# Machine Learning Meets Term Life Insurance: Targeting High-Value Customers

# 1. Introduction

## 1.1 Background of HashSysTech Insurance

Well known to be associated with creativity in the market, HashSysTech Insurance is an innovative and optimistic Insurance Company. The company offers several insurance products, and one of the most popular product lines is term life insurance. The development of HashSysTech as a young company is striving in a competitive environment, which implies investing in consumer advertising programs to promote the products. Initially, one very effective way of reaching out to potential new customers is through telemarketing. However, since such initiatives are getting more costly, the business wants to find out how to reduce costs while maintaining the effectiveness of such programs. This necessitated the development of Project Greenlight, which aims to apply data mining principles to improve the chances of accurately identifying term life insurance consumers.

## 1.2 Problem Statement: Optimizing Telemarketing Campaigns

Several challenges were encountered while identifying the telemarketing concept for HashSysTech Insurance, one major issue highlighted is the difficulty in adopting telemarking, primarily because of the high costs. The company must consider reducing its targeting and marketing to low-value customers since they often offer less conversion. This means that, based on clients' information and conditions and the spectral's interactive design, the predictive model can accurately show customers with a higher probability of purchasing term life insurance. This will let HashSysTech effectively use telemarketing on the definite segment of the customers, which will bring profits and eliminate unprofitable costs. The use of this approach will assist the company in achieving adequate control over its resources.

## 1.4 Overview of the Dataset

This report employs a dataset of HashSysTech Insurance, containing client data and interaction records extricated from preceding telemarketing campaigns. The features are divided into demographical features, contact information features specific to the campaign, and the target variable, equal to one for every customer who has opted for term life insurance. The data set will be transformed to a ready state for building machine learning models by undergoing pre-processing steps: data cleaning, missing values handling, and data visualizations.

# 2. Task 1: Data Exploration and Preparation

## 2.1 Data Exploration

### ****2.1.1 Descriptive Statistics for Numerical Features****

The database holds quantitative values that construct the customer profile and aspects of telemarketing sale campaigns. Statistical summaries comprising average, median, and variation are computed to describe the distribution of the variables about the probability of converting customers for term life insurance.

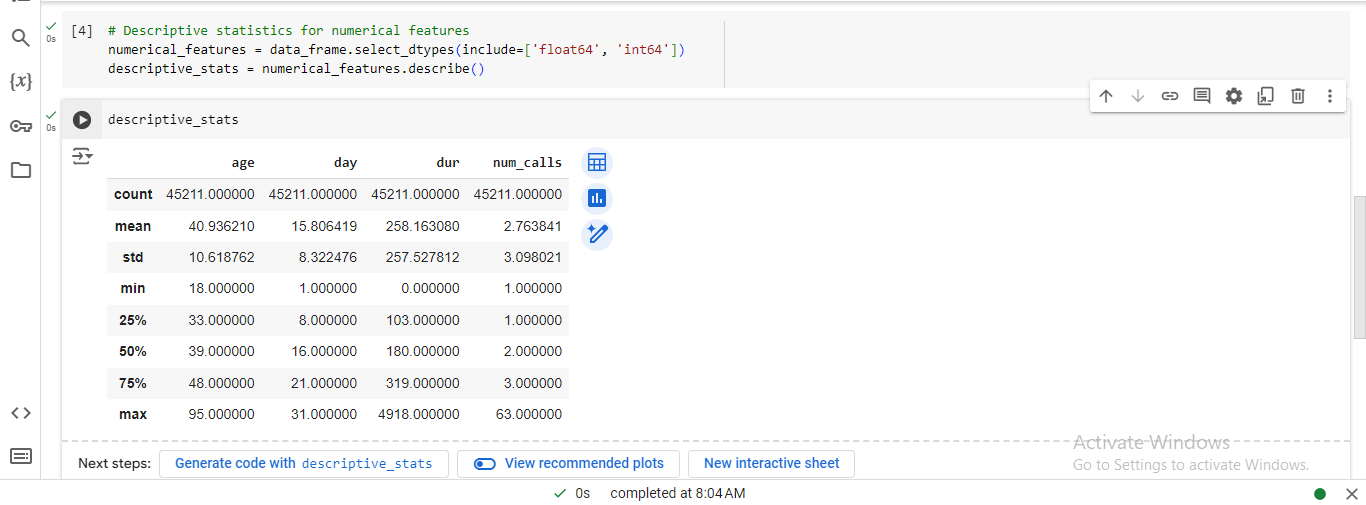


Figure : Descriptive Statistics for Numerical Features

The statistical measures used include mean, median, mode, range, and standard deviation for the numerical features in the dataset, which allows an improved appreciation of each feature as a determinant of customer behavior.

### 2.1.2 Frequency Analysis for Categorical Features

The categorical variables provide some of the most valuable information about the customers, including their occupation, marital status, and method of contact. A frequency analysis was used to evaluate how often these categorical variables appear in the dataset.

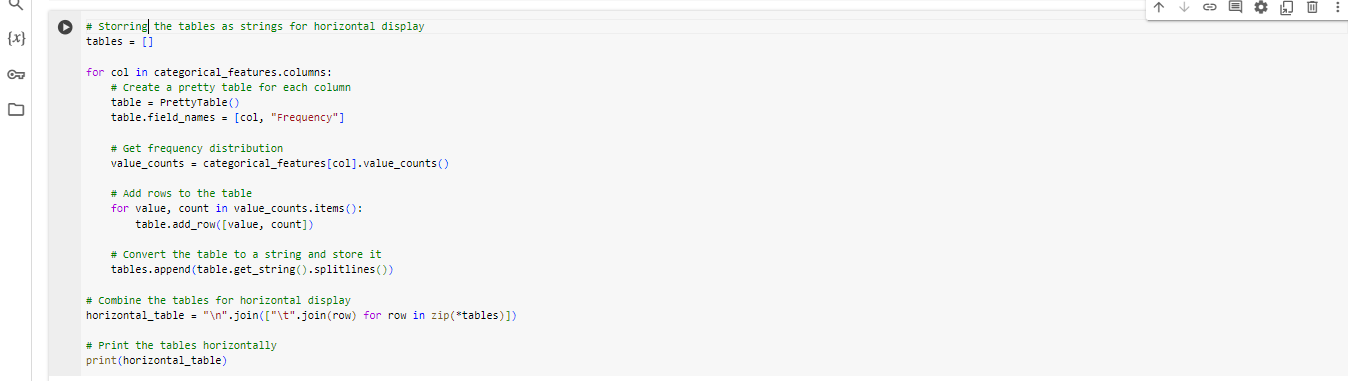


Figure : Jupyter Notebook Code: Frequency Distribution Tables

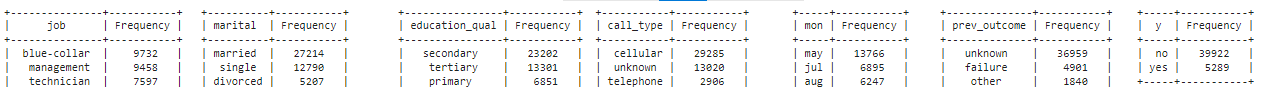


Figure : Frequency Distributions of Categorical Features

The frequency analysis lessens the degree of dispersion and gives an insight into the distribution of categorical features and how various customer segments engage the company’s telemarketing sales. This data will be helpful so that several high-value customers will be easily recognizable.

## 2.2 Handling Missing Values and Outliers

### 2.2.1 Identifying Missing Values

Since missing data can impact the machine learning model performance and lead to biased results, the solution to the problem is considered further. Hence, missing values should first be examined and dealt with before analysis is conducted on them.

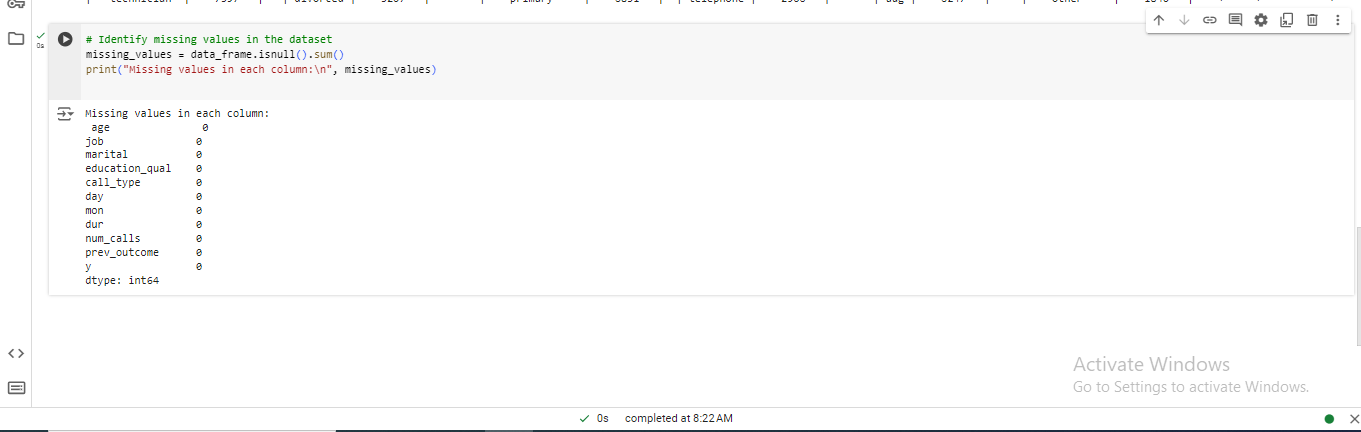


Figure : Jupyter Notebook Output: Missing Values Check

The output above summarizes missing values in the dataset, as displayed above. These missed values must be covered appropriately to maintain the credibility of the dataset in the analysis. The operationalization of the data showed that no missing values were found in any of the columns; therefore, no action is needed for the missing data.

### 2.2.2 Methods for Handling Missing Data

There are various approaches to dealing with missing data, such as deleting records that contain missing values or computing those using statistical measures. To avoid data missing issues in this analysis, imputation techniques, for instance, filling with the mean for numerical features and filling with mode for categorical features, will be used.



Figure : Jupyter Notebook Code: Missing Value Imputation

This imputation strategy enables the dataset to stay as rich as possible while unmasking the analysis with biases from neglectable datasets.

### 2.2.3 Detecting and Treating Outliers

Outliers can also significantly influence machine learning models; hence, they should be handled appropriately. Outliers were identified and handled in numerical features using the Interquartile Range (IQR) method.



Figure : Jupyter Notebook Coding and Output: Outlier Detection and Handling

Outliers distort the estimated expected customer behavior; therefore, their early identification and removal enhance the accuracy of any prediction made by the model.

#### 2.2.3.1 Addressing Outliers

The outlier counts you provided indicate the number of extreme values detected in some columns:

* Age ----- 487 outliers
* Day ----- 0 outliers
* Dur ----- 3235 outliers
* num\_calls ------ 3064 outliers

Here are steps to handle these outliers:

1. **Identifying Outliers**: Some techniques for identifying outliers include the Interquartile Range (IQR). If you note any outliers within the column, you must choose the proper treatment technique, which may either cap or delete the said value.
2. **Treating Outliers**: Depending upon the type of outliers, it can be either the deletion or the application of the capping process. Here’s an example of how to handle outliers using the IQR method: Here’s an example of how to handle outliers using the IQR method:

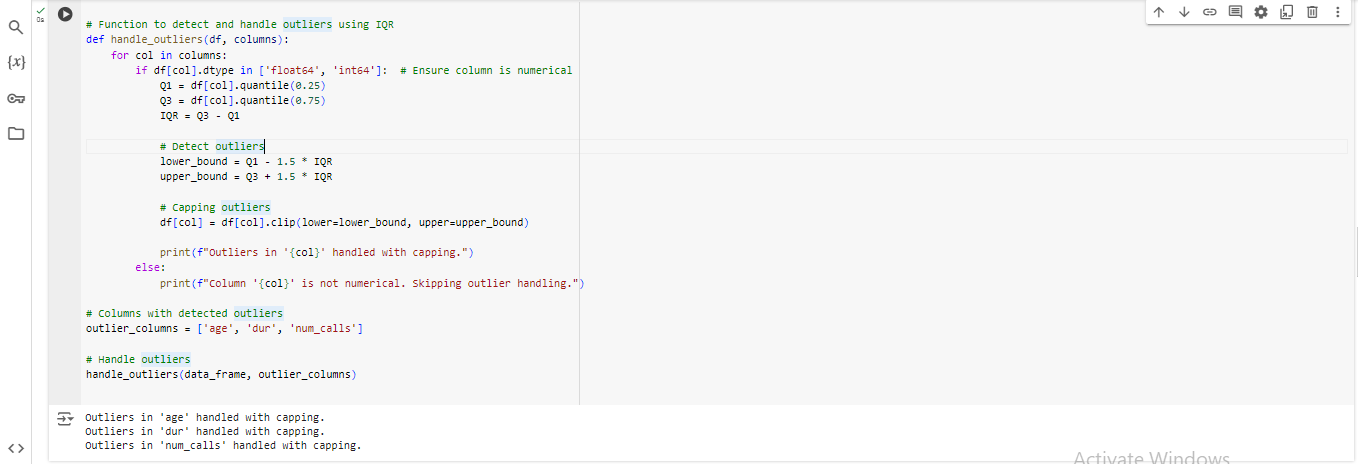


Figure : Jupyter Notebook Code: Outlier Detection and Handling

#### 2.2.3.2 Explanation

* **Identifying Outliers**: The method used to detect outliers is known as the IQR method. Values below Q1 - 1.5 \* IQR and above Q1 + 1.5 \* IQR are considered outliers.
* **Handling Outliers**: The **clip**() method limits values within the calculated range, reducing the impact of the outlying values on the model.

Thus, applying these methods to the dataset will make it more robust than when the structure of the data is violated, and the performance of an ML model is likely to be worse. If outliers are essential in the analysis or if they are valid cases, you may decide to analyze them rather than remove them.

## 2.3 Data Visualization

### 2.3.1 Visualizing Relationships between Features

Because of data visualization analysis, it is possible to determine the relative connection between the features and the target variable, customer conversion. Scatter plots, histograms, and box plots visualized these relationships.

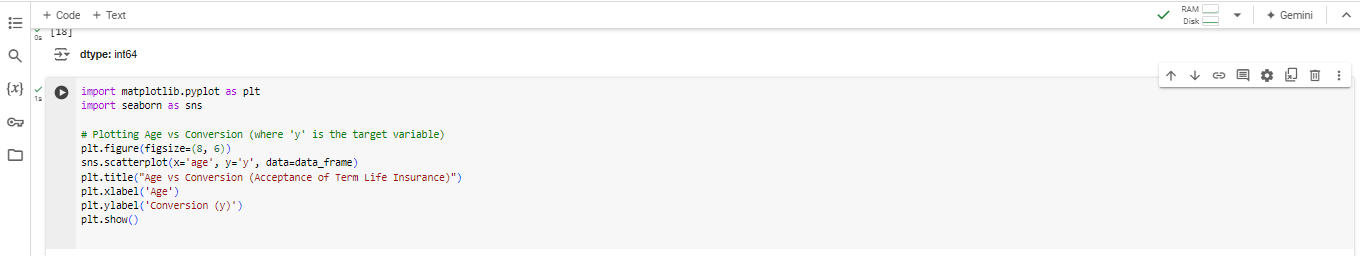


Figure : Jupyter Notebook Code: Plotting Age vs. Y (Target variable)

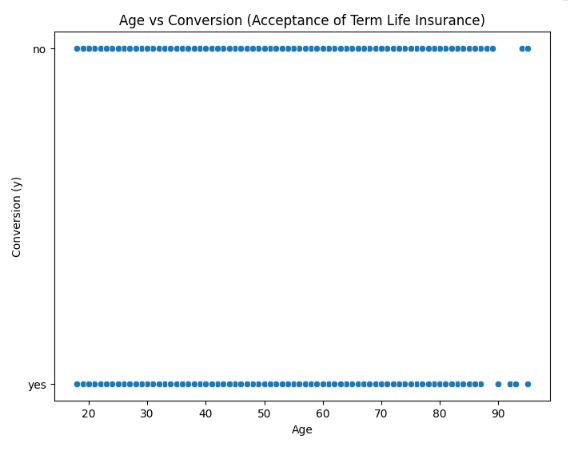


Figure : Scatter Plot: Age vs. Conversion (Acceptance of Term Life Insurance)

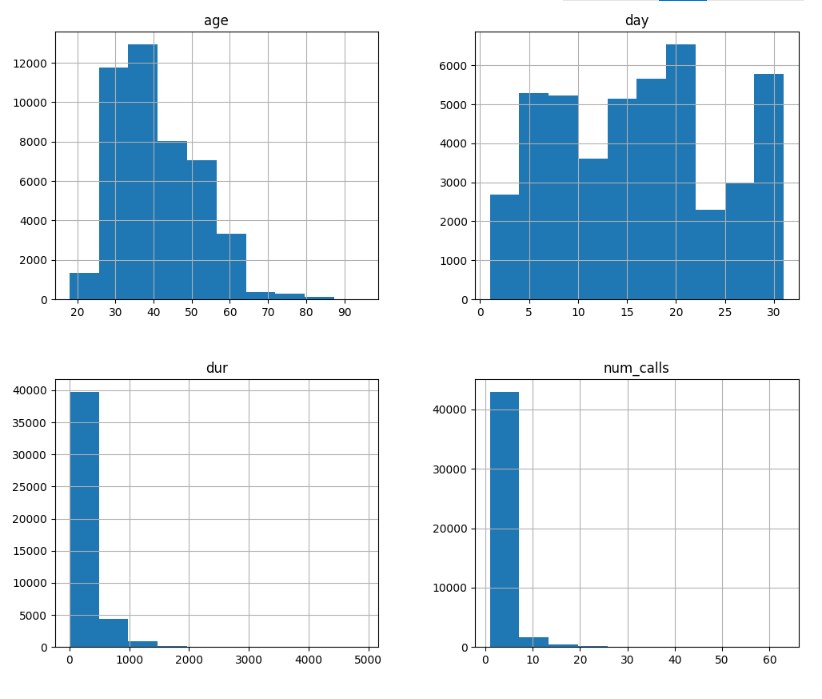


Figure : Histograms of Numerical Features

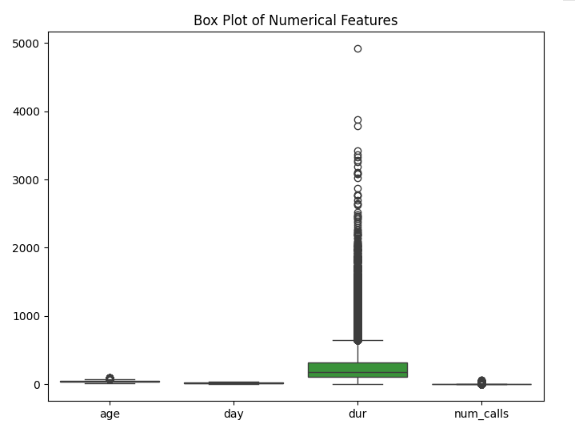


Figure : Box Plot of Numerical Features

These intuitive charting models provide another perspective on how various attributes, such as age and occupation, may affect customer conversion rates.

### 2.3.2 Insights from Data Visualizations

As a result of the visualization, it was noted that specific ages generate higher conversion rates, thus can be the angles targeted by HashSysTech Insurance while designing its telemarketing strategies. Similarly, some outliers were observed, which supports how moderate extreme values in the data set need to be handled. Graphs, especially time series graphs, also helped identify other patterns that may not be seen in descriptive statistics.

The next section of this report expands on how data was prepared and explored; this includes steps such as data cleaning, imputation of missing values, outlier detection, and data visualization, all of which are critical in formulating a robust predicting model.

# 3. Task 2: Model Selection and Training

## 3.1 Model Selection

### 3.1.1 Description of Selected Algorithms

Two widely used machine learning algorithms, Logistic Regression and Random Forest, have been selected to predict term life insurance conversions.

* Logistic Regression: Logistic Regression is a statistically valuable model for binary classification problems where the output can only be of two types (Yes or No, Buy or No buy, etc). It estimates the target variable posterior probability with a weighted sum of input variables. The generalized algorithm supposes a linear association between the independent variables and the logarithm of odds of outcome, thus yielding the algorithm more straightforward in interpretation and computational complexity.
* **Random Forest:** Random Forest is another learning technique developed from a decision tree learning algorithm. It constructs many decision trees using different subsets of the given data and features, and the output combines the suggested decisions. Random Forest can discover interactions with the features without imposing linearity and works well in classification and regression. Unlike other models, k-NN is less sensitive to overfitting, particularly when handling large datasets and noise.

### 3.1.2 Justification for Algorithm Choice

The decision is to use logistic regression because it is straightforward to interpret and easy to determine how the features affect the likelihood of conversion. Random Forest was selected since it can model non-linear relationships and is relatively immune to the problem of overfitting. This combination of models will allow us to identify linear behavior and higher-order higher-order patterns in the data set.

## 3.2 Data Splitting

### 3.2.1 Process of Splitting the Dataset

In addition, to compare the accuracy of models developed here and their ability to generalize and recognize patterns within new data, the dataset has been subsequently divided into training and testing datasets. This study implements a specific type of sampling, known as the stratified sampling technique, to match the raw distribution of positive and negative classes within the target variable of the train and test datasets. The dataset is conventionally divided into roughly 80/20, with the former meant for training and the latter for testing. This should ensure that models have adequate data to learn and some data to use to test them.



Figure : Jupyter Notebook Coding: Data Preprocessing and Model Training

In this process, the endogenous variable y will indicate whether the customer converted or accepted the insurance offer. The feature matrix X contains all the features that will, in turn, be used to predict the outcome.

### 3.2.2 Stratified Sampling for Target Variable Balance

Regarding the binary classification problem, dividing the target variable classes (converted/non-converted, for example) in the same ratio in both the training and testing datasets is essential. Hence, one is guaranteed that the model can perform on a distribution other than the training distribution of the target variable. Stratified sampling ensures that the ratios of positive and negative classes within the target variable are also followed in the training set and testing set. This reduces the problem of data imbalance during the evaluation and guarantees that the models are trained based on the variation in the target variable in both sets.

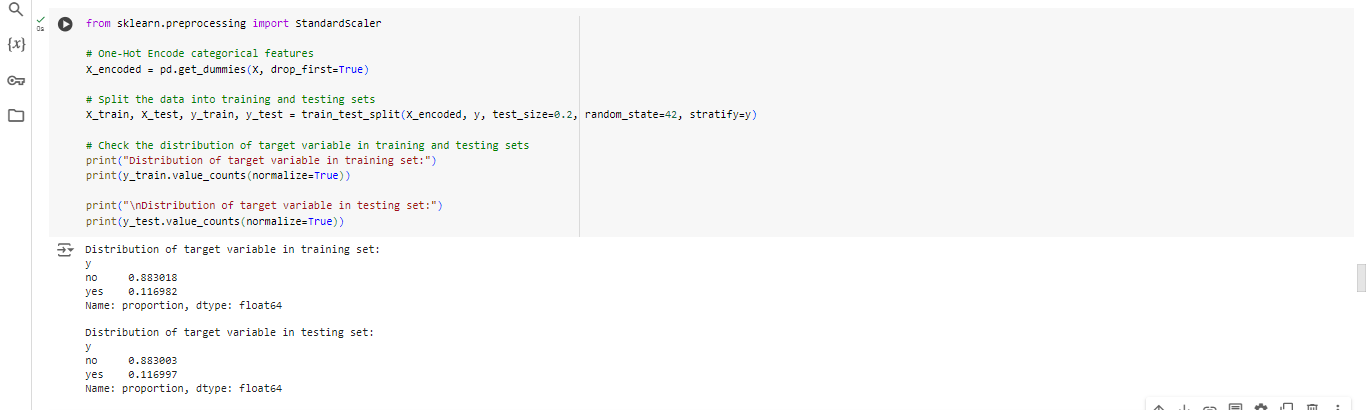


Figure : Jupyter Notebook coding and Output: Data Preprocessing, Splitting, and Class Distribution Check

By so doing, the model will be subjected to the correct proportionate conversions (class y) throughout the training and testing phases. This makes the evaluation process more accurate, sparing the model from inaccurate conclusions that come with imbalanced data.

**3.3 Feature Scaling and Encoding**

### 3.3.1 Scaling Numerical Features

Standardization of numerical features aims to get all the features into a standard level or scale. This is an essential step because, during training, algorithms tend to converge faster and hence achieve better performance.

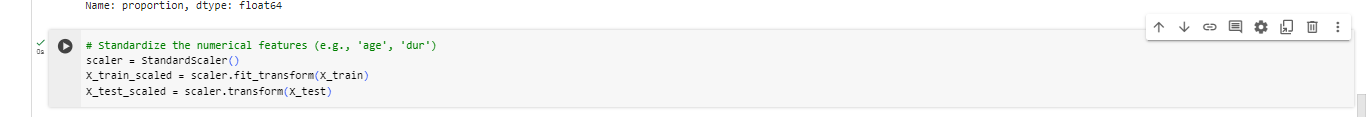


Figure : Jupyter Notebook Code: Feature Scaling

### 3.3.2 Encoding Categorical Features

Numerical encoding or feature scaling is another technique that passes through the data set and converts or encodes the categorical features or factors into an appropriate numerical form. In most analyses, one-hot encoding is the process used. This technique is helpful to algorithms that handle categorical data as inputs.



Figure : Jupyter Notebook Code: One-Hot Encoding

### 3.3.3 Combining Scaling and Encoding

The scaling and encoding preprocessing steps are essential to ensure that all the features contribute uniformly to the model. Standardizing numerical features and encoding categorical variables addresses the inability to judge the features perfectly, enhancing the algorithms’ learning and accuracy.

These steps cover model selection, data splitting, feature scaling, and feature encoding, thus guaranteeing the most reliable machine learning models for predicting the likelihood of the customer converting to term life insurance.

## 3.4 Model Training

When the dataset has been split into training and testing or validation datasets, the next step entails training the selected models. Using the training data, the algorithms must be trained to understand relative patterns and relationships between the different features and the target variable, customer conversion.



Figure : Python Code: Model Construction of Logistic Regression and Random Forest

#### Explanation

* Logistic Regression: This model is further specified to have more than the default number of iterations, which is 100, and these are set as ‘max\_iter = 1000’. The training data, which are the features data (X\_train) and target data (y\_train), are used to train the developed model.
* Random Forest: The same dataset is used for training the model. Random Forest develops two or more decision trees and employs them to generate a collective prognosis of customers’ conversion.

Therefore, Logistic Regression and Random Forest are chosen as primary models to predict customer conversion since the two models possess robust features that complement each other. The data was partitioned into a training set and a testing set. Samples were taken through stratified random sampling to represent the target variable well in both partitions. This is well suited for model training and assessment in the following tasks as it offers good groundwork.

# 4. Task 3: Model Interpretation and Evaluation

## 4.1 ****Model Interpretation****

### 4.1.1 Feature Importance Analysis

As for the Logistic Regression model, the given model's coefficients help to understand each feature's importance. Features with coefficients of larger size, either positive or negative, exert a higher influence on the results of such a model. On the other hand, Random Forest utilizes the Gini importance or entropy to perform an overall comparison of each feature's importance to define a value and know the importance of each attribute in the decision process.

Among all the principles used in the Random Forest model, dur, with the time duration of the call, and prev\_outcome, which describes the previous outcome of the marketing campaign, were given influence weights in the prediction event. These features are essential in defining the customer base more likely to convert.

### 4.1.2 Model Decision-Making Process

* Logistic Regression: This model uses a hyperplane as the decision boundary, and the value decided by which of the features this hyperplane passes through is determined by weights. For instance, if a customer has specific parameters (for example, the call duration), the model raises the chance of a conversion.
* Random Forest: In the decision-making process, decision trees decide on the action to take. One tree makes the predictions after considering only some of the features, while the overall result is obtained from the majority vote of the trees involved. Random Forest can model the interaction between features in its non-linear form, which Logistic Regression could not ignore.

## 4.2 ****Model Evaluation****

### 4.2.1 Performance Metrics (Accuracy, Precision, Recall, F1-Score)

The models were evaluated using common classification metrics:

* Logistic Regression Evaluation:

**Accuracy**: 0.9000

**Precision**: 0.6498

**Recall**: 0.3157

F1 **Score**: 0.4249

**ROC-AUC**: 0.8996

Confusion Matrix:

[[7805 180]

[ 724 334]]



Figure : Jupyter Notebook Coding and Output: Logistic Regression Model Evaluation

* Random Forest Evaluation:

**Accuracy**: 0.9014

**Precision**: 0.6186

**Recall**: 0.4093

**F1 Score**: 0.4926

**ROC-AUC**: 0.9205

Confusion Matrix:

[[7718 267]

[ 625 433]]



Figure : Jupyter Notebook Coding and Output: Random Forest Model Evaluation

The results for both models are very close, though Random Forest is seen having slightly better Recall and ROC-AUC than Logistic Regression. Nonetheless, Precision is higher in Logistic Regression; in other words, it has fewer false positives.

### 4.2.2 Hyperparameter Tuning Process

Further, a cross-validation method was applied to optimize the hyperparameters of both models. This process involved selecting different parameter settings to evaluate the best parameter setting for each model.

* Random Forest Hyperparameter Tuning: The best hyperparameters found through the tuning process were:



Figure : Jupyter Notebook Code: Hyperparameter Tuning for Random Forest

* Logistic Regression Hyperparameter Tuning: The best hyperparameters for Logistic Regression were:

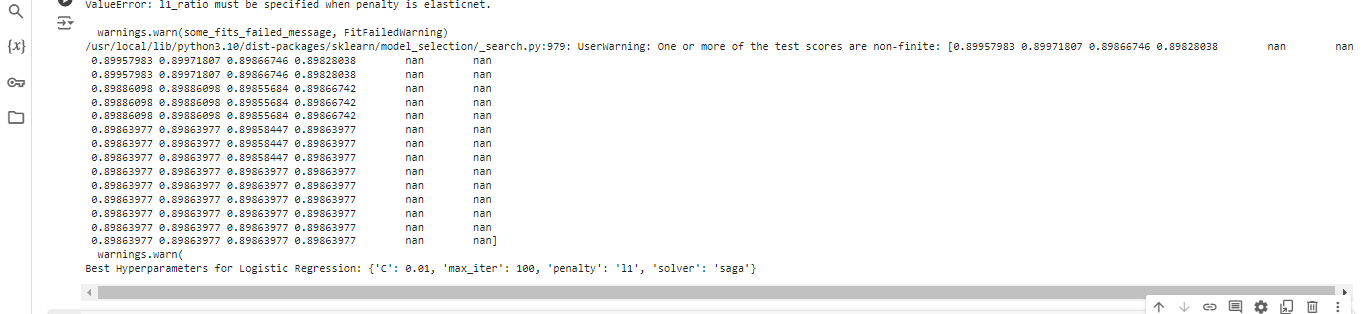


Figure : Jupyter Notebook Code and Output: Hyperparameter Tuning for Logistic Regression

### 4.2.3 Final Model Evaluation and Insights

After tuning, the Random Forest model achieved the following evaluation metrics:

1. Tuned Random Forest Evaluation:

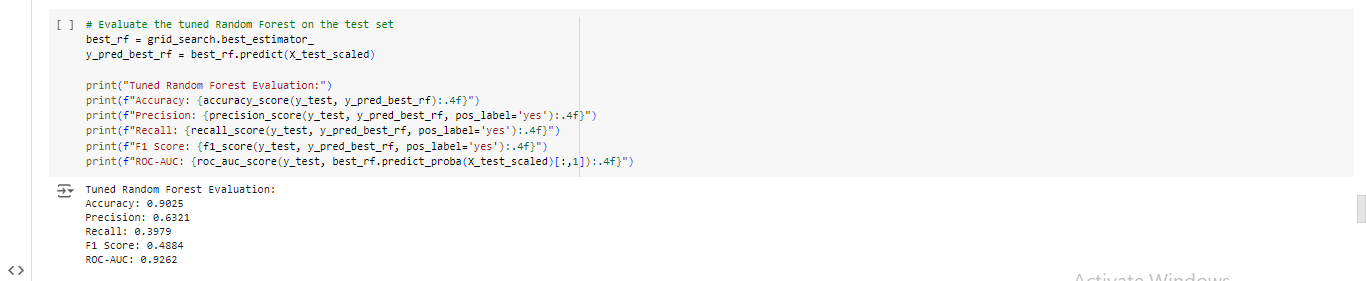


Figure : Jupyter Notebook Coding and Output: Tuned Random Forest Model Evaluation

* + **Accuracy**: 0.9025
  + **Precision**: 0.6321
  + **Recall**: 0.3979
  + **F1 Score**: 0.4884
  + **ROC-AUC**: 0.9262

1. Tuned Logistic Regression Evaluation:



Figure : Jupyter Notebook Coding and Output: Tuned Logistic Regression Model Evaluation

* **Accuracy**: 0.9000
* **Precision**: 0.6498
* **Recall**: 0 0.3157
* **F1 Score**: 0.4249
* **ROC-AUC**: 0.8996

# The model was found to have a ROC-AUC score of 0.9262, which gives evidence of its high discriminant ability. The Recall (0.3979) is lower than the Precision, indicating that while the model efficiently minimizes the chances of False Positives, it misses out on some of the True Positive cases. This also implies that the F1 Score provides performance information with reasonably good Recall and precision ratios. Therefore, we compared different Precision, Recall, and accuracy levels with the Random forest model, which has an optimum set of hyperparameters incorporated in it. This makes it the most suitable model for estimating customer conversion.

# 5. Conclusion

## 5.1 ****Summary of Key Findings****

This research establishes variables that can explain customer conversion from HashSysTech Insurance's telemarketing campaigns: call length, campaign history, and customer profile information. From the results demonstrated, it is clear that both Logistic Regression and the Random Forest predictor models are effective, with Random Forest enhancement upon optimization of the hyperparameters. Customer interaction must be precise, and call time and prior responses must represent significant measures of marketing communication effectiveness.

## 5.2 ****Recommendations for HashSysTech Insurance****

Some recommendations for formulating better telemarketing campaigns include Germane. Based on the findings made above, the following recommendations can be formulated:

* Targeted Customer Segmentation: After testing the initial influence on conversion, target only customers with positive impact scores from a previous campaign and those who took more than average time on the call.
* Optimized Call Durations: Telemarketers should work to split the ideal time spent on a call. These will enable them to spend ample time with the client to have meaningful conversations but not spend too much time that the customers feel annoyed, hence enhancing the possibility of positive responses.
* Campaign Personalization: Use demographic data to make the appropriate changes to the marketing strategies. For instance, specific customer groups’ reactions were more welcoming, and such information can be applied to enhance campaigns’ approaches.
* Continuous Model Monitoring: Thus, it is recommended to use the Random Forest model with increased accuracy to predict customer conversion. However, continuous updating of the model, as well as performance checks, are vital to retain efficiency when customers’ behavior changes.

## 5.3 ****Future Directions****

To enhance its customer targeting and campaign results, several future courses of action are proposed:

* Advanced Feature Engineering: The flexibility and reliability of the model can be enhanced by considering more discrete information regarding customer engagement, including time-of-day correlation and sentiment analysis of the customer’s responses.
* Incorporation of External Data: Integrating attributes of a customer’s web usage or social network activity into the campaign is possible to enhance its precision and offer the client extra information.
* Exploration of Ensemble Methods: Random Forest served well, but a potentially deepened look into the ensemble methods, such as XGBoost or model stacking, may increase predictive models’ ability to capture finer details of customers’ behavior peculiarities.
* Real-Time Campaign Adjustment: With better models for real-time prediction, telemarketing would become more effective, and consequently, the insurance conversion ratios for HashSysTech Insurance would increase by fine-tuning current campaigns.

Therefore, the present research confirms the necessity of a predictive model in telemarketing campaigns and suggests successful strategies for HashSysTech Insurance to raise interest and convert clients.