**Bank Marketing Prediction**

**Abhijeet Dutta**

**Data science trainees,**

**Alma Better, Bangalore**

**Abstract:**

A recommendation system is **a subclass of Information filtering Systems that seeks to predict the rating or the preference a user might give to an item**. In simple words, it is an algorithm that suggests relevant items to users.

***Keywords: unsupervised machine learning, recommendation, kindle, books, amazon***

**1.Problem Statement**

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe a term deposit (variable y).

The main objective is to build a predictive model, which could help them in predicting the success rate as well as important factors which increases the chance of subscription. This would in turn help them in focusing on the factors and estimate the success of campaign accordingly.

* Input variables:
* Bank Client data:
* age (numeric) job: type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
* marital: marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
* education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
* default: has credit in default? (categorical: 'no', 'yes', 'unknown')
* housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
* loan: has personal loan? (categorical: 'no', 'yes', 'unknown')
* Related with the last contact of the current campaign:
* contact: contact communication type (categorical: 'cellular', 'telephone')
* month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
* day\_of\_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
* duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
* Other attributes:
* campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
* pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
* balance: account balance of the client
* previous: number of contacts performed before this campaign and for this client (numeric)
* poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
* Social and economic context attributes
* Output variable (desired target):
* y - has the client subscribed a term deposit? (binary : 'yes','no')

**2. Introduction**

### Term deposits are **a low-risk way to invest your money and earn a fixed rate of interest**. They lock away your money for the time that you choose (the term), usually between one month and five years. If you need your money before the term ends, you have to pay a penalty fee.

### The classification algorithm observes all the inputs collected during the campaign and helps the bank to predict the probable chance of success (in this case subscription for the term deposit) which in turn will help the bank to decide to be more organized during the upcoming campaigns.

## **3. Types of Outcome**

* Yes (Subscribed)
* No (Not Subscribed)

## **4. Reasons for subscription**

* Campaign is more successful between the age group 20 to 60.
* Cellular contact is more effective than any other mode of contact.
* It is most popular among persons with stable income and those who are about to retire.
* unseen emergencies and so on.

# **5. How Term deposit works**

## **Demand for term insurance**

## Term deposits **let you invest for a set amount of time and get a fixed interest rate**. They can be useful when saving for bigger items like a car, or investing when you want to be certain about the interest you'll earn.

## **Low risk contract:**

Risk in Term Deposit is very low as compared to any other investment instruments. So people who are near to

their retirement and someone who wants

has any short duration goals are more inclined towards the subscription of this.

**6. Steps involved:**

**Exploratory Data Analysis**

After loading the dataset we performed this method to explore the hidden pattern of our target variable that is Y with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

It also helps to take decisions about feature engineering.

**Null values Treatment**

There is no missing data in our dataset.

**Encoding of categorical columns**

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features as well as manually map different categories of the variables to numerical values. Because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

**Feature Selection**

In these steps we used algorithms like Extra Tree classifier to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

**Treating Class imbalance:**

Since the classes in our dependent variable

are highly imbalanced (88%,12%).

In order to overcome this bias, we have applied SMOTE oversampling technique.

**Fitting different models**

For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **Decision Tree**
3. **Random Forest Classifier**
4. **XGBRF Classifier**
5. **XGBoost Classifier**

**Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models like Random Forest Classifier and XGBoost classifier.

* **SHAP Values for features**

We have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn’t put much weight on the performance of our model.

**7.1. Algorithms:**

1. **Logistic Regression:**

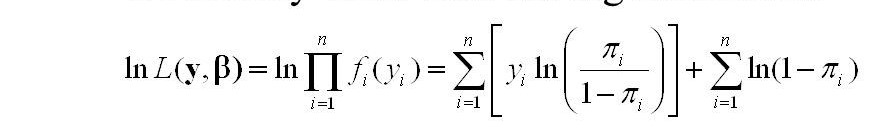
Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:



.**2.Random Forest Classifier:**

Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set, collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.



1. **XGBoost-**

To understand XGBoost we have to know gradient boosting beforehand.

* **Gradient Boosting-**

Gradient boosted trees consider the special case where the simple model is a decision tree

****

In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).

When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

**XGBoost** is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

**7.2. Model performance:**

Model can be evaluated by various metrics such as:

**1. Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification models at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

**2. Precision/Recall**-

Precision is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP

Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

**3. Accuracy**-

Accuracy is given by the number of correctly classified examples divided by the total number of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

**4. Area under ROC Curve (AUC)**-

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

**7.3. Hyper parameter tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Randomized Search CV for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into 5 folds.

**Randomized Search CV-** In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist’s control

**8. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA, encoding of categorical columns, feature selection, SMOTE oversampling and then model building.

In all of these models our accuracy revolves in the range of 69 to 75%.

After hyperparameter tuning with RandomizedSearchCV for random forest Classifier, the accuracy of our model improved to 76%. Hyper parameter tuned RandomForest modeling is proved to be the best model for our dataset which incurred 76% accuracy.

We used Shapely additive explanation method (SHAP) to find the importance as well as contribution of the independent features to classify the dependent variable.

Local interpretability is also explained by SHAP.

**References-**

1. sklearn documentation

2. GeeksforGeeks

3. Analytics Vidhya

4. Towards Data Science