Business Problem

Target Problems

Call Center Resource Allocation

Forecasts the number of times a policyholder is likely to call, which aims to optimize resource allocation and enhance cost-efficiency in call center operations.

Annual Premium Amount Decomposition

Break down the independent variables behind the annual premium amount to optimize the pricing strategy of the company's insurance products.

Approach

Predictive model on the number of calls made by customers that can draw insights to allocate resources

Interpretation of the predictive model on annual premium amount to understand the key price drivers

Interpretable regression models that can be easily integrated into current business practice

Boosting Algorithm

Catboost

- Incorporating all independent features
- Experimenting **Poisson** and **RMSE** loss function

Catboost on Feature Interactions

- Incorporating all independent features
  - Capture the **interaction** among features via 2-degree polynomial transformation
- Experimenting **Poisson** and **RMSE** loss function

Stacked Catboost – Address the Imbalance

- Incorporating all independent features
- Building **2 stacked models** to maximize accuracy
  - **Classifier** to predict if policyholders will call
    - **Regressor** to predict call counts, trained only on policyholders w/ call counts > 0
- Experimenting **Poisson** and **RMSE** loss function

Hyperparameters Tuning

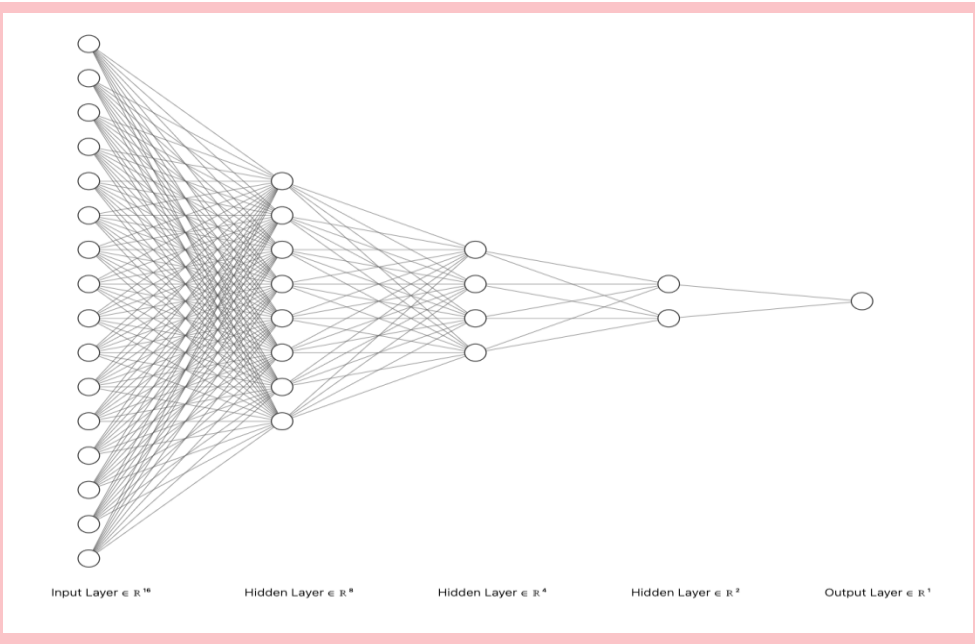
Catboost on Feature Interactions

The feature-interactions-augmented model's relative Gini on public leaderboard improved from 0.25513 to 0.25699 compared to the baseline Catboost model

Model Name	Validation Set	Private Leaderboard	Public Leaderboard
	Relative Gini		
Catboost	0.339316	0.2448	0.25513
Catboost - Interaction	**0.341306**	**0.2474**	**0.25699**
CatBoost - Stacked	0.78201	0.23207	0.23365

Classifier is not accurate enough in detecting policyholders w/ no call counts, while **Regressor** is powerful

Neural Network



- **4 dense layers** with decreasing size (**1024, 512, 256, 128**) + **1 output layer**
- **Activation Function**: Exponential Linear Unit
- **Dropout Rate**: 0.2 after each of first three dense layer --- Prevent Overfit
- **Optimizer**: Adam Learning Rate: 0.001
- **Loss Function**: MAE
- **Metrics**: Mean Absolute Error

Neural Network Model

- Log-transformed **target variable** to reduce weighting on 0 values and handle skewness
- Imputed Missing value by using mean of each feature
- A learning rate scheduler reduces the learning rate by 5% every epoch after the 50th epoch

Model Name	Validation Set	Private Leaderboard	Public Leaderboard
	Relative Gini		
Regression	0.310573	0.24787	0.25483
Neural Network	**0.3423**	**0.24902**	**0.25579**