Health Insurance Lead Prediction

LAL BABU SAH

FEB 28, 2021

Table of contents

Introduction: Business Problem

A financial services company that provides various financial services like loan, investment funds, insurance etc. to its customers wishes to cross-sell health insurance to the existing customers who may or may not hold insurance policies with the company.

The company recommend health insurance to it's customers based on their profile once these customers land on the website. Customers might browse the recommended health insurance policy and consequently fill up a form to apply. When these customers fill-up the form, their Response towards the policy is considered positive and they are classified as a lead.

Once these leads are acquired, the sales advisors approach them to convert and thus the company can sell proposed health insurance to these leads in a more efficient manner.

In this project, I have to build a model to predict whether the person will be interested in their proposed Health plan/policy.

Used classification machine learning algorithm ‘Random Forest Classifier’ for the prediction of dependent variable (‘Response’).

Data Acquisition

Train and Test dataset given in the csv format, read the train and test dataset using pandas.

Data Wrangling

```
data_analysis(df_train,  df_test)
```

```
train_shape: (50882, 14)
test_shape: (21805, 13)
```

	train_dtype	test_dtype	train_isnull	test_isnull	train_unique	test_unique
Holding_Policy_Duration	object	object	20251	8603	16	16
Holding_Policy_Type	float64	float64	20251	8603	5	5
Health Indicator	object	object	11691	5027	10	10
ID	int64	int64	0	0	50882	21805
City_Code	object	object	0	0	36	36
Region_Code	int64	int64	0	0	5316	4694
Accomodation_Type	object	object	0	0	2	2
Reco_Insurance_Type	object	object	0	0	2	2
Upper_Age	int64	int64	0	0	58	58
Lower_Age	int64	int64	0	0	60	60
Is_Spouse	object	object	0	0	2	2
Reco_Policy_Cat	int64	int64	0	0	22	22
Reco_Policy_Premium	float64	float64	0	0	6977	5226

Feature: 'Accomodation_Type', 'Reco_Insurance_Type' and 'Is_Spouse' have datatype ‘object’ and containing two unique categories, converted these categorical columns in numeric type(0, 1).

Feature: 'City_Code' have datatype ‘object’ and containing 36 unique categories, converted these categorical columns in numeric type using ‘label encoder’.

Data Exploration

Dealing with missing values:

Feature: 'Health Indicator' have datatype ‘object’ and containing approx. 20% missing data and have 9 unique categories, we can’t drop this feature, this can be a important feature, assigned the missing value with new variable and converted these columns in numeric type using ‘label encoder’.

Feature: ‘Holding_Policy_Type’ have datatype ‘float’ and containing approx. 40% missing data and have 4 unique type, we can’t drop this feature, this is an important feature. So, predicted the missing value using ‘XGBClassifier’.

Feature: ‘Holding_Policy_Duration’ have datatype ‘object’ and containing approx. 40% missing data and have 15 unique categories, we can’t drop this feature, this is an important feature. So, predicted the missing data using ‘XGBClassifier’.

Correlation and p-value:

Calculated the Pearson Correlation Coefficient and the p-value of different columns and dropped the columns which are not important for prediction on the basis of correlation and p-value.

Dealing with outliers:

Feature: 'Reco_Policy_Premium' is continues data, visualised the data using ‘distplot’ and ‘boxplot’.

Model Development

After data pre-processing and one_hot_encoding of categorical features, normalized the train and test dataset, and split the train dataset into train and validation dataset using ‘train_test_split’ for model development and model evaluation.

Used classification machine learning algorithm ‘Random Forest Classifier’ for the prediction of dependent variable (‘Response’).

For hyper-parameter tuning used ‘RandomizedSearchCV’ and ‘StratifiedKFold’ technique.

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.