PREDICTION OF CUSTOMERS SUBSCRIBING TO TERM DEPOSITS IN A BANKING INSTITUTION

Post Graduate Program in Data ScienceEngineering

Location:
Chennai

Batch:
PGPDSE-FT-JUL'22

Mentored By:
Ms. Anjana Agrawal

SUBMITTED BY:
Anupam Dash
Ishwarya. B
Ranjith Kumar. A
Suganesh. R
Sundar Rajan Seshadri

Table of Contents

1. INTRODUCTION

- Term deposits are conventionally known for their preference for safe investments. The people in general, are risk avert and are interested in saving money and investing in low-risk investments.
- Nowadays, term deposit is considered to be the main business of banks but due to the rapid development of the Internet which has a great impact on the financial service products of banks.
- Rate of Return (ROR) and Interest margin are one of the key factors for the customers to choose and continue with the types of term deposits available. With respect to the business perspective of the deposits it's an indispensable factor. They come with less risk factor as they are not influenced by market fluctuations likewise other investments.
- Main objective of this project is to find the features/variables that affect the customers with respect to subscribing to term deposits. The aim of this project is focus on the customers who will subscribe the term deposits and to maximize the positive effect on the business.

1.1. Dataset Information

- This dataset contains telephonic marketing survey of customers with 17 variables describing the 45211 observations of customers who are already existing customers of the bank and the survey was to gain insights on whether the customers will predict to their term deposits or not.

1.2. Problem Statement

- For any business to successful, customer retention is important. With current rate of 78% being the retention rate, customer retention is much more an effective way than that of finding new customers.

- The bank collects feedback from its existing customers, they manuallycollect the feedback through telephonic marketing whether the customer will subscribe to the term deposits or not and by the timethe end results are predicted by machine learning model, we can easily sort out the issues that led to negative fallbacks of the policy.

1.3. Variable Categorization with Description

- The dataset consists of 17 variables. Out of these variables 16 are independent variables and 1 is a target variable. The variables are amixture of both numerical and categorical type.

1.3.1. Numeric Variables:

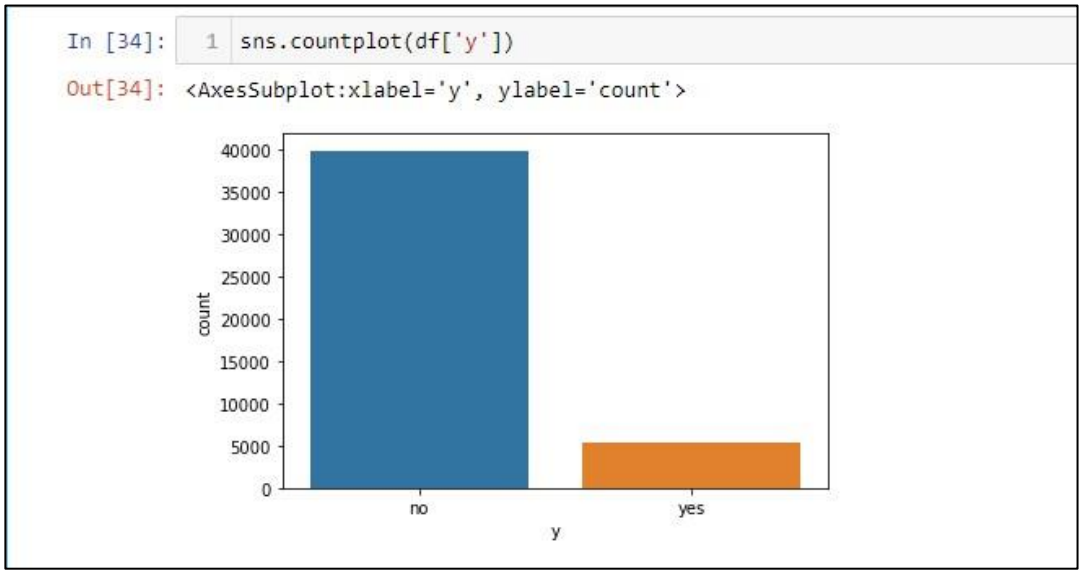
Sr No.	Variable	Datatype	Description
1	Age	int64	Age of the customers
2	Balance	int64	The actual bank balance of the customers (given in yearly basis)
3	Day	int64	The number of days that passed by after the client was last contacted from a previous campaign
4	Duration	int64	The last contact duration, in seconds (numeric).
5	Campaign	int64	The number of contacts performed during this campaign and for this client (numeric, includes last contact).
6	P-Days	int64	The number of days that passed by after the client was last contacted from a previous campaign.
7	Previous	int64	The number of contacts performed before this campaign and for this client.

1.3.2 Categorical Variable:

Sr No.	Variable	Datatype	Description
1	Job	object	Describes the type of job
2	Marital	object	Marital status of the customer
3	Education	object	Educational qualification of the customer
4	Default	object	The number of contacts performed before this campaign and for this client.
5	Housing	object	Contains information about whether the customer has housing loan or not.
6	Loan	object	Contains information about whether the customer has personal loan or not.
7	Contact	object	It gives the contact communication type.
8	Month	object	This feature contains information about the last contact month of year.
9	P-Outcome	object	It gives information about the outcome of the previous marketing campaign.
10	Y (Target Variable)	object	Gives information on whether the client has subscribed to term deposit or not.

1.4 Target Variable

The target variable of the above dataset is ‘Y’. We have to predict whether the customer has subscribed to the term deposits or not.



INFERENCE:

We find that around 80% of the customers have said ‘no’ to subscribing to term deposits whereas only 10% have said ‘yes’ to subscribing to term deposits. We can observe **a huge amount of class imbalance.**

2. DATA PRE-PROCESSING

- Data pre-processing is a process of preparing the raw data and makingit suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this we use data pre-processing task.
- A real-world data generally contains noises, missing values, and maybein an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning thedata and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.
- The data consists of 45211 rows and 17 columns. Out of these we have7 categorical columns and the rest as numerical.

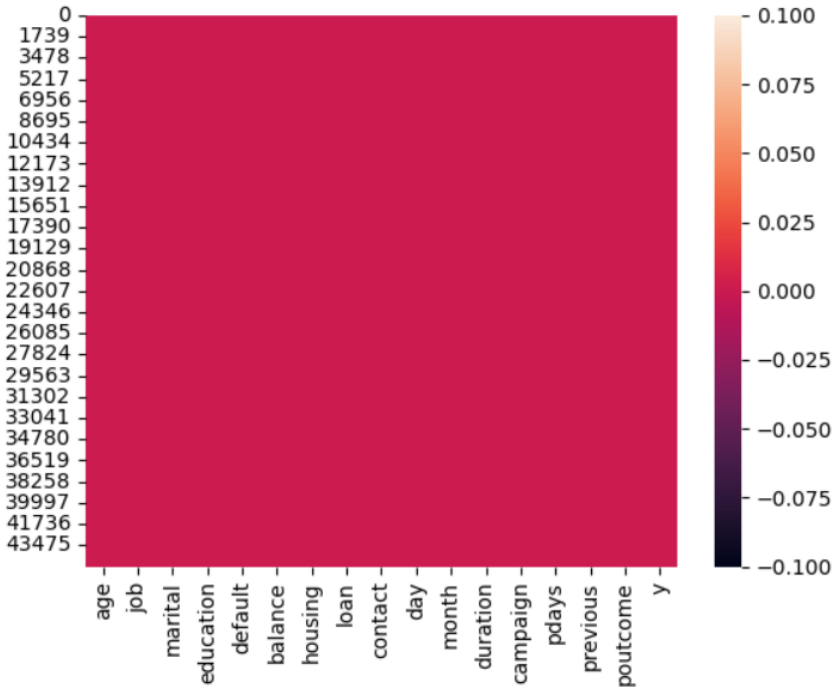
2.1. Datatype Verification

We first check the data types of each of the columns of the data.

Variable	Datatype
Age	int64
Job	object
Marital	object
Education	object
Default	object
Balance	int64
Housing	object
Loan	object
Contact	object
Day	int64
Month	object
Duration	int64
Campaign	int64
P-Days	int64
Previous	int64
P-Outcome	object
Y	object

2.2. Missing Value Treatment

- The next step of data pre-processing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.
- While checking out for null values, it's found out that the dataset had no null values.



INFERENCE:

From the above heatmap, we can infer that the dataset has no null values.

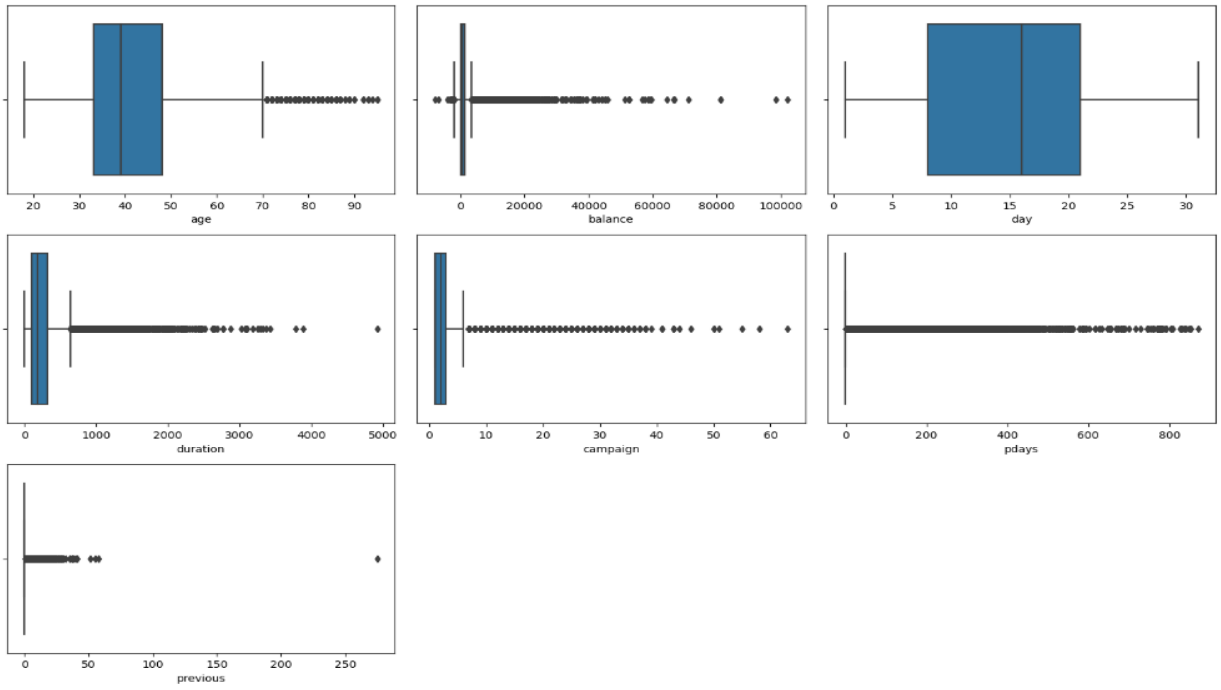
2.3. Redundant Features Removal

- Checking and removal of duplicate rows is important because presence of duplicates can lead us to make incorrect conclusions by leading us to believe that some observations are more common than they really are.
- From the 5-point summary of the data, we observe that there are no duplicate rows or any such redundant features present in the data.

- We finally have data which has 45211 rows and 17 columns as the data does not have any duplicate values or other features.

2.4. Check for Outliers

- Data has outliers present in the numerical columns. For making the base model, we do not perform any outlier treatment and retain all therows present in the data.



INFERENCE:

All the columns such as Age, Balance, Duration, Campaign, P-Days, Previous have heavy outliers in them as per the data analysis. Its necessary to remove all the outliers to balance the data going forward.

3. EXPLORATORY DATA ANALYSIS

3.1 Numerical & Categorical Columns

```
a = df.select_dtypes(include='O').columns
b = df.select_dtypes(exclude='O').columns
print(f'There around {len(a)} categorical columns')
print(f'There around {len(b)} Numerical columns')
```

```
There around 10 categorical columns
There around 7 Numerical columns
```

Target Variable

```
In [11]: # target column
         df.y.value_counts()

Out[11]: no      39922
         yes      5289
         Name: y, dtype: int64
```

Inference:

In this dataset the target variable is the column 'Y', this column it shows the count of the client's response of 'yes' or 'no' for the subscription of term deposits. The count of 'no' response is higher than that of 'yes' response which is 39922 and 5289 respectively

3.2 Checking for Null Values

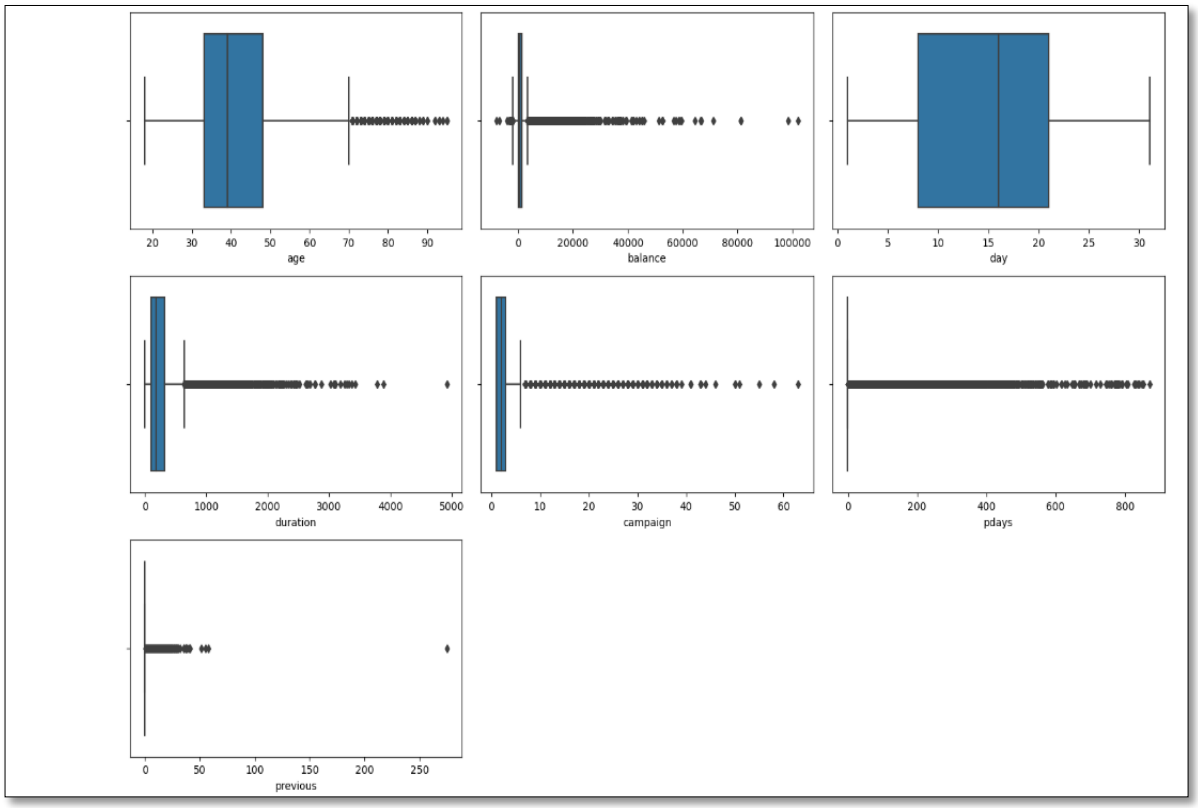
```
In [10]: df.isnull().sum() # we dont have any missing values in the dataset

Out[10]: age          0
         job          0
         marital      0
         education    0
         default      0
         balance      0
         housing      0
         loan         0
         contact      0
         day          0
         month        0
         duration     0
         campaign     0
         pdays        0
         previous     0
         poutcome     0
         y            0
         dtype: int64
```

Inference:

We can see that from the above image attached, the dataset has no null values in any of the columns.

3.3 Outlier Treatment



Inference

We can observe that majority of the columns have outliers in them. Columns such as age, balance, duration, campaign, p-days, previous have outliers.

Count of Outliers

```
In [30]:
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).sum()

Out[30]: age          487
         balance     4729
         campaign    3064
         contact        0
         day            0
         default        0
         duration    3235
         education      0
         housing        0
         job            0
         loan           0
         marital        0
         month          0
         pdays       8257
         poutcome       0
         previous    8257
         y              0
         dtype: int64
```

Inference

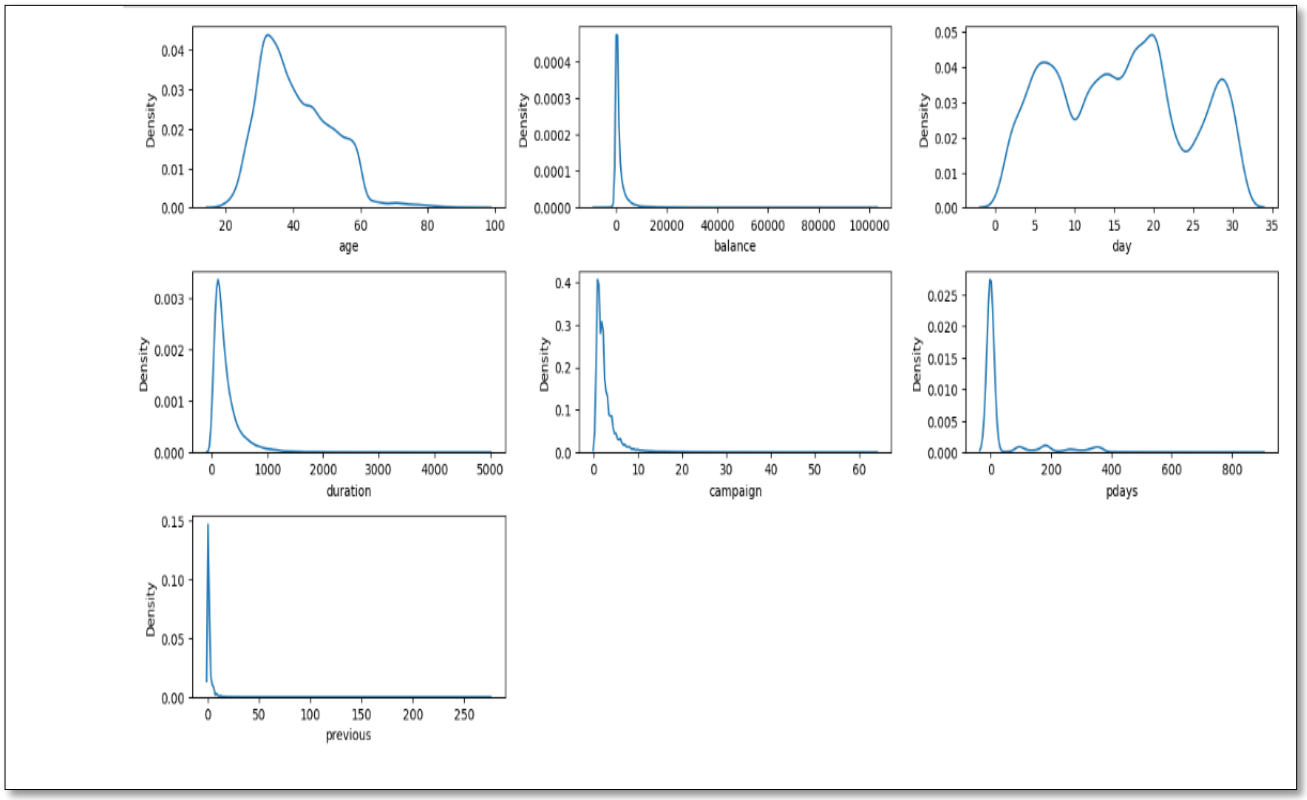
The count of outliers in few features such as Age, Balance, Campaign, Duration, P-Days, Previous can be seen.

Note:

Here in this dataset, we have not treated/removed the outliers as with respect to bank every client is important and hence considering them as outliers and removing them may not be appropriate for the business and to proceed with the data as well

3.4 Relation between Variables

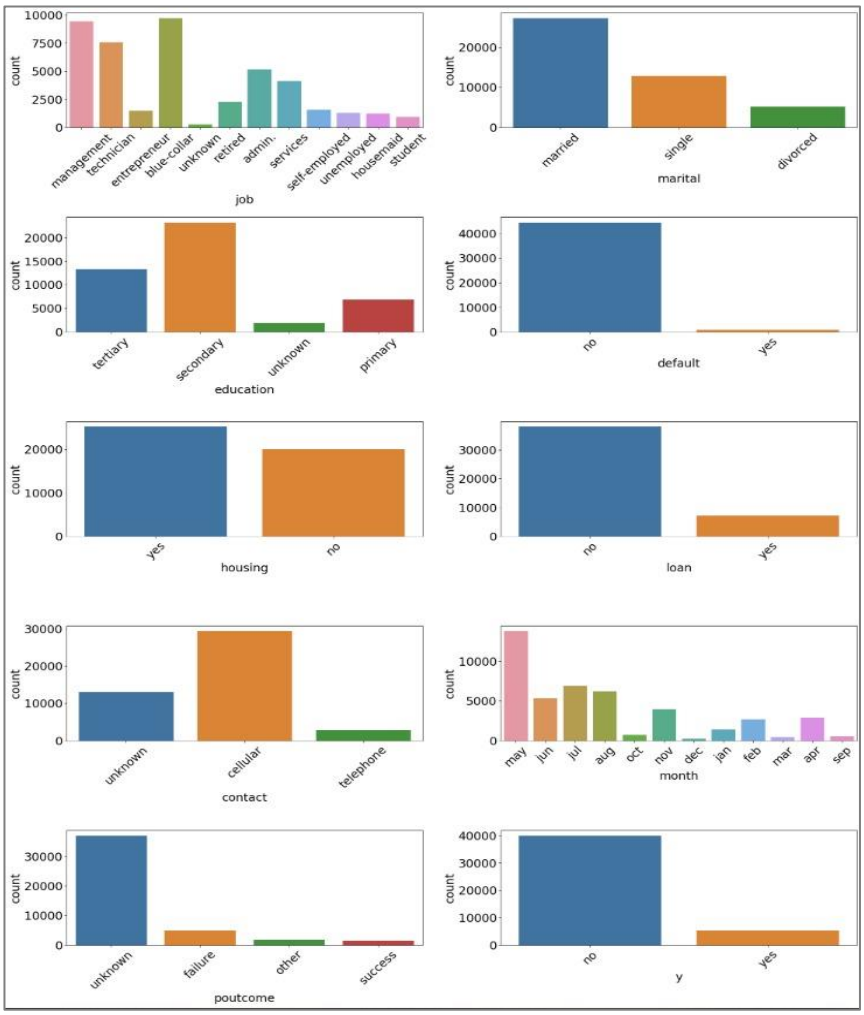
3.4.1 Univariate Analysis on Numerical Variables



Inference

- Age – 75 % of the people are the age between 20 to 50
- Balance – Average balance in most of the people rs.3000 only
- Day – All the days in the month are equally distributed
- Duration – Average call duration between 400 seconds
- Campaign – Average of 3 calls contacted per person
- Pdays – Most of the people are not contacted
- Previous – Most of the people are not contacted

3.4.2 Univariate Analysis on Categorical Variables

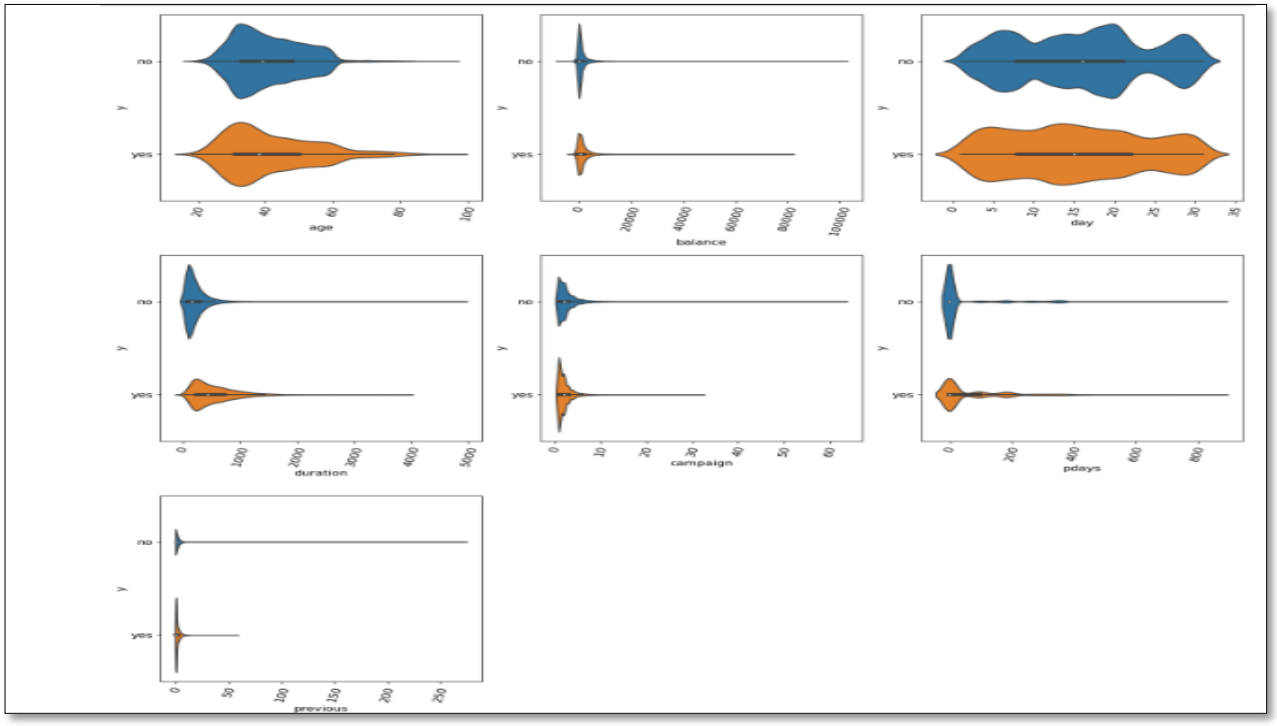


Inference

- job -blue collar and management have high working data when compared to other jobs.
- Marital - married are high than single and widowed.
- Education - most people are secondary educated.
- Default - most of the people do not has credit default.
- Housing - people partially have less housing loans.
- Loan -most of people do not have loans.
- Contact - bank has contacted on cellular than other modes.
- Month - May is Last contacted month for most peoples.
- P-outcome - Previous outcome has high unknown results.
- Y - most of the people said no for term deposits

3.4.3 Bi Variate Analysis based on Target Variable

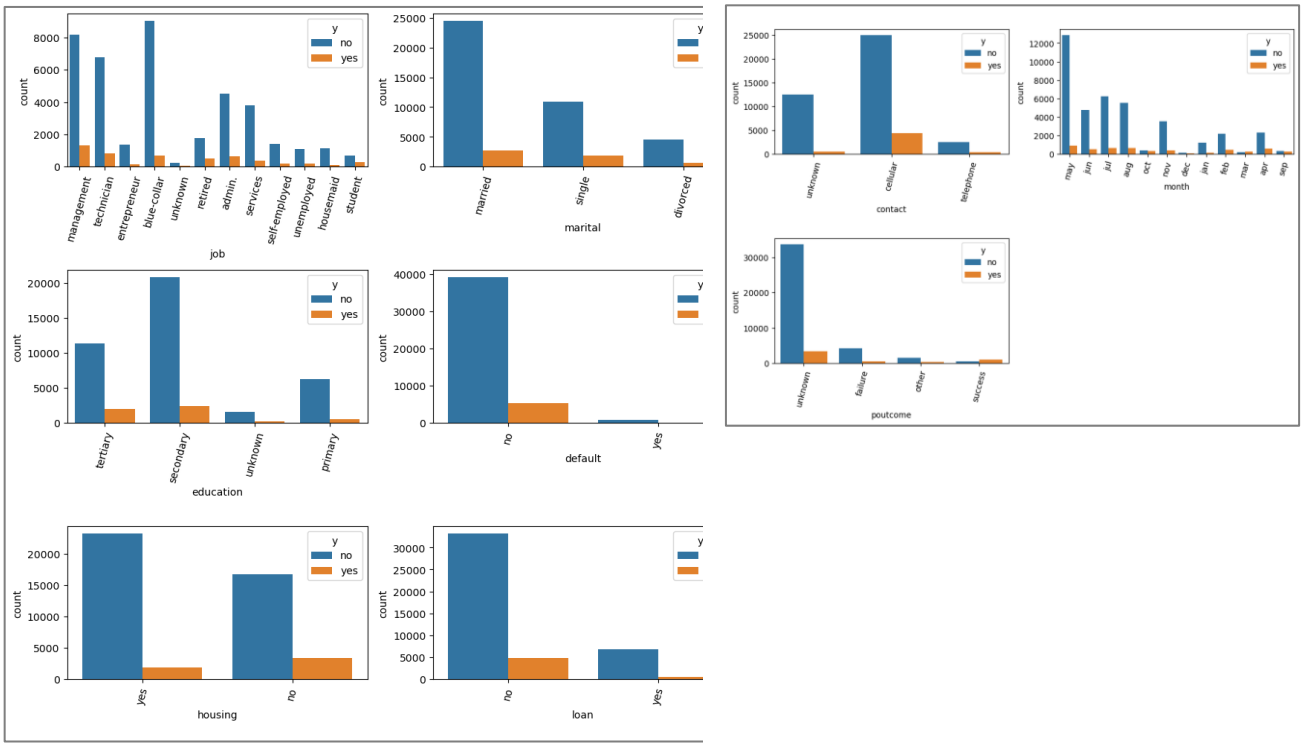
Relationship between Numerical variables and Target variable



- **Inference**

- The average age between 25 to 40 have more no of customers with both ‘yes’ and ‘no’ responses.
- The average bank balance of customers between 0 to 3000 have recorded both ‘yes’ and ‘no’ responses.
- The day between 1 to 32 are equally distributed for both ‘yes’ and ‘no’ responses.
- The average duration of calls of customers between 0 to 1000 have said ‘yes’ to the subscription and average duration of calls of customers between 0 to 600 have said ‘no’ for the subscription.
- The average marketing campaign (telephonic marketing) for a person is around 3 calls.

3.4.4 Relationship between Categorical Variables and Target Variable



Inference:

- In job category, management, technology, blue-collar have highly subscribed to term deposits.
- In marital category both married and single have subscribed more in numbers.
- In the education category both secondary and tertiary sub-categories have subscribed more in numbers than to those working in management & blue-collar categories.
- In default category most of the customers have not subscribed.
- In housing category most of the customers with no housing loan have subscribed more in numbers.
- In loan category the customers with no loan have subscribed more.
- In contact category the customers with cellular phone have subscribed more.

3.4.5 Categorical vs Categorical

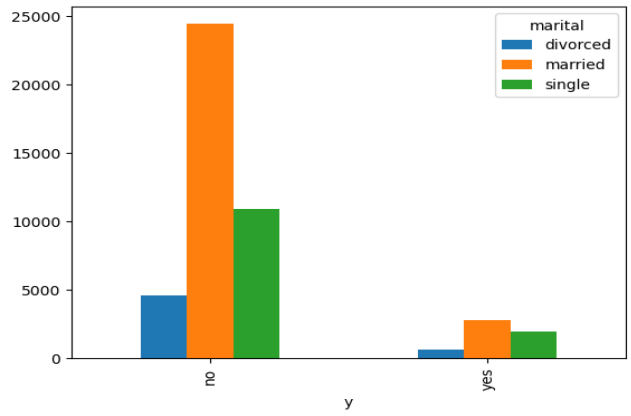


Fig 1. Marital w.r.t to target

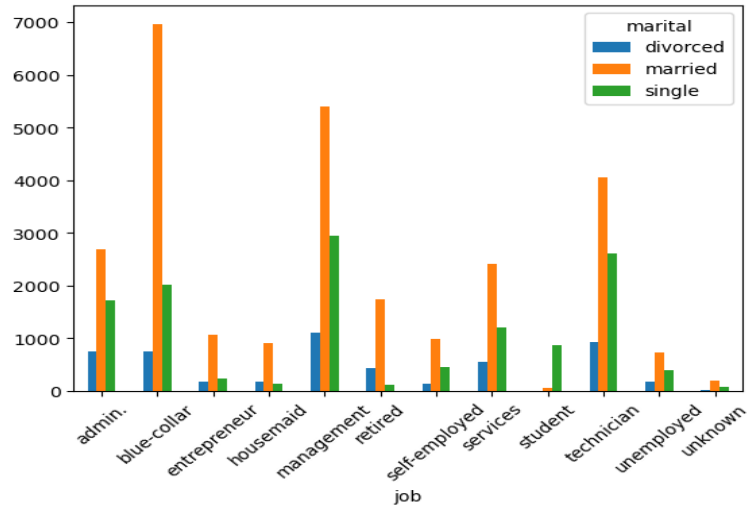


Fig 2. Marital w.r.t to Job

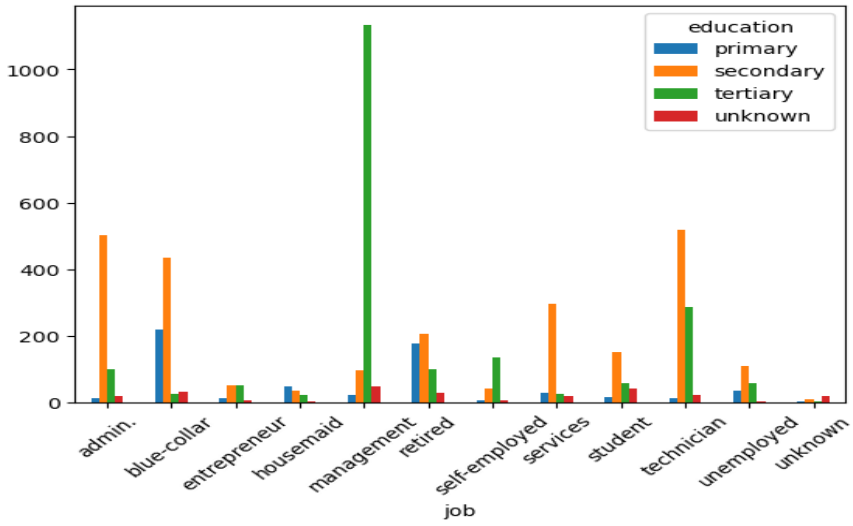


Fig 3. Job w.r.t to Education

1. The married people 60.19%, single 28.29%. About 90% people is married and single only 10% is divorced

2.From Fig 1 we can infer that the married and single people most people who subscribed to term deposit

3. From Fig 2 the married and single people are working in the admin, blue-collar, management, services, technician. So, we should focus our campaign on these kind jobs working people.
In this management and blue collar are high in number

4. From Fig 3 people working on the above job categories have completed a minimum of secondary education. If a person has completed secondary education, they are more likely to subscribe to term deposits

5. From our 39922 customers who said no have, if they have completed up to secondary education, they are more likely to subscribe to term deposits.

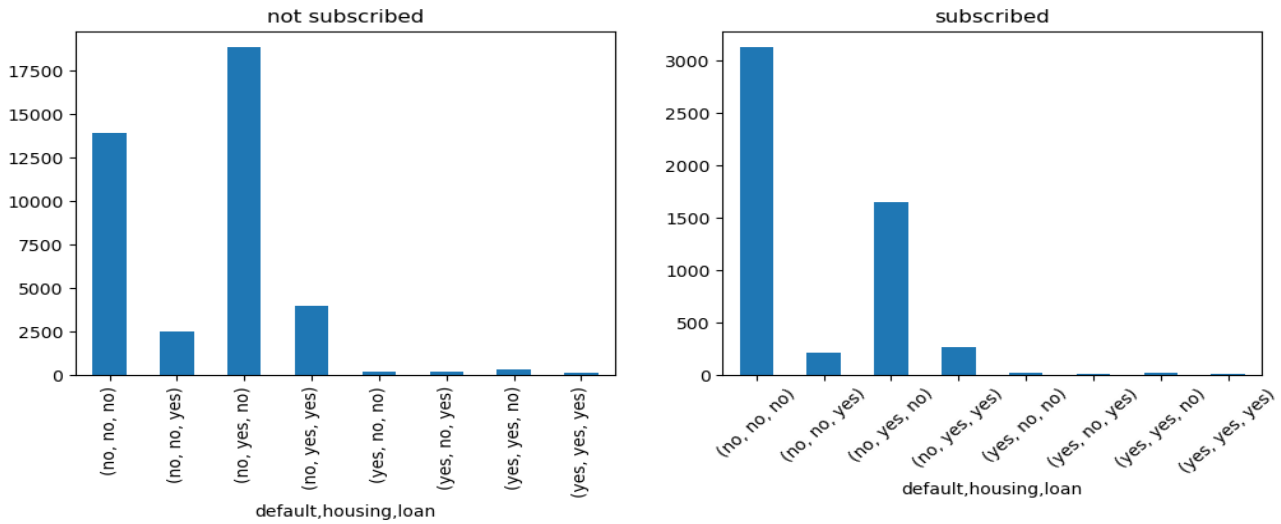


Fig 4. Loans w.r.t to subscribed and mot subscribed

From this Fig 4 we infer that people

- who don't have credit, housing and personal loan are more likely to subscribe to term deposits,
- who don't have credit, personal but have housing are also more to subscribe
- If customer who has credit default, they not subscribing to term deposits, even if they don't have housing or personal loan.
- We should focus on customers who are not having credit default will subscribe to term to deposits

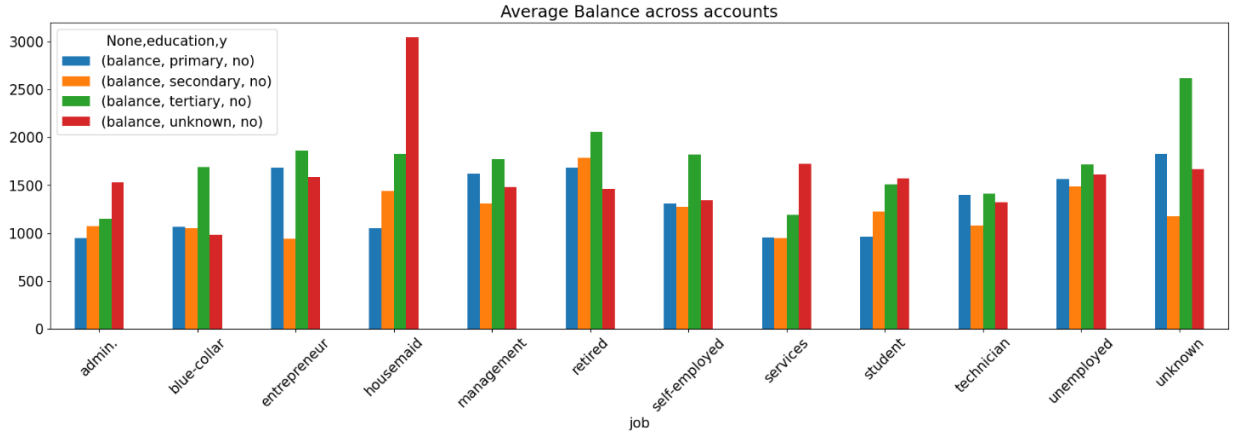


Fig 5. Balance of each job and education categories w.r.t not subscribed

From this Fig 5

- The customers who completed secondary and tertiary has a very high bank balance
- They are more customers we should concentrate to improve our term deposits subscriber count

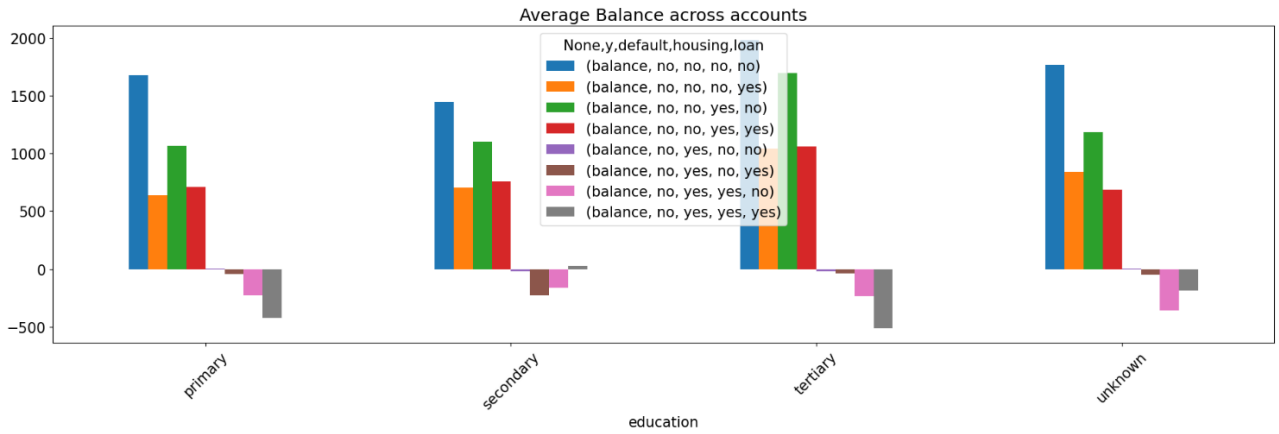


Fig 6. Balance of each job and education categories w.r.t not subscribed

From this Fig 6

- The customers who don't have no loan having a good balance.
- The customers having personal, housing and credit having negative balance.
- The people having less no of loans is our target customers for increasing our subscribers.

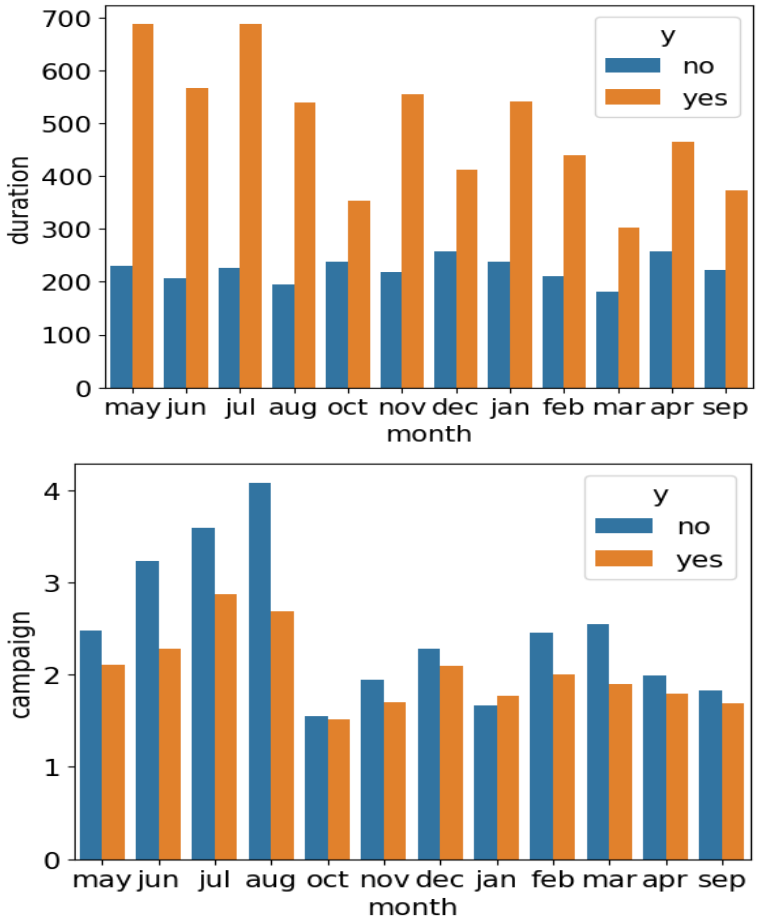


Fig 7. Month w.r.t to duration & Campaign

1.From the Figure 7 we infer that increase in number of duration and number of calls have led to more subscriber.
We can see that people have less call duration and a smaller number of calls have been subscribed less to our term deposits.

2.We should increase our focus on more calls and the calls must be very effective to increase our subscriber for term deposits. Contact people in May, July month has an increased our subscriber counts. So, we should focus on these months for campaign

3.4.6 Numerical vs Numerical

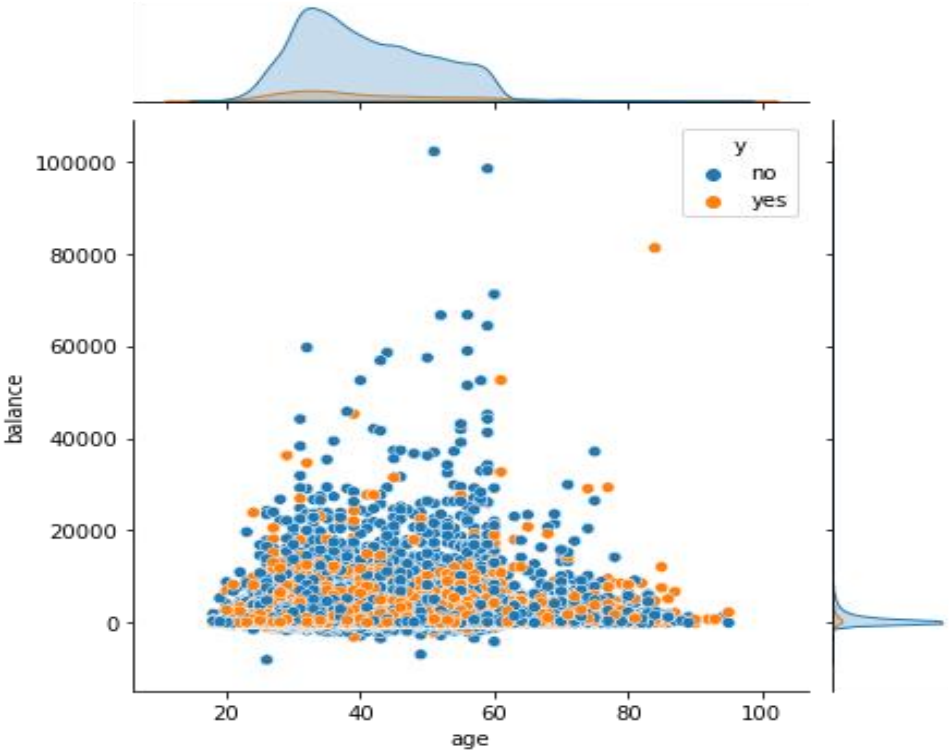


Fig 1. Age w.r.t to balance

1.From Figure 1 we can say that customers with age between 20 to 60 with high balance said no for term deposits and we can target those customers for subscribing term deposits. Their balance is also very high and they have not been subscribed for term deposits.

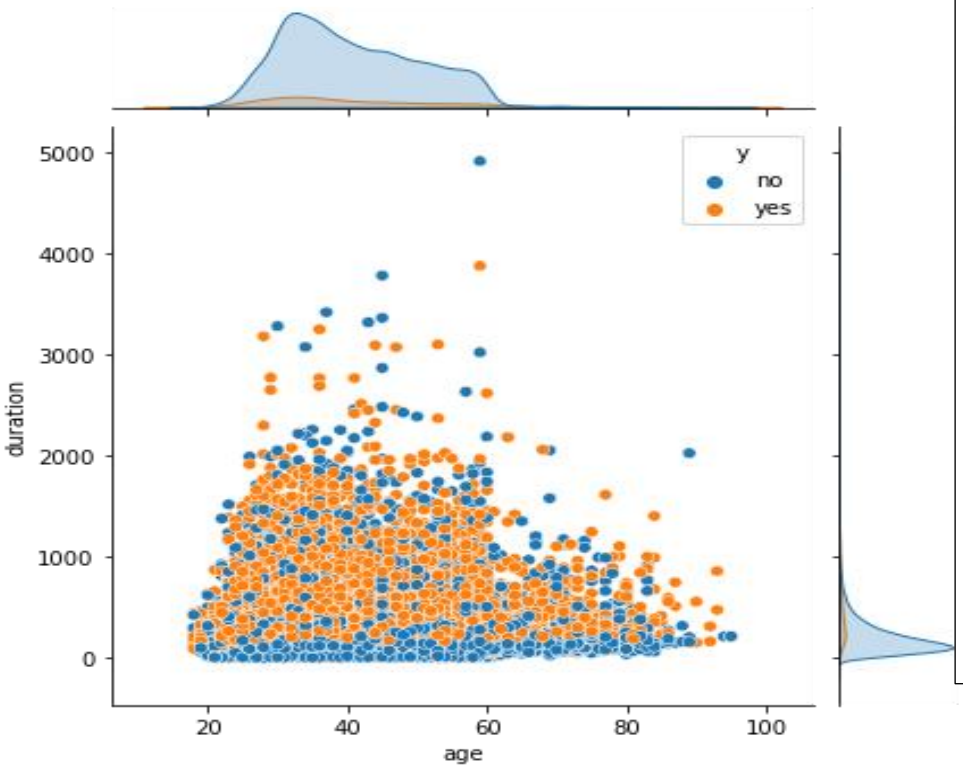


Fig 2. Age w.r.t to duration

2.From figure 2 this graph we can say that customers between age 20 to 60 have maximum duration on conversation and said yes for the term deposits. Who has less call duration have said no for the term deposits.

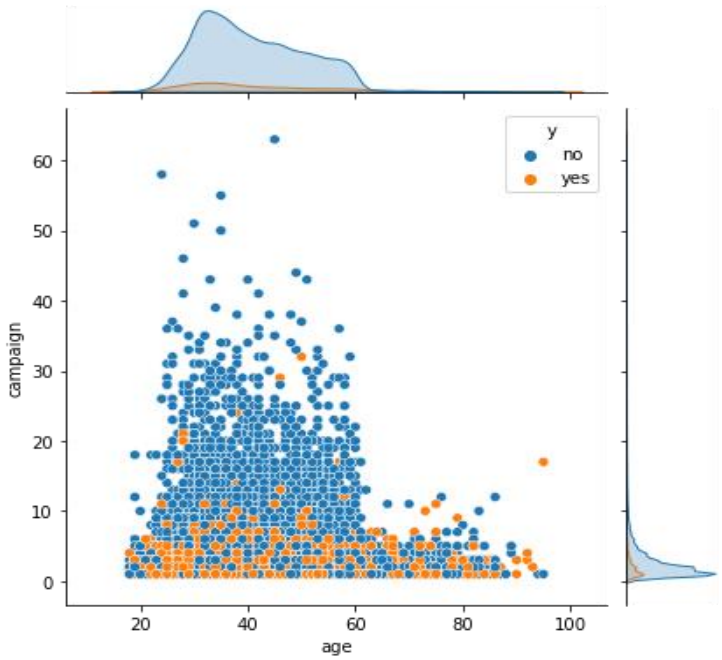


Fig 3. Age w.r.t to campaign

From this Figure 3 we can say that customers age between 20 to 50 said no on the campaign (including previous contacted).

People with more campaign calls not subscribed for term deposits, the max calls for customer is 15 calls. This is the amount calls will decide customer subscribed to term deposit or not. more that are customer not likely to subscribe

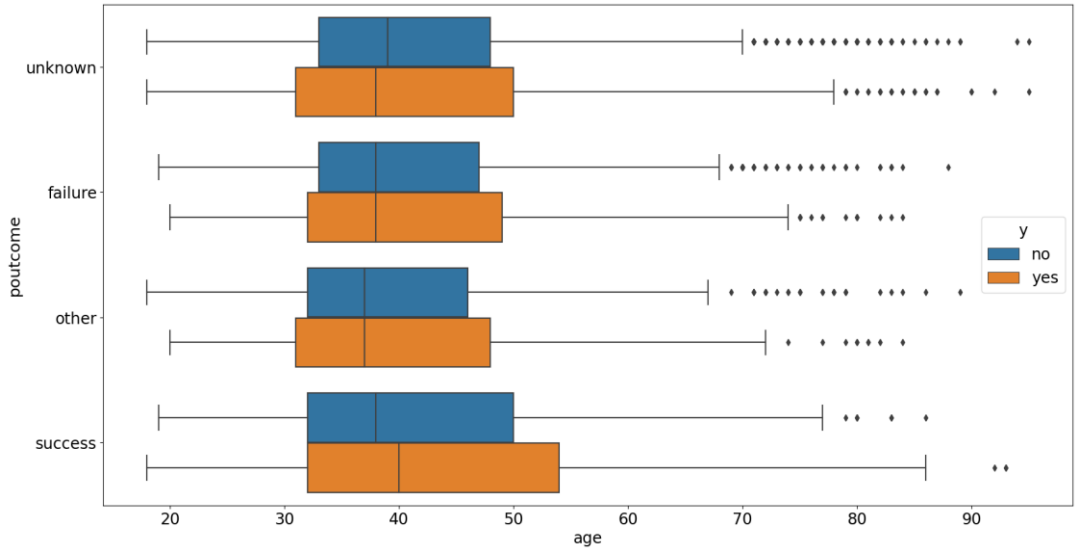


Fig 4. Age w.r.t to pout come

- From the above plot we can say that success on previous campaign outcome for age between 32 and 54 has good counts for term deposit subscription whereas on failure customers age range between 32 to 49 said no for term deposits subscription.
- Results like unknown and other has the customers age range between 31 to 50. we need to target these customers again to get result on the term deposit subscription

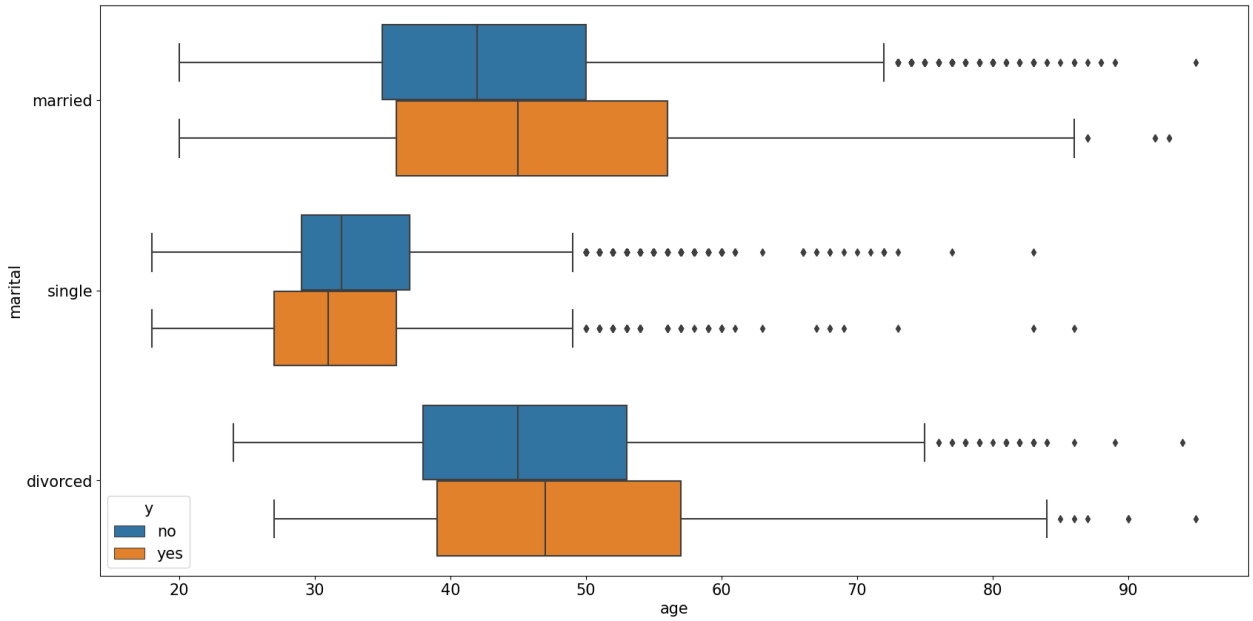


Fig 5. Age w.r.t to marital

- From the above plot we can say that customers who are married and divorced, age between 37 to 58 have subscribed to term deposits.
- We need to target customers married, divorced and single who were in range 25 to 50 to subscribe term deposits.

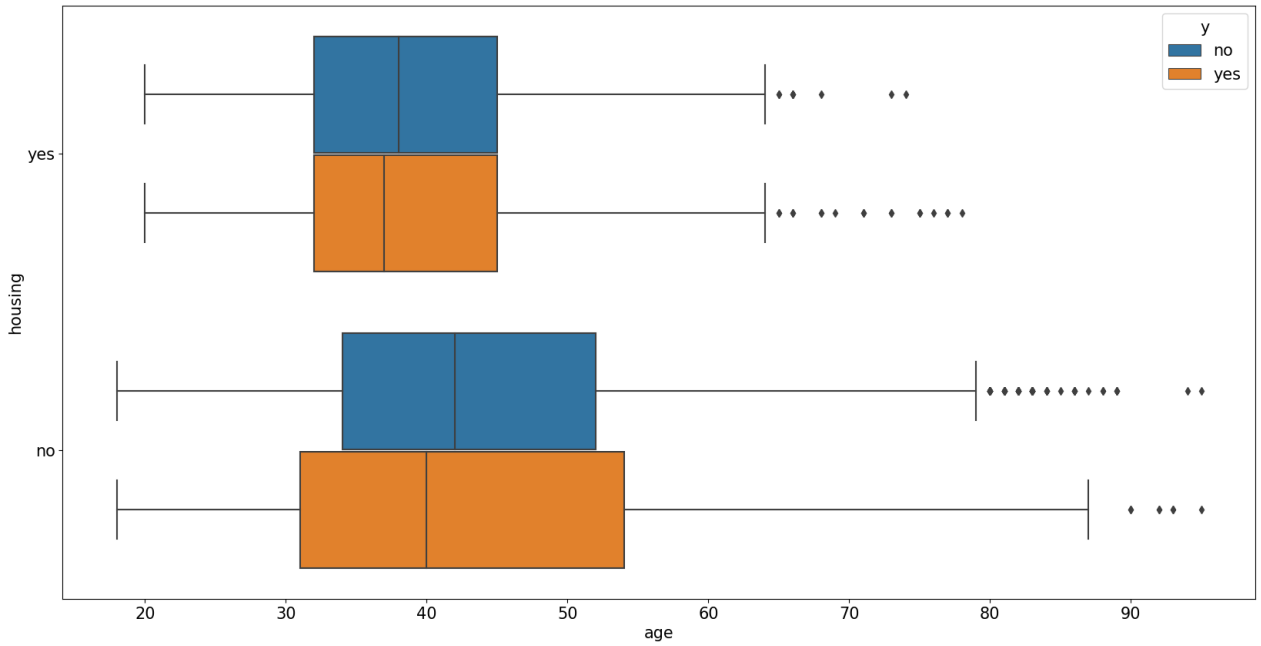


Fig 6. Age w.r.t to housing

- From this plot we can say that customers with age range 32 to 43 with housing loan said have not subscribed to term deposits and customers without housing loan said yes with age range between 31 to 54

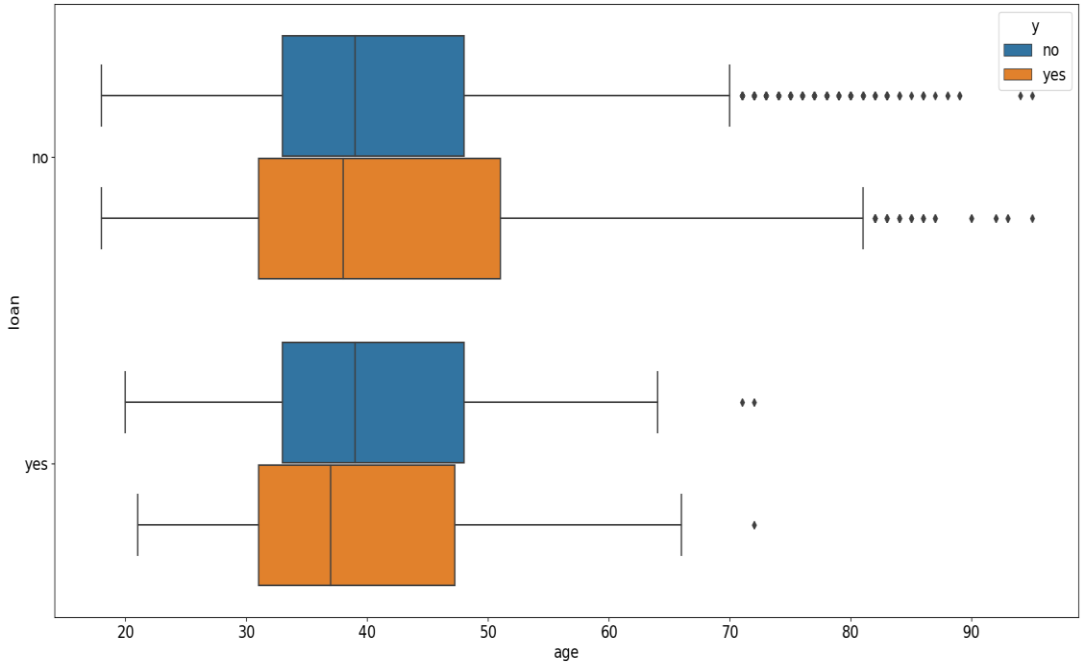


Fig 7. Age w.r.t to loan

From this plot we can say that customers with age range 32 to 51 with loan said have not subscribed to term deposits and customers without loan said yes for term deposits with age range between 31 to 47.

3.5 Correlation of Dataset

Inference

- On comparing with the target variable, there isn't much correlation between the variables.
- All the features are equally important to predict the target.
- From the above plot, we can conclude that p days, previous, duration has some correlation.

3.6 FEATURE ENIGNEERING:

- Age is continuous variable, from the above code, we split the age groups to
- (18 to 39) age customers are considered young, (40 to 60) age customers are considered as adult, (above 61) age customers are considered as old.
- Job category the categories are split into Unemployed (student, unemployed, retired and unknown), Blue-collared (blue-collar, technicians, services, housemaid and self-employed) and White-collared (management, admin and entrepreneur).
- Months are split into F1 it includes (jan, feb, mar), F2 includes (apr, may, jun), F3 includes (jul, aug, sep) and F4 includes (oct, nov, dec).
- In P-Outcome category 'unknown' is categorized as 'previous non-contact' due to category of the data not available.
- Per call time feature is the duration divided by number of calls.

3.7 ENCODING TECHNIQUE:

3.7.1 ONE HOT ENCODING:

- Marital and Contact are categorical variables(nominal), from the above code we have encoded the categorical variable into numerical variable (0 and 1) because machine doesn't understand the categorical variables.

3.7.2 LABEL ENCODING:

- Education, Marital, Target variable (Y), Default, Housing, Contact and Loan are categorical variables, from the above code we have encoded the categorical variables into numerical variables such as default, housing and loan into (‘no’:0,’ yes’:1), contact into (‘cellular’:0, ‘telephone’:1), marital into (‘married’:0,’single’:1), education into (‘primary’:2, ‘secondary’:1 and ‘tertiary’:0) and target variable(‘Y’) into (‘no’:0 , ‘yes’:1) to ease the understanding of the machine.

3.7.3 FREQUENCY ENCODING:

- Age, Job, P-Outcome, Month categories are converted into numerical variables using frequency encoding.

3.8 SCALING TECHNIQUE:

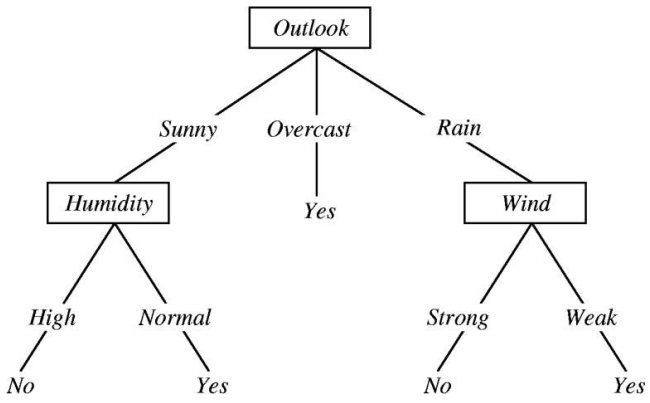
- The columns such as Balance, Duration, Day, Campaign, P-Days, Previous are scaled using techniques such as Standard Scaler
- After Scaling the model able to read well

4.MODEL BUILDING:

4.1 (Before Smote)

4.1.1 DECISION TREE ALGORITHM

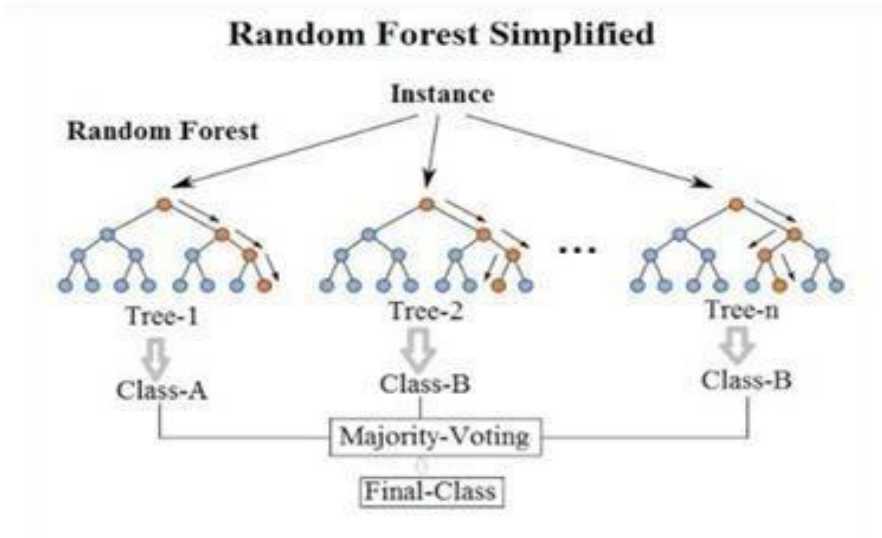
- A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g., whether a coin flip comes up heads ortails), each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of featuresthat lead to those class labels. The paths from root to leaf represent classification rules. Below diagram illustrate the basic flow ofdecision tree for decision making with labels (Rain (Yes), No Rain (No)).



- Decision tree is one of the predictive modelling approaches usedin statistics, data mining and machine learning.
- Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric **supervised learning** method usedfor both **classification** and **regression** tasks.
- Tree models where the target variable can take a discrete set of values are called **classification trees**. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**. Classification And Regression Tree (CART) isgeneral term for this.

4.1.2 RANDOM FOREST CLASSIFIER

- Random forests or random decision forests are an ensemble learningmethod for classification, regression and other tasks that operates byconstructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- To say it in simple words: Random Forest builds multiple decision treesand merges them together to get a more accurate and stable prediction.
- One big advantage of random forest is, that it can be used for bothclassification and regression problems.
- Random Forest has nearly the same hyper parameters as a decision treeor a bagging classifier.
- Fortunately, we don't have to combine a decision tree with a baggingclassifier and can just easily use the classifier-class of Random Forest. Like I already said, withRandom Forest, you can also deal with Regression tasks by using the Random Forest repressor.
- Random Forest adds additional randomness to the model, while growingthe trees. Instead of searching for the most important feature while splitting a node, itsearches for the best feature among a random subset of features. This results in a wide diversity thatgenerally results in a better model.



Advantages of Random Forest:

- There is no need for feature normalization
- Individual decision trees can be trained in parallel
- Reduced over fitting
- Require almost no input preparation
- Performs implicit feature selection
- It's very quick to train
- Modeling and Predicting Online Purchasing Intention of Shopper
- Disadvantages of Random Forest:
- No interpretability

4.1.3 K-NEAREST NEIGHBOURS

- The KNN algorithm assumes that similar things exist in close proximity.In other words, similar things are near to each other.

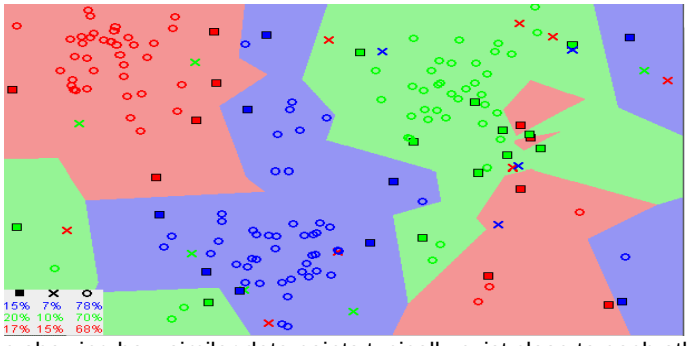


Image showing how similar data points typically exist close to each other

- Notice in the image above that most of the time, similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculatingthe distance between points on a graph.

- KNN's main disadvantage of becoming significantly slower as the volume of data increases makes it an impractical choice in environments where predictions need to be made rapidly. Moreover, there are faster algorithms that can produce more accurate classification and regression results

4.1.4 ADA BOOST CLASSIFIER

- AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called **Decision Stumps**
- What this algorithm does is that it builds a model and gives equal weights to all the data points. It then assigns higher weights to points that are wrongly classified. Now all the points which have higher weights are given more importance in the next model. It will keep training models until and unless a lower error is received.

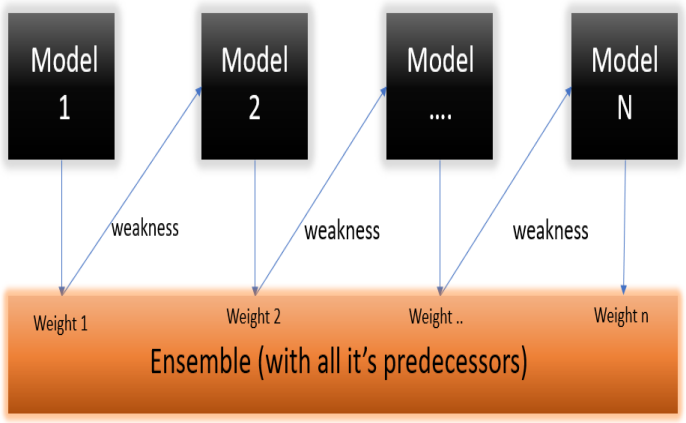
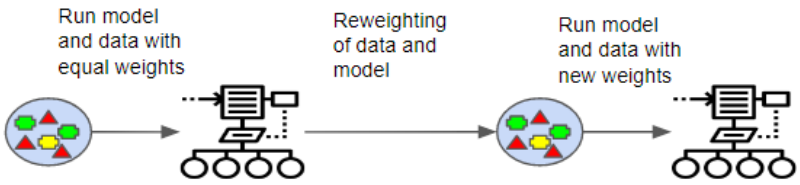


Image showing Working of Ada Boost Technique

4.1.5 GRADIENT BOOSTING CLASSIFIER

- Gradient boosting classifiers are specific types of algorithms that are used for classification tasks z, as the name suggests.
- Features are the inputs that are given to the machine learning algorithm, the inputs that will be used to calculate an output value. In a mathematical sense, the features of the dataset are the variables used to solve the equation. The other part of the equation is the *label* or target, which are the classes the instances will be categorized into. Because the labels contain the target values for the machine learning classifier, when training a classifier, you should split up the data into training and testing sets. The training set will have targets/labels, while the testing set won't contain these values.
- In AdaBoost, the predictions are made through majority vote, with the instances being classified according to which class receives the most votes from the weak learners.

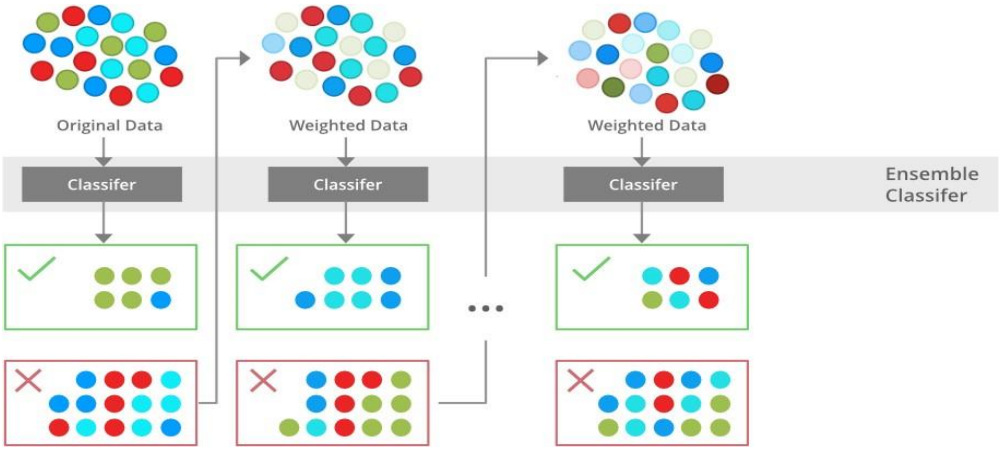


- Gradient boosting classifiers are the Ada Boosting method combined with weighted minimization, after which the classifiers and weighted inputs are recalculated. The objective of Gradient Boosting classifiers is to minimize the loss, or the difference between the actual class value of the training example and the predicted class value. It isn't required to understand the process for reducing the classifier's loss, but it operates similarly to gradient descent in a neural network.
- Refinements to this process were made and Gradient Boosting Machines were created.
- In the case of Gradient Boosting Machines, every time a new weak learner is added to the model, the weights of the previous learners are frozen or cemented in place, left unchanged as the new layers are introduced. This is distinct from the approaches used in Ada Boosting where the values are adjusted when new learners are added.

- The power of gradient boosting machines comes from the fact that they can be used on more than binary classification problems, they can be used on multi-class classification problems and even regression problems.

4.1.6 XGB CLASSIFIER

- Boosting is an ensemble modeling, technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built



- From the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continuedand models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

- In XG Boost, decision trees are created in sequential form. Weights playan important role in XG Boost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree isincreased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

4.1.7 LGBM CLASSIFIER

- A Gradient Boosting Decision tree is a very popular machine learning algorithm that has effective implementations like Boost and many optimization techniques are actually adopted from this algorithm. It uses two types of techniques which are gradient Based on side sampling or GOSS and Exclusive Feature bundling or EFB.
- It uses two types of techniques which are gradient Based on side sampling or GOSS and Exclusive Feature bundling or EFB. So, GOSS will actually exclude the significant portion of the data part which have small gradients an d only use the remaining data to estimate the overall information gain. The data instances which have large gradients actually play a greater role for computation on information gain. GOSS can get accurate results with a significant information gain despite using a smaller dataset than other models.
- Light is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

The main features of the LGBM model are as follows:

- Higher accuracy and a faster training speed.
- Low memory utilization
- Comparatively better accuracy than other boosting algorithms and handles overfitting much better while working with smaller datasets.
- Parallel Learning support.
- Compatible with both small and large datasets

(Before Smote)

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
7	XGBClassifier	0.953661	0.910096	7705	289	524	525	0.50	0.64	0.56
8	LGBMClassifier	0.929579	0.910870	7720	274	532	517	0.49	0.65	0.56
2	RandomForestClassifier	0.999972	0.905120	7751	243	615	434	0.41	0.64	0.50
6	GradientBoostingClassifier	0.908980	0.905894	7764	230	621	428	0.41	0.65	0.50
1	DecisionTreeClassifier	1.000000	0.875373	7436	558	569	480	0.46	0.46	0.46
5	AdaBoostClassifier	0.900852	0.898043	7745	249	673	376	0.36	0.60	0.45
4	GaussianNB	0.804274	0.806701	6802	1192	556	493	0.47	0.29	0.36
0	LogisticRegression	0.892363	0.889749	7787	207	790	259	0.25	0.56	0.34
3	KNeighborsClassifier	0.913045	0.885768	7776	218	815	234	0.22	0.52	0.31

- Before SMOTE
- From the above table we can clearly see how the models are performing
- The Xtreme gradient boosting, Light Gradient Boosting machine performed better with the F1 score of 0.56
- The Random Forest and Gradient Boosting performed with the F1 score of 0.50
- But Random Forest model is overfitting
- The decision tree got a f1 score of 0.46, Ada boost got a score of 0.45
- The gaussian Naïve bayes, Logistic Regression, K Nearest neighbors got a f1 score of 0.36,0.34,0.31
- Our target is highly biased it is hard for model to predict, so we are doing smote on our data

(After Smote)

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.947710	0.903572	7529	465	407	642	0.61	0.58	0.60
7	XGBClassifier	0.969979	0.903904	7603	391	478	571	0.54	0.59	0.57
6	GradientBoostingClassifier	0.915811	0.875705	7196	798	326	723	0.69	0.48	0.56
2	RandomForestClassifier	1.000000	0.888533	7437	557	451	598	0.57	0.52	0.54
5	AdaBoostClassifier	0.896752	0.860997	7118	876	381	668	0.64	0.43	0.52
0	LogisticRegression	0.851228	0.814110	6614	1380	301	748	0.71	0.35	0.47
3	KNeighborsClassifier	0.936999	0.816654	6668	1326	332	717	0.68	0.35	0.46
1	DecisionTreeClassifier	1.000000	0.852483	7170	824	510	539	0.51	0.40	0.45
4	GaussianNB	0.751926	0.627778	4916	3078	288	761	0.73	0.20	0.31

- We have done a Smote with minority sampling, it splits our target into 50% , biased target is converted to unbiased
- The Light Gradient Boosting performed better at a F1 score of 0.60
- In Base model after smote the maximum f1 score achieved is 0.60,0.57,0.56 by LGBM, XGB, Gradient Boosting
- The highest TP is achieved by Logistic Regression, Gaussian NB Model at a f1 score of 0.47,0.31

5. FEATURE ENGINEERING AND FEATURE EXTRACTION: (Per call Time)

```
df5['per_call_time'] = round((df1.duration/(df1.campaign*60)),2)
```

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.947429	0.903461	7528	466	407	642	0.61	0.58	0.60
2	RandomForestClassifier	1.000000	0.890302	7426	568	424	625	0.60	0.52	0.56
6	GradientBoostingClassifier	0.916186	0.873383	7179	815	330	719	0.69	0.47	0.56
7	XGBClassifier	0.969823	0.904235	7618	376	490	559	0.53	0.60	0.56
5	AdaBoostClassifier	0.895578	0.860334	7102	892	371	678	0.65	0.43	0.52
0	LogisticRegression	0.848566	0.814221	6612	1382	298	751	0.72	0.35	0.47
3	KNeighborsClassifier	0.938142	0.816654	6662	1332	326	723	0.69	0.35	0.47
1	DecisionTreeClassifier	1.000000	0.850824	7151	843	506	543	0.52	0.39	0.45
4	GaussianNB	0.784860	0.678425	5357	2637	271	778	0.74	0.23	0.35

- Per call time feature engineered column is added to smote data and checked for any improvement
- In Base model after smote the maximum f1 score achieved is LGBM (0.60), XGB (0.57), Gradient Boosting (0.56)
- Adding per call time feature, the f1_score of LGBM (0.60), GBM (0.56), XGB (0.56) the model performance remains the same

(Combining the Job category)

```
unemp = ['student','unemployed','retired','unknown']
blue  = ['blue-collar','technician','services','housemaid','self-employed']
white = ['management','admin.','entrepreneur']
```

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.948337	0.904567	7551	443	420	629	0.60	0.59	0.59
6	GradientBoostingClassifier	0.919193	0.878359	7224	770	330	719	0.69	0.48	0.57
7	XGBClassifier	0.968210	0.905562	7632	362	492	557	0.53	0.61	0.57
2	RandomForestClassifier	1.000000	0.890634	7470	524	465	584	0.56	0.53	0.54
5	AdaBoostClassifier	0.899524	0.867411	7171	823	376	673	0.64	0.45	0.53
0	LogisticRegression	0.854438	0.817317	6648	1346	306	743	0.71	0.36	0.47
1	DecisionTreeClassifier	1.000000	0.856685	7202	792	504	545	0.52	0.41	0.46
3	KNeighborsClassifier	0.936513	0.813557	6665	1329	357	692	0.66	0.34	0.45
4	GaussianNB	0.756562	0.634082	4972	3022	287	762	0.73	0.20	0.32

- In Base model after smote the maximum f1 score achieved is LGBM (0.60), XGB (0.57), Gradient Boosting (0.56)
- Adding the job category maximum f1 score achieved is LGBM (0.59), XGB (0.57), Gradient Boosting (0.57)
- After doing the model performance is reduced

(Binning the Age into different groups)

```
l=[]
for i in df1['age']:
    if (i>=18) & (i<=39):
        a='young'
        l.append(a)
    elif (i>=40) & (i<=60):
        a='adult'
        l.append(a)
    elif (i>=61) :
        a='old'
        l.append(a)
```

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.948713	0.904678	7554	440	422	627	0.60	0.59	0.59
6	GradientBoostingClassifier	0.917001	0.876921	7197	797	316	733	0.70	0.48	0.57
7	XGBClassifier	0.967646	0.903572	7602	392	480	569	0.54	0.59	0.57
2	RandomForestClassifier	1.000000	0.891297	7448	546	437	612	0.58	0.53	0.55
5	AdaBoostClassifier	0.897833	0.861772	7129	865	385	664	0.63	0.43	0.52
0	LogisticRegression	0.854062	0.815659	6630	1364	303	746	0.71	0.35	0.47
1	DecisionTreeClassifier	1.000000	0.858012	7217	777	507	542	0.52	0.41	0.46
3	KNeighborsClassifier	0.936451	0.813889	6667	1327	356	693	0.66	0.34	0.45
4	GaussianNB	0.778752	0.673781	5313	2681	269	780	0.74	0.23	0.35

- In Base model after smote the maximum f1 score achieved is LGBM (0.60), XGB (0.57), Gradient Boosting (0.56)
- Adding the age category maximum f1 score achieved is LGBM (0.59), XGB (0.57), Gradient Boosting (0.57)
- After doing that the model performance remains the same

(Binning of Month)

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.948619	0.905120	7536	458	400	649	0.62	0.59	0.60
2	RandomForestClassifier	1.000000	0.898706	7512	482	434	615	0.59	0.56	0.57
6	GradientBoostingClassifier	0.921433	0.883003	7288	706	352	697	0.66	0.50	0.57
7	XGBClassifier	0.969337	0.904899	7620	374	486	563	0.54	0.60	0.57
5	AdaBoostClassifier	0.900761	0.863651	7157	837	396	653	0.62	0.44	0.51
0	LogisticRegression	0.852669	0.813447	6608	1386	301	748	0.71	0.35	0.47
1	DecisionTreeClassifier	1.000000	0.860113	7211	783	482	567	0.54	0.42	0.47
3	KNeighborsClassifier	0.937469	0.813779	6651	1343	341	708	0.67	0.35	0.46
4	GaussianNB	0.752083	0.627447	4912	3082	287	762	0.73	0.20	0.31

- In Base model after smote the maximum f1 score achieved is LGBM (0.60), XGB (0.57), Gradient Boosting (0.56)
- Adding the month category maximum f1 score achieved is LGBM (0.60), XGB (0.57), Gradient Boosting (0.57)
- After adding the month, the overall model performance is increasing

(Adding all the features and checking for improvement)

```
best6.sort_values(by='F1 score',ascending=False)
```

	Model	Train_ACC	Test_ACC	TN	FP	FN	TP	recall	Precision	F1 score
8	LGBMClassifier	0.948932	0.906005	7549	445	405	644	0.61	0.59	0.60
7	XGBClassifier	0.970903	0.908548	7640	354	473	576	0.55	0.62	0.58
2	RandomForestClassifier	1.000000	0.893951	7468	526	433	616	0.59	0.54	0.56
6	GradientBoostingClassifier	0.923985	0.880018	7258	736	349	700	0.67	0.49	0.56
5	AdaBoostClassifier	0.904285	0.867080	7168	826	376	673	0.64	0.45	0.53
0	LogisticRegression	0.853765	0.816101	6635	1359	304	745	0.71	0.35	0.47
1	DecisionTreeClassifier	1.000000	0.861661	7240	754	497	552	0.53	0.42	0.47
3	KNeighborsClassifier	0.938158	0.816764	6665	1329	328	721	0.69	0.35	0.47
4	GaussianNB	0.802196	0.715913	5699	2295	274	775	0.74	0.25	0.38

- After adding all the category maximum f1 score achieved is LGBM (0.60), XGB (0.58), Gradient Boosting (0.56)
- Based on feature engineering we created multiple models, in that feature engineering with month binned has good scores.
- From that LGBM, XGB, GB performed good, so were tunning the models to get better scores

6. HYPERPARAMETER TUNING:

For month binned feature engineered data

- Building a Light Gradient Boosting model

```
0.9477417940365823
0.9046776512219397
[[7514  480]
 [ 382  667]]
              precision    recall  f1-score   support

           0       0.95      0.94      0.95      7994
           1       0.58      0.64      0.61      1049

    accuracy                           0.90      9043
   macro avg       0.77      0.79      0.78      9043
weighted avg       0.91      0.90      0.91      9043
```

- After doing the hyper parameter tuning the FN is increased 0.61 and the TP is increased
- But the model predictions have been improved
- Building a Gradient Boosting model

```
0.9433688949789566
0.8958310295255999
[[7424  570]
 [ 372  677]]
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      7994
           1       0.54      0.65      0.59      1049

    accuracy                           0.90      9043
   macro avg       0.75      0.79      0.77      9043
weighted avg       0.90      0.90      0.90      9043
```

- After doing the hyper parameter tuning the TP is increased the f1 score 0.59 also increased for Gradient Boosting

- Building a Xtreme Gradient Boosting model

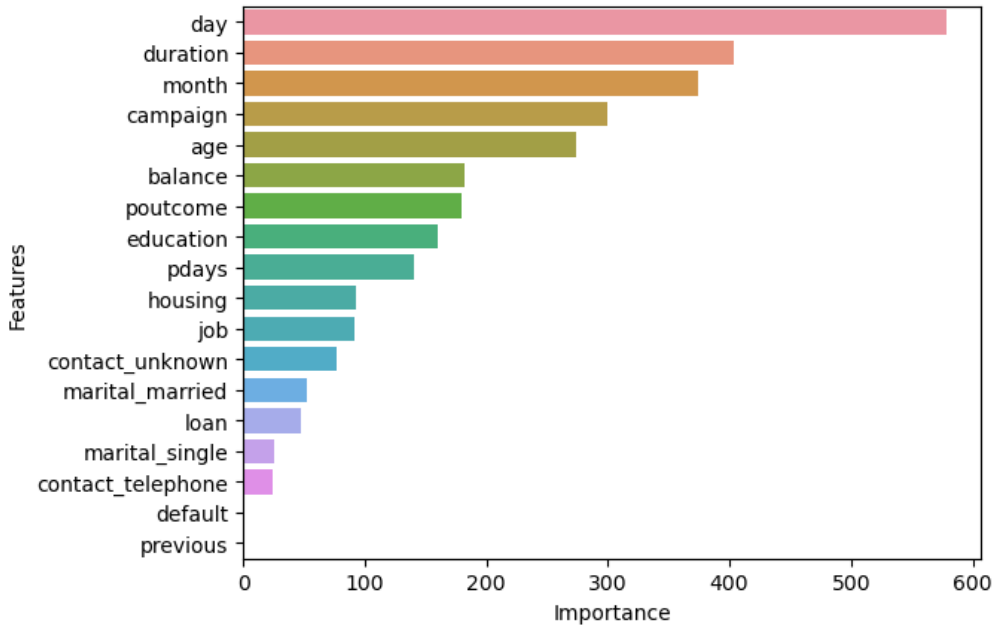
```
0.9727904472937261
0.906668141103616
[[7613  381]
 [ 463  586]]
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      7994
           1       0.61      0.56      0.58      1049

    accuracy                           0.91      9043
   macro avg       0.77      0.76      0.76      9043
weighted avg       0.90      0.91      0.91      9043
```

- After doing the hyper parameter tuning the TP is increased the f1 score (0.58) also increased for xgb

7. FEATURE EXTRACTION:



- Removing the default and previous from data and checking for improvement

```
0.9475851916812829
0.9042353201371226
[[7506  488]
 [ 378  671]]
              precision    recall  f1-score   support

           0       0.95      0.94      0.95      7994
           1       0.58      0.64      0.61      1049

    accuracy                           0.90      9043
   macro avg       0.77      0.79      0.78      9043
weighted avg       0.91      0.90      0.91      9043
```

- After removing the feature, the f1 score remains the same and the TP has been increased
- So, we proceed by extracting the default and previous feature and build models for good predictions

8. SUGGESTIONS FOR PREDICTION OF SUBSCRIBING TO TERM DEPOSITS:

Based on EDA observations the following suggestions have been made:

- Job, Credit default, Housing and Loan significantly affect the customer experience along with the several other variables considered.

- The bank should highly focus on their marketing strategy on promoting their new scheme of term deposits instead of using the old technique of marketing through phone calls as adapting different new strategies may attract customers effectively and can be time and cost reducing.

- Considering the customer's financial liability and stability is important for the business as the customer with huge liabilities and financial risks such as housing loans, personal loans, credit card defaults are prone to risks which may play a major in subscribing to the term deposits.

- The financial institutions must ensure that their customers have potential qualifications before enrolling them into any kind of policies or schemes introduced such as work background as many of their customers might be students or retired personnel too and they might not have the potential and necessity at present in enrolling/subscribing to such schemes.
- Duration of calls impact our customers, higher the call duration better the term deposits subscriptions

9. REFERENCES:

- Dataset - https://archive.ics.uci.edu/ml/datasets/bank+marketing
- Towards Data science – https://towardsdatascience.com/decision-tree-in- machine- learninge380942a4c96

  https://towardsdatascience.com/random-forests-algorithm-explained-with-areal-life-example-and-some-python-code-affbfa5a942c

  https://www.analyticsvidhya.com/blog/2021/06/understanding- randomforest/

10. Notes For Project Team

Sample Reference for Datasets (to be filled by team and mentor)

Original owner of data	Paulo Cortez, Sergio Moro
Data set information	Bank Marketing
Any past relevant articles using the dataset	Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier,62:22-31, June2014
Reference	Kaggle
Link to web page	https://archive.ics.uci.edu/ml/datasets/bank+marketing