Project Report-

**Bank Campaign Prediction**

Problem Statement :-

We are going to predict the results of a Marketing Campaign conducted by a Portuguese Bank based on phone calls using different combination of different Machine Learning algorithms and imbalanced data handling techniques to solve the problem.

Our Dataset has 41188 observations and 21 attributes ordered by date (from May 2008 to November 2010)

We are trying to build the models without the use of the variable Duration(which contributes greatly to general models) as the duration of a call is not known before making the call. So using this model the bank should be able to decide whether or not to call the customer and minimize its losses.

Attribute Information

* **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’
* **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 (numeric, include)
* **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 (numeric)
* **poutcome**: outcome of the previous marketing campaign (categorical: 'failure’, 'nonexistent’, 'success’)
* **emp.var.rate**: employment variation rate - quarterly indicator (numeric)
* **cons.price.idx**: consumer price index - monthly indicator (numeric)
* **cons.conf.idx**: consumer confidence index - monthly indicator (numeric)
* **euribor3m**: euribor3 month rate - daily indicator (numeric)
* **nr.employed**: number of employees - quarterly indicator (numeric)
* **y** - has the client subscribed a term deposit? (binary: 'yes’, 'no')

Unknown values imputation

Attributes Job, education, marital and housing were having unknown values which were imputed using the mode of each attribute.

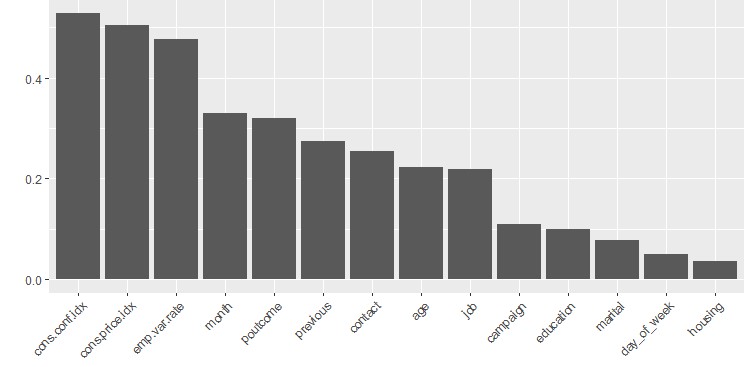
Correlation analysis

Based on correlation analysis attributes nr.employed and eurobor3 were dropped as they were highly correlated(>0.90) with the attribute emp.var.rate .

Attributes **Loan** and **Default** were dropped because one of the class in each attributed was distributed over more than 80% of the attribute, hence not much of the variance was explained by these variables.

Attribute **Duration** is dropped because from business point of view the duration of a call is not known before making the call, and usually the result is know after the call.

Attribute **pdays** is dropped when we check for attributes importance using chi-square. Below is the variable importance graph



Class Imbalance Problem:

Most real-world classification problems display some level of class imbalance, which is when each class does not make up an equal portion of your data-set.

In this project we have focused on Sampling based methods to deal with the problem at hand and used the following methods along with different ML algorithms.

Under Sampling

Over Sampling

Synthetic Minority Oversampling TEchnique(SMOTE)

.

The following ML Algorithms were applied in our Project :-

* Random Forest
* Decision Trees
* Logistic Regression
* Naïve Bayes
* SVM
* Xgboost

Every one of these algorithms has been used with the combination of sampling techniques to see which of the combination works better on our dataset despite divided into Imbalanced Classes.

What is Under Sampling?

This method works with majority class. It reduces the number of observations from majority class to make the data set balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.

Library used : ROSE

Function used :Ovun.sample , Parameter : Method = UNDER

**Disadvantage** of Under Sampling : Potential Loss of Information..

What is Over Sampling?

This method works with minority class. It replicates the observations from minority class to balance the data. It is also known as Upsampling.

**Disadvantage** of Up Sampling :After Upsampling, the models are more prone to Overfit.

Library used : ROSE

Function used : Ovun.sample , Parameter : Method = over

What is SMOTE?

In simple words, instead of replicating and adding the observations from the minority class, it overcome imbalances by generates artificial data. It is also a type of oversampling technique.

In regards to synthetic data generation, synthetic minority oversampling technique (SMOTE) is a powerful and widely used method. SMOTE algorithm creates artificial data based on feature space (rather than data space) similarities from minority samples. We can also say, it generates a random set of minority class observations to shift the classifier learning bias towards minority class.To generate artificial data, it uses bootstrapping and k-nearest neighbours.

**Advantage**: As the data gets generated is not exactly the replica of the minority class, the models are less prone to overfit.

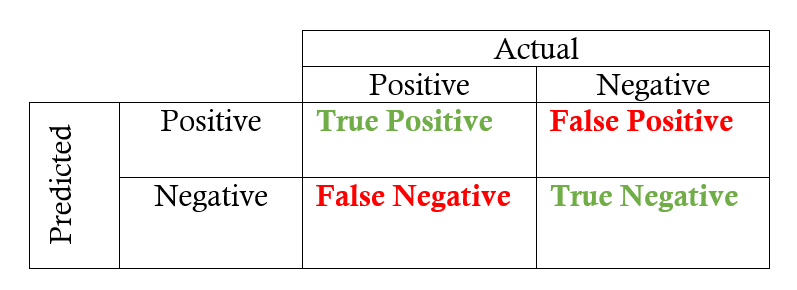
**Disadvantage**: The generated data is not Legitimate Data but just a resemblance of the Original Data.

Library used :DMwR

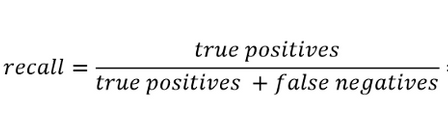
Function used : SMOTE , Parameter : perc.under = 200 , perc.over = 100

Evaluation Matrix

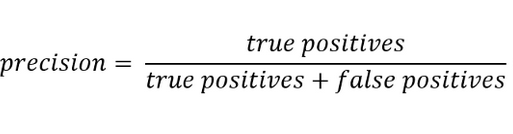
*CONFUSION MATRIX :-*



**Recall** : The precise definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are data point classified as positive by the model that actually are positive (meaning they are correct), and false negatives are data points the model identifies as negative that actually are positive (incorrect).



**Precision** : Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative



**F-Measure**: The F1 score isthe harmonic mean of precision and recall taking both metrics into account in the following equation:



**AUC**: An AUC curve plots the true positive rate on the y-axis versus the false positive rate on the x-axis. The true positive rate (TPR) is the recall and the false positive rate (FPR) is the probability of a false alarm.

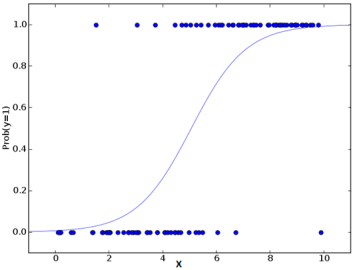
**MODELS**

*LOGISTIC REGRESSION :-*

The logistic function is a Sigmoid function, which takes any real value between zero and one. It is defined as

https://cdn-images-1.medium.com/max/800/1*xtIMx01jy4myAo1N8bbSCA.png

And if we plot it, the graph will be S curve,



Function used : generalised linear model ,Parameter : family = binomial

model=glm(data=train,y~.,family=binomial())

Base model : TP: 131, FP: 49, FN:774, TN: 7063

Precision: 0.728, Recall: 0.145, F-measure: 0.121, AUC: 0.751

Undersampledmodel : TP: 529, FP: 871, FN:376, TN: 6241

Precision: 0.378, Recall: 0.582, F-measure: 0.23, AUC: 0.782

Oversampled model : TP: 529, FP: 871, FN:376, TN: 6241

Precision: 0.378, Recall: 0.582, F-measure: 0.23, AUC: 0.782

SMOTE model : TP: 526, FP: 847, FN:379, TN: 6265

Precision: 0.383, Recall: 0.581, F-measure: 0.231, 0.761

DECISION TREES:-

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

Function used :- rpart

Library Used:-rpart

Base model : TP: 154, FP: 83, FN:751, TN: 7029

Precision: 0.65, Recall: 0.17, F-measure: 0.135, AUC: 0.582

Undersampledmodel : TP: 547, FP: 1022, FN:358, TN: 6090

Precision: 0.349, Recall: 0.604, F-measure: 0.221, AUC: 0.756

Oversampled model : TP: 533, FP: 996, FN: 372, TN: 6116

Precision: 0.349, Recall: 0.589, F-measure: 0.219, AUC: 0.752

SMOTE model : TP: 511, FP: 897, FN: 394, TN: 6215

Precision: 0.363, Recall: 0.565, F-measure: 0.221, 0.751

RANDOM FOREST:-

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it’s simplicity and the fact that it can be used for both classification and regression tasks. In this post, you are going to learn, how the random forest algorithm works and several other important things about it.

Base model : TP: 209, FP: 135, FN:626, TN: 6430

Precision: 0.606, Recall: 0.253, F-measure: 0.178, AUC: 0.779

Undersampledmodel : TP: 528, FP: 1044, FN:307, TN: 5521

Precision: 0.336, Recall: 0.632, F-measure: 0.219, AUC: 0.781

Oversampled model : TP: 472, FP: 826, FN: 363, TN: 5739

Precision: 0.364, Recall: 0.565, F-measure: 0.221, AUC: 0.771

SMOTE model : TP: 472, FP: 672, FN: 363, TN: 5893

Precision: 0.413, Recall: 0.565, F-measure: 0.239,AUC: 0.786

randomForest(y~.,data=f\_data\_training,nTrees=500,mtry=3)

Library Used :randomforest

Function Used :randomforest , Parameter: ntree=500,mtry=3

Naïve Bayes:-

**Naive Bayes** is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.

Base model : TP: 209, FP: 750, FN:461, TN: 6362

Precision: 0371, Recall: 0.49, F-measure: 0.423, AUC: 0.652

Undersampledmodel : TP: 558, FP: 1592, FN:347, TN: 5520

Precision: 0.259, Recall: 0.616, F-measure: 0.3653, AUC: 0.696

Oversampled model : TP: 528, FP: 1561, FN: 350, TN: 5551

Precision: 0.262, Recall: 0.613, F-measure: 0.367, AUC: 0.697

SMOTE model : TP: 517, FP: 1337, FN: 388, TN: 5775

Precision: 0.278, Recall: 0.5712, F-measure: 0.37441,AUC: 0.692

naiveBayes(x = train[,-15],y = train[,15])

Library Used : Caret, e1071

Function used :NaiveBayes

Support Vector Machines:-

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

(radial kernel)

Base model : Precision: 0.8125, Recall: 0.147, F-measure: 0.249, AUC: 0.572

Undersampled model: Precision: 0.341, Recall: 0.670, F-measure: 0.452, AUC: 0.756

Oversampled model: Precision: 0.3617, Recall: 0.647, F-measure: 0.459, AUC: 0.748

SMOTE model :Precision: 0.356, Recall: 0.647, F-measure: 0.459, AUC: 0.752

(Polynomial kernel)

Base model : Precision: 1, Recall: 0.056, F-measure: 0.107, AUC: 0.528

Undersampled model: Precision: 0.472, Recall: 0.193, F-measure: 0.274, AUC: 0.583

Oversampled model: Precision: 0.367, Recall: 0.534, F-measure: 0.435, AUC: 0.715

SMOTE model :Precision: 0.425, Recall: 0.42, F-measure: 0.42, AUC: 0.65

svm(y~., data = f\_data1\_training)

Library Used : e1071

Function used :tune , svm ( Parameters : kernel, cost,gamma )

XgBoost:-

It is a boosting technique which uses Gradient Descent for Optimization. Hence named Extreme Gradient Boosting.

Undersampled model : TP: 545, FP: 1013, 360, TN: 6099

Precision: 0.35, Recall: 0.602, F-measure: 0.221, AUC: 0.782

Oversampled model : TP: 521, FP: 1015, FN: 339, TN: 6142

Precision: 0.34, Recall: 0.607, F-measure: 0.222, AUC: 0.7828

SMOTE model : TP: 474, FP: 627, FN: 431, TN: 6485

Precision: 0.431, Recall: 0.524, F-measure: 0.236, AUC: 0.78

Library Used :xgboost

Function used :xgboost

parameters: list(booster = "gbtree", objective = "binary:logistic", eta=0.01, gamma=1, max\_depth=4, min\_child\_weight=1, subsample=0.7, colsample\_bytree=0.6)

Optimisation: xgb.cv( params = params, data = dtrain, nrounds = 200, nfold = 6, showsd = T, stratified = T, print.every\_n = 10, early\_stopping\_rounds= 50, maximize = T, eval\_metric = "auc")

RESULTS AND INTERPRETATION

Accuracy is not considered in our project as an evaluation metrics due to presence of Imbalanced classes and will lead to misleading results.

Evaluating the models based on Precision, Recall, F-measure and AUC.

Based on our metrics the XGBoost model with oversampling should be considered as a better model because it has the highest AUC of 0.7828 and a recall of 0.607.

And, the random forest model with Under sampling which has the AUC of 0.78, and the recall of 0.632.

Recall is the measure, we should try to maximize as it has the FN term in the denominator, which if is a huge value will cause the bank more loss, as it shows the positive class as negative (i.e. The customers who could have bought the product our model will classify them as they will not buy).

We tested the above two models on the test set for our final model selection.

Random Forest – Precision 0.332, Recall 0.645 and AUC 0.799

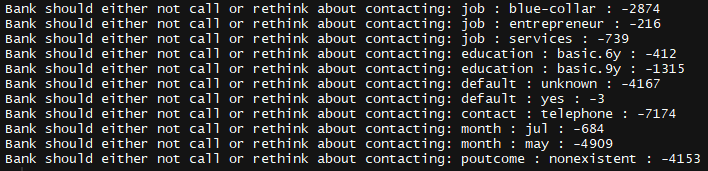
XGBoost – precision 0.35, Recall 0.63 and AUC 0.791.

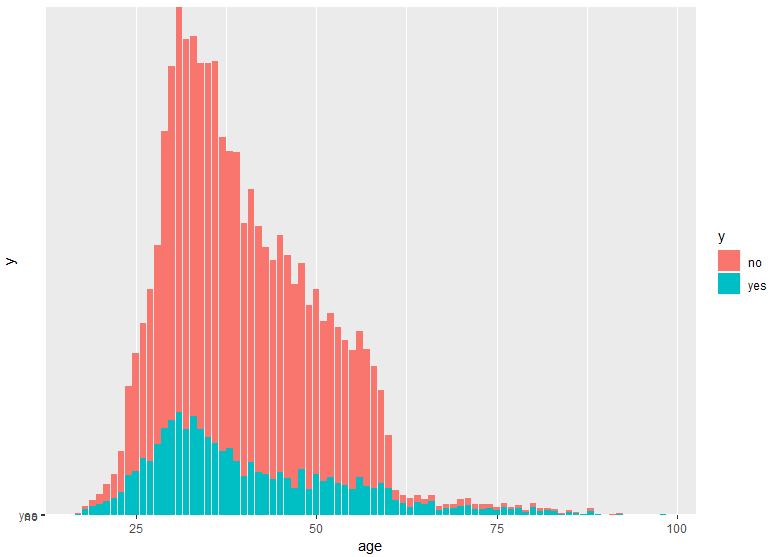
Turns out Random Forest with undersampling has the best results.

Along with opting for the best model, the bank should also discretize their customers as per certain categories.

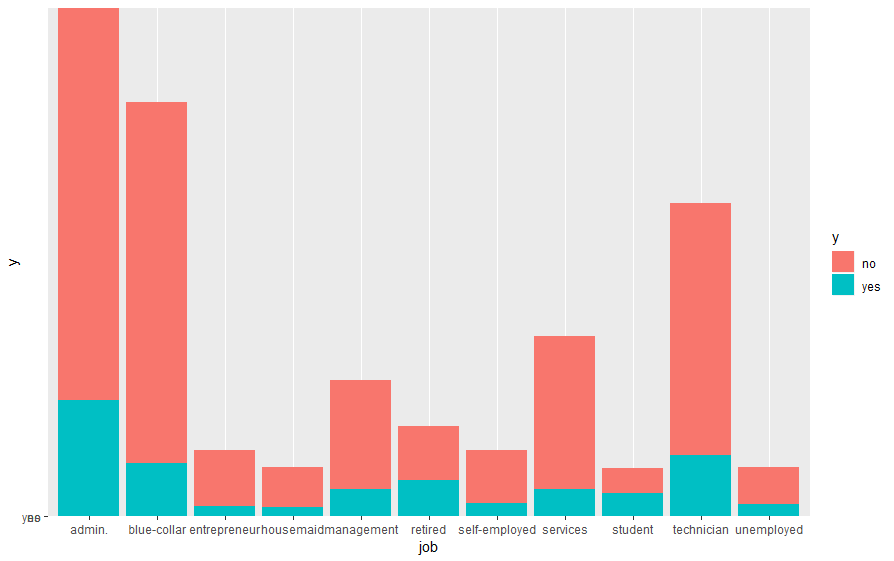
On an average per person, we have considered that the loss faced by the campaign per rejection is 1$ and gain a profit of 10$ per purchase of the product.

On that basis, we have segregated the attributes which the bank should avoid contacting or rethink about in terms of which category of people causes a loss to the bank.

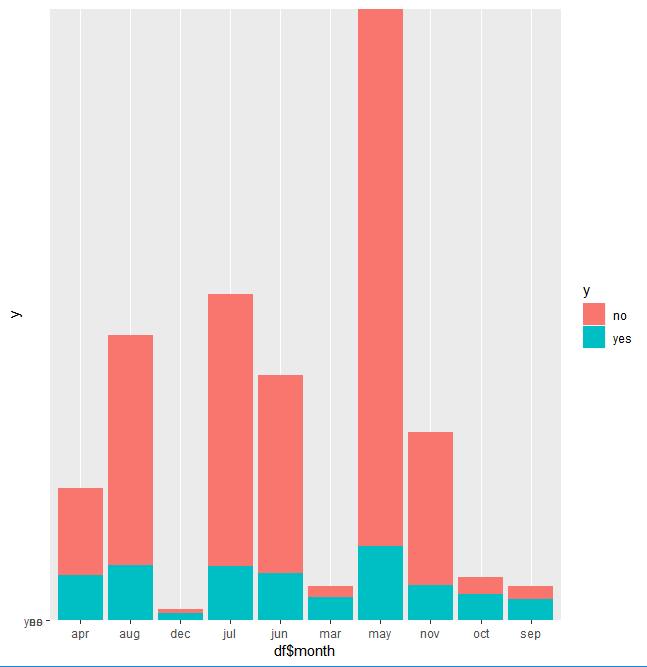
**



The % of older people (above 60) and people between the age Of (17-20) years have the higher tendency to go for the bank deposit. These people should be a priority target customers for the bank.

**

Retired and students category have the highest % of opting the campaign. This also coincides with our Age distribution observation.

**

The month of Oct, Sep, Dec and Mar have the highest % for the campaign success. This is the time for spring And fall vacation, this period should be considered for calling the respective customers.