Health Insurance Lead Prediction

Data Understanding:

- The shape of the data was (50882, 14), with lots of categorical variables and few numerical.
- Data is divided into two parts one describing customer's demographics and the second one describing its business with a company.

EDA:

Important findings:

- None of the continuous variables were helping to distinguish between positive and negative classes.
- There is a **Target Imbalance**, only **23% positive** responses.
- the number of **positive responses** from **C1 and C2 city is high**, and the conversion rate is also **near 25%.** quite decent if we compare it with other cities.
- the number of **positive responses** from the **X1 and X2 categories is high**, and the conversion rate is also near **25%**. quite decent if we compare it with other categories.
- more **positive responses** from a customer **holding another policy for +14, 1, 2,3 years**. conversion rate is also good for these categories, **above 20%.**
- Recommended **policy no. 22** have **high positive responses**. conversion rate is also quite high for policy no. 22, above 30%. policy no.15 has a **conversion rate of about 45%.**

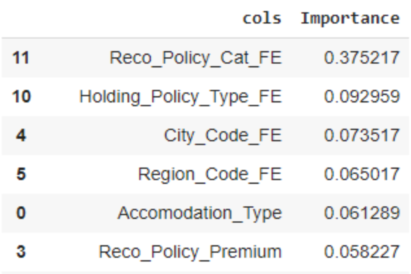
Other Interesting Observations:

- on **average premium amount** of **c1 and c30** city are **high,** and on **average Age** of customers **of C1 and C30 city is above 50.** maybe at a higher age health risk is high, and that's why high premium
- recommended avg. the **premium amount** for **joint customers** is very **high** compare to an individual one**.**
- Customers who **owned accommodation**, get on an average **high recommendation premium amount**.
- Customers with **X7 health indicators** have on average **high recommended premium amount**. also, the **average age** of customers with the **X7 indicator is above 50 years.**
- customers with **14+ years** of **holding other policy** got an on avg. **high amount premium amount** recommendation. customers **holding policy type 4** get high on an avg. **high amount recommendation. type 5** recommended policy looks **expensive**.

(More on EDA and Observations available in the EDA Notebook)

Approach:

1. At the very start after doing basic pre-processing, I directly use XGBoost, to find out feature importance. So that I can Focus on these Features for Feature Engineering. Found below Features.



	cols	Importance
11	Reco_Policy_Cat_FE	0.375217
10	Holding_Policy_Type_FE	0.092959
4	City_Code_FE	0.073517
5	Region_Code_FE	0.065017
0	Accomodation_Type	0.061289
3	Reco_Policy_Premium	0.058227

Pre-processing:

- Found no duplicate values.
- **Missing values:**
  - when there are missing values at Holding_Policy_Duration, Holding_Policy_Type also has missing values. and their percentage values missing matches.
  - From the data dictionary, we can see that Holding_Policy_Duration and Holding_Policy_Type indicate to already existing customers. what if there is someone new who didn't hold any policy. we can conclude that these null values indicate someone new. for Holding_Policy_Duration we can put 0 years and for Holding_Policy_Type we can put -1 to indicate that person does not hold any.
  - Creating an unknown category for health indicator missing values.
  -
- **Feature Engineering:**
  - I used feature tools for creating features automatically.
  - In Manual feature engineering, created some features like new customers, average age, age difference. And run a baseline Xgboost model where I got a score of 64%.
  - Using important features found earlier created some features of Ratios, mean, standard deviation transformations. Tried different ratios. Created some log, square transformations features. And Used CatBoost classifier, as more data was in categorical form. Got a score of 76%. A good Improvement from the last score.
  - Then Join some categorical features together to see how they behave for Age, Premium Amount. They behaved differently for different regions, cities, and other categories. Using Catboost got a score of 0.80. A good Improvement from the last score.
  - Then created some more features by combining 3 categories, got a slight improvement in score.

- **Encoding and Scaling:**
  - Tried Label and frequency encoding, frequency encoding worked better. After splitting the data scaled it using MinMax scaler.

Model Building:

- Tried Two models, Xgboost and CatBoost.
- For Tuning Xgboost Model I used the Hyperopt package, for bayesian Optimization. Here I used various hyperparameters to tune the model and try to avoid, overfitting. Here I got a 0.77 score on Val and 0.76 on a public leaderboard.
- For tuning the Catboost Model, I used Skopt package, for bayesian Optimization. Given a list of categorical features, it worked well on this data. I got a 0.81 score on Val and 0.80 on a public leaderboard.