Health Insurance Lead Prediction

Approach

Steps:

1) I choose **google colab** platform for the project. Importing all the relevant libraries. Load the training dataset. Look for the shape and info of the dataset. It has total 50882 rows with 14 columns. Our target column is **Response**(0 or 1) which makes it a **classification** problem.

2) Checking for the null values and found 3 categorical columns having nan values. Getting the sum of unique values in each column. Getting the datatype for each column.

3) Separating the variables in continuous and categorical data. We have maximum variables as categorical. Separating the categorical columns into nominal and ordinal.

4) Plotting frequency distribution and count plots for continuous and categorical variables respectively. Plotting boxplots with respect to Target column. Boxplots allows us to check for outliers. There are not many outliers in this dataset.

5) Taking out correlation among all the variables. Which gives the ID, City_Code, Region_code are least correlated variables to the target variable. Hence dropping these 3 columns from the main dataset.

6) Feature Engineering / Data preprocessing:-

- *Nan values*- We had 3 columns with Nan values. Since the Nan rows are large in number, so cannot drop the rows. So applying **frequency encoder** : it replaces the Nan values with mode. This method is used for categorical variables.
- *Ordinal categorical columns*- We had 4 ordinal categorical columns. I encoded them with **Label Encoder** : In Label encoding, each label is converted into an integer value.
- *Nominal categorical columns :* We had 5 nominal categorical columns. I encoded them with **Dummy encoding**: This categorical data encoding method transforms the categorical variable into a set of binary variables (also known as dummy variables).

7) Splitting my training dataset into x_train, x_test, y_train, y_test in the ratio of 70:30. Dropping the target column from X and putting it in Y.

8) Now my dataset is ready for Model training. I am applying various models one by one and checking which model gives me highest Evaluation metric which was **roc_auc_score** as stated in the problem statement.

9) Models used for Training**:**

Model	**Accuracy**	**Roc_Auc_Score**	**AUC(Area)**
Logistic Regression	0.757	0.5	0.567
Gaussian Naïve Bayes	0.757	0.5	0.556
K-Nearest Neighbors	0.712	0.514	0.545
Random Forest Classifier	0.707	0.522	0.561
Random Forest Regressor		0.584	0.584
Decision Tree Classifier	0.749	0.507	0.631
Decision Tree Regressor		0.6315	0.6315

10) Outcome: Decision Tree Regressor gave me the maximum value of roc_auc_score. Hence this is my final model for predicting values on the testing dataset given.

11) Now importing the testing Dataset. And applying all the steps that were done on training dataset to this. This dataset too has 3 columns containing Nan values. Now training the model on whole 50882 rows of Training dataset .

12) Applying Decision Tree Regressor (max_depth = 10). I got this value of max_depth by applying a loop on the DTR model with max_depth in range of (5-15). And I got maximum roc_auc_score for max_depth = 10 .

13)Finally creating a csv file for the predicted values.

Name: Ansheeta Singh