BANK MARKETING ANALYSIS

-Preksha Dhoot
Sneh Bhandari
Samyak Nayak

Abstract

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The **classification** goal is to predict if the client will subscribe to a term deposit.This project will enable the bank to develop a more granular understanding of its customer base, predict customers' response to its telemarketing campaign and establish a target customer profile for future marketing plans.
By analyzing customer features, the bank will be able to predict customer saving behaviours and **identify which type of customers is more likely to make term deposits**. The bank can then focus its marketing efforts on those customers. This will not only allow the bank to secure deposits more effectively but also increase customer satisfaction by reducing undesirable advertisements for certain customers.

Dataset Description

This dataset is about the direct phone call marketing campaigns, which aim to promote term deposits among existing customers, by a Portuguese banking institution from May 2008 to November 2010. It is publicly available in the UCI Machine Learning Repository retrieved from http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#.

This dataset has features about the clients to whom the calls were made.The class information is **binary yes or no for a term deposit with the bank**.With this data availability,the aim is to predict if the client would agree for a term deposit with the bank or not.
The dataset has **45211 values for 17 different features** out of which **7** are numerical features and **10** are categorical features. The feature information is given as follows

No.	Feature Name	Feature Type	Feature Description
1.	Age	Numeric ; Bank Client Data	Age of the client
2.	Job	Categorical ; Bank Client Data	Type of job the client has
3.	Marital	Categorical ; Bank Client Data	Marital status of the client

4.	Education	Categorical ; Bank Client Data	Education level of the client
5.	Default	Categorical ; Bank Client Data	If the client has credit in default
6.	Balance	Numeric ; Bank Client Data	Balance of the client
7.	Housing	Categorical ; Bank Client Data	If the client has housing loan
8.	Loan	Categorical ; Bank Client Data	If the client has personal loan
9.	Contact	Categorical ; Related to last contact	Contact communication type
10.	Day	Numeric; Related to last contact	Last contact day of the week
11.	Month	Categorical ; Related to last contact	Last contact month of the year
12.	Duration	Numeric ; Related to last contact	Last contact duration
13.	Campaign	Numeric ; Other Attributes	Number of contacts performed
14.	Pdays	Numeric ; Other Attributes	Number of days passed after last contact
15.	Previous	Numeric ; Other Attributes	Number of contacts performed previously
16.	Poutcome	Categorical ; Other Attributes	Outcome of previous campaign
17.	y	Categorical ; Output Variable	Subscription of term deposit

Data Preprocessing

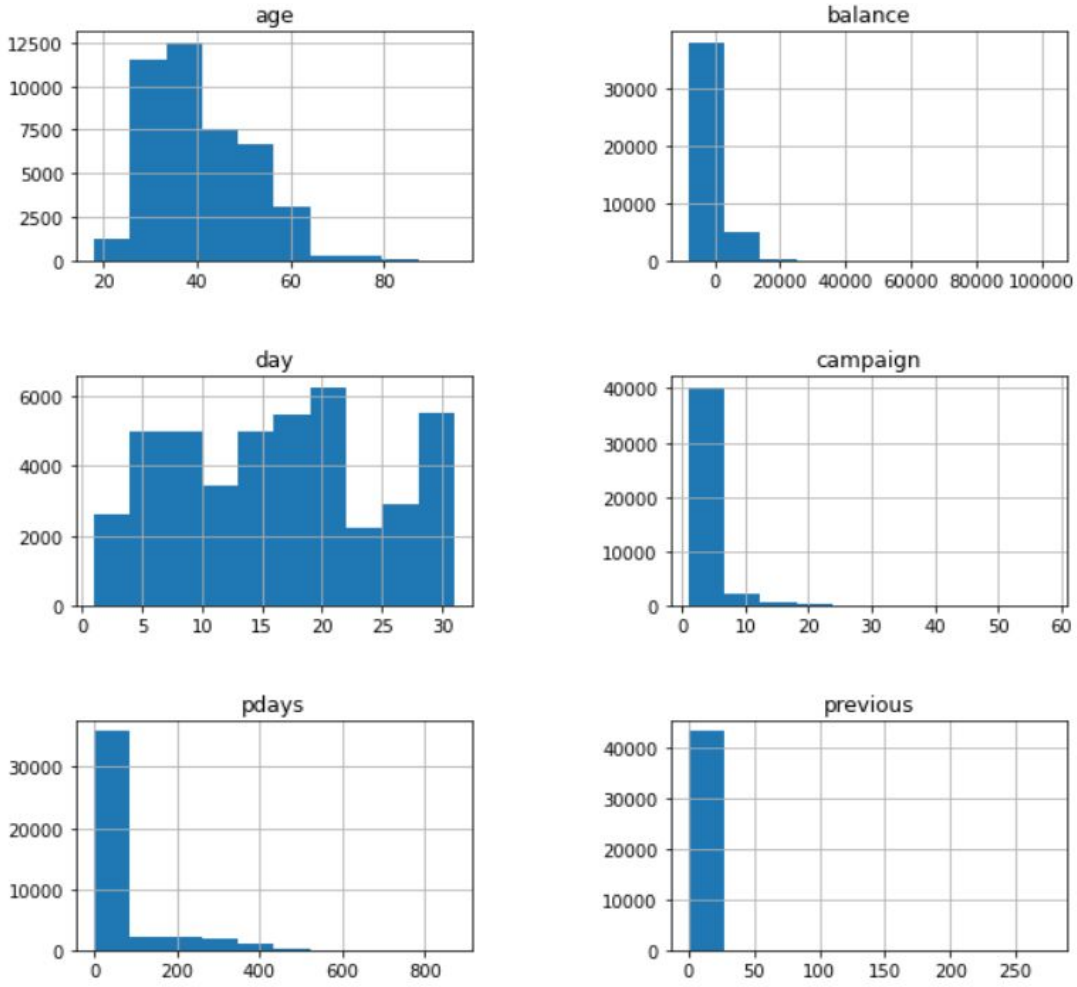
- REMOVING NULL VALUES

  There were a lot of unknown values in the dataset,particularly in the **'contact'** and the '**poutcome**' features,so we neglected the two and also removed the remaining rows with any null values in them.

On checking the autocorrelation between the variables, we found that the feature **‘duration**’ is **correlated** with the output class ’y’,and it highly affects the output, so we removed it to make a realistic predictive model .

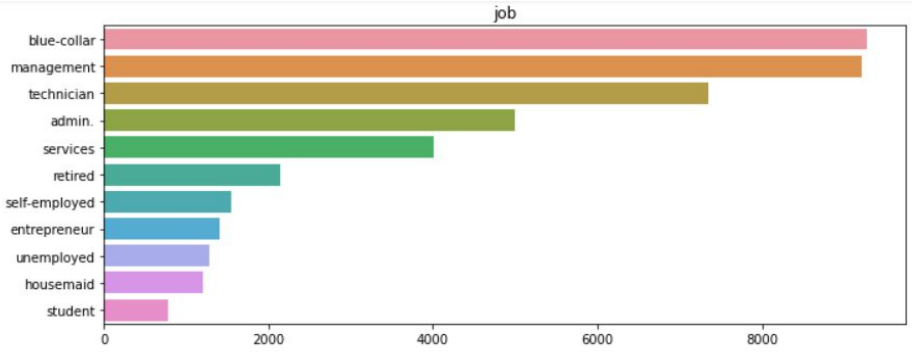
- NUMERICAL VARIABLES

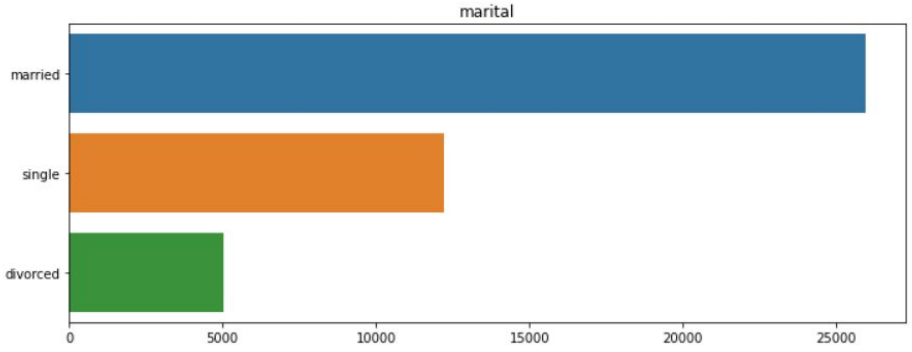
  Visualizing numerical variables

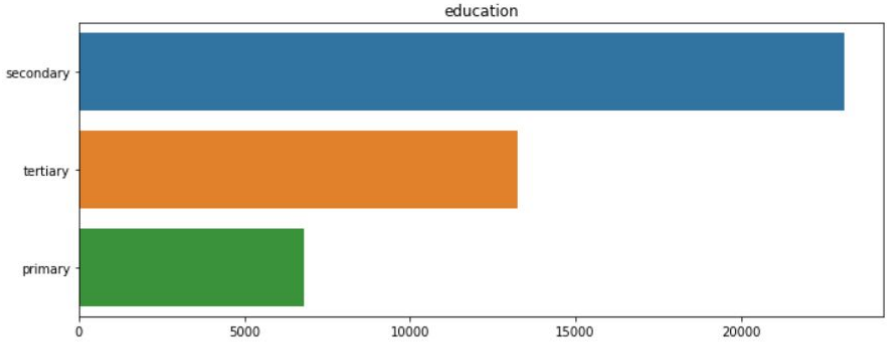


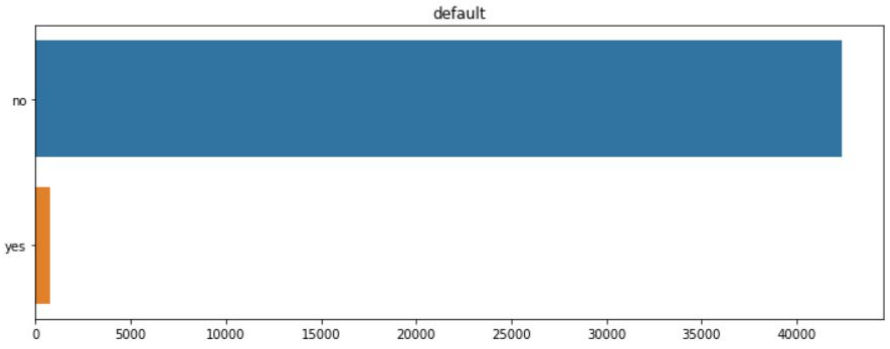
  Though all the features are continuous**,the range of values** among the features **vary** a lot which is why we have **normalized** our dataset while working on the models.

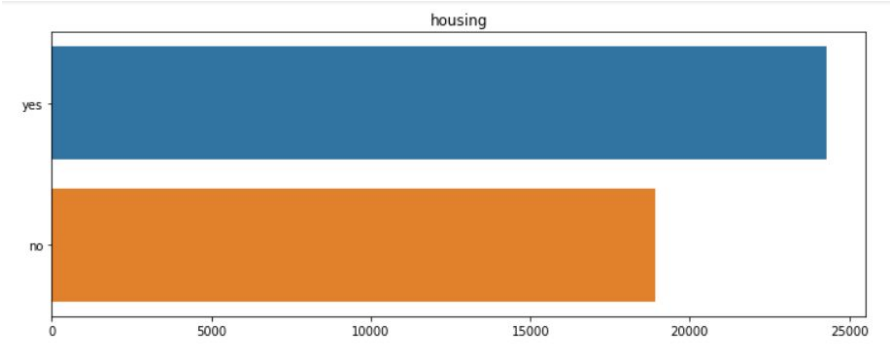
- CATEGORICAL VARIABLES

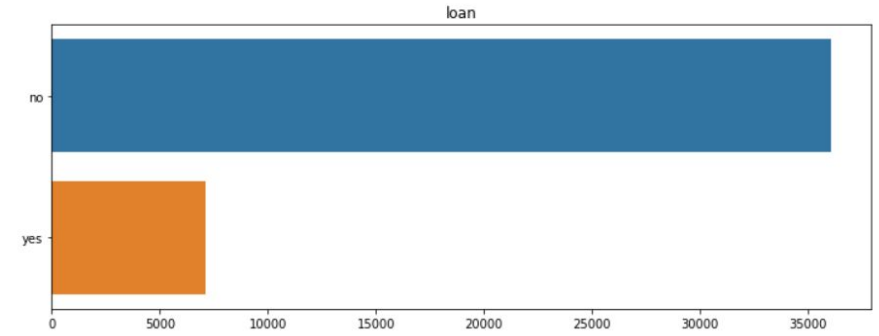

job
blue-collar
management
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
0
2000
4000
6000
8000

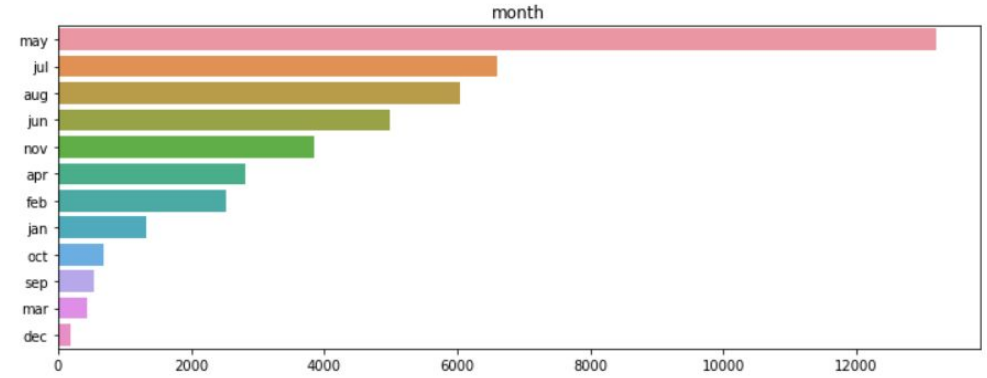

marital
married
single
divorced
0
5000
10000
15000
20000
25000


education
secondary
tertiary
primary
0
5000
10000
15000
20000


default
no
yes
0
5000
10000
15000
20000
25000
30000
35000
40000

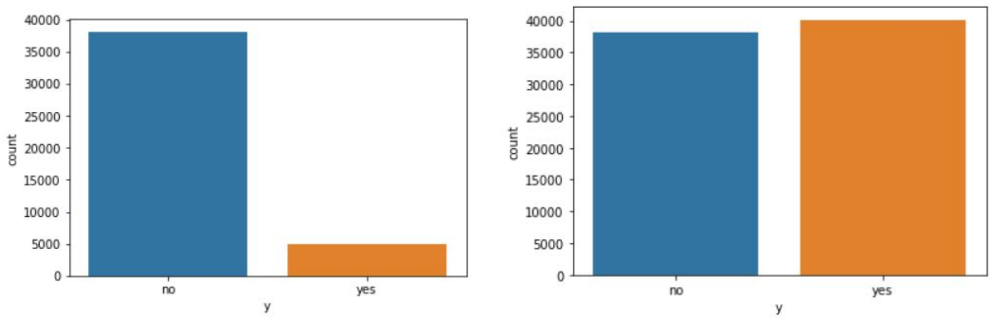






We converted the categorical variables into meaningful numerical values by: **label encoding** the **ordinal features**(‘default’,’housing’,’loan’) and **one-hot encoding** the **nominal features**(‘job’,’marital’,’education’,’month’)

- IMBALANCED CLASSIFICATION



Visualisation of output variable(original dataset, after up sampling)

For the output class, the values are **highly imbalanced** with the number of **'no'** values being **38172** and **'yes'** only being **5021**. So,to get better analysis we have **upsampled** and **downsampled** the data separately and checked the performance of various models on all three datasets- the original dataset(with **45211 values**),the downsampled dataset(with **10042 values**) and the upsampled dataset(**78319 values**)

MODELS

I. PNN

We consider d independent variables $X_1,X_2....X_d$ and $Y_1,Y_2...Y_k$ as the response variable with k classes.We are required to find the value of j for which the probability of Y belonging to a particular class j given X=x is maximum

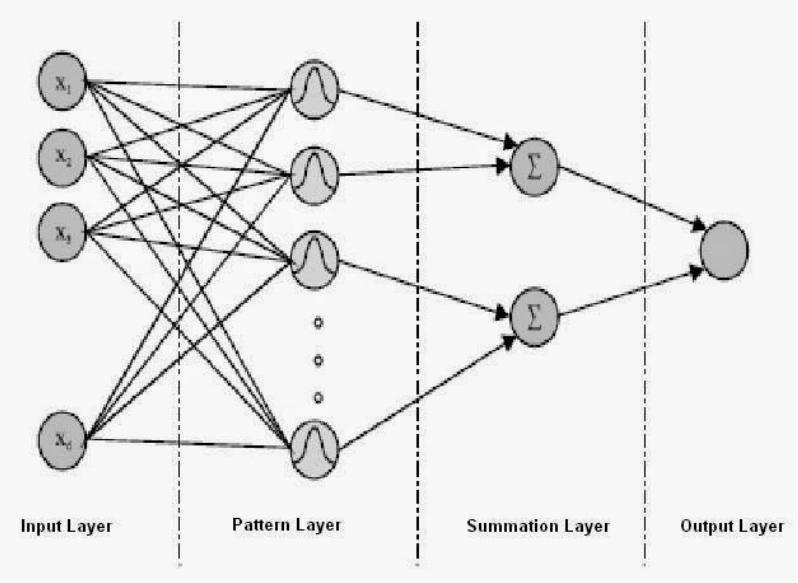
$$p(j):=P(\ Y=j \mid X=x\ )$$

The most probable label will be predicted for observation X using the equation:

$$argmax_{i<=j<=k}p(j)$$

In a PNN, the operations are organized into a multilayered feedforward network with four layers:

- Input layer
- Pattern layer
- Summation layer
- Output layer

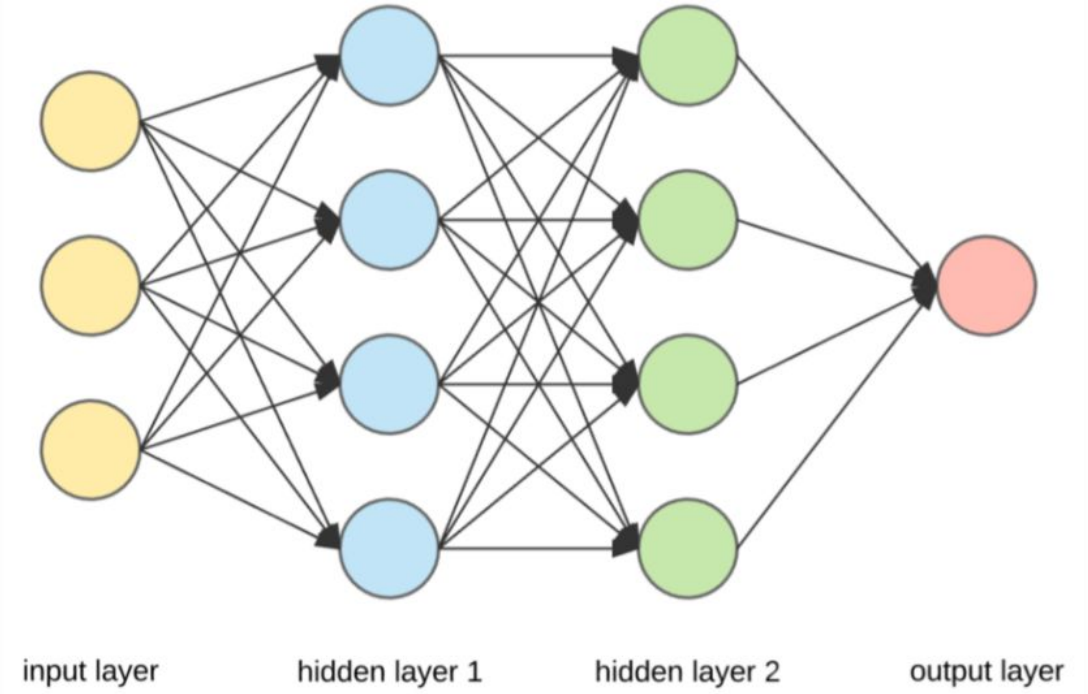


Each neuron in the **input layer** represents a predictor variable.The **pattern layer** is designed to contain one neuron (node) for each training case available and the neurons are split into the classes.The PNN executes a training case by first presenting it to all pattern layer neurons. Each neuron in the pattern layer computes a

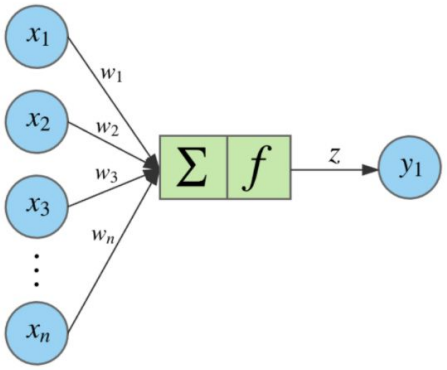
distance measure between the presented input vector and the training example represented by that pattern neuron. The PNN then subjects this distance measure to the Parzen window (weighting function, W) and yields the activation of each neuron in the pattern layer.The **summation layer** contains one neuron for each class and computes p(j).The results from the summation neurons are then compared and the largest is fed forward to the **output neuron** to yield the computed class and the probability that this observation will belong to that class.

II. ANN

An Artificial Neural Network is a deep learning model. It contains the input layer, with the number of neurons as the features, single(called a perceptron) or multiple hidden layers, and an output layer.



We get a deep neural net if we significantly increase the hidden layer.
Every neuron in a layer is connected to every neuron in the next layer and these connections have weight.
Every neuron while taking the inputs sums all the weights.



If the inputs are X=(X1, X2, X3 …. Xn) and the respective weights of the connections are W=(W1, W2, W3 …. Wn) to a particular neuron, y1, then, the value of the neuron y1 will be

$$y1=f\,(b + w.x)= f\,(b + \sum_{i=1}^{n} xiwi\,)$$

Where b is the bias term and is added to all the layers. And f is the activation function defined by us. The most commonly used are RELU, softmax and sigmoid.

The number of hidden layers, the neurons in them and the activation function is all decided by the user.

While training the model, the *ANN* uses the method called back-propagation to adjust the weights based on the output provided.

Result

Since our dataset was imbalanced, we have used two different sampling methods to fit our models:

We have divided the dataset into 80:20 for training and testing.

We have used the following metrics to check the accuracy of our models and find the best models

- Accuracy — measure of true positive and true negative results
- Sensitivity — measure of part of true results our model classify to be right
- Specificity — measure of part of false results our model classify to be right
- AUC-ROC — the closer the value is to 1, the better the model
-

Original dataset

	Specificity	Sensitivity	AUC-ROC	Accuracy
PNN	0.1512	0.8308	0.4910	0.8823
ANN	0.097	0.9896	0.5434	0.6719
Random Forest	0.1955	0.9797	0.5876	0.8854
Support Vector	0.097	0.989	0.50	0.8823

Upsampling

	Specificity	Sensitivity	AUC-ROC	Accuracy
PNN	0.1549	0.8243	0.4898	0.4839
ANN	0.7497	0.6759	0.7128	0.7134
Random Forest	1.0	0.9454	0.9727	0.9731
Support Vector	0.82	0.2589	0.5420	0.5469

Downsampling

	Specificity	Sensitivity	AUC-ROC	Accuracy
PNN	0.1627	0.8321	0.4974	0.4922
ANN	0.6372	0.7077	0.6599	0.6719
Random Forest	0.6382	0.7644	0.7013	0.7003
Support Vector	0.0284	0.9777	0.5030	0.4957

Conclusion

The imbalance of the classes in the output variables gives us an **imbalance** dataset(more no than yes). This gives our models a much higher accuracy to predicting 'no' output compared to 'yes' output, leading to a **huge difference between the specificity and sensitivity** values, as we can see in the table for ***Original Dataset***.

This problem is solved using *upsampling* and *downsampling.*

We can see from these respective tables that **ANN** and **Random Forest** in the **down sampling** table have a **higher AUC-ROC** scores (**70.13%** and **65.99%**) as well as **improved specificity and sensitivity**. We see that **Random Forest** performs exceptionally well in the **up sampled dataset**, with an AUC-ROC score of **97.27%**.

References

1.https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

2.https://github.com/Bharat-Reddy/Bank-Marketing-Analysis

3. https://github.com/mcabinaya/Bank-Marketing-Data-Analysis

4.https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

Github project link- https://github.com/prekshadhoot/Bank-Marketing