# Capstone Project

## Bank Marketing Effectiveness Prediction

**Team Members**

Saurabh Hase

Aakash Sharma

# Problem Statement

**Will the contacted client subscribe to a term deposit?**

# Data Description

**Input variables:**

**Bank Client data:**

- **age (numeric)**
- **job:** type of job (categorical: admin, blue-collar, entrepreneur, housemaid, management, retired, self-employed, services, student, technician, unemployed, unknown)
- **marital :** marital status (categorical: divorced, married, single, unknown ; note: divorced means divorced or widowed)
- **Education:** (categorical: basic-4y, basic-6y, basic-9y, high school, illiterate, professional course, university degree, unknown)
- **default:** has credit in default? (categorical: no, yes, unknown)
- **housing:** has housing loan? (categorical: no, yes, unknown)
- **loan:** has personal loan? (categorical: no, yes, unknown)

- **contact:** contact communication type (categorical: cellular, telephone)
- **month:** last contact month of year (categorical: jan, feb, mar, ..., nov, dec)
- **day_of_week:** last contact day of the week (categorical: mon, tue, wed, thu, fri)
- **duration:** last contact duration, in seconds (numeric).
- **campaign:** number of contacts performed during this campaign and for this client (numeric, includes last contact)
- **pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- **previous:** number of contacts performed before this campaign and for this client (numeric)
- **poutcome:** outcome of the previous marketing campaign (categorical: failure, nonexistent, success)
- **y/deposit** - has the client subscribed a term deposit? (binary: yes, no)

# Exploratory Data Analysis

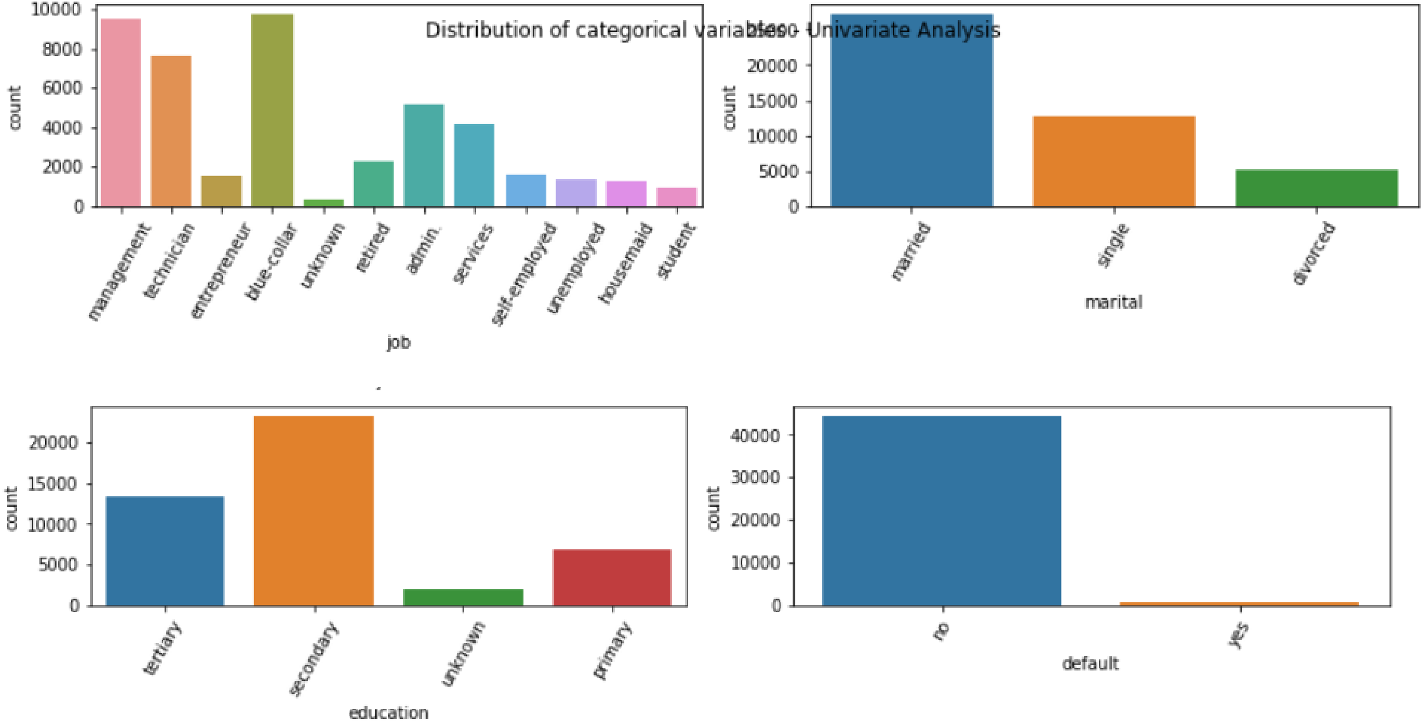
Exploratory Data Analysis, EDA for short, is simply a ‘first look at the data’. It forms a critical part of the machine learning workflow and it is at this stage we start to understand the data we are working with and what it contains

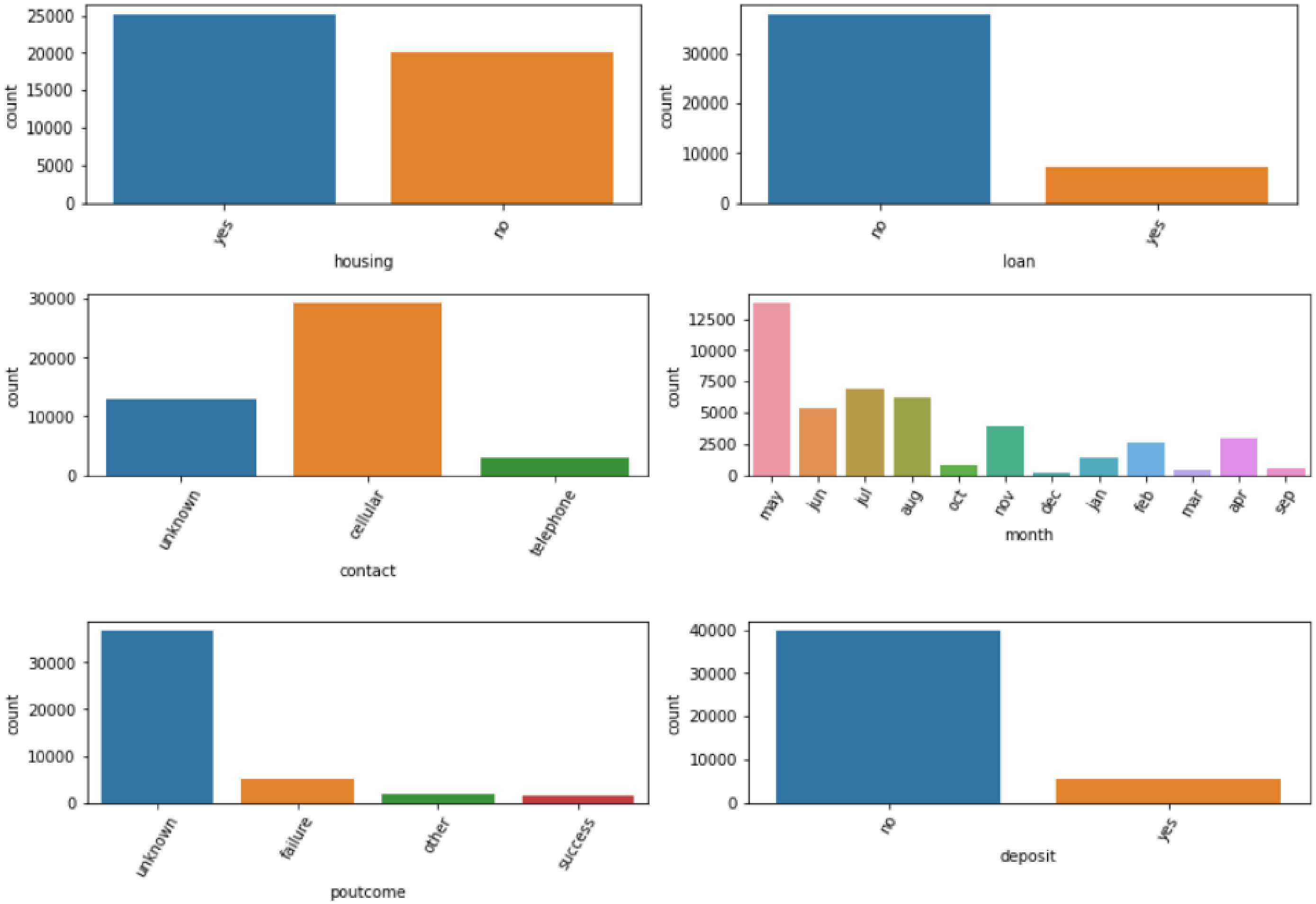
## Performed steps

### 1. Importing CSV file and analysing data

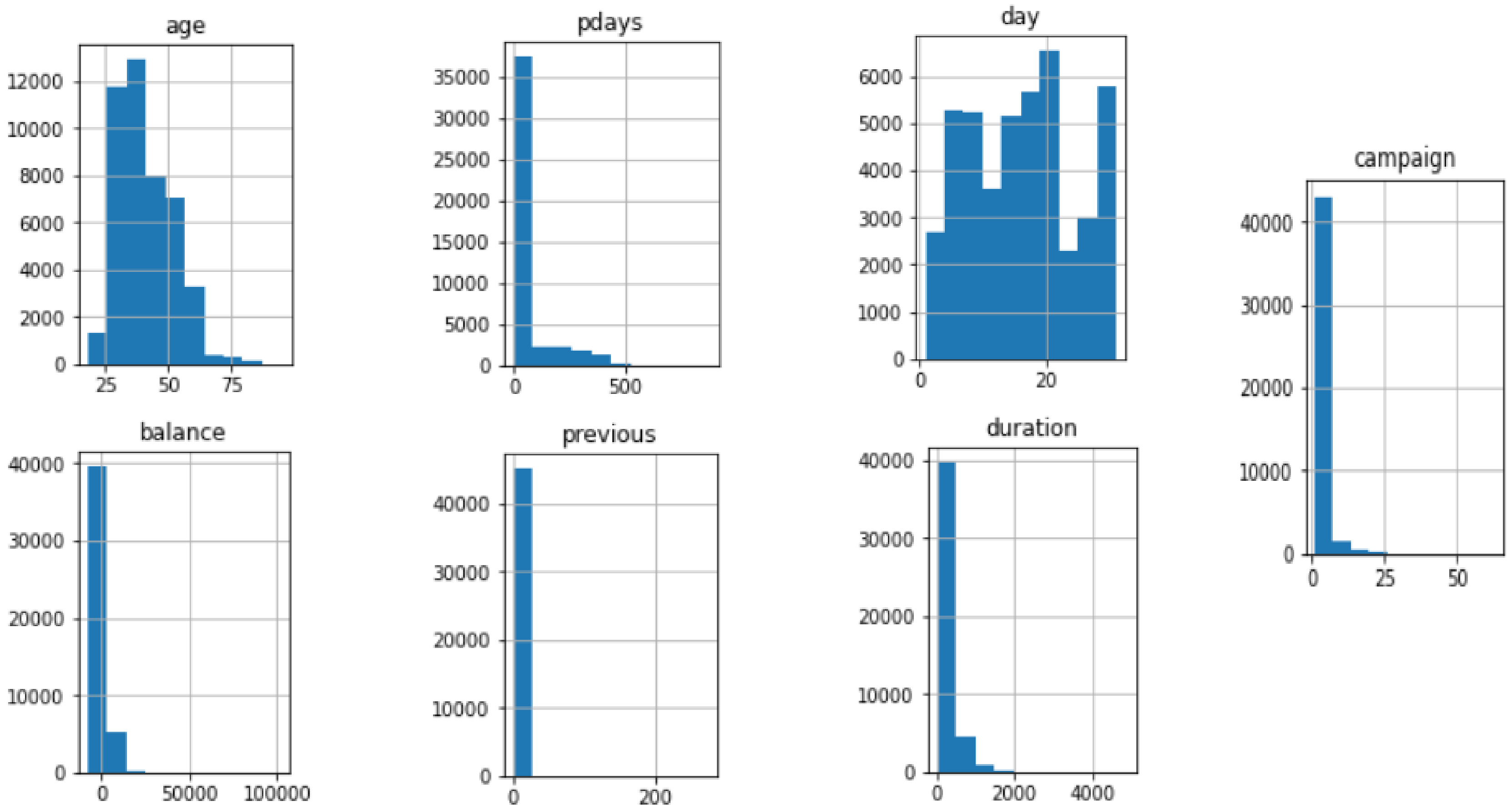
Using the pandas library we imported the CSV file to read the dataset. We used the delimiter function inside the read_csv to separate out the columns of the dataset.

# 2. Categorical variable distribution



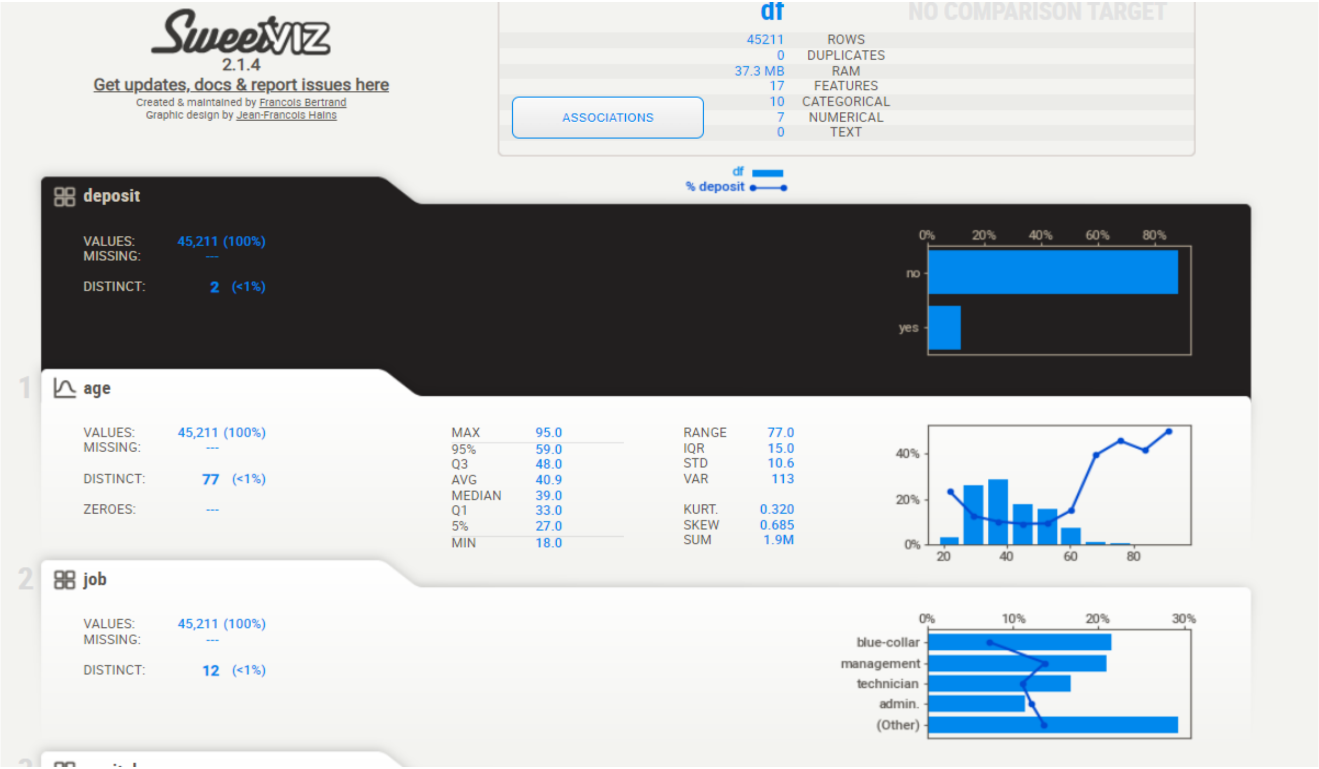

count
25000
20000
15000
10000
5000
0
yes
no
housing
count
30000
20000
10000
0
no
yes
loan
count
30000
20000
10000
0
unknown
cellular
telephone
contact
count
12500
10000
7500
5000
2500
0
may
jun
jul
aug
oct
nov
dec
jan
feb
mar
apr
sep
month
count
30000
20000
10000
0
unknown
failure
other
success
poutcome
count
40000
30000
20000
10000
0
no
yes
deposit

# 3. Numerical Variables Distribution



# 4. Sweetviz

Sweetviz is an open-source Python library that generates beautiful, high-density visualizations to kickstart EDA



Sweetviz
2.1.4
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains

```
df                NO COMPARISON TARGET
45211     ROWS
0         DUPLICATES
37.3 MB   RAM
17        FEATURES
10        CATEGORICAL
7         NUMERICAL
0         TEXT
```

ASSOCIATIONS

deposit

```
VALUES:     45,211 (100%)
MISSING:    ---
DISTINCT:   2  (<1%)
```

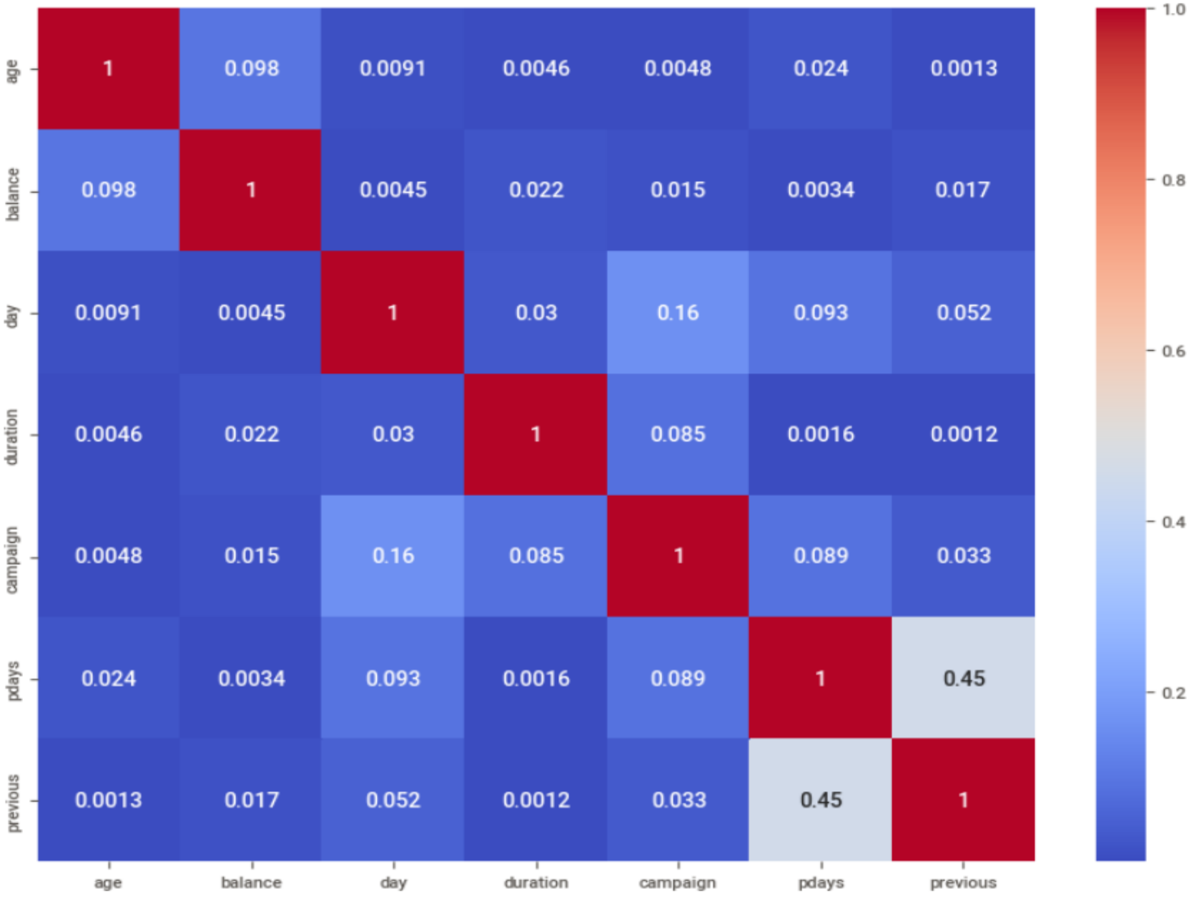
1 age

```
VALUES:     45,211 (100%)     MAX      95.0     RANGE   77.0
MISSING:    ---               95%      59.0     IQR     15.0
                              Q3       48.0     STD     10.6
DISTINCT:   77  (<1%)         AVG      40.9     VAR     113
                              MEDIAN   39.0
ZEROES:     ---               Q1       33.0     KURT.   0.320
                              5%       27.0     SKEW    0.685
                              MIN      18.0     SUM     1.9M
```

2 job

```
VALUES:     45,211 (100%)
MISSING:    ---
DISTINCT:   12  (<1%)
```

# Correlation



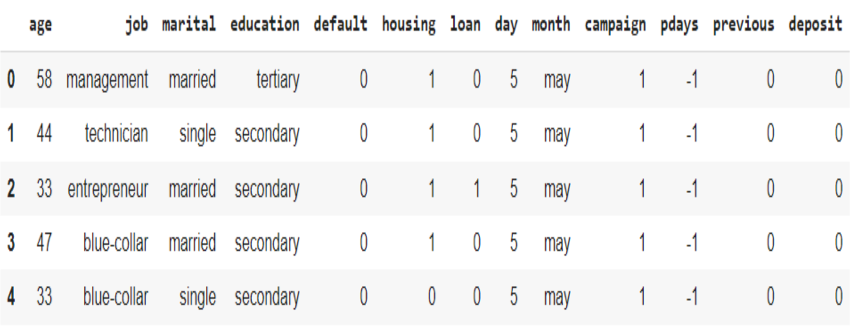
|  | age | balance | day | duration | campaign | pdays | previous |
|---|---|---|---|---|---|---|---|
| age | 1 | 0.098 | 0.0091 | 0.0046 | 0.0048 | 0.024 | 0.0013 |
| balance | 0.098 | 1 | 0.0045 | 0.022 | 0.015 | 0.0034 | 0.017 |
| day | 0.0091 | 0.0045 | 1 | 0.03 | 0.16 | 0.093 | 0.052 |
| duration | 0.0046 | 0.022 | 0.03 | 1 | 0.085 | 0.0016 | 0.0012 |
| campaign | 0.0048 | 0.015 | 0.16 | 0.085 | 1 | 0.089 | 0.033 |
| pdays | 0.024 | 0.0034 | 0.093 | 0.0016 | 0.089 | 1 | 0.45 |
| previous | 0.0013 | 0.017 | 0.052 | 0.0012 | 0.033 | 0.45 | 1 |

# One Hot encoding

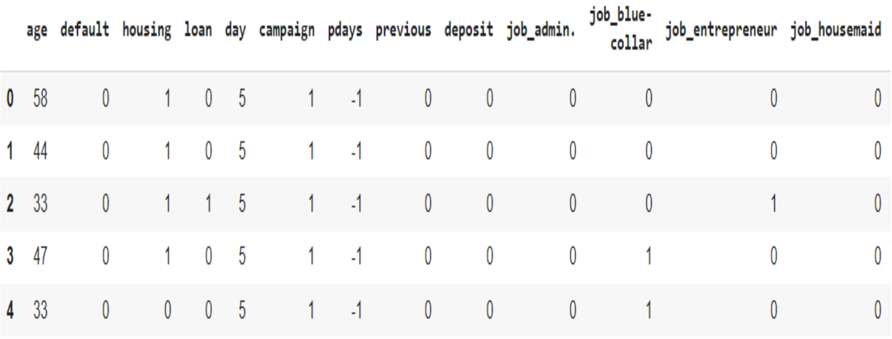
One Hot Encoding is method to produce binary integers of 0 and 1 to encode categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

# Label Encoding

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.



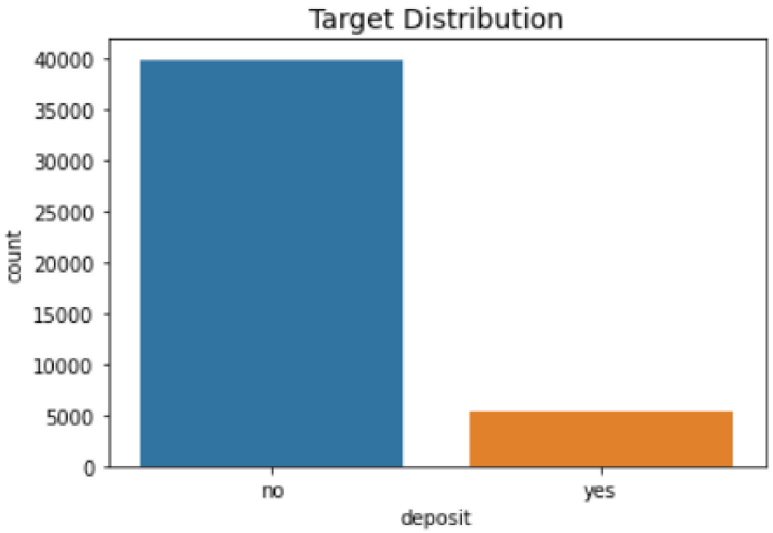
| | age | job | marital | education | default | housing | loan | day | month | campaign | pdays | previous | deposit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 58 | management | married | tertiary | 0 | 1 | 0 | 5 | may | 1 | -1 | 0 | 0 |
| 1 | 44 | technician | single | secondary | 0 | 1 | 0 | 5 | may | 1 | -1 | 0 | 0 |
| 2 | 33 | entrepreneur | married | secondary | 0 | 1 | 1 | 5 | may | 1 | -1 | 0 | 0 |
| 3 | 47 | blue-collar | married | secondary | 0 | 1 | 0 | 5 | may | 1 | -1 | 0 | 0 |
| 4 | 33 | blue-collar | single | secondary | 0 | 0 | 0 | 5 | may | 1 | -1 | 0 | 0 |



| | age | default | housing | loan | day | campaign | pdays | previous | deposit | job_admin. | job_blue-collar | job_entrepreneur | job_housemaid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 58 | 0 | 1 | 0 | 5 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 44 | 0 | 1 | 0 | 5 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 33 | 0 | 1 | 1 | 5 | 1 | -1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 47 | 0 | 1 | 0 | 5 | 1 | -1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4 | 33 | 0 | 0 | 0 | 5 | 1 | -1 | 0 | 0 | 0 | 1 | 0 | 0 |

# Imbalance Data set

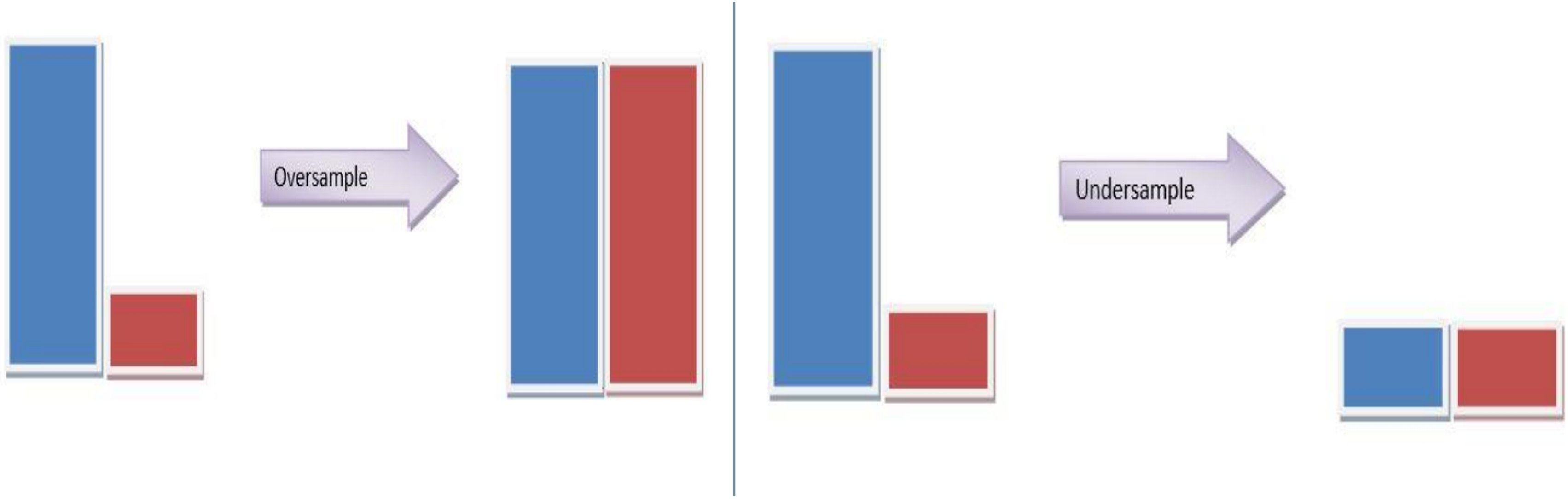
Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations



# Resampling

### Oversampling & Undersampling

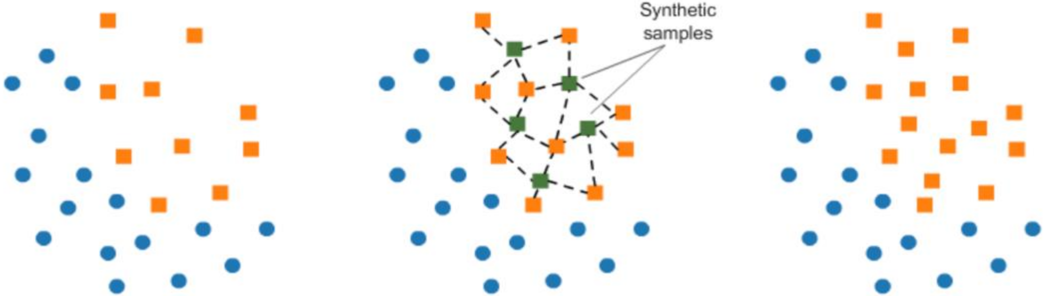
This technique is used to upsample or downsample the minority or majority class. When we are using an imbalanced dataset, we can oversample the minority class using replacement.

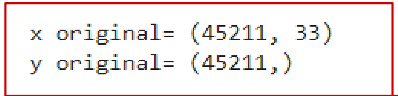


# SMOTE

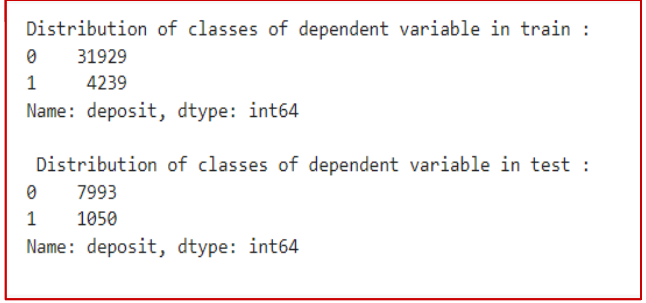
SMOTE stands for **Synthetic Minority Oversampling Technique**.

SMOTE is an algorithm that performs data augmentation by creating **synthetic data points** based on the original data points.



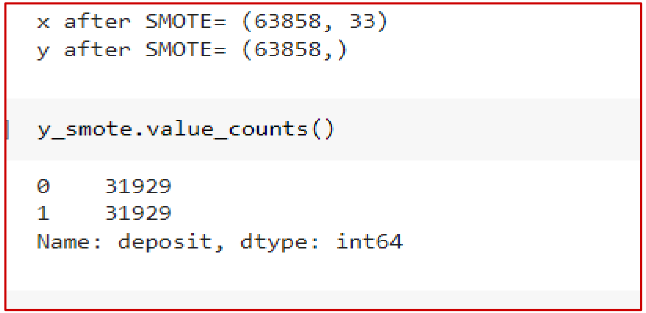


```
x original= (45211, 33)
y original= (45211,)
```



```
Distribution of classes of dependent variable in train :
0    31929
1     4239
Name: deposit, dtype: int64

 Distribution of classes of dependent variable in test :
0     7993
1     1050
Name: deposit, dtype: int64
```



```
x after SMOTE= (63858, 33)
y after SMOTE= (63858,)
```

```
y_smote.value_counts()
```

```
0    31929
1    31929
Name: deposit, dtype: int64
```

# Scaling The Data

Differences in the scales across input variables may increase the difficulty of the problem being modelled so we used scaling method to bring the values in the same scale

## Standard scaler

Standardizing a dataset involves rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1.

**y = (x – mean) / standard_deviation**

**mean = sum(x) / count(x)**

**standard_deviation = sqrt( sum( (x – mean)^2 ) / count(x)**

# Machine learning models

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- eXtreme Gradient Boosting
- CAT Boosting
- K-Nearest Neighbour

# Evaluation Matrices

## Confusion Matrix

Confusion matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values

| | | Actual Values | |
|---|---|---|---|
| | | Positive (1) | Negative (0) |
| Predicted Values | Positive (1) | TP | FP |
| | Negative (0) | FN | TN |

$$\textit{Recall} = \frac{TP}{TP + FN}$$

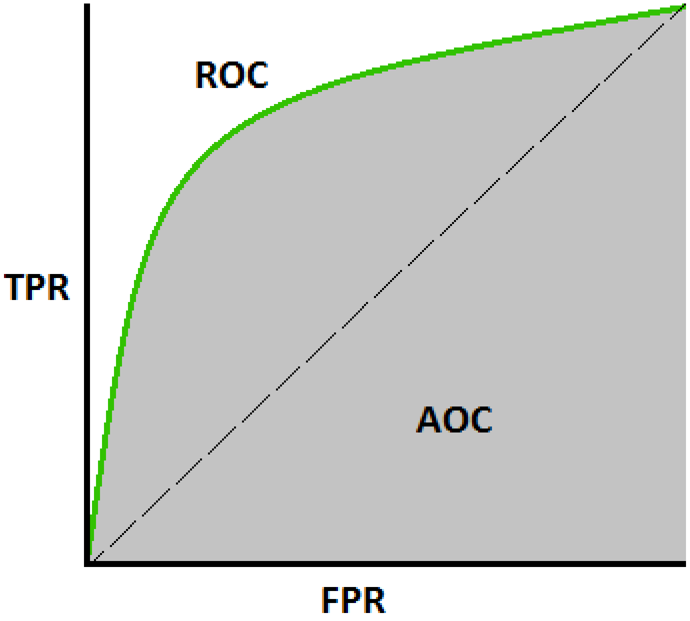
$$\textit{Precision} = \frac{TP}{TP + FP}$$

$$\textit{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

$$\textbf{Accuracy} = \frac{(TP+TN)}{Total}$$

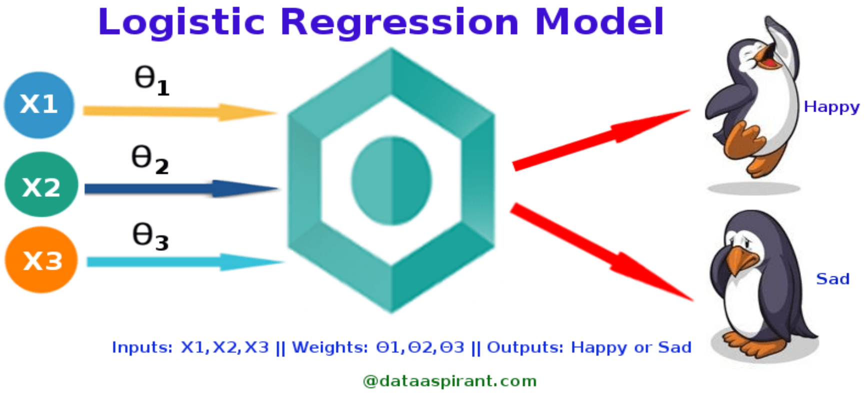
## AUC - ROC Curve

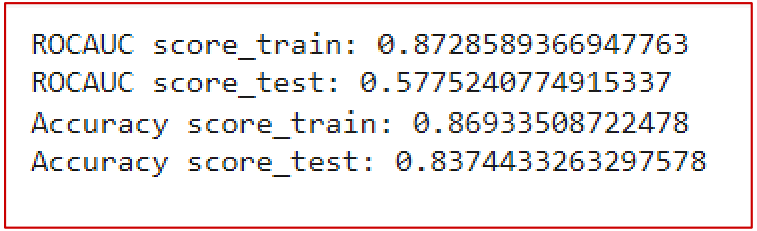
AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.



# Logistic Regression

- Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable.
- The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.
- In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).


Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: Θ1,Θ2,Θ3 || Outputs: Happy or Sad
@dataaspirant.com

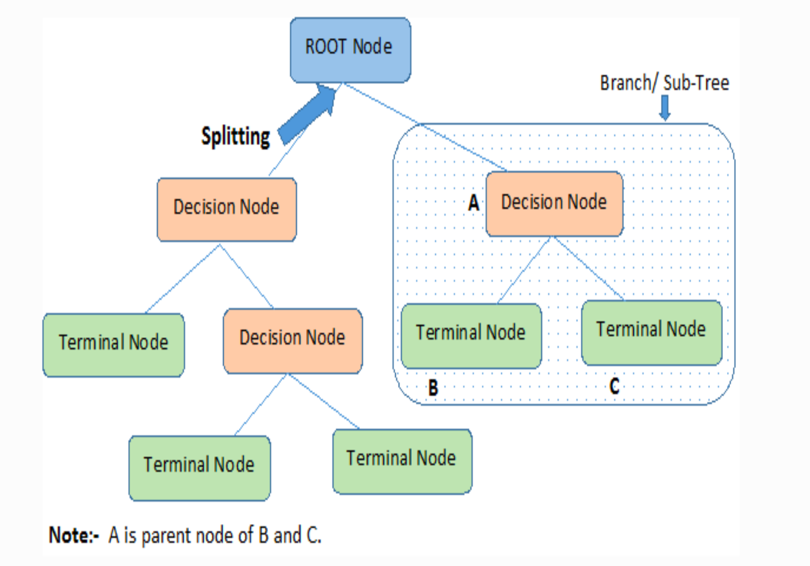


```
ROCAUC score_train: 0.8728589366947763
ROCAUC score_test: 0.5775240774915337
Accuracy score_train: 0.86933508722478
Accuracy score_test: 0.8374433263297578
```

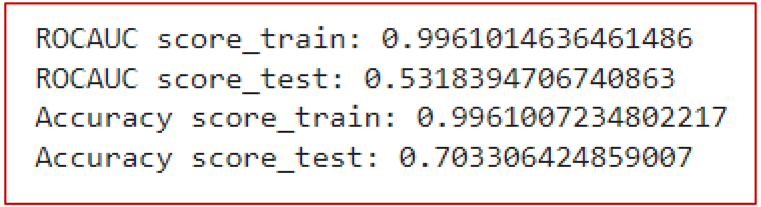
# Decision Tree Regression

Decision tree builds regression or classification models in the form of a **tree structure**. It breaks down a dataset into smaller and smaller subsets to reach the result by taking maximum votes

- **Step-1:** Begin the tree with the root node
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the root node into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.



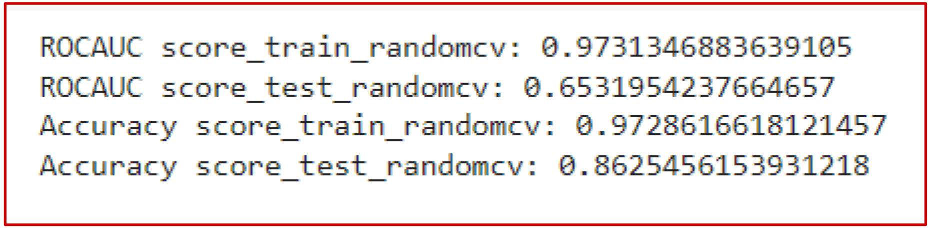
**Note:-** A is parent node of B and C.



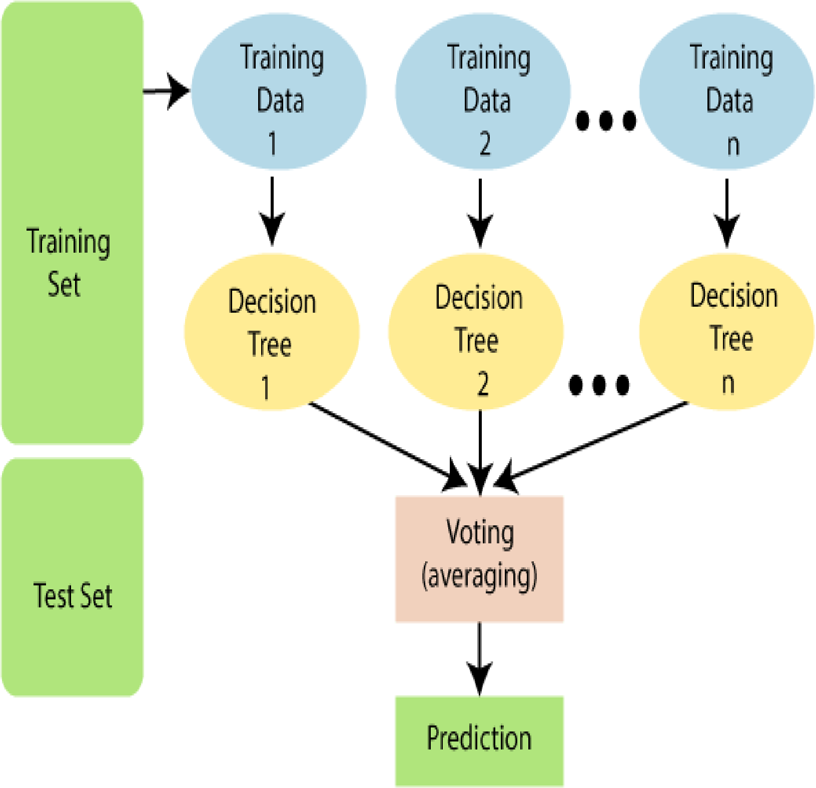
```
ROCAUC score_train: 0.996101463646148б
ROCAUC score_test: 0.531839470674086З
Accuracy score_train: 0.9961007234802217
Accuracy score_test: 0.703306424859007
```

# Random Forest Regression

- Random forest, as its name suggests, comprises an enormous amount **of individual decision trees** that work as a group or as they say, an ensemble.
- Every individual decision tree in the random forest lets out a class prediction and the class with the **most votes** is considered as the model's prediction.

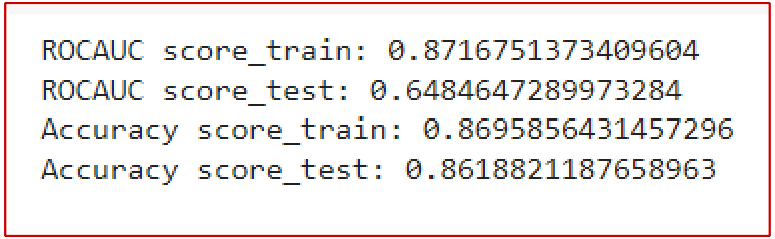


```
ROCAUC score_train_randomcv: 0.9731346883639105
ROCAUC score_test_randomcv: 0.6531954237664657
Accuracy score_train_randomcv: 0.9728616618121457
Accuracy score_test_randomcv: 0.8625456153931218
```

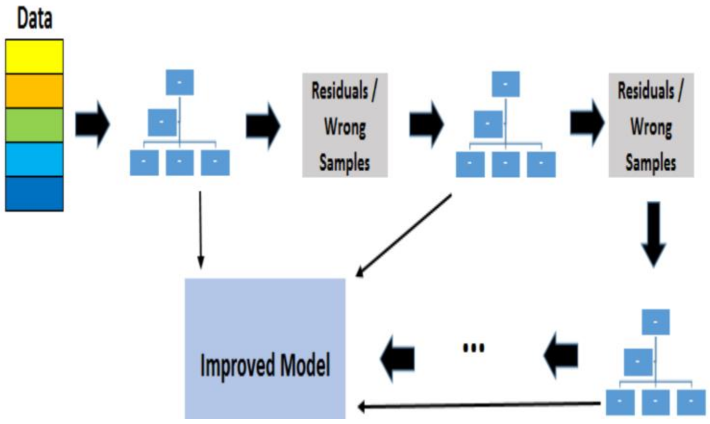


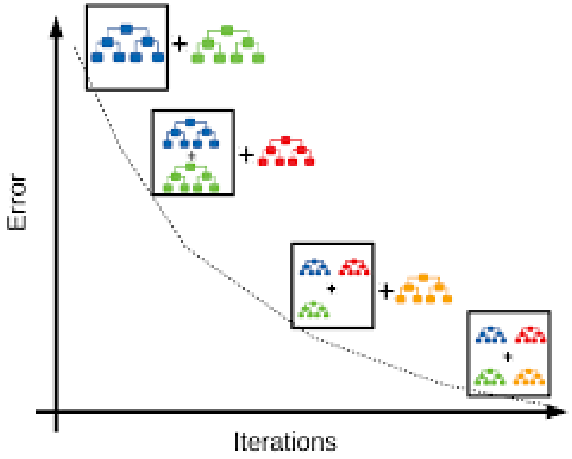
# Gradient Boost

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its **predecessor's error**. In gradient boosting decision trees, we combine many weak learners to come up with one strong learner. The weak learners here are the individual decision trees. All the trees are conncted in series and each tree tries to minimise the error of the previous tree.



```
ROCAUC score_train: 0.8716751373409604
ROCAUC score_test: 0.6484647289973284
Accuracy score_train: 0.8695856431457296
Accuracy score_test: 0.8618821187658963
```

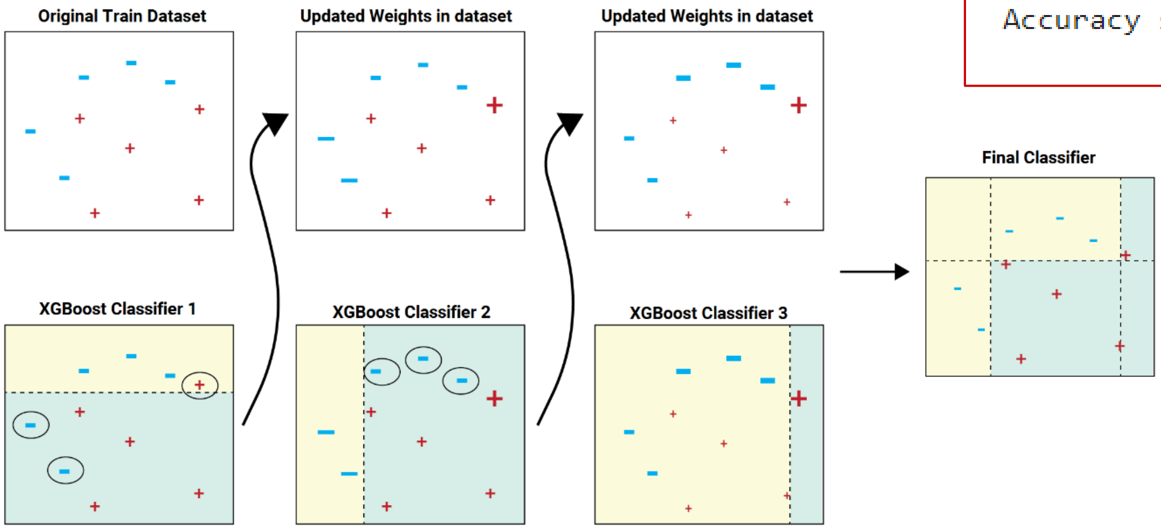




# eXtreme Gradient Boost

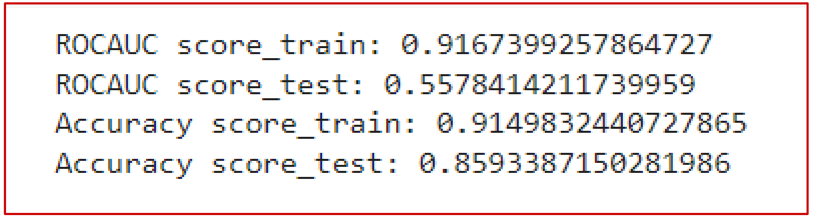
In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results

```
ROCAUC score_train: 0.8738415243719928
ROCAUC score_test: 0.6341879187304705
Accuracy score_train: 0.8719502646496915
Accuracy score_test: 0.8555789008072542
```



# CAT Boosting

- CatBoost name comes from two words “**Cat**egory” and “**Boost**ing
- As discussed, the library works well with multiple **Cat**egories of data, such as audio, text, image including historical data.
- Handling Categorical features automatically

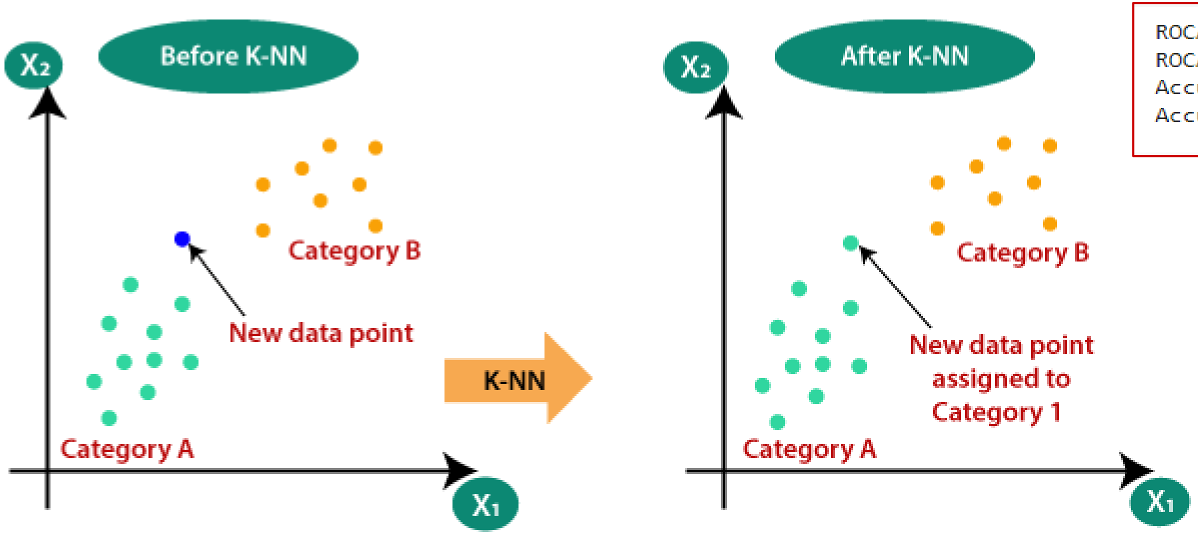


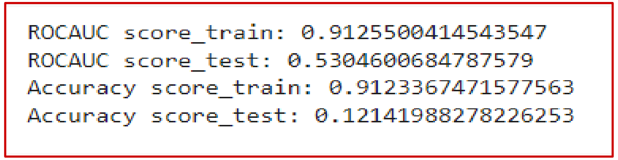
```
ROCAUC score_train: 0.9167399257864727
ROCAUC score_test: 0.5578414211739959
Accuracy score_train: 0.9149832440727865
Accuracy score_test: 0.8593387150281986
```

# K Nearest Neighbour

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories

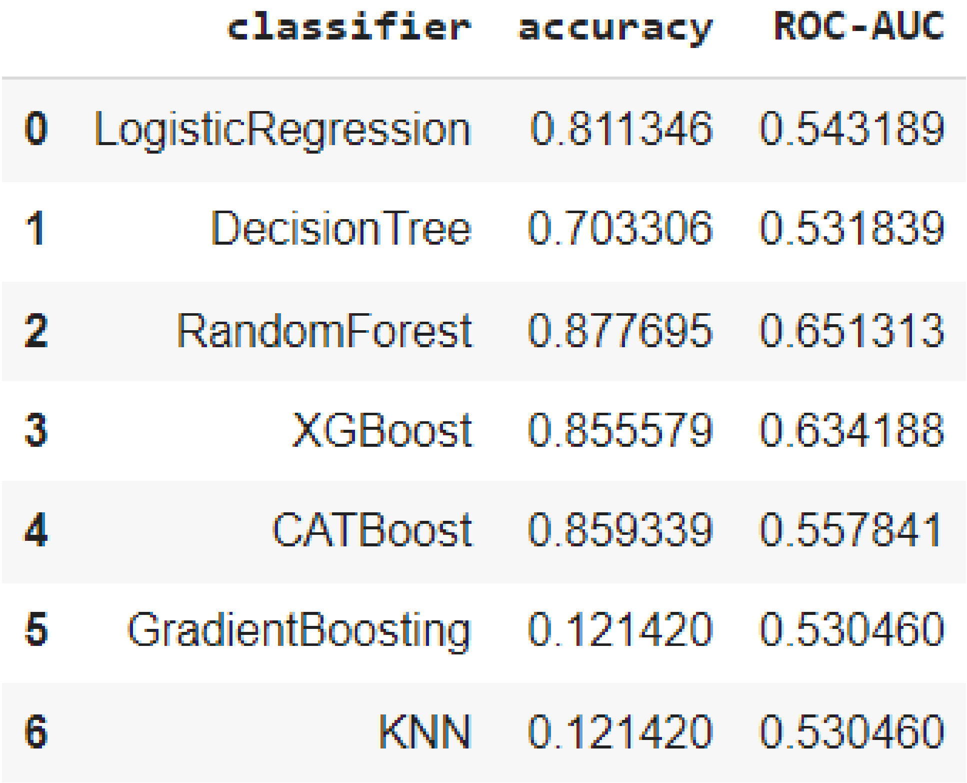
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.





```
ROCAUC score_train: 0.9125500414543547
ROCAUC score_test: 0.5304600684787579
Accuracy score_train: 0.9123367471577563
Accuracy score_test: 0.12141988278226253
```

# RESULTS



| | classifier | accuracy | ROC-AUC |
|---|---|---|---|
| **0** | LogisticRegression | 0.811346 | 0.543189 |
| **1** | DecisionTree | 0.703306 | 0.531839 |
| **2** | RandomForest | 0.877695 | 0.651313 |
| **3** | XGBoost | 0.855579 | 0.634188 |
| **4** | CATBoost | 0.859339 | 0.557841 |
| **5** | GradientBoosting | 0.121420 | 0.530460 |
| **6** | KNN | 0.121420 | 0.530460 |

# Conclusion

With this we have completed our experiments with the help of following steps.

- EDA (Exploratory Data Analysis)
- Checking correlation
- One hot encoding/ Label encoding
- SMOTE
- StandardScalar
- Applying various models
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - Gradient Boosting
  - eXtreme Gradient Boosting
  - CAT Boosting
  - KNN.
- Evaluation matrices
  - Confusion Matrix
  - AUC-ROC Curve

**We found that Random Forest model gives the us best result of ROC_AUC score of 65%**

THANK YOU.