BANK TELEMARKETING – CLASSIFICATION ANALYSIS

Nittin Aggarwal (111401512), Sachin Arunkumar (111447888), Sharad Sridhar (111492675)

State University of New York – Stony Brook

Abstract

Here, we analyze various classification algorithms and their use in classifying a given dataset that is skewed. We try and determine the set of people that will subscribe to a term deposit after learning from the given dataset.

Introduction

A typical strategy for marketing utilized by companies to enhance business, is to use direct marketing when targeting segments by contacting them to meet a goal. It is required usually to centralize the remote interactions in a contact center to ease operational management of campaigns. Telephone (fixed line or mobile) is one of the most widely used medium for contacting customers. This methodology is termed as telemarketing.

With technology, marketing campaigning strategies can be properly charted out to maximize customer lifetime value through the evaluation of available information and customer metrics, thus allowing to build longer and tighter relations in alignment with business demand. However, the task of selecting clients that are more likely to subscribe to an instrument is considered to be NP-hard.

Specifically, decision support systems use technology managed by bank managers and entrepreneurs to market bank and other financial instruments such as loans, bonds, mutual funds etc. DSSs are usually small-scale systems that cater to decision making process involving a single bank manager whereas for scalable systems commands use of Artificial Intelligence domain expertise to make use of various dynamic and static, intrinsic and extrinsic parameters to conclude a proper decision.

AI can play a key role in personal and intelligent DSS, allowing the extraction of predictive knowledge from raw data. The goal here is to build a data-driven model that learns an unknown underlying function that maps several input variables, which characterize an item (e.g., bank client), with one labeled output target (e.g. if client subscribes loan or not).

Our aim was to analyze and learn different classification techniques for a given dataset. We have a program written in Python using the Scikit-Learn library that we use for running these classifications.

Description

Key Highlights

Skewed Data (Class Imbalance problem)

The famous class imbalance problem is observed in the data. The target labels do not have equal number of instances for the machine learning classifier algorithms to learn appropriately and the data is highly skewed in the order 80:20 with 80 per cent clients not subscribing the deposits. Therefore, careful feature engineering is required with domain expertise to curate the features in such a way that machine learning can work appropriately in order to avoid overfitting of data.

Feature engineering

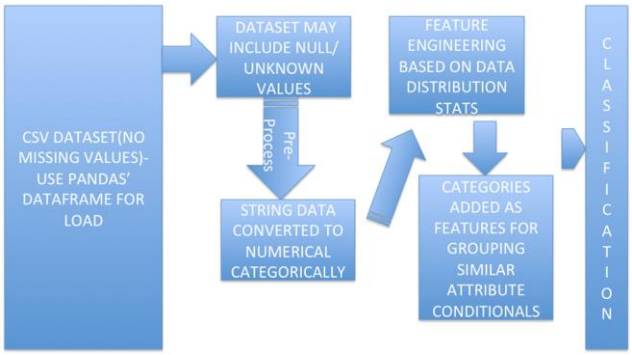
Careful preprocessing of the data is followed by feature engineering step that is allowing us to observe, study and analyze the distribution of a particular attribute and its correlation to the output target variable. This info along with some intelligent feature selection based on domain expertise has allowed to generate certain rules that have been utilized to tackle the class imbalance problem by assigning same attribute values to records satisfying some criteria based on a set of attributes resulting in categorization.

Exploration of Deep learning NN (Multi-layer perceptron)

A Taste of unsupervised learning which is a tough domain to explore and requires careful construction and use for

better results has been explored with satisfactory results though due to presence of inadequate data, a slight under performance is observed compared to the traditional classification supervised techniques. This can be attributed small dataset and result class being binary.

Structure of System developed for the Problem



Pandas data-frames are utilized to load and store data from the csv file. Though the data does not have any NULL values, unknown values exist within the data, which need to be handled. The data also has a lot of variables used as strings which need to be converted to numeric data type to be available for running most of the models which would define our baseline. This would also help in performance gains since handling integer would be cost effective against handling strings. This marks the preprocessing step of our solution.

Post this preprocessing, it is highly essential to modify data to tackle the class imbalance problem. Thus, the step commands feature engineering combined with utilization of domain expertise.

Since the number of features are not large, we can try to use some intuitive guesses and domain expertise to arrive at a bunch of features that would help in better prediction.

Following features have been modified:
Age variable has been categorized as:

- 0<age<19
- 19<age<40
- age>40

with mean age as 40 and standard deviation of 10.
Different types of campaign have been categorized as 1,2,3,4 with mean of 2.

The previously 'contacted' variable has been segregated based on the number of days contacted from current day. It is categorized as:

- if prev=0; we assign 0
- if prev>=1 and pdays<30 then 1
- if prev>=30 and pdays<35 then 2
- if prev>=35 then 3 with av pdays = 40

The above feature has been generated using combination of two related features.

The balance variable has been categorized as:

- bal<500 then 1
- 500<=bal<1000 then 2
- bal>=1000 then 3 with a mean of 1362 and std of 3044

The job types of bank-clients have been intuitively classified as:

- {management, admin,blue-collar} as 2
- {retired,services,housemaid,technician} as 1
- {unemployed,unknown} as 3
- {student,entrepreneur,self-employed} as 4

The marital job field has been classified as
married - 1
single - 2
divorced – 3

The poutcome field has been categorized as :

- other and unknown as 0
- failure as 1
- success as 3

The education field has been modified to group primary and secondary education into level 1 and tertiary education into level 3. A category for unknown is created separately to keep its impact separate.

- Other binary data was classified as 0 and 1.
- Moreover, the day of week field has been dropped as it has no relevance with the predicted variable

Following the feature engineering which is done gradually and in an adaptive fashion to curate better results, we applied various classifiers over the data.
For running the classifiers, it is quite important to decide how do we partition our data set so that we can train the model with ample samples. For this, we have divided the dataset into 2 splits with a ratio of 80:20 for each partition for test and train data sets. Following this we have to begin our exploration step for classification where we work out and tune our hyper parameters for specific models to better predict the results.

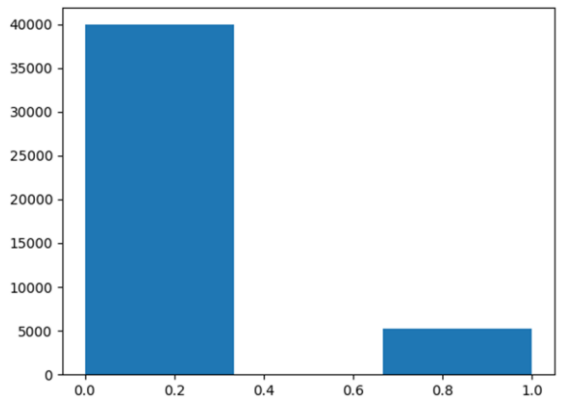
The classifiers used are:

- Logistic Regression
- KNN(K Nearest Neighbour)
- Decision Tree Classifier
- LDA(Linear Discriminant Analysis)
- Random Forest Classifier
- Gradient Boost
- Adaptive Boosting Classifier
- Neural Network(Adam solver) for multi perceptron model

EVALUATION

We ran several classification algorithms on the given data to test their performance. The data chosen for the is from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Bank+Marketing).
The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.
The data that was obtained from the aforementioned link is quite imbalanced. The distribution of data is skewed towards a single class, which made parts of the classification a little challenging. Most of the classification algorithms need samples to be distributed among the classes evenly.

DATASET



The above graph shows the skewness observed in the dataset.

The data bank-full.csv used contains 45211 records with 17 attributes which can be separated into separate categories. The attributes can be categorized as bank-client personal information attributes, social and economic context attributes and campaign related attributes. Along with this each record is labeled as yes or no which is our target decision variable ‘y’ signifying whether the bank-client has subscribed to the instrument bank-deposit or not. The problem more subtly stated is to predict the result of a phone call intended to market bank deposits that may lead to subscription or no subscription.

An important detail observed in the dataset is that there are no NULL values or missing values although unknown values can be observed.

RESULTS

Logistic Regression
Accuracy: 0.899613
Precision Score: 0.784066
Recall Score: 0.643050
F1-Score: 0.681775
ROC-AUC Score: 0.784066

K-Means
Accuracy: 0.896226
Precision Score: 0.799181
Recall Score: 0.594848
F1-Score: 0.626689
ROC-AUC Score: 0.799181

Decision Tree Classifier
Accuracy: 0.901161
Precision Score: 0.781285
Recall Score: 0.662841
F1-Score: 0.700022
ROC-AUC Score: 0.781285

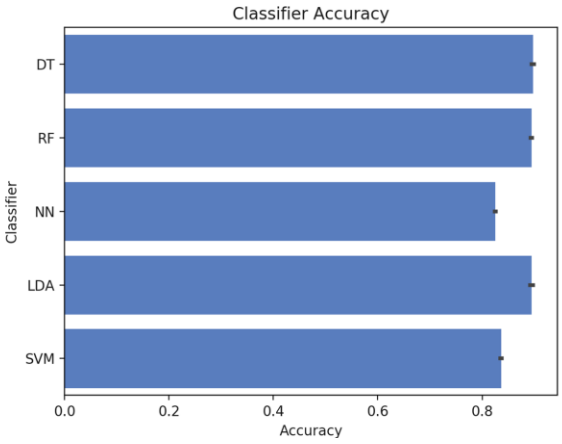
Linear Discriminant Analysis
Accuracy: 0.898963
Precision Score: 0.764939
Recall Score: 0.680820
F1-Score: 0.711533
ROC-AUC Score: 0.764939

Random Forest Classifier
Accuracy: 0.899129
Precision Score: 0.836610
Recall Score: 0.596933
F1-Score: 0.632170
ROC-AUC Score: 0.836610

Gradient Boost
Accuracy: 0.905820
Precision Score: 0.799535
Recall Score: 0.677492
F1-Score: 0.716523
ROC-AUC Score: 0.799535

Adaptive Boosting Classifier
Accuracy: 0.899309
Precision Score: 0.773933
Recall Score: 0.659694
F1-Score: 0.695706
ROC-AUC Score: 0.773933

Neural Network
Accuracy: 0.893531
Precision Score: 0.815755
Recall Score: 0.571669
F1-Score: 0.594074
ROC-AUC Score: 0.815755



Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. On most occasions, this is an acceptable performance measure to evaluate, but this is highly dependent on the dataset being symmetrical, i.e., the values of false positive and false negatives are almost same, and, the class distribution is 50/50. But seeing as how our dataset is heavily skewed, we can conclude that is not particularly useful here.

Precision - Precision looks at the ratio of correct positive observations. The formula is True Positives / (True Positives + False Positives).

Recall - Recall is also known as sensitivity or true positive rate. It's the ratio of correctly predicted positive events. Recall is calculated as True Positives / (True Positives + False Negatives).

F1 Score: The F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

ROC – AUC: The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example,
i.e. P(score(x+)>score(x−)).
ROC curves also give us the ability to assess the performance of the classifier over its entire operating range. The most widely-used measure is the area under the curve (AUC). An AUC of less than 0.5 might indicate that something interesting is happening. A very low AUC might indicate that the problem has been set up wrongly, the classifier is finding a relationship in the data which is, essentially, the opposite of that expected. In such a case, inspection of the entire ROC curve might give some clues as to what is going on.
In an imbalanced set (which is true in our case), therefore, it is better to use the ROC-AUC curve as an evaluation method, since any abnormalities are more easily detected by the score.

Analysis

SVMs generally perform good for relatively small data sets with fewer outliers. But it requires significant memory and time (one of our metrics for evaluation). Random forests comes next and may require more data but they almost always come up with a pretty robust model. Whereas, deep learning algorithms requires large datasets to work well, and it is also required to train them in reasonable time. Moreover, deep learning algorithms require much more tuning based on domain knowledge. We employed sophisticated neural networks to observe the impact they can make without much of the feature engineering. But due to the smaller training data size the performance of the Neural networks is not robust and useful with low accuracy. Moreover, the classification problem is a binary problem whereas the NN works best for multi-class classification problems.

As the ROC analysis provides the tools to select possibly the optimal models and discard sub-optimal ones independently from the class distribution. It helps in our case to evaluate the classifiers better as we have a skewed data. It

is observable that picking the Random Forest Classifiers and Neural Networks provides us with the highest area of curve with a comparatively lower accuracy. This also helps in dropping Decision tree classifier and Logistic regression classifier which are providing the higher accuracies as they depend more on class distribution.

It is also observed that the feature importance of duration field is the highest followed by the poutcome field as obtained from the random forest classifier. Duration field tells in seconds what was the call duration and poutcome which tells outcome of the previous marketing campaign. It can be inferred that longer the duration more the client is more convinced to buy the product. Also depending on the result of the previous campaign (success or failure or non-existent) would influence the result of the current one.

Constraints

As client's propensity to subscribe to the bank deposit would change over time, our model depends on the economic variables of the particular time frame and would fail to work for a dataset of another time frame. For the model to work for such a data, we would require the model to train with a sliding window of fixed number of days and then adapt the model continuously so as to make it work across time frames.

Conclusion

The most crucial objective of this project was to learn the application of various classification algorithms and their usage and impact on the predicted variable in case of highly imbalanced dataset. Moreover, it was learnt that it is not always true that higher accuracy means better prediction. Those attributes that are irrelevant to the problem need to be removed. There will be some features that will be more important than others to the model accuracy. There will also be features that will be redundant in the context of other features. In our case it can be seen the feature that was redundant was 'day_of_week'.

We anticipate that the prediction measure of the model can be greatly improved if we can link external environment data such as credit score of the client, bank rate offered during the time. Also, it is possible that if we are able to retrieve more historical data then the training can be robust and would lead to maximum potential of Neural Networks.

References

- https://pdfs.semanticscholar.org/4a27/709545cfa225d8983fb4df8061fb205b9116.pdf
- http://scikit-learn.org/stable/

To run the code, please place the downloaded data set given in the link in the same folder and run the python program as usual