversion in terms of positive response.

Clients must provide proper resources and training regions where conversion rate is less than 5%. And try to expand customer base where it has more than 12% conversion in terms of positive response

## **Sales of vehicle insurance policy depends upon previous damage of the the vehicle**

All people who had vehicle damage are mostly showing positive response in comparison to people with no vehicle damage who are showing negligible response.

## **Sales of vehicle insurance policy depends upon insured status**

People who were previously insured are mostly showing positive response in comparison to people were not previously insured are showing negligible response

**5. Steps involved:**

* **Data wrangling**

In these steps the dataset is loaded, null value is treated, duplicated rows are removed, dependent variable distribution is checked, feature engineering is performed, correlation analysis is performed, outlier removal is performed.

* **Exploratory Data Analysis**

In this step univariate analysis in which we observe each variable individually, bivariate analysis is performed to check the relation between two features.

* **Null values Treatment**

Our dataset contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.

* **Data preprocessing for ML model**

At first VIF analysis is performed to remove multicollinearity. After that categorical encoding is performed. At last final tuning of data is done to finally use data efficiently for building ML models.

* **Feature Selection**

In these steps we used algorithms like ExtraTree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

Next we used Chi2 for categorical features and ANOVA for numerical features to select the best feature which we will be using further in our model.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment. Here MinMaxScaler() is used for standardization.

* **Handling imbalanced dataset and Building ML models**

Dependent variable i.e. Response is heavily imbalanced. To solve this issue we used three different types of implementation. Following are the three type of the implementation:

1. Building ML models over imbalanced data and performing hyperparameter tuning using Cross-Validation over baseline model.
2. Building ML models after oversampling data using SMOTE implementation and performing hyperparameter tuning using Cross-Validation over baseline model
3. Building ML models after oversampling and undersampling data using SMOTETomek implementation and performing hyperparameter tuning using Cross-Validation over baseline model

* **Fitting different models**

For modeling we tried various classification algorithms with or without cross validation like:

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

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in 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.

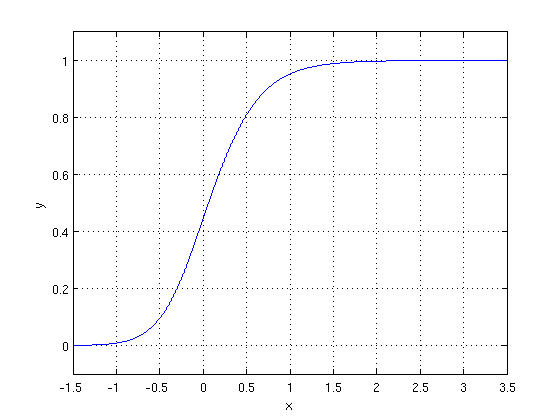
**6.1. Algorithms:**

1. **Logistic Regression:**

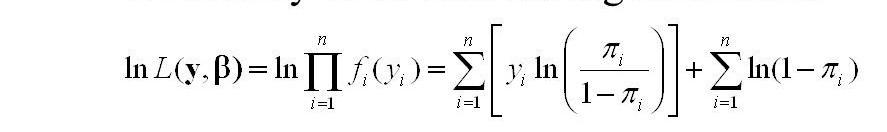
Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:

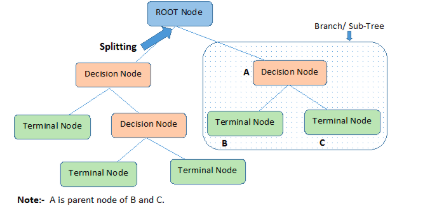


.

1. **Decision Tree Classifier:**

Decision tree is a type of supervised learning algorithm that is mostly used in classification problems. It works for both categorical and continuous input and output variables.

Let’s look at the basic terminology used to study decision trees:



**Root Node:** It represents the entire population or sample and this further gets divided into two sets.

**Splitting:** It is a process of dividing a node into two sub-nodes.

**Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.

**Leaf/ Terminal Node:** Nodes that do not split are called leaf or terminal nodes.

**Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.

**Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.

**Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria is different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable. Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

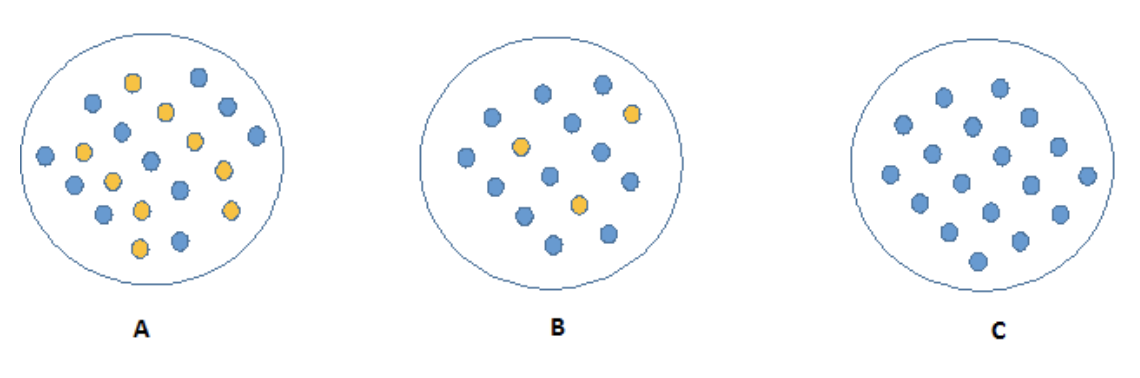
**Methods to determine best split:**

1. Information Gain
2. Gini
3. Chi Square
4. Reduction in Variance

Let’s understand about one of the features:

## Information Gain

Look at the image below and think which node can be described easily. I am sure, your answer is C because it requires less information as all values are similar. On the other hand, B requires more information to describe it and A requires the maximum information. In other words, we can say that C is a Pure node, B is less Impure and A is more impure.



Entropy can be calculated using formula:-



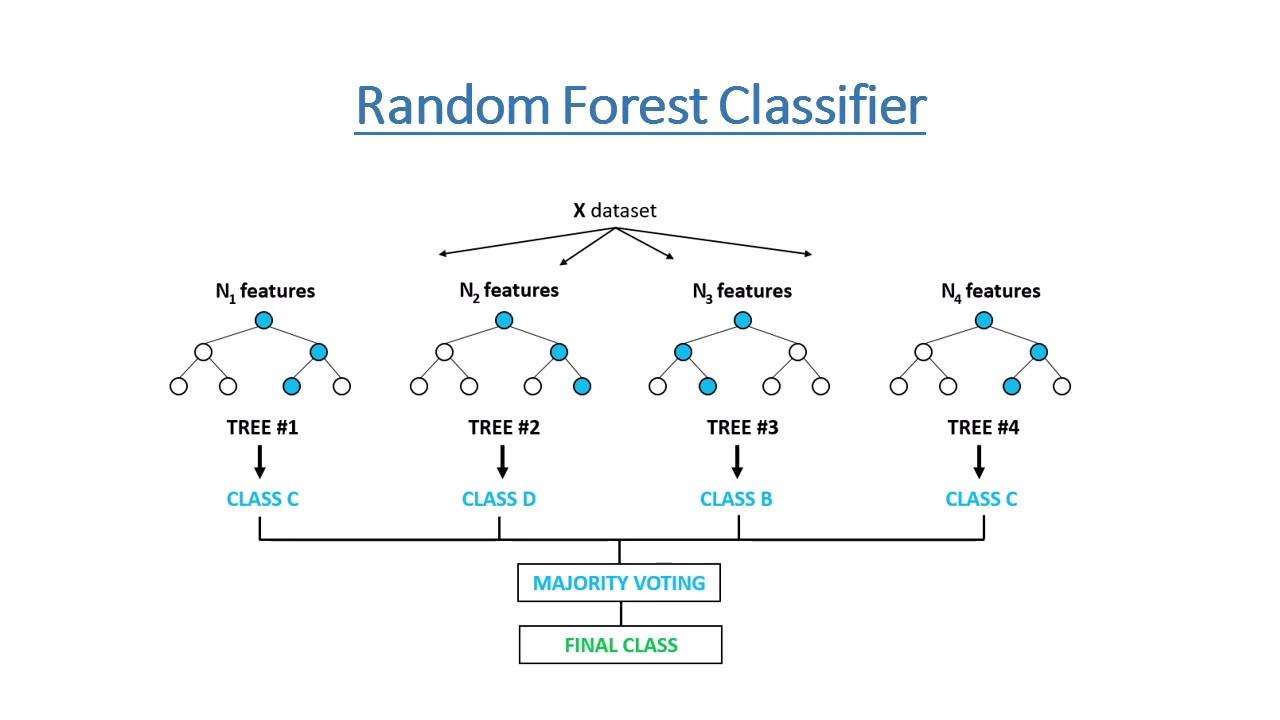
Here p and q are the probability of success and failure respectively in that node. Entropy is also used with categorical target variables. It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

The decrease in entropy after a split is called **Information Gain**. Steps to calculate information split for a split:

* Calculate entropy of parent node
* Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.
* Calculate the difference in entropy before and after split.

**3. 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.

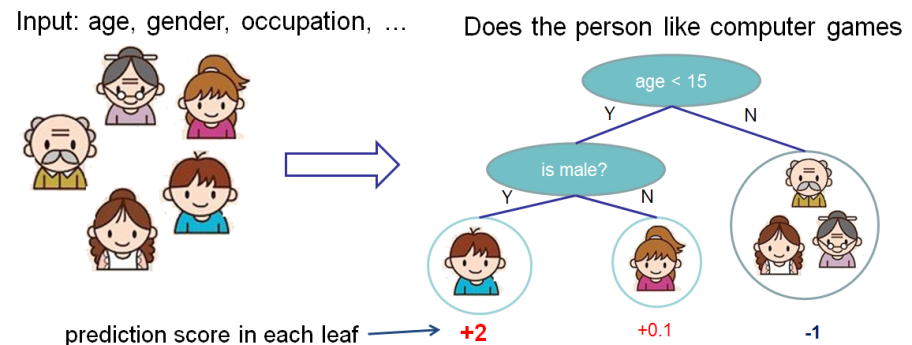


1. **XGBoost-**

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.

**6.2. Model performance:**

Model can be evaluated by various metrics such as:

1. **Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification modelis 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

1. **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

1. **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.

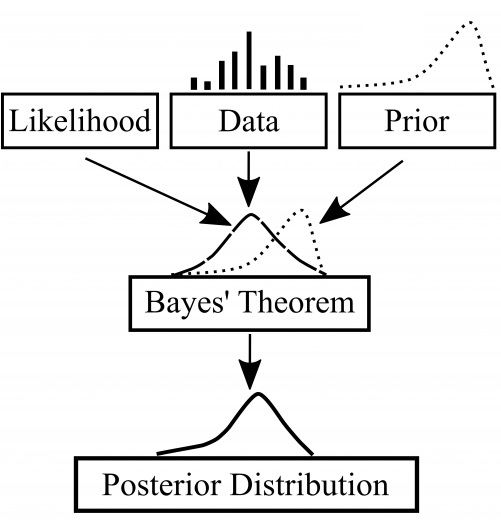
**6.3. Hyper parameter tuning:**

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

# **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.

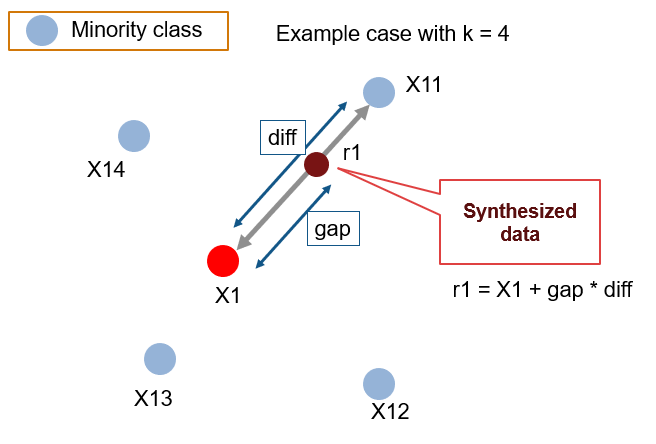
**6.4. Sampling using Implementation technique:**

**1. SMOTE:** Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling.

#### Working Procedure:

At first the total no. of oversampling observations, N is set up. Generally, it is selected such that the binary class distribution is 1:1. But that could be tuned down based on need. Then the iteration starts by first selecting a positive class instance at random. Next, the KNN’s (by default 5) for that instance is obtained. At last, N of these K instances is chosen to interpolate new synthetic instances. To do that, using any distance metric the difference in distance between the feature vector and its neighbors is calculated. Now, this difference is multiplied by any random value in (0,1] and is added to the previous feature vector. This is pictorially represented below:



Source: <https://github.com/minoue-xx/Oversampling-Imbalanced-Data>

**2. SMOTETomek : SMOTE + Tomek Links -**

Hybridization techniques involve combining both undersampling and oversampling techniques. This is done to optimize the performance of classifier models for the samples created as part of these techniques.

SMOTE+TOMEK is such a hybrid technique that aims to clean overlapping data points for each of the classes distributed in sample space. After the oversampling is done by SMOTE, the class clusters may be invading each other’s space. As a result, the classifier model will be overfitting. Now, Tomek links are the opposite class paired samples that are the closest neighbors to each other. Therefore the majority of class observations from these links are removed as it is believed to increase the class separation near the decision boundaries. Now, to get better class clusters, Tomek links are applied to oversampled minority class samples done by SMOTE. Thus instead of removing the observations only from the majority class, we generally remove both the class observations from the Tomek links.

**8. Conclusion:**

We need to target middle aged, senior people and focus more on converting youth into our possible customers. People who did not have vehicle damage needed to be guided more to purchase policy more as most of the customers showing positive response had vehicle damage. Thirty-six policy channels had generated no response,they needed to be trained robustly. Regions generating less than 8% conversion rate in terms of customer policy purchase interest must focus on increasing brand presence in their respective area. While building ML models, SMOTETomek must be used for resampling data after that Random Forest Classification model with hyperparameter tuning can predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company with 94.26 % ROC-AUC score and 88.27% f1 score.

**References-**

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya