Approach:

i. A brief on the approach used to solve the problem.

Since it's a classification problem of classifying a user as – whether he/she will be **churned** or **not**, I compared various classification algorithms with cross validation score of macro f1-score. It's shown with plots in my code (python jupyter notebook).

And, I selected top 4 algorithms to train - LGBM, XGBoost, Random Forest and Linear Discriminant Analysis. After trying multiple training rounds, found that weighted ensemble of XGBoost and LGBM is doing consistently good. So, concluded the approach.

ii. Which Data-preprocessing / Feature Engineering ideas really worked? How did you discover them?

XGBoost top 5 features: Balance, Income_Age_Ratio, Age, Balance_Income_Ratio, Balance_Product_holdings_ratio

LGBM Top 5 features: Age, Transaction_Status, Age_Product_Holdings_ratio, Balance_Product_Holdings_ratio, Gender_num

As their names specify the meanings, some ratio features along with Balance, Age and Gender features, worked really well.

I discovered them by creating various pairs of ratio features, since sometimes the ratio tells us the significant difference between classes and they even increase the difference if difference in numerator is increased or, difference in denominator is decreased or, it's even great when both happens simultaneously.

iii. What does your final model look like? How did you reach it?

My final model is weighted ensemble of XGBoost Model and LightGBM Model i.e.

The final probability of Churn = 0.4*XGB + 0.6*LGBM

XGB = probabilities for class 1(i.e. Churn) from XGBoost model

LGBM = probabilities for class 1(i.e. Churn) from LightGBM model

And, the cut-off probability is 0.42 I.e., if "The final probability of Churn"**>0.42,** it'll be classified as **Churn,** otherwise not.

I reached it via starting with multiple classification algorithms, tuning their hyper parameters, then trying many ensembles of top 4 algorithms and found that this ensemble/model, at the specified probability cut-off (0.42) is **consistently doing good across train, validation and test set.**

Used Ensemble of algorithms to create robustness. In point .ii., we can see the difference between top 5 features of the two algorithms, which is creating additional robustness to the final model, along with their different learning processes.