**Smart banking customer targeting using ML Ensemble for improved business operational efficiency through reduced cost per call (CPC)  
EDA, Data Preparation, Feature Engineering and Modeling Report**

By Sai Sharan

University of Memphis

# Introduction

Traditionally businesses reach out to prospect customer for encashing potential opportunities (can be in terms of cross sell or upsell). Though this process of targeting customer for potential cross sell or upsell in observed in many industries, it is very wide and frequent seen in banking industry were a customer holding a with a bank will be targeted for cross selling opportunities like loans, fixed deposits and term deposits etc., Banks traditionally uses various channels to reach their customer and one of the major such channel is Telemarketing. Though telemarketing seems like a very easy way to reach a customer at the same time it is very costly. As per industry standard typical Cost per Call (CPC) is around $2.7 to $5.6 [1] (it might difference from business to business), based this statistic we can estimate the possible impact of targeting a wrong prospect and importance of accurate targeting strategy. In the current project we plan to address this problem of high operating cost due to inaccuracy customer targeting using machine learning. As part of the analysis, we will be using banking telemarketing call data for predicting the propensity of a customer opting for cross selling, which can later be used by banking businesses for making better call plan as well customer target list.

Keywords: Business optimization, Machine Learning for CPC reduction, Call Center Optimization

Contents

[Introduction 2](#_Toc130750974)

[Data Overview 6](#_Toc130750975)

[Source 6](#_Toc130750976)

[Data Description 6](#_Toc130750977)

[Collection Process 7](#_Toc130750978)

[Data Summary Stats 7](#_Toc130750979)

[Metadata, Counts and Datatypes 7](#_Toc130750980)

[Quick Summary Stats 8](#_Toc130750981)

[Univariate Analysis 9](#_Toc130750982)

[Insights and Key Distributions from Univariate Analysis 9](#_Toc130750983)

[Age 9](#_Toc130750984)

[Balance, Duration & Campaign 9](#_Toc130750985)

[Pdays and Previous 10](#_Toc130750986)

[Categorical Variable Distribution 10](#_Toc130750987)

[Bi-Variate Analysis 12](#_Toc130750988)

[Insights from Bi-Variate Analysis 12](#_Toc130750989)

[Numerical Variable 12](#_Toc130750990)

[Categorical Variable 14](#_Toc130750991)

[Correlation Analysis 15](#_Toc130750992)

[One to One Correlation Analysis using Pearson’s Correlation Coefficient 15](#_Toc130750993)

[Multicollinearity Analysis using Variance Inflation Factor (VIF) 16](#_Toc130750994)

[Target Variable Class Imbalance 16](#_Toc130750995)

[EDA Conclusion and Next steps 16](#_Toc130750996)

[Data Preparation based in Insights from EDA 17](#_Toc130750997)

[Missing Value Treatment 17](#_Toc130750998)

[Outlier Treatment 17](#_Toc130750999)

[Outlier Treatment for Age 18](#_Toc130751000)

[Outlier Treatment for Call Duration 18](#_Toc130751001)

[Outlier Treatment for Balance 19](#_Toc130751002)

[Outlier Treatment for Campaigns 19](#_Toc130751003)

[Train and Test Data Preparation 19](#_Toc130751004)

[Feature Engineering 20](#_Toc130751005)

[Addressing Multicollinearity 20](#_Toc130751006)

[Treating Categorical Features 20](#_Toc130751007)

[Model Development 20](#_Toc130751008)

[Decision Tree 20](#_Toc130751009)

[Parameter Tuning 21](#_Toc130751010)

[Loss Metrics & Model Final Model Results 22](#_Toc130751011)

[Random Forest 22](#_Toc130751012)

[Parameter Tuning 23](#_Toc130751013)

[Loss Metrics & Model Results 24](#_Toc130751014)

[XGboost 25](#_Toc130751015)

[Parameter Tuning 25](#_Toc130751016)

[Loss Metrics & Model Final Model Results 26](#_Toc130751017)

[Ensemble of Decision Tree, Random Forest and XGboost 27](#_Toc130751018)

[Bias Validation 27](#_Toc130751019)

[Loss Metrics & Final Model Results 28](#_Toc130751020)

[Cost of ML Model Development 28](#_Toc130751021)

[Pre-existing model research and comparison 28](#_Toc130751022)

[References 29](#_Toc130751023)

# Data Overview

## Source

Data for current project has been procured from UCI Machine learning repository. Following is the description of repository (as per the website) “*The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. The archive was created as an ftp archive in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning datasets.*”

## Data Description

We are using “Bank Marketing Dataset” hosted in UCI repository for the current analysis. Data actually belongs to a Portuguese bank where existing bank customers are targeted for term deposit subscription over phone calls. The data provided has information related to customers and past behavior when targeted with marketing campaign. Following are key fields and their description as per UCI repository

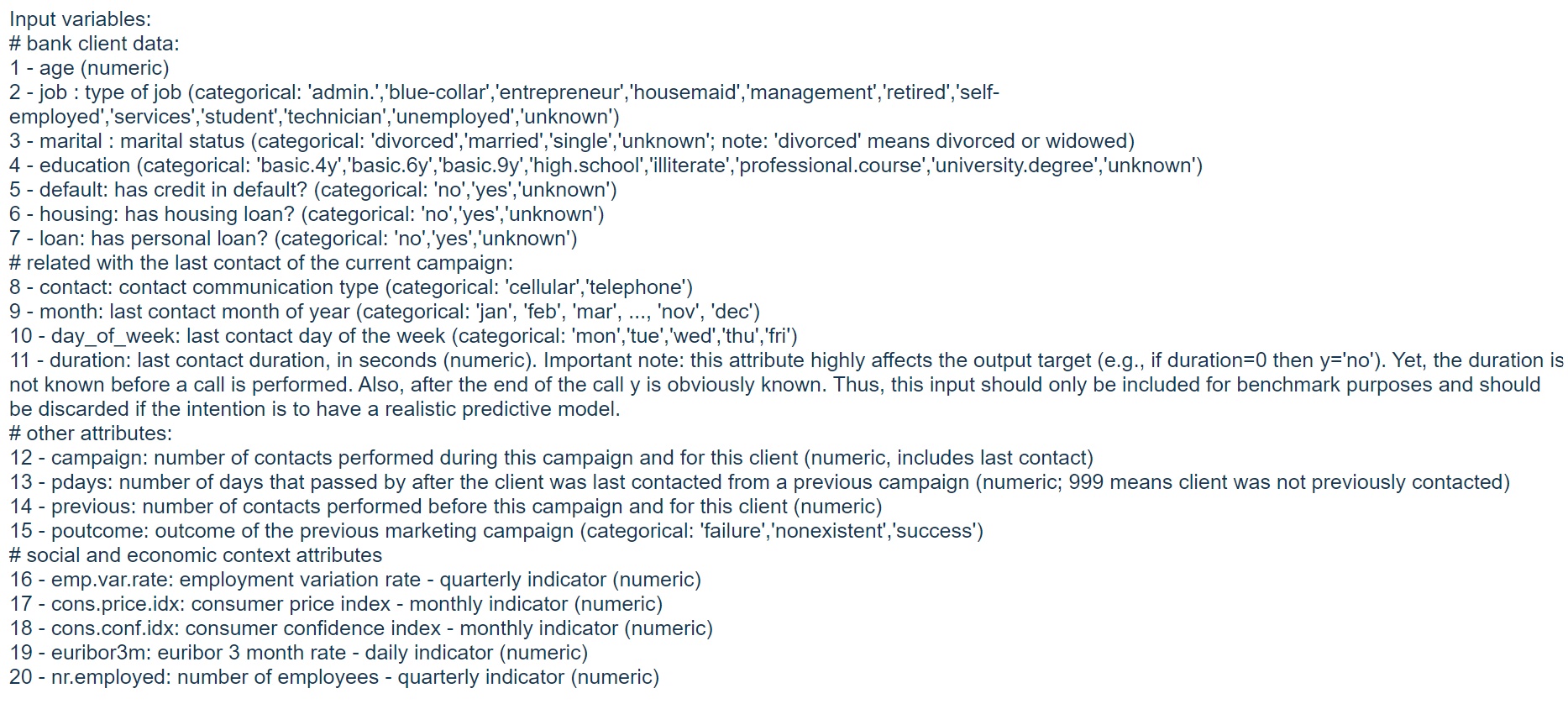


Image showing description of key fields in the dataset

## Collection Process

Data is directly downloaded from UCI website. As required information is available as single file there is no need for data merges. Data is available as CSV file which can be directly loaded into python for model development.

# Data Summary Stats

Understanding metadata and other key attributes like data size, data types only helps in making data preparation process smoother but also makes it faster as proactive data treatments can be made.

## Metadata, Counts and Datatypes

Dataset has close to 45K records with 17 columns. A good split of both categorical (10 fields) and continuous (7 fields) variables are available. Table 1 below show the datatype of each field.

|  |  |
| --- | --- |
| **Variable** | **Datatype** |
| job | object |
| marital | object |
| education | object |
| default | object |
| housing | object |
| loan | object |
| contact | object |
| month | object |
| poutcome | object |
| y | object |
| age | int64 |
| balance | int64 |
| day | int64 |
| duration | int64 |
| campaign | int64 |
| pdays | int64 |
| previous | int64 |

Table 1: *Datatypes of key fields*

## Quick Summary Stats

A high-level glance at the quick summary stats below (Table 2) shows that

1. All categorical fields have limited set of factors (max factor = 12 for job) hence no need for further bucketing
2. All the numerical feature ranges are in expected limits except balance and pdays which has negative values



Table 2: *High-level summary key summary states of data fields*

# Univariate Analysis

To understand the distribution of individual features univariate analysis has been performed on each feature of the dataset. Insights from the univariate analysis will be used for data treatment (if required for missing values, outliers and feature engineering) as well as model approach selection. To understand the univariate distribution Histogram have been used for numerical fields (post binning) and frequency plots are using for categorical fields.

Though univariate analysis has been performed on all the features only insightful distribution are reported in the current document. For detailed univariate EDA charts refer to EDA html file.

## Insights and Key Distributions from Univariate Analysis

Age**:** Age of the customer is nearly normally distributed with few outliers having age above 65 years which can be treated during data preprocessing stage (after careful observation)

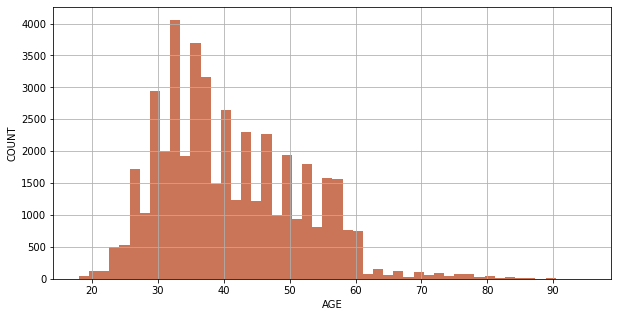


Chart 1: *Histogram of Customer Age*

Balance, Duration & Campaign**:** Balance, Duration and & Campaign variables are exhibiting right skewed distribution which kind of makes sense from intuition perspective. But these variables might need further treatment if models like linear regression are using for prediction**.** Additionally, balance has negative numbers which can be loans etc.,

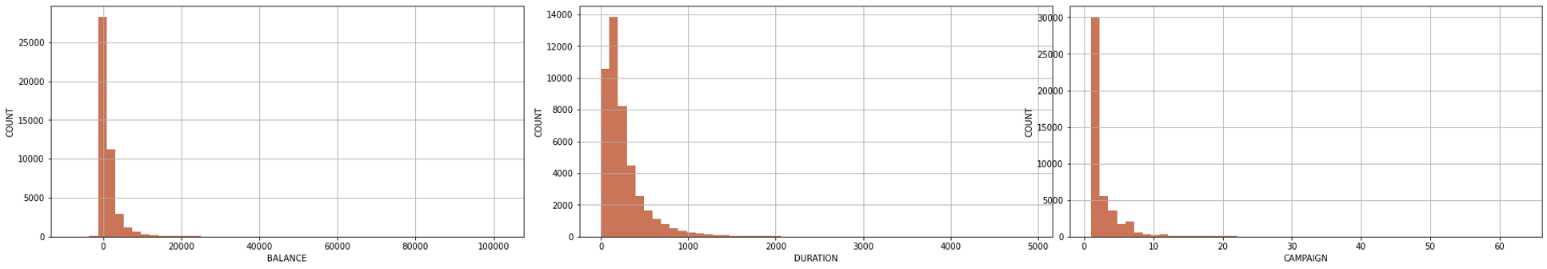
****

Chart 2: *Right Skewed Distribution of Balance, Duration & Campaign*

Pdays and Previous**:** Most of the values in Pdays and Previous are mostly zero. Non-zero values in these are of importance as not every customer is retarget during any campaign.

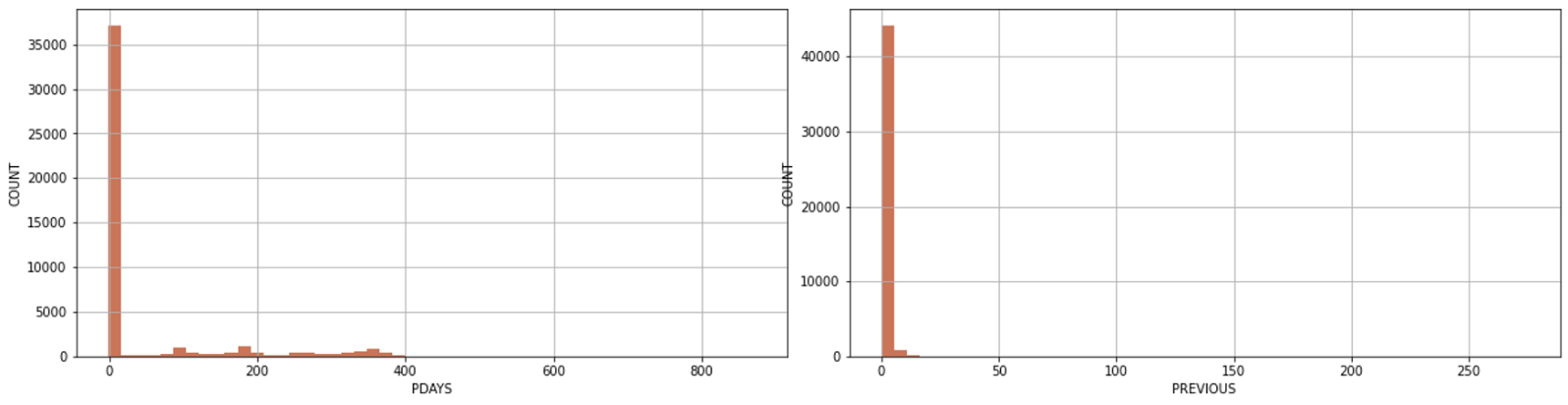
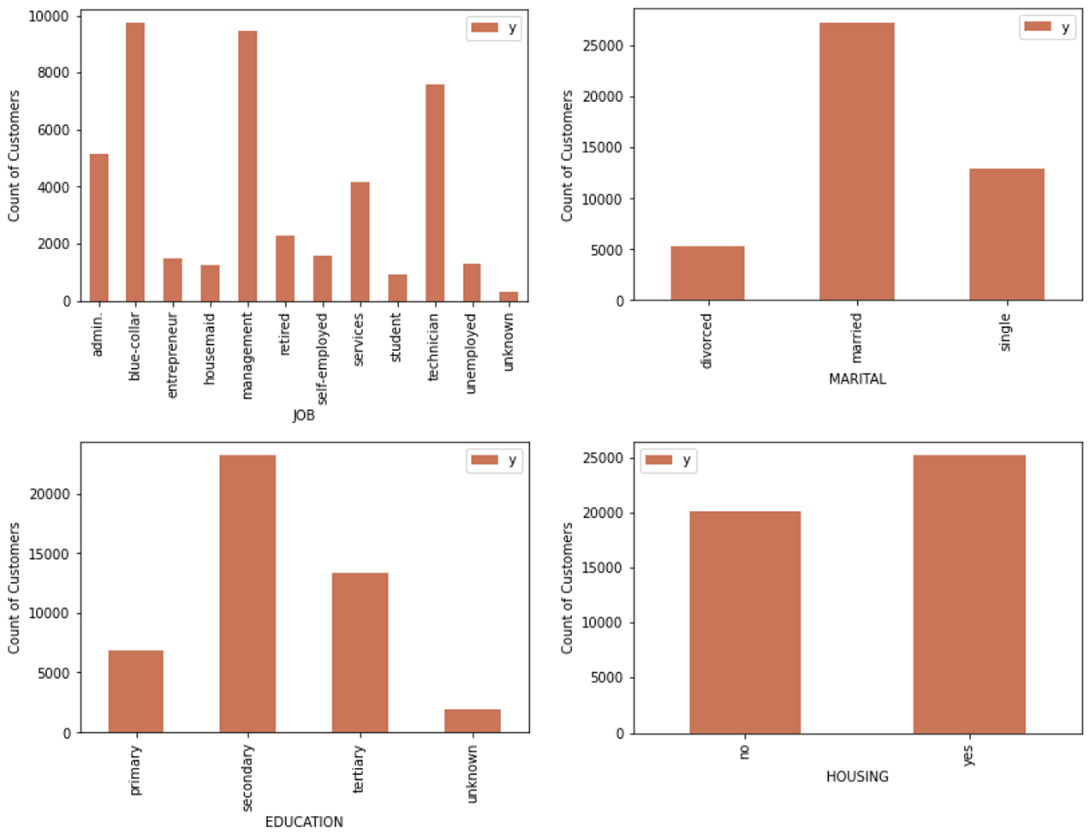


Chart 3: *Distribution of Pdays and Previous*

Categorical Variable Distribution**:**

1. All categorical variables have limited set of factors hence no need for further bucketing of categories before dummy coding of variables.
2. Very few customers are defaulters (default = ‘Yes’). Based on business intuition this can an important feature for model development
3. Job among all categorical variable has relatively higher number of factors, some of the factors can be excluded during preprocessing/model development stage based on their importance
4. Job and Education have factor values as ‘Unknow’, we need to be cautious while using this factor in the model as any insight on this factor is not actionable



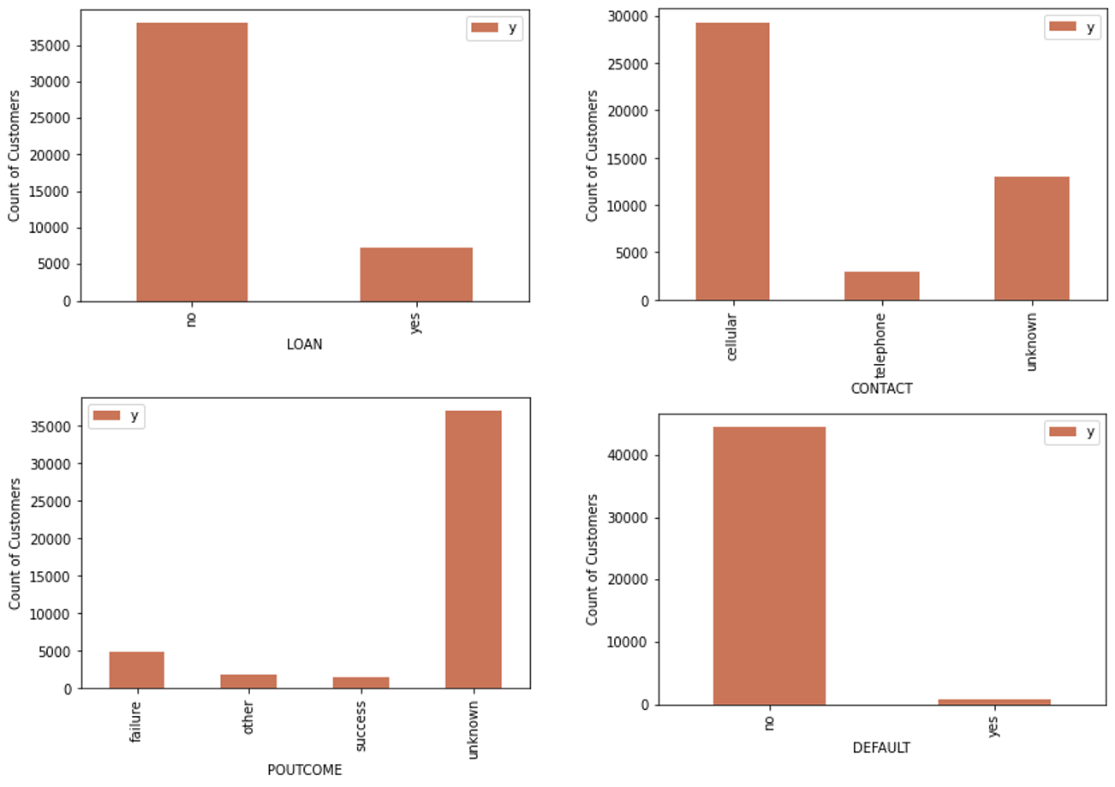


Chart 3: *Frequency plot of categorical variables across various factors*

# Bi-Variate Analysis

## Insights from Bi-Variate Analysis

Bi-Variate analysis on each feature (both categorical and continuous) has been performed for identifying possible the key features/factors impacting term deposit subscription. For categorical variables percentage term deposit distribution and for continuous variable average value feature value for with and without term deposit subscription has been used.

### Numerical Variable

* Average of age and day doesn’t show any significant difference for positive vs negative class [Refer to Chart 4]
* Balance, Duration, Campaign and Pdays variance shows very significant difference for positive vs negative classes [Refer to Chart 5]
  + People with high balance are showing higher tendency to opt for term deposit
  + Customers spending more time when an agent called for the first time are having higher tendency to opt for term deposits
  + Reaching a customer who has been targeted in previous campaigns are showing higher receptiveness in comparison with other customers

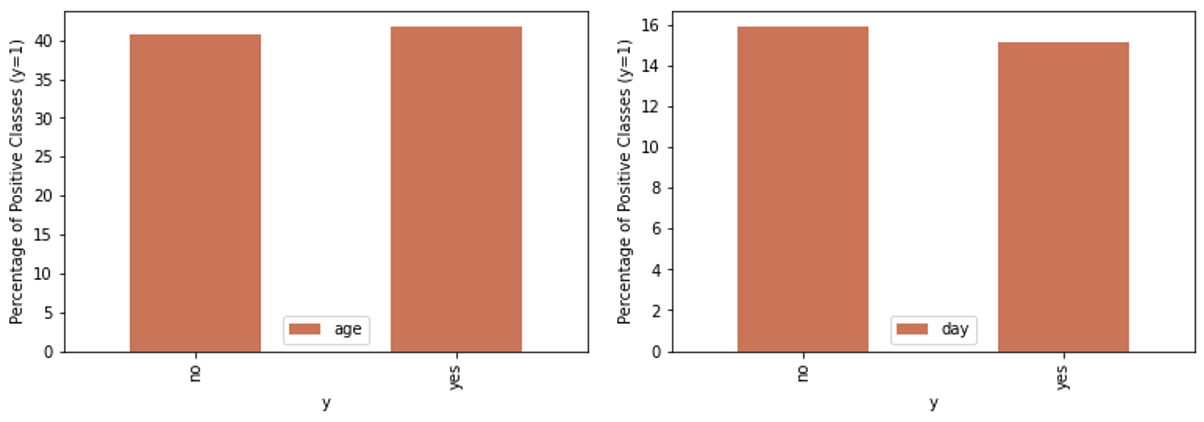


Chart 4: *Average of age, day vs positive & negative classes*

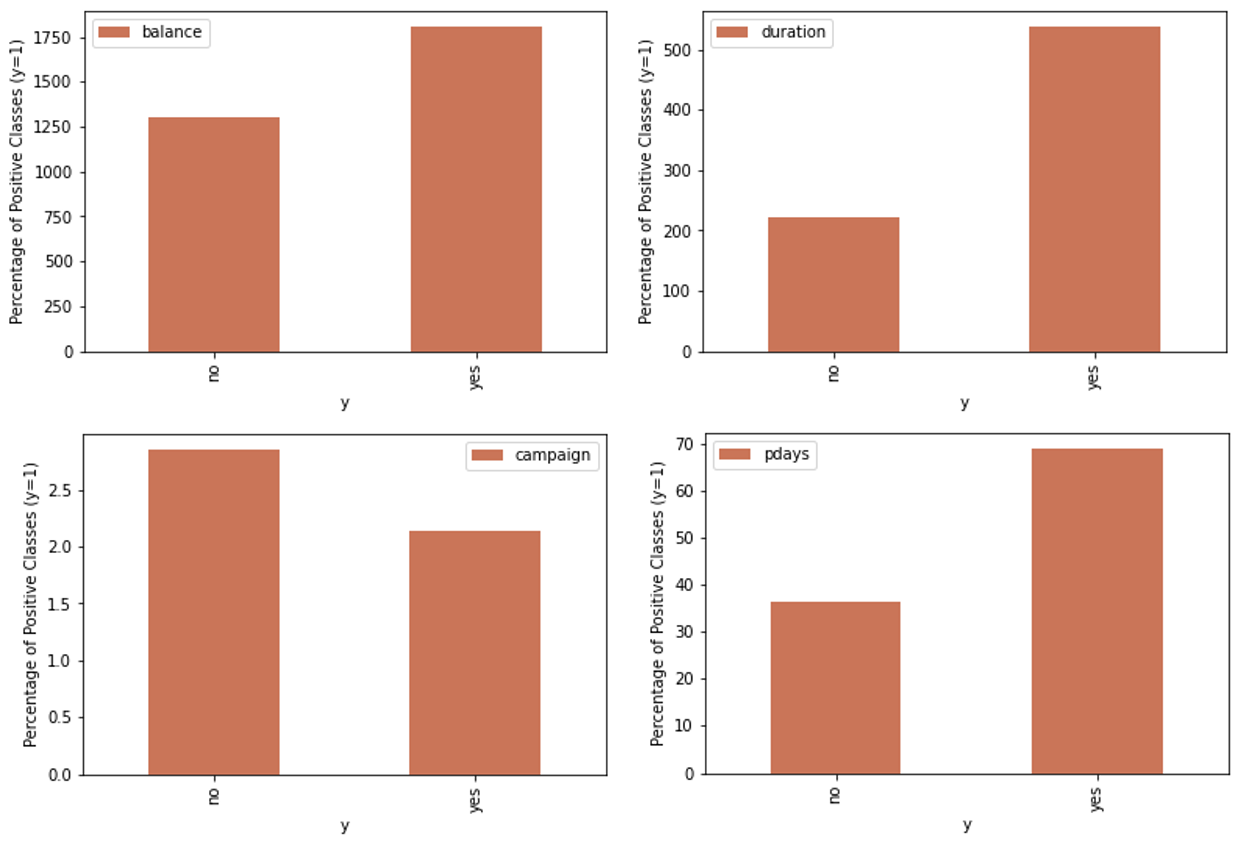


Chart 5: *Average of Balance, Duration, Campaign, Pdays vs positive & negative classes*

### Categorical Variable

* Customers who successfully opted in previous marketing campaign (poutcome) have higher changes of opting for term deposit if targeted
* Students and retired employees have higher percentage of term deposit opter than that of other job groups
* Customer who doesn’t own a house are preferring term deposit than that of customer with own house
* Lower the education level lower are the chances for taking term deposit
* Loan defaulting customers are showing lower receptance to term deposit
* Customers who don’t how an existing loan are opting more for term deposit than that of other customers

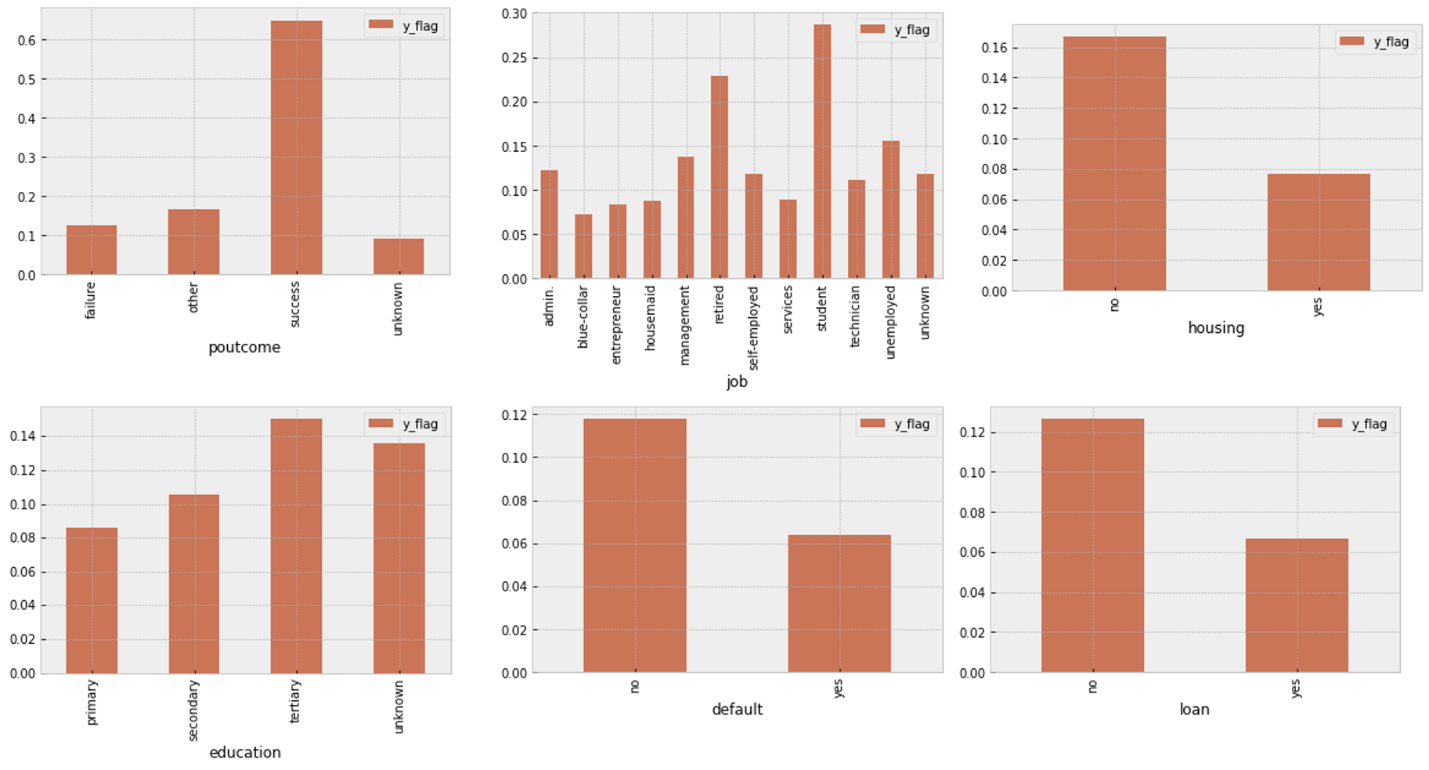


Chart 6: *Percentage of positive classes across different factors of categorical variable*

# Correlation Analysis

Correlation analysis has been performed on the given dataset post dummy coding categorical variables. We have used Pearson’s correlation coefficient for understanding variable one on one correlation and VIF has been used to check the multi collinearity of the variables. Following are key observations from correlation analysis

* poutcome – pdays, Education secondary - Education tertiary and Married\_single - marital\_married are very highly correlated with each other hence they to be appropriately treated while modeling [Refer to Chart 7]
* poutcome\_unknown, month\_may, day and marital\_married feature have > 5 VIF suggesting a strong presence of multicollinearity [Refer to Table 3]

## One to One Correlation Analysis using Pearson’s Correlation Coefficient



Chart 7: *Variable correlation heatmap*

## Multicollinearity Analysis using Variance Inflation Factor (VIF)



Table 3: *Variables with High VIF*

# Target Variable Class Imbalance

Only ~12% of the total customers have subscribed to term deposit (Refer to Chart 8) which means for every 100 records we have only 12 positive classes. This suggests a slight presence of class imbalance issues. During model development incase we observe large amount false negative class balancing might have to be performed.

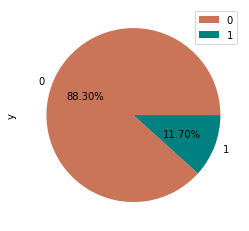


Chart 8: *Distribution customers subscribing for term deposit*

# EDA Conclusion and Next steps

1. Data has no missing values hence no need for missing value treatment
2. Certain outliers have been observed for age variable which can be treated during preprocessing stage
3. Majority for the numeric variables are showing skewed distribution, it is suggested to avoid methods like linear regression as data is violating modeling assumptions
4. High Balance, High Duration, customer targeted in previous campaign, customer last targeted, customers without house, customer without existing loans and customers without any defaults are characters showing high tendency to opt for term deposits – (Observation made from initial EDA) Hence these can be import variables for model development
5. poutcome – pdays, Education secondary - Education tertiary and Married\_single - marital\_married are very highly correlated with each other hence they to be appropriately treated while modeling
6. poutcome\_unknown, month\_may, day and marital\_married feature have > 5 VIF suggesting a strong presence of multicollinearity
7. Data shows the presence of slight class imbalance issue which has to be addressed using appropriate class balancing techniques like over sampling

# Data Preparation based in Insights from EDA

## Missing Value Treatment

No missing value treatment performed on the data as procured data doesn’t contain any missing values

## Outlier Treatment

Of all features analyzed Age, Call duration, Balance and Campaign had relatively high degree of outliers and have been treated. We have applied very tight threshold for outlier treatment and detection as outlier can be key for prediction of subscription due to low class ratio. Following is the approach followed and distribution before and after outlier treatment

### Outlier Treatment for Age

Based on the visual observation of age, we see that there are very few customers with age > 60 and it is really rare for someone to opt for term plan with age greater than 60. Hence, we have capped age to 60 i.e., if age is greater than 60, we cap it to 60. Refer to Chart 9 below

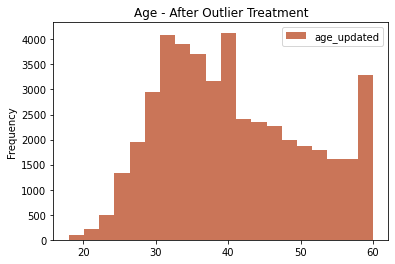
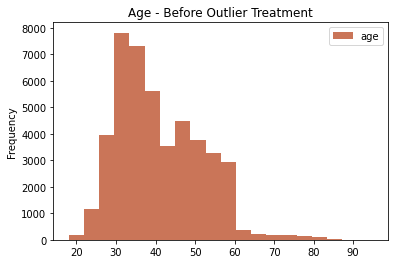


Chart 9: *Distribution of age before and after capping*

### Outlier Treatment for Call Duration

We observe that the call duration is ranging from 0 to 5000 sec in the dataset. A call duration of 2000 sec (33Min) is very high for a customer to interact with agents over phone. Hence, we treated anything above 2000 sec as outlier and capped it. Refer to Chart 10

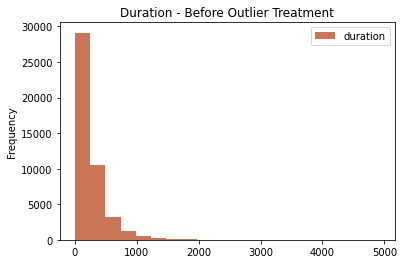
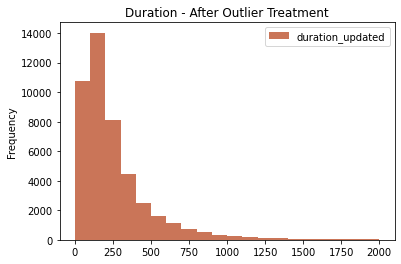
 

Chart 10: *Distribution of call duration before and after capping*

### Outlier Treatment for Balance

There are very few data point with balance greater than 40K. Hence, records with greater than 40K balance are capped at 40K. Refer to Chart 11 below for distribution before and after

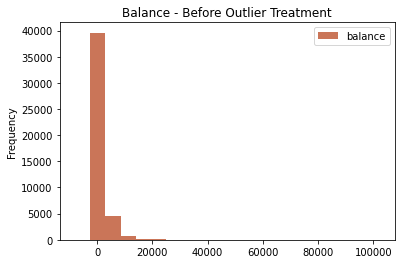
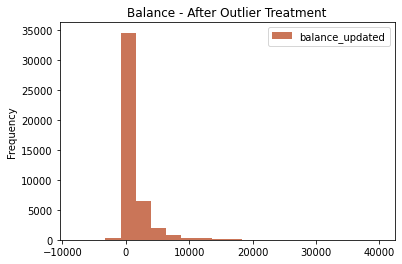
 

Chart 11: *Distribution of balance before and after capping*

### Outlier Treatment for Campaigns

Number of campaigns is ranging from 0 to 60 in the dataset, which means that a customer reached anywhere between 0 to 60 times is the current campaign. Based on distribution of data we few data point have campaign value >30. Hence, all data point with campaign value > 30 are capped at 30. Refer to Chart 12 below for distribution before and after

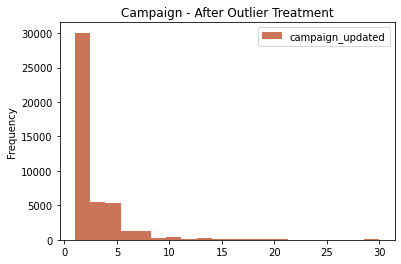
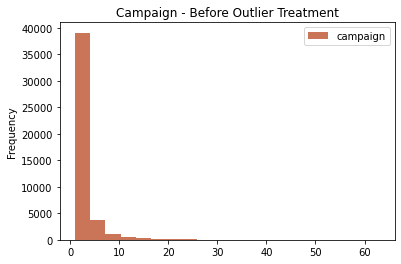


Chart 12: *Campaign value before and after capping*

## Train and Test Data Preparation

Data has been split into train and test in 4:1 ration through random sampling. Model will be trained on train dataset and performance will be evaluated on test dataset.

# Feature Engineering

## Addressing Multicollinearity

Of all the correlation observed Poutcomes is very highly correlated with Pdays both statistically and in business sense and hence Pdays is been excluded from the modeling. Though there are other correlated variables we are not excluding then as we are using tree-based models for model building and are robust to outliers and has no assumptions on multicollinearity

## Treating Categorical Features

Data has lot categorical features like Education, Job etc., which cannot be used as is. Categorical variables are dummy coded for making data model ready

# Model Development

## Decision Tree

As per the proposal decision tree model was developed as baseline model for current project. We have implemented and tuned decision trees using Grid Search and performance has been evaluated on Test and Train datasets. Following are the initial model results.

### Parameter Tuning

Decision tree model has been majorly tuned on max\_depth and max\_features parameter. Gridsearch with cross validation is used for parameter tuning. Max depth parameter is search in the range of 1 to 20 while max\_features is search in the range of 10 to 20. Best model was found at max depth = 19 and max\_features = 19. [Refer to Chart 13 and 14 below for understanding model performance with change in max\_feature and max\_depth]

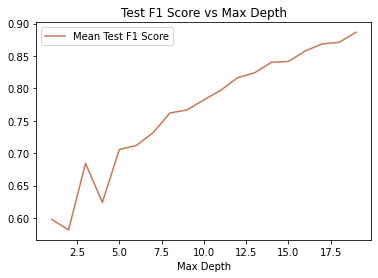


Chart 13: *Model F1 Score vs Max Depth*

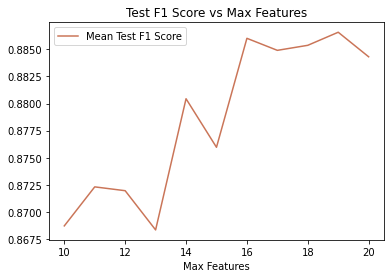
**

Chart 14: *Model F1 Score vs Max Features*

### Loss Metrics & Model Final Model Results

* Decision tree model has exhibited a good train and test accuracy of 97% and 87% respectively on support ratio of 50% (post class balancing)
* Though model is exhibiting a good accuracy of more than 90% but due to presence of class imbalance it is not good enough (especially in test)
  + When we look at recall score, 97 percent of the term plan subscribers are being identified during testing while only 53% are identified during testing. Which suggest that model is biased towards non-subscribers due to data size limitations [Refer to Chart 15 below]

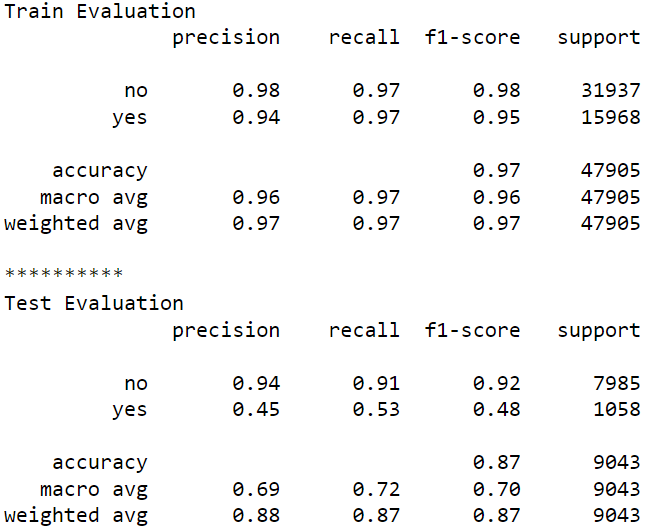


Chart 15: *Decision Tree – Train and Test Results*

## Random Forest

As per the initial proposal Random Forest model was developed as second model in ensemble of 3 model stack for current project. We have implemented and tuned random forest using Grid Search and performance has been evaluated on Test and Train datasets. Following are the initial model results.

### Parameter Tuning

Random Forest model has been majorly tuned on max\_depth and n\_estimators parameters. Gridsearch with cross validation is used for parameter tuning. Max depth parameter is search in the range of 10 to 20 while n\_ estimators is search in the range of 1 to 200. Best model was found at max depth = 15 and n\_ estimators = 101. [Refer to Chart 16 and 17 below for understanding model performance with change in n\_estimators and max\_depth]

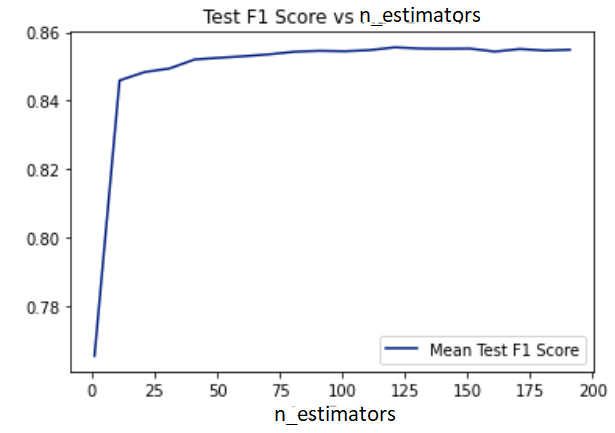


Chart 16: *Model F1 Score vs Number of Estimators*

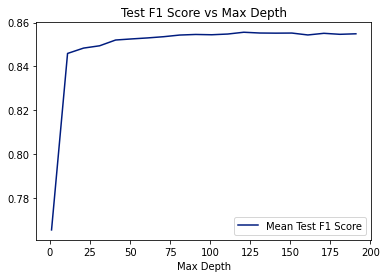


Chart 17: *Model F1 Score vs Max Depth*

### Loss Metrics & Model Results

* Random Forest model has exhibited a good train and test accuracy of 95% and 90% respectively on support ratio of 50% (post class balancing)
* Though model is exhibiting a good accuracy of more than 90% (in test) but exhibited lower precision and recall
  + It can be observed from recall scores that 93 percent of the term plan subscribers are being identified during testing while only 67% are identified during testing. [Refer to Chart 18 below]
* Random forest model show better performance than that of Decision tree in identifying possible customers purchasing term plan
  + Decision tree showed an recall score of 53% while random forest exhibited 67% (an increment of 14%)

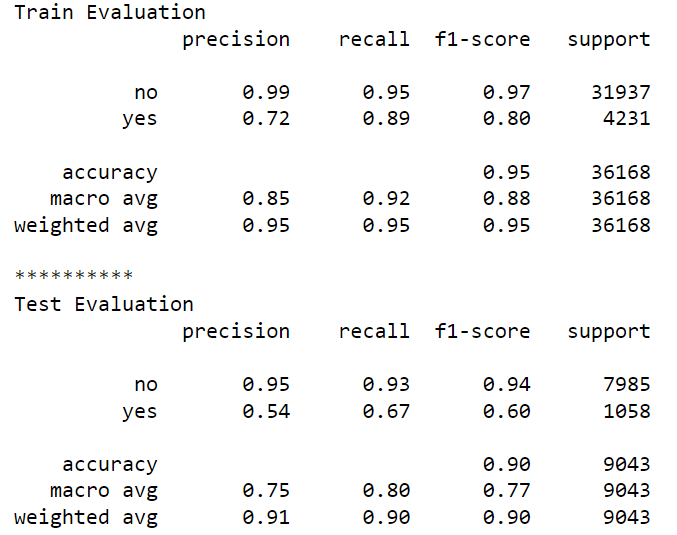


Chart 18: *Random Forest – Train and Test Results*

## XGboost

XGboost model was developed as third model in ensemble of 3 model stack for current project. We have implemented and tuned Xgboost using Grid Search and performance has been evaluated on Test and Train datasets. Following are the initial model results.

### Parameter Tuning

XGboost model has been majorly tuned on max\_depth and n\_estimators parameters. Gridsearch with cross validation is used for parameter tuning. Max depth parameter is search in the range of 10 to 20 while n\_ estimators is search in the range of 50 to 150. Best model was found at max depth = 12 and n\_ estimators = 130. [Refer to Chart 19 and 20 below for understanding model performance with change in n\_estimators and max\_depth]

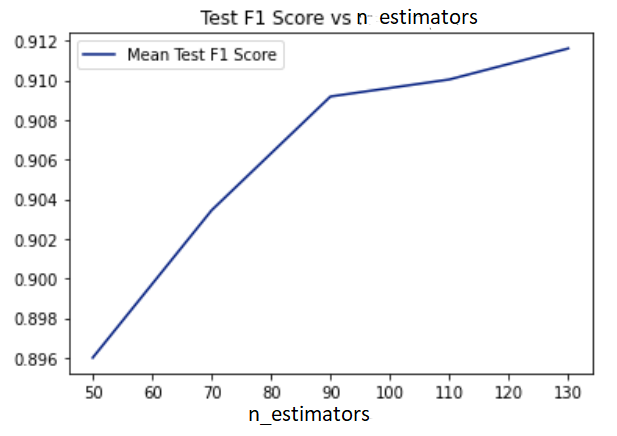


Chart 19: *Model F1 Score vs Number of Estimators*

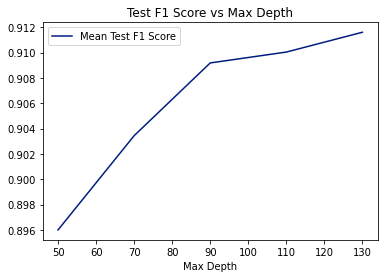


Chart 20: *Model F1 Score vs Max Depth*

### Loss Metrics & Model Final Model Results

* XGboost model has exhibited a good train and test accuracy of 95% and 89% respectively on support ratio of 50% (post class balancing)
* Though model is exhibiting a good accuracy of more than 95% (in test) but exhibited lower precision and recall
  + It can be observed from recall scores that, 92 percent of the term plan subscribers are being identified during testing while only 66% are identified during testing. [Refer to Chart 18 below]
* XGboost model showed similar performance to that of Random Forest in identifying

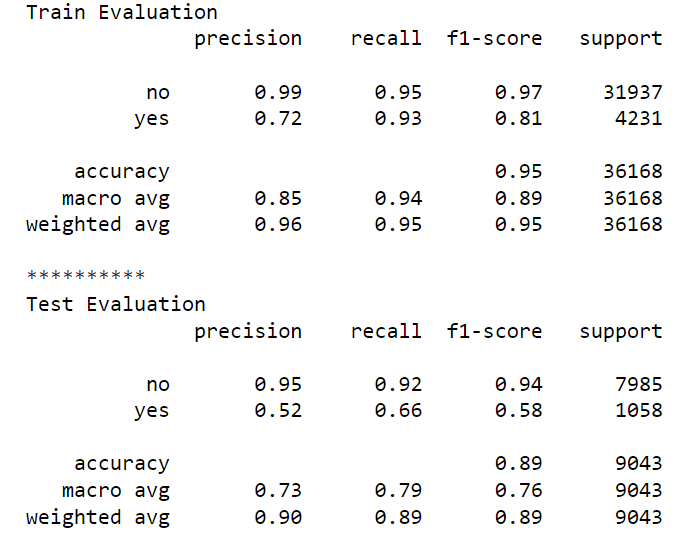


Chart 21: *XGboost – Train and Test Results*

## Ensemble of Decision Tree, Random Forest and XGboost

Initial idea of the proposal was to use the ensemble of all the three models but decision tree showed very different performance than that of random forest hence ensemble of Random Forest and Decision Tree is used as final model.

### Bias Validation

Train ensemble model showed good accuracy similar to that of Random Forest and XGboost. Train accuracy of ensemble model is 92%. [Refer to Chart 22 below]

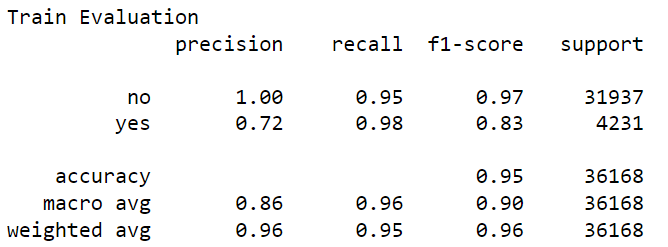


Chart 21: *Ensemble – Train Results*

Test ensemble model showed better performance that of Random Forest and XGboost. Train accuracy of ensemble model is 89% and positive class recall of 70%. [Refer to Chart 22 below]

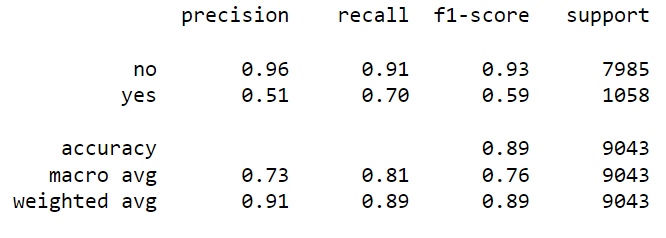


Chart 22: *Ensemble – Test Results*

No significant model bias has been observed during modeling as train and test results are more or less similar.

### Loss Metrics & Final Model Results

Final model evaluation on test data showed an recall of 70% which is better than random forest with 67% and XGboost 66%. Which basically means our model could successfully identify 70% of total customers who would opt for term plan. Also this 70% recall rate is observed at a precision of 51% which means for every one positive customers identified model is falsely identifying two customers who in actual wont opt for term plan.

### Cost of ML Model Development

Model Train Time: XGboost – 2Sec; Random Forest – 1Sec

Model Prediction Time: Negligible

# Conclusion

“Machine Learning Approaches for Marketing Campaign in Portuguese Banks” paper by Jennifer Alexandra; Kristina Pestaria Sinaga (2021) show the implementation of Decision Tree, Naïve Bayes, and Random Forest as well as clustering techniques for predicting. Afore mentioned paper evaluated each of these models in isolation our approach stands out in comparison with other in the through the usage of ensemble.

Ensemble model performed well, achieving a 92% train accuracy, and showed better performance than the individual Random Forest and XGboost models, with a 70% recall rate on identifying potential term plan customers. No significant model bias observed, and the model had a negligible prediction time. Precision of the model is a bit low, with the model falsely identifying two customers for every one positive customer identified as a potential term plan customer.

# References

**Reference Description:** Source for understanding Cost per Call in a typical call center

**Reference Link:** <https://www.liveagent.com/customer-support-glossary/cost-per-call/>

**Reference Description:** Research paper showing how ensemble of machine learning model can be used to achieve better prediction accuracies

**Reference Link:** Stefan Lessmann a, Johannes Haupt a, Kristof Coussement b, Koen W. De Bock c *Targeting customers for profit: An ensemble learning framework to support marketing decision-making* [*https://doi.org/10.1016/j.ins.2019.05.027*](https://doi.org/10.1016/j.ins.2019.05.027)

**Reference Description:** Research paper on different techniques for addressing class imbalances

**Reference Link:** Justin M. Johnson & Taghi M. Khoshgoftaar *Survey on deep learning with class imbalance,* Journal of Big Data volume 6, Article number: 27 (2019)

**Reference Description:** Resource explaining the technique and need for dummy code for machine learning model development

**Reference Link:** <https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/#:~:text=Dummy%20Coding%3A%20Dummy%20coding%20is,absence%20is%20represented%20by%200>.

**Reference Description:** Resource capturing implementation of decision tree in Python

**Reference Link:** https://www.geeksforgeeks.org/decision-tree-implementation-python/