APPROACH

## Data Analysis

❖ The first step of the process is to analyze the given data for any error and Null values and deal with them.
  - This is done by using "**Sklearn-SimpleImputer**" with "Most Frequent" strategy which places the most repeated values instead of the nulls.

❖ The next process is check if there are any "Categorical Values" and convert them into "Numerical values"
  - This is done my using "**Sklearn-OrdinalEncoder**", it changes the categorical values into numerical values. (Ex. A,B,C → 0,1,2)

❖ After that, the "Correlation" between the data and the target class is checked to find if there useful information that is needed in order to Proceed with "**Feature Engineering**"

Feature Engineering

❖ After Analyzing the Correlation data its found the target data "CLTV" has high positive correlation with "Claim Amount".
   o Now that data that highly correlates with Claim Amount are "Area" and "Num_Policies".
   o With the help of these to data a new data "Claim_Prob" which denotes the chances of claiming amount from the company is newly created.
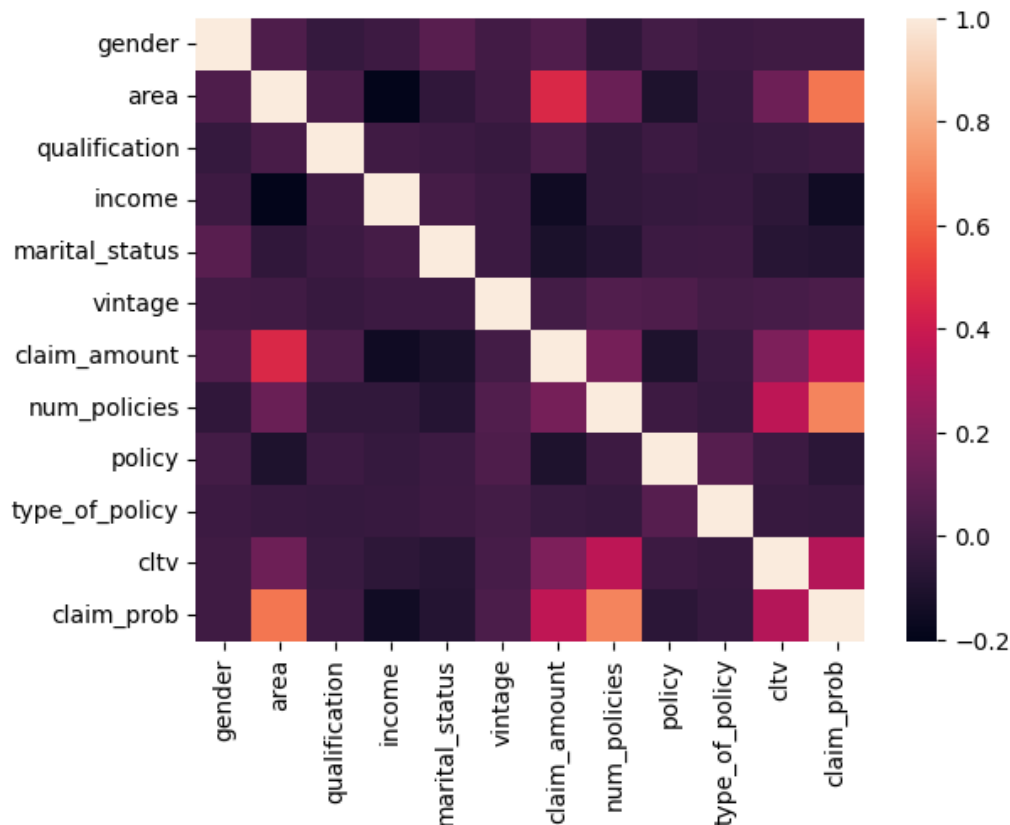
Data Splitting

❖ Now the Data is split into Training and Validation set from the existing Training data that is provided. (Training set – 99.95% and Validation set – 0.05% .

Model Selection

❖ The models chosen for this dataset are,
  - ○ XG Boost
  - ○ Light Boost
  - ○ Cat Boost

Hyper Parameter Tuning

❖ Hyper parameter tuning is done the help of "HyperOPT" Library which searches best parameters in the given range of options provided by user.
❖ After finding the Optimal Hyper Parameters, we are moving to the next step that is "Fine Tuning of Hyper parameter"

Fine-Tuning Hyper Parameters

❖ After finding the optimal parameters, we use "AutoML" Library to further optimize the parameters we got from the previous step to get the best results.

Voting Regressor

❖ After Training the models with the hyper parameters we got now, we are using the "Voting Regressor" to get even more accuracy instead of just using one Model.
  - ○ XG Boost, Light Boost as well as Cat Boost is provided to the Voting Regressor to find the final Result.

Test Data Predication

❖ Now with the help of Voting regressor we are predicting the test data provided and writing the results as ".CSV" in the submission format.

Prepared by : Sairam H

Email : 26thofficial.creator@gmail.com

GitHub : https://github.com/26th-Official/

LinkedIn : https://www.linkedin.com/sairam-h/