

# HEALTH INSURANCE

## Post Graduate Program in Data Science Engineering

Location: **Hyderabad**

Batch: **PGPDSE-FT Nov23**

### Submitted by

Prachi Jain

S. Neha

Diana Caroline

R.V Varsha

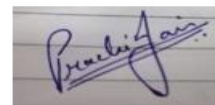
Ch. Sai Kanaka Sireesha

### Mentored by

Mr. P V Subramanian

Date: 10-06-2024

Signature of Mentor



*P.V. Subramanian*  
Signature of Mentor

Signature of Team Leader

## **ACKNOWLEDGEMENT**

Any endeavor in a specific field requires the guidance and support of many people for successful completion. The sense of achievement on completing anything remains incomplete if the people who were instrumental in its execution are not properly acknowledged. We would like to take this opportunity to verbalize our deepest sense of indebtedness to our project mentor, Mr. P V Subramanian, who was a constant pillar of support and continually provided us with valuable insights to improve upon our project and make it a success. Further, we would like to thank our parents for encouraging us and providing us a platform wherein we got an opportunity to design our own project.

## **DECLARATION**

We hereby declare, that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

## **Table of Contents**

1. <a href="#">INTRODUCTION</a> .....	3
<a href="#">1.1 Dataset Information</a> .....	3
<a href="#">1.2 Problem Statement</a> .....	3
2. <a href="#">VARIABLE CATEGORIZATION WITH DESCRIPTION</a> .....	4
<a href="#">2.1 Numerical</a> .....	4
<a href="#">2.2 Categorical</a> .....	5
<a href="#">2.3 Target Variable</a> .....	5
3. <a href="#">DATA PRE-PROCESSING</a> .....	6
<a href="#">3.1 Data Dictionary</a> .....	6
<a href="#">3.2 Datatype Verification</a> .....	6
<a href="#">3.3 Missing Value Treatment</a> .....	7
<a href="#">3.4 Duplicate and Noisy Value Removal</a> .....	7
<a href="#">3.5 Check for Outliers</a> .....	8
4. <a href="#">EXPLORATORY DATA ANALYSIS</a> .....	8
<a href="#">4.1 Univariate Analysis</a> .....	8
<a href="#">4.2 Bi-Variate Analysis</a> .....	15
<a href="#">4.3 Correlation Matrix</a> .....	20
<a href="#">4.4 Multi-Variate Analysis</a> .....	23
<a href="#">4.5 Statistical Tests</a> .....	27
5. <a href="#">BASE MODEL</a> .....	30
<a href="#">5.1 Decision Tree Model</a> .....	30
<a href="#">5.2 Assumptions Check</a> .....	36
6. <a href="#">IMPROVING THE MODELS</a> .....	46

6.1 K-Fold Cross Validation Models.....	46
6.2 Evaluating the KFold Models .....	46
7. <u>HYPERPARAMETER TUNED MODELS</u> .....	48
7.1 Hyper Parameter Tuned Models .....	48
7.2 Evaluating the Tuned Models .....	49
8. <u>INTRODUCING SMOTE</u> .....	51
8.1 Application of SMOTE .....	51
8.2 Models with SMOTE .....	52
9. <u>MODEL'S CONCLUSION</u> .....	52
10. <u>PROJECT'S CONCLUSION</u> .....	53
11. <u>RECOMMENDATIONS</u> .....	54
12. <u>LIMITATIONS, CHALLENGES &amp; SCOPE</u> .....	56

# 1. INTRODUCTION

In recent years, there has been a significant surge in global health insurance subscriptions, paralleled by a rise in health concerns ranging from basic ailments to severe medical emergencies. With the unpredictability and potential financial burden associated with health issues, insurance coverage has become increasingly indispensable.

Health Insurance serves as a vital safety net, offering financial protection and access to essential healthcare services for individuals and families. Insurance companies are keen on capitalizing on this trend, aiming to expand their policyholder base by leveraging data-driven insights. By leveraging the wealth of data available to them, insurance companies can gain valuable insights into customer behavior, preferences, and risk profiles.

In our project, our objective is to analyze a dataset related to Health Insurance policies with the aim of understanding factors influencing policyholder outreach. By analyzing various factors influencing insurance subscriptions and customer behavior, we aim to provide actionable insights that will optimize marketing strategies, enhance customer engagement, and drive competitive advantage. Through data analysis and modeling techniques, we seek to uncover key drivers of insurance uptake and refine policy offerings to cater to evolving consumer needs. By using data analytics, we aim to help insurance companies succeed in the healthcare industry and grow sustainably despite competition.

## 1.1 Dataset Information

The dataset comprises insurance policy recommendations and responses from individuals, with various demographic and policy-related attributes recorded. Each row represents a unique individual, with features such as age, accommodation type, health status, and policy details. It consists of two subsets: Individual policies and Joint policies. The structure of the subsets consists of [40536 number of rows] observations for Individual policies and [10346 number of rows] observations for Joint policies, resulting in a total of 50882 observations. The dataset aims to predict individuals' responses to policy recommendations, aiding insurance companies in targeting strategies and customer engagement. Key variables include city code, region code, recommended insurance type, and policy premium.

## 1.2 Problem Statement:

To develop a predictive model that accurately predicts who are likely to show interest in the recommended policy based on their demographic and policy-related attributes. This model aims to optimize the marketing efforts of the insurance company by targeting individuals who are more likely to subscribe to the recommended policy. Ultimately, the goal is to increase the conversion rate of policy recommendations, leading to higher sales and revenue for the insurance company while also providing suitable insurance coverage to interested individuals.

Health Insurance												
City_Code	Region_Code	Accommodation_Type	Reco_Insurance_Type	Upper_Age	Lower_Age	Is_Spouse	Health_Indicator	Holding_Policy_Duration	Holding_Policy_Type	Reco_Policy_Cat	Reco_Policy_Premium	Response
C3	3213	Rented	Individual	36	36	No	X1	15	3.0	22	11628	0
C5	1117	Owned	Joint	75	22	No	X2	3	2.0	22	30510	0
C5	3732	Owned	Individual	32	32	No	X2	1	1.0	19	7450	1
C24	4378	Owned	Joint	52	48	No	X1	15	3.0	19	17780	0
C8	2190	Rented	Individual	44	44	No	X2	3	1.0	16	10404	0
C9	1785	Rented	Individual	52	52	No	X2	5	1.0	22	15264	0
C3	679	Owned	Individual	28	28	No	X1	6	3.0	17	10640	0

With the increasing importance of Health Insurance amid rising health concerns globally, insurance companies are keen to expand their market reach. They require actionable insights derived from data analysis to effectively promote their policies and attract more customers. Our task is to analyze company data and identify factors that significantly impact the uptake of health insurance policies. By doing so, we aim to provide clear recommendations on how companies can enhance their marketing efforts and increase their customer base's deposit, or at least until the maturity date, unless the investor is willing to pay a penalty.

## 2. VARIABLE CATEGORIZATION WITH DESCRIPTION:

The dataset comprises 14 variables, with 13 independent variables and 1 target variable. These variables encompass a mix of numerical and categorical types. Below is the categorization of variables along with their descriptions for the Health Insurance Classification dataset:

### 2.1 Numerical Columns:

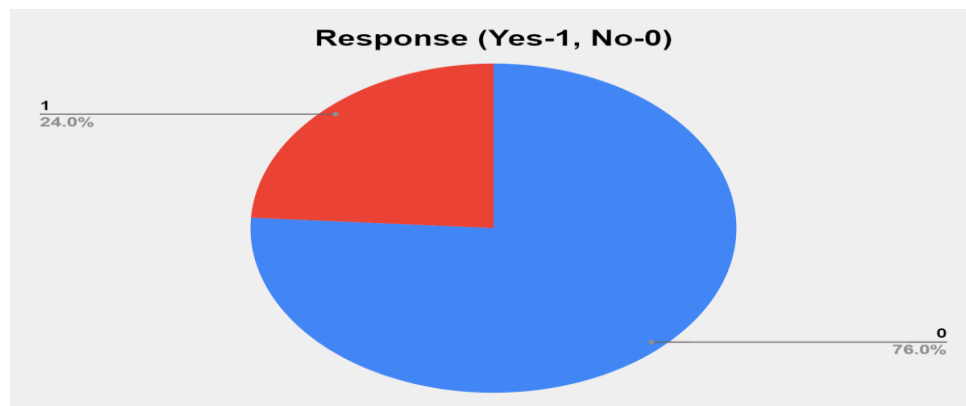
Column Name	Data Type	Column description
ID	Integer	Unique identifier for each individual
Region_Code	Integer	Code representing the region of the individual.
Upper_Age	Integer	Upper age limit of the individual.
Lower_Age	Integer	Lower age limit of the individual.
Holding_Policy_Type	Float	Type of holding policy.
Reco_Policy_Cat	Integer	Category of recommended policy.
Reco_Policy_Premium	Float	Premium amount for the recommended policy.
Response	Integer	Customer showing interest in the recommended policy recommendation (0 = No, 1 = Yes)

## 2.2 Categorical Columns:

Column Name	Data Type	Column description
City_Code	Character	Code representing the city of the individual.
Accommodation_Type	Character	Type of accommodation (Owned, Rented).
Reco_Insurance_Type	Character	Type of recommended insurance (Individual, Joint).
Is_Spouse	Character	Whether the individual is a spouse or not.
Health Indicator	Character	Indicator of the individual's health status.
Holding_Policy_Duration	Character	Duration of holding policy.

## 2.3 Target Variable:

The target variable of the above dataset is Response. This variable has two classes, namely: 0 (Customer did not show interest in the recommended policy) and 1 (Customer showed interest in the recommended policy).



- In the above dataset, 76% of the Responses are No, where the customers did not show interest in the recommended policy and 24% of the Responses are Yes, where the customers showed interest in the recommended policy. We observe that there is a moderate amount of class imbalance.



### 3. DATA PRE-PROCESSING:

Data preprocessing is essential for preparing the Health Insurance Dataset for analysis and modeling. It involves handling missing values, encoding categorical variables, scaling numerical features, handling outliers, and splitting the dataset. Missing values are addressed by replacing them with appropriate measures, such as mean or median for numerical variables and a new category label for categorical variables. Categorical variables are encoded into numerical format using techniques like one-hot encoding or label encoding. Feature scaling techniques like min-max scaling or standardization are applied to ensure numerical features are on a similar scale. Outliers are identified and either removed or transformed based on their impact on model performance.

Finally, the dataset is split into training and testing sets for model evaluation. The data consists of 50882 rows and 14 columns. Out of these we have 6 categorical columns and 8 numerical columns. These preprocessing steps ensure that the dataset is cleaned, transformed, and ready for further analysis and modeling to derive meaningful insights for health insurance policy recommendation strategies.

#### 3.1 Data Dictionary:

SINo	Column Name	Column description
1	ID	Unique identifier for each individual
2	City_Code	Code representing the city of the individual
3	Region_Code	Code representing the region of the individual
4	Accommodation_Type	Type of accommodation (Owned, Rented)
5	Reco_Insurance_Type	Type of recommended insurance (Individual, Joint)
6	Upper_Age	Upper age limit of the individual
7	Lower_Age	Lower age limit of the individual
8	Is_Spouse	Whether the individual is a spouse or not
9	Health Indicator	Indicator of the individual's health status
10	Holding_Policy_Duration	Duration of holding policy
11	Holding_Policy_Type	Type of holding policy
12	Reco_Policy_Cat	Category of recommended policy
13	Reco_Policy_Premium	Premium amount for the recommended policy
14	Response	Customer showing interest in the recommended policy recommendation (0 = No, 1 = Yes). Our <b>Target variable</b>

- There are a total of 14 variables. This data dictionary provides a comprehensive overview of the variables included in the health insurance dataset, laying the foundation for data exploration, analysis, and modeling.

## 3.2 Datatype Verification

We first check the data types of each of the columns of the data.

Column Name	Data Type
ID	int64
City_Code	object
Region_Code	int64
Accommodation_Type	object
Reco_Insurance_Type	object
Upper_Age	int64
Lower_Age	int64
Is_Spouse	object
Health Indicator	object
Holding_Policy_Duration	object
Holding_Policy_Type	float64
Reco_Policy_Cat	int64
Reco_Policy_Premium	float64
Response	int64

## 3.3 Missing Value Treatment

The next step of data pre-processing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Column Name	Null Value Percentage
Health Indicator	22.976691

Holding_Policy_Duration	39.799929
Holding_Policy_Type	39.799929

The missing value treatment used is a form of random imputation for the variables: 'Health\_Indicator', 'Holding\_Policy\_Duration' and 'Holding\_Policy\_Type'. This approach involves replacing missing values in the variable with randomly selected values from the existing categories, weighted by their respective proportions in the dataset.

### 3.4 Duplicate and Noisy Value Removal

Checking and removal of duplicate rows is important because presence of duplicates can lead us to make incorrect conclusions by leading us to believe that some observations are more common than they really are. In our dataset, we do not have any duplicate rows.

```
data[data.duplicated()==True]
```

ID	City_Code	Region_Code	Accommodation_Type	Reco_Insurance_Type	Upper_Age	Lower_Age	Is_Spouse	Health_Indicator	Holding_Policy_Duration	Holding_Policy_Type

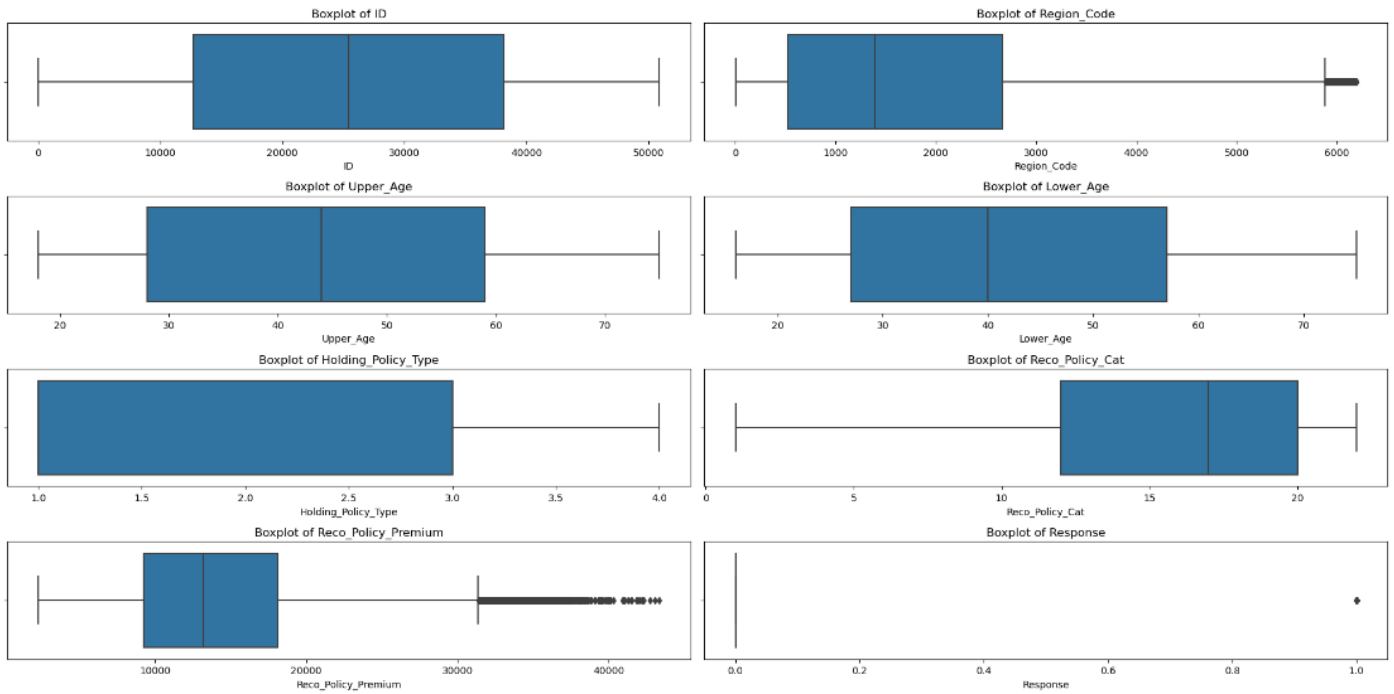
No Duplicate rows observed.

```
data.duplicated().sum()
```

0

### 3.5 Check for Outliers

The variables 'Region\_Code' and 'Reco\_Policy\_Premium' exhibit outliers, yet we choose not to address them, as doing so could result in the loss of valuable data and potentially compromise the efficacy of our modeling approach.



## 4. EXPLORATORY DATA ANALYSIS:

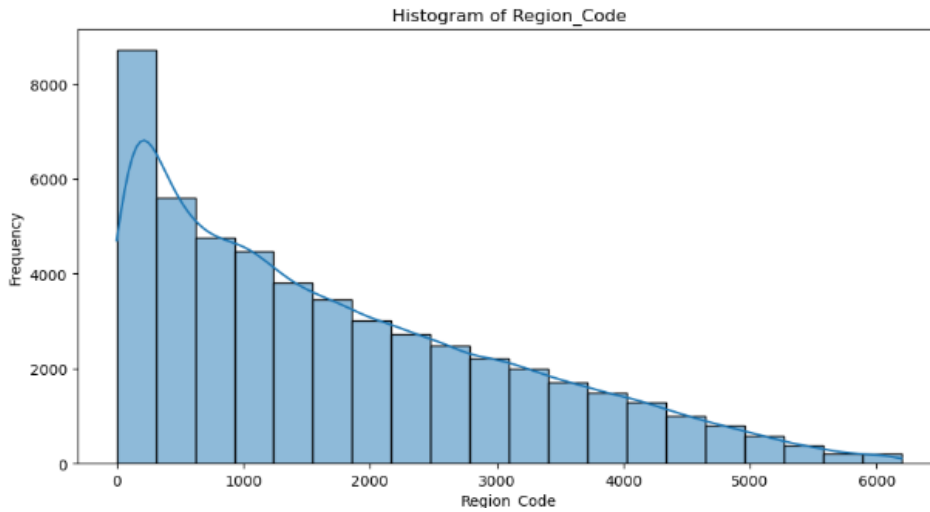
### 4.1 Univariate Analysis

**For Numerical Variables:** Plotting numeric variables with a histogram (hist plot) provides a visual representation of the distribution of values within a dataset.

Column Name	Skewness		Column Name	Kurtosis
ID	0.00000		ID	-1.200000
Region_Code	0.798096		Region_Code	-0.20224
Upper_Age	0.21737		Upper_Age	-1.233867
Lower_Age	0.330594		Lower_Age	-1.170556
Holding_Policy_Duration	0.629064		Holding_Policy_Duration	-0.912926
Reco_Policy_Cat	-0.928224		Reco_Policy_Cat	-0.310533

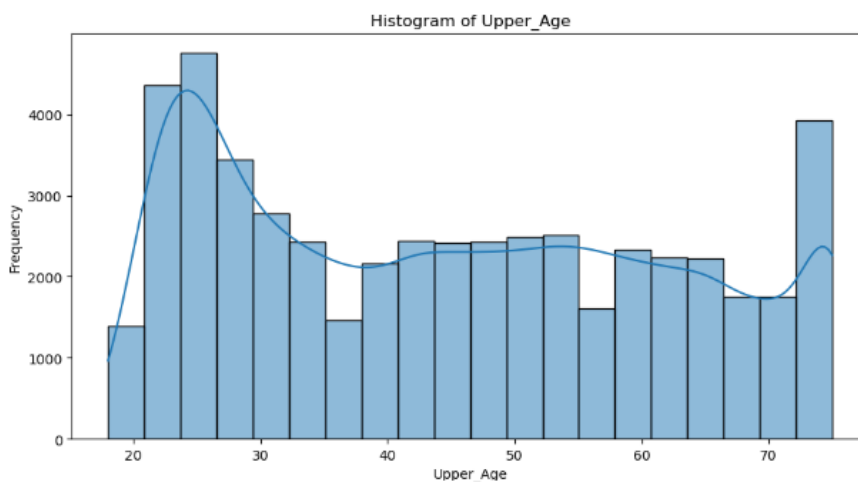
Reco_Policy_Premium	0.782463		Reco_Policy_Premium	0.423053
Response	1.217936		Response	-0.516653

## 1) Region\_Code:



- Skewness = 0.798096
- Kurtosis = -0.202240
- Region\_Code is right skewed.
- It is leptokurtic, and has a wide tail

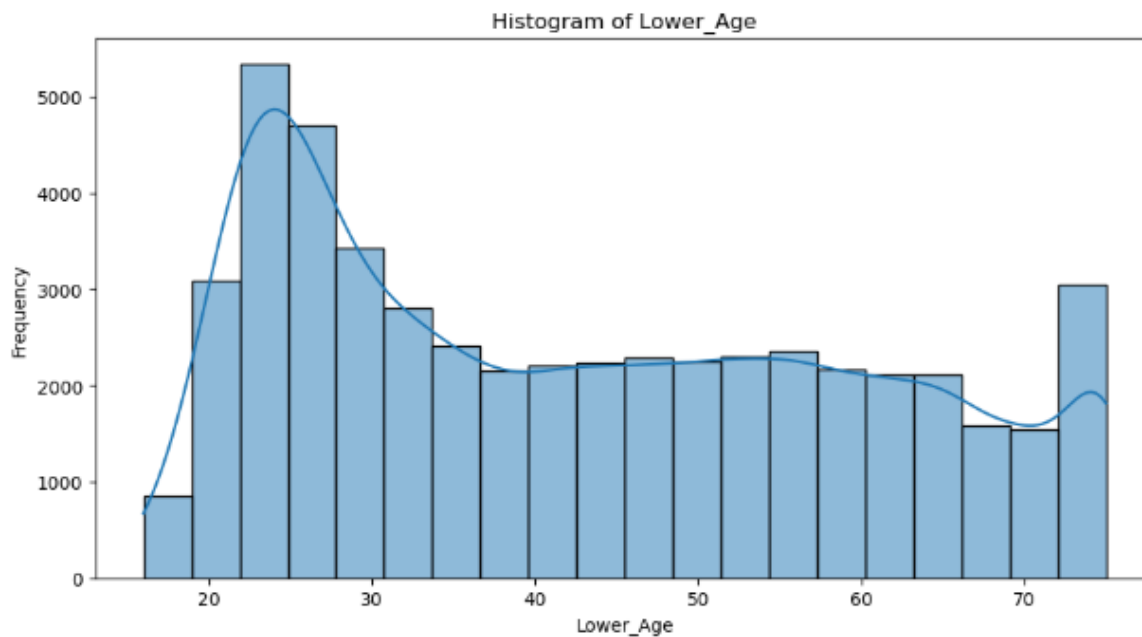
## 2) Upper\_Age:



- Skewness = 0.217370

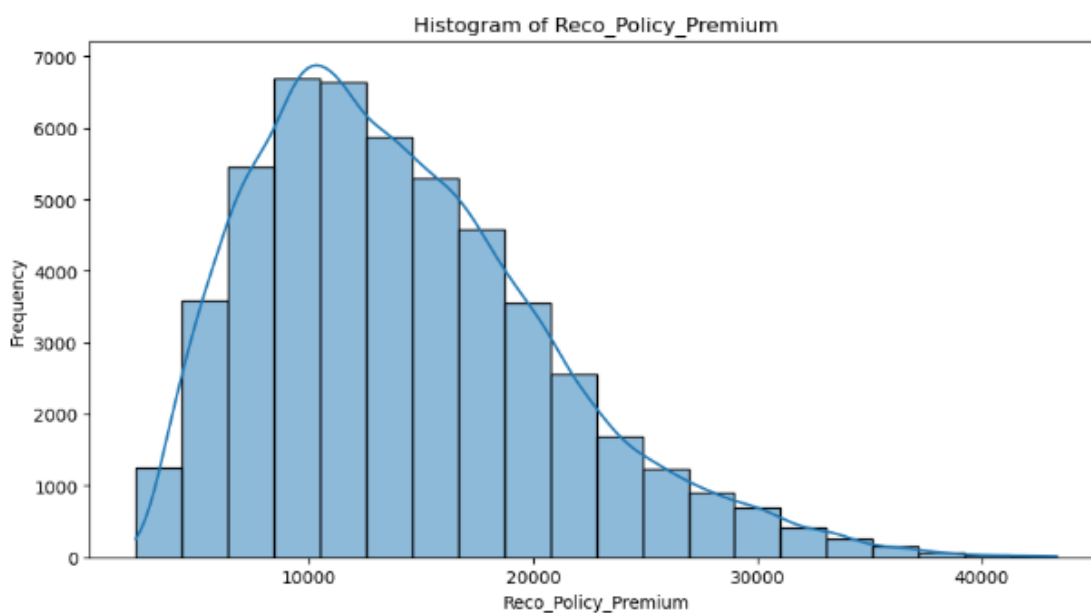
- Kurtosis = -1.233867
- Upper\_Age is a platykurtic curve; positive skewness.

### 3) Lower\_Age:



- Skewness = 0.330594
- Kurtosis = -1.170556
- Lower\_Age is a platykurtic curve; positive skewness.

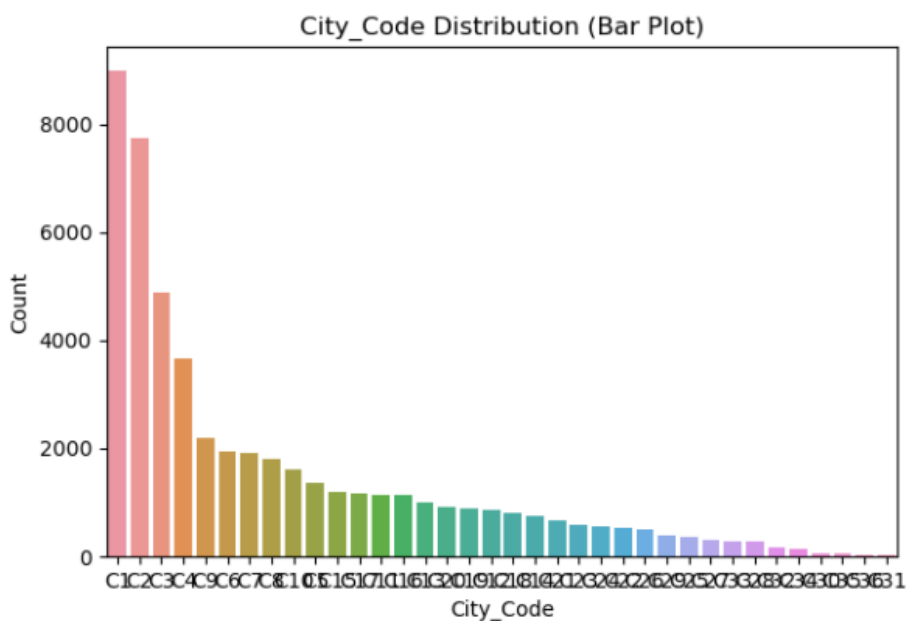
### 4) Reco\_Policy\_Premium:



- Skewness = 0.782463
- Kurtosis = 0.423053
- Reco\_Policy\_Cat demonstrates slight negative skewness, suggesting a concentration towards higher policy categories with fewer instances of lower categories.
- It is leptokurtic, and has a wide tail.

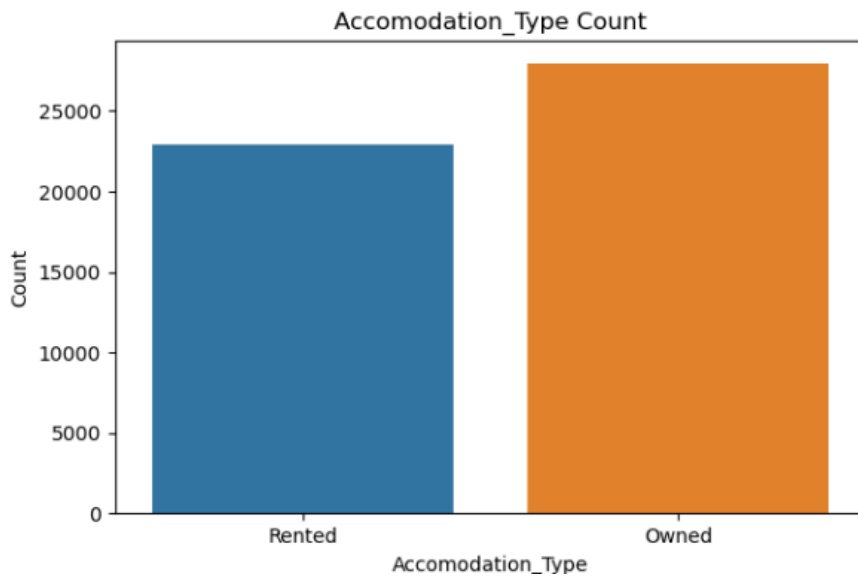
**For Categorical Variables:** We plot bar graphs to understand the distribution of categorical data in the dataset.

### 1) City\_Code:



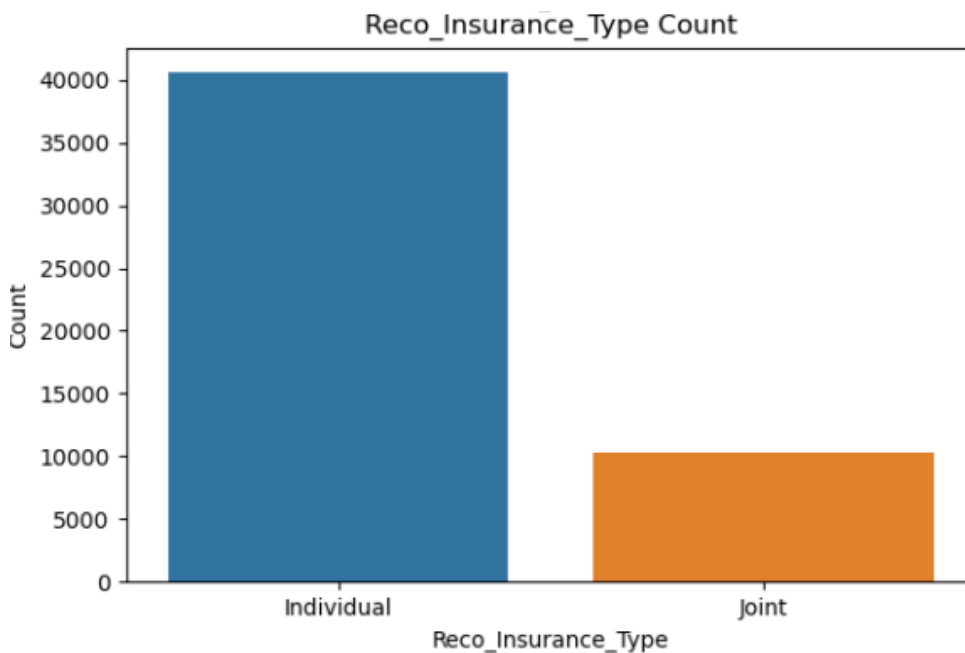
- The distribution of customers varies across different city codes, with some cities(C1,C2,C3,C4) having higher counts than others.

## 2) Accomodation\_Type:



- Majority of the customers own their accommodation rather than renting.

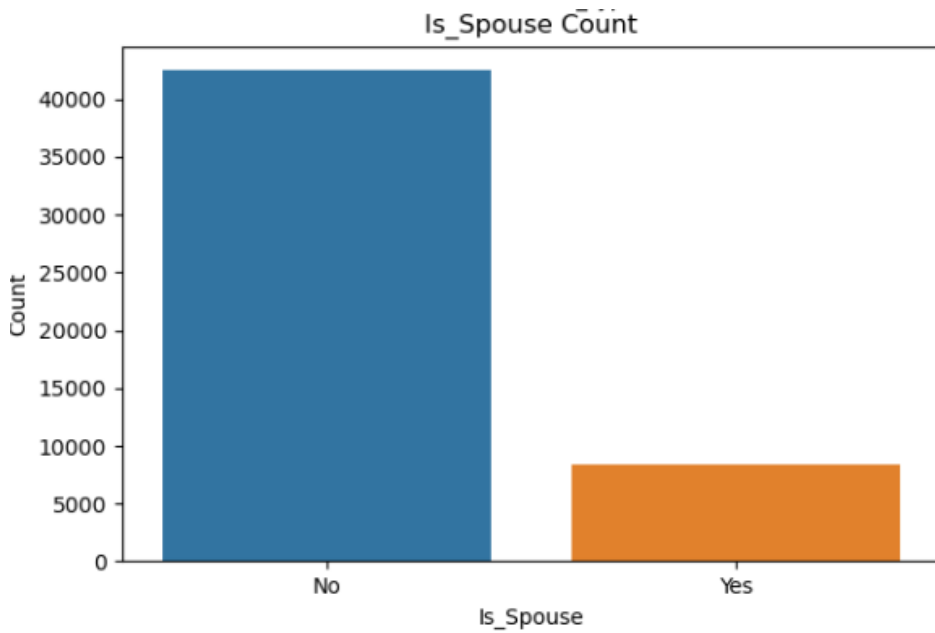
## 3) Reco\_Insurance\_Type:



- Most customers have individual insurance rather than joint insurance.

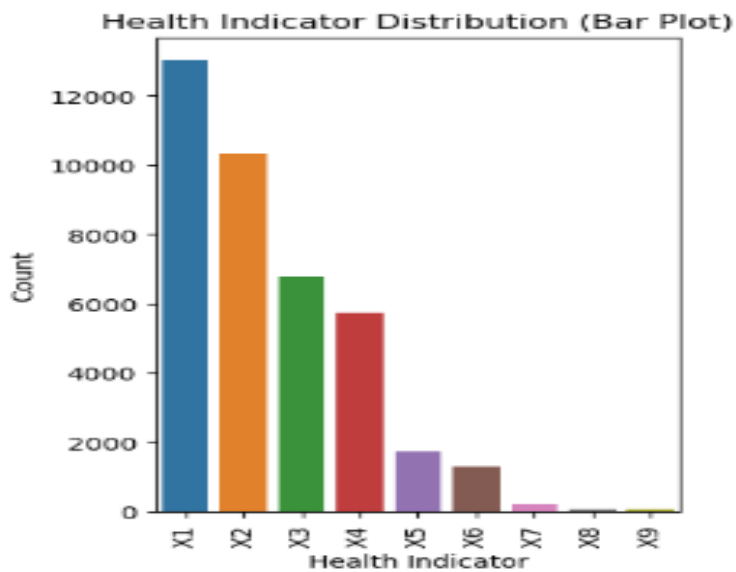
## 4) Is\_Spouse:





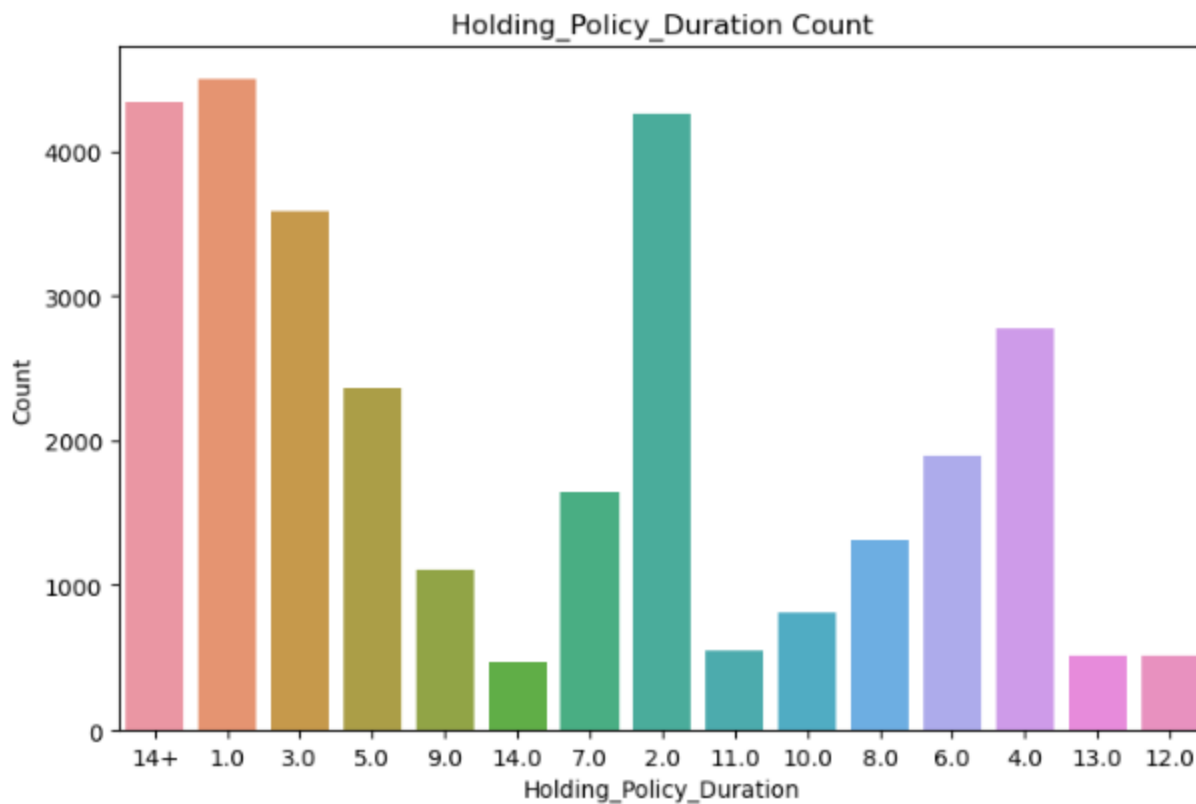
- The majority of customers are not spouses.

#### 5) Health Indicator:



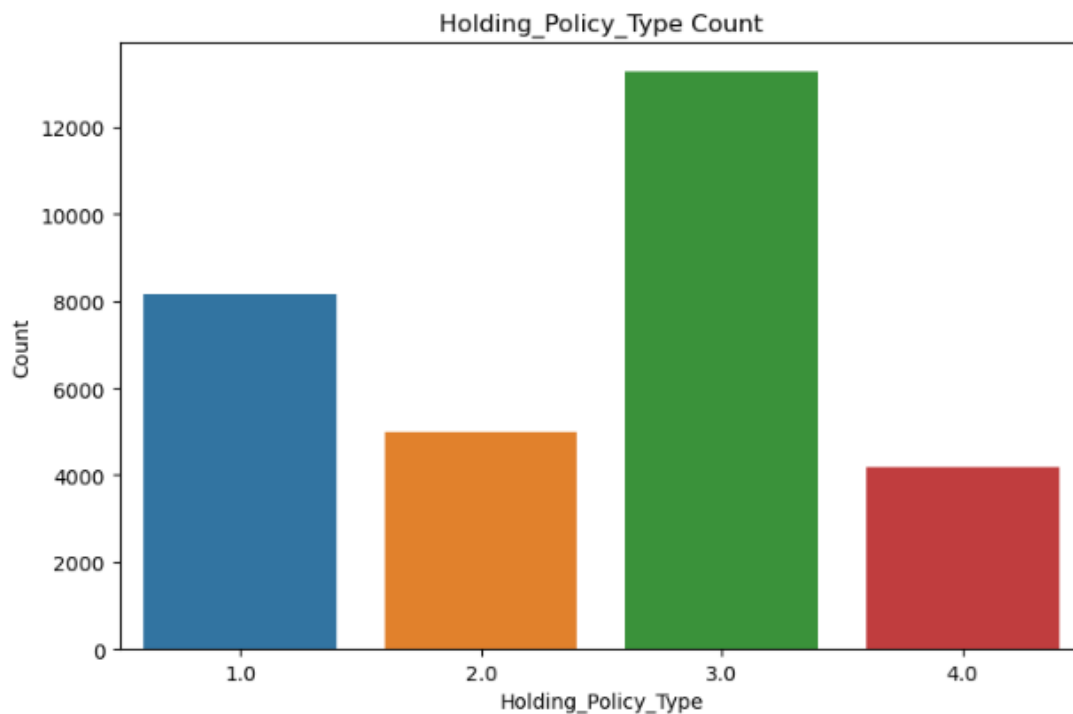
- The distribution of health indicators shows that most customers have health indicator X1 followed by X2, X3 etc.

#### 6) Holding\_Policy\_Duration:



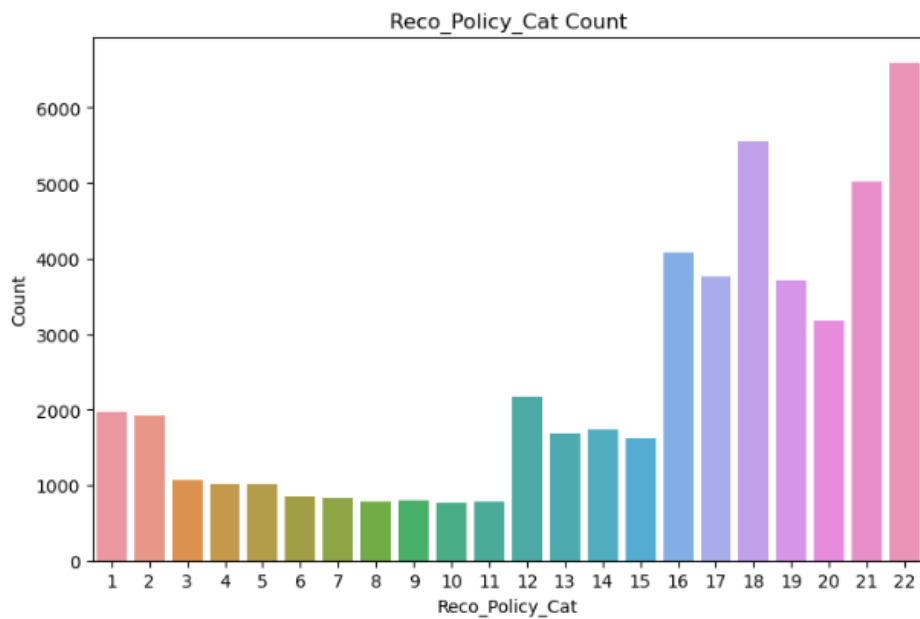
- There is a wide range of holding policy durations, with some durations being more common than others.

#### 7) Holding\_Policy\_Type:



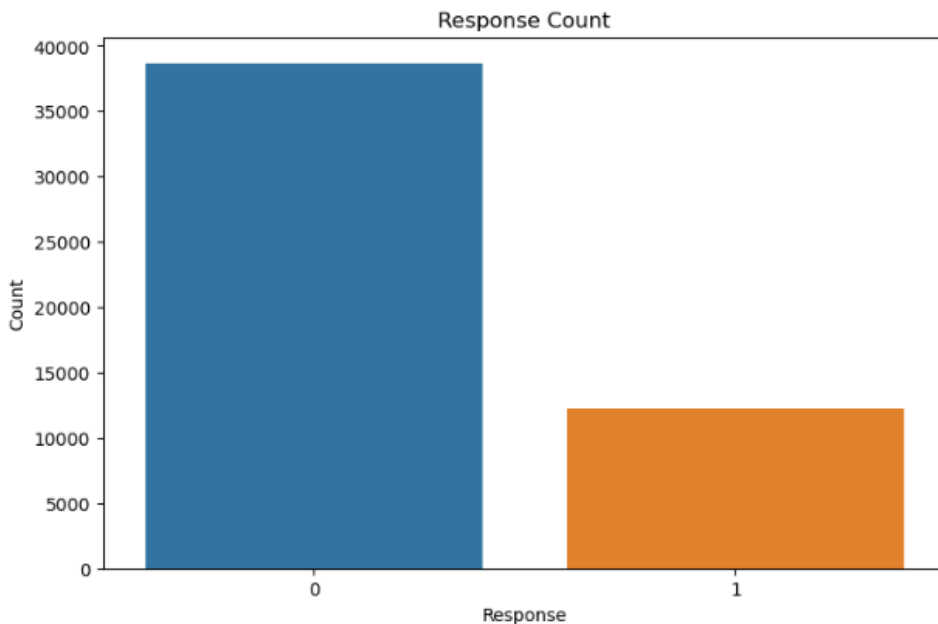
- Holding policy type 3 is the most common among customers.

## 8) Reco\_Policy\_Cat:



- Distribution of recommended policy categories varies, with some categories being more common than others.

## 9) Response:

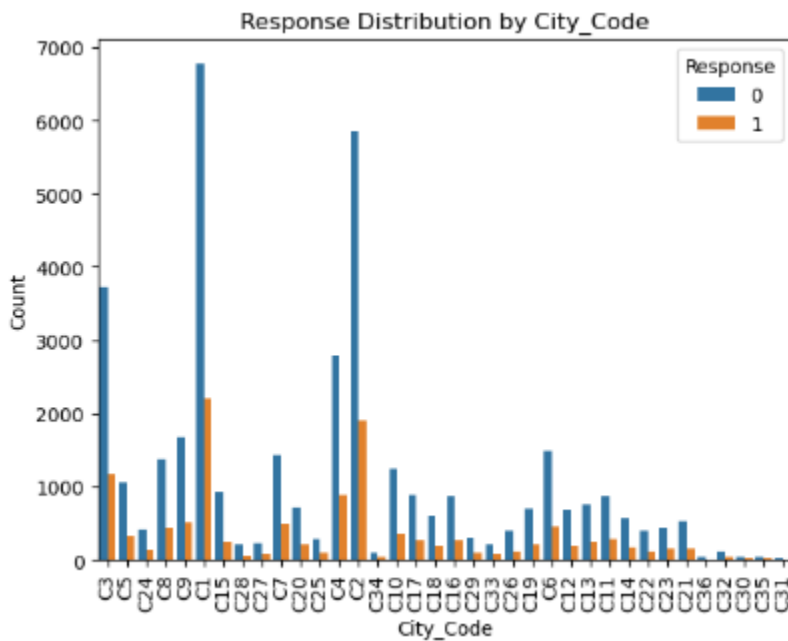


- The distribution of responses indicates that the dataset is imbalanced, with a higher number of negative responses (0) compared to positive responses (1).

## 4.2 Bi-Variate Analysis:

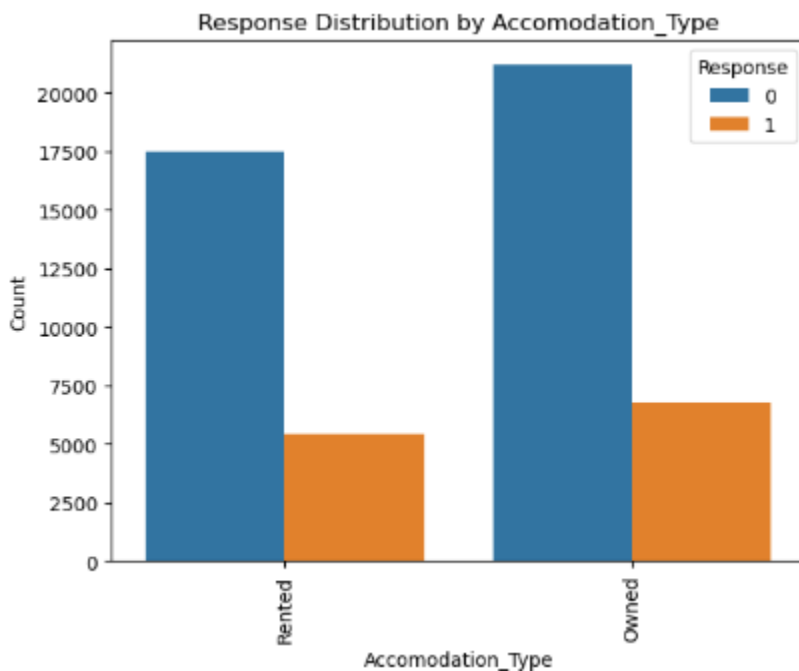
**For Categorical Variables:** We plot count plots to understand the distribution of categorical data with the Target Variable in the dataset.

### 1) City\_Code vs Response:



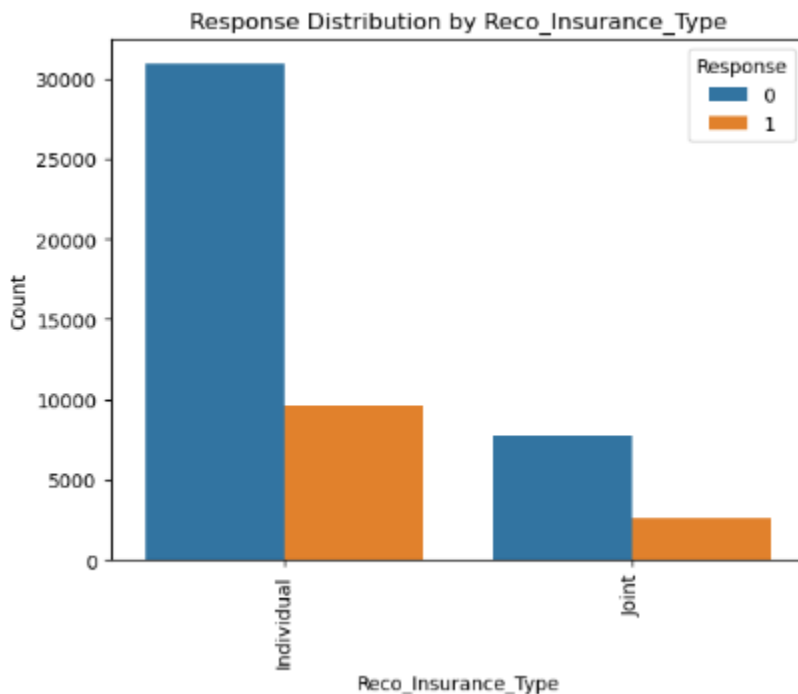
- Certain city codes may have higher response with zero(C3,C1,C2) compared to others. This could indicate regional variations in customer behavior or preferences.

### 2) Accommodation\_Type vs. Response:



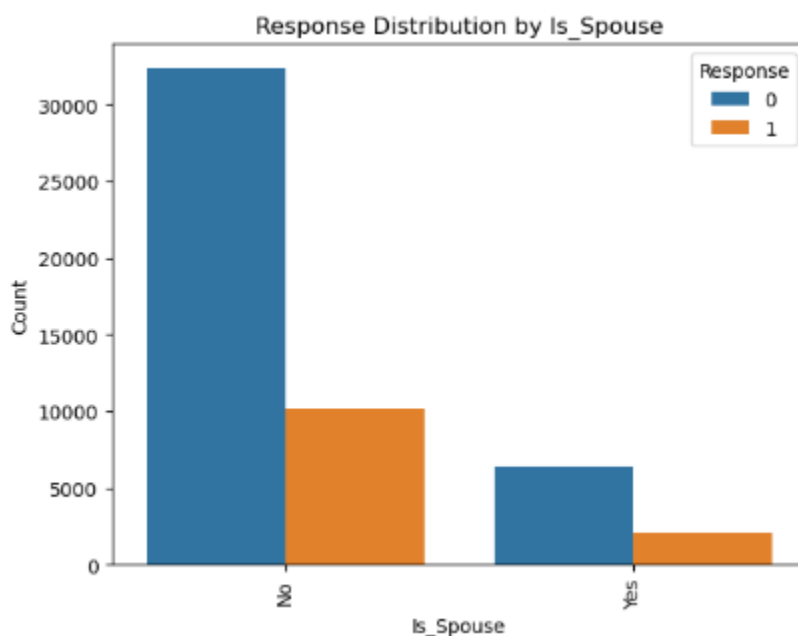
- Customers with different accommodation types (e.g., Owned vs. Rented) may exhibit varying response rates. This could reflect differences in financial stability or risk perception. Accomodation\_Type of Owned have a high response where insurance is not taken.

### 3) Reco\_Insurance\_Type vs. Response:



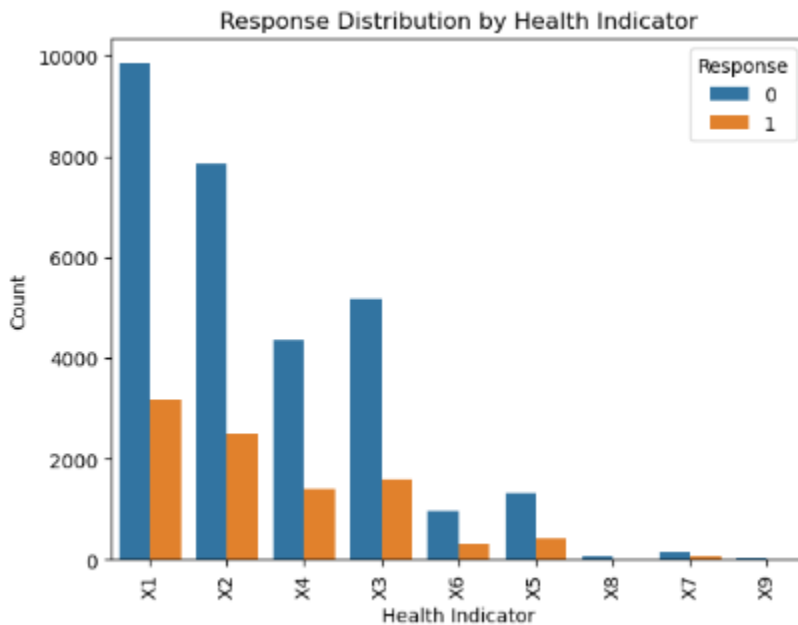
- Individuals recommended for different insurance types (e.g., Individual vs. Joint) may respond differently to the recommendations, possibly influenced by factors like family size or financial planning. Most of the individuals have not preferred any insurances.

### 4) Is\_Spouse vs. Response:



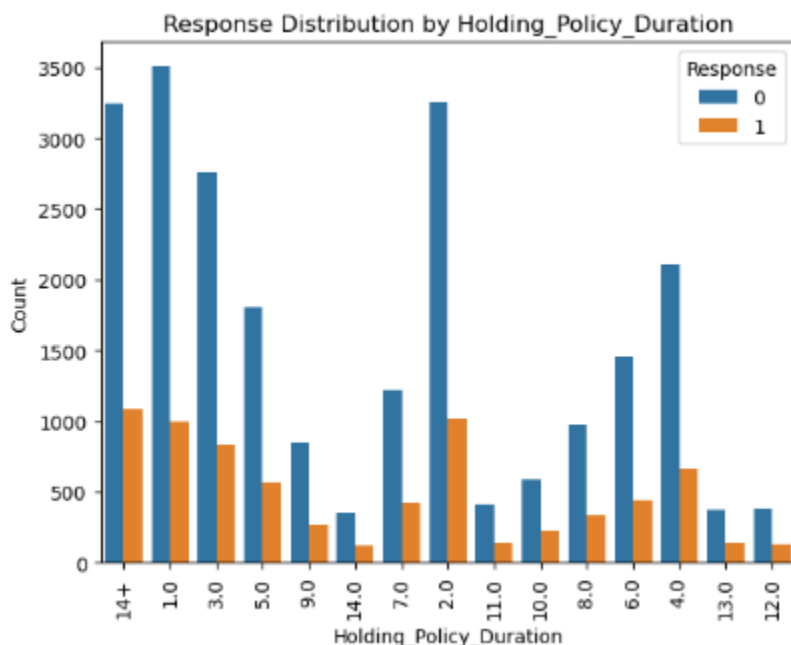
- The response to insurance recommendations might differ based on whether the individual is a spouse or not. This could relate to family-oriented decision-making or financial considerations. Those who don't have a spouse have not preferred to take insurance.

### 5) Health Indicator vs. Response:



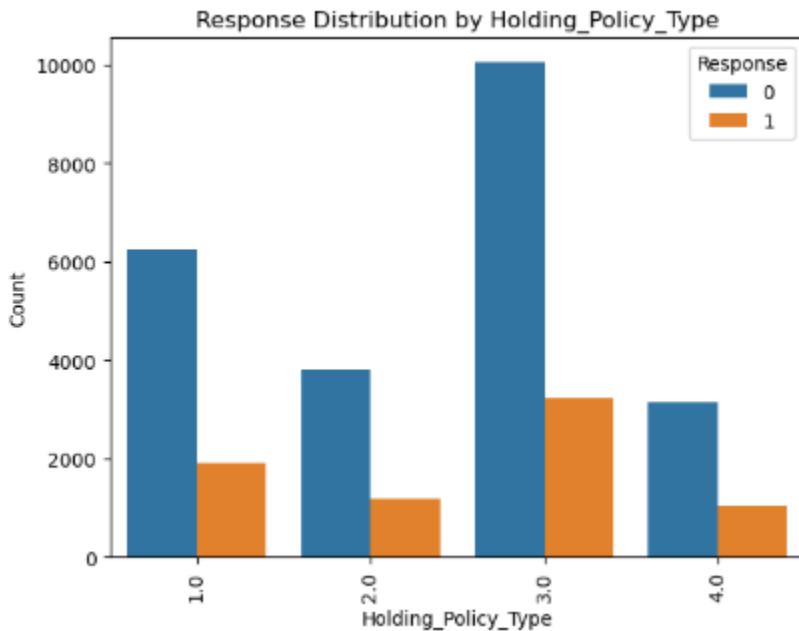
- Health indicators may play a role in determining the response to insurance recommendations, with individuals in better health potentially being more receptive to certain policies.

### 6) Holding\_Policy\_Duration vs. Response:



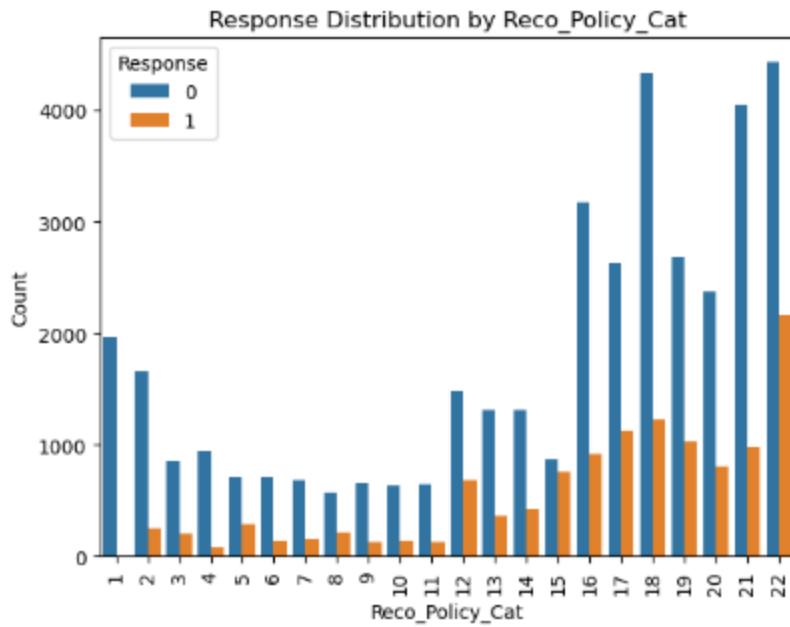
- The duration of holding policies may influence the response to new insurance recommendations, indicating whether individuals with longer-held policies are more or less likely to switch.

### 7) Holding\_Policy\_Type vs. Response:



- Individuals with Holding\_Policy\_Type 3.0 are the most common among the policy holders. This could imply that Holding\_Policy\_Type 3.0 might be a standard or commonly offered policy type by the insurance company, indicating a relatively higher interest among individuals with this policy type.
- Then followed by Holding\_Policy\_Type 1.0, then Holding\_Policy\_Type 2.0 and lastly Holding\_Policy\_Type 4.0

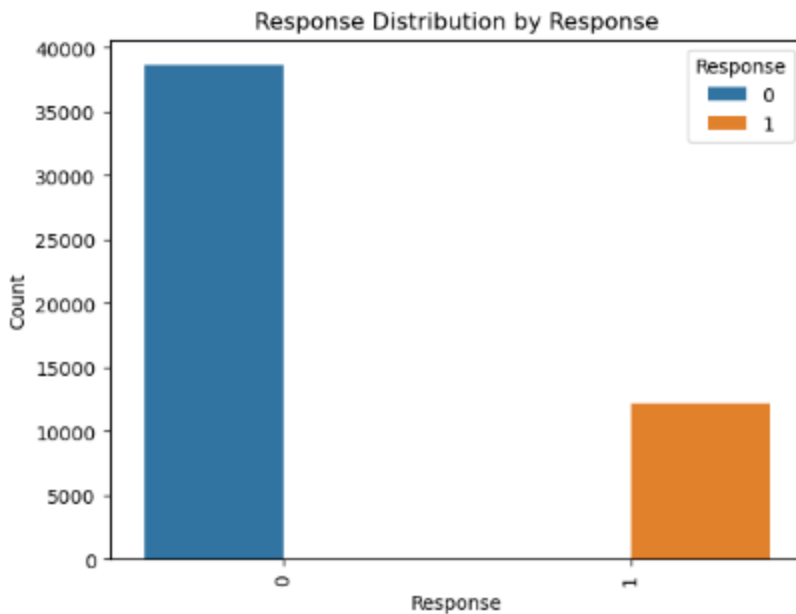
### 8) Reco\_Policy\_Cat vs. Response:



- Categories 22, 18, 21, 16, 17 are the most common among policyholders.
- The distribution of Response across different Reco\_Policy\_Cat categories indicates that certain policy categories may elicit a higher interest level (Response=1) compared to others.

## 9) Response vs Response:





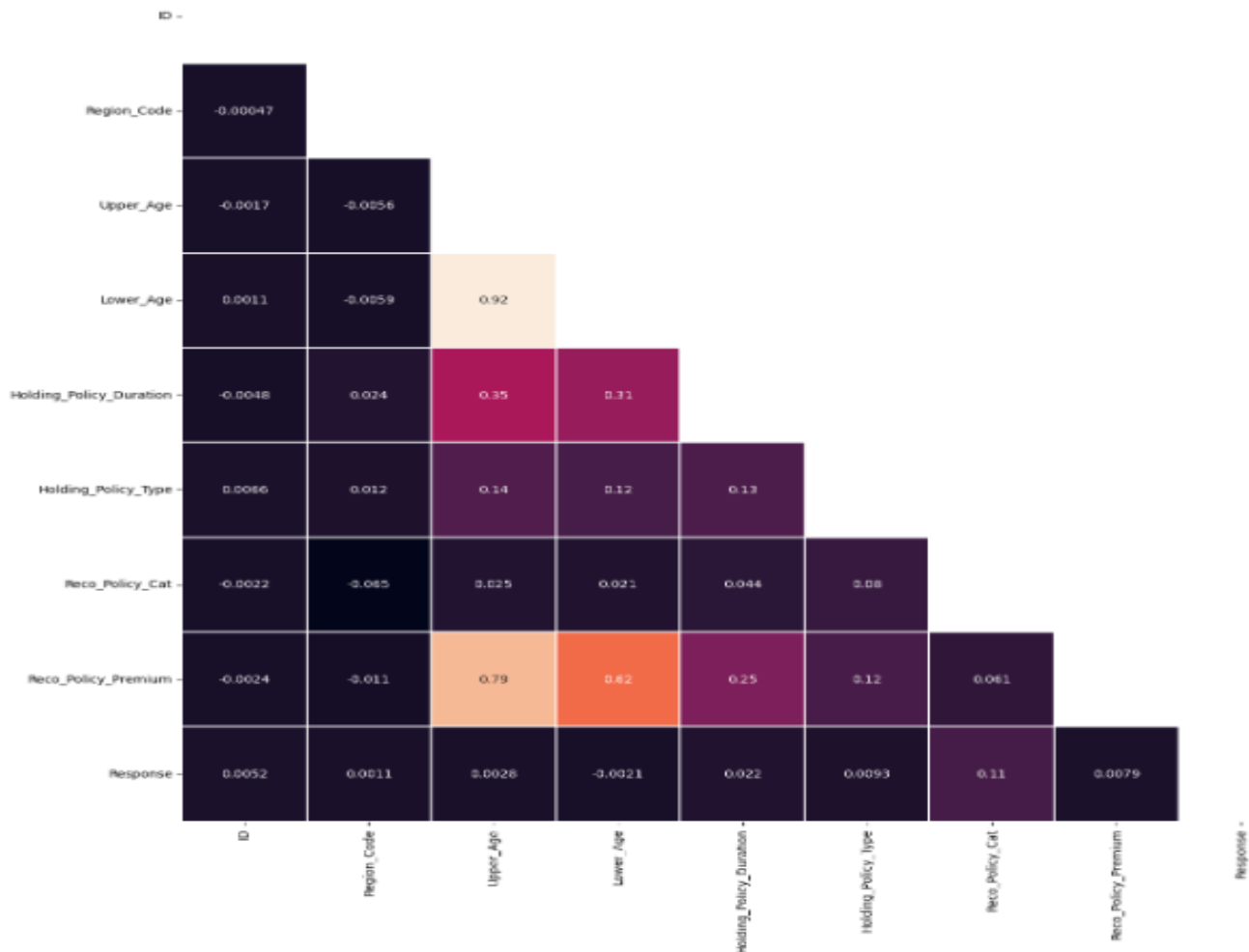
- The majority of the observations in the dataset have a Response value of 0, indicating that the majority of individuals are not interested in the insurance product or offer.
- The lower count for Response value 1 suggests that there is a relatively smaller proportion of individuals who have shown interest in the insurance product or offer.

### 4.3 Correlation Matrix:

#### 1) Heat-Map – Pearson Correlation Matrix:

The correlation matrix heatmap, constructed for the Health Insurance dataset, visually illustrates the relationships between variables. It utilizes the Pearson, Kendall and Spearman correlation coefficient to quantify the strength and direction of linear associations. In the context of this heatmap, it offers insights into how variables such as 'Region\_Code', 'Region\_Code', 'Upper\_Age', 'Lower\_Age', 'Holding\_Policy\_Type', 'Reco\_Policy\_Cat', 'Reco\_Policy\_Premium' and 'Response' interact.

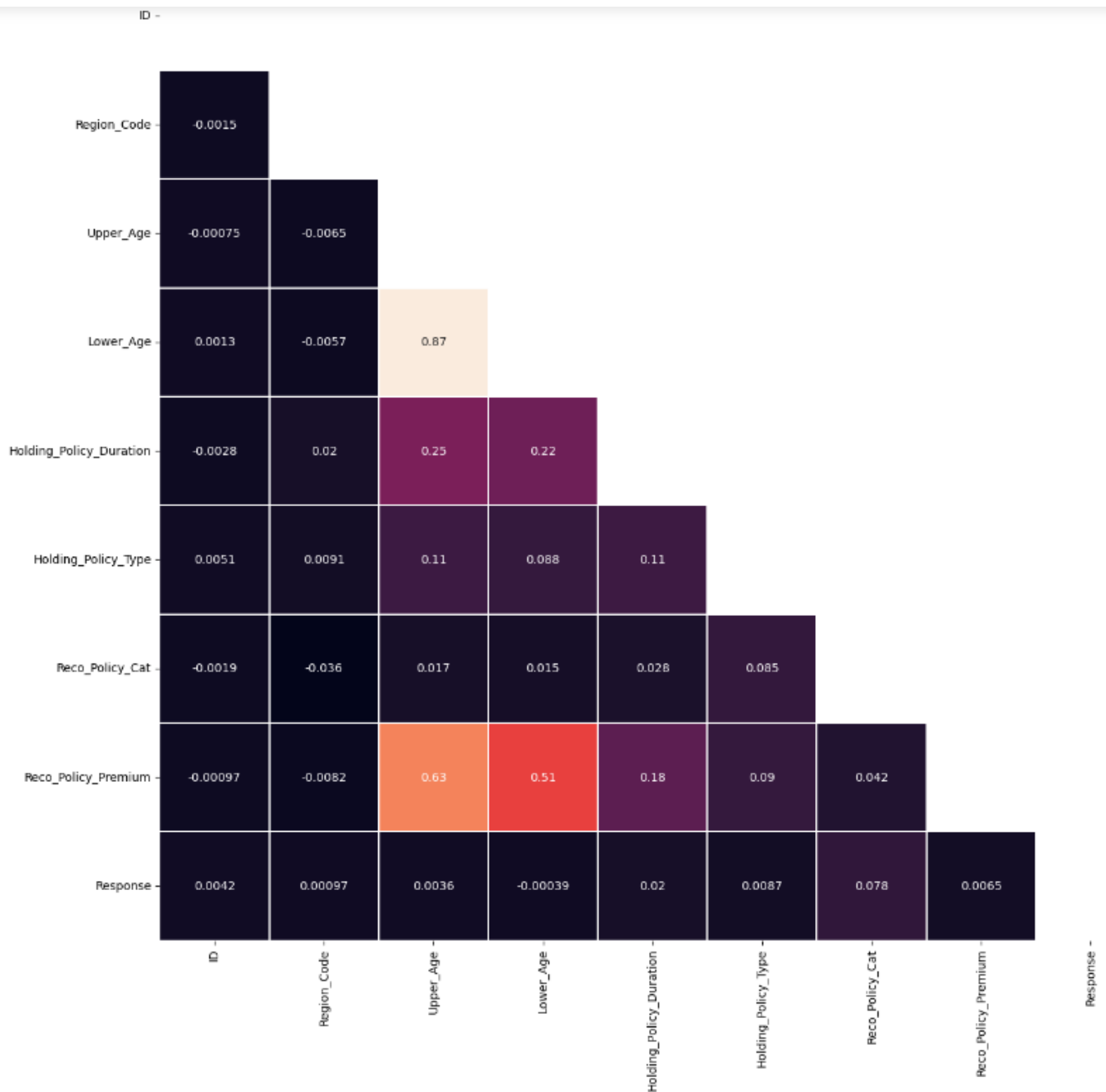
Darker hues denote stronger correlations, while lighter shades indicate weaker correlations. This visualization aids in identifying potential patterns, dependencies, and associations within the dataset, informing strategic decisions regarding risk assessment, pricing strategies, and customer engagement initiatives.



## Pearson Correlation:

- The Pearson correlation heatmap shows the linear relationships between numerical variables.
- Strong positive correlations (values close to 1) are observed between certain pairs of variables, indicating a direct linear relationship. Similarly, strong negative correlations (values close to -1) suggest an inverse linear relationship between variables.
- Variables with weak correlations (values close to 0) are less linearly related. For example, Upper\_Age and Lower\_Age exhibit a strong positive correlation, as expected since they are likely to be closely related. Reco\_Policy\_Cat and Region\_Code also show some level of correlation, albeit weaker.

## 2) Heat-Map – Kendall Correlation:

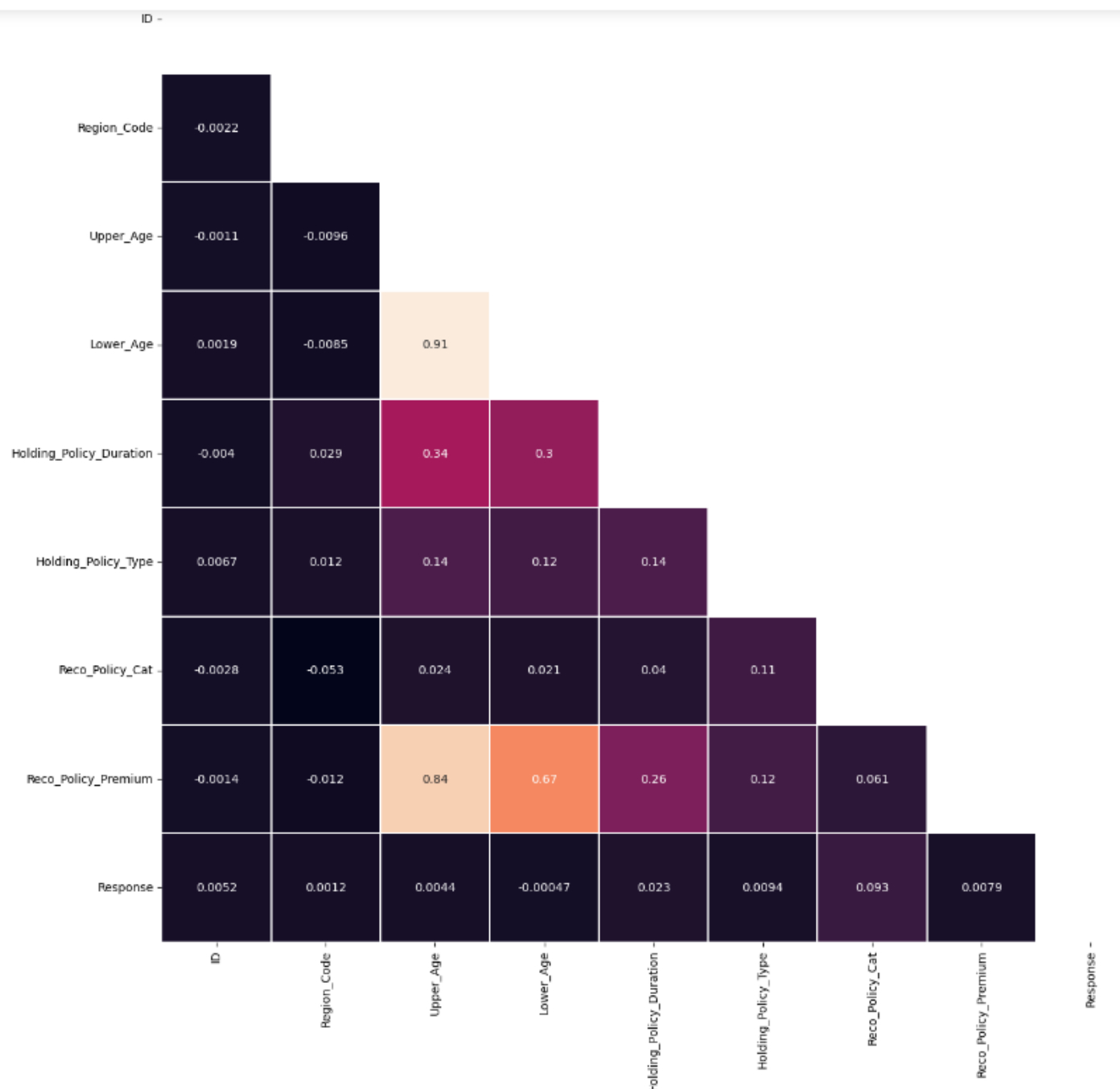


### Kendall Correlation:

- The Kendall correlation heatmap displays the strength and direction of monotonic relationships between numerical variables.
- Similar to Pearson correlation, strong positive and negative correlations are visible, albeit with different values due to the method's focus on monotonicity rather than linearity.

- The heatmap reveals relationships that may not be captured by Pearson correlation, especially for non-linear associations.
- Overall, the patterns and trends observed in the Kendall correlation heatmap complement those in the Pearson correlation heatmap.

### 3) Heat-Map – Spearman Correlation:



### Spearman Correlation:

- The Spearman correlation heatmap illustrates the monotonic relationships between numerical variables, similar to Kendall correlation. It provides insights into the strength and direction of

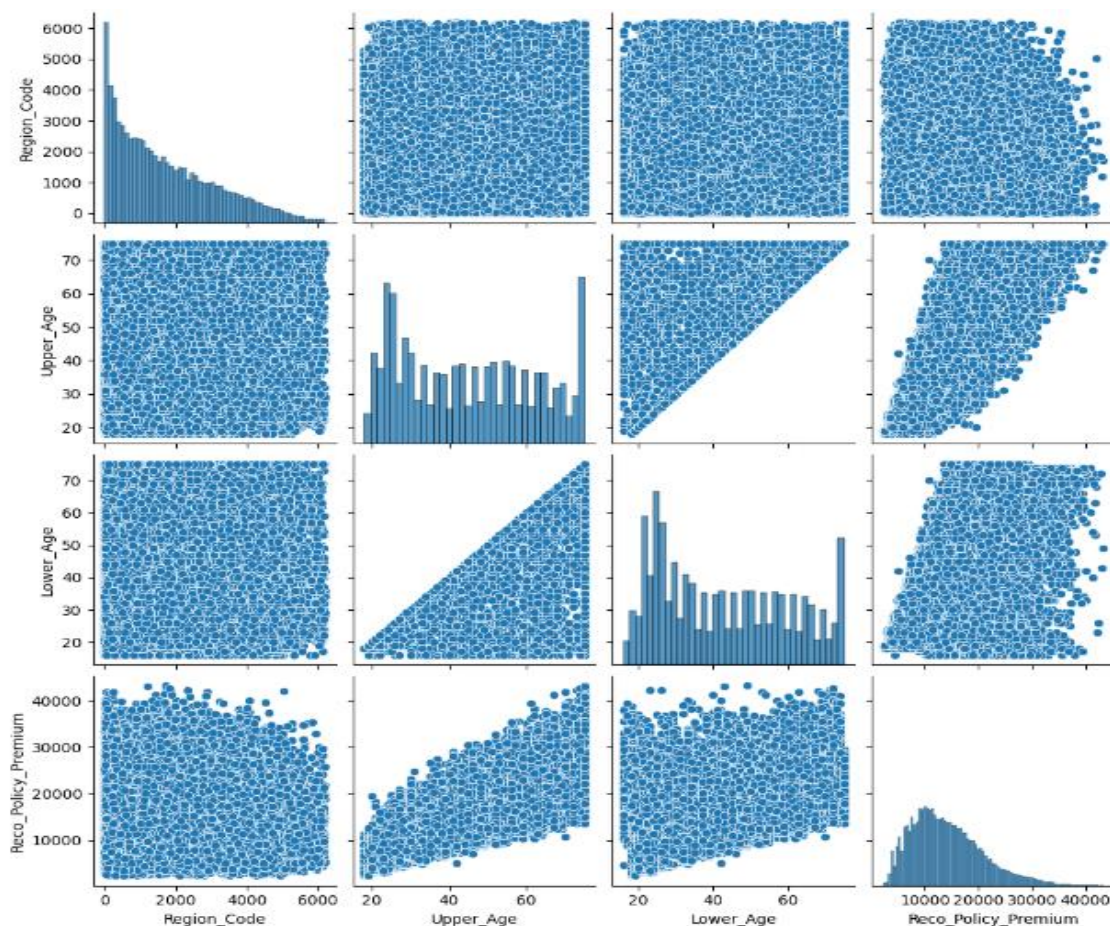
monotonic correlations, which may differ from linear correlations captured by Pearson correlation.

- The Spearman correlation heatmap helps identify ordinal relationships and assesses the consistency of rankings between variables. Like the Kendall correlation heatmap, it offers a different perspective on variable associations, particularly for non-linear relationships.
- By examining these three correlation heatmaps, we can gain a comprehensive understanding of the relationships between numerical variables in the dataset, considering both linear and non-linear associations.

## 4.4 Multi-Variate Analysis:

### For Numeric Variables – Pair Plot:

Plot pair plot to visualize relationships between numerical variables.

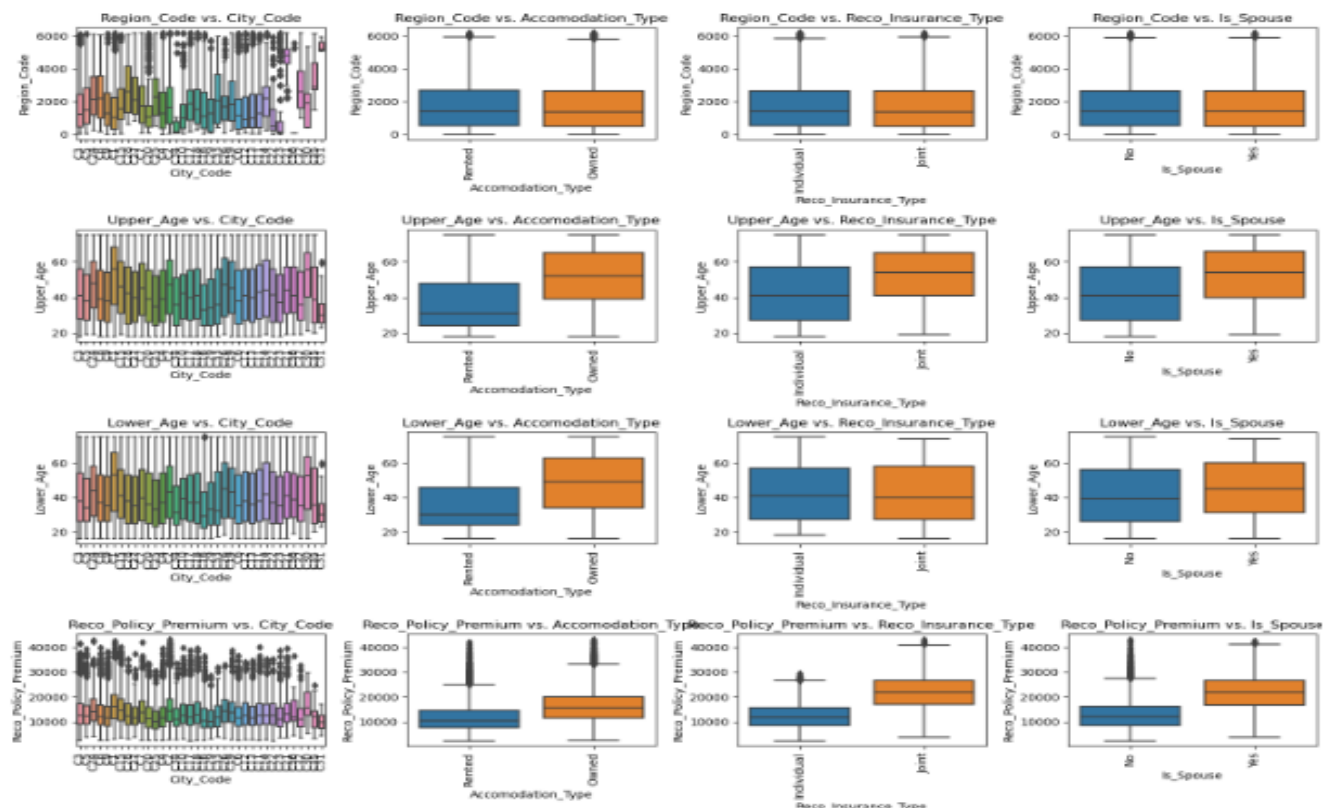


- Region Code vs. Reco Policy Premium: There doesn't seem to be a clear linear relationship between region code and recommended policy premium. The scatter plot appears to show random distribution of points, indicating no significant correlation between these variables.

- Upper Age vs. Lower Age: The scatter plot between upper age and lower age shows a strong positive linear relationship, which is expected since these two variables represent the same individual's age. The points form a straight line with a slope of 1, indicating a perfect positive correlation.
- Upper Age vs. Reco Policy Premium: There appears to be a slight positive correlation between upper age and recommended policy premium. As the upper age increases, the recommended policy premium also tends to increase, although the relationship is not very strong.
- Lower Age vs. Reco Policy Premium: Similar to upper age, there is a slight positive correlation between lower age and recommended policy premium. As the lower age increases, the recommended policy premium also tends to increase, although the relationship is not very strong.
- Overall, the pair plot provides insights into the relationships between numerical variables in the dataset. While some variables exhibit clear correlations, others show more complex patterns or no significant relationships.

## 2) Numeric Vs Categorical – Box Plot:

Box Plots for Numerical vs Categorical variables.

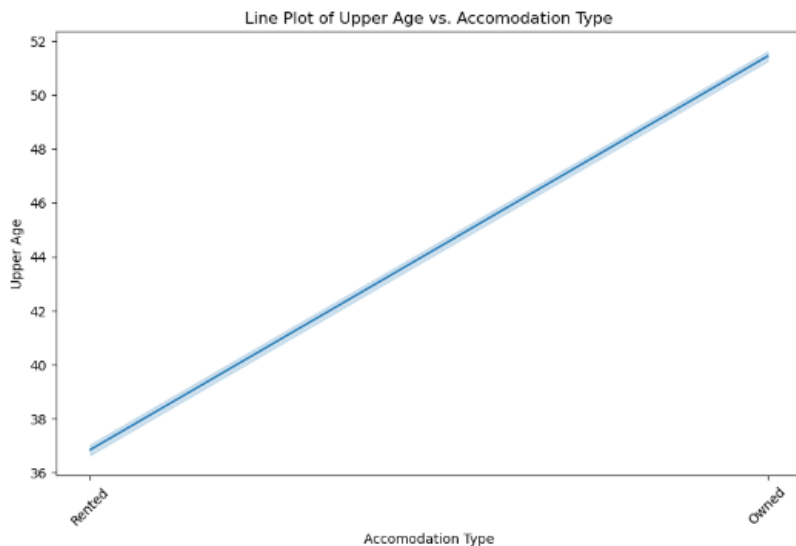


- Boxplots for each numerical variable (e.g., Region\_Code, Upper\_Age, etc.) are plotted against each categorical variable (e.g., City\_Code, Accommodation\_Type, etc.). These boxplots help visualize the distribution of numerical variables within different categories of the categorical

variables. Inferences can be drawn based on the comparison of median values, interquartile ranges, and presence of outliers across different categories of the categorical variables.

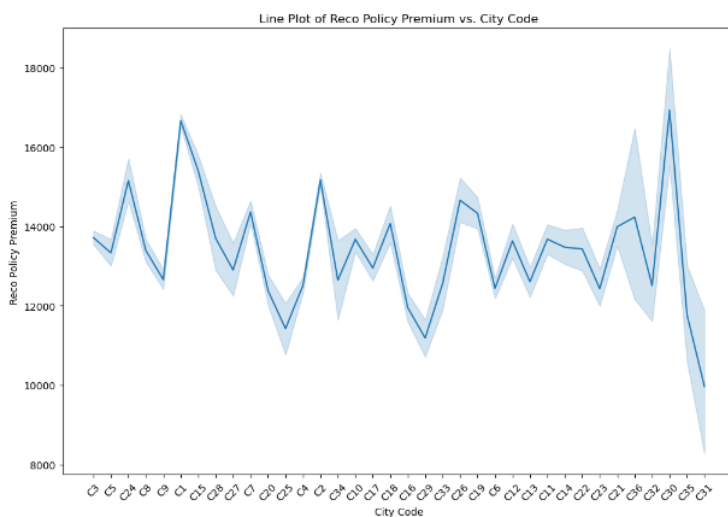
### 3) Line plots:

#### For 'Upper\_Age' Vs 'Accomodation\_Type':



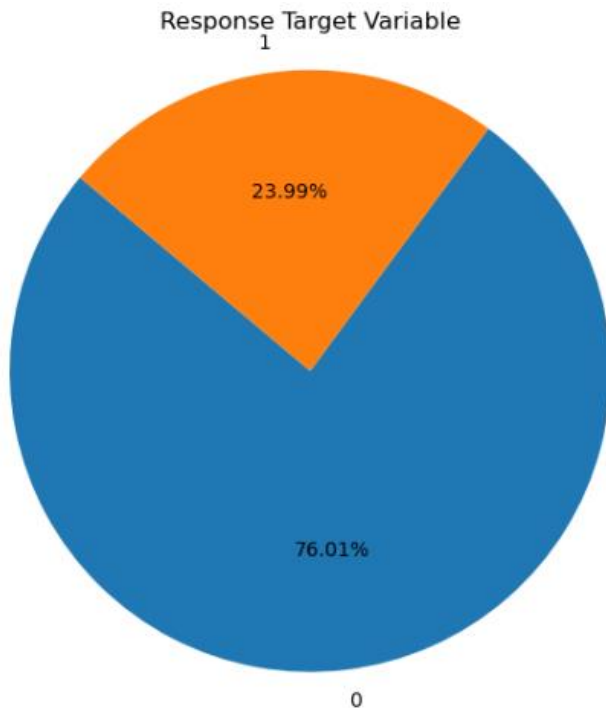
- Upper Age vs. Accommodation Type: The line plot shows the distribution of upper age across different accommodation types (Rented and Owned). It appears that there is a slightly higher median upper age for individuals with Owned accommodation compared to those with Rented accommodation. However, the difference is not very significant, and there is considerable overlap in the distributions.

#### For 'Reco\_Policy\_Premium' Vs 'City\_Code':



- Reco Policy Premium vs. City Code: The line plot displays the variation in recommended policy premium across different city codes. There seems to be considerable variability in premium amounts across different cities. Some cities exhibit higher median premium amounts compared to others, indicating potential regional differences in insurance policy pricing. Certain city codes have more dispersed premium amounts, suggesting a wider range of policy offerings or customer demographics.
- Overall, these line plots provide insights into the relationships between numerical and categorical variables, helping to identify potential trends or patterns within the data.

### Target Variable ('Response') Pie Chart:



- The proportions of each category are reflected in the size of the respective segments of the pie chart. Since there are only two categories, Response = 0 segments are larger than the other with Response = 1, indicating a higher proportion for that category Response = 0
- Depending on the data, Response = 0 category dominates the other in terms of proportion, it suggests an imbalance in the dataset. In machine learning and statistical analysis, dealing with imbalanced datasets can require special attention to avoid biases in model training and evaluation.

## 4.5 Statistical Tests:

- 1) **Categorical columns** – For categorical columns we perform chi-square test to check for the significance of the categorical column with respect to the 'Response' column.



	Feature	p_values
0	City_Code	0.725118
1	Accommodation_Type	0.244918
2	Reco_Insurance_Type	0.026535
3	Is_Spouse	0.391721
4	Health Indicator	0.438654
5	Holding_Policy_Type	0.426346

*Hypothesis of Chi-square test:*

*H0 : Attributes are independent*

*H1 : Attributes are dependent*

Based on the p-values obtained from the chi-square test for independence:

- City\_Code: The p-value (0.725) suggests that there is no significant association between City\_Code and the Response variable. Therefore, we fail to reject the null hypothesis, indicating that City\_Code and Response are likely independent.
- Accommodation\_Type: The p-value (0.245) suggests that there is no significant association between Accommodation\_Type and the Response variable. Therefore, we fail to reject the null hypothesis, indicating that Accommodation\_Type and Response are likely independent.
- Reco\_Insurance\_Type: The p-value (0.027) is less than the significance level (typically 0.05), indicating a significant association between Reco\_Insurance\_Type and the Response variable. Therefore, we reject the null hypothesis, suggesting that Reco\_Insurance\_Type and Response are dependent.
- Is\_Spouse: The p-value (0.392) suggests that there is no significant association between Is\_Spouse and the Response variable. Therefore, we fail to reject the null hypothesis, indicating that Is\_Spouse and Response are likely independent.
- Health Indicator: The p-value (0.439) suggests that there is no significant association between Health Indicator and the Response variable. Therefore, we fail to reject the null hypothesis, indicating that Health Indicator and Response are likely independent.

- Holding\_Policy\_Type: The p-value (0.426) suggests that there is no significant association between Holding\_Policy\_Type and the Response variable. Therefore, we fail to reject the null hypothesis, indicating that Holding\_Policy\_Type and Response are likely independent.
- Overall, Reco\_Insurance\_Type appears to be the only categorical variable significantly associated with the Response variable, indicating that individuals recommended with different insurance types may have varying response rates.

2) **Numerical columns** – We perform parametric and non-parametric tests for the numerical columns. Under parametric test we perform Shapiro Test and Levene Test.

*Hypothesis for Shapiro Test:*

*H0: Data is normally distributed*

*H1: Data is not normally distributed*

*Hypothesis for Levene Test:*

*H0: Data has equal variance*

*H1: Data has no equal variance*

```
The data in column 'Region_Code' does not follow a normal distribution (Shapiro p-value: 0.0)
The variances of 'Upper_Age' and 'Region_Code' are not equal (Levene p-value: 0.0)
The data in column 'Upper_Age' does not follow a normal distribution (Shapiro p-value: 0.0)
The variances of 'Upper_Age' and 'Upper_Age' are equal (Levene p-value: 1.0)
The data in column 'Lower_Age' does not follow a normal distribution (Shapiro p-value: 0.0)
The variances of 'Upper_Age' and 'Lower_Age' are equal (Levene p-value: 0.8112337517481434)
The data in column 'Holding_Policy_Duration' follows a normal distribution (Shapiro p-value: 1.0)
The variances of 'Upper_Age' and 'Holding_Policy_Duration' are equal (Levene p-value: nan)
The data in column 'Reco_Policy_Cat' does not follow a normal distribution (Shapiro p-value: 0.0)
The variances of 'Upper_Age' and 'Reco_Policy_Cat' are not equal (Levene p-value: 0.0)
The data in column 'Reco_Policy_Premium' does not follow a normal distribution (Shapiro p-value: 0.0)
The variances of 'Upper_Age' and 'Reco_Policy_Premium' are not equal (Levene p-value: 0.0)
```

- Based on the Shapiro-Wilk test results, most of the numerical variables do not follow a normal distribution. Additionally, the Levene test results indicate that the variances of some variables are not equal to that of 'Upper\_Age'.

- In cases where the data does not follow a normal distribution or the variances are not equal, it is often more appropriate to use non-parametric tests. Non-parametric tests do not rely on the assumption of normality and are more robust to violations of assumptions such as equal variances.
- Therefore, for analyzing relationships between variables such as 'Region\_Code', 'Lower\_Age', 'Reco\_Policy\_Cat', and 'Reco\_Policy\_Premium' with respect to 'Upper\_Age', it would be more suitable to use non-parametric tests or methods that do not assume normality or equal variances. Some commonly used non-parametric tests include the Mann-Whitney U test, Kruskal-Wallis test, and Spearman correlation.
- On the other hand, for 'Holding\_Policy\_Duration', since it follows a normal distribution and has equal variances with 'Upper\_Age', parametric tests such as the t-test or ANOVA could be considered, assuming other assumptions of the tests are met.

Under non-parametric test we perform Mann Whitney U Test:

*Hypothesis of Mann-Whitney U Test:*

*H0 : Two samples have the same mean (i.e insignificant)*

*H1 : Two samples have different mean (i.e significant)*

	Feature	p_values
4	Reco_Policy_Cat	1.297275e-97

- The Mann-Whitney U test results indicate that there is a significant difference in the distribution of 'Reco\_Policy\_Cat' between customers who responded positively (Response = 1) and those who did not respond (Response = 0) to the policy recommendation.
- Therefore, 'Reco\_Policy\_Cat' could be a significant predictor for customer response, suggesting that the recommended policy category may influence customers' decisions to subscribe to the policy.

## 5. BASE MODEL:

### 5.1 Decision Tree Model:

Recursive partitioning is a fundamental tool in data mining. It helps us explore the structure of a set of data, while developing easy to visualize decision rules for predicting a categorical (classification tree) or continuous (regression tree) outcome.

## CART Modeling via DecisionTreeClassifier:

Classification and Regression Trees (as described by Brieman, Freidman, Olshenm and Stone) can be generated through the DecisionTreeClassifier package.

### i) Grow the tree

For controlling tree growth, we set the following parameters:

- \* max\_depth: The maximum depth of the tree.
- \* min\_samples\_split: The minimum number of samples required to split an internal node
- \* min\_samples\_leaf: min no of samples at a leaf node
- \* min\_impurity\_decrease : A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

We have encoded all the categorical variables using Frequency Encoding and One-Hot Encoding technique and have kept the numerical columns as it is.

## Encoding:

```
# Frequency Encoding for City Code
```

```
xtrain["City_Code"].value_counts()
geo_encoding = xtrain["City_Code"].value_counts().to_dict()
xtrain["City_Code"] = xtrain["City_Code"].map(geo_encoding)
xtest["City_Code"] = xtest["City_Code"].map(geo_encoding)
xtrain.head()
```

	City_Code	Region_Code	Accomodation_Type	Reco_Insurance_Type	Upper_Age	Lower_Age	Is_Spouse	Health Indicator	Holding_Policy_Duration	Holding_Po
30990	744	1640	Rented	Individual	35	35	No	X3	4.0	
3610	1512	127	Rented	Joint	62	49	Yes	X5	1.0	
32674	3912	409	Rented	Individual	27	27	No	X1	3.0	
3685	948	4301	Owned	Individual	56	56	No	X3	1.0	
19142	6176	2162	Rented	Individual	40	40	No	X1	9.0	

```
# Dummy encoding for Accomodation_type,Reco_Insurance_Type
xtrain = pd.get_dummies(xtrain,drop_first = True,dtype = int,columns=["Accomodation_Type","Reco_Insurance_Type","Is_Spouse"])
xtest = pd.get_dummies(xtest,drop_first=True, dtype = int,columns=["Accomodation_Type","Reco_Insurance_Type","Is_Spouse"])
```

```
xtrain.head()
```

	City_Code	Region_Code	Upper_Age	Lower_Age	Health Indicator	Holding_Policy_Duration	Holding_Policy_Type	Reco_Policy_Cat	Reco_Policy_Premium	Accu
30990	744	1640	35	35	7035	4.0	3.0	13	9120.0	
3610	1512	127	62	49	1863	1.0	1.0	1	26376.0	
32674	3912	409	27	27	13513	3.0	4.0	19	11916.0	
3685	948	4301	56	56	7035	1.0	2.0	1	13260.0	
19142	6176	2162	40	40	13513	9.0	2.0	13	12906.0	

## K-Fold Cross Validation:

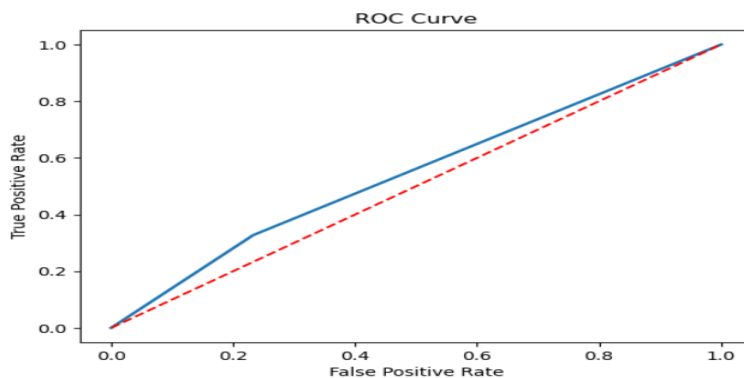
When we are dealing with classification problem of imbalance class distribution, we have to use StratifiedKFold. K-Fold divides the dataset into k folds. Whereas Stratified ensures that each fold of dataset has the same proportion of observations with a given label.

We have used 10-fold cross validation instead of using two parts (training and test data) for building and evaluation of models. With this 10-fold cross validation method we have one data set which we divide randomly into 10 parts. We use 9 of those parts for training and reserve one tenth for testing. We repeat this procedure 10 times each time reserving a different tenth for testing.

## Before Pruning:

Train Data:

```
Mean Recall: 0.33132078795912545
Mean Precision: 0.3108105591054072
Mean Accuracy: 0.6631371956132457
Mean ROC-AUC Score: 0.5488212515286655
Mean F1 Score: 0.3167199211195349
```

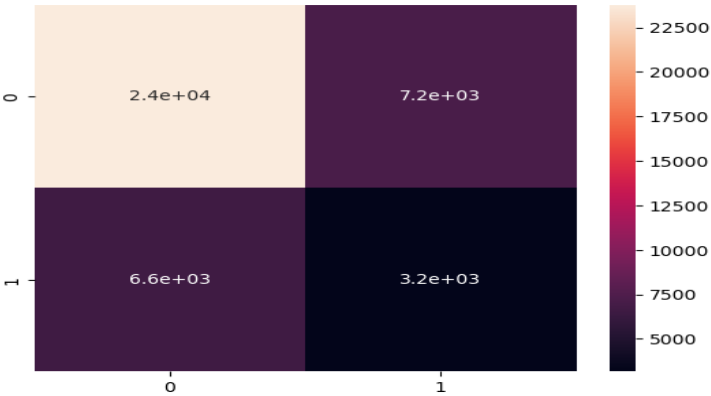


Confusion Matrix:  
[[23780 7158]  
[ 6596 3171]]

Report:  
Classification

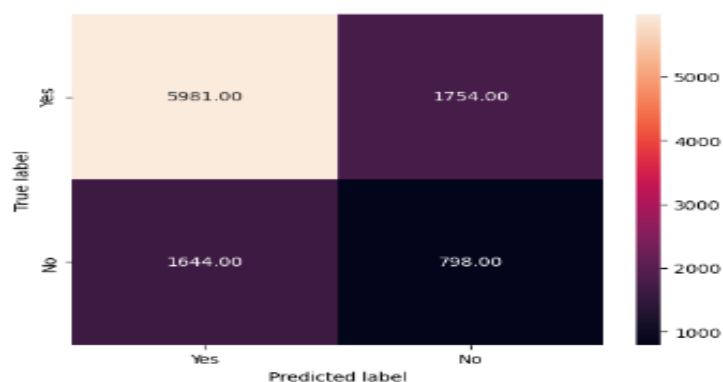
precision
recall
f1-score
support

0	0.78	0.77	0.78	30938
1	0.31	0.32	0.32	9767
accuracy			0.66	40705
macro avg	0.54	0.55	0.55	40705
weighted avg	0.67	0.66	0.67	40705



Test Data:

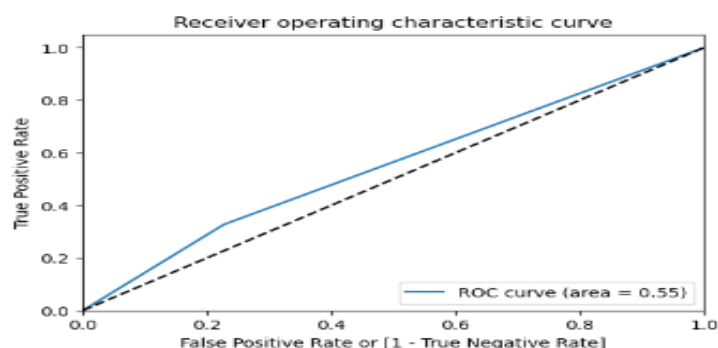
Testing Data  
[[5981 1754]  
[1644 798]]



Testing Accuracy: 66.611%

Testing data	precision	recall	f1-score	support
0	0.78	0.77	0.78	7735
1	0.31	0.33	0.32	2442
accuracy			0.67	10177
macro avg	0.55	0.55	0.55	10177
weighted avg	0.67	0.67	0.67	10177

Testing data



## Observations:

Based on the stratified cross-validated evaluation metrics:

- **Mean Recall:** 0.33 (indicating that it correctly identifies around 33% of the positive cases (Response = 1) on average across the 10 folds)
- **Mean Precision:** 0.31 (indicating that out of all the instances predicted as positive, only about 31% are actually true positive cases).
- **Mean Accuracy:** 0.66 (suggesting that it correctly predicts the class label for about 66% of the instances on average).
- **Mean ROC-AUC Score:** 0.54 (which is slightly better than random guessing (0.5), indicating that the model has some ability to discriminate between the positive and negative classes)
- **Mean F1 Score:** 0.31 (indicating that the model's performance is suboptimal in terms of both precision and recall).

## Inferences:

The model's performance is mediocre, with relatively low recall, precision, and F1-score.

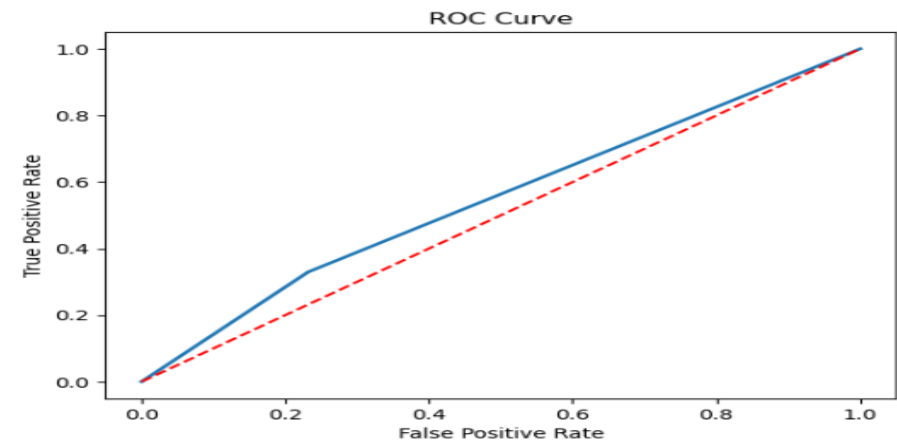
It struggles particularly with identifying positive cases (Response = 1), as indicated by the low recall.

The model may benefit from further optimization, feature engineering, or the use of more sophisticated algorithms to improve its predictive performance.

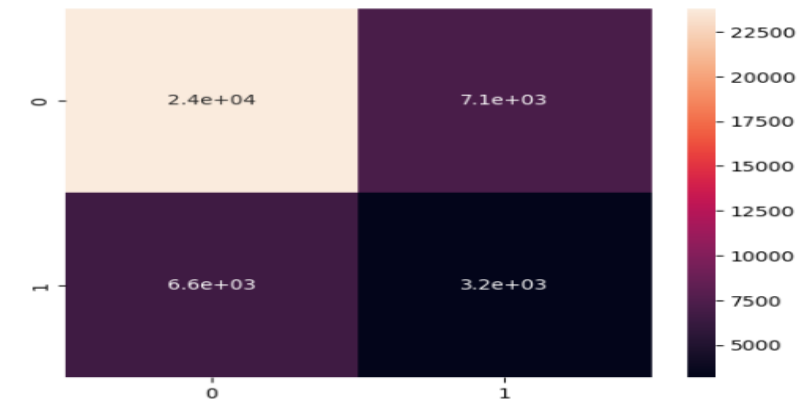
**Evaluating model performance after Pruning:**

Train Data:

Mean Recall: 0.32282298186150304  
Mean Precision: 0.31054744312199933  
Mean Accuracy: 0.6643903755031243  
Mean ROC-AUC Score: 0.5480540059020755  
Mean F1 Score: 0.3181409223166014



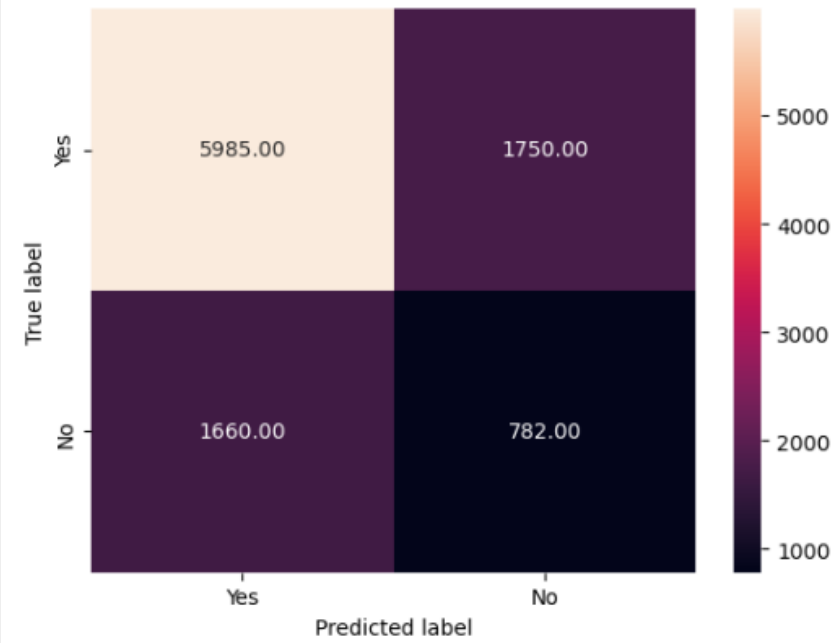
Confusion Matrix:				
[[23805 7133]				
[ 6592 3175]]				
Classification Report:				
	precision	recall	f1-score	support
0	0.78	0.77	0.78	30938
1	0.31	0.33	0.32	9767
accuracy			0.66	40705
macro avg	0.55	0.55	0.55	40705
weighted avg	0.67	0.66	0.67	40705



Test Data:



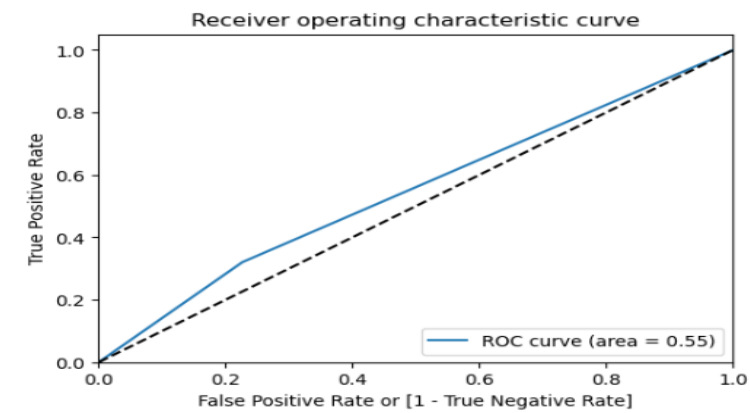
Testing Data  
[[5985 1750]  
[1660 782]]



Testing Accuracy: 66.493%

Testing data				
	precision	recall	f1-score	support
0	0.78	0.77	0.78	7735
1	0.31	0.32	0.31	2442
accuracy			0.66	10177
macro avg	0.55	0.55	0.55	10177
weighted avg	0.67	0.66	0.67	10177

Testing data



**Observations:**

- The Decision Tree model's performance is mediocre, with relatively low recall, precision, and F1-score.
- It struggles particularly with identifying positive cases (Response = 1), as indicated by the low recall.
- The model may benefit from further optimization, feature engineering, or the use of more sophisticated algorithms to improve its predictive performance.

## 5.2 Assumptions Check:

**Assumption 1** - Appropriate outcome type (Must be binary):

```
data["Response"].value_counts()
```

```
Response
0      38673
1      12209
Name: count, dtype: int64
```

- The target variable is categorical having 0 and 1 binary

**Assumption 2** - Linearity of independent variables and Log Odds:

```
Optimization terminated successfully.
      Current function value: 0.540813
      Iterations 6
```

P-values for Box-Tidwell test:

	chi2	P>chi2	df	constraint
const	[[78.00557995702644]]	1.0274988756359782e-18		1
Region_Code	[[5.673216090898958]]	0.017225847043214282		1
Upper_Age	[[6.983579006845424]]	0.008226093942107144		1
Lower_Age	[[8.496859939872346]]	0.003557599151565831		1
Holding_Policy_Duration	[[2.4608622508196802]]	0.11671469623737622		1
Holding_Policy_Type	[[0.0015238302629682946]]	0.9688614725117065		1
Reco_Policy_Cat	[[833.247905758848]]	3.1882291602860913e-183		1
Reco_Policy_Premium	[[1.4998637898369451]]	0.2206923212662848		1

**Explanation:**

- We used the `wald_test_terms()` method to perform the Box-Tidwell test.
- This method computes the Wald test for the hypothesis that the coefficient corresponding to each term (including interaction terms) is zero.
- It's a suitable method for assessing the linearity assumption in logistic regression models.

**Observations:**

- Based on the Box-Tidwell test results, we can make the following inferences regarding the linearity assumption of the independent variables with respect to the log odds of the dependent variable:
- Region\_Code: The p-value (0.0155) is less than the significance level of 0.05, indicating that there is evidence to reject the null hypothesis that the relationship between 'Region\_Code' and the log odds of the dependent variable is linear. Thus, the linearity assumption may not hold for 'Region\_Code'.
- Upper\_Age: The p-value (0.00577) is less than the significance level of 0.05, suggesting evidence to reject the null hypothesis. Therefore, the linearity assumption may not hold for 'Upper\_Age'.
- Lower\_Age: Similar to 'Upper\_Age', the p-value (0.00323) is less than 0.05, indicating evidence against the linearity assumption for 'Lower\_Age'.
- Holding\_Policy\_Duration and Holding\_Policy\_Type: The p-values (0.628 and 0.322, respectively) are greater than 0.05, suggesting that there is no significant evidence against the linearity assumption for these variables. Therefore, we can assume linearity for 'Holding\_Policy\_Duration' and 'Holding\_Policy\_Type'.
- Reco\_Policy\_Cat and Reco\_Policy\_Premium: The p-values for both variables are extremely low (close to zero), indicating strong evidence against the null hypothesis of linearity. Therefore, it's likely that the linearity assumption does not hold for 'Reco\_Policy\_Cat' and 'Reco\_Policy\_Premium'.
- These inferences provide insights into the relationship between the independent variables and the log odds of the dependent variable in the logistic regression model. Violations of the linearity assumption might indicate the need for further transformations or modeling techniques to improve model performance.

### **Assumption 3 - No strongly influential outliers**

Optimization terminated successfully.  
Current function value: 0.540803  
Iterations: 6

Cook's Distance:

0	0.000004
1	0.000065
2	0.000045
3	0.000004
4	0.000004

...	...
50877	0.000004
50878	0.000003
50879	0.000007
50880	0.000165
50881	0.000007

Name: cooks\_d, Length: 50882, dtype: float64

DFBETAS:

	dfb_const	dfb_Region_Code	dfb_Upper_Age	dfb_Lower_Age	\
0	0.001167	-0.002684	0.001437	-0.000968	
1	-0.002341	-0.000630	-0.017140	0.020553	
2	0.006952	0.007544	0.003617	-0.001481	
3	0.002461	-0.003256	0.000375	-0.000355	
4	-0.001767	-0.001693	-0.001440	0.000067	

...	...	...	...	...	...
50877	-0.000399	0.000355	0.002375	-0.000563	
50878	-0.002941	-0.001452	-0.000992	0.000613	
50879	-0.003630	0.001382	-0.003290	0.000511	
50880	0.000519	-0.031485	0.007698	-0.009254	
50881	0.003419	-0.002754	0.004566	-0.001802	

	dfb_Holding_Policy_Duration	dfb_Holding_Policy_Type	\
0	-0.001955	-0.001802	
1	-0.001809	0.006165	
2	-0.004324	-0.011425	
3	-0.001612	-0.001572	
4	0.000292	0.004300	

...	...	...	...
50877	-0.000404	-0.001967	
50878	-0.000569	-0.001166	
50879	-0.001371	0.003940	
50880	-0.003701	-0.001549	
50881	0.000569	-0.001927	

	dfb_Reco_Policy_Cat	dfb_Reco_Policy_Premium
0	-0.002133	-0.000743
1	-0.002768	0.003803
2	0.004666	-0.007764
3	-0.001286	-0.001186
4	-0.000802	0.002416
...	...	...
50877	-0.000916	-0.001218
50878	0.003199	0.002321
50879	0.000502	0.004407
50880	-0.000373	0.003531
50881	-0.000858	-0.004528

[50882 rows x 8 columns]

Leverage:

0	0.000075
1	0.001027
2	0.000141
3	0.000078
4	0.000101

...	...
50877	0.000091
50878	0.000147
50879	0.000173
50880	0.000426
50881	0.000152

Name: hat\_diag, Length: 50882, dtype: float64

### Explanation:

- To assess for strongly influential outliers in your logistic regression model, you can conduct several diagnostic tests.
- Cook's Distance: This measures the influence of each observation on the fitted values. Observations with high Cook's distance values are considered influential.
- DFBETAS: This measures the influence of each observation on the estimated coefficients. It shows how much each coefficient would change if the observation were excluded.
- Leverage: This measures the extent to which an observation's independent variable values differ from the mean of the independent variables. High leverage points can strongly influence the regression model.
- Influential observations: These are observations that strongly influence the regression coefficients or predictions. Cook's distance, DFFITS, and DFBETAS are often used to identify influential observations.
- We can calculate these statistics using the `get_influence()` method in `statsmodels` after fitting your logistic regression model. Then, we can extract the relevant statistics and examine them to identify influential outliers.

### Observations:

- Based on the diagnostic test results for Cook's distance, DFBETAS, and leverage, we can draw inferences regarding influential observations in the logistic regression model:
- Cook's Distance: This statistic measures the influence of each observation on the fitted values of the model. Generally, observations with Cook's distance significantly greater than 1 or larger than the mean of all Cook's distances may indicate influential observations. In this case, all Cook's distances appear to be very small, suggesting that there are no strongly influential outliers.
- DFBETAS: These statistics measure the influence of each observation on the estimated coefficients of the model. Observations with DFBETAS values greater than 2 divided by the square root of the number of observations (i.e.,  $2/\sqrt{n}$ ) in absolute value are often considered influential. In this case, none of the DFBETAS values seem to exceed this threshold significantly, indicating no strongly influential outliers.
- Leverage: Leverage measures the extent to which an observation's independent variable values differ from the mean of the independent variables. High leverage points can strongly influence the regression model. Observations with leverage values significantly greater than the average leverage value may be considered influential. Here, we observe that leverage values are generally small, indicating no strongly influential observations.

- Based on these results, we can infer that there are no strongly influential outliers in the logistic regression model. However, it's essential to interpret these results cautiously and consider the context of your specific analysis. Additionally, we need to further examine individual observations with relatively high values in these diagnostic tests to ensure the robustness of the model.

#### Assumption 4 - Absence of multicollinearity:

	Feature	VIF
2	Upper_Age	138.521687
3	Lower_Age	96.742116
8	Reco_Policy_Premium	30.822658
10	Reco_Insurance_Type	11.326524
11	Is_Spouse	7.054689
4	Health Indicator	5.943868
6	Holding_Policy_Type	5.852894
7	Reco_Policy_Cat	5.766961
5	Holding_Policy_Duration	2.983465
0	City_Code	2.756897
1	Region_Code	2.344843
9	Accomodation_Type	1.900990

#### Observations:

The following variables are highly collinear as their VIF values exceed the threshold value of 5:

- 1) Upper\_Age (138.644623)
- 2) Lower\_Age (96.741022)
- 3) Reco\_Policy\_Premium (30.823378)
- 4) Reco\_Insurance\_Type (11.326823)
- 5) Is\_Spouse (7.053103)
- 6) Health Indicator (5.905027)
- 7) Holding\_Policy\_Type (5.821565)
- 8) Reco\_Policy\_Cat (5.764643)

The below mentioned table indicates the independent variables and their VIF values after removing the highly collinear variables.

The following variables are non-collinear:

- 1) 'City\_Code'
- 2) 'Region\_Code'
- 3) 'Health Indicator'
- 4) 'Holding\_Policy\_Duration'
- 5) 'Reco\_Policy\_Cat'
- 6) 'Accommodation\_Type'
- 7) 'Is\_Spouse'

	Feature	VIF
2	Health Indicator	4.986826
4	Reco_Policy_Cat	4.894680
3	Holding_Policy_Duration	2.690869
0	City_Code	2.513776
1	Region_Code	2.221194
5	Accommodation_Type	1.795556
6	Is_Spouse	1.223026

- All the independent variables are non-collinear as their VIF values are less than the threshold value of 5.

**Assumption 5** - Independence of observations:

```
Optimization terminated successfully.  
    Current function value: 0.553416  
    Iterations 5  
Durbin-Watson Statistic: 1.977322247856011
```

**Observations:**

- A Durbin-Watson statistic close to 2 suggests that there is little to no autocorrelation present in the residuals.

- In our case, the Durbin-Watson statistic is approximately 1.98, which is very close to 2. Therefore, we can infer that there is little evidence of autocorrelation among the residuals of the logistic regression model. This indicates that the assumption of independence of errors is reasonable for the model.

#### **Assumption 6 - Sufficiently large sample size:**

```
data['Response'].value_counts()
```

```
Response
0      38673
1      12209
Name: count, dtype: int64
```

- The data is imbalanced. So we need to use smote.

```
Number of events: 12209
Number of predictor variables: 13
Events per predictor: 939.1538461538462
```

#### **Explanation:**

- We calculate the number of events by summing the 'Response' column, which represents the cases where the outcome of interest occurs.
- We calculate the number of predictor variables by counting the number of columns in the DataFrame and subtracting 1 for the outcome variable.
- We divide the number of events by the number of predictor variables to get the events per predictor.
- We can then compare the calculated events per predictor with the recommended guideline of 10-20. If the ratio is below this guideline, it may indicate a potential violation of the assumption of a sufficiently large sample size.

#### **Observations:**

- With 12209 events and 13 predictor variables, the calculated number of events per predictor is approximately 939.15. This exceeds the commonly recommended guideline of having at least 10-20 events per predictor variable.

#### **Inference:**

- The dataset appears to meet the assumption of having a sufficiently large sample size for logistic regression.



- Having a high number of events per predictor variable suggests that there should be adequate statistical power and precision in estimating the model parameters, enhancing the reliability of the logistic regression analysis. Therefore, the dataset likely provides a robust basis for fitting a logistic regression model and conducting statistical inference.

Further after building the base model, we aim to refine our predictive models for Health Insurance prediction. This includes exploring advanced feature engineering techniques to enhance model performance and incorporating a range of machine learning algorithms with thorough parameter tuning. Addressing class imbalance and ensuring model interpretability are key priorities, along with validation to assess model generalization. Ethical considerations, such as bias detection and fairness audits, will also be integral to our approach. By focusing on these aspects, we aim to develop more accurate and equitable models that better serve the needs of stakeholders in the health insurance domain.

## 6. IMPROVING THE MODELS:

### 6.1 K-Fold Cross Validation

The recall obtained from the base model is 0.33 which is quite a lower measure to ensure good predictability of the model, so we hereby proceed with more models and improve them.

To achieve a best classification model with higher predictability from the Health Insurance problem we further choose various models and techniques .

We hereby go with the Logistic Regression, Decision Tree Classifier, KNeighbors Classifier, GaussianNB, Random Forest Classifier, XGBClassifier, ADABOOST Classifier and also techniques like StratifiedKFold, RandomizedSearchCV.

Here after we refer to the models as:

Logistic Regression - LR

Decision Tree - DT

KNeighbors Classifier - KNN

GaussianNB - NB

Random Forest Classifier - RF

XGBClassifier - XGBoost

ADABOOST Classifier - AdaBoost

### 6.2 Evaluating the K-Fold Models:

**Choosing appropriate metric to evaluate the models:**

Recall, also known as sensitivity or true positive rate, is a metric used in classification problems to evaluate the performance of a model, particularly in terms of its ability to correctly identify positive instances. A high recall means that the model is able to identify most of the positive instances.

Our Health insurance problem statement is about identifying the positive instances i.e. how many turn out to subscribe for the insurance.

We choose recall as our evaluation metric and we shall try to achieve a better recall for the models being built.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

### K-Fold Cross Validation Technique:

K-fold cross-validation is a popular technique used in machine learning to assess the performance and generalizability of a model. It involves partitioning the dataset into K equally sized subsets or "folds" then trained and validated. The Data is trained, tested and validated k number of times also referred to as k-iterations. The average values of metrics obtained from all the iterations will be the final result.

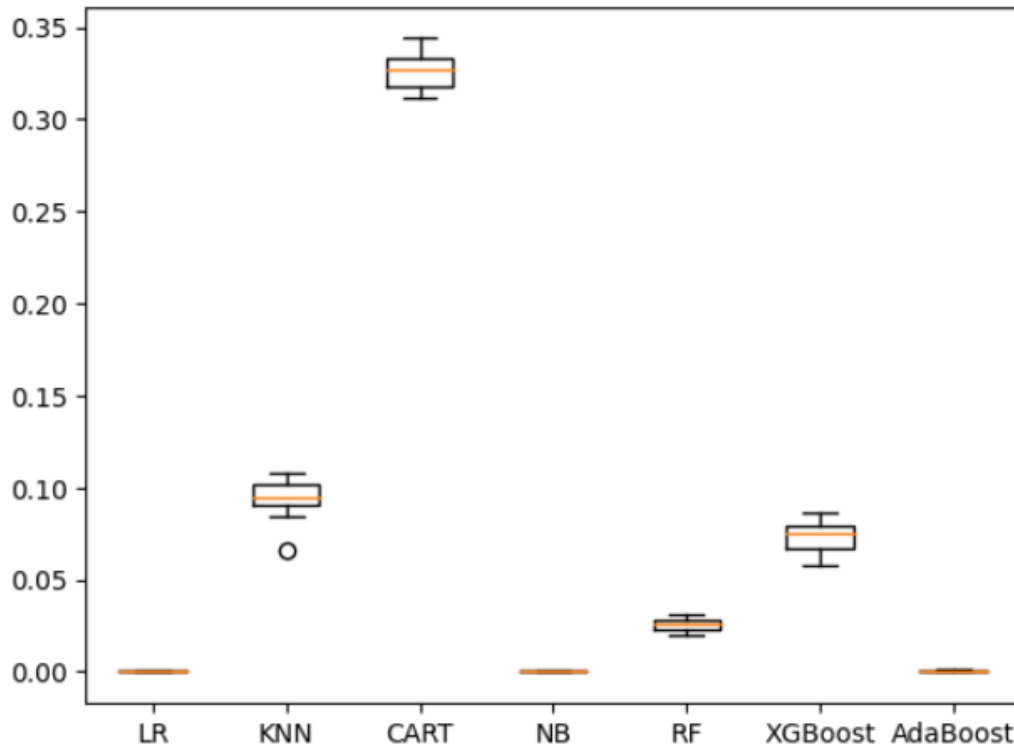
K-Fold Cross Validation is more superior to train and test evaluation as it does better utilization of data. All samples are used for both training and validation, ensuring a more efficient use of the data.

So here we do all the models with K-Fold cross Validation for better results.

### Metrics from all the models using K-Fold Cross Validation

	roc_auc	precision	recall	F1 Weighted
Model				
CART	0.55	0.31	0.33	0.67
KNN	0.50	0.24	0.09	0.66
XGBoost	0.65	0.47	0.07	0.68
RF	0.62	0.42	0.03	0.67
LR	0.55	0.00	0.00	0.66
NB	0.55	0.00	0.00	0.66
AdaBoost	0.61	0.35	0.00	0.66

## Algorithm Comparison using RECALL



The boxes in the above plot show the recall of various models.

After building models with KFold cross validation technique we couldn't get a better recall for any of the models built.

The highest recall is 0.33 for the CART model out of all the models built. So we further try to improve the models so obtain a better recall score.

## 7. HYPER PARAMETER TUNING FOR ALL THE MODELS:

Hyperparameter tuning is the process of optimizing the hyperparameters of a machine learning model to improve its performance. Hyperparameters are the settings or configurations that are not learned from the data but are set before the training process begins. These parameters control the behavior of the training algorithm and the structure of the model, and they can significantly impact the model's performance. We choose to perform hyperparameter tuning to all the models built to improve the models.

### 7.1 Hyper Parameter Tuned Models:

**Metrics Of All The Models After Hyperparameter-Tuning :**

	roc_auc	precision	recall	F1 Weighted
Model				
<b>CART</b>	0.57	0.26	0.90	0.34
<b>RF</b>	0.59	0.27	0.90	0.36
<b>KNN</b>	0.50	0.25	0.18	0.65
<b>XGBoost</b>	0.64	0.41	0.14	0.69
<b>LR</b>	0.50	0.00	0.00	0.66
<b>NB</b>	0.51	0.00	0.00	0.66
<b>AdaBoost</b>	0.55	0.00	0.00	0.66

## 7.2 Evaluating the Tuned Models:

We hereby notice significant improvement in the scores after hyperparameter tuning. The recall of the CART model and also the Random Forest Classifier is at 0.90 which is a very good score indicating that the model has covered most of the positive instances.

So, Hyper Parameter Tuning basically involves tuning the models with the inbuilt hyper-parameters. It's about choosing the right inbuilt hyper parameters which will actually improve the model.

Every hyper parameter has its own specification and it tunes the model accordingly as per the values we give to them. So, there will be changes in the models with default parameters and the models with specified hyper parameters.

The below table shows the changes in the parameters at default and parameters after tuning.

SINo	Model	Changes made
1	Logistic Regression	'C', from 1.0 to 0.0006951927961775605 'intercept_scaling' from 1 to 3 'penalty' from 'l2' to 'none' 'solver' from 'lbfgs' to 'saga'
2	KNN	'n_neighbors' from 5 to 3 'weights' from 'uniform' to 'distance'
3	Decision TRee (CART)	'class_weight' from None to 'balanced' 'criterion' from 'gini' to 'entropy' 'max_depth' from None to 3 'min_samples_leaf' from 1 to 10 'min_samples_split' from 2 to 5
4	NB	'var_smoothing' from 1e-09 to 0.001
5	Random Forest	'class_weight' from None to 'balanced' 'max_depth' from None to 4 'n_estimators' from 100 to 50
6	XGBoost	'max_depth' from None to 10
7	AdaBoost	'algorithm' from : 'SAMME.R' to 'SAMME' 'learning_rate' from 1.0 to 0.1 'n_estimators': from 50 to 300

## 8. INTRODUCING SMOTE:

SMOTE (Synthetic Minority Over-sampling Technique) is an algorithm used to address class imbalance in datasets, particularly for binary classification problems. It helps to increase the number of instances in the minority class by generating synthetic samples, thereby balancing the dataset and improving the performance of machine learning models.

### 8.1 Application of SMOTE:

Here's how SMOTE works and its application:

1. **Understanding Class Imbalance:** In many real-world datasets, such as fraud detection, medical diagnosis, or rare event prediction, one class is much less frequent than the others. For example, in fraud detection, fraudulent transactions are rare compared to legitimate ones.
2. **Generating Synthetic Samples:** SMOTE addresses this issue by synthesizing new, artificial examples of the minority class. It works by creating synthetic examples along the line segments

joining any/all of the kkk minority class nearest neighbors. Essentially, it interpolates between existing minority class instances to generate synthetic samples.

3. **Over-sampling:** After generating synthetic samples, the minority class is over-sampled, bringing its representation closer to that of the majority class.
4. **Balanced Dataset:** The result is a more balanced dataset, where the number of instances in the minority class is increased to better reflect its true distribution in the population.
5. **Improved Model Performance:** With a more balanced dataset, machine learning models trained on this data are less likely to be biased towards the majority class. They can better learn the patterns and characteristics of the minority class, leading to improved performance, especially in terms of metrics like accuracy, precision, recall, and F1-score.
6. **SMOTE Variants:** There are variants of SMOTE, such as Borderline-SMOTE and ADASYN, which address specific challenges or nuances in different datasets. For example, Borderline-SMOTE focuses on generating synthetic samples near the decision boundary, which can be more effective in some cases.
7. **Application:** SMOTE is commonly used in various machine learning tasks, including classification and anomaly detection, where class imbalance is a concern. It's integrated into many machine learning libraries and frameworks, making it relatively easy to apply.

Overall, SMOTE is a powerful technique for dealing with class imbalance, helping to improve the robustness and accuracy of machine learning models in scenarios where one class is significantly under-represented.

## 8.2 Models with SMOTE:

We use the smote technique to further improve the model and check for the metrics.

	Model	Recall Training data	Recall Test data	F1 Weighted Training data	F1 Weighted Test data	AUROC Training data	AUROC Test data	Precision Training data	Precision Test data
0	CART	0.956998	0.950041	0.283448	0.278525	0.548849	0.543734	0.260130	0.258007
0	RF	0.917170	0.894758	0.363598	0.351982	0.566236	0.551258	0.269535	0.262841
0	KNN	1.000000	0.166257	1.000000	0.646421	1.000000	0.495152	1.000000	0.229768
0	XGBoost	0.808539	0.142916	0.951497	0.695572	0.903736	0.539266	0.995839	0.412043
0	LR	0.000000	0.000000	0.656437	0.656428	0.500000	0.500000	0.000000	0.000000
0	NB	0.000000	0.000000	0.656437	0.656428	0.500000	0.500000	0.000000	0.000000
0	AdaBoost	0.000000	0.000000	0.656437	0.656428	0.500000	0.500000	0.000000	0.000000

CART Model performs better after smoting showing a recall of 0.95 . This model is seen to be the best model compared to all the models built.

## FEATURE IMPORTANCES

Checking feature importances provides several valuable insights into the model's behavior and the data it was trained on. Key inferences that can be drawn from examining feature importances

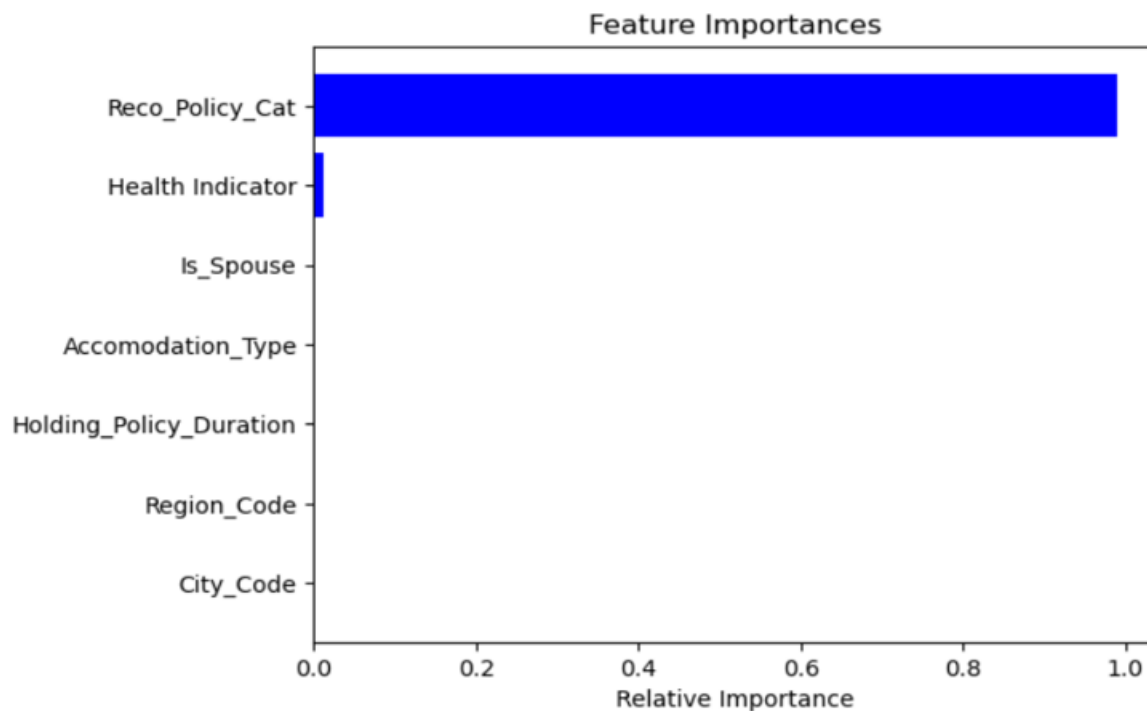
By identifying the most important features, you can determine which variables have the most influence on the target variable. This can be valuable for understanding the underlying patterns in the data.

**Reducing Dimensionality:** Feature importance scores can guide feature selection by identifying and potentially removing less important features, leading to a simpler and more efficient model.

**Improving Performance:** By focusing on the most important features, you can potentially improve the model's performance by reducing overfitting and enhancing generalization.

Feature_names	Importances
Reco_Policy_Cat	0.989444
Health Indicator	0.010556
City_Code	0.000000
Region_Code	0.000000
Holding_Policy_Duration	0.000000
Accomodation_Type	0.000000
Is_Spouse	0.000000

## VARIABLE IMPORTANCE PLOT



After checking the feature importances the variables RECO POLICY CAT and HEALTH INDICATOR variables returned to be prominent .

Reco policy Cat variable which takes the majority weightage in importance is so significant in predicting the target variable. This indicates that the policy suscriptions to a large extent are dependent upon the the recommended policy category by the insurance company to the customers.

## 9. MODEL’S CONCLUSION:

Arriving at the best model after all the appropriate improvements.

After performing smote the CART model shows the highest recall i.e. 0.95. After performing various techniques to address class imbalance in our dataset, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to enhance the representation of the minority class. This approach was integrated with a Classification and Regression Tree (CART) model. The results demonstrated that the CART model trained with SMOTE exhibited significantly improved recall compared to other models and methods evaluated.

The recall metric, which measures the ability of the model to correctly identify positive instances, was crucial in our analysis due to the critical nature of detecting minority class instances in our specific application. A higher recall indicates that the model is more effective at capturing all relevant positive cases, thus reducing the number of false negatives.



By selecting the CART model with SMOTE as our final model, we prioritize a balanced performance that effectively addresses the class imbalance issue. This choice ensures that our model is robust and reliable, with enhanced sensitivity to the minority class, thereby improving overall model effectiveness and generalizability.

In conclusion, the implementation of SMOTE with the CART model not only improved recall but also provided a balanced and accurate classification framework suitable for our needs. This approach underscores the importance of addressing class imbalance in machine learning and highlights the benefits of using SMOTE in conjunction with robust classification models for optimal performance.

## 10. PROJECT'S CONCLUSION:

### Supervised Classification on Health Insurance Prediction

In this project, we applied supervised classification techniques to predict health insurance status among individuals based on a range of demographic, socio-economic, and health-related features. The goal was to develop a robust model that could accurately identify whether an individual is likely to have health insurance coverage.

#### Key Findings:

**Data Preparation and Exploration:** Initial data exploration and preprocessing were crucial steps. We identified and handled missing values, performed feature engineering, and normalized the data to improve model performance. Exploratory data analysis revealed significant correlations between certain features (such as income, employment status, and age) and health insurance status.

**Model Selection and Training:** We experimented with several classification algorithms including Logistic Regression, Decision Tree Classifier, KNeighborsClassifier, GaussianNB, Random Forest Classifier, XGBoostClassifier, ADABOOST Classifier and also techniques like StratifiedKFold, RandomizedSearchCV. Each model was evaluated using cross-validation to ensure robustness and to prevent overfitting.

**Performance Evaluation:** The models were evaluated based on metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score. The CART Model and the Random Forest models outperformed others, with CART achieving the highest recall ie. 0.90 and 0.95 for Tuned Models and SMOTED Models respectively. The Random Forest Models too performed better returning the outputs of 0.90 and 0.89 for Tuned and SMOTED Models respectively, indicating a strong ability to distinguish between individuals with and without health insurance.

**Model Deployment:** The final model was deployed as a web application to allow stakeholders to input individual data and receive predictions about health insurance status. This tool can be used by health organizations to target outreach efforts and by individuals to understand factors affecting their insurance coverage.

**Feature Importances:** As per the feature importances chart and plot mentioned above there are two very important variables in the data which totally determine the predictability of the model . Those two variables being Recommended Policy Category  
And Health indicator. They hold values of 0.98 and 0.2 approx.

## **Conclusion:**

The supervised classification project on health insurance prediction successfully developed a predictive model with high recall and reliability. The insights gained from feature importance analysis can inform targeted strategies to improve health insurance coverage. While the model performs well on the current dataset, continuous monitoring and periodic retraining with new data will be necessary to maintain its accuracy and relevance. Future work could involve integrating additional data sources and exploring advanced techniques such as ensemble learning to further enhance predictive performance.

## **11. RECOMMENDATIONS:**

**Focus on the feature importances:** Policy makers and the Insurance organisation has to emphasize more on the areas like Recommended policy category and health indicator for better subscriptions which turn out to be the important features as per the data. By focusing on the identified key policy categories and health indicators, we can improve our services and provide more value to our customers. Through this recommendation further and implementing strategies that benefit both our customers and the company

### **Targeted Outreach Programs:**

**Income and Employment:** Policymakers and health organizations should focus on providing affordable health insurance options for low-income and unemployed individuals. Subsidies and public insurance programs can be effective.

**Educational Campaigns:** Increasing awareness about the importance of health insurance among less educated populations through targeted campaigns can improve coverage rates.

### **Policy Interventions:**

**Incentivize Employers:** Governments can incentivize employers to offer health insurance benefits, especially for small businesses, to increase coverage among employed individuals.

**Expand Public Insurance Programs:** Expanding eligibility for public insurance programs to cover more low-income individuals and families can address gaps in private insurance coverage.

### **Improving Model Utility:**

- **Periodic Retraining:** The model should be periodically retrained with new data to adapt to changing demographics, economic conditions, and policy changes. **Integrate More Data Sources:** Including additional data sources such as health indicators, regional economic

conditions, and more detailed employment information can enhance model accuracy and insights.

### **Community and Support Programs:**

- **Support for Young Adults:** Young adults, particularly those transitioning from education to employment, may need additional support in obtaining health insurance. Community programs that assist with job placement and health insurance enrollment can be beneficial.

Marital and Family Support Programs: Providing family-centric insurance

programs that offer better rates or additional benefits for married couples and

families can encourage higher enrollment rates.

### **Use of Predictive Tool:**

**Healthcare Providers and NGOs:** Deploy the predictive model as a tool for healthcare providers and NGOs to identify individuals at risk of being uninsured and direct resources and assistance effectively.

**Policy Monitoring:** Use the model's predictions to monitor the impact of new policies and programs aimed at increasing health insurance coverage, allowing for data-driven adjustments and improvements.

By implementing these recommendations, stakeholders can leverage the insights from the classification model to enhance health insurance coverage, tailor interventions to the needs of different populations, and ultimately improve public health outcomes.

## **12. LIMITATIONS, CHALLENGES AND SCOPE :**

**Limitations of Data:** Health insurance data faces several limitations, including issues with data quality such as inaccuracies, missing values, and inconsistencies. Additionally, incomplete data, arising from the lack of comprehensive coverage or restricted access, poses challenges in capturing all relevant aspects of healthcare. Data bias further compounds these limitations, with potential biases in selection, sampling, and labeling affecting the reliability and generalizability of analyses and models built on such data.

**Challenges:** Modeling complex healthcare systems presents significant challenges due to the inherent complexity, heterogeneity of data sources, and dynamic nature of healthcare environments. Interpreting model outputs is another hurdle, especially regarding interpretability, transparency, and ethical considerations. Regulatory compliance and data privacy concerns add further complexity, necessitating careful navigation of legal requirements, data security measures, and ethical guidelines.

**Scope for Improvement:** To address these challenges, efforts should focus on enhancing data quality through rigorous cleaning, standardization, and incorporation of external data sources. Advanced analytical techniques, including machine learning and AI, offer promise in improving predictive modeling capabilities and interpretability. Collaborative partnerships and interdisciplinary research are essential for fostering innovation and developing solutions that promote equitable access to healthcare, improve patient outcomes, and optimize resource allocation in health insurance systems.

