

A more detailed explanation is provided in the EDA section of the code.

Feature selection:

1. The features like `Is_Active`, `Credit_Product`, `Age`, `Vintage`, `Avg_Account_Balance` seemed like some of the important features after EDA.
2. People with `Credit_Product` value `NaN` look to have a better chance of conversion, therefore ignoring `NaN` wasn't an option so I added that as a new category.
3. Distribution of `Avg_Account_Balance` was almost a power-law distribution so changing that to normal seemed like a good strategy.
4. People having lower value of `Age` and `Vintage` have smaller chances of turning to lead so creating a new feature to manifest this property made sense. We created a new feature $\text{Age} * (\text{Vintage} // 12)^2$ and is shown to have a good value of feature importance.
5. There were so many values for the categorical feature `Region_Code` so it seemed like a better way to encode them by imputing their respective mean of `Is_Lead`.
6. `Gender` alone is not that important feature but might become a valuable feature if used with other features like `Channel_Code`.
7. I also tried using different scaling methods like Min-Max and Standard scaler to see how our models respond to that.
8. For Categorical features, I tried Label encoding (for binary features) and one-hot encoding.

Model Selection:

I have tried multiple different models such as Logistic Regression, Random Forest, XgBoost, LightGBM, CatBoost, Deep Learning. All performed well but XgBoost seemed to have outperformed all of them. Even though I was really torn between CatBoost (Because of its unique way of handling categorical features) and XgBoost but CatBoost tended to overfit a little probably needed some better featurization which could have been possible had time permitted.