Data Analysis on Bank Additional Dataset From UCI Repository.

https://archive.ics.uci.edu/ml/datasets/bank+marketing

Anshul Gupta

Summary-

The data is related with direct marketing campaigns of a Portuguese banking institution. The DataSet is available at https://archive.ics.uci.edu/ml/datasets/bank+marketing . The marketing campaigns were based on phone calls and with purpose to recognize potential customers to buy products or services from the bank. Given the size of banking customers it is neither practical nor cost-efficient for bank to contact every customer. The selection of the potential customers make this a classification problem where we have a dependent variable **'y'** which can be categorized as 'yes' or 'no' based on certain attributes. Yes is affirmative sign where the customer is ready to business with bank and No indicates that customer is not ready to avail any services from the bank.

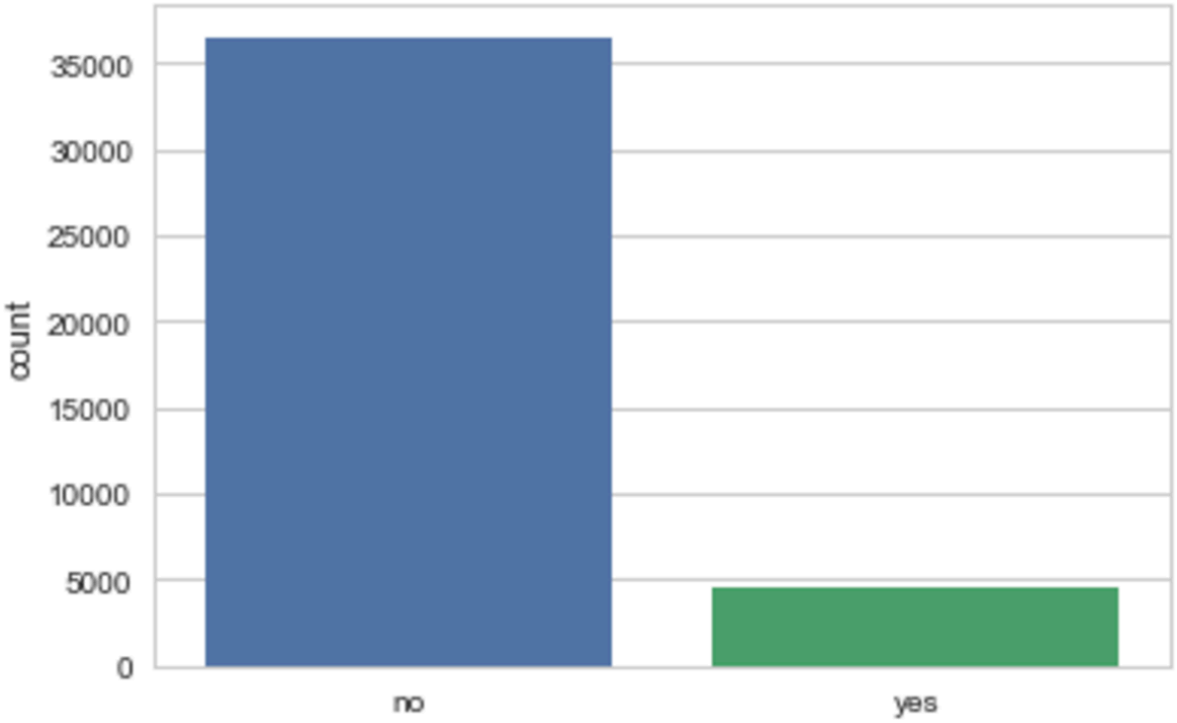
Introduction –

The bank-additionals dataset has about 41188 instances and 20 features and a response variable **'y'**. In this dataset I have implemented various Machine Learning Models to predict the product of the bank would be subscribed or not. The Data has following attributes.

Attibutes	DType
Age	Numeric(Int)
Job	Categorical
Marital	Categorical
Education	Categorical
Default	Categorical
Housing	Categorical
Loan	Categorical
Contact	Categorical
Month	Categorical
Day_Of_Week	Numeric(Int)
Duration	Numeric(Int)
Campaign	Numeric(Int)
PDays	Numeric(Int)
Previous	Numeric(Int)
POutcome	Categorical
Emp.var.price	Numeric(Float)

Cons.price.idx	Numeric(Float)
Cons.conf.idx	Numeric(Float)
Euribor3m	Numeric(Float)
Nr.Employed	Numeric(Float)
Y	Categorical

The Dataset contains 36548 'No' and 4640 'Yes'. In our case 'Yes' is vital as these are customers who are going to subscribe to bank products. So this is a class imbalanced problem. Below figure describes value counts for Yes or No.



The Evaluation Metrics to evaluate the performance of our model will be True Positive Rate(Sensitivity or TPR) and True Negative Rate(Specificity or TNR) . As the Accuracy Score will not give accurate judgement of our model as the response variable is highly imbalanced. Higher the TPR better the model can be used to predict the outcome.

TPR = TP/(TP + FN)
TNR = TN/(TN + FP)

Problem Statement-

The classification goal is to predict if the customer is going to avail the banking service or not. There are about 21 attributes out of which Age, Job, Marital Status, Education, Default, Housing and Loan are the customer features. Contact, Month, Day Of Week and Phone Call Duration are data recorded by Telemarketing campaign based Phone calls. Employment Variation Rate, Consumer Price Index and Consumer Confidence index are the social economic factors included in the Dataset . The response variable has a categorical values ‘yes’ or ‘no’.

Approach-

Data Wrangling-

There are about 10 features which have categorical values. Categorical Features are needed to be one-hot encoded to train the model. For that pandas provide a function called pd.get_dummies to One-Hot encode the categorical features to fit with various Classification and Regression Problems. By performing this our attributes increase from 21 to 64.

Correlation of attributes to determine any positive or negative correlation with respect to response variable ‘y’. Correlation describes the degree of linear relationship between two set of variables.

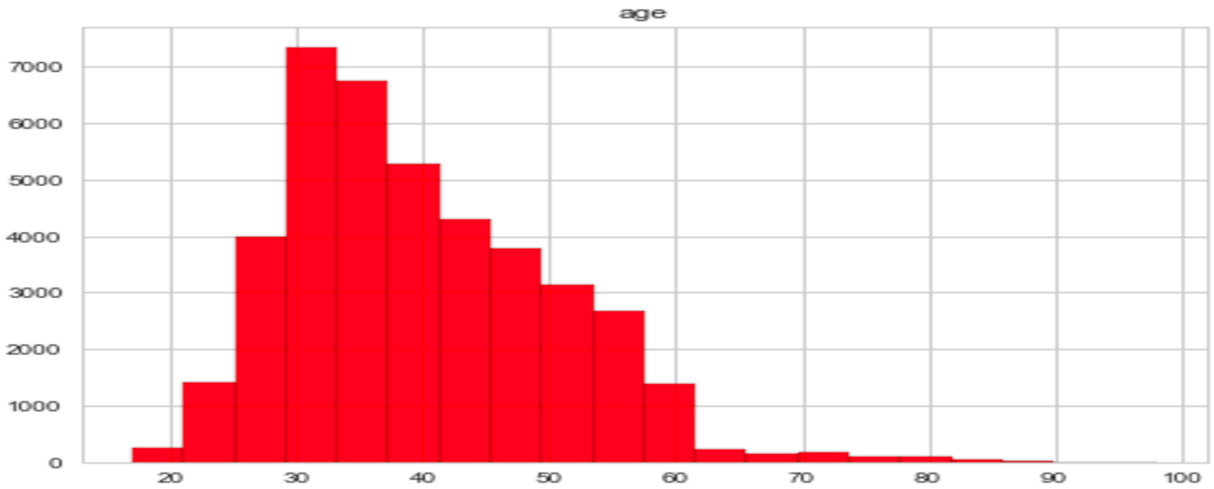
Following correlation chart shows correlation of numerical columns with response variable and among themselves. The diagram illustrates no significant correlation with respect to response variable.

Summary Statistics for Numeric Data-

	age	duration	campaign	pdays	previous
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963
std	10.42125	259.279249	2.770014	186.910907	0.494901
min	17.00000	0.000000	1.000000	0.000000	0.000000
25%	32.00000	102.000000	1.000000	999.000000	0.000000
50%	38.00000	180.000000	2.000000	999.000000	0.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000
max	98.00000	4918.000000	56.000000	999.000000	7.000000

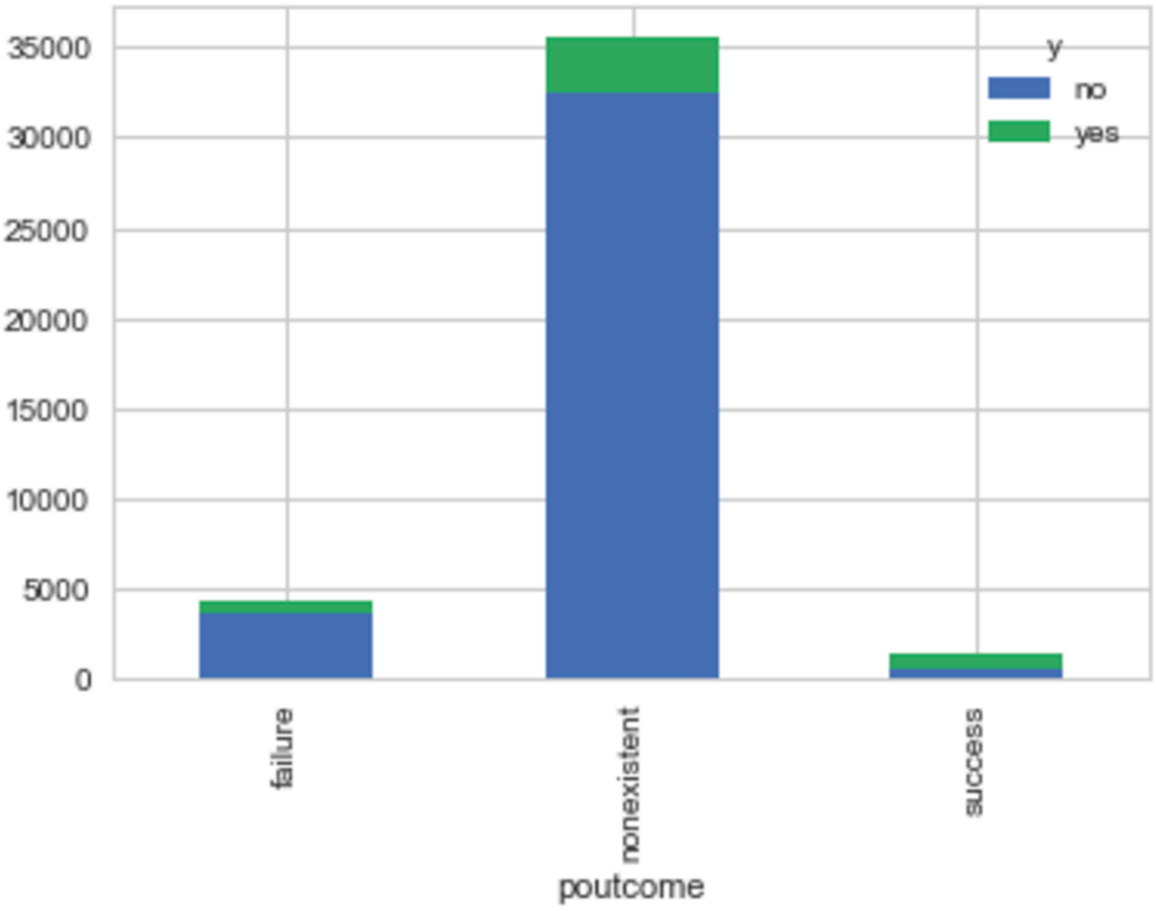
	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	1.570960	0.578840	4.628198	1.734447	72.251528
min	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	1.400000	94.767000	-26.900000	5.045000	5228.100000

Age Histogram-

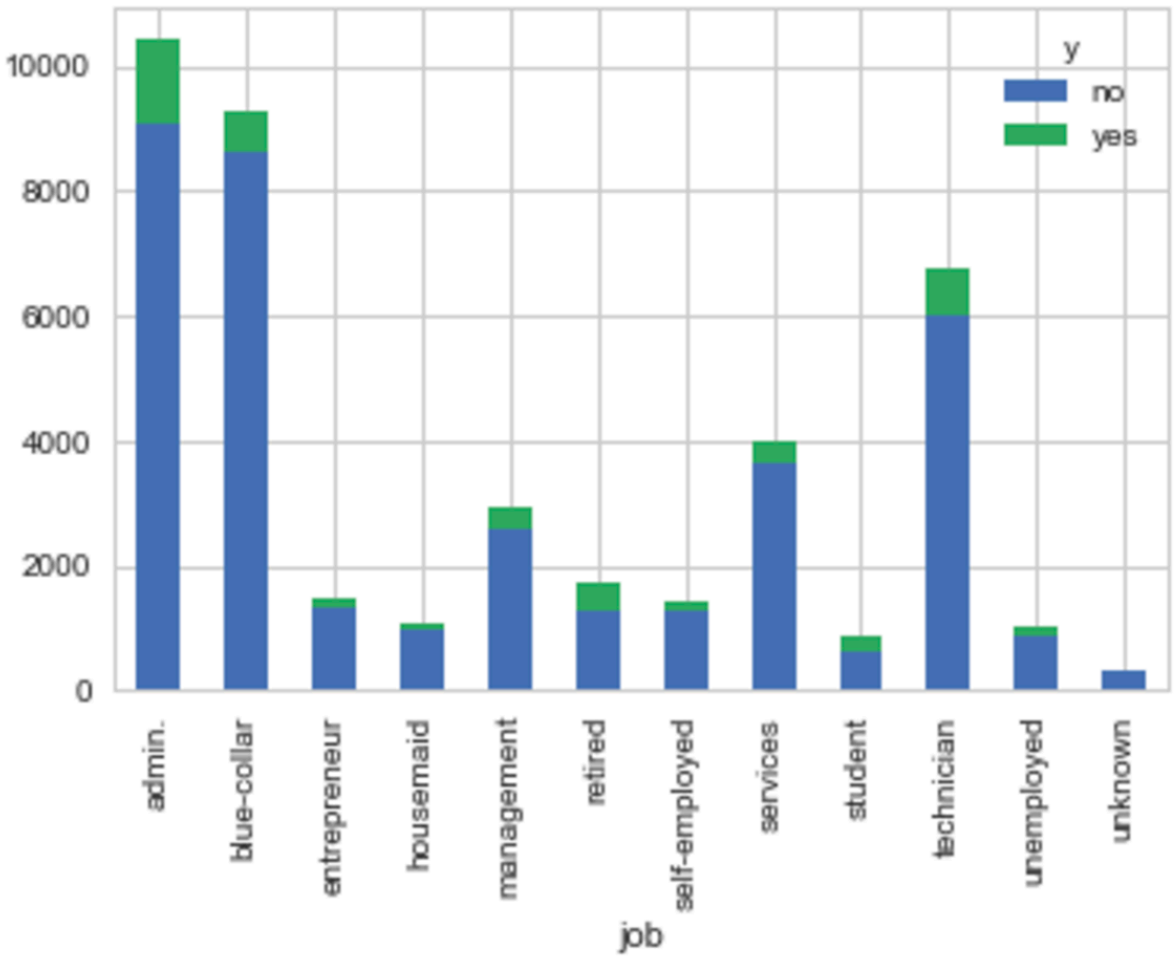


The histogram shows age group 30-40 shows maximum number of customers in the Dataset.

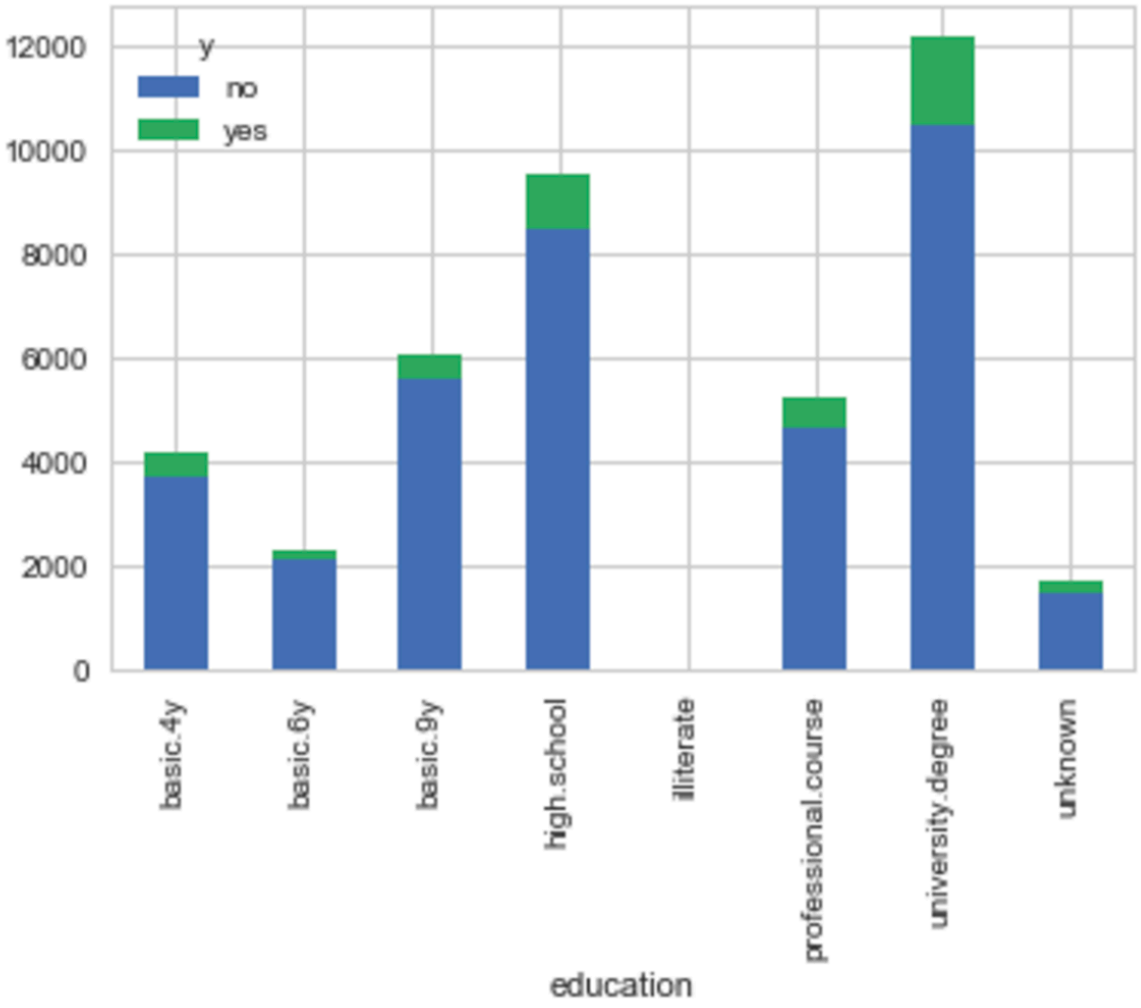
Correlation Diagram of Numerical Attributes.



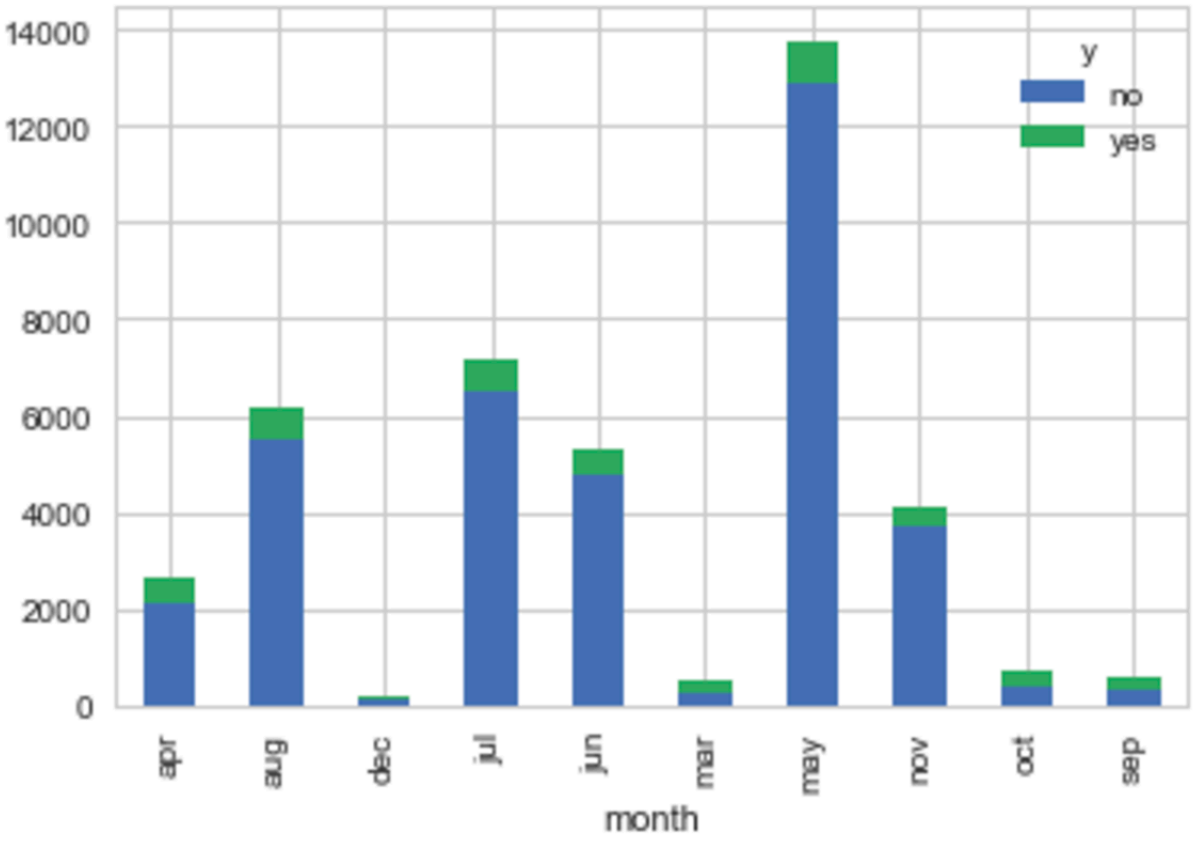
OutCome Of Previous Marketing Campaign-The plot shows that the OutCome of previous Martketing campaign is mostly non-existent.



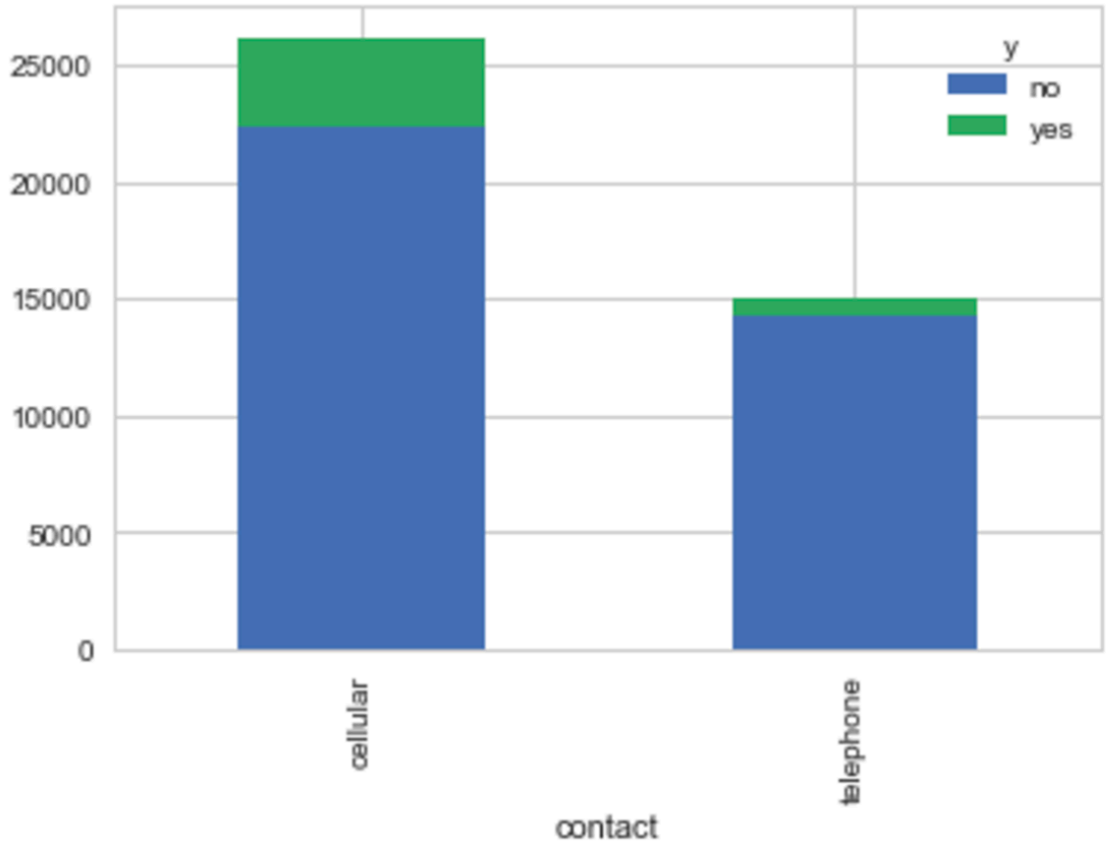
Job-The three top most jobs people were contacted here were Admin, Blue-Collar and technician which comprise more than half of the total people contacted by the TeleMarketing Company



Education-The plot shows that people with university degree were contacted the most **.**



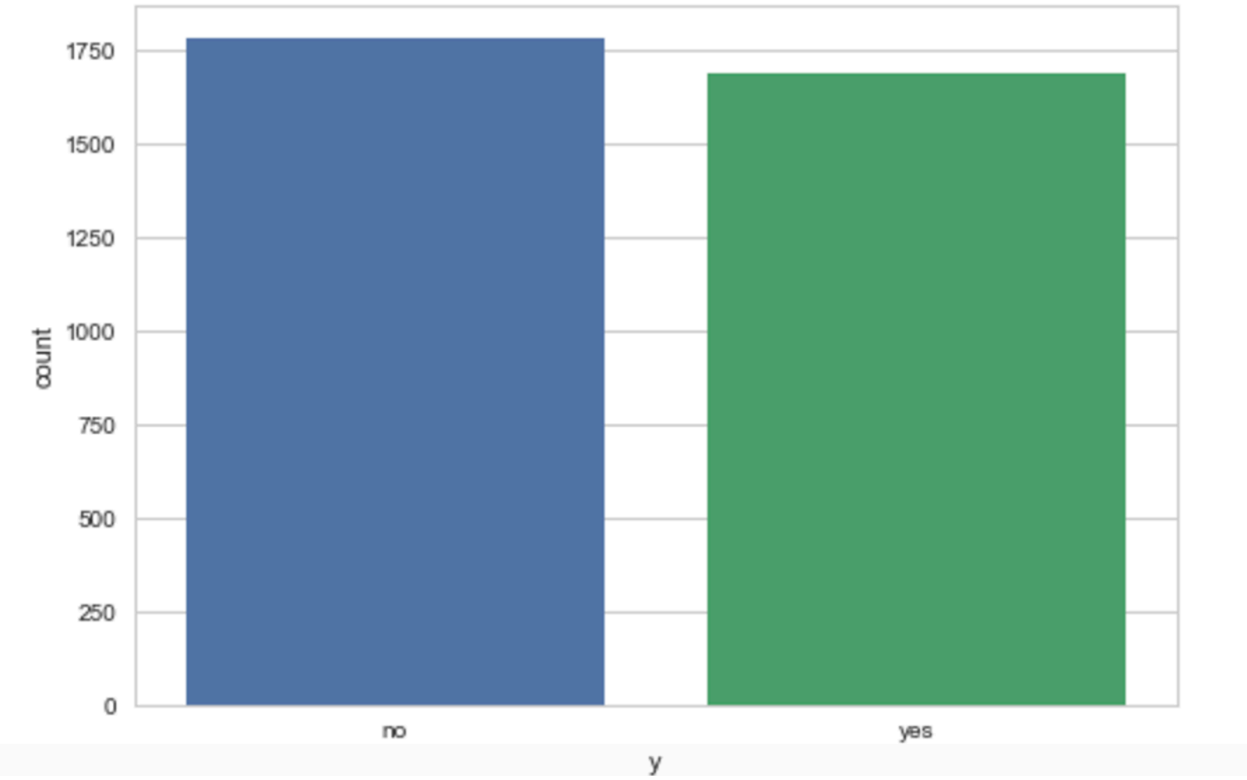
Month- The countplot shows that the month of May was where maximum calls made which is almost double than second highest month that is July.



Contact-The countplot shows that there were two types contact one was cellular and other was telephone. The bars shows yes or no for cellular and telephone. It shows cellular communication was preferred over telephone.

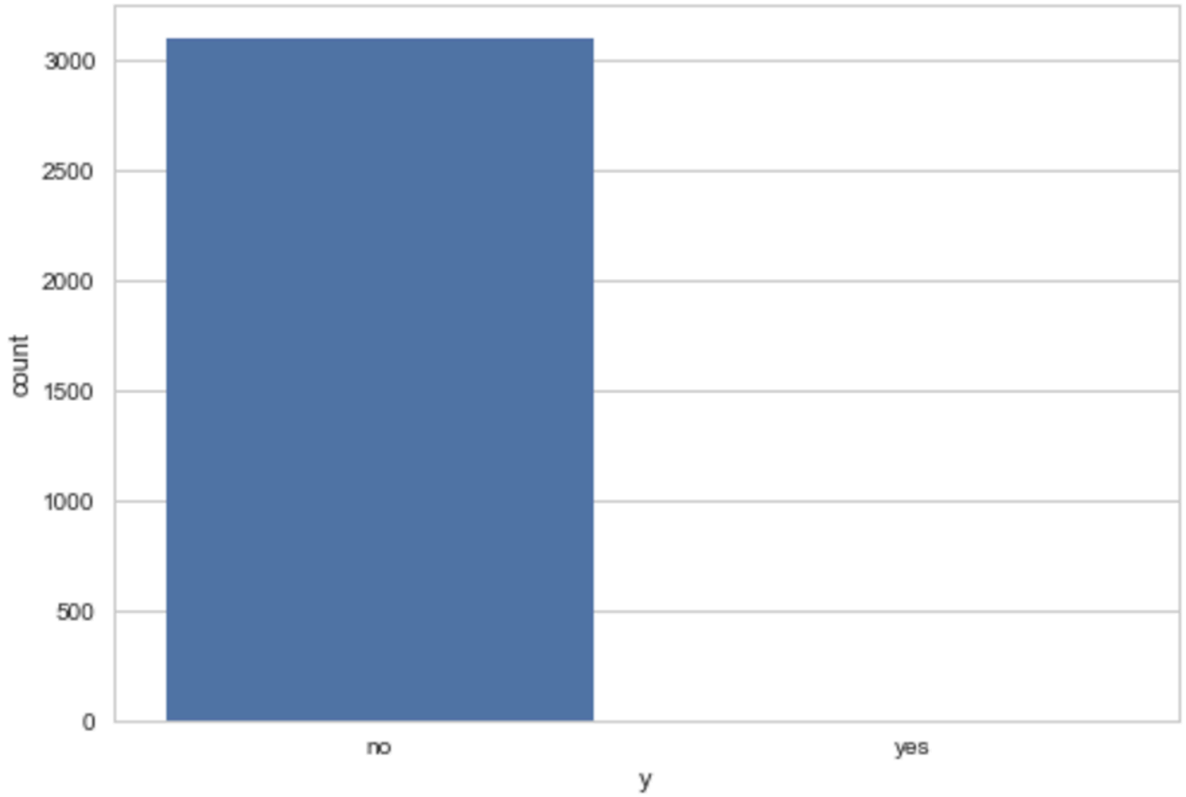
Call Duration- This attribute could highly influence the response variable. Longer the call duration the more chances are for a yes.

Two categories –
1.Duration of call is more than 600s
Response variable ‘no’ is this case is 1780 and ‘yes’ is 1684.



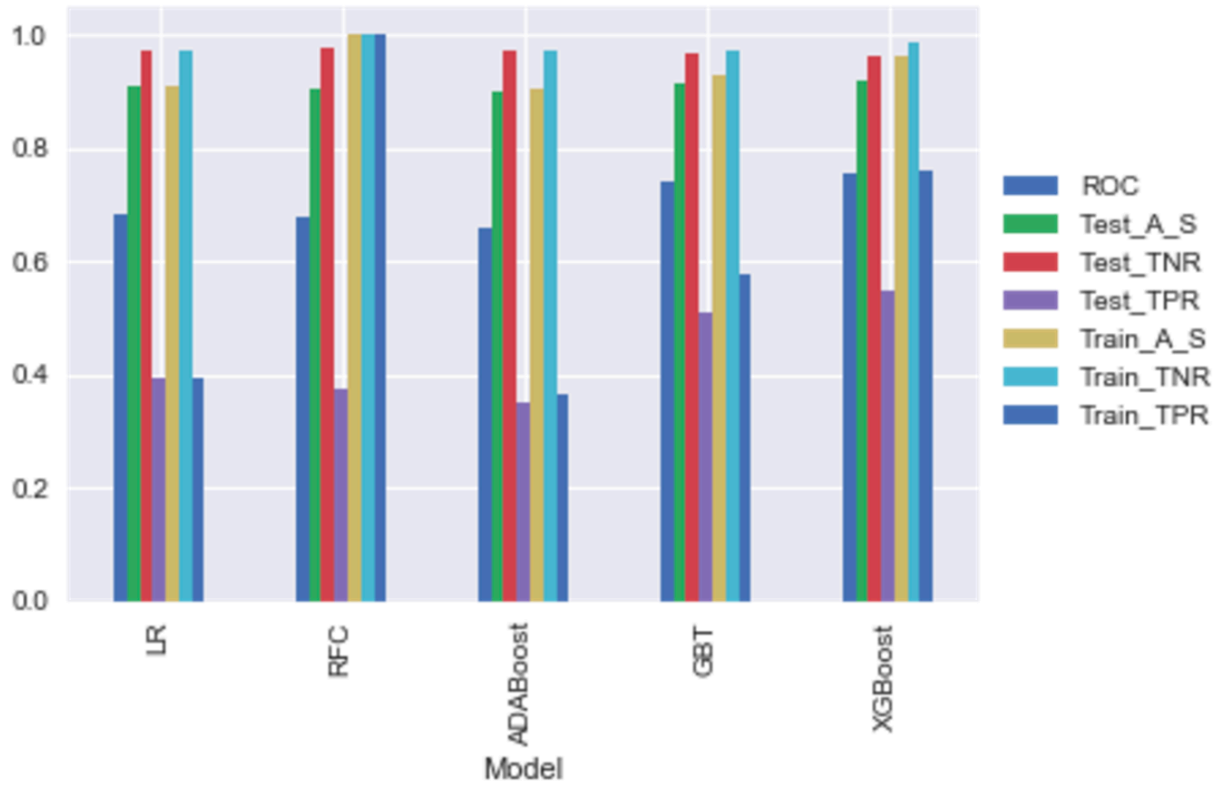
2.Duration with calls less than 50 seconds
Response variable is ‘No’ 3101 cases and yes is just one case

So we can infer that call duration plays an important in getting a positive sign for the bank. But as soon as the call is over we get to know the result and hence this can be used as bench mark that longer call duration is a sign that customer is going to avail a bank product.



Models

As this is classification problem with imbalanced Dataset so we need to fit a classification Algorithms such as Logistic Regression, Random Forest , ADABoost, Gradient Boosting and XGBoost. For tuning the hyperparameters used Grid Seach CV and cross validation to optimize the parameters such as regularization strength (C) for Logistic regression and penalize the algorithm with L2 parameter. This was done for each and every algorithm.
The below figure indicates that model is predicting a poor True Positive Rate which is the most important evaluation metric in our scenario as it indicates customer who are willing to subscribe to bank products.

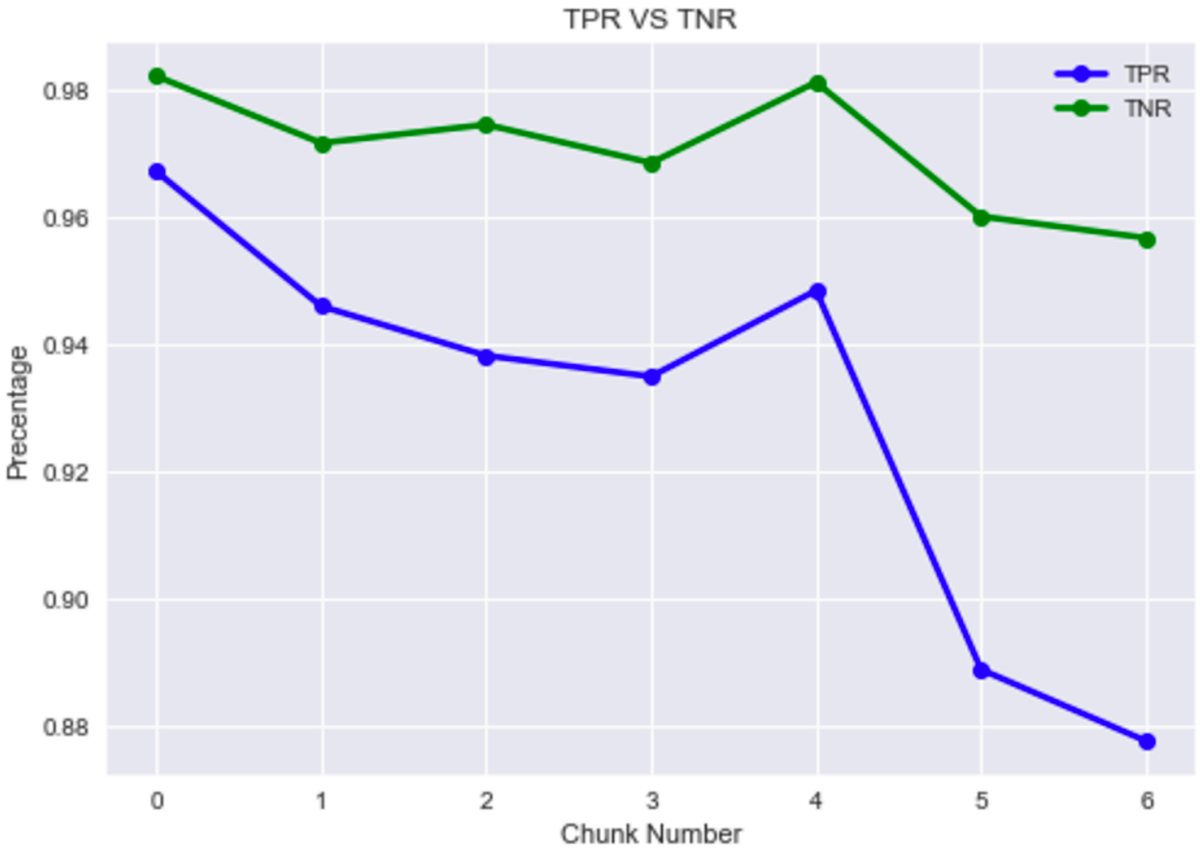


	ROC	Test_A_S	Test_TNR	Test_TPR	Train_A_S	Train_TNR	Train_TPR
Model							
LR	0.682994	0.908797	0.974464	0.391523	0.909611	0.975062	0.394089
RFC	0.676593	0.907826	0.976955	0.376231	1.000000	1.000000	1.000000
ADABoost	0.660882	0.901675	0.973663	0.348101	0.906594	0.974544	0.365755
GBT	0.739225	0.915271	0.967901	0.510549	0.928272	0.972631	0.575202
XGBoost	0.757588	0.918184	0.964888	0.550287	0.962367	0.987883	0.761392

From the above table one can conclude XGBoost best to predict the response variable but it has 55% TPR.

Optimization of Results – To optimize the results we can perform the resampling techniques. The techniques used are oversampling and undersampling.

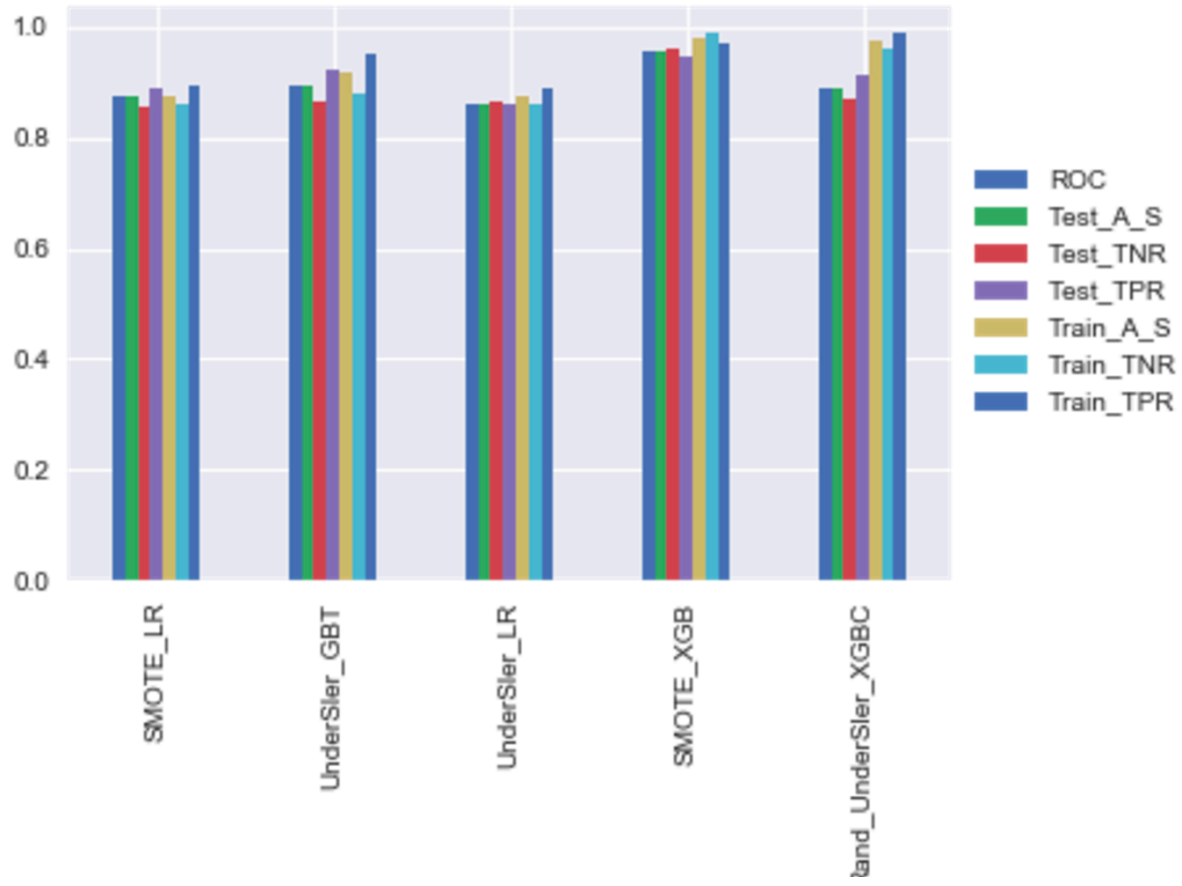
UnderSampling Technique – This method works with majority class .In this we take equal minority and majority class ie. 4640 total instances of each class as we have and then fit our model. This was done throughout the dataset and seven chunks were formed.



So on using Logistic Regression the TPR ranges from about 88% to 97% and TNR Ranges from 96% to 98%.This shows that UnderSampling has shown significant improvements in the results.

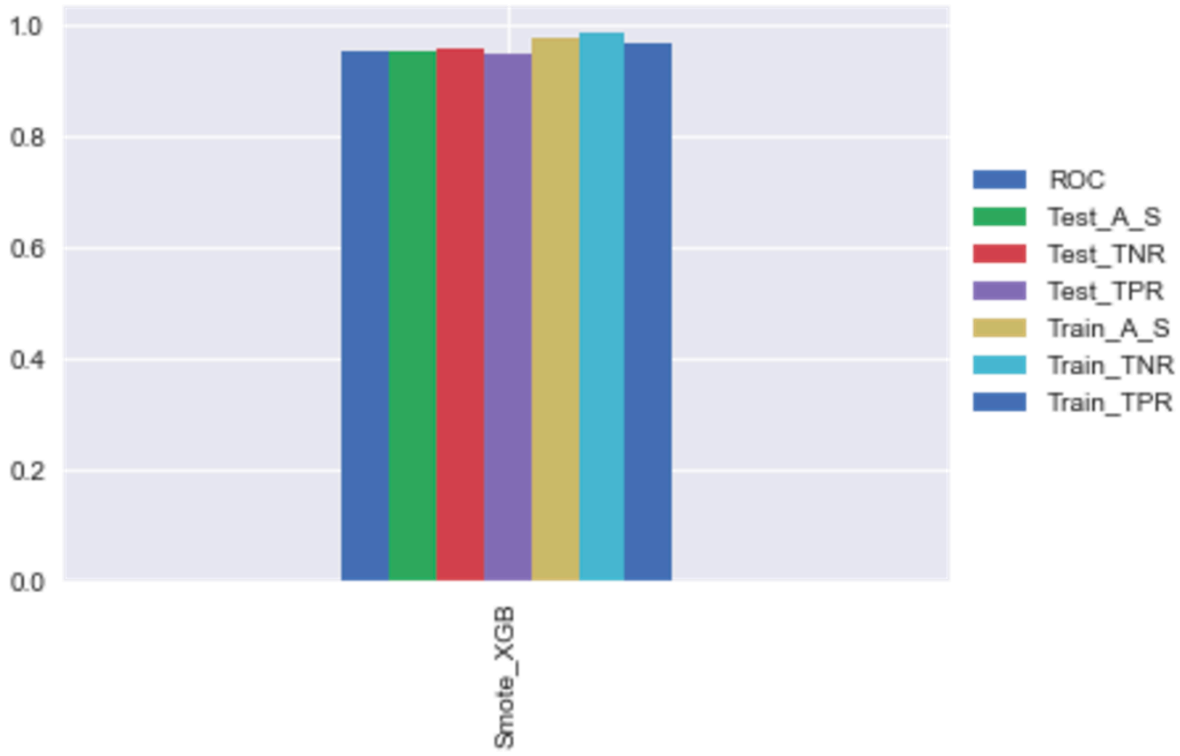
OverSampling Technique –This method works with minority class. In this we take bootstrap samples(random with replacement) of minority class. So we have 36548 samples for majority and 36548 samples of minority class. For this we use a Synthetic Data Generation which overcome the class imbalances by generating artificial data . Synthetic minority oversampling technique(SMOTE) is a synthetic data generation technique where it creates artificial data based on minority class.

	ROC	Test_A_S	Test_TNR	Test_TPR
Model				
SMOTE_LR	0.873289	0.873273	0.856674	0.889903
UnderSler_GBT	0.891915	0.891523	0.863218	0.920612
UnderSler_LR	0.861679	0.861710	0.863926	0.859432
SMOTE_XGB	0.953253	0.953258	0.958815	0.947690
Rand_UnderSler_XGBC	0.889661	0.889368	0.868179	0.911143



As we can see the results with resampling techniques are much better than algorithms applied on normal Dataset.

From the last figure we can conclude that Smote_XGB produces the best results for prediction of the response variable.



Comparison of Results With Prusty Sagariska and Hany A. Elsalamony-

The most efficient algorithm will be which has the highest True Positive Rate.

Most efficient algorithm for Prusty Sagariska was Decision Tree. The results are

Precision, Recall, F Measure,ROC Area, Performance with test data

Model / Parameter	Naïve Bayes(Modified Training set)	Decision Tree (C4.5)	Decision Tree(Modified training set)
Overall Accuracy	78.15%	93.7%	89.9%
Recall (Class Yes)	0.805	0.627	0.932
Precision (Class Yes)	0.769	0.794	0.875
F Measure (Class YeS)	0.786	0.701	0.903
ROC Area	0.851	0.931	0.939
TPR of test set	80%	48%	88%

Hany A. Elsalamony

Most efficient algorithm for Multi Layer Percepton(MLPNN)

Model	Partition	Accuracy	Sensitivity	Specificity
MLPNN	Training	90.92%	65.66%	93.28%
	Testing	90.49%	62.20%	93.12%
TAN	Training	89.16%	55.87%	91.97%
	Testing	88.75%	52.19%	91.73%
LR	Training	90.09%	64.83%	91.76%
	Testing	90.43%	65.53%	92.16%
C5.0	Training	93.23%	76.75%	94.92%
	Testing	90.09%	59.06%	93.23%

	ROC	Test_A_S	Test_TNR	Test_TPR
Model				
SMOTE_LR	0.873198	0.873182	0.856401	0.889995
UnderSler_GBT	0.891915	0.891523	0.863218	0.920612
UnderSler_LR	0.861679	0.861710	0.863926	0.859432
SMOTE_XGB	0.953253	0.953258	0.958815	0.947690
UnderSler_XGB	0.889661	0.889368	0.868179	0.911143

Results-The figure above indicates that XGBoost with Synthetic Data Generation(Smote) produces the best results.

TPR for Smote_XGB is about 95%

Recommendation To the client-

XGBoost model is the best algorithm to predict output variable. Using XGBoost with Smote gives the best TPR which is about 94.7% and is much much better than 55% which was without using any oversampling technique. Even the TNR is 95.8% which is far better than any other undersampling technique.

Future Work –

To improve the results we can use a Neural Network. We can add multiple hidden layers and make it a deep learning model and train it on a GPU to get high processing power and optimize it by using different optimizers.
We can use higher n_estimators in XGBoost with a PC with higher processing power to further improve the TPR.