JOBATHON JANUARY 2023

Problem Statement

VahanBima is one of the leading insurance companies in India. It provides motor vehicle insurances at best prices with 24/7 claim settlement. It offers different types of policies for both personal and commercial vehicles. It has established its brand across different regions in India.

Around 90% of the businesses today use personalized services. The company wants to launch different personalized experience programs for customers of VahanBima. The personalized experience can be dedicated resources for claim settlement, different kinds of services at doorstep, etc. Inorder to do so, they would like to segment the customers into different tiers based on their customer lifetime value (CLTV).

Inorder to do it, they would like to predict the customer lifetime value based on the activity and interaction of the customer with the platform. So, as a part of this challenge, your task at hand is to build a high performance and interpretable machine learning model to predict the CLTV based on the user and policy data.

Approach

- After importing data, I did some **EDA** (check null values, check distributions and skewness, checked unique values etc), I found that there was skewness in two numerical columns.
- Following that I made baselines of two models LightGBM, Catboost and checked which model is performing best.
- Then, I trained all the models in **10-fold Cross-validation**, experimented various techniques by trusting OOF score and taking mean of predictions on test set of all the folds
- I made observation that Catboost is performing better than LightGBM.
- After that I experimented various type of pre-processing-
- **Encoding technique** – initially encoded with .factorize() but subsequently went one-hot-encoding, I also tried training models by not encoding categorical columns but OHE was giving me best local score.
- **Transformations** – Power transformations and exponent transformations, did not gave me good score.
- **Feature Engineering** – I tried transforming numerical columns by grouping various cat columns and taking mean but did not gave me good score.
- I combined some categorical columns which gave me good score.
- **Hyperparameter tunning** – I tunned hyper parameters with the help of **optuna**.
- Taking about hyper parameters, I took n_estimators high but provided **early stopping parameter** as well in order to avoid overfitting.
- Subsequently I tried other models such – Neural Network, Random Forest, etc. in order to combine them to ensemble but decided to go with my best model i.e. catboost.