**Bank Marketing Effectiveness Prediction**

**ANKUSH**

**AlmaBetter, Bangalore**

**Abstract:**

One of the industries that is being transformed the most by the recent Machine learning advances is the finance industry. Be it predicting the stock prices, or in our case predicting, a customer willingness to subscribe to a term deposit. Therefore, in our project we have come up with a solution that increases the efficiency by making fewer calls but improves the success rate.

Our experiment can help understand what could be the reason for the classification of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct classification.

**1.Problem Statement**

The data is related to 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 assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe to a term deposit (variable y).

**2.Data Summary**

* Age - Age of customer .
* Job - type of job customers do.
* Marital - Weather are customer is married, single or Divorced
* Education - Qualification of our customer.
* Default - Customer has credit in default or not.
* Housing - Customers have a housing loan or not.
* Loan - Customer has a personal loan or not.
* Contact: contact communication type.
* Month: last contact month of year.day of week: last contact day of the week .
* Duration: last contact duration, in seconds.
* Campaign: number of contacts performed during this campaign
* Pdays: number of days that passed by after the client was last contacted from a previous campaign
* Previous: number of contacts performed before this campaign and for this client
* Poutcome: outcome of the previous marketing campaign
* y : has the client subscribed a term deposit or not.

**3. Introduction**

One of the industries that is being transformed the most by the recent Machine learning advances is the finance industry. Be it predicting the stock prices, or in our case predicting, a customer willingness to subscribe to a term deposit. Therefore, in our project we have come up with a solution that increases the efficiency by making fewer calls but improves the success rate.

Our goal here is to build a predictive model, which could help to predict if the client will subscribe to a term deposit or not.

**4. Steps involved:**

* **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is y (deposit) 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.

* **Null values Treatment**

Our dataset contains no null values which is very good because null values might tend to disturb our accuracy.

* **Encoding of categorical columns**

We used One Hot Encoding and Label Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* **Fitting different models**

For modeling we tried various classification algorithms like:

* **Logistic Regression**
* **SVM Classifier**
* **KNN Classifier**
* **Random Forest Classifier**
* **Light GBM**
* **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 Light GBM.

**5. Algorithms:**

* **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)



.

* **Support Vector Machine Classifier:**

SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyperplane at a higher order dimension.



* **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.



* **KNN:**

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand



* **LightGBM-**

LightGBM, short for Light Gradient Boosting Machine, is a free and open source distributed gradient boosting framework for machine learning originally developed by Microsoft. It is based on decision tree algorithms and used for ranking, classification and other machine learning tasks.



**6. Model performance:**

Model can be evaluated by various metrics such as:

* **Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification modelis 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.

* **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

* **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

* **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. 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 Grid Search CV for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds.

* **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

**8. Conclusion:**

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

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

In all of these models our accuracy revolves in the range of 79 to 94%.

And there is no such improvement in accuracy score even after hyperparameter tuning.

So the accuracy of our best model is 92% which can be said to be good for this large dataset. This performance could be due to various reasons like: no proper pattern of data, too much data, not enough relevant features.

**References-**

* GeeksforGeeks
* Stack\_overflow
* Stat\_quest