Medumo - Identifying Patient Engagement Patterns

Zhonghao Guo, Zihao Yuan, Tianzhi Wu, Jialiang Shi
{gzh1994,yuan1z,tianzhi,markshi}@bu.edu

1. Project Description

This project aims to build a model to predict whether a customer will show up to an appointment or not. Features includes demographics characteristics (age, gender, etc), time-related factors, engagement channels (mobile, web, email, etc.), engagement events (notifications, web-clicks, etc).

The main question to be answered is “Which factors (and how much) affect patients that don’t show up to an appointment”. This is critical to determining whether a patient’s medical care will be successful or not.

Other strategic questions to be answered includes:

(1) “Are demographic variables indicative of patients more/less likely to cancel appointments at the last minute”

(2) “What other factors are responsible for driving patient engagement?”

2. Proposed Work

2.1 Proposed Phases of Work

There are roughly four major phases of this project:

(1) Data Representation
(2) Comparative Analysis Research on Models
(3) Models Building and Finetuning
(4) Feature Importance Analysis

2.2 Model

(1) Logistic Regression
(2) Support Vector Machine
(3) Decision Tree
(4) Neural Networks

2.3 Objective

The Ideal objective is to implement all these models by ourselves, and try to finetune the hyperparameters to achieve better performance if time allows. Otherwise, we would only implement part of these models and use existing libraries to do the job.

3. Exploratory Data Analysis

3.1 Dataset Overview

(1) Size

The size of training set is 1996, and test set size is 221. Although the number of training sample is relatively small, the number engagement events associated with each training sample is quite large, which includes over 170,000 events records in total, and about 88 events for each training sample on average.

(2) Data Imbalance

There is major data imbalance problem in the training set that about 90% of training samples are labeled negative (showed up), and only 10% are labeled positive (cancellations within 3 days of colonoscopies and no show).

(3) Missing Data

Engagement Events		Enrollment	
Hospital Id	0.000	Hospital Id	0.000
Patient Id	0.000	Patient Id	0.000
Event Date	0.000	Registration Date	0.000
Event Time	0.000	Procedure Date	0.000
Event Name	0.000	Email	0.000
Message Id	0.620	SMS	0.000
Module Id	0.740	Date of Birth	0.298
Event_Desc	0.063	Gender	0.469

Table 1. Missing Data

According to Table 1, we know that for engagement events, there are a lot of missing data in Meassage_Id, Module_Id, and Event_Desc. However, we claim that it’s not an issue, because by looking into the data structure, we understand that whether a data is missing or not is strongly connected with the type of event it belongs to. For example, Message_Id is not missing only if the type of event is Notification_Sent. We can use some representation methods to eliminate the missing data in the engagement events data.

On the other hand, for the enrollment information, 29.8% Data of Birth is missing, and 46.9% of Gender is missing. That’s an issue that we need to handle.

Methods that can deal with missing data includes filling missing values with some prior values(eg. mean), forward/backward padding, interpolation or simply dropping samples with missing data.

3.2 Feature Representation

There are 3 separate tables to be integrated, and how represent data in a feature vector can make a huge difference in the modeling performance.

We want to try out several data representation techniques to preprocess our data, and see how they behave in different models separately.

4. Approaches Analysis

4.1 Logistic Regression

The hyper parameters that we need to choose within ***sklearn.linear_model.LogisticRegression*** includes tol (Tolerance for stopping criteria.), and class_weight (since our data is unbalanced).

As we know, logistic regression is very interpretative in the features. We can interpret the regression coefficients as indicating the strength that the associated factor (i.e. explanatory variable) has in contributing to the utility - or more correctly, the amount by which a unit change in an explanatory variable changes the utility of a given choice.

We can also use nonlinear feature mapping methods to classify a possibly non-linear separable case for the training set.

4.2 Support Vector Machine

We plan to use the Support Vector Machine Classifier implemented in scikit-learn. The ***sklearn.svm.svc*** has several neccessary parameters which are in need of finetuning. We would choose the default 'rbf' kernel. The cache_size should be as large as possible because we have enough memory on SCC. If there is data inbalance, the class_weight should be carefully chosen. The gamma(G) and Penalty parameter of the error term(C) weight most among all parameters. We would use *Grid Search* Module in sklearn. By setting a dict of {C,G}, the optimizer will automatically evaluate the precision of model by cross-validation and return the optimal params back.

4.3 Decision Tree

Another effective module for classification problem is decision tree. The most important parameter we will focus on, is the information entropy Ent(D). Ent(D) of a feature represents how much this feature is related to the category. We always choose the feature with smallest Ent(D) to make decision.

In case of numerous number of samples, pruning strategy is helpful to reduce the scale of the tree which estimate the accuracy to decide whether to mark this node as a leaf-node.

The loss of feature is another problem to face, which could be solved by giving lower weight to sample with a lost feature.

Son-node could not only be decided by one feature. We can use linear combination of multiple features to make a more accurate decision and a more complicated decision boundary.

4.4 Neural Networks

The neural network has been proved effective on non-linear separable data. Our dataset is relatively small (1996 training + validation samples), without much high dimensional input features, so we assume very deep neural networks, convolutional net, and recurrent net are not that applicable in this project. A simple MLP shall be good enough. Finetuning hyperparameters is mainly what we would do. We will test on: # of hidden layers, # of nodes in each layer, learning rate, activation function (tanh or relu), optimizer (bgd, sgd or adam) and batch size. We will try different combinations and choose the model that gives best prediction.

5. Evaluation Criterion

By trying out different models to classify whether a patients would show up or not, we hope to show that the best model can achieve accuracy, precision, and recall are all higher than the 70%.

6. Timeline and Leaders

Task	Deadline	Lead
Exploratory Data Analysis	Nov.20	Jialiang Shi
Research on Models	Nov.27	Zhonghao Guo
Implement & Finetune Models	Dec.12	All
Feature Importance Analysis	Dec.25	Zihao Yuan
Report and Presentation	Dec.20	All

References

[1] Working with missing data, pandas 0.23.4 documentation