# CareerEra PGP (DataScience) – Capstone Project

# Banking Domain: Term Deposit Marketing Prediction

## Contents

# Capstone Project Bank Marketing Prediction

By - Naveen Nainegali

# Project Overview

The requirement is for identifying potential customer, which the bank can use for telemarketing to sell their products.

Telemarketing is a mean to advertise and educate the customer about the product and services which bank can sell to the customer. The selection of the customer plays an important role as bank pays the cost for telemarketing, and it will get returns on it only if the specific customer opts for the product or the service. The product should be of value to the customer only then they will chose to take the product.

If customer selection for the marketing is not done, it will :

Waste time and efforts of the bank , may create spoil the brand image, customer may block communication channel, perception of unnecessary harassment.

Telemarketing scope is not just via telephone, but also covers SMS, messenger apps, telephone, mobile, emails, chatbots etc.

The scope of the project is related to the telemarketing.

In telemarketing, customer is contacted as part of telemarketing process, customer is provided with information about the product and often requires to be contacted more than one time.

The problem in telemarketing domain is company may have to reach out to all of the customers be it a potential customer or not. This become a lot of logistical work and lot of efforts are wasted on the customers which don’t fit in the potential buyer category for that product.

This problem requires solution so that company can target only the potential customers which have higher probability to buy that product instead of wasting efforts on the customers which have lower probability of buying that product.

So past data can be used for the data analysis and ultimately machine learning techniques can be applied to predict the potential buyers as per various attributes related to customers.

Take advantage of data science has already been researched and published here.

## Objective

This dataset has been sourced from Paulo Cortez (Univ. Minho) and Sérgio Moro (ISCTE-IUL) @ 2012.

The collected data is related with direct marketing campaigns 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 (or not) subscribed.

There are two datasets:

1) bank-full.csv with all examples, ordered by date (from May 2008 to November 2010).
2) bank.csv with 10% of the examples (4521), randomly selected from bank-full.csv.

***The smallest dataset is provided to test more computationally demanding machine learning algorithms*

Following steps will be used to solve this problem.

1. Get Data:
    - Get data in a consumable format
2. Load Data
    - Load the data in Jupyter Notebook
3. Features Engineering -
    - We will be analyzing the data to find if any strong relation between any of the feature and outcome or within feature themselves.
4. Pre-Processing -
    - Data will be pre-processed like numerical features will be transformed so that they should be on single scale and categorical features will be encoded to numerical by using one-hot encoding and feature engineering and feature selection will be performed.
5. Model Selection
    - This is classification problem so various classification algorithms will be used like logistic regression, Support vector classifier, Random Forest Classifier, Ada-Boost Classifier, Gradient-Boost Classifier and K-Nearest Neighbors.
6. Model Training and Testing
    - Best classifier will be selected among mentioned classifiers on the basis of F score and various other parameters like training and testing times.
7. Model Finalization
    - Best classifier will be further tuned by tuning various hyper-parameters.
8. Model Validation
    - Stability of the classifier will be tested by using K-Fold.
9. Hyper-tuning
    - Top features will be selected on the basis of feature importance and accuracy will be checked with and without feature selection and accordingly feature selection will be done.

## Solution

We need to create a model which can accurately predict whether client is going to subscribe to term deposit or not.

Accurate prediction can help the bank to put efforts towards the potential clients which are going to subscribe the term deposit instead of calling each client.

So, suggested model should be created which can accurately predict whether client is going subscribe to term deposit or not.

## Data Analysis

Dataset contains information related to marketing campaigns of the bank. The marketing campaigns were based on phone calls and often more than one contact to the same client was required, to access if the product would be yes or no for the subscription.

Dataset contains 45211 examples and 16 input variables. Following are the input followed by output variables.

***Input Variable***

1. Age: Age is the factor which can impact client interest in the term deposit.
2. Job: This is type of job client have.
3. Marital: This is marital status of the client.
4. Education: This gives educational background of the client.
5. Default: Whether client has credit in default.
6. Balance: Average yearly balance (in euros)
7. Housing: Client has housing loan or not.
8. Loan: Client has personal loan or not.
9. Contact: contact communication type. Cellular or telephone.
10. Month: Last contact month of the year.
11. Day of week: Last contact day of the week.
12. Duration: Last contact duration in seconds. This attribute highly affects the output target (if duration close to 0 seconds then y is definitely “no”). This input will only be included for the benchmarking purposes and will be discarded for predictive modelling.
13. Campaign: Number of contacts performed during this campaign for this client including last contact.
14. Pdays: Number of days that passed by after the client was last contacted from a previous campaign. 999 means client was not previously contacted.
15. Previous: number of contacts performed before this campaign and for this client.
16. Poutcome: Outcome of the previous marketing campaign.

***Output Variable***

17. **Y:** Output variable. Has client subscribed to term deposit or not?

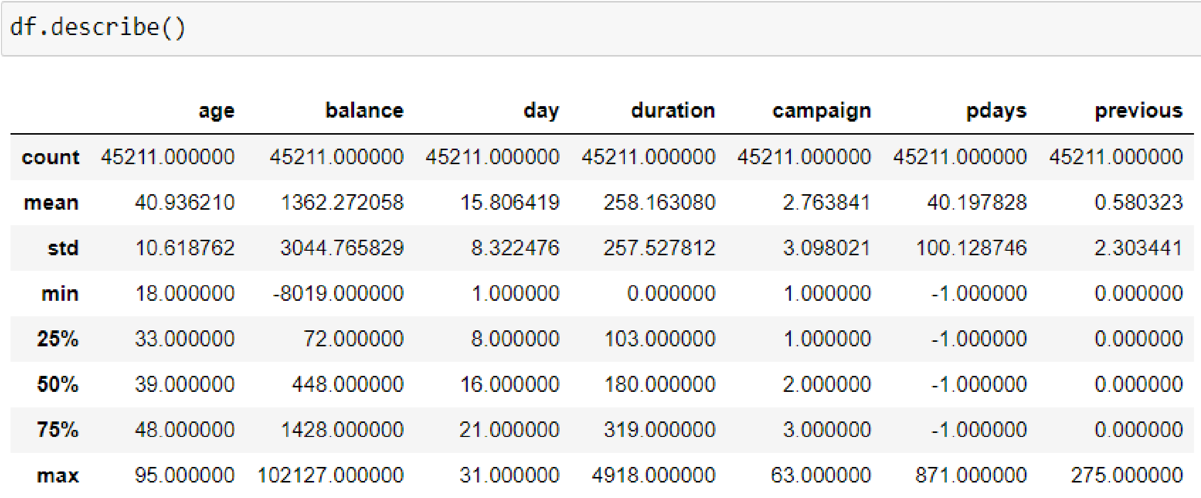
**A sample dataset with 10% of whole dataset is also made available to prepare and test the model.

All the above mentioned will be used for analysis and the prediction model building except the “Duration”.

- Data set contains total 7 numeric features and 10 categorical features.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        45211 non-null  int64 
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64 
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64 
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64 
 12  campaign   45211 non-null  int64 
 13  pdays      45211 non-null  int64 
 14  previous   45211 non-null  int64 
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

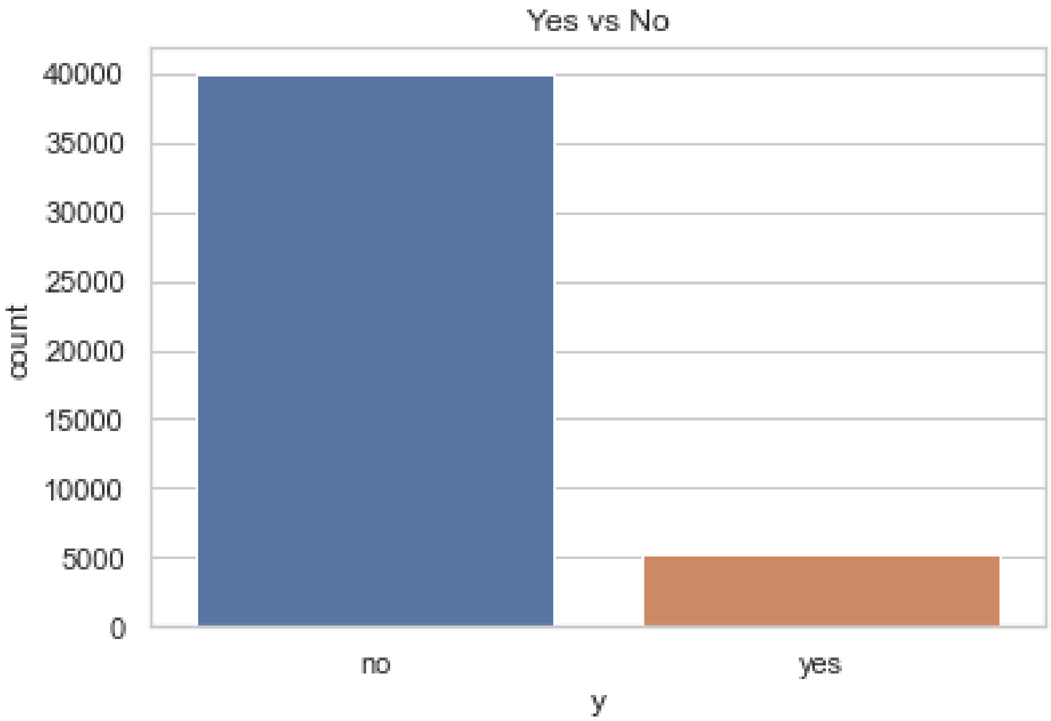
- Following are the stats of various numeric features.



```
df.describe()
```

| | age | balance | day | duration | campaign | pdays | previous |
|---|---|---|---|---|---|---|---|
| **count** | 45211.000000 | 45211.000000 | 45211.000000 | 45211.000000 | 45211.000000 | 45211.000000 | 45211.000000 |
| **mean** | 40.936210 | 1362.272058 | 15.806419 | 258.163080 | 2.763841 | 40.197828 | 0.580323 |
| **std** | 10.618762 | 3044.765829 | 8.322476 | 257.527812 | 3.098021 | 100.128746 | 2.303441 |
| **min** | 18.000000 | -8019.000000 | 1.000000 | 0.000000 | 1.000000 | -1.000000 | 0.000000 |
| **25%** | 33.000000 | 72.000000 | 8.000000 | 103.000000 | 1.000000 | -1.000000 | 0.000000 |
| **50%** | 39.000000 | 448.000000 | 16.000000 | 180.000000 | 2.000000 | -1.000000 | 0.000000 |
| **75%** | 48.000000 | 1428.000000 | 21.000000 | 319.000000 | 3.000000 | -1.000000 | 0.000000 |
| **max** | 95.000000 | 102127.000000 | 31.000000 | 4918.000000 | 63.000000 | 871.000000 | 275.000000 |

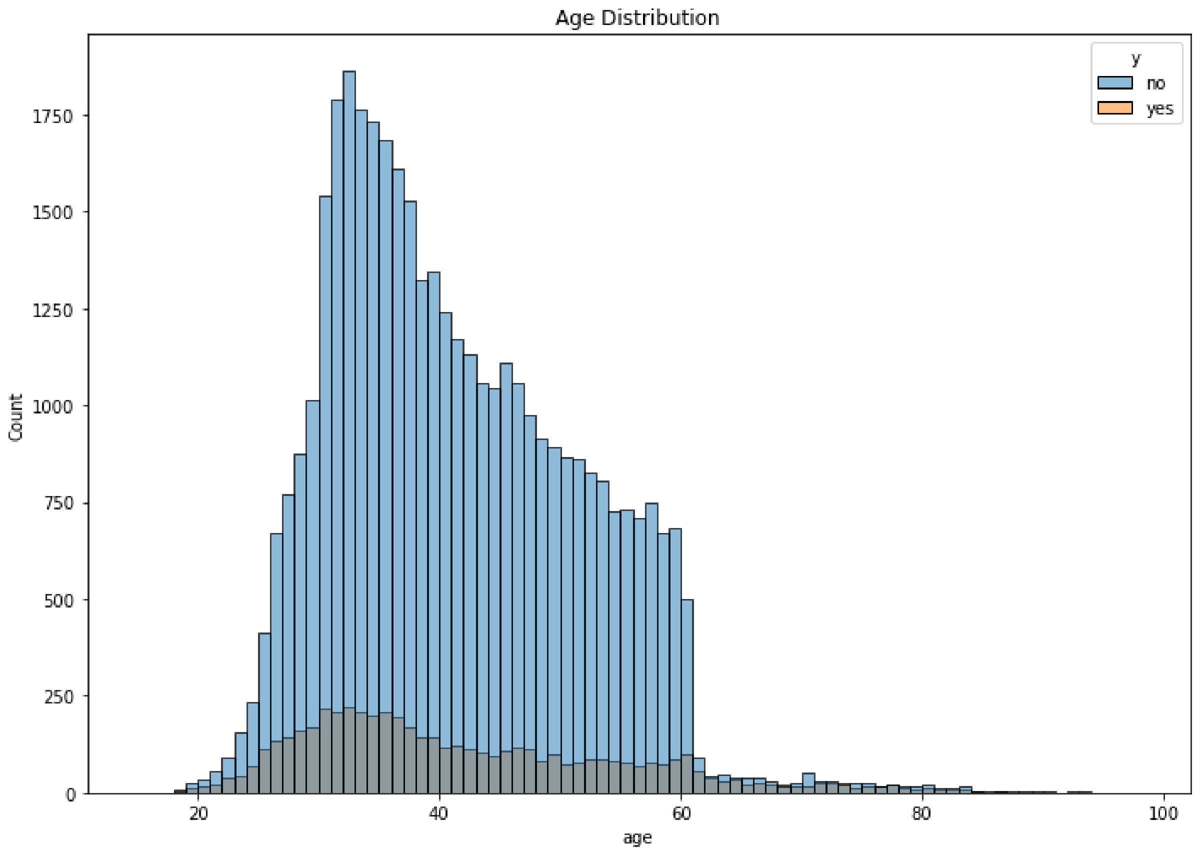
- This is the distribution of outcome variable and we can clearly see outcome is skewed towards “no” as compared to “yes”.



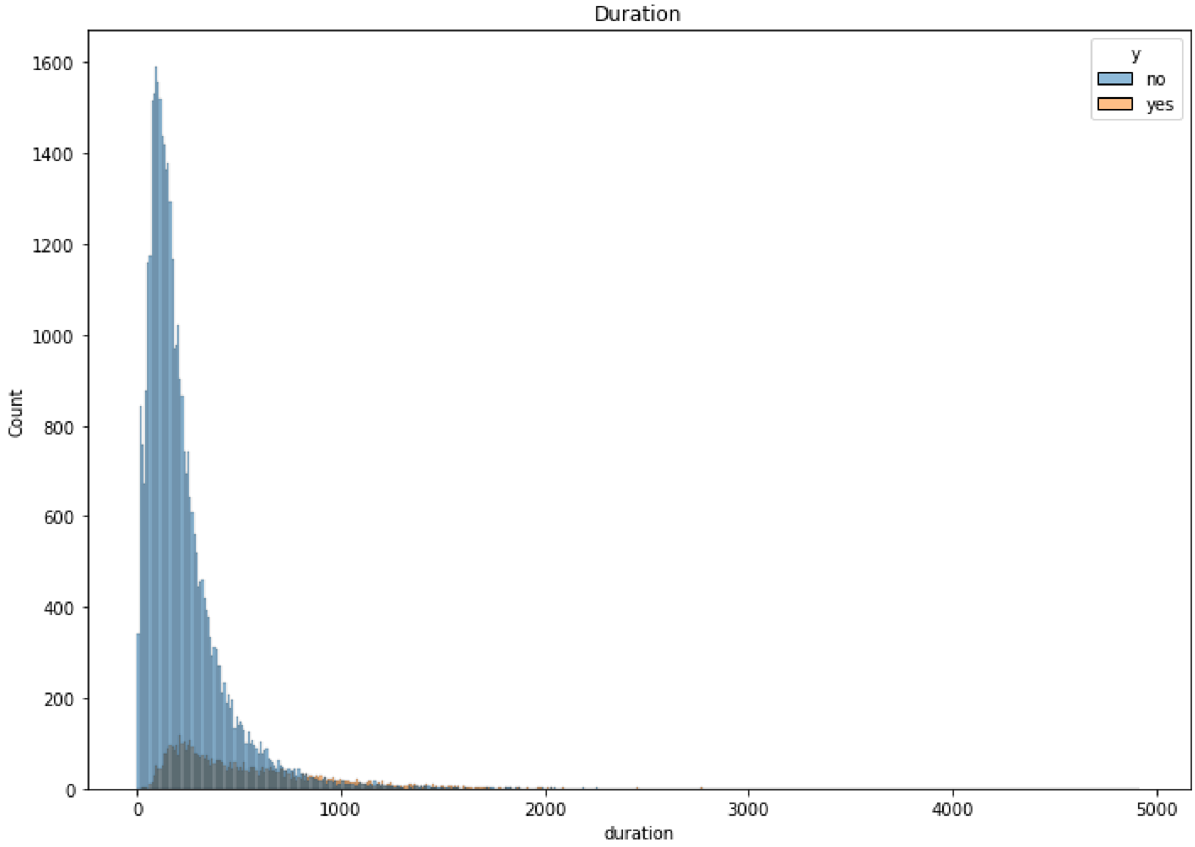
## Visualization

### Numeric Features

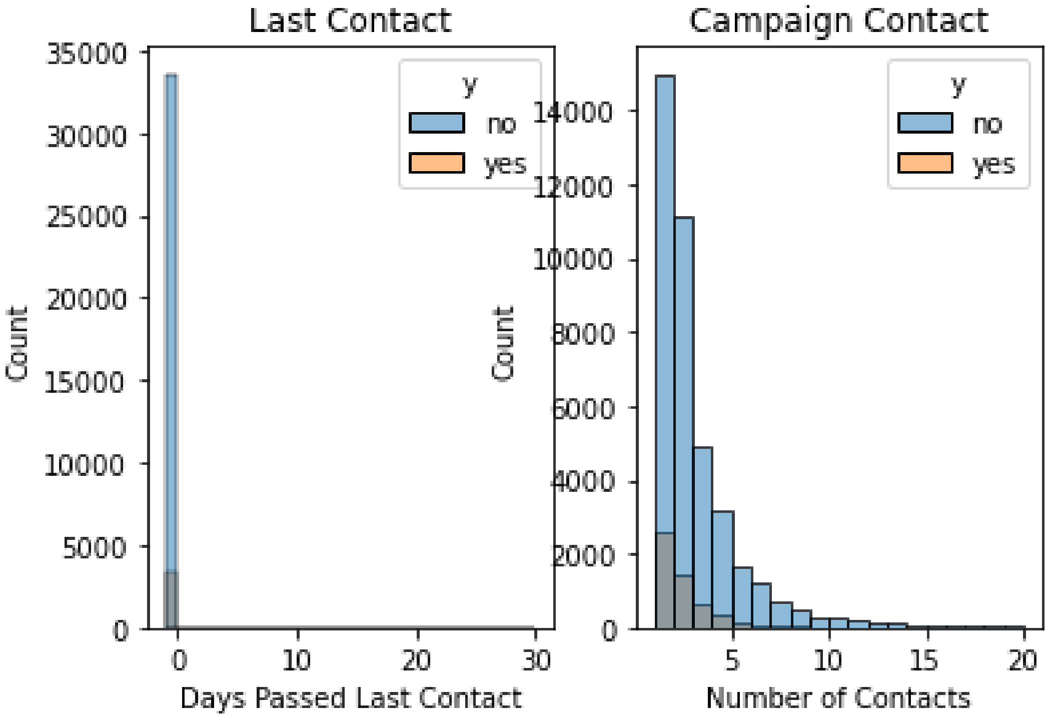
- Age is spread from 17 years to 98 years, Following are the histogram of age distribution.
    - Hue has been added to indicate count of people opening term deposit with the bank.
    - Success ration seems to be high before 25years and after 60 years.

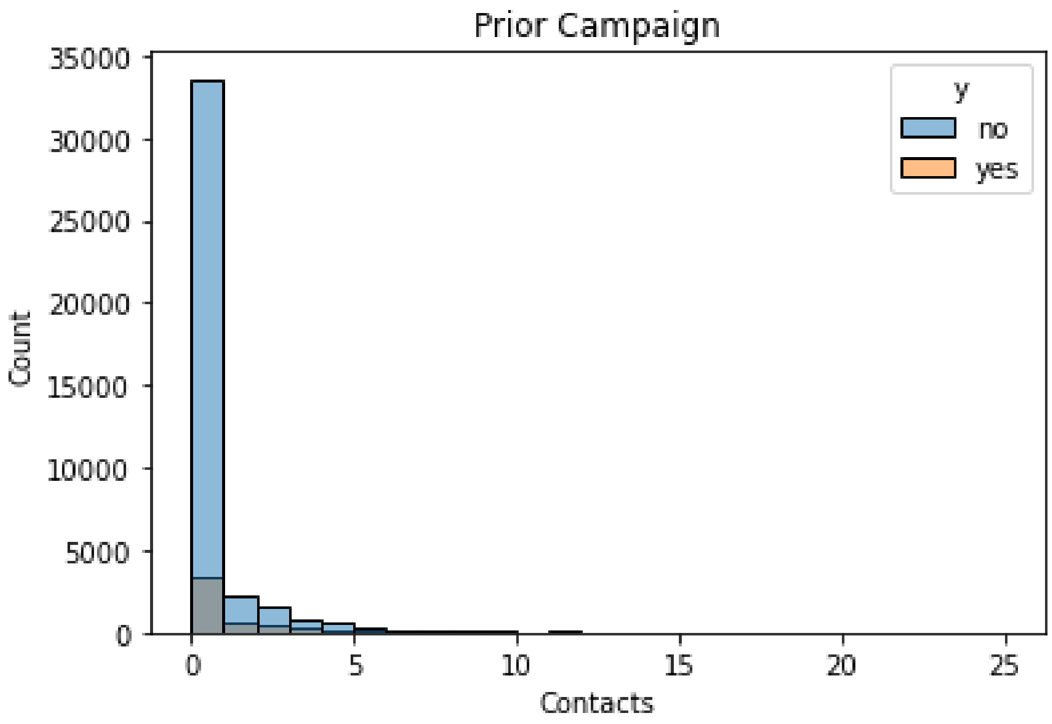


- Duration feature is also skewed.
    - High number of data on the left indicates call were disconnected within few seconds.
    - Those around 0 seconds duration have not resulted in any success possibly the customer were not contacted or the campaign objective could not be delivered.



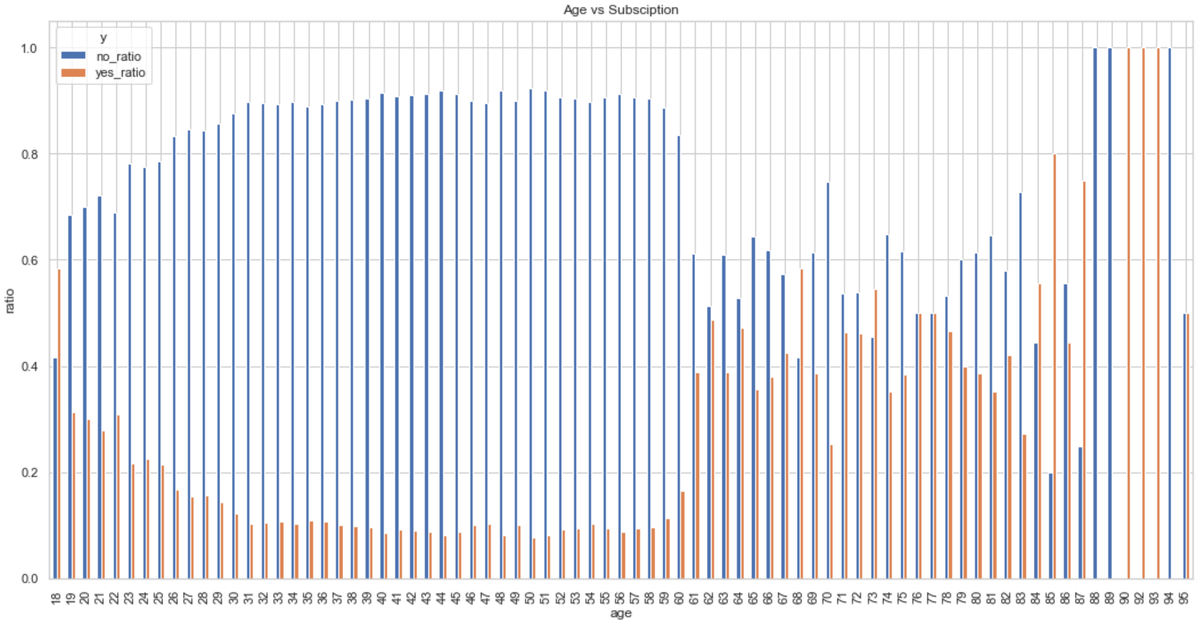
- Campaign feature is also showing that most of the new customer are targeted for marketing.
  - Pdays : Last contact as -1 indicate a new customer
  - Campaign : Number of contacts performed during this campaign
  - Previous : Number of contacts performed before this campaign, high number of 0 indicates that the customer was not contacted before.



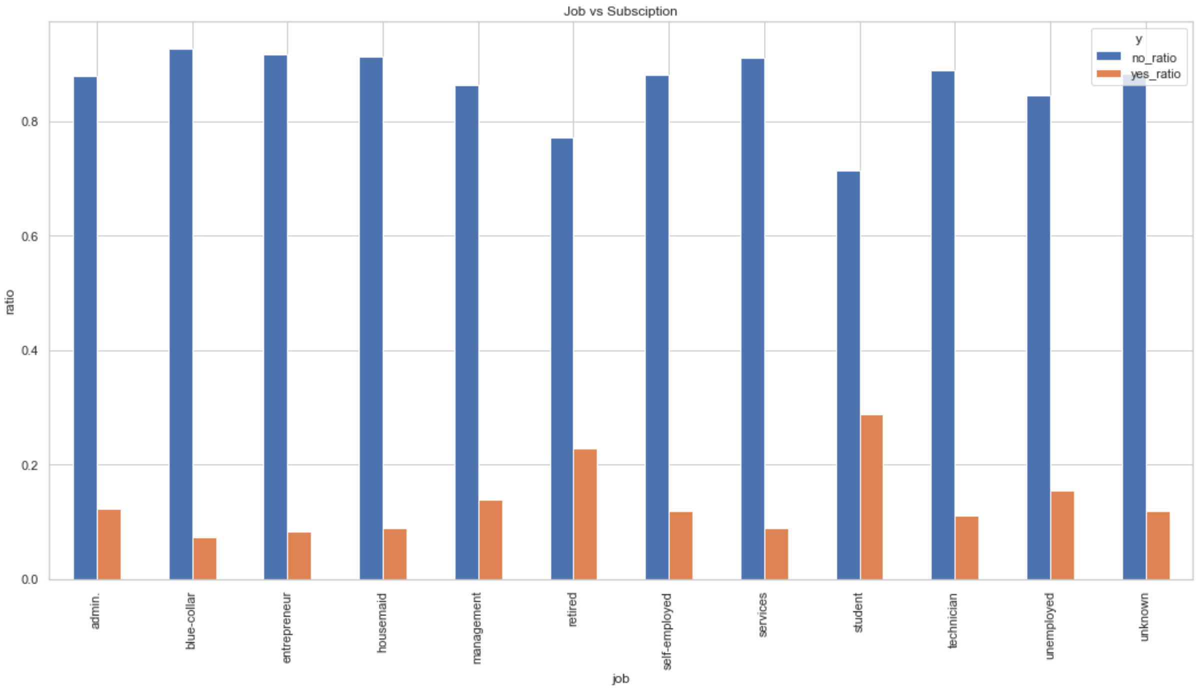


## Feature value count to success ratio exploratory analysis

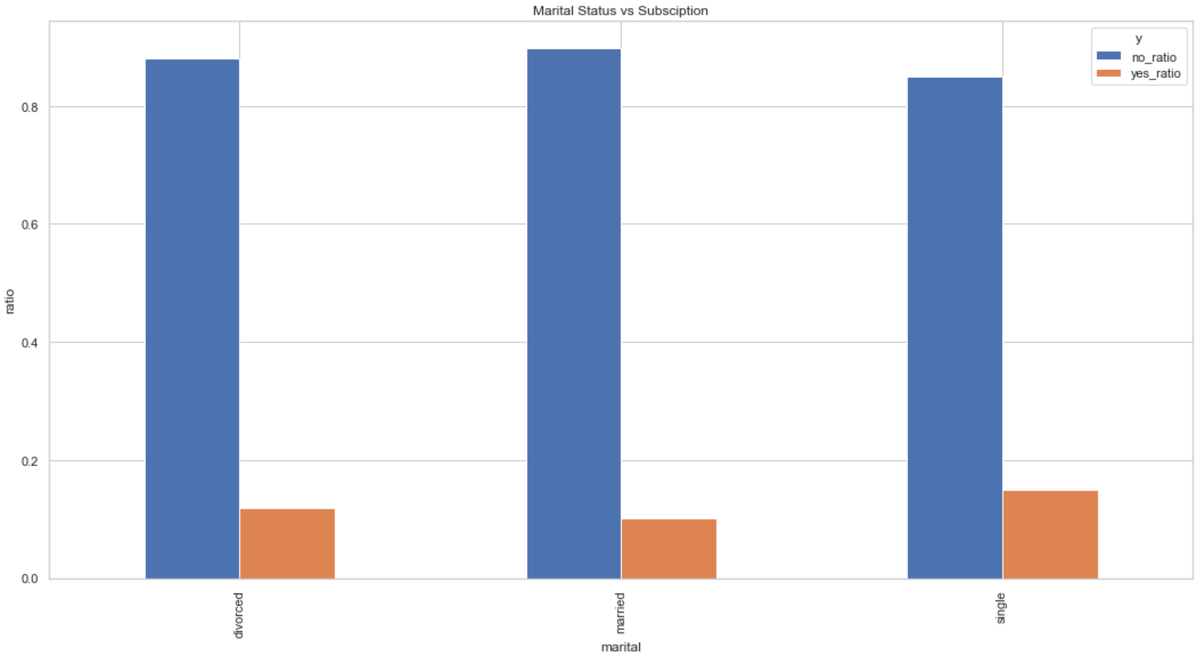
- Age vs y ratio: We can see from the above plot that success ratio is higher for the clients with age below 25 and above 60. In the mid age group success ratio is less.



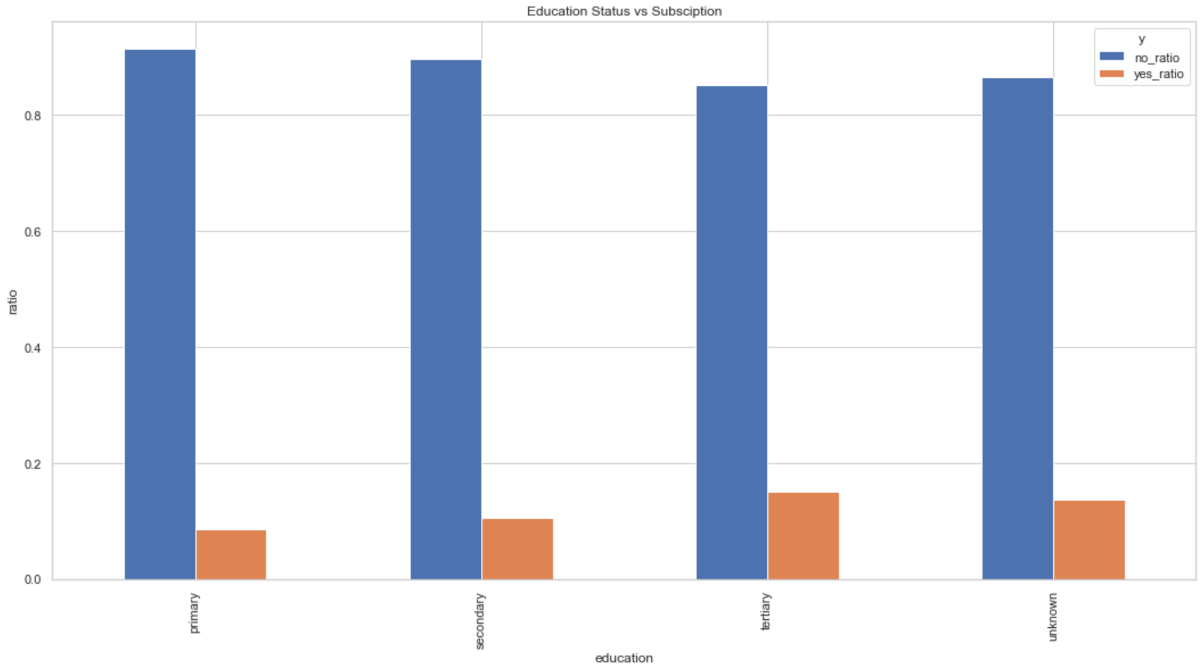
- Job type vs y ratio: Success ratio among retired and students are higher as compared to other professions.



- Marital status vs job: Not very high difference among the success ratios of different relationship types but it seems success ratio of Singles and unknowns are comparatively high.



- Education vs y ratio: Success ratio is higher for illiterate, but sample size is very small. There is not strong correlation between success and client education, but it seems higher the education better are the chances for success.



- Other variables do not contribute significantly to the decision of

## Algorithms and Techniques

This is a classification problem in which we need to identify whether client will opt for term deposit or not based on various other features.

We will use following 6 classification methodology and brief of all of these techniques is also given.

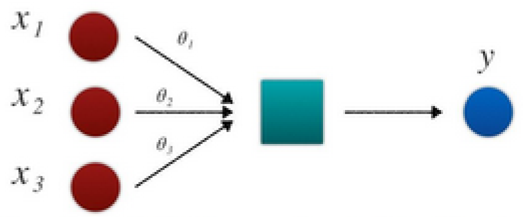
- **Logistic Regression:** Logistic regression training is all about the finding the relationship between independent variables or features to the class. Logistic regression tries to find which class is most likely to occur for the given values of features. This is done by finding the probability.

  In our example we have 2 classes, that is, individual will opt for term deposit or not. So, these are 2 classes. We have various independent variables or features like age, education, job, etc. Some of the features affect outcome more and some of the features affect income less.

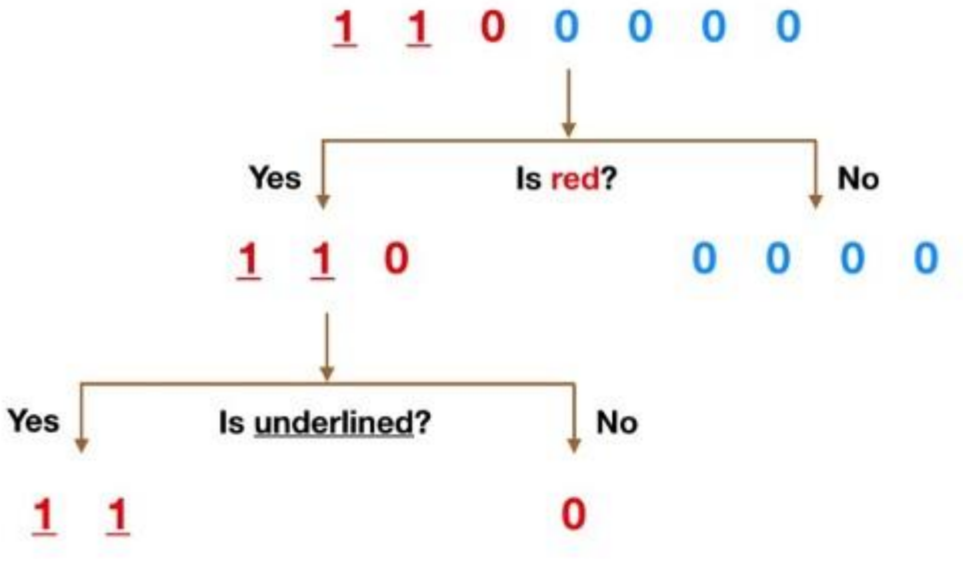
  While training for the logistic regression, we try to establish a relationship between all features and outcome to find the probability as yes is higher or no is higher.

  This relationship gives us a model or a mathematical relationship which calculates the probability of outcome as per given values of feature variables.

Usually logistic regression is used for binary classification which means output variable contains only 2 classes like "yes/no", "true/false", etc but that can be extended to use logistic regression to classify data to more than 2 classes.



- **K-Nearest Neighbor:** In this technique outcome is found based on the nearest neighbors of the data points. K is the number of neighbors and outcome will be decided based on the number of neighbors belong to that class. Let’s say any data point have more nearest neighbors belong to class “no” then outcome will be made as “no”.
- **Decision Tree:** They are the building blocks of the random forest model. Fortunately, they are pretty intuitive. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision trees in data mining.



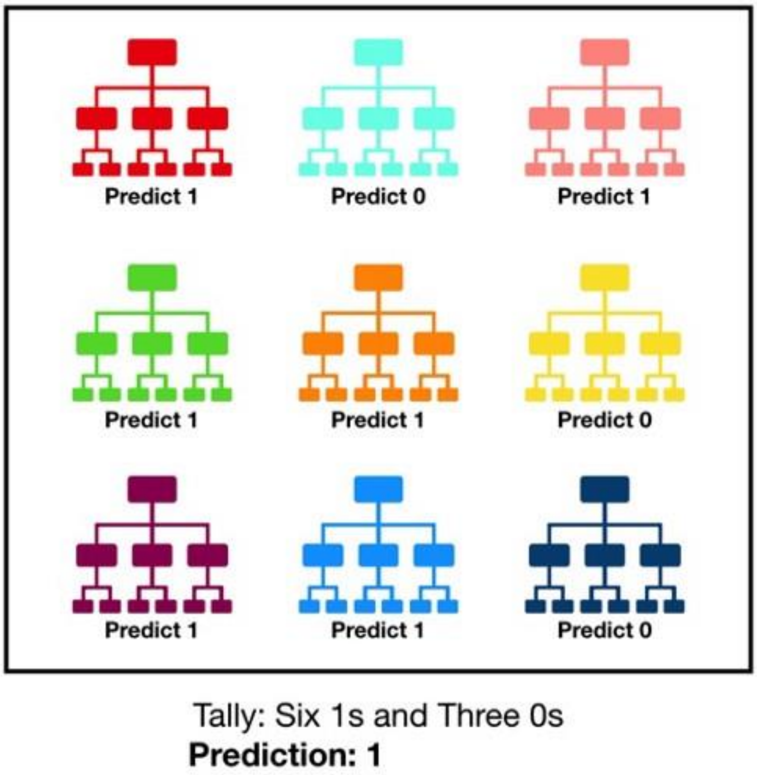
- **Naïve Bayes Gaussian :** In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.

  Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics literature, naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naïve Bayes is not (necessarily) a Bayesian method

- **Random Forest classifier:** Random Forest is a supervised learning algorithm. Like you can already see from its name, it creates a forest and makes it somehow random. The “forest” it builds, is an ensemble of Decision Trees, most of the time trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

  Model consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction



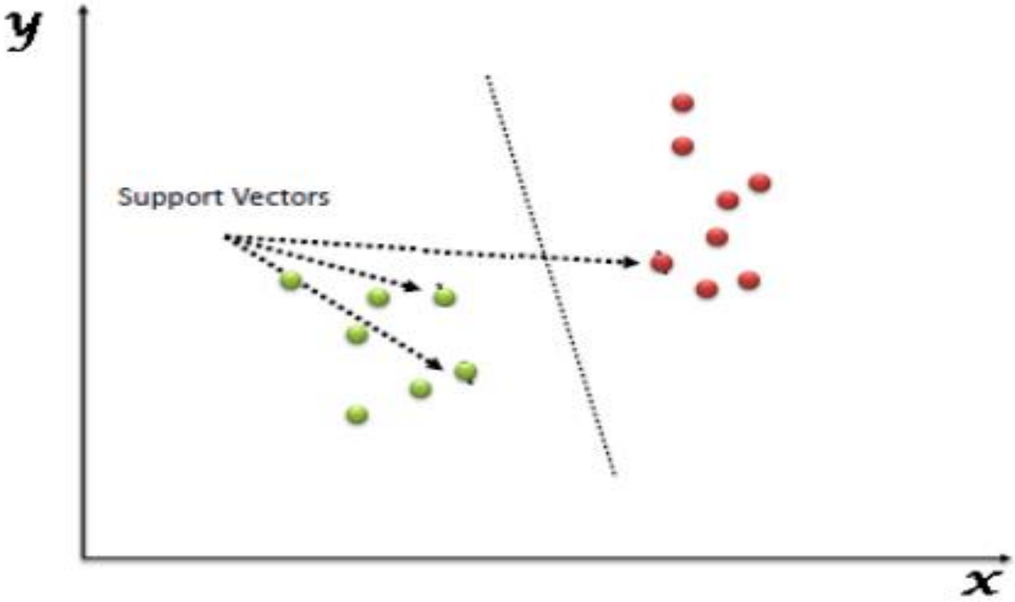
  Random forest method builds multiple decision trees and merges them together to get a more accurate and stable prediction.

  The prerequisites for random forest to perform well are:

  a. There needs to be some actual signal in our features so that models built using those features do better than random guessing.

b. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other

- **Support Vector Classifier:** “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

## Benchmark Model

Simple Naive predictor will be used as the benchmark model which will consider that every client is going to subscribe the term deposit because this is how current process of the organization is working. Currently organization is calling every client.

Predictive model which will be used as a solution should have way higher accuracy than the benchmark simple Naive predictor.

## Evaluation Metrics

Counts of clients who said “yes” is 5289 and those who said “no” is 39922. So this is clearly an unbalanced distribution. If we consider all “yes” which is usually the case and call to every client then we will get 11.69% of accuracy and same is the F1 score when beta = 0.

### *F-beta score and accuracy*

This is used as evaluation metrics for the predictive model and same will be compared with the benchmark model. Following is the formula of the F-beta score.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

**Accuracy** measures how often the classifier makes the correct prediction. It's the ratio of the number of correct predictions to the total number of predictions (the number of test data points).

**Precision** tells us what proportion of clients we classified as "yes" were actually "yes".. It is a ratio of true positives (clients which classified as "yes", and which are actually "yes") to all positives (all clients classified as "yes"), in other words it is the ratio of

**[True Positives/(True Positives + False Positives)]**

**Recall(sensitivity)** tells us what proportion of clients are "yes" were classified as "yes". It is a ratio of true positives (clients classified as "yes", and which are actually "yes") to all the words that were actually spam, in other words it is the ratio of

**[True Positives/(True Positives + False Negatives)]**

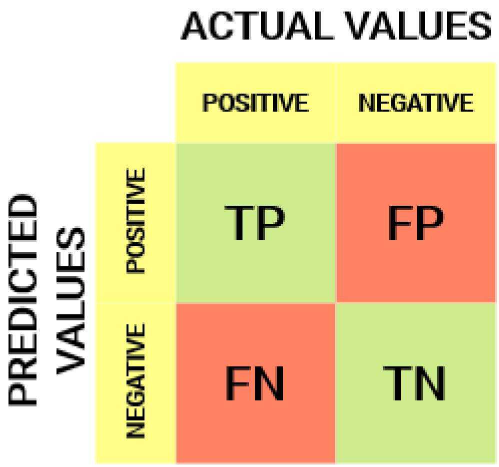
We don't want to miss any client which can say "yes" so we would like to focus more on the Recall. So I would like to keep value of beta as 0 so that recall can be focused.

If we will go with beta = 0 then we will be working on accuracy which is same as F score in our scenario.

### *Confusion Matrix and accuracy*

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:



|  |  | ACTUAL VALUES |  |
|---|---|---|---|
|  |  | POSITIVE | NEGATIVE |
| PREDICTED VALUES | POSITIVE | TP | FP |
|  | NEGATIVE | FN | TN |

**True Positive (TP)**

- The predicted value matches the actual value
- The actual value was positive and the model predicted a positive value

**True Negative (TN)**

- The predicted value matches the actual value
- The actual value was negative and the model predicted a negative value

**False Positive (FP) – Type 1 error**

- The predicted value was falsely predicted
- The actual value was negative but the model predicted a positive value
- Also known as the **Type 1 error**

**False Negative (FN) – Type 2 error**

- The predicted value was falsely predicted
- The actual value was positive but the model predicted a negative value
- Also known as the **Type 2 error**

Calculating accuracy for a given confusion matrix:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

# Methodology

## Data Pre-processing

Algorithm which depends on distance based requires data transformation so that feature value shouldn't be different from other feature value otherwise that feature will weigh more in that algorithm and will overshadow that other features.

We have used label encoder on the categorical features.

## Implementation

Following steps were used to implement the various algorithms.

- Data set was broken into 3 parts:
    - Training set
    - Validation set
    - Test set
- Training set will contain 70% part of the data set and will be used to train the data set.
- Validation set will be used for validation of model on unseen data and to visualize if model is overfitting or underfitting.
- Test set was used for further testing on the data and is used as a data for which outcome is not known.
- All 6-mentioned classification will be fitted on 10%, 50% and 100% training set.
- Then fitted model will be validated on model set and finally used for predicting on test set.
- Accuracy, training and testing times and F-beta score with beta = 0.001 is used because sklearn fscore metrics doesn't accept beta = 0.
- Then finally algorithm with highest F-score was selected.

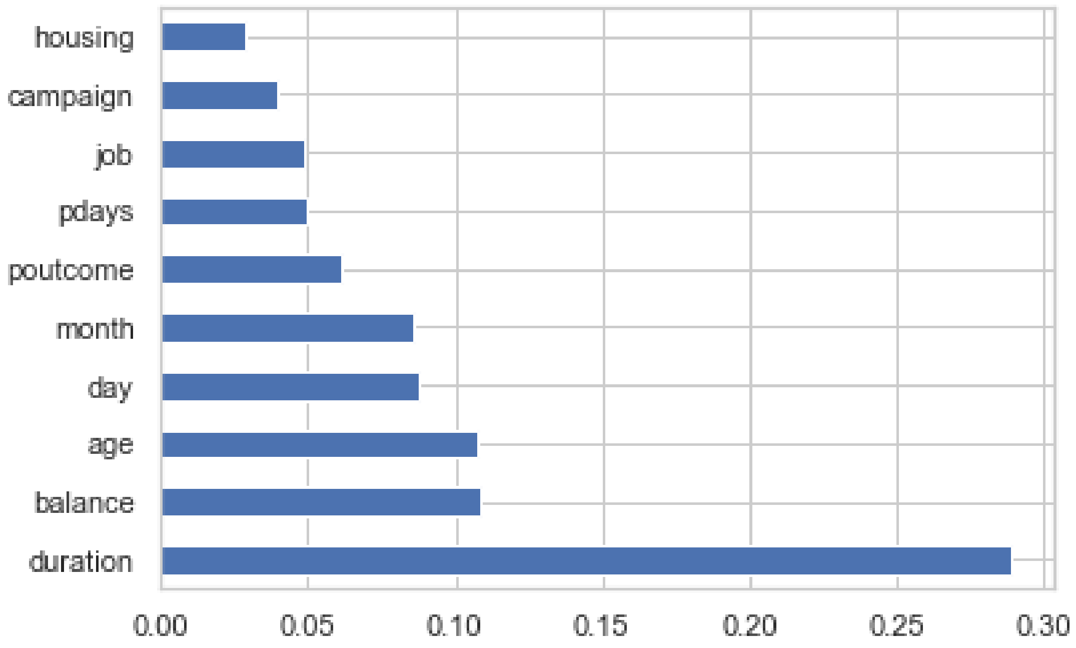
## Difficulty during implementation and possible complications

- Accuracy, training and testing times and F-beta score with beta = 0.001 is used because sklearn fscore metrics doesn't accept beta = 0.
- Function "train_predict" and "evaluation_graphs" are the core of the process where multiple classification algorithms will be tried for training and predicting. Both of these functions are related like output of "train_predict" will be used as input for "evaluation_graphs". If anything changes in one function can impact functionality of other function. So, one must be careful while changing these functions.
- During coding, algorithm was planned but while coding lot of too and for changes were done in the mentioned 2 functions to make it work.
- Any changes in the above mentioned functions can bring whole system on halt.

## Feature Selection

As per feature importance of best Random Forrest classifier, we have top  features which are covering major weightage. So, we just need to use only top features. This will reduce the training and prediction time significantly.

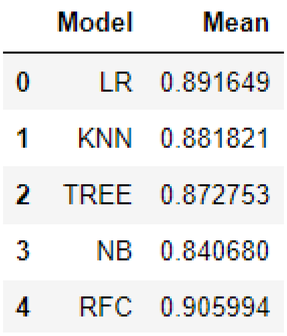
Following is the graph of feature importance.



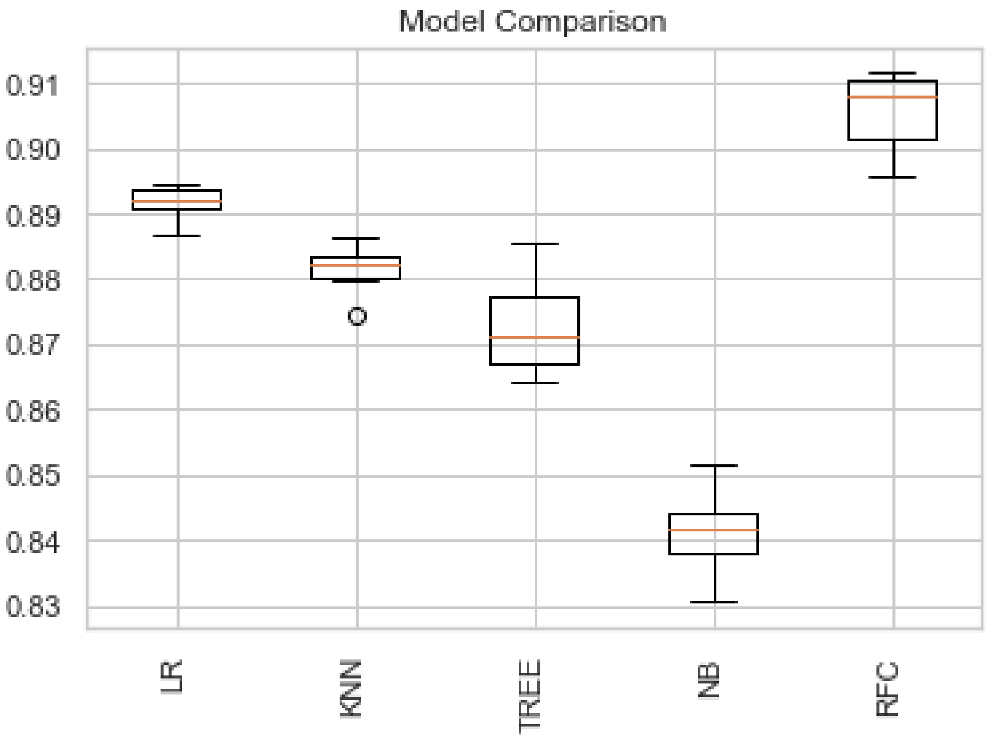
# Results

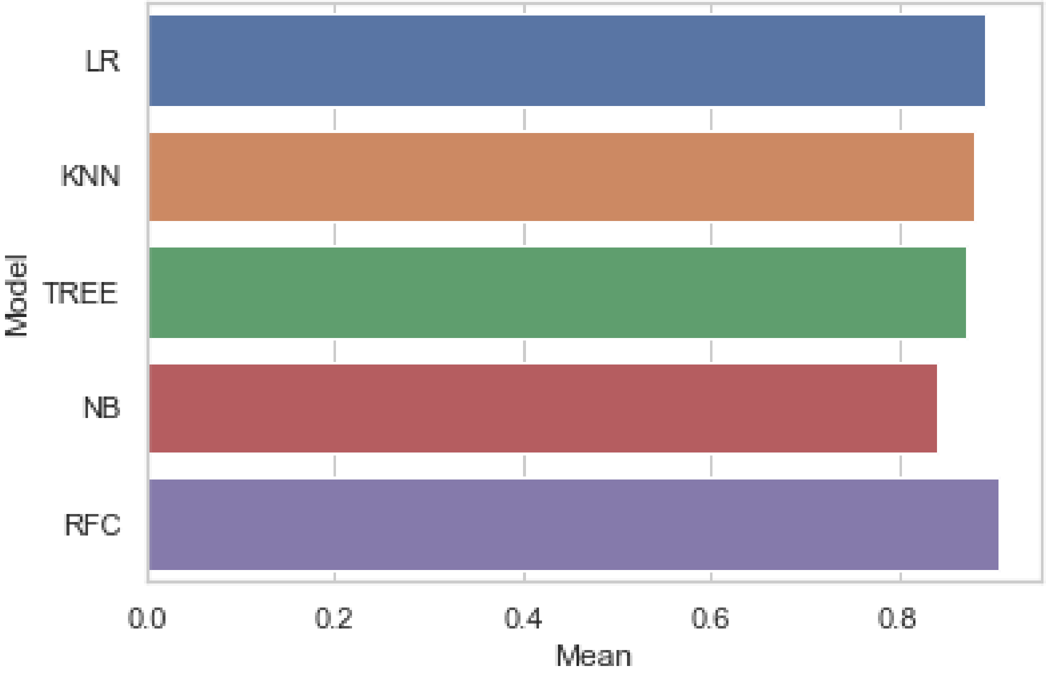
## Model Evaluation and Validation

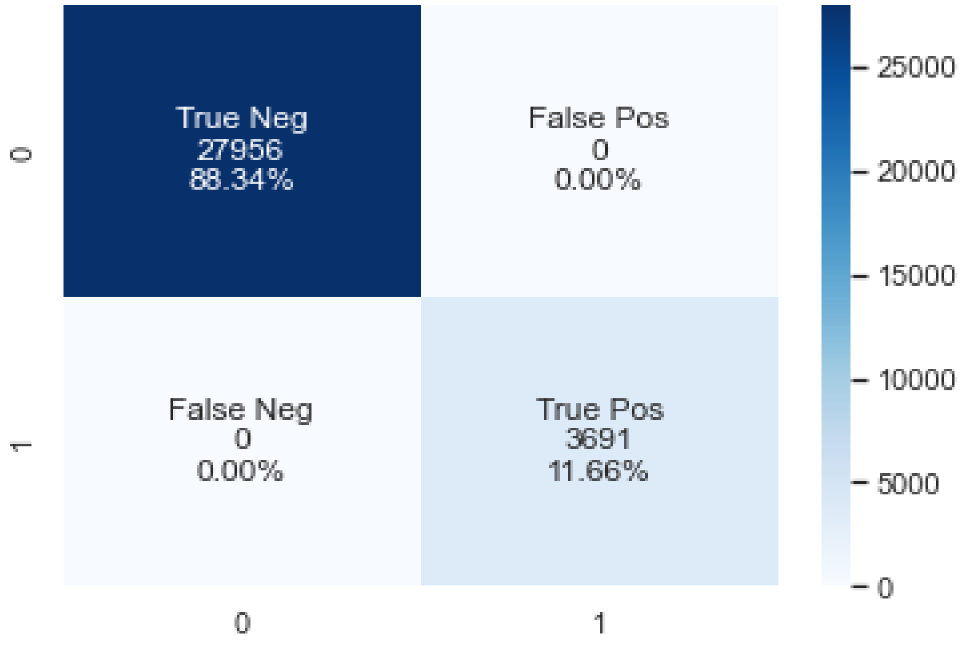
Based on the model evaluation made . we can say the Rando Forrest Classifier provides the best accuracy



|   | Model | Mean |
|---|---|---|
| 0 | LR | 0.891649 |
| 1 | KNN | 0.881821 |
| 2 | TREE | 0.872753 |
| 3 | NB | 0.840680 |
| 4 | RFC | 0.905994 |


Model Comparison
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
0.83
LR
KNN
TREE
NB
RFC


LR
KNN
TREE
NB
RFC
Model
0.0
0.2
0.4
0.6
0.8
Mean



Final model has significantly solved the problem because earlier F-score was very low and telemarketing team needs to call each customer. Now with this model, outcome can be predicted and only potential customers who have high probability of buying the term deposit will be called which will surely improve the productivity of the team.

# Conclusion

Based on the data set provided, Random Forrest Classifier is the best model which an be applied to get the best accuracy.

## Reflection

Following is the summary of the entire project:

- Data was acquired and loaded in the python.
- Exploratory data analysis was done, and features were analyzed.
- Data pre-processing and feature engineering was also done.
- Multiple models were tried, and confusion matrix, accuracy and F score was checked on the validation and test set.
- Finally, best model among tried models was selected.
- Grid search and Random cv was used to further improve the model by selecting the best hyper-parameter.
- Features were reduced by selecting top features and accuracy was checked on reduced feature models.
- K-fold cross-validation was used to check the model consistency and it was found that model generalized well.

Difficult part of this project was feature selection as features were not very strongly impacting the outcome.

Final model solves the problem and can be used for these kinds of issue provided features and distribution of features remains same. We can’t fit same model to another problem be it similar type of issue or not.

### Improvements

There is always scope of improving a model and same is true with final model as well. Following are some of the ways which can further improve the model:

- Feature engineering is one aspect which can be used to generate new features and further improve this model.
- Feature selection is another aspect which can be used to further improve the model after feature engineering.
- Hyper-parameter and base_estimator related changes in pipeline model can also be used to further improve this model.

# References

- Google Searches
- Towardsdatascience (Medium.) articles
- Stackoverflow for quick fixes