Bank Marketing Effectiveness Prediction

Ganeshkumar Patel, Yaman Saini, Akanksha Agarwal,
Sanjay Kumar, Saurabh Funde
Data Science Trainees, AlmaBetter, Bangalore

Abstract

Finance industry is one of the leading industries globally and have the potential to bring huge impact in the growth of nation. Thus, it is important to analyze the data or information that banking sector records about the clients. This data can be used to create connection and keep professional relationship with the customers in order to target them individually for any banking schemes.

Usually, the selected customers are contacted directly through: personal contact, telephone cellular, email or any other means of contact to advertise the new services or give an offer. This kind of marketing is called direct marketing and is one of the leading marketing techniques.

Thus, in this project we trained a model that can predict that whether the client will opt for a term deposit or not using given bank-client data, data related with the last contact of the current campaign and some other useful attributes

Keywords: Supervised Machine Learning, Classification, Predictions, duration.

1. Problem Statement

The given dataset is of a direct marketing campaign (Phone Calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (Target variable y).

We were provided with following dataset:

Bank Client data:

- age (numeric)
- job : type of job
- marital : marital status
- education
- default: has credit in default?
- housing: has housing loan?
- loan: has personal loan?

Related with the last contact of the current campaign:

- 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 (numeric).

Other attributes:

- 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
- poutcome: outcome of the previous marketing campaign

Output variable (desired target):

- y - has the client subscribed a term deposit? (binary: 'yes','no')

2. Introduction

Marketing is the most common method which many companies are using to sell their products, services and reach out to the potential customers to increase their sales. Telemarketing is one of them and most useful way of doing marketing for increasing business and build good relationship with customers to get business for a company. It's also important to select and follow up with those customers who are most likely to subscribe product or service.

There are many classification models, such as Logistic Regression, Decision Trees, Random Forest, KNN, ANN and Support Vector Machines (SVM) that can be used for classification prediction.

3. Classification Approach

After understanding the problem statement, we loaded the dataset for following operations:

- **Data Exploration**
- **Exploratory Data Analysis**
- **Feature Engineering**
- **Feature selection**
- **Balancing Target Feature**
- **Building Model**
- **Hyperparameter Tuning**

3.1 Dataset Exploration

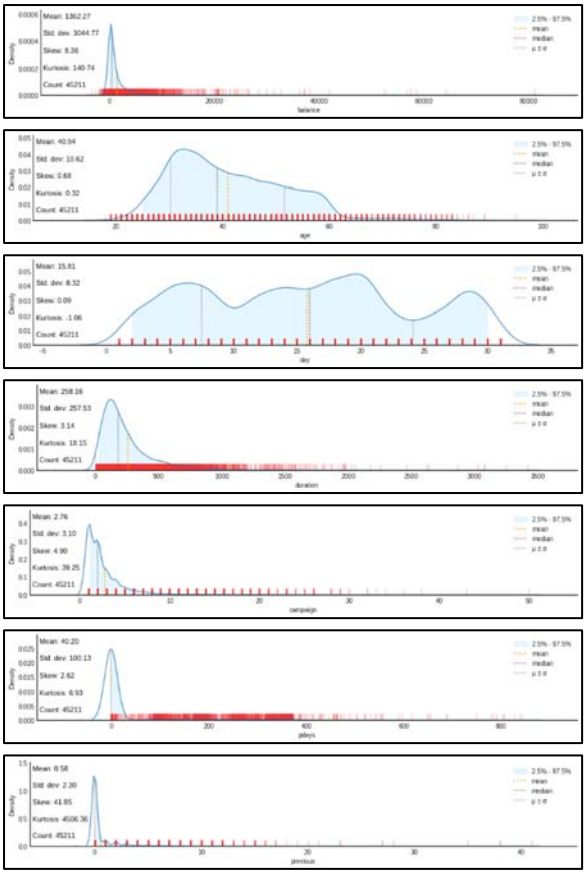
The given dataset was initially loaded for a quick overview. It was observed that our dataset contains 45211 records and 17 features. Datatypes of features was then checked and it was found that there are 7 numerical (int) and 10 Categorical (object) datatypes among which no null values and duplicated records were found in our dataset.

3.2 Exploratory Data Analysis

After data wrangling, we did univariate and multivariate analysis on features to understand their pattern and how they relate with target class.

A. Univariate Analysis

We initially plotted the numerical features using klib library.

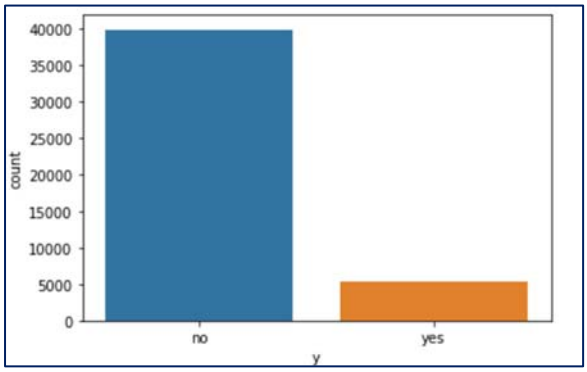


All 7 numerical data types graphs were right skewed and except for 'age' and 'day' all have outliers which should be removed.

B. Bivariate Analysis

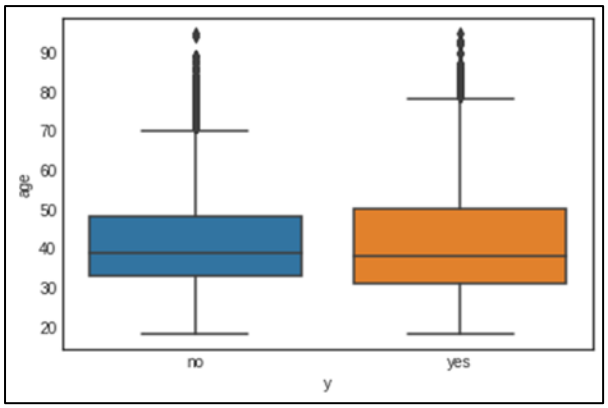
a) Target Class Distribution

We first plotted target class distribution and found that it is highly imbalanced with ratio of 88.3:11.6 i.e out of 100 clients only 12 of them opted for term deposit.



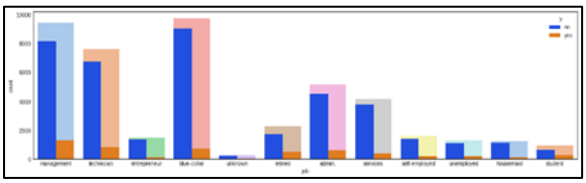
We then plotted each categorical feature with target variable to get some insights.

b) Age of clients

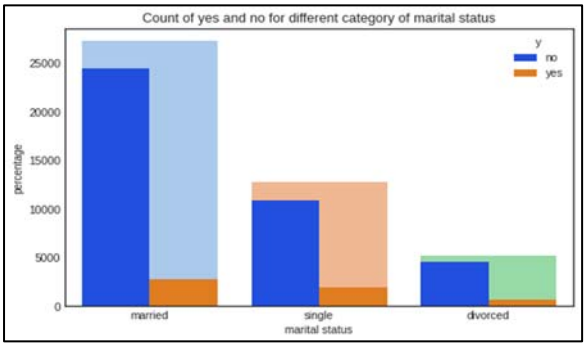


In the above plot it is clear that a majority of customers called is in the age of 30s and 40s (33 to 48 years old fall within the 25th to 75th percentiles) and for each of the target variable the age feature is not linearly separable. Thus, age will be of less importance to us.

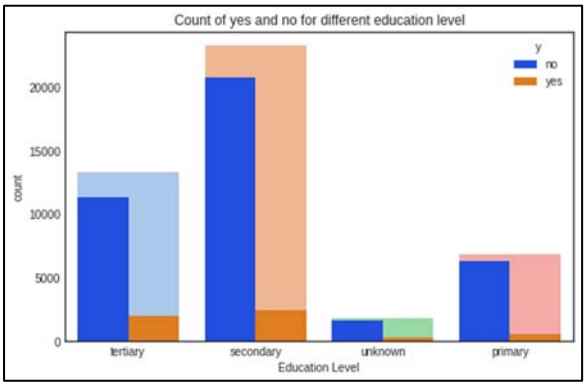
c) Job type- Customers with Blue-collar, management and technician showed maximum interest in subscription. We can also observe the large variance in our data among all categories.



d) Marital status

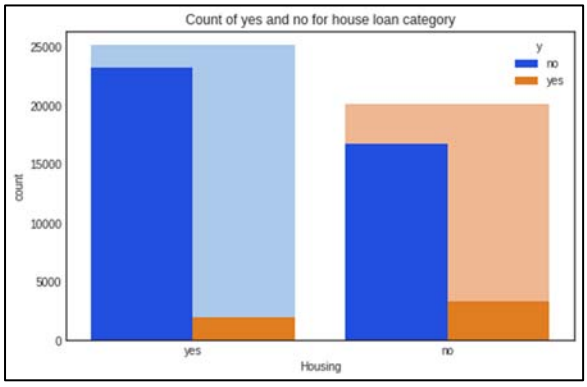


e) Education



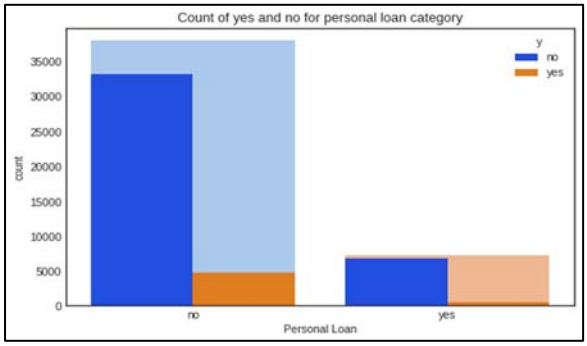
In comparison to primary and some unknown education, people with secondary and tertiary education were more driven towards paying term deposit in bank.

f) Housing Loan



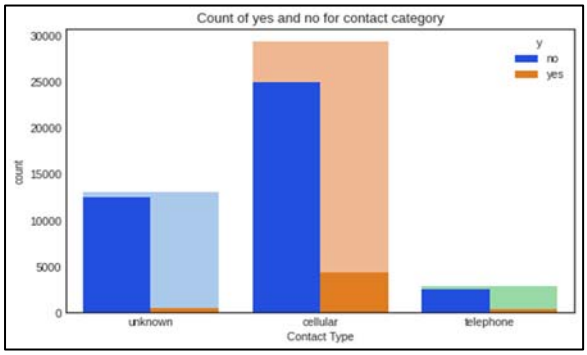
Majority of the people had previous housing loans and thus very few of them opted for term deposit.

g) Personal Loan



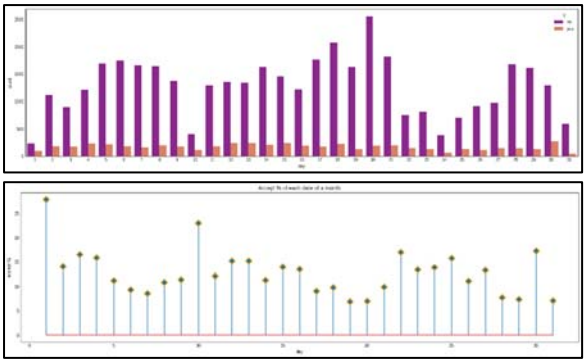
Majority of the people had personal loans and thus very few of them opted for term deposit.

h) Contact



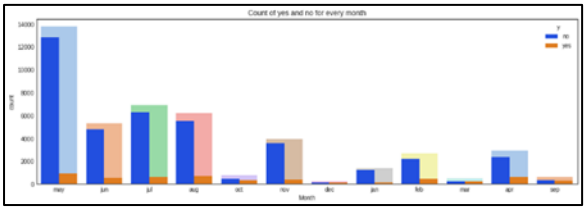
Majority of the people were contacted through cellular medium and were converted to the subscription. Thus, cellular medium of contact is more effective in comparison to telephone and other mediums.

i) Day



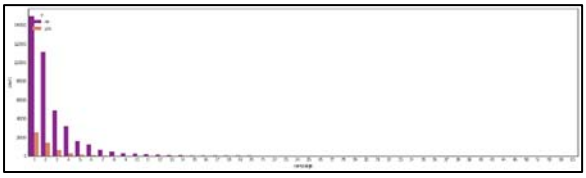
From the above plot of accepted term deposit vs day wise graph, we can predict that on first, tenth and near to the end of the month people took the term deposit. This may be due to the fact that various organizations have their schedules to release salary and then after people opted to pay deposit.

j) Month



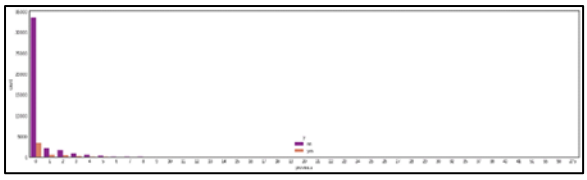
During the month of may there were maximum subscriptions with relatively good subscriptions in june, july and august. During other months we can se less subscription and so we can combine few of them later on.

k) Campaign



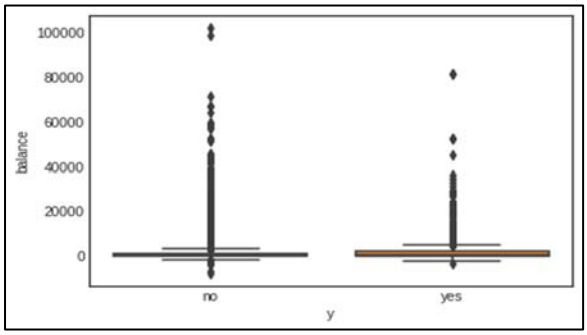
People were mostly contacted once who subscribed to it, while others were contacted more number of times but the conversion rate reduced after 20 we did not see any significant conversions thus we drop those observations later on.

l) Previous

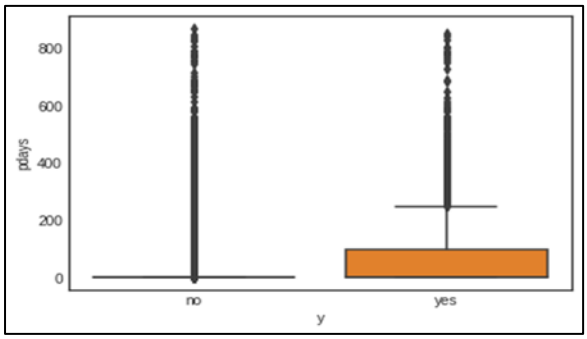


We can see above that majority of people were not contacted previously before this campaign and there are no significant contacts after 11 times already done.

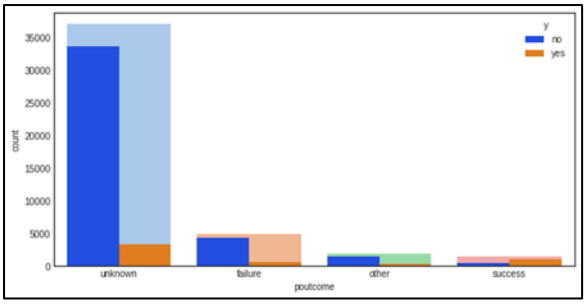
m) Balance- Balance of customers is more than 1 lacs and thus we need to remove outliers as the median is very less near to 450.



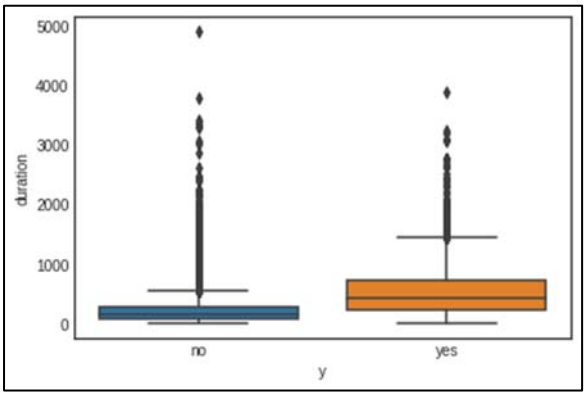
n) Pdays- pdays have large outliers and will have to look upon more closely.



o) Poutcome- Looking at poutcome we can infer that the success rate was high for some unknown category.



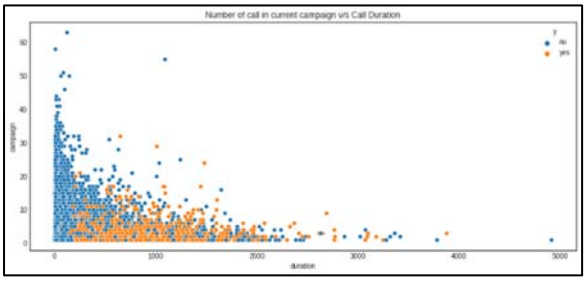
p) Duration



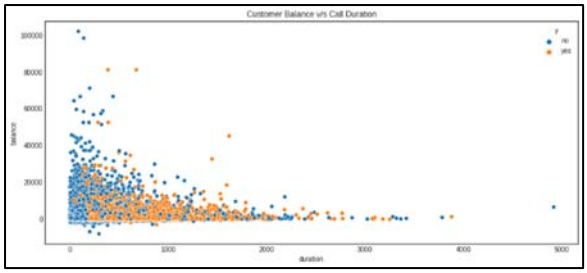
The above box plot shows that calls with large duration has more tendency for conversion.

C. Multivariate Analysis

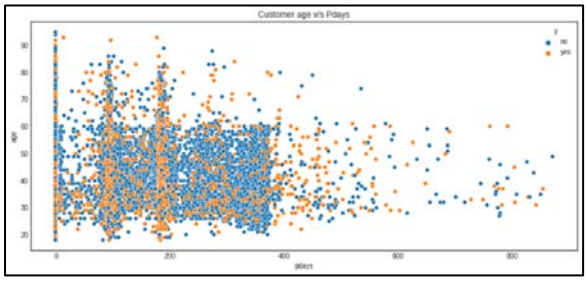
After performing above univariate analysis we tried to dig deep into the data to see the relation between multiple numerical features with others of same data types.



The success rate is more for less number of contacts in this campaign



Most of the clients who have taken a term deposit do not have very high balance (mostly in between 0-20000)



Above scatter plot is depicting that most of the client that had been last contacted falls in 0-400(days)

3.3 Feature Engineering

Feature engineering is one of the important steps in model building and thus we focused more into it. We performed the following in feature engineering

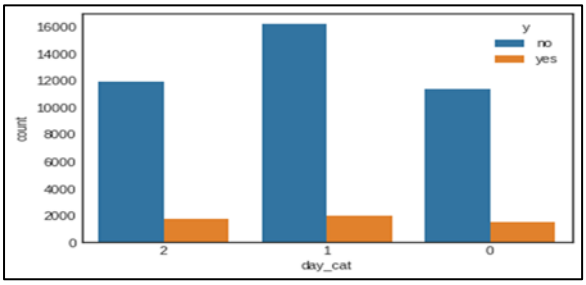
a) Dealing with outliers

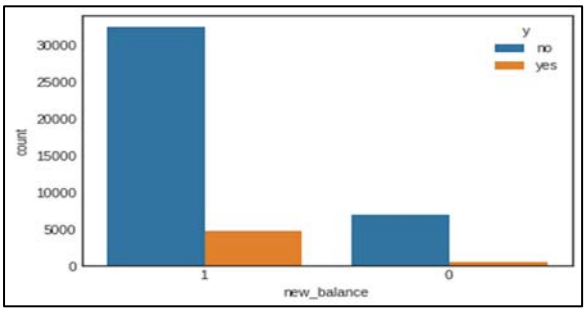
After looking at the plots above we removed the outliers

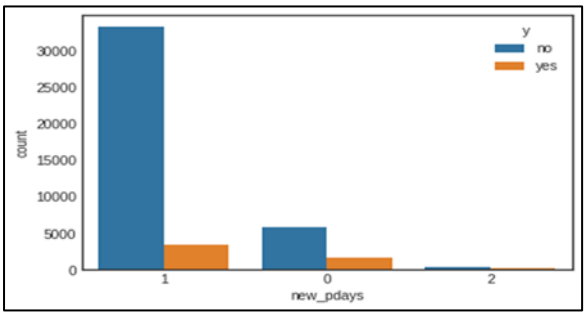
- In **duration** we removed those observation with no output and duration> 2000s
- In **campaign** we removed campaigns> 20
- In **previous** we removed observations for previous contacts> 11

b) Converting some discrete variable into categorical variable

Here we converted days, balance and pdays to ordinal forms and later converted them to categorical. By this we reduced the feature as well as did not assigned weights to them. After reducing we got following plots of them.







c) Reduce Job Categories

As we saw above, job has 12 categories, thus it will be good to reduce them. For this we combined few categories into one such as

- Admin + services = adms
- enterpenure + selfemployed + unemployed + unknown + housemade = others
- retired + student = rstd

d) Converting categorical variables into numeric- Here we converted all categorical features to numeric and then later converted them to categories so that the model don't assume them as weights.

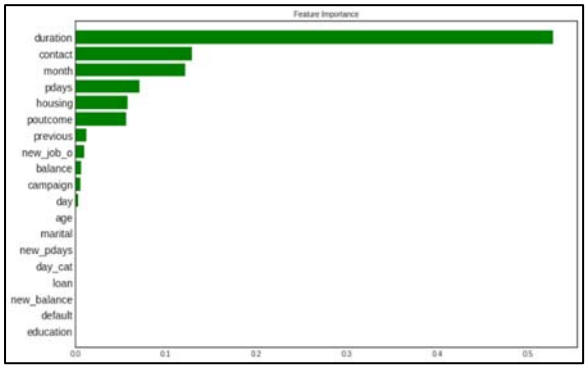
3.4 Correlation Analysis

We can see that **'pdays' and 'previous' are highly corelated features**, therefore, it will be good to remove any one of them to reduce multicollinearity.

3.5 Feature Selection

In feature selection we took the decision very wisely so that our model is trained well enough to predict the correct output. Thus, for selecting the right features we looked at the feature importance of decision tree.

Feature Importance after applying decision tree



From the above Decision tree feature importance we came to conclusion that only 'marital', 'age', 'day', 'campaign', 'balance', 'new_job_o', 'previous','poutcome', 'housing', 'pdays', 'month', 'contact' **are showing significant importance to be considered for model building. But from these features we removed** 'previous' **as this feature is showing high collinearity with pdays.**

3.6 Model Building

To start with building first we dealt with highly imbalanced data using SMOTE and then feature standardization.

- **Balancing Target Variable**

The target variable contains highly imbalanced labeled data in the 88:12 ratio. Using SMOTE which is basically used to create synthetic class samples of minority class to balance the distribution of target variable. The target variable balanced for modeling.

```
Before OverSampling, counts: Features (31296, 30) and Label (31296, 1)
yes
0      27624
1       3672
dtype: int64
yes
0      11839
1       1574
dtype: int64
After OverSampling, counts: Features (55248, 30) and Label (55248, 1)
yes
0      27624
1      27624
dtype: int64
```

- **Feature Standardization**

Standardization typically means rescales data to have mean of 0 and standard deviations of 1. To bring all values from independent variables in same scale. Using standard scalar, the independent variables transformed.

- **Fitting Different model's**

There are several classification models available for prediction/classification. In this

project we used following models for classification Algorithm's

1. **KNN**
2. **Random Forest**
3. **LGBM**
4. **ANN**

3.6.1 K-Nearest neighbors (KNN)

K-Nearest Neighbor is a non-parametric supervised learning algorithm both for classification and regression. The principle is to find the predefined number of training samples closest to the new point and predict the correct label from these training sample. It's a simple and robust algorithm and effective in large training datasets.

Following are steps involved in KNN.

1. Select the K value.

2. Calculate the Euclidean distance between new point and training point.

3. According to the similarity in training data points, distance and K value the new data point gets assigned to the majority class.

```
Cross_validation score [0.75221719 0.95701357 0.95728507 0.95447552 0.9572812 ]
KNN Test accuracy Score 0.861179452769701
              precision    recall  f1-score   support

           0       0.95      0.89      0.92     11839
           1       0.44      0.62      0.51      1574

    accuracy                           0.86     13413
   macro avg       0.69      0.76      0.72     13413
weighted avg       0.89      0.86      0.87     13413

array([[10572,  1267],
       [  595,   979]])
```

3.6.2 Random Forest

Random forest is a Decision Tree based algorithm. It's a supervised learning algorithm. This algorithm can solve both type of problems i.e. classification and regression. Decision Trees are flexible and it often gets overfitted. To overcome this challenge Random Forest helps to make classifications more efficiently. It creates a number of decision trees from a randomly selected subset of the training set and averages the-final outcome. Its accuracy is generally high. Random forest has ability to handle large number of input variables.

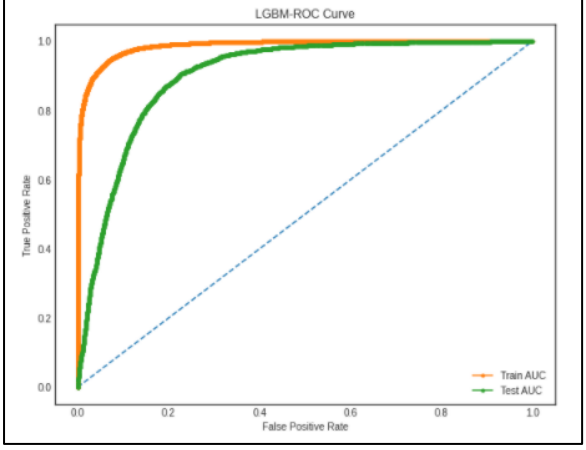
```
Cross_validation score [0.80361991 0.93475113 0.93402715 0.93203005 0.93257308]
RandomForest Test accuracy Score 0.7828971892939686
              precision    recall  f1-score   support

           0       0.97      0.77      0.86     11839
           1       0.33      0.85      0.48      1574

    accuracy                           0.78     13413
   macro avg       0.65      0.81      0.67     13413
weighted avg       0.90      0.78      0.82     13413

array([[9164, 2675],
       [ 237, 1337]])
```

3.6.3 LGBM

Light GBM is a gradient boosting algorithm based on Decision trees to increases the efficiency of the model. It is a fast-processing algorithm. It takes less memory to run and is able to deal with large amounts of data. It's a go to algorithm for huge datasets as other algorithm takes huge computational time to give output.

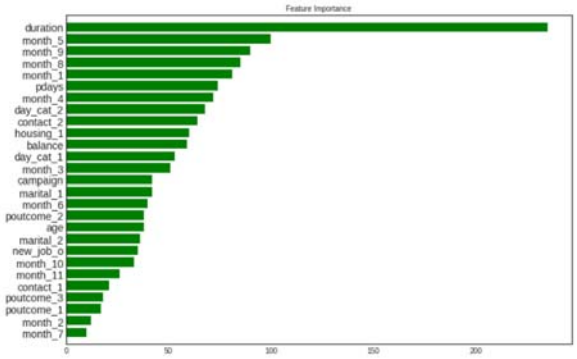
```
Cross_validation score [0.75475113 0.9519457  0.95429864 0.95447552 0.95230338]
LGBM Test accuracy Score 0.8417952732423768
              precision    recall  f1-score   support

           0       0.97      0.85      0.90     11839
           1       0.41      0.81      0.55      1574

    accuracy                           0.84     13413
   macro avg       0.69      0.83      0.72     13413
weighted avg       0.91      0.84      0.86     13413

array([[10018,  1821],
       [  301,  1273]])
```

```
**************************************************
Train AUC: 0.986
Test AUC: 0.902
**************************************************
Train Accuracy: 0.938
Test Accuracy: 0.842
```



The above **ROC curve** shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). As we can see the train data curve is closer to the top left and is having high AUC of 0.982 while on test data the AUC is reduced a bit but still the area is 0.897.

- **Feature Importance from LGBM**



The above plot clearly shows that *'duration' has the highest feature importance***.**

But in our dataset, it is mentioned to avoid 'duration' for building more realistic model**. Thus we now removed the 'duration' feature from our list of features and trained the model to see the accuracy score.**

3.6.4 Model Building without using 'duration' feature

Here, we will compare LGBM (as we got best score with duration) and ANN (neural nets may lead to higher accuracy) models

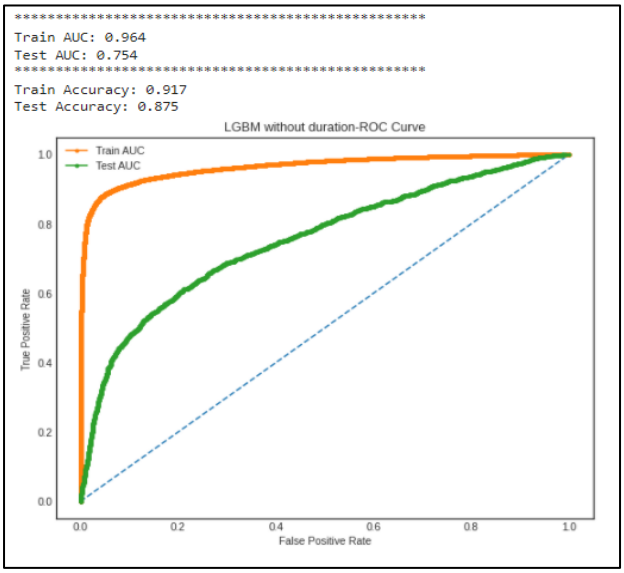
- **LGBM (without 'duration' feature)**

```
Cross_validation score [0.70334842 0.94479638 0.94371041 0.93945153 0.94433885]
LGBM Test accuracy Score 0.87541936926862
              precision    recall  f1-score   support

           0       0.92      0.94      0.93     11839
           1       0.46      0.39      0.42      1574

    accuracy                           0.88     13413
   macro avg       0.69      0.66      0.68     13413
weighted avg       0.87      0.88      0.87     13413

array([[11129,   710],
       [  961,   613]])
```

For our models with 'duration' **feature included we focussed on** recall score **to see the actual conversion rate, but after** removing this feature **we are more focussing on** precision score**, because now we are actually trying to predict the chances of conversion from the list of customers provided.**

Here we can clearly see that after dropping 'duration' feature, there is steep decrease in recall score, but precision score is increased**.**

```
************************************************
Train AUC: 0.964
Test AUC: 0.754
************************************************
Train Accuracy: 0.917
Test Accuracy: 0.875
```



We can also see the decrease in AUC on test data after removing the 'duration' feature.

- **Hyperparameter tuning using LGBM**

  Hyperparameters are important part in the Machine Learning and model improvement process. Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm.

```
Cross_validation score [0.70334842 0.94479638 0.94371041 0.93945153 0.94433885]
LGBM_Hypertunning Test accuracy Score 0.8764631327816298
              precision    recall  f1-score   support

           0       0.92      0.94      0.93     11839
           1       0.47      0.39      0.42      1574
```

There was no noticeable effect on score with hyperparameter tuning.

3.6.5 ANN

Artificial neural networks are representative of the human brain and they are specifically made to recognize patterns. ANN interpret data through various models. The patterns that these models detect are all numerical and are in the form of vectors. Artificial Neural Networks (ANN) have many different coefficients, which it can optimize. Hence, it can handle much more variability as compared to traditional models.

Without 'duration' Feature

```
ANN test Test accuracy Score 0.8777305599045702
              precision    recall  f1-score   support

         0.0       0.92      0.95      0.93     11839
         1.0       0.47      0.37      0.41      1574

    accuracy                           0.88     13413
   macro avg       0.70      0.66      0.67     13413
weighted avg       0.87      0.88      0.87     13413

array([[11194,   645],
       [  995,   579]])
```

4. Model Evaluation

For classification problems we have different metrics to measure and analyze the model's performance. In highly imbalanced target feature accuracy metrics doesn't represents true reality of model.

4.1 Confusion Matrix

The confusion matrix is a tabular form metrics which tell us the truth labels classified versus to the model predicted labels. True Positive signifies the how many positive classes samples model able to predict correctly. True Negatives signifies how many negative class samples the model predicted correctly.

4.2 Precision/Recall

Precision is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP. It focus on Type 1 error.

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

4.3 Accuracy

Accuracy is one of the simplest metrics to use. It's defined as the number of correct predictions divided by the total number of predictions and multiplied by 100.

4.3 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.

Model	Test accuracy	Precision	Recall	F1 Score
KNN	0.8611	0.44	0.62	0.51
RF	0.7828	0.33	0.85	0.48
LGBM With duration	0.8417	0.41	0.81	0.55
LGBM Without duration	0.8744	0.46	0.39	0.42
ANN Without duration	0.8777	0.47	0.37	0.41

Thus, we can conclude that Light GBM **is showing the best score of**

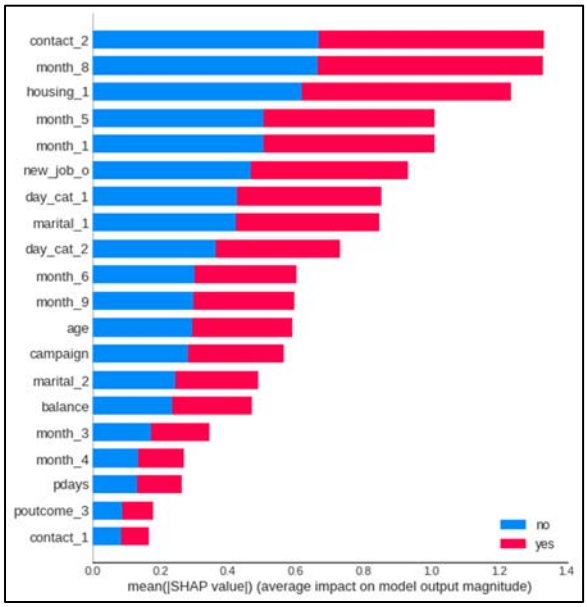
1) recall **of 0.81** using 'duration' **feature with precision of 0.41**

2) precision **of 0.46** without 'duration' **feature with recall of 0.39**

5. Feature Importance using SHAP

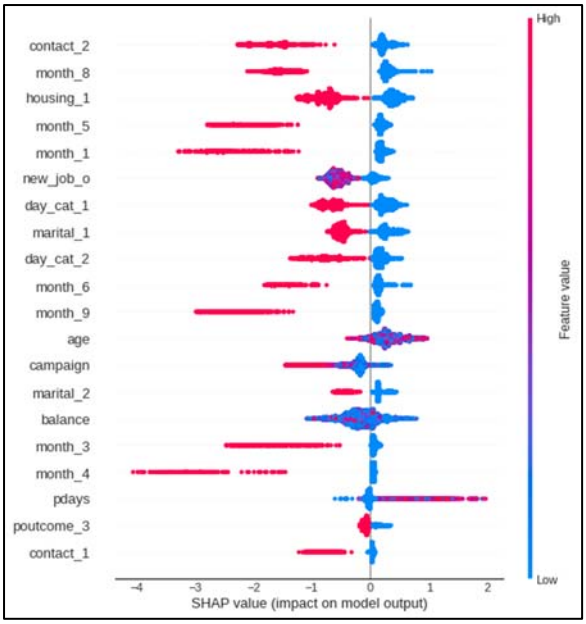
The SHAP framework has proved to be an important advancement in the field of machine learning model interpretation. SHAP combines several existing methods to create an intuitive, theoretically sound approach to explain predictions for any model. SHAP values quantify the magnitude and direction (positive or negative) of a feature's effect on a prediction

- **Summary plot**



In this plot, the impact of a feature on the target class is stacked to create the feature importance plot. From here we can see that contact2 has highest importance on the target 'yes', while other features such as month_8, housing_1 and subsequently all other features showed importance in decreasing order.

In the below plot we can see that for high positive shap value of duration, is impacting the model the most to predict the target value. While other remaining features is negative relation of value vs impact.



6. Conclusion and Future scope

Conclusion- Thus we come to an end of our analysis and model building to predict if the client will subscribe a term deposit or not. The most important takeaways are:

- **If the clients are contacted twice in first 10 days of the month, then it is more likely that call will be converted to subscription.**
- **Clients take more subscriptions in a month of January, May or August.**
- **If clients are already paying housing loans, then it is more likely that they will opt for term deposit.**

In our dataset we were provided with call 'duration' but in real world practice that won't be the case. So, to build more realistic model **we trained our models for both the cases i.e with or without 'duration'**. For both the cases **LGBM showed best scores of recall with 'duration' and precision without 'duration'.**

Future Scope- Our main objective is to get good precision score for without 'duration' models and good recall score for 'duration' included model.

So, we can initially formulate the required time to converge a lead using 'duration' included models and then sort out precise leads for 'duration' excluded models using this formulated time.

Here, the idea is to find out responses for any particular record with varying assumed predefined duration range.

For example, let's say, to converge a call, duration ranges between 60 to 2000 sec, then using this range we can predict all responses for each lead while iterating through this duration range. If we get positive response for any value of 'duration' we can assign that duration time to that particular lead.

In this way we can help marketing team to get precise leads along with time required to converge that lead and also, those leads that have least probability to converge (if we get no positive response for any assumed duration). Thus, an effective marketing campaign can be executed with maximum leads converging to term deposit.

References

1. For feature selection:

https://scikitlearn.org/stable/modules/feature_selection.html

2. For imbalance data handling:

https://imbalancedlearn.org/stable/#:~:text=Imbalancedlearn%20%28imported%20as%20imblearn%29%20is%20an%20open%20source%2C,tools%20when%20dealing%20with%20classification%20with%20imbalanced%20classes.

3. For metrics:

https://scikitlearn.org/stable/modules/model_evaluation.html

4. For shapley explainations:

https://towardsdatascience.com/explainable-ai-xai-with-shap-multi-class-classification-problem-64dd30f97cea