Predict Customer Responses using Marketing Campaigns Data

By: Li Li

April 2022

SpringBoard Data Science Intense Training Program

1. Introduction

1.1Problem statement

Customer personality analysis is an interesting topic for companies. Marketing campaigns aim to encourage customers to act in some manner. The business would like to better understand if customers would respond to the marketing campaign. It is a very common problem many retail businesses are facing to.

This project was inspired by a Kaggle competition. The direct link to the data is as follows: (https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis). Records of 2240 between the years 2012 and 2014 were analyzed alongside customer personality and their spending behaviors.

1.2 Questions of interest

This is a binary classification problem. To solve the problem, we'll answer the questions as follows:

- Who are the customers more responsive to the last marketing campaign?
- What is the customers' demographic information, such as age?
- Which factors, such as income and shopping behaviors, significantly influence the customers' responses to the campaign?

1.3 Objective

This will be done through exploratory data analysis, plotting visualizations, and creating a model that allows us to predict the customers' responses to a campaign, considering a specific promotion. The main goal is to provide insights for campaign strategies and even leads to more conscious decision.

Five different classifiers are trained on the training data and used to predict if a customer would accept the offer in the last campaign. For each technique and its corresponding model, I will assess its performance by comparing the predicted output with the actual result. Finally, all models will be compared to each other, and we will find out which model works the best.

2. Data Wrangling

2.1 Data loading

The first step of analysis consists of reading and loading the data into a pandas dataframe.

The dataset contains 2240 records and 29 features, which can be classified into 4 groups:

(1) **Customer information**

- Year_Birth: Customer's birth year
- Education: Customer's education level
- Marital_Status: Customer's marital status
- Income: Customer's yearly household income
- Kidhome: Number of children in customer's household
- Teenhome: Number of teenagers in customer's household
- Dt_Customer: Date of customer's enrollment with the company
- Complain: if the customer complained in the last 2 years

(2) **Amount spent on products in last 2 years**

- MntWines: Amount spent on wine
- MntFruits: Amount spent on fruits
- MntMeatProducts: Amount spent on meat
- MntFishProducts: Amount spent on fish
- MntSweetProducts: Amount spent on sweets
- MntGoldProds: Amount spent on gold

(3) **Places where customers made the purchase**

- NumWebPurchases: Number of purchases made through the company's website
- NumCatalogPurchases: Number of purchases made using a catalog
- NumStorePurchases: Number of purchases made directly in stores
- NumWebVisitsMonth: Number of visits to the company's website in the last month

(4) **Promotion activities response**

- NumDealsPurchases: Number of purchases made with a discount
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Response: 1 if the customer accepted the offer in the last campaign, 0 otherwise

2.2 Data cleaning

Use **pandas.DataFrame.info()** method to print a summary of the data frame.

- **Missing values**

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   MntWines             2240 non-null   int64
 10  MntFruits            2240 non-null   int64
 11  MntMeatProducts      2240 non-null   int64
 12  MntFishProducts      2240 non-null   int64
 13  MntSweetProducts     2240 non-null   int64
 14  MntGoldProds         2240 non-null   int64
 15  NumDealsPurchases    2240 non-null   int64
 16  NumWebPurchases      2240 non-null   int64
 17  NumCatalogPurchases  2240 non-null   int64
 18  NumStorePurchases    2240 non-null   int64
 19  NumWebVisitsMonth    2240 non-null   int64
 20  AcceptedCmp3         2240 non-null   int64
 21  AcceptedCmp4         2240 non-null   int64
 22  AcceptedCmp5         2240 non-null   int64
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Complain             2240 non-null   int64
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

There are no null values on the dataset. We observe that the column 'Income' has 24 missing values. The missing values are filled in with the statistical value (median). Based on the visualization, incomes greater than 110,000 are considered 'outliers' in this dataset and removed.

- **Drop the columns of no use**

Two columns of 'Z_CostContact' and 'Z_Revenue' were dropped as they don't have the data description.

The Id column is not useful to explain whether the customer would respond. Therefore, this column is dropped from the dataset.

- **Make the categories consistent within the features**
  'Education' and 'Marital_status': organize the various class names within the features into two classes
- **Rename the columns**
  'Response' is renamed as 'AcceptedCmp6'

2.3 Features engineering

- 'Age' is engineered from the 'Year_Birth' column to gain more information.
- 'Dt_Customer' are converted to DateTime objects instead of objects. By subtracting it from the collection year 2015, 'Days_Enrolled' is generated, the number of days since the customers enrolled with the company.
- 'Children' is generated by summing 'Kidhome' and 'Teenhome'
- 'TotalExpenses' is generated by aggregating all the expenses of 'Wines', 'Fruits', 'Meat', 'Fish', 'Sweets', and 'Gold' per customer.
- 'TotalAcceptedCmp5' is generated by aggregating the total number of promotion acceptance for each customer (champaign 1-5).
- 'TotalNumPurchases' is generated by aggregating all the purchases of 'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases', 'NumDealsPurchases'

Save the cleaned dataset as a new CSV file.

3. Exploratory Data Analysis (EDA)

Exploratory data analysis consists of analyzing the main characteristics of a data set using **visualization methods** and **summary statistics**.

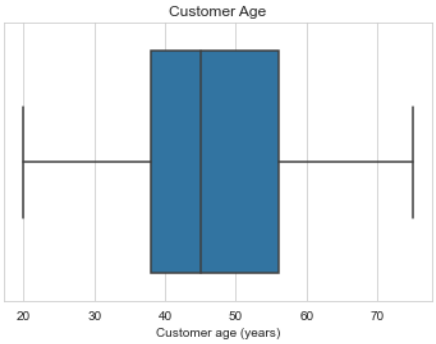
The following questions guided my data exploratory analysis:

- How do the customer's responses vary with income, the days enrolled with the company) and their expenses?
- Are there any significant differences among groups? Does a specific group respond more than others?
- Is there any consistent relationship between the total amount of expenses of the customers with their responses to the marketing campaign? We should expect this.

3.1 Univariate analysis

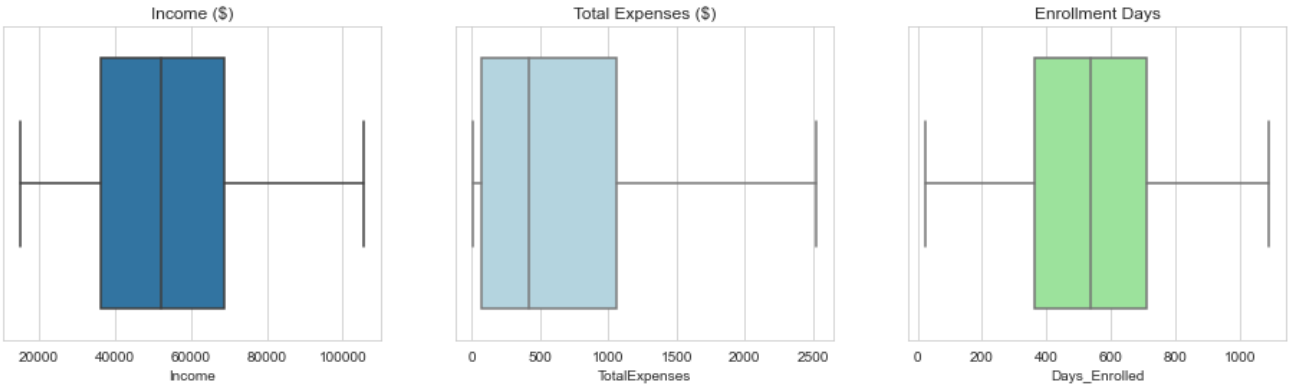
Univariate analysis is done by EDA to determine if the features should be further modified or engineered.

- **Investigating customer demographics (age)**



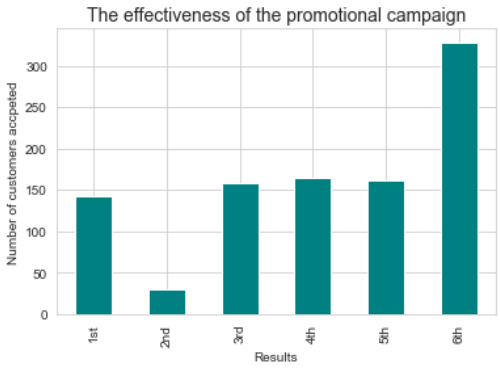
Most customers in this data are mid-aged between 35~55.

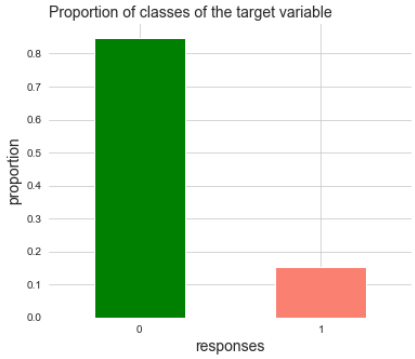
- **Investigating customer Income, total expenses, and days enrolled**



Most people in this dataset are royal customers enrolled between 360 ~ 720 days. And most of them have income between 35000~70000 and total expenses between 100~1000.

3.2 Investigating the target variable: "AcceptedCmp6"





The effectiveness of the 6 marketing campaigns (left)

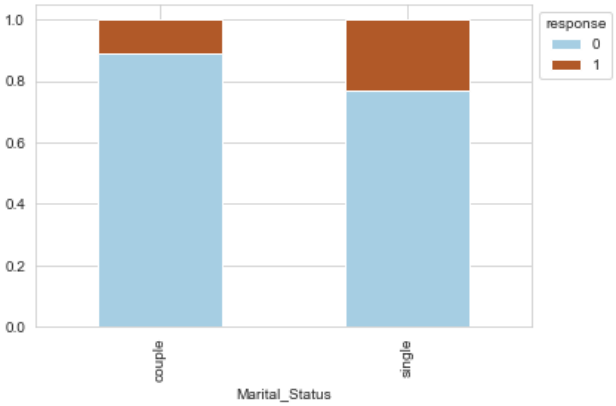
A severe class imbalance is seen in the target variable(right)

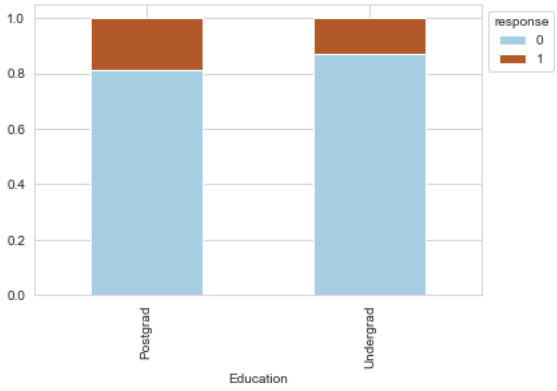
Accepted: 1 328
Not_Accepted: 0 1824

3.3 Investigating the categorical variables

A normalized stacked percentage bar chart is used to analyze the influence of the independent categorical variable (Marital_status and Education) on the target variable, showing the percentage of Responses for each category of the feature.

- **"AcceptedCmp6" v**ariation with “Marital Status”





On average, single people accept the last campaign more often.

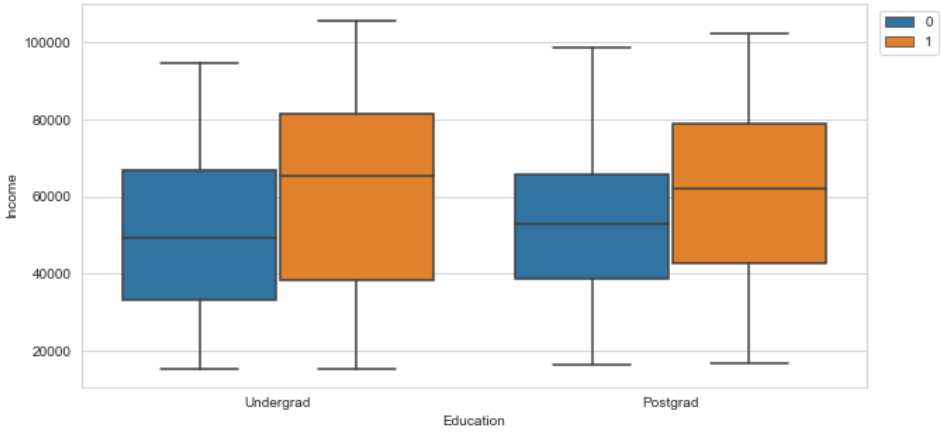
- **"AcceptedCmp6" v**ariation with “Education”

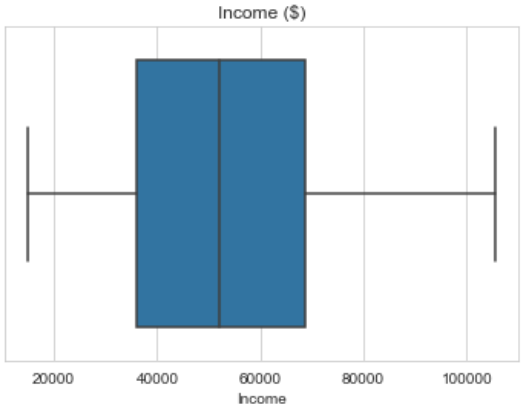
On average, customers with postgraduate accept the last campaign more often.

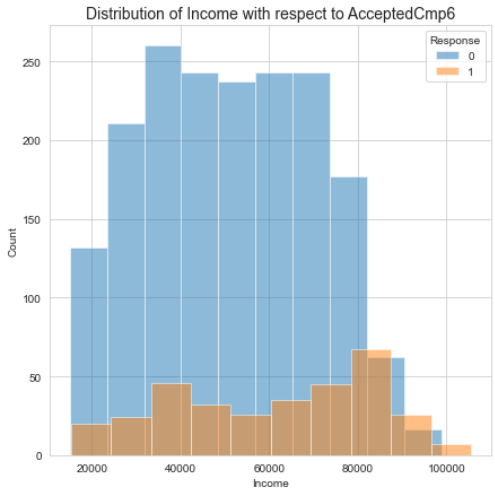
3.4 Investigating the numeric variables

On the other hand, I use **histograms** and **boxplots** to evaluate the **influence of each independent numeric variable on the target variable**.

- **"AcceptedCmp6" v**ariation with '**Income**'

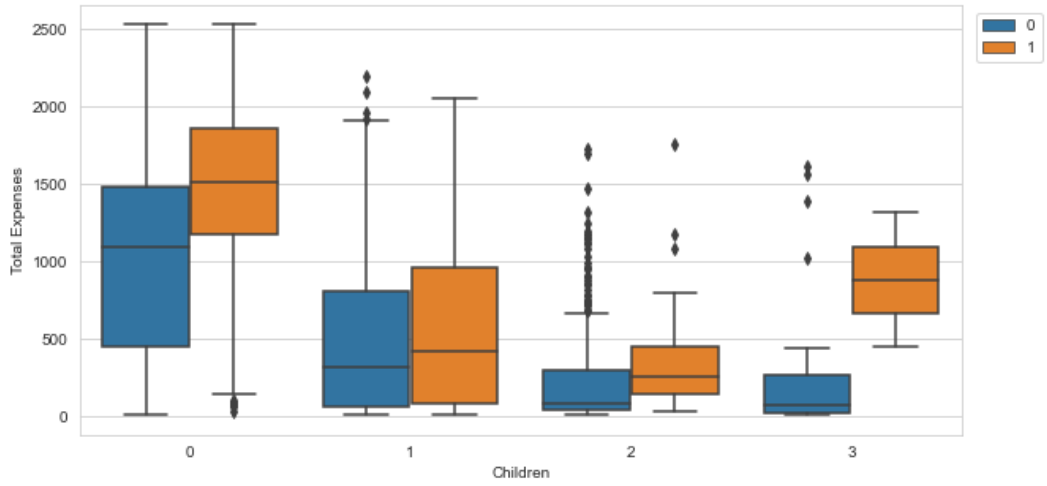


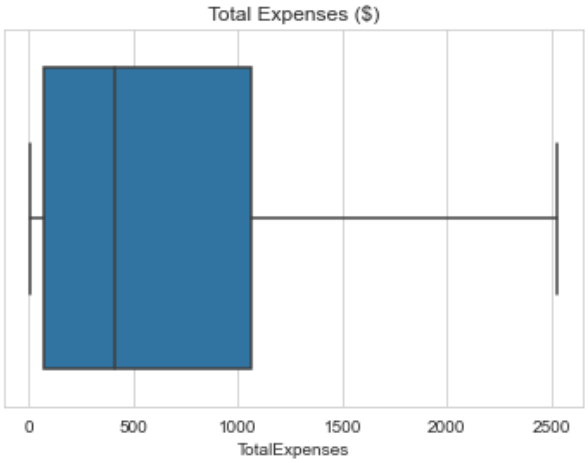


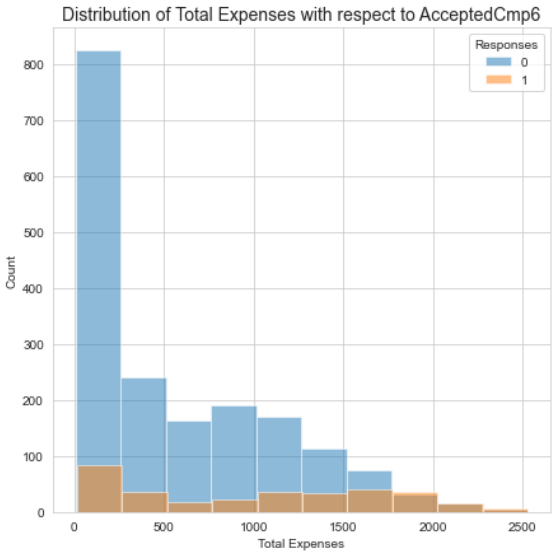


The response rate tends to be larger when incomes are higher.
Customers with higher income are more active in the last campaign compared to those with less income.

- **"AcceptedCmp6" v**ariation with "**TotalExpense**"







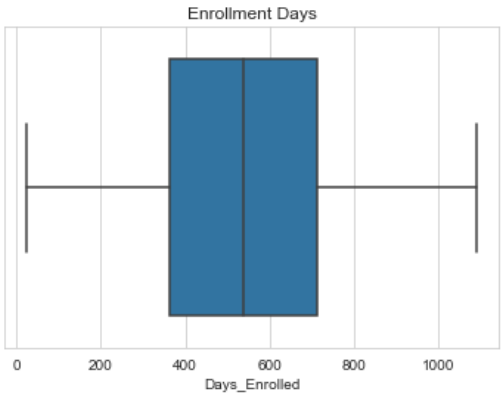
Most customers are spent less than 1000.

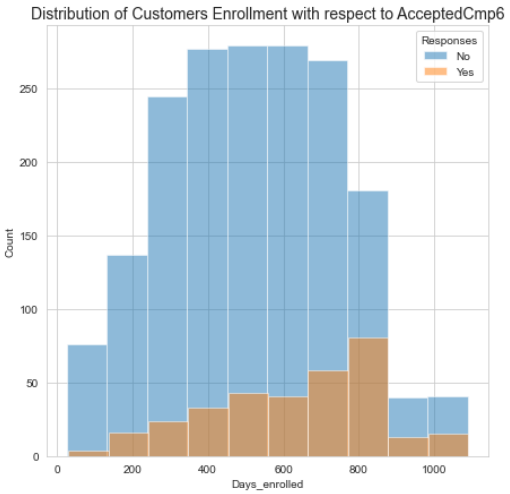
The response rate is slightly higher with more total expenses.

- **"AcceptedCmp6" v**ariation with "**Days_Enrolled**"

Most customers are enrolled between 360-720 days.

The response rate is higher with more enrollment days.



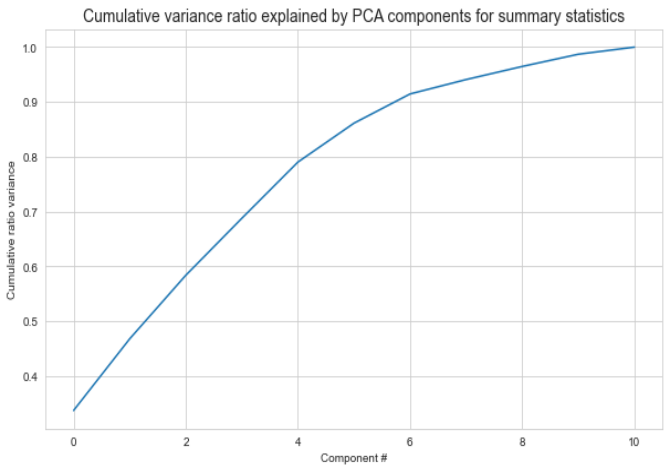


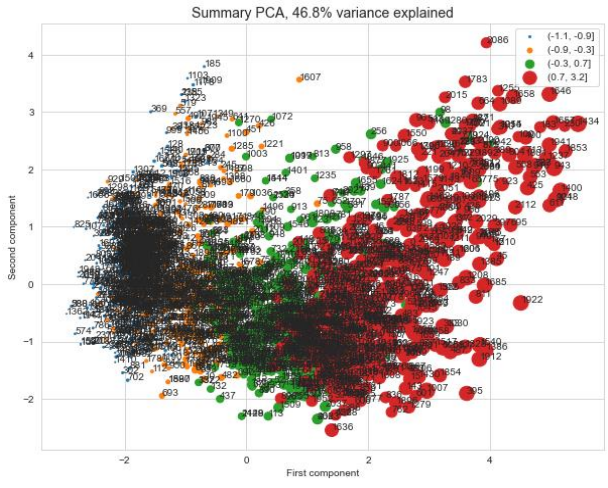
3.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique that allows us to view the data from the most informative viewpoint. Any features in this data that were less than 85% correlated were fed into PCA analysis.

The first two components in the new dimension seem to account for about 50% of the variance of the data, and the first five for over 80%.

The red points represent the upper quartile of 'TotalExpense' and they spread across the first dimension (>0). There's also a spread of the other quartiles as well.





4. Data Preprocessing

Use Scikit-Learn label encoding in categorical variables as the order of classes within the feature does not matter. Two columns are encoded: Education and Marital_status.

5. Machine Learning Modeling

In this session, 5 different machine learning models are trained and tuned to achieve the optimized results. Based on the metric 'roc_auc" scores, the random forest classifier was selected as the best model.

5.1 Splitting data into training and testing sets

To fit the model on the data, I first ensured that all categorical features were converted to dummy variables or hot encoded. I then split the data into training (70%) and testing sets (30%). The model was then allowed to learn from the training set, make predictions, and report model performance on the test set.

5.2 Setting a baseline model

In this binary classification problem, I first use a simple classifier as a baseline to evaluate the performance of a machine learning model. The rate of customers that did not respond to the marketing campaign (most frequent class) can be used as a baseline to evaluate the quality of the models generated. The model to be considered for future predictions should be outperforming the baseline model.

```
              precision    recall  f1-score   support

           0       1.00      0.85      0.92       646
           1       0.00      0.00      0.00         0

    accuracy                           0.85       646
   macro avg       0.50      0.42      0.46       646
weighted avg       1.00      0.85      0.92       646

 [[547  99]
 [  0   0]]
```

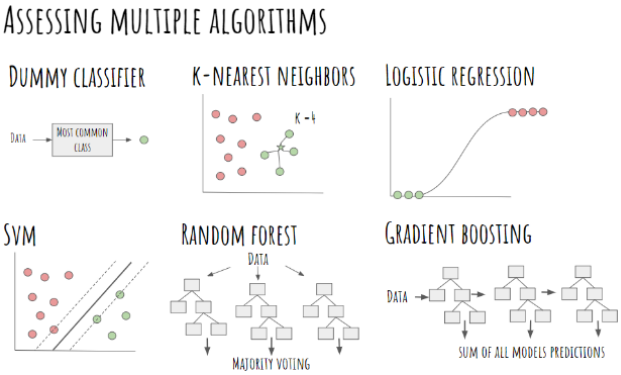
While the accuracy is high, the severe class imbalance in the target variable makes the precision, recall, f1-score, and auc_roc score more important metrics of a good model. Thus, this baseline model performs poorly.

5.3 Assessing multiple algorithms

Generally, we need to evaluate a set of algorithms and select the final model that provides the best performance.

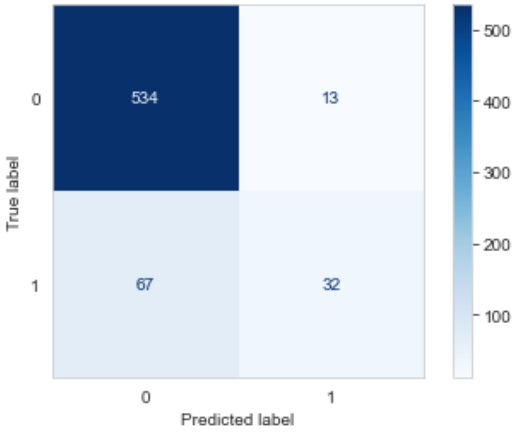
In this project, I compare 5 different popular classifiers for solving binary classification problem:

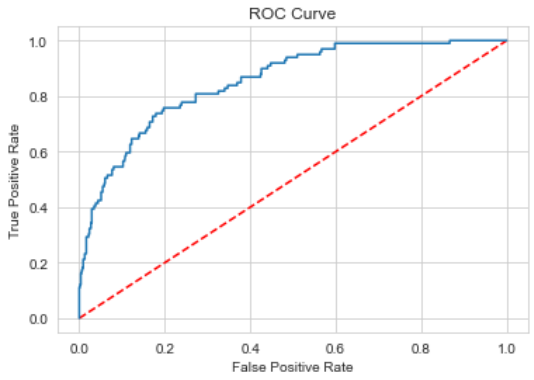
- Logistic Regression
- K Nearest Neighbors (KNN)
- Random Forest
- Gradient boosting
- XGBoost



All these models can be built feasibly using the algorithms by the Scikit-learn package. Only for the XGBoost model, we need to use the xgboost package.

- Logistic Regression

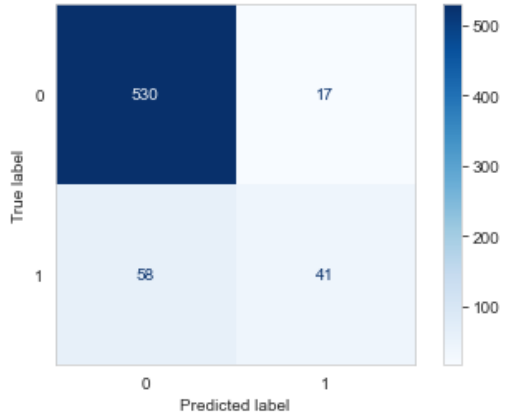


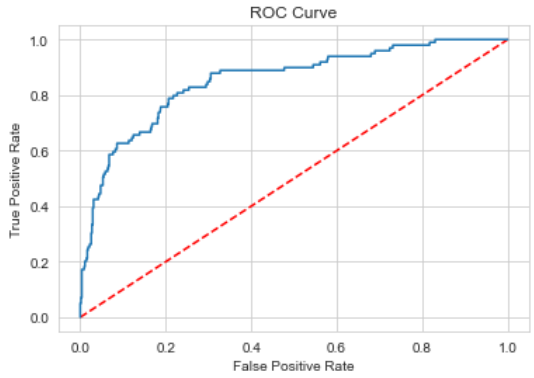


Though there is a significant improvement in precision (number of correct predictions for a given predicted class/ total number of predictions in the same class), recall (number of correct predictions for a given class / total actual number in class) still suffers from this model, given the decision threshold is 0.5. The auc_roc score is 85.3%.

- Random Forest

The random forest uses randomly selected features to construct multiple models. I used a randomized grid search of values for 'n_estimators', and 'max_depth' around the base classifier model parameters. ‘max_depth’ to be ‘10’ which means we are allowing the tree to split ten times. I performed 5 fold cross-validation loop to find the best values for these parameters. This is the best performing model, with a good accuracy rate, decent auc_roc of 85.3%, and values for important metrics across the board.





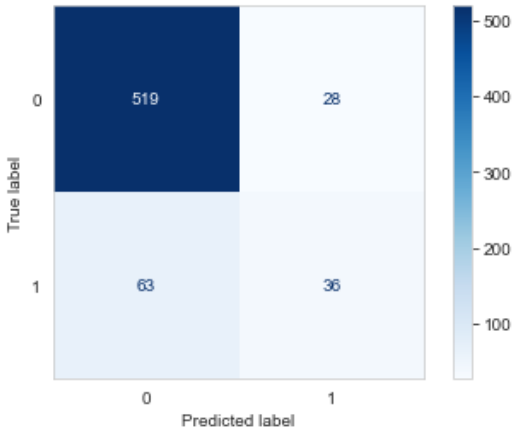
- Gradient boosting classifier

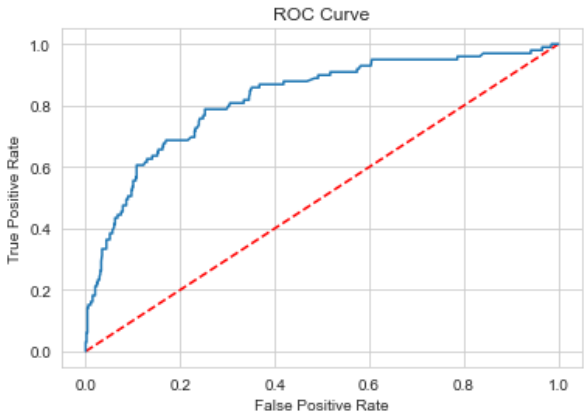
Before training the features were scaled for gradient boosting. Six different classifiers were trained and tested on training and validation data, with each classifier's learning rate parameter set to a unique value. Based on the accuracy scores and corresponding learning rate values, the learning rate of 0.75 lends the highest accuracy scores on both training and validation sets, this value is fed into the gradient boosting classifier.

```
[[519  28]
[ 63  36]]
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       547
           1       0.56      0.36      0.44        99

    accuracy                           0.86       646
   macro avg       0.73      0.66      0.68       646
weighted avg       0.84      0.86      0.85       646
```





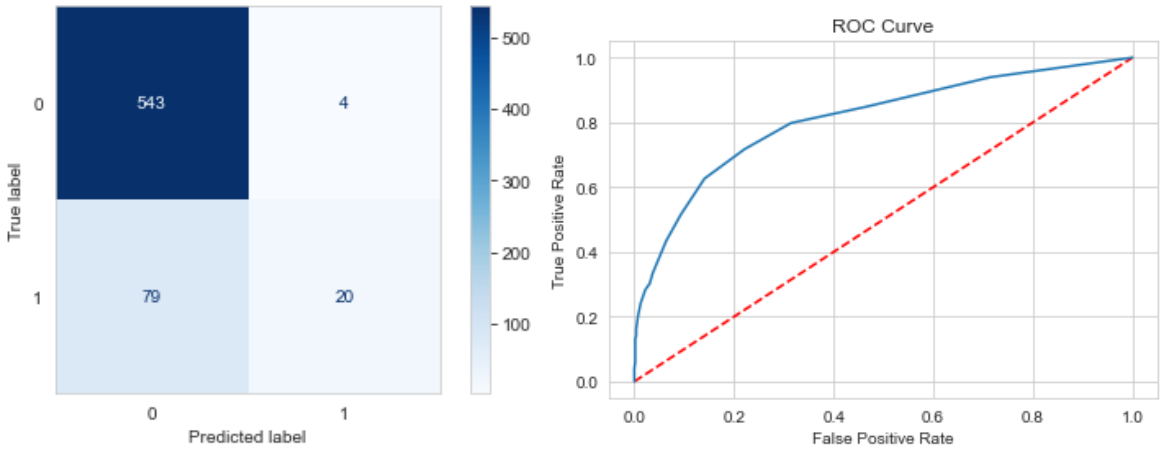
- K Nearest Neighbors (KNN)

The optimal value of the ‘n_neighbors k’ is chosen by the elbow method.

```
[[543   4]
 [ 79  20]]
              precision    recall  f1-score   support

           0       0.87      0.99      0.93       547
           1       0.83      0.20      0.33        99

    accuracy                           0.87       646
   macro avg       0.85      0.60      0.63       646
weighted avg       0.87      0.87      0.84       646
```



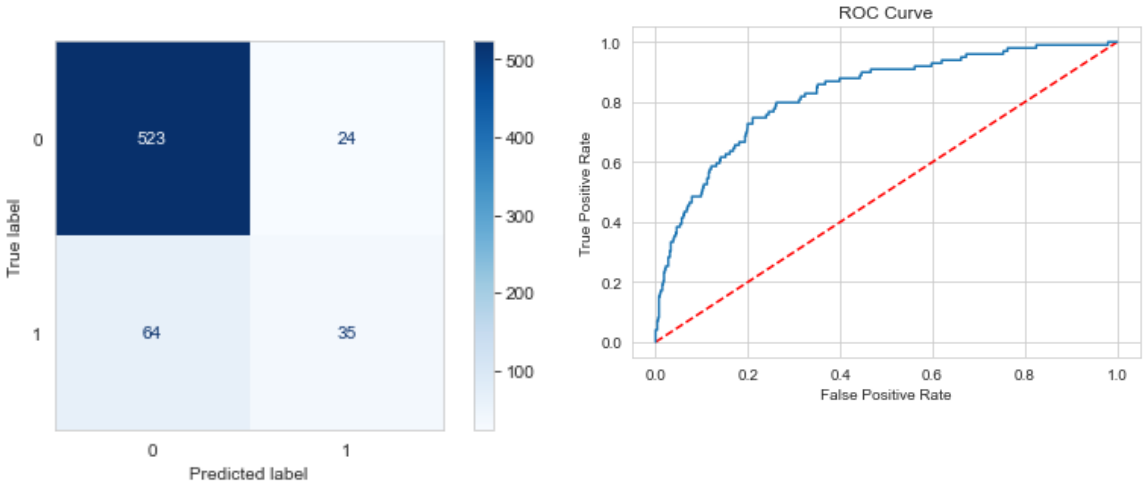
- XGBoost

My final model is the XGBoost model which was built using the ‘XGBClassifier’ algorithm provided by the xgboost package. objective='binary:logistic' is used to return probability rather than decisions. The ‘max_depth’ to be 3 is chosen and finally, the model is fitted and the predicted values are evaluated.

```
[[523  24]
 [ 64  35]]
              precision    recall  f1-score   support

           0       0.89      0.96      0.92       547
           1       0.59      0.35      0.44        99

    accuracy                           0.86       646
   macro avg       0.74      0.65      0.68       646
weighted avg       0.85      0.86      0.85       646
```

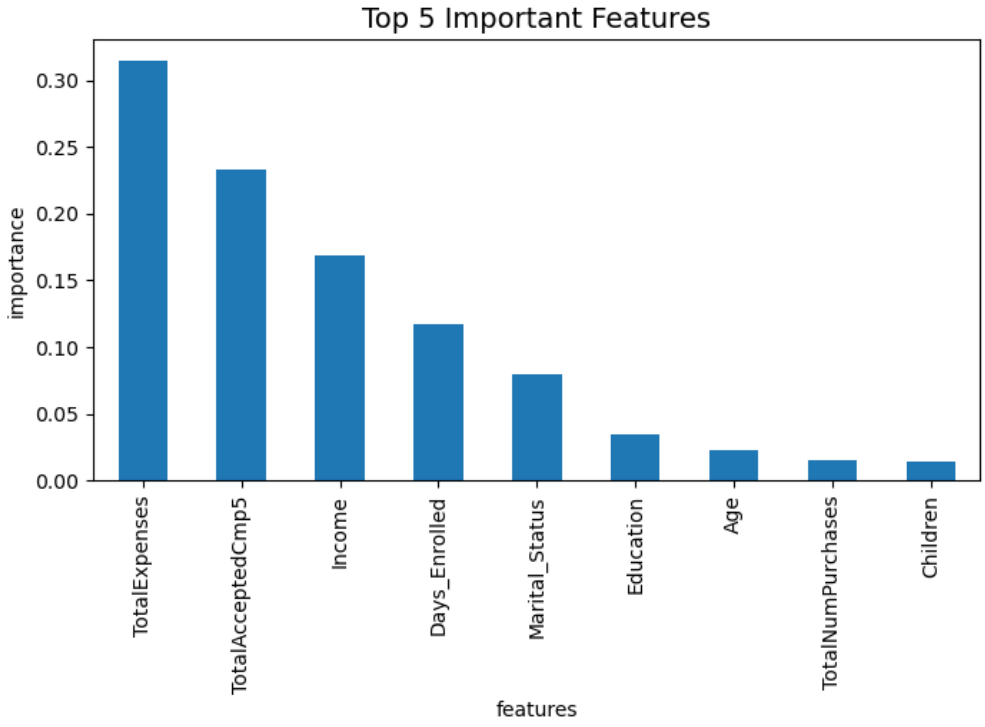


5.4 Evaluation of the model performance and Model selection

Each machine learning model conducted hyperparameter tuning and cross-validation to improve model accuracy.

While comparing the confusion matrix and auc_roc of all the models, the random forest model has performed a very good job of classifying the accepted customers from the non-accepted customers followed by the Logistic regression classifier. So, the most appropriate model which can be used for this case is the random forest classifier.

5.5 Importance Features



The dominant top five features are:

- 'TotalExpenses'
- 'TotalAcceptedCmp5' - total accepted response except the last champaign
- 'Income'
- 'Days_Enrolled'
- 'Marital Status'

6. Summary

- Label encoding is used to convert categorical variables into binary numeric variables.
- After hyperparameter tuning, the models are slightly improved.
- The best classifier that works well on this dataset is the random forest classifier which achieved a strong auc_roc score of 85.3%. Since the severe imbalance is observed in the target variable, the auc_roc score is the best metric to be used to evaluate the classification model performance given any decision threshold.

7. Future direction

- Additional data on the minority class (accepted) could improve the accuracy of the model.
- Explore more models, like SVM, math, and statistics behind this.