HEALTH INSURANCE CROSS SELL PREDICTION

Lucky Jain, Debashish Das, Vivek Katolkar

Alma Better

Abstract: The dataset contains data on demographics (gender, age, region code type), vehicles (vehicle age, damage), policies (premium, sourcing channel), and so on. We predicted that the customer who has medical insurance from the company will or will not be interested in purchasing a vehicle insurance policy based on this feature. This model is extremely beneficial to the company because it allows it to plan its communication strategy to reach out to those customers and optimize its business model and revenue accordingly.

Problem Statement: 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. Just like medical insurance, there is vehicle insurance where every year 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. So we tried our best to build a model to predict whether a customer would be interested in Vehicle Insurance or not.

Data Summary:

- Id: Unique ID for the customer
- Gender: Gender of the customer
- Age: Age of the customer
- Driving License 0: Customer does not have DL, 1: Customer already has DL
- Region Code: Unique code for the region of the customer
- Previously_Insured: 1: Customer already has Vehicle Insurance, 0: Customer doesn't have Vehicle Insurance
- Vehicle_Age : Age of the Vehicle
- Vehicle_Damage :1: Customer got his/her vehicle damaged in the past. 0: Customer didn't get his/her vehicle damaged in the past.
- Annual Premium: The amount customer needs to pay as premium in the year
- Policy Sales Channel: Anonymized Code for the channel of outreaching to the customer i.e. Different Agents, Over Mail, Over Phone, In Person, etc.
- Vintage: Number of Days, Customer has been associated with the company
- Response: 1: Customer is interested, 0: Customer is not interested

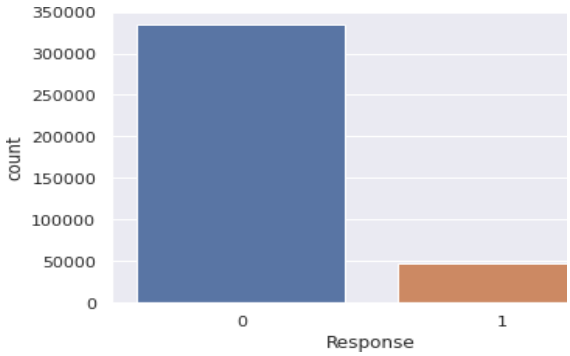
Dataset Shape: (381109, 12)

	Name	dtypes	Missing	Uniques	First Value	Second Value
0	id	int64	0	381109	1	2
1	Gender	object	0	2	Male	Male
2	Age	int64	0	66	44	76
3	Driving_License	int64	0	2	1	1
4	Region_Code	float64	0	53	28	3
5	Previously_Insured	int64	0	2	0	0
6	Vehicle_Age	object	0	3	> 2 Years	1-2 Year
7	Vehicle_Damage	object	0	2	Yes	No
8	Annual_Premium	float64	0	48838	40454	33536
9	Policy_Sales_Channel	float64	0	155	26	26
10	Vintage	int64	0	290	217	183
11	Response	int64	0	2	1	0

Exploratory Analysis and Visualization:

Exploratory data visualizations (EDVs) are the types of visualizations we create when we have no idea what information is contained in our dataset. Let us now begin exploring the first dataset's insights and understanding the relationship between columns or features. Let us now begin exploring insights into understanding the relationship between columns or features.

1. Dependent Variable: Response is the dependent variable or the target variable in the dataset. We plotted count plot on it.

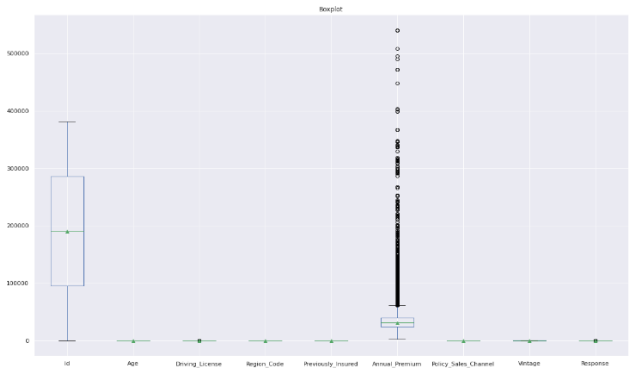


The data is highly imbalanced. As we can see in above graph, there are very few interested customers whose stats are less than 50000 and those above 300000 are not interested

2. Checking outlier in all the numerical columns: Plotted boxplot on the numerical features and the results are:

   As you can see

   1. Annual Premium has the highest outliers present in this dataset
   2. Driving License has very less outliers.
   3. Response has very less outliers.



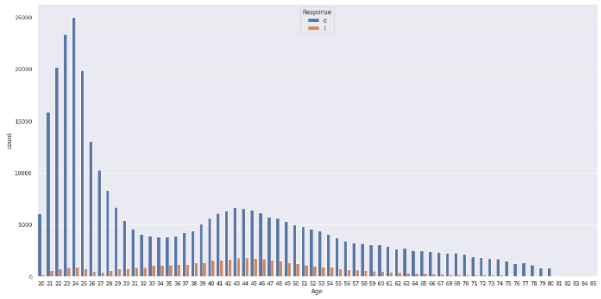
Response in male and female categories: Count plot on male and female categories versus response. We can conclude from the plots that: -

- The gender variable ratio in the dataset is nearly equal, male category is slightly more than female, and male category has slightly higher chances of purchasing insurance than female.
- The male population exceeds 200000, while the female population is close to 175000.The number of males interested is greater than 25000, while the number of females interested is less than 25000.The male category is slightly larger than the female category, and the chances of purchasing insurance are also slightly higher.

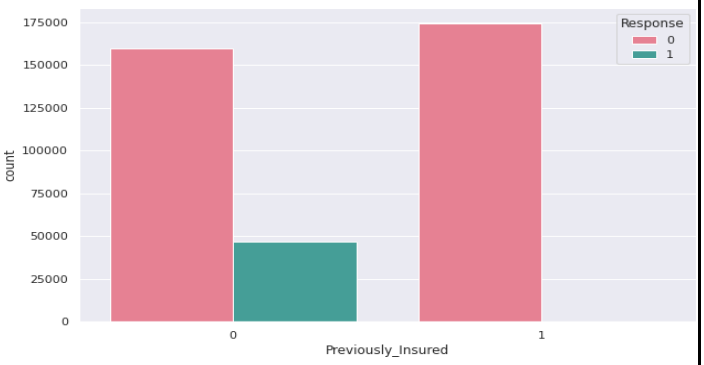
Response in different ages: Plotted count plot on different ages against response.

Results:

- Young people under the age of 30 are uninterested in auto insurance. Reasons could include a lack of experience, a lack of maturity, and a lack of expensive vehicles
- People aged between 30-60 are more likely to be interested.
- From the boxplot we can see that there no outlier in the data as you can see there is no outliers present in Age



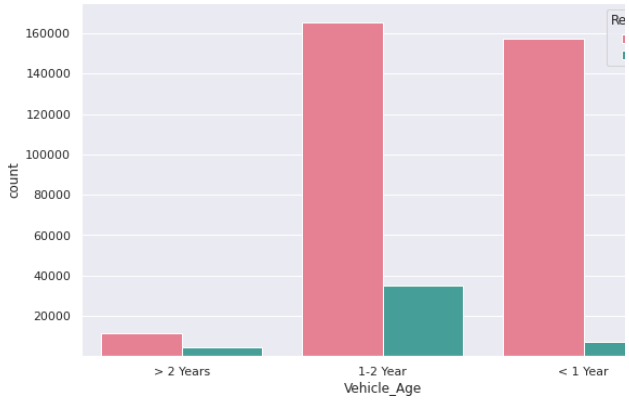
3. We plotted a count plot on driving licenses against responses and discovered that almost all customers interested in Vehicle Insurance have a driving license.
4. Plotted count plot on previously insured against response and we can see that those who have not insurance some of them are taking insurance



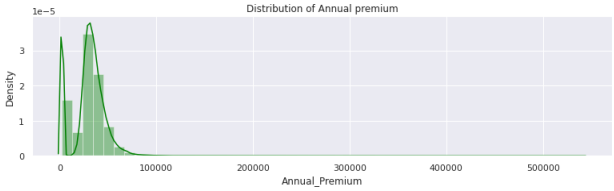
5. **Response in vehicle age:** Plotted count plot on vehicle age category against response. From the plots we can conclude that:

   Results:

   - From seeing this graph, we can say that if the vehicle's age is in between 1 to 2 years, those vehicle owners are more likely to buy insurance
   - The number of customers with Vehicle Age >2 is greater than the number of customers with Vehicle Age 1.



6. Plotted distribution and box plot of annual premium and we can conclude from plots
    - Based on the distribution plot, we can conclude that the annual premium variable is skewed to the right.
    - As you can see that in the column Annual premium there are many outliers present



1. Correlation matrix: Vintage has little effect on the target variable (response). We can eliminate the least correlated variable.

.

Data Preprocessing: The process of converting raw data into an understandable format is known as data preprocessing. Before using machine learning or data mining algorithms, the data quality should be checked.

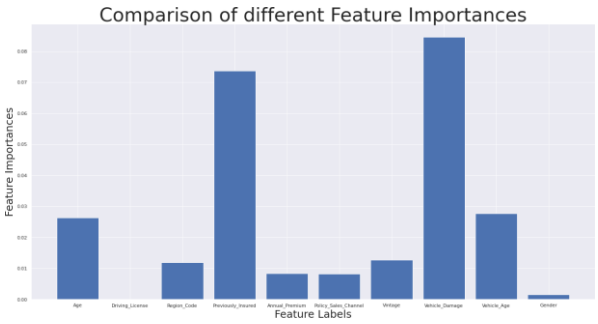
We tried to remove the duplicated rows of the dataset but there are no duplicate rows

1. **Label encoding:** Label encoding is the process of converting labels into a numeric format so that they can be read by machines. Machine learning algorithms can then make better decisions about how those labels should be used. In supervised learning, it is an important preprocessing step for the structured dataset. The categorical columns -vehicle damage, vehicle age, and gender - were converted to numeric using level encoding.

Dataset Shape: (381109, 11)

	Name	dtypes	Missing	Uniques	First Value	Second Value
0	Age	int64	0	66	44.0	76.0
1	Driving_License	int64	0	2	1.0	1.0
2	Region_Code	float64	0	53	28.0	3.0
3	Previously_Insured	int64	0	2	0.0	0.0
4	Annual_Premium	float64	0	48838	40454.0	33536.0
5	Policy_Sales_Channel	float64	0	155	26.0	26.0
6	Vintage	int64	0	290	217.0	183.0
7	Response	int64	0	2	1.0	0.0
8	Vehicle_Damage	int64	0	2	1.0	0.0
9	Vehicle_Age	int64	0	3	2.0	0.0
10	Gender	int64	0	2	1.0	1.0

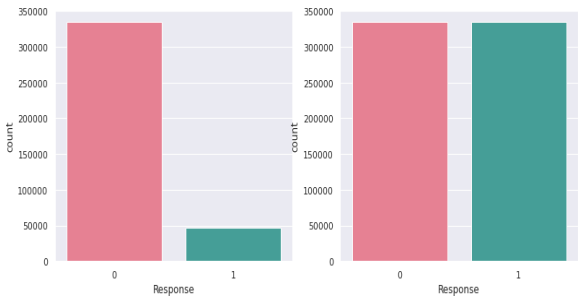
2. **Feature Selection**: The Extremely Randomized Trees Classifier (Extra Trees Classifier) is a type of ensemble learning technique that aggregates the classification results of multiple de-correlated decision trees collected in a "forest" to produce its classification result. In concept, it is very similar to a Random Forest Classifier and differs only in the way the decision trees in the forest are constructed.
The Extra Trees Forest's Decision Trees are built from the original training sample. Then, at each test node, each tree is given a random sample of k features from the feature-set, from which each decision tree must choose the best feature to split the data using some mathematical criteria (typically the Gini Index). This random selection of features results in the construction of multiple de-correlated decision trees. To perform feature selection using the above forest structure, during the forest's construction, the normalized total reduction in the mathematical criteria used in the feature of split decision (Gini Index if the Gini Index is used in the forest's construction) is computed for each feature To perform feature selection, each feature is ordered in descending order based on its Gini Importance, and the user selects the top k features based on his/her preferences.



We can remove less important features from the data set. Driving License, Gender is contributing very less that's why I'm removing those columns.

3. **Handling Imbalance dataset:** Handling Imbalanced Datasets: A class imbalance exists when observations in one class outnumber observations in other classes.

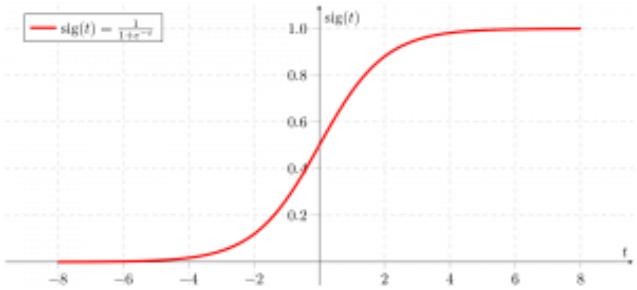
We can see that there is a significant difference between the data sets. We use the resampling technique to solve this problem. Random resampling is a crude technique for rebalancing an imbalanced dataset's class distribution. Overfitting can occur when random oversampling duplicates examples from the minority class in the training dataset. The method is known as "naive resampling" because it is based solely on data and does not employ any heuristics. This makes it easy to implement and quick to execute, which is ideal for very large and complex datasets. This technique is applicable to two-class (binary) classification problems as well as multi-class classification problems involving one or more majority or minority classes. Importantly, the class distribution change is only applied to the training dataset. The goal is to influence the models' fit. The test or holdout datasets used to evaluate a model's performance are not resampled. These naive methods can be effective in general, but it depends on the specifics of the dataset and models involved. To resample the dataset, we used a random over sampler. As we can see, our response has the same number of both classes. Dataset is being divided in an 80:20 ratio.



Machine Learning Models:

1. **Logistic regression**: It is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features (or inputs), X. Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as "1". Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

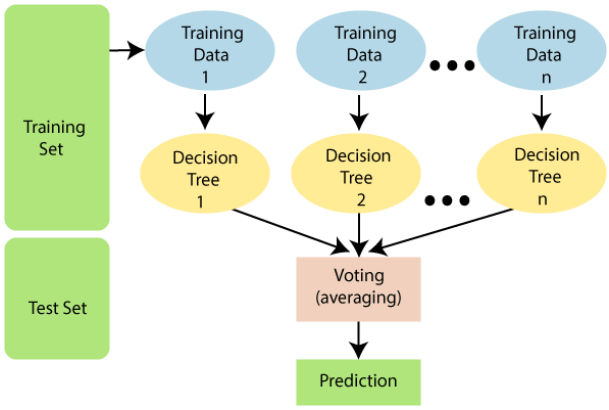
$$g(z) = \frac{1}{1+e^{-z}}$$



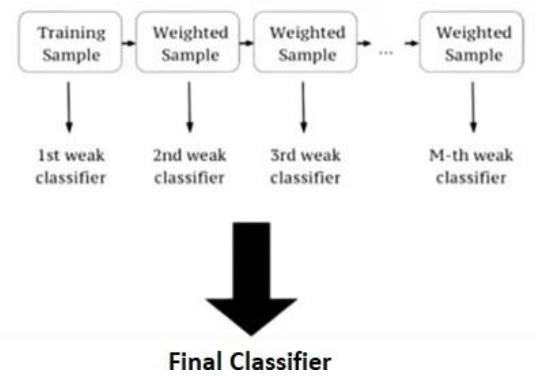
Only when a decision threshold is introduced into the equation does logistic regression become a classification technique. The threshold value is an important aspect of Logistic regression and is determined by the classification problem itself. The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this seldom is the case. In the case of a Precision-Recall tradeoff, we use the following arguments to decide upon the threshold.

2. **Random Forest Algorithm**: Random Forest is a well-known machine learning algorithm from the supervised learning technique. It can be applied to both classification and regression problems in machine learning. It is based on the concept of ensemble learning, which is a process that involves combining multiple classifiers to solve a complex problem and improve the model's performance. "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset," as the name implies. Instead of relying on a single decision tree, the random forest takes the predictions from each tree and predicts the final output based on the majority vote of predictions. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The

below diagram explains the working of the Random Forest algorithm:



3. **XGBoost:** Gradient Boosted decision trees are implemented in XGBoost. C++ was used to create this library. It is a type of software library that was created to improve model speed and performance. This algorithm generates decision trees in a sequential fashion. Weights are very important in XGBoost.All of the independent variables are given weights, which are then fed into the decision tree, which predicts results. The weight of variables incorrectly predicted by the tree is increased, and these variables are then fed into the second decision tree. These individual classifiers/predictors are then combined to form a more powerful and precise model. It can solve problems involving regression, classification, ranking, anduser-definedprediction.

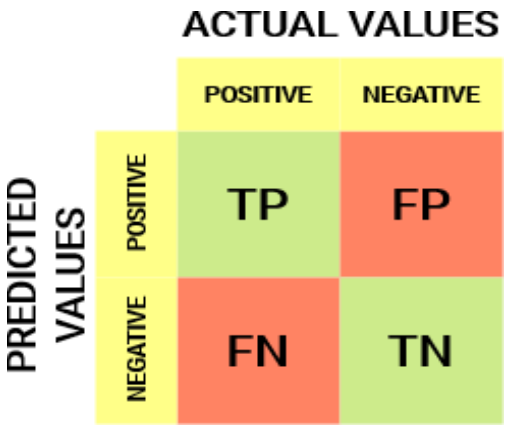


4. **<u>Hyper parameter tuning:</u>** Hyper parameters are sets of information that are used to control how an algorithm learns. Their definitions influence model parameters, which change as a result of the new hyper parameters. This set of values influences a model's performance, stability, and interpretation. Each algorithm necessitates its own hyper parameter grid, which can be tailored to the business problem at hand. The way a model learns to trigger this training algorithm after parameters to generate outputs is altered by hyper parameters. We used Grid Search CV, Randomized Search CV and Bayesian Optimization for hyper parameter 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.

5. **GridSearchCV:** It is the process of performing hyper parameter tuning in order to determine the optimal values for a given model. As mentioned above, the performance of a model significantly depends on the value of hyper parameters. GridSearchCV is a model selection function included in the Scikit-learns (or SK-learn) package.
So, an important point to note here is that the Scikit-learn library must be installed on the computer.
This function assists you in looping through predefined hyper parameters and fitting your estimator (model) to your training set. Finally, we can choose the best parameters from the list of hyper parameters. GridSearchCV evaluates the model for each combination of the values passed in the dictionary using the Cross-Validation method.
As a result of using this function, we can calculate the accuracy/loss for each combination of hyper parameters and select the one with the best performance.

Different Cost Functions: The cost function is a method for assessing "the performance of our algorithm/model."It takes both the model's predicted and actual outputs and calculates how far off the model was in its prediction.It returns a higher value if our predictions differ significantly from the actual values.

1. **Confusion Matrix:** A Confusion matrix is a N x N matrix used to assess the performance of a classification model, where N represents the number of target classes. The matrix compares the actual target values to the machine learning model's predictions. This provides us with a comprehensive picture of how well our classification model is performing and the types of errors it is making.
For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:



Let's break down the matrix:

- There are two possible values for the target variable: positive or negative.
- The columns represent the target variable's actual values; the rows represent the target variable's predicted values.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

2. **F1 score:** F1 is a precision and recall function. It is required when attempting to strike a balance between Precision and Recall .If we need to strike a balance between Precision and Recall AND there is an uneven class distribution, it might be a better metric to use (large number of ActualNegatives).

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

3. **Accuracy**: It can be largely contributed by a large number of True Negatives which in most business circumstances, we do not focus on much whereas False Negative and False Positive usually has business costs (tangible & intangible)
4. **ROC Curve:** An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

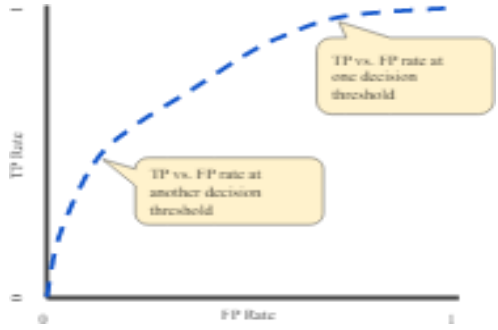


Figure: TP vs. FP rate at different classification thresholds.

We could evaluate a logistic regression model many times With different classification thresholds to compute the points in a ROC curve, but this would be inefficient. Fortunately, there is an efficient, sorting-based algorithm called AUC thatcanprovideuswiththisinformation.

Model

Evaluation:

Logistic Regression:

Accuracy: 0.784

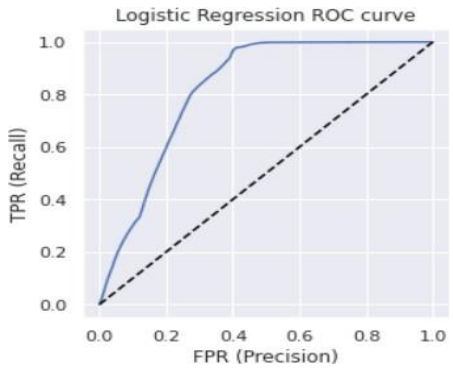
Precision: 0.705

Recall: 0.978

F1-Score: 0.819
ROC_AUC Score: 0.834

```
[[39510 27337]
 [ 1500 65413]]
```

Figure 1: Confusion Matrix (LOGR)



Random Forest :-

Classifier Accuracy:

0.948

```
[[60037   119]
 [ 6810 66794]]
```

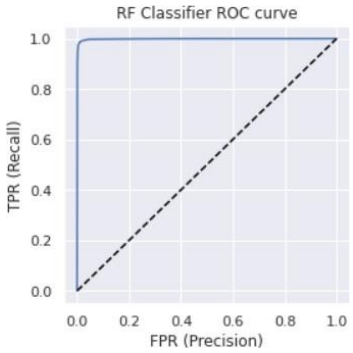
Precision: 0.907

Recall: 0.998

Figure 2: Confusion Matrix (RFC)

F1-Score: 0.951

ROC_AUC Score: 0.834



XGB Classifier:-

Accuracy : 0.797

```
[[44469 22378]
 [ 4786 62127]]
```

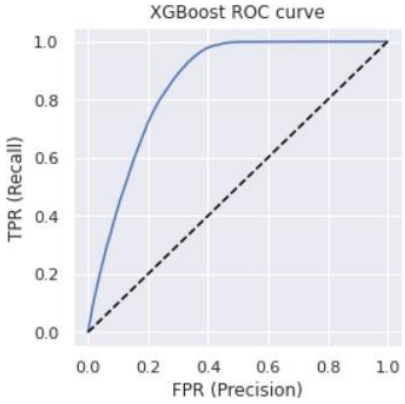
Precision: 0.735

Figure 3: Confusion Matrix (XGBC)

Recall: 0.928

F1-Score: 0.821

ROC_AUC Score: 0.819



Best Model: Because Random Forest Classifier has the highest accuracy, I will use GridSearchCV to set Hyperparametersvalues.

Best Parameters: {'criterion': 'gini', 'max_depth': 50, 'min_samples_split': 2, 'n_estimators': 10}

Accuracy : 0.951

```
[[60037   119]
 [ 6810 66794]]
```

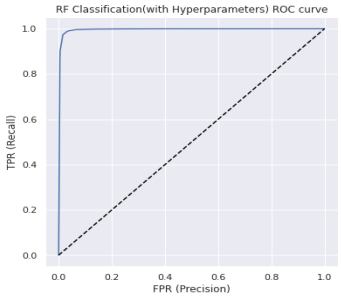
Precision: 0.912

Recall: 0.997

Figure 4: Confusion Matrix

F1-Score: 0.953

ROC_AUC Score: 0.834



As we can see, using Hyper parameters increased Accuracy, Precision, f1 score, ROC AUC, and decreased Recall. However, the change is very small; if you prefer, we can ignore it as well.

Conclusion:

- Models are built after loading the dataset, cleaning it, performing EDA, Feature Engineering, and feature selection.
- XGBoost Classifier produced the best results in terms of Training and Testing Accuracy and ROC-AUC score.
- The most important features for the model were Vehicle damage and Previously Insured.
- In this case, we will select XGBoost Machine Learning.
- This model can predict the label of the target customer who responds and does not respond at a recall model with a value of 93 percent and an AUC score of 96 percent.
- This means that our model can improve our response rate when predicting customers who are interested in purchasing vehicle insurance.

References: -

- Data science for business: what you think about data mining
- Hands-On Exploratory Data Analysis with Python Perform EDA techniques to understand, summarize, and investigate your data by Suresh Kumar Mukhiya, Usman Ahmed (z-lib.org)
- https://bunker2.zlibcdn.com/dtoken/01c5fc197a94283bfb0c0943bd5b2d0c