UDACITY NANODEGREE MACHINE LEARNING CAPSTONE PROJECT

TOPIC:BANK MARKETTING CAMPAIGN

By

K.guru tejaswini

DOMAIN BACKGROUND:

Generally, banks used to collect a lot of information of customers and the analysis of data of a bank gives the customer relationship with the bank. Handling,of so much amount of data will really be a typical thing so in order to avoid that we go with machine learning tasks to simply our work like classification which we did in this project.

Analysis of campaign includes a lot of things to be classified and it can better be solved by using machine learning techniques one can refer the sample example for campaign analysis using machine learning is https://www.aarki.com/blog/using-machine-learning-to-predict-campaign-performance by this link

In addition, some of the banks and financial-services companies may depend only on strategy of mass marketing for promoting a new service or product to their customers. In this strategy, a

single communication message is broadcasted to all customers through media such as television, radio or advertising firm, etc... In this approach, companies do not set up a direct relationship to their customers for new-product offers. In fact, many of the customers are not interested or just don't respond to this kind of sales promotion.

The main benefit of solving this kind of problem with the help of machine learning lie in the essence of machine learning which has various techniques and approaches to deal with a complex in an easy way by classifying the things,tuning the parameters,performing evaluation metrics and many more

So,I believe that the ml has power to deal with this problem.

The similar which was solved using machine learning was as follows: https://nycdatascience.com/blog/student-works/machine-learning/machine-learning-retail-bank-marketing-data

PROBLEM STATEMENT:

The customers data is a crucial thing to analysis th campaign of a bank.So I decided to deal with the problem of finding whether a person will participate in the campaign or not.Data mining models help us achieve this. The purpose is to increase the campaign effectiveness by identifying significant characteristics that affect the success based on a handful of algorithms that

we will test (e.g. Logistic Regression, Gaussian Naive Bayes, Decision Trees and others). The experiments will demonstrate the performance of models by statistical metrics like accuracy, sensitivity, precision, recall, etc

The problem which we are going to deal is a purely classification problem because we want to classify the response of a customer as yes or no so I will use the machine learning classification algorithms like naïve_bayes,svm,decision trees.

DATA SETS AND INPUTS:

The data set was extracted from the UCI repository which has a handful of datasets to explore and select the interested dataset Data set contains the following characteristics:

Data Set Characteristics: Multivariate

Number of Instances :45211

Area :Business

Attribute Characteristics: Real

Number of Attributes :17

Date Donated :2012-02-14

Associated Tasks Classification : Missing Values? Yes,

labelled as “unknown”

Number of Web Hits 386732

The following are the inputs for our data set to solve the problem:

Attribute Information:

Input variables:

Bank client data:

1. age (numeric)

2. job : type of job (categorical: 'admin.','bluecollar','entrepreneur','housemaid','management','retired', 'selfemployed','services','student','technician','unemployed','unknown')

3. marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4. education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course', 'university. degree','unknown') 5. default: has credit in default? (categorical: 'no','yes','unknown')

6. housing: has housing loan? (categorical: 'no','yes','unknown')

7. loan: has personal loan? (categorical: 'no','yes','unknown')

Related with the last contact of the current campaign:

1. contact: contact communication type (categorical: 'cellular','telephone')

2. month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

3. duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known.

Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other attributes:

1. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) 2. 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) 3. previous: number of contacts performed before this campaign and for this client (numeric) 4. poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Output variable (desired target):

1. y - has the client subscribed a term deposit? (binary: 'yes','no')

Missing Attribute Values:

There are several missing values in some categorical attributes, all coded with the "unknown" label. These missing values can be treated as a possible class label or delete.

The distribution of classes for the attributes vary in wider range which I mentioned in the brackets of the attribute.

Expected output to solve this problem:

The problem can be solved by finding whether a particular person is interested in the campaign or not.

Solution Statement

We check for quality of given data and perform data cleaning. To check if the model created is good, We will split the data into training and validation sets to check the accuracy of the best model.(i.e. wel split the given training data into two parts,70% of which will be used to train our models and 30% we will hold back as a validation set)

As described in above section, there are several non-numeric columns that need to be converted. Many of them are simply yes/no categories (like housing). These can be converted into 1/0 (binary) values. Other columns, like profession and marital, have more than two categories, and are known as categorical variables. The recommended way to handle such a column is to create as many columns as possible values (e.g.

profession_admin, profession_blue-collar, etc.), and assign a 1 to one of them and 0 to all others. These generated columns

are sometimes called dummy variables, and we will use the pandas.get_dummies() function to perform this transformation.

So let's pick few algorithms to evaluate.

- Logistic Regression (LR)
- Classification and Regression Trees (CART)
- Gaussian Naive Bayes (NB)
- Support Vector Machines (SVM)
- Random Forests (RF)
- XGBoost (XGB)

We are using 5-fold cross validation to estimate accuracy. This will split our dataset to 5 parts, train on 4 and test on 1 and repeat for all combinations of train-test splits. Based on the info used to split my data into training and testing and find out the results on them by applying machine learning algorithm.

Benchmark Model

The given dataset is a supervised learning problem for which tree type models perform a lot better than the rest. So we will pick Extreme Gradient Boosting (XGB) as benchmark and try to beat the benchmark with hyperparameter tuning. We will also

try Ensemble methods if the hyperparameter tuning does not improve the scores.

The missing values in the data are marked as unknown so I used to delete them because I believe that there is no use of such value present in the data.

Actually I would like to test the benchmark model accuracy with the different classifiers I used to compare with benchmark model like decision trees,Gaussian naïve bayes and also on ada boost etc

Evaluation Metrics

The data set which I have taken was purely an imbalanced dataset. An imbalanced data set is nothing but the data set has the classes interval is not uniform and has distribution of classes over the data set so I used the following metrics to evaluate

The performance of each classification model is evaluated using three statistical measures; classification accuracy, sensitivity and specificity. It is using true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The percentage of Correct/Incorrect classification is the difference between the actual and predicted values of variables. True Positive (TP) is the number of correct predictions that an instance is true, or in other words; it is occurring when the positive prediction of the classifier coincided with a positive

prediction of target attribute. True Negative (TN) is presenting a number of correct predictions that an instance is false, (i.e.) it occurs when both the classifier, and the target attribute suggests the absence of a positive prediction. The False Positive (FP) is the number of incorrect predictions that an instance is true. Finally, False Negative (FN) is the number of incorrect predictions that an instance is false..

Classification accuracy is defined as the ratio of the number of correctly classified cases and is equal to the sum of TP and TN divided by the total number of cases (TN + FN + TP + FP).

Precision is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP).

Recall is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN).

Sensitivity refers to the rate of correctly classified positive and is equal to TP divided by the sum of TP and FN. Sensitivity may be referred as a True Positive Rate.

Specificity refers to the rate of correctly classified negative and is equal to the ratio of TN to the sum of TN and FP

Project Design

I used to solve this problem will be in the following order:

- Exploring the Data
  - Loading Libraries and data
  - Peek at the training data
  - Dimensions of data
  - Overview of responses and overall response rate
  - Statistical summary
- Data preprocessing/cleaning
  - Preprocess feature columns
  - Identify Feature and Target columns
  - Data cleaning
  - Training and Validation data split
  - Feature Scaling - Standardization/Normalizing data
- Evaluate Algorithms
  - Build models
  - Select best model
  - Make predictions on the validation set
  - Feature importance and feature selection
- Tuning the model to Improve Result
- Final conclusion through evaluation metrics