Bank Term Deposition Subscription Prediction

Binary Classification Model Demo

Ren Hwai, 2024 June

Information Links

- **Github page for the code:**

  https://github.com/Ren1990/house_price_reg_model

- **Dataset from Kaggle:**

  https://www.kaggle.com/datasets/thedevastator/bank-term-deposit-predictions

- **My Linkedin:**

  https://www.linkedin.com/in/renhwai-kong/

- **My Tableau:**

  https://public.tableau.com/app/profile/kyloren.kong/viz/Demo_2024InvestmentPortfolio/DBPortfolio

- **My GenAI Job Interviewee Agent :**

  https://renhwaichatbot.streamlit.app/

About Myself

Hi! This is me, Ren Hwai, chilling in Iceland. Happy family trip during my career break!

"You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future." - Steve Jobs

After working in top US semicond company for 8 years as Senior Technology Development Process Engineer & Smart Manufacturing Analyst (Eng. IV), I take a long break to sharpen my Python skill in data science & analysis, and study for CFA (Chartered Finance Analyst) to look for new industry exposure and work opportunity.

Executive Summary

- XGB Classification Model is trained to predict whether the customer will subscribe bank term deposit during bank marketing campaign with 0.83 Balanced Accuracy.
- The target response is binary (‘yes’ or ‘no’ subscription). Main challenge from raw dataset is imbalanced dataset with only 11% successful subscribe cases
- Data transformation is performed before model training. 5-fold Cross Validation is used to reduce overfit risk.
- Optuna package is used to optimize model hyperparameter; less important features are removed (from 37 to 31) during model optimization.
- Based on model feature importance, factors which affect the outcome are:
  - How long since the last marketing campaign had contacted the customer
  - Whether the customer has subscribed the bank term deposit before
  - What are the month and day the campaign contact customer

Introduction

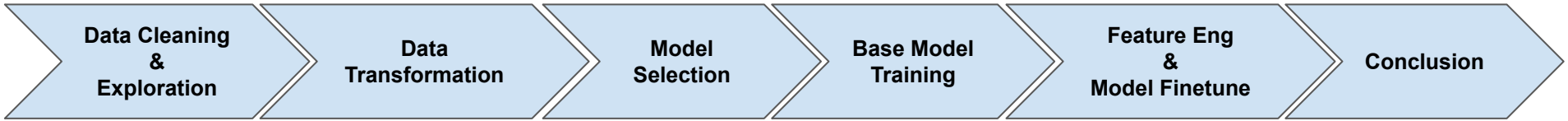
- This dataset, titled ‘Direct Marketing Campaigns for Bank Term Deposits’ from Kaggle, is a collection of data related to the direct marketing campaigns conducted by a Portuguese banking institution.
- These campaigns primarily involved phone calls with customers, and the objective was to determine whether or not a customer would subscribe to a term deposit offered by the bank.

Objective

- The goal is to predict the outcome of the marketing campaign and study what are the key factors affecting the result.

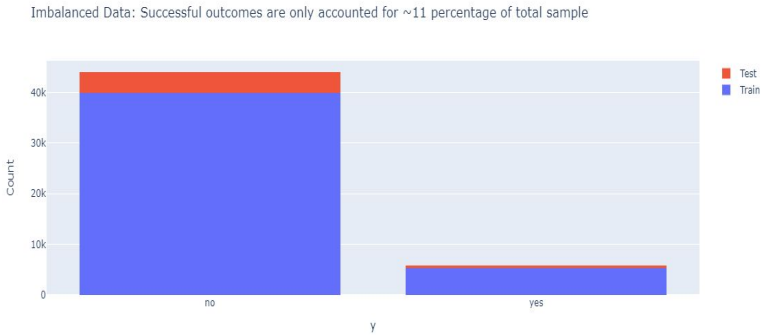
Model Training

- The model targets to predict whether the customer will subscribe the bank term deposition from marketing campaign.
- The target response, ‘y’ has binary values:
  - ‘yes’ : customer subscribes the bank term deposition
  - ‘No’: customer does not subscribe the bank term deposition
- This is a **Binary Classification Problem** using supervised learning.
- Below is the model training flow:



Dataset Overview

1. Data provided:
   a. Train.csv (45,211 rows x 17 cols).
   b. Test.csv (4,521 rows x 17 cols).
2. Good data quality, does not contain duplication, missing or single value data.
3. Imbalanced data: only 11% of the data are 'yes' (i.e. success subscription). **Balanced Accuracy score will replace Accuracy to assess model performance.**



Balanced Accuracy is average accuracy of both 'yes' and 'no' accurate prediction rate. In imbalance dataset, Accuracy will be heavily screwed by 'no' prediction accurate rate

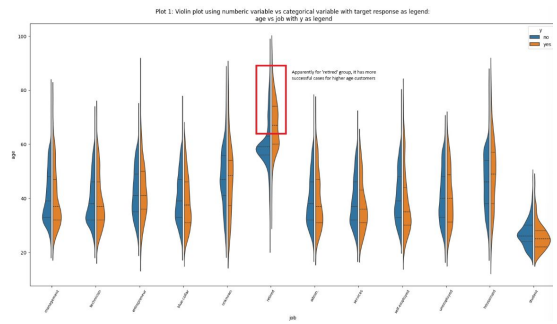
Accuracy:
[Correct 'Yes' Prediction+Correct 'No' Prediction] / [All 'Yes' Prediction+All 'No' Prediction]

Balanced Accuracy:
0.5 * [Correct 'Yes' Prediction/All 'Yes' Prediction+Correct 'No' Prediction/All 'No' Prediction]

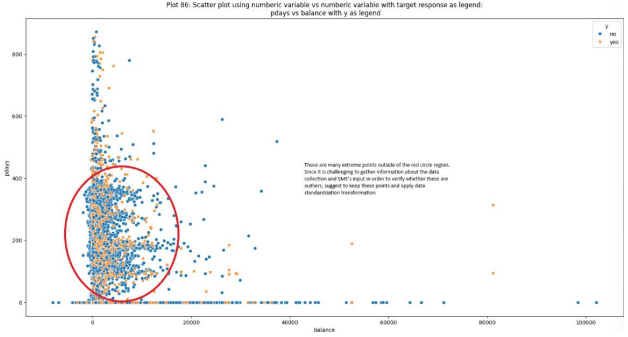
Data Exploration

1. No obvious distinct information is uncovered from data exploration
2. Data is fairly evenly distributed, but there are outliers for few features. Robust Scaling will be used for data transformation
3. Below are examples of data visualization:

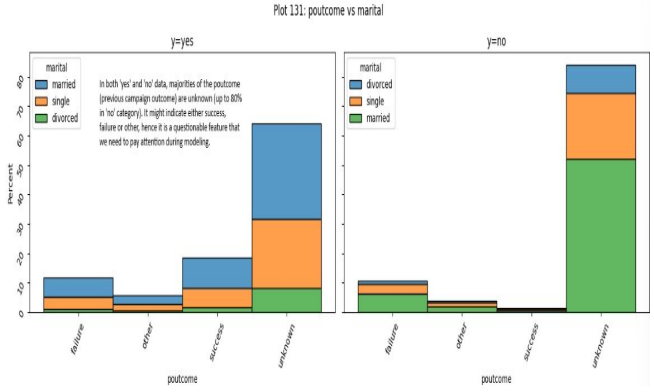
Violin Plot



Scatter Plot



Stacked Bar



Data Transformation is performed before fitting data for model training

1. Convert 'month' into numeric number of month.

Index	Month
0	May
1	June
2	July
3	Aug
4	Sept

Index	Month
0	5
1	6
2	7
3	8
4	9

2. Binary value categorical feature -> label encoder.

Index	default	y
0	no	yes
1	no	no
2	yes	yes
3	no	yes
4	no	no

Index	default	y
0	0	1
1	0	0
2	1	1
3	0	1
4	0	0

3. Multiple value categorical feature -> one-hot encoder. New columns will be created

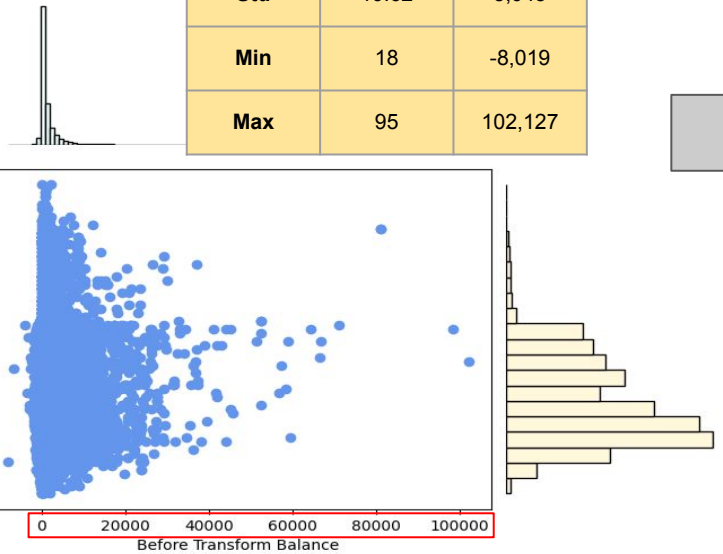
- In example below, if label encoder is used, ‘unknown’ will be assigned by 0, ‘primary’ -> 1, ‘secondary’ -> 2 and ‘tertiary’ is -> ‘3’, and total columns will remain same.
- Downside of using label encoder is, model will treat ‘primary’ is higher than ‘unknown’ (1>0) etc.
- Since this is not the true relation or comparison, label encoding should be avoid to prevent machine learning to pick up this relation

Index	education
0	unknown
1	primary
2	secondary
3	tertiary

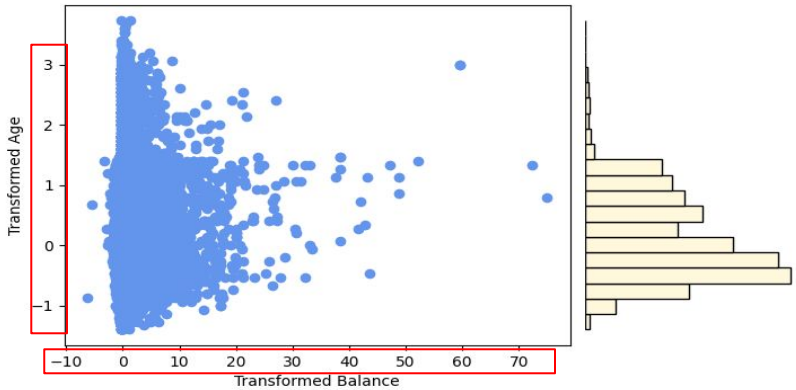
Index	education_unknown	education_primary	education_secondary	education_tertiary
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

4. Numeric feature -> robust-scaler encoder.

	age	balance
Mean	40.9	1,362
Median	39.0	448
Std	10.62	3,045
Min	18	-8,019
Max	95	102,127



	age	balance
Mean	0.1291	0.6742
Median	0.0000	0.0000
Std	0.7079	2.2454
Min	-1.4000	-6.2441
Max	3.7333	74.9845



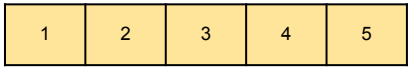
Model Selection

- There are many classification models for selection. Instead of trying all models and perform fine tuning for all, Model Selection is applied to select potential model candidate for optimization.
- Below are 6 classification models used in Model Selection:
  a. SVC (Support Vector Classification)
  b. Decission Tree Classification
  c. XGB (ExtraGradientBoost) Classification
  d. RandomForest Classification
  e. ExtraTree Classification
  f. LogisticRegression

K-fold Cross Validation

- 5-fold (i.e. K=5) Cross Validation (CV) is applied in model training to obtain generalized model to avoid overfitting caused by machine learning
- The train.csv provided by dataset is randomly split into 5 equal sets (test.csv is not used).
  - 4 sets are used for training model and 1 set is used for validating model performance
  - 1 set is used for model result validation
  - Rotate the set so that all 5 sets have been used for validation
- Average Balanced Accuracy of 5 folds result is used to assess model performance

1. Split train.csv randomly into 5 sets



train.csv

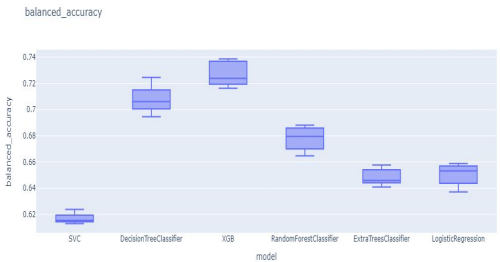
2. Split train.csv randomly into 5 sets for cross validation

Train Data

Validation Data

K=1	1	2	3	4	5
K=2	1	2	3	4	5
K=3	1	2	3	4	5
K=4	1	2	3	4	5
K=5	1	2	3	4	5

3. Assess the 5-fold model performance result

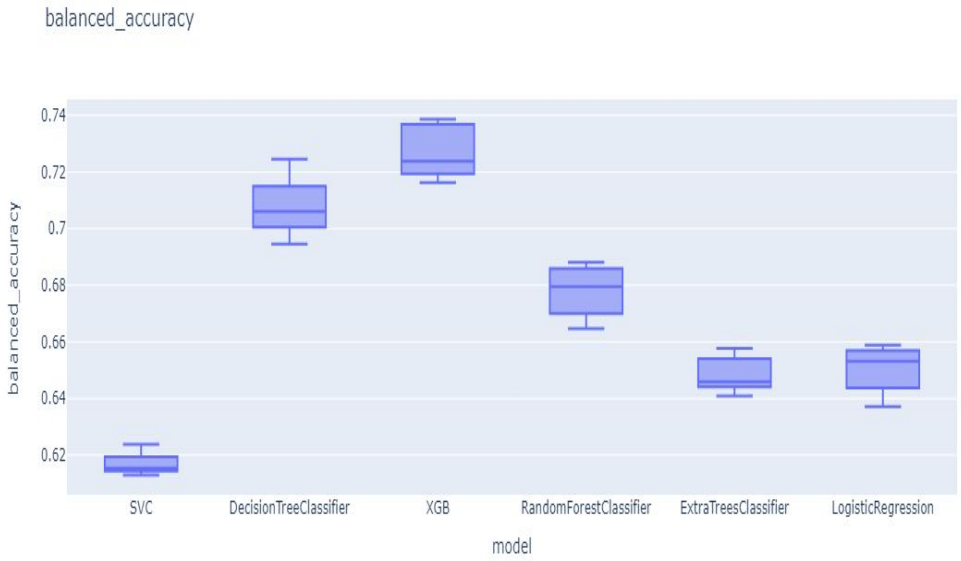


Model Selection Outcome: XGB Model is selected

1. Summary of average score: XGB has highest average balanced accuracy in 5-fold assessment.

Model	Balanced Accuracy	Precision	Recall	F1 Score
SVC	0.6170	0.6563	0.2516	0.3635
DecisionTree Classifier	0.7079	0.4786	0.4859	0.4821
XGB Classifier	**0.7271**	**0.6244**	**0.4936**	**0.5512**
RandomForest Classifier	0.6778	0.6550	0.3825	0.4827
ExtraTrees Classifier	0.6485	0.6487	0.3201	0.4285
Logistic Regression	0.6503	0.6500	0.3237	0.4230

2. Box plot of 5-fold result doesn't show abnormal wide spread (variation) for all models.



Create Base XGB Model ('Model1') with Optuna

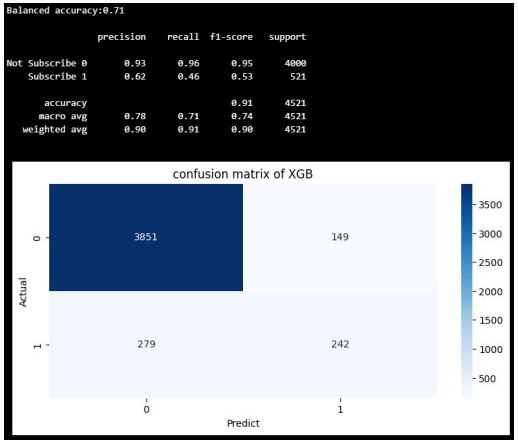
- Optuna is a Python Library used for Model Hyperparameter Tuning:
  - Setup hyperparameter range for optimization search space
  - Setup Cross Objective function: maximize average Balanced Accuracy calculated from 5-fold cross validation
  - Leverage Optuna Algo to search best hyperparameter
- 'Model1' has achieved 0.77 Balanced Accuracy, further breakdown shows:
  - Precision to predict 'unsuccessful subscription' is increased from 93% to 95%
  - Precision to predict 'successful subscription' is dropped from 62% to 60%

1. Standard hyperameter achieve 0.71 Balanced Accuracy using Test.cvs

hp_example={'gamma': 3,
'max_depth': 6,
'reg_alpha': 5,
'reg_lambda': 5,
'subsample': 0.5,
'importance_type': 'gain',
'seed': 1000,
'booster': 'gbtree',
'eta': 0.1,
'objective': 'binary:hinge'}

Balanced accuracy:0.71

	precision	recall	f1-score	support
Not Subscribe 0	0.93	0.96	0.95	4000
Subscribe 1	0.62	0.46	0.53	521
accuracy			0.91	4521
macro avg	0.78	0.71	0.74	4521
weighted avg	0.90	0.91	0.90	4521

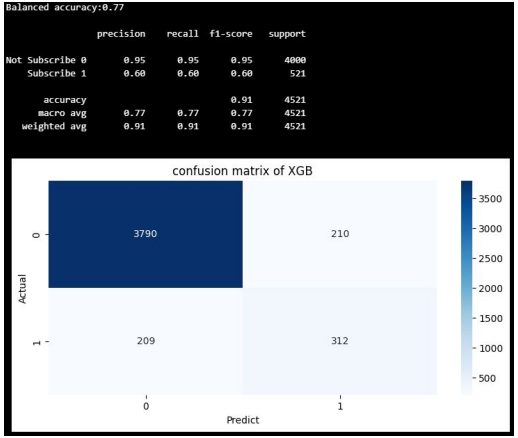


2.After Optuna Hyperparameter Tuning, Balanced Accuracy is increased to 0.77

hp01={ 'gamma': 8.142983479923078,
'max_depth': 17,
'reg_alpha': 1.0040684463586094,
'reg_lambda': 0.6804433417084434,
'subsample': 0.9472737264861752,
'importance_type': 'total_gain',
'seed': 156,
'eta': 0.07515307644937705,
'booster': 'gbtree',
'objective': 'binary:hinge'}

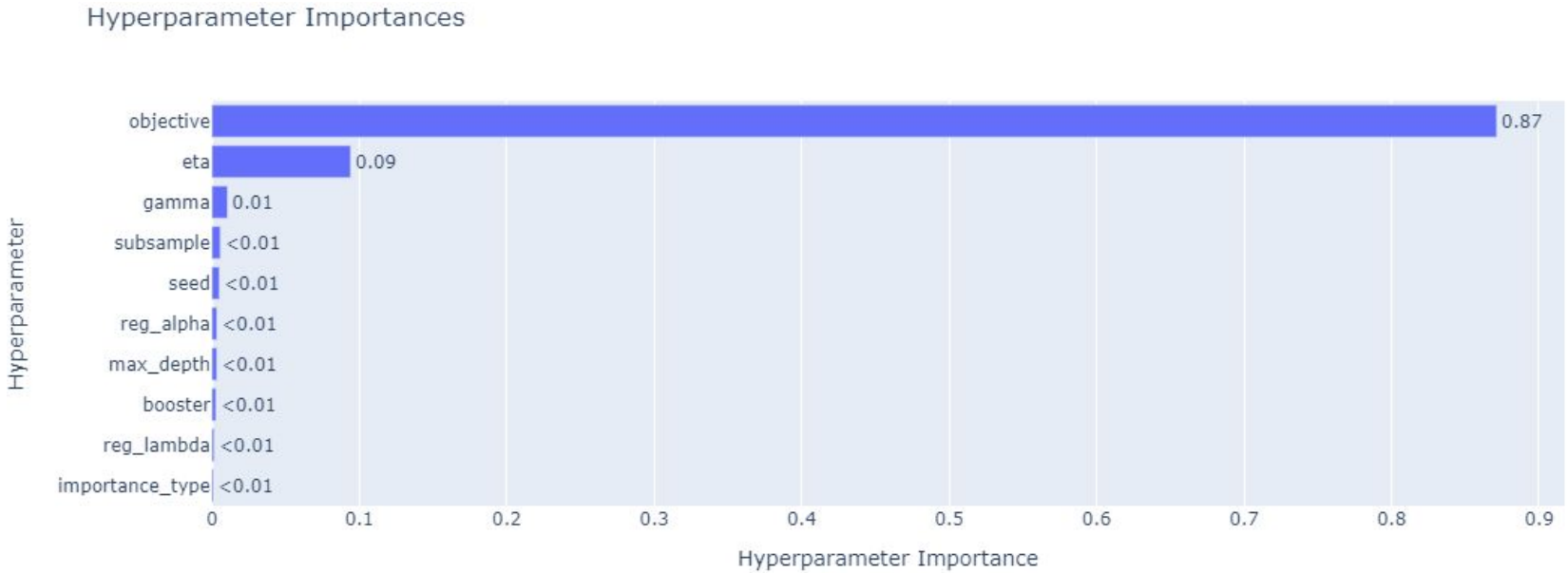
Balanced accuracy:0.77

	precision	recall	f1-score	support
Not Subscribe 0	0.95	0.95	0.95	4000
Subscribe 1	0.60	0.60	0.60	521
accuracy			0.91	4521
macro avg	0.77	0.77	0.77	4521
weighted avg	0.91	0.91	0.91	4521



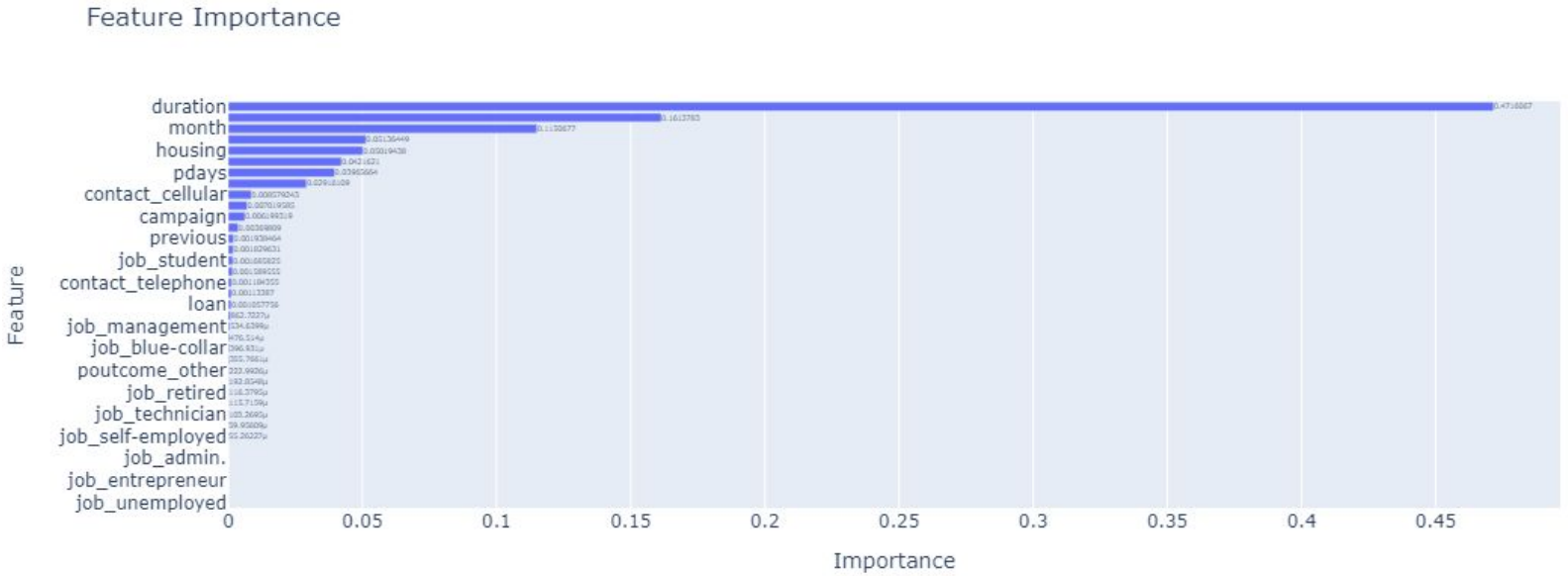
XGB Hyperparameter Importance of ‘Model1’

- Through Optuna hyperparameter importance feature, it is found that ‘objective’ and ‘eta’ are the most important hyperparameter.



Feature Importance of ‘Model1’

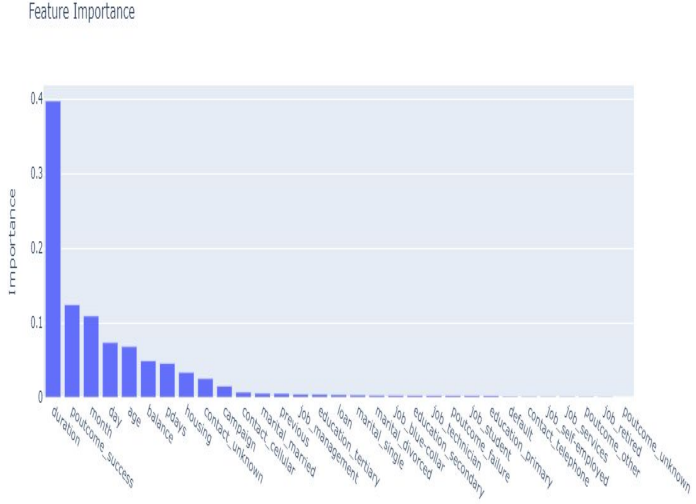
- ‘Model1’ contains 37 features. Most important features are:
  - ‘duration’: 'Duration (in seconds) of the last contact with customers during the previous campaign.'
  - ‘poutcome’: 'Outcome from the previous marketing campaign.'
  - ‘month’ and ‘day’: Month and day when the customers were contacted



Final Model

- Low importance features are removed
- Progressively fixing hyperparameter ‘objective’ and ‘eta’, and finetune other hyperparameters
- Final model has reduce features from 37 to 31. With optimized hyperparameter test data Balanced Accuracy is improved from 0.77 to 0.83
- Further features removal reduces the Balanced Accuracy, hence stop at 31 feature as Final Model

1. Final Model 31 features

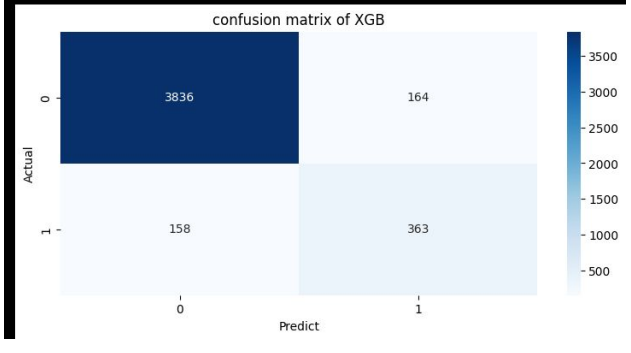


2. Final model achieved 0.83 Balanced Accuracy with optimized hyperparameter

final_hp={ 'gamma': 4.123684419610142,
'max_depth': 17,
'reg_alpha': 0.5487894695587288,
'reg_lambda': 0.7363883398550661,
'subsample': 0.7483511743637902,
'importance_type': 'total_gain',
'seed': 388,
eta': 0.05048155869197371,
'booster': 'dart',
'objective': 'binary:hinge'}

Balanced accuracy:0.83

	precision	recall	f1-score	support
Not Subscribe 0	0.96	0.96	0.96	4000
Subscribe 1	0.69	0.70	0.69	521
accuracy			0.93	4521
macro avg	0.82	0.83	0.83	4521
weighted avg	0.93	0.93	0.93	4521

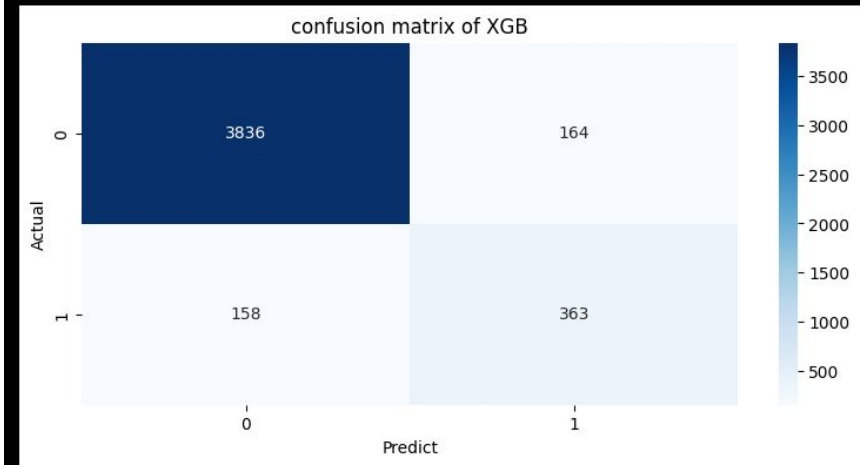


Final Model is fairy well in predicting the outcome

- Final model has achieved 0.83 balanced accuracy. Further breakdown shows:
  - Precision in predicting ‘unsuccessful subscription’ is increased from 95% to 96%.
  - Precision in predicting ;’successful subscription’ is increased from 60% to 69%.
- With 70% Recall for successful subscription, the model feature importance should reasonably explain the factors related to 70% successful subscription cases.

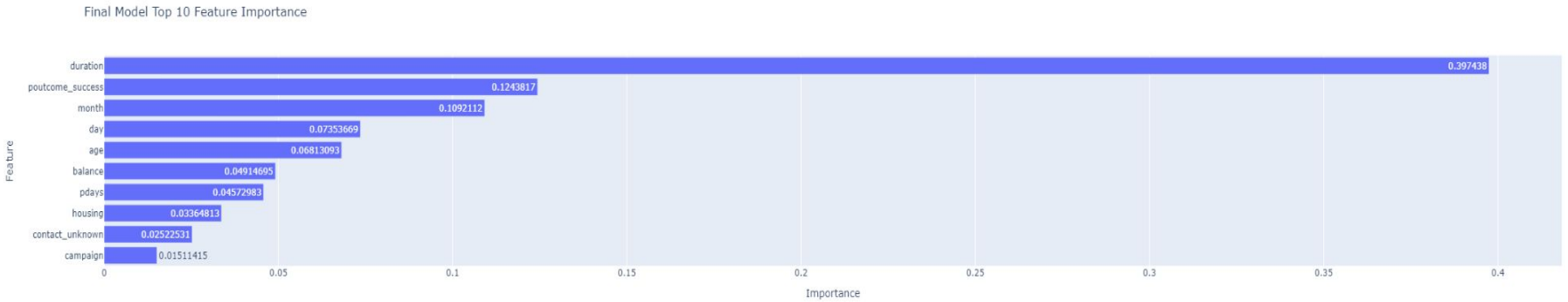
Balanced accuracy:0.83

	precision	recall	f1-score	support
Not Subscribe 0	0.96	0.96	0.96	4000
Subscribe 1	0.69	0.70	0.69	521
accuracy			0.93	4521
macro avg	0.82	0.83	0.83	4521
weighted avg	0.93	0.93	0.93	4521



Hypothesis on Success Subscription

- The model is trained with historical data, it might not be learning the causation relation between the factors and Success Subscription.
- Nonetheless, hypothesis is made based on the top model feature:
  - **'duration' (~40%)**: How long since the last marketing campaign was made with the customer. The customer may be displeased by too frequent marketing contact
  - **'poutcome_success'(~12%)**: The customer has subscribed the campaign in previous marketing campaign. This group of customers are likely the savvy customer who are looking for such bank term deposit or are experience with similar service
  - **'Month' and 'Day'(~11%+7%)**: The customer might have higher chance to subscribe the service after receiving salary/bonus, or less chance to subscribe due to planning to spend the cash for festival celebration
- A/B Testing on new data can be used to validate hypothesis above



Doubts on ‘poutcome’

- Although there is no abnormality found during Data Cleaning, however it is discovered that there ‘unknown’ value, potentially these are missing value and replaced with ‘unknown’.
- ‘poutcome_success’ is the 2nd important factor, however further deepdive shows that ‘success’ only accounts for 3.34%, and 81.7% is unknown.
- Through cross validation and out-of-sample test data validation, the model performance and feature importance are quite stable, hence the Final Model should be generalized and low overfit risk
- Although there are doubts, the model performance should be reliable