Bank Marketing Effectiveness Prediction

Ritik Gupta
Data science trainee,
AlmaBetter

Abstract:

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. Often, more than one contact to the same client was required, in order to access if the bank term deposit would be subscribed or not.
This project will enable the bank to develop a more granular understanding of its customer base, predict customers' response to its telemarketing campaign and establish a target customer profile for future marketing plans.

Keywords: Marketing, Effectiveness prediction, bank, classification, machine learning.

Problem Statement:

Bank needs to spend a lot for marketing operations, targeting right set of client and multiple campaigns, but sometimes its not possible to find out who can be potential subscriber for product, despite having a lot of data about clients, previous subscribers and different campaigns, so bank needs to find out an optimal model that predict whether a client can be potential subscriber of product or not, so that they focus on micro targeting and in similar niche for better results.

Dataset Information

- Instances: 45,211
- Attributes: 17

Attributes information:

The dataset contains features like:

- **age**: age of client
- **job**: profession of client
- **marital**: client marital status
- **education**: educational qualifications
- **default**: has credit in default ?
- **housing**: has house loan ?
- **loan**: has personal loan ?
- **contact**: communication type
- **month**: last contact month
- **day**: last contact day of month (numerical)
- **duration**: last contact duration (in seconds)
- **campaign**: number of contacts performed during this campaign
- **pdays**: days passed after last contact.
- **previous**: number of contacts performed before this campaign.
- **Poutcome**: outcome of previous marketing campaign.

Target Variable :

- **'Y'**: whether client will subscribe term deposit (Yes) or not subscribe (No)

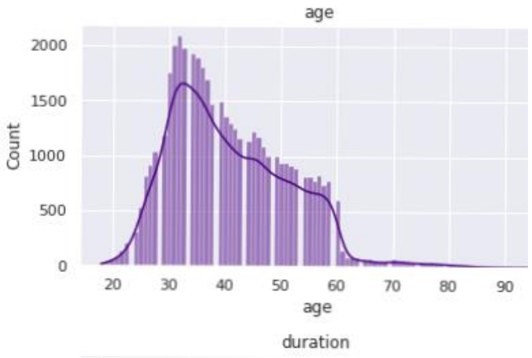
Steps involved:

- **Hands-on Data**
  Statistical analysis and summary of data. checking and correcting the dtypes for further analysis.

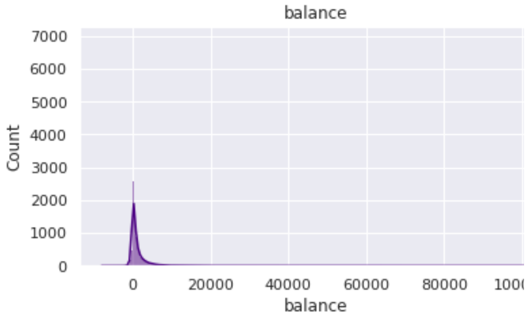
- **Exploratory Data Analysis**
  We started by EDA from comparing our independent variables to the target variable subscription rate and this process help us to identify hidden relations between our data and give some interesting insights on the data.

- **Data Distribution**

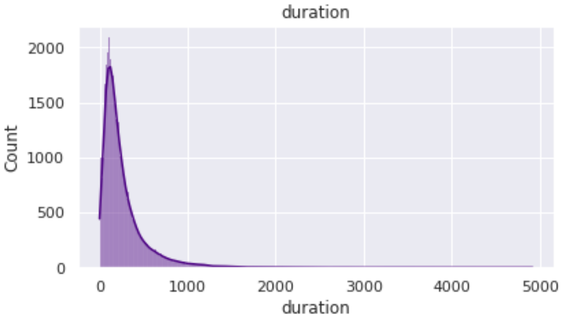
➢ Age



➢ Balance



➢ Duration



- **Null/Missing values Treatment**
  This data has no missing values but has some value which is stated as ‘unknown’ and we have converted them to ‘other’ as it makes some sense than unknown.

- **Outlier Treatment**
  Balance feature has some outliers and we have removed outliers with help of IQR range and a threshold values.

- **Feature Engineering**
  Converted days number into week number, defined quarters from month number, label and target encoding features.

- **Correlation and Feature importance**
  Not others but duration of call feature showed a high relation with target variable but since duration of call can only find after a campaign and we build this to predict before campaign so, to build realistic model we have to remove this feature.

- **Splitting Data**
  We have to split the data in train and test sets, with stratify because there's an imbalance in class labels.

- **Handling Imbalance Data**
  Majority class is 9 times higher than minority class and balancing the data is the biggest challenge, and we choose 2 approach to move further

  - **SMOTEtomek (Oversampling) technique**
  - **Nearmiss (Under sampling) technique**

- **Scaling Data**
  After the handling imbalance, we have to scale our variables to input in model, here we used standardization technique to scale.

- **Fitting Models**
  For modelling we tried various classification algorithms:
  - **Oversampling**:-
  1. Random Forest Classifier
  2. Naïve Bayes Classifier

  - **Under sampling**:-
  3. K Nearest Neighbor
  4. XG Boost
  5. Support Vector Machine
  6. Neural Network Classifier

- **Training Models with Hyperparameter Tuning**
  Tuning the hyperparameters of respective algorithms is necessary for low error and high accuracy and to avoid overfitting, mostly in tree based algorithms.

Handle Imbalanced Data:

This problem is binary classification between subscribed or not subscribed and our class has 88.3% unsubscribed and 11.7% is subscribed, so from the model perspective data needs a balance of both classes. And to balance data we used two techniques.

1. SMOTEtomek Oversampling:- this method combines the SMOTE ability to generate synthetic data for minority class and Tomek Links ability to remove the data that are identified as Tomek links from the majority class (that is, samples of data from the majority class that is closest with the minority class data).

2. Nearmiss Under sampling:
The algorithm works by looking at the class distribution and randomly eliminating samples from the larger class. When two points belonging to different classes are very close to each other in the distribution, this algorithm eliminates the datapoint of the larger class thereby trying to balance the distribution.

Hyper-parameter tuning:

We used Grid Search CV, and Random search CV but with Halving Technique of scikit lean library of python, which is experimental feature to reduce hyperparameter tuning computational time.

1. **Halving Grid Search CV-**
   A new class of successive Halving, where training is performed on the subsets of data, rather than on all the data. The worst performing data are filtered out by training them on a small subset of data. After N number iterations select the best data/candidates leading to a faster evaluation time.

2. **Halving Random Search CV-**
   In this concept training is performed on the subsets of data rather then whole data and on the subsets, random search cv is using random combinations of hyperparameters given prior in dictionary to find best combinations of parameters.

Evaluation Metrics:

We evaluated model on basis of-

- **ROC_AUC score-**

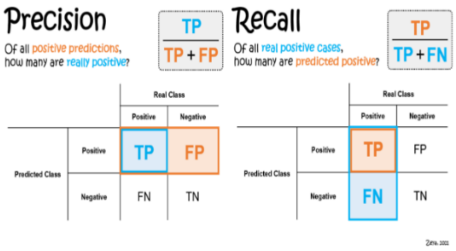
ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. It gets influenced by outliers.

- **F1 Score-**

It is used to evaluate binary classification systems, which classify examples into positive or negative; The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

- **Recall (for class-1)-**

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.


Precision
Of all positive predictions, how many are really positive?
TP / TP + FP
Recall
Of all real positive cases, how many are predicted positive?
TP / TP + FN
Real Class
Positive
Negative
Predicted Class
Positive
Negative
TP
FP
FN
TN
Zeya, 2021

- **Matthews correlation coefficient -**

The Matthews correlation coefficient (MCC), is a reliable statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally both to the size of positive elements and the size of negative elements in the dataset.

Algorithms:

These algorithms are used to build different predictive models, and compared their scores.

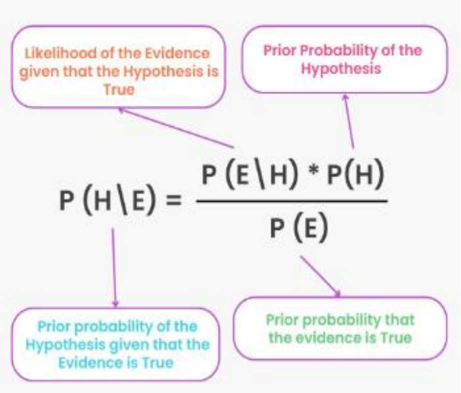
- ### Oversample data:-

1. Random Forest Classifier:

Random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms, this algorithm establishes the outcome based on the predictions of the decision trees.

2. Naive Bayes Classifier:

The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. therefore they are considered as naive. you can derive probability models by using Bayes' theorem, depending on the nature of the probability model, you can train the Naive Bayes algorithm in a supervised learning setting.


Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
P (H\E) = P (E\H) * P(H) / P (E)
Prior probability of the Hypothesis given that the Evidence is True
Prior probability that the evidence is True

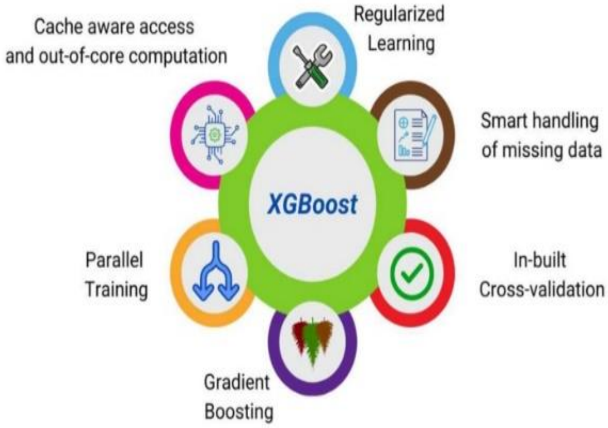
- ### Under sample data:-

1 K Neighbor Classifier:

This is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

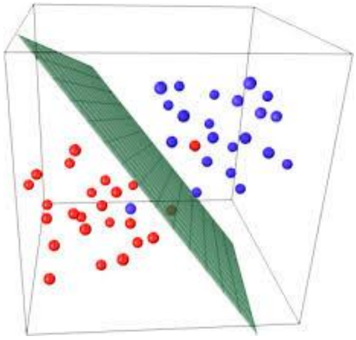
2 XG Boost:

This falls under the category of Boosting techniques in Ensemble Learning, it consists of a collection of predictors which are multiple models to provide better prediction accuracy. In Boosting technique the errors made by previous models are tried to be corrected by succeeding models by adding some weights to the models. this is a faster algorithm compared to others because of its parallel and distributed computing.


Cache aware access and out-of-core computation
Regularized Learning
Smart handling of missing data
XGBoost
Parallel Training
In-built Cross-validation
Gradient Boosting

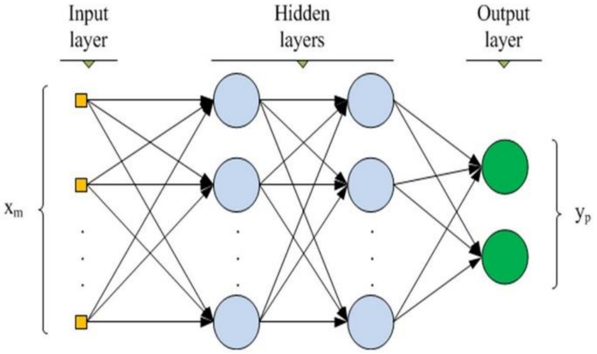
3 Support Vector Classifier:

Here, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being value of a particular coordinate. Then, perform classification by finding the hyper-plane that differentiates the two classes very well, Support Vectors are simply the coordinates of individual observation.

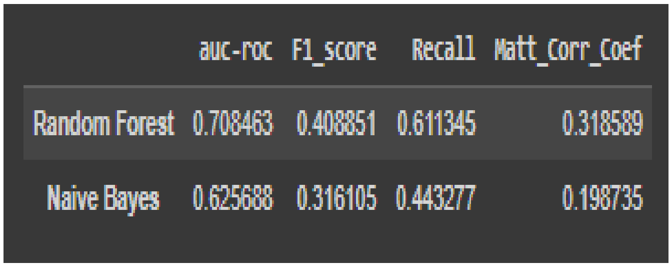


4 Neural Network Classifier:

Neural networks are loosely representative of the human brain learning. Artificial Neural Network consists of Neurons which in turn are responsible for creating layers. the output from each layer is passed on to the next layer. There are different nonlinear activation functions to each layer, which helps in the learning process and the output of each layer. The output layer is also known as terminal neurons, The weights associated with the neurons and which are responsible for the overall predictions are updated on each epoch. The learning rate is optimised using various optimisers. Each Neural Network is provided with a cost function which is minimised as the learning continues. The best weights are then used on which the cost function is giving the best results.



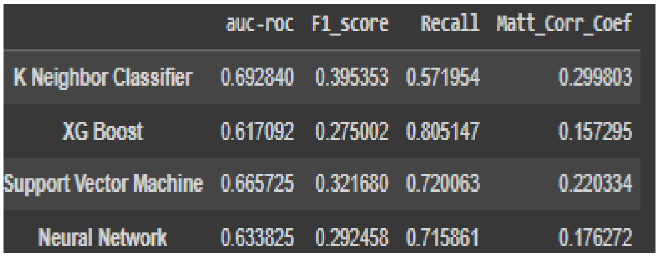
Model Scores (over-sampled):



	auc-roc	F1_score	Recall	Matt_Corr_Coef
Random Forest	0.708463	0.408851	0.611345	0.318589
Naive Bayes	0.625688	0.316105	0.443277	0.198735

When we used oversampling for balancing class labels, Random forest gave us 70% roc_auc score and 61% recall, but here we more focused towards recall because of business use case we have to reduce False Negetives and compromise on False positives.

Model Scores (undersampled):



	auc-roc	F1_score	Recall	Matt_Corr_Coef
K Neighbor Classifier	0.692840	0.395353	0.571954	0.299803
XG Boost	0.617092	0.275002	0.805147	0.157295
Support Vector Machine	0.665725	0.321680	0.720063	0.220334
Neural Network	0.633825	0.292458	0.715861	0.176272

Undersampled data has given us high recall scores, and XG boost is giving us highest 80% recall.

Conclusion:

Class imbalance is a big problem in classification, but there are more techniques that may help in catering this problem more accurately like 'Ensembling' technique for imbalance class to create different models from different sets and aggregate the results. experiments are the only way we can improve. but due to limited resources and time we have made our conclusion. We have focused on Recall to be more as per the business usecase, recall is important here more than precision, because a business can run marketing campaign even for customers who are not supposed to subscribe but cannot lose potential customer because of the predictive model failure; so our selection criteria depends on the Recall score more than anything.

Highlighted Insights:

1. 'Duration' of call is highly correlated with target variable but for creating a realistic model remove duration for modeling, as duration is only known after call and we already knew the result after call has happened. And for that reason f1, roc_auc score is low and we focused more on Recall.

2. Using Halving grid search and random search we were able to reduce training and computation time.

3. Previous loans, marital status, age, balance, contact month and weeks played an important role in predicting.

4. Random Oversampling can produce uneven results, so synthetic over sampling is primarily considered.

5. Clients who received calls less than 5 times, and conversation lasts under 30 minutes duration are the most subscribed clients.

6. 1st, 3rd and 4th quarter has high subscription rate. and these quarters has comparatively low contact rate.

References-

1. Towards data science
2. Almabetter blogs
3. Stack Overflow
4. Scikit learn documentation