GROUP #8: Medumo - Identifying Patient Engagement Patterns

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

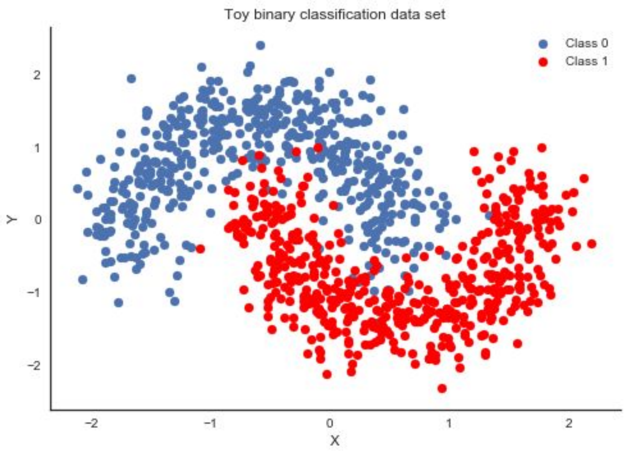

Figure 1.binary classification problem

1. Project Task

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(registration date, procedure date), engagement channels (E-mail, SMS etc.), engagement events (notifications, etc).

One thing we are interested 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. Related Work

Machine learning has become a major approach for data analysis, pattern recognition, classification, and natural language processing (NLP). Various research efforts have been put into machine learning algorithm development and apply machine learning techniques to practical problems. For the discriminative models in machine learning, the logistic regression, SVM, random forest and neural network are four remarkable tools. Schuldt et al. apply support vector machines (SVM) to recognize human actions in videos through local space time features capturing [1]. Liaw et al. describe using random forest to do classification and regression [2]. [3] use neural network to classify subcellular structures in fluorescence microscope image. For a specific classification problem like identifying Patient engagement patterns, we will try all these four algorithms and pick the one that result in best prediction. Following that, we will optimize the approach we select to further improve our model’s performance.

3. Approach

The given dataset consists of m training samples. We denote them by $(x^{(i)}, y^{(i)})$, which $x^{(i)}$ is an input with n-dimensional features, and $y^{(i)}$ is the label of the datapoint, $y^{(i)} \in (0, 1)$. What we are asked to implement is a classification model, so that we plan to adopt supervised learning algorithms. Here are 4 typical algorithms:

3.1 Logistic Regression

-Hypothesis:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

“y=1” if $h_\theta(x) \geq 0.5$, “y=0” if $h_\theta(x) \leq 0.5$

θ: learnable parameters

-Cost Function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}, y^{(i)})$$

$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)} + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

-Goal: Minimize cost $minJ(\theta)$

3.2 Support Vector Machine

SVM aim to find the hyperplane with the maximum margin to classify positive and negative datasets. Take the hard-margin SVM for example:

-Hyperplane model:

$f(x) = \omega^T x + b$

-Objective Function:

$$L(\omega, b, \alpha) = \frac{1}{2}||\omega||^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(\omega^T x_i + b))$$

$\alpha = (\alpha_1; \alpha_2; ...; \alpha_m)$ is Lagrange multiplier

-Goal: Minimize L

3.3 Random Forest

A decision tree is a tree-like structure, where each non-leaf node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf node holds a class label. Algorithms for constructing decision trees: choosing a variable at each step that best splits the set of items. For example, the information gain is used by the ID3, C4.5 and C5.0 tree-generation algorithms.

-Entropy:

$$H(T) = I_E(p_1, p-2, ..., p_J) = -\sum_{i=1}^{J} p_i \log p_i$$

-Information Gain:

$$-\sum_{i=1}^{J} p_i \log p_i - \sum_{a} p(a) \sum_{i=1}^{J} -Pr(i|a) \log Pr(i|a)$$

3.4 Neural Network

-Input: $x = [x_1, x_2, ..., x_m]^T$

-Output: $h_\theta(x) = g(\theta^{(i-1)}a)$, where g(z) is activation function and i is the # of layers.

-Cost function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\theta(x^{(i)}))_k + (1 - y_k^{(i)}) \log(1 - (h_\theta^{(i)})_k)]$$

$$+\frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(\theta_{ji}^{(l)})^2$$

4. Dataset and Evaluation Metric

4.1 Dataset

The raw dataset has 1996 training samples (labeled) and 222 test samples, as is shown in Table1. Among these total 2217 labeled data, we have complete hospital id, patient id, registration date, procedure date. Another four very important features, gender, birthday, SMS setting and E-mail setting, are stored as enrollment data. However, bunch of them are missing. Table2 summarizes the missing data situation. The reference sheet contains totally 408 rows of caution message. There is no patient id for each caution and they are represented in natural language. In engagement events, we have user activities, like node view, user clicked buttons, notification sent, etc. Totally over 170k events are recorded.

Training (pos/neg)	Test (pos/neg)	Total (pos/neg)
234/1762	0/222	234/1984

Table 1. Dataset summary

Engagement Events		Enrollment	
Hospital Id	0.0%	Hospital Id	0.0%
Patient Id	0.0%	Patient Id	0.0%
Event Date	0.0%	Registration Date	0.0%
Event Time	0.0%	Procedure Date	0.0%
Event Name	0.0%	Email	0.0%
Message Id	62.0%	SMS	0.0%
Module Id	74.0%	Date of Birth	29.8%
Event_Desc	6.3%	Gender	46.9%

Table 2. Missing Data

4.2 Pre-processing

As we mentioned before, age, gender, SMS and E-mail settings are four key features that can not be ignored. However, only part of them are included in enrollment dataset. So we have to do data cleaning. *Date of birth* are transformed to *Age* (in days) and the *enrollment date*, *procedure date* together are transformed to *Waiting Time*. We find that *notification message* is the most determinant item among all engagement data. There are over 200 notifications with just serial number but no exact notification content. We keep the first 30 most notifications and combine them into our training dataset. Thus, the training dataset $D^{\text{train}} \in R^{964 \times 37}$ and test dataset .$D^{\text{test}} \in R^{208 \times 37}$ Table 3 and 4 give us a summary and tiny example of dataset we use.

Training (pos/neg)	Test (pos/neg)	Total (pos/neg)
103/861	0/208	103/1069

Table 3. Clean Dataset summary

Patient ID	Hospital	Waiting	Email	SMS	Age	Gender	Msg 73	Msg 73	...	No Show/

	ID	Time					85	86		Late Cancel
87255	19	1	0	0	22585	1	1	0	...	0
85877	19	5	1	1	20408	0	1	0	...	0
85281	19	1	0	1	28128	1	0	0	...	1
85267	19	8	0	1	24411	1	0	1	...	0
...	...	...	...	...	...	...	...	...	...	...

Table 4. Dataset Sample

4.3 Evaluation Metric

The task of the project is to test the accuracy of patient engagement pattern in four different approaches. The raw test case has 222 negative samples. When enrollment and engagement features bring in, the number of samples will accordingly reduce. For each algorithm, we pre-process the data slightly differently. The number of negative prediction will be divided by the whole test case, giving us the test accuracy.

5. Results

5.1 Random Forest

We adopted Random Forest Classifier from sklearn library.

Input

964 training samples with 16 features, including Hospital_19, Hospital_24, Hospital_30, Registration_Date, Procedure_Date, Date_of_Birth, Email_False, Email_True, Email_Missing, SMS_False, SMS_True, SMS_Missing, Gender_Female, Gender_Male, and Gender_NaN.

Output

Showup Flag (0/1)

Grid_Search_Grid

'max_depth': (2,3,4,5,6,7,8,9,10)

'Min_samples_split': (2,3,4,5,6,7,8,9,10)

'Min_samples_leaf': (1,2,3,4,5)

Best Model Params

Max_depth = 2

Min_samples_leaf = 3

Min_samples_split = 2

Best Model Accuracy

Training Accuracy = 88.3%

Testing Accuracy = 85.3%

5.2 Logistic Regression

With same input/output format as Random Forest Classifier, the second classifier we used is Logistic Regression Classifier from sklearn library.

Grid_Search_Grid

C = (1e-3,1e-2,1e-1,1,1e1,1e2,1e3)

Penalty = (‘l1’,’l2’)

Best Model Params

C = 1e-1

penalty = ‘l2’

Best Model Accuracy

Training Accuracy = 88.3%

Testing Accuracy = 88.3

Feature Importance Analysis

- Positive Correlation Features:
  Hospital_24: 0.156
  Gender_Famale: 0.018
- Negative Correlation Features:
  Hospital_30: -0.134
  Hospital_19: -0.034
  Gender_Male: -0.032
  Procedure_Date: -0.032

5.3 SVM

The module we use is the soft margin support vector classifier embedded in scikit learn.

Input: 964 training samples. 208 test samples. x:6 features, including Hospital Id, Waiting Time, Email(0/1), SMS(0/1), Age, Gender(0/1). y:Label(0/1).

Output: 208 prediction labels in 0/1.

Parameters: Fine Tuned using grid search.

C=1, class_weight=’balanced’, gamma=0.125, kernel=’poly’.

Relative importance of each attribute:

Age, Waiting Time, Gender, SMS, Email, Hospital Id.

Result:

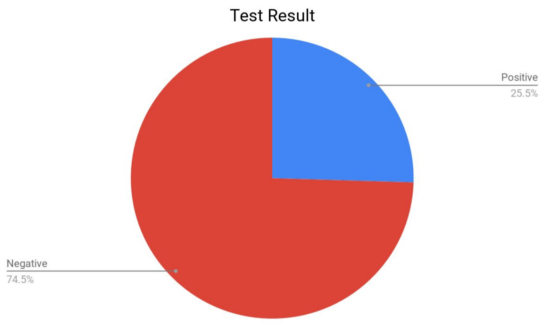

Figure 2. Prediction result by SVC

training accuracy: 0.71, test accuracy: 0.75

5.4 Neural Network

We implement the neural network on tensorflow.

Network Structure:
Input layer: 37 nodes
Hidden layer 1: 32 nodes
Hidden layer 2: 16 nodes
Output layer: 2 nodes
Activation function: ReLU
Optimizer: Adam, batch gradient descent(BGD)
Learning rate: 0.001
Loss function: cross entropy loss
Training epoch: 200

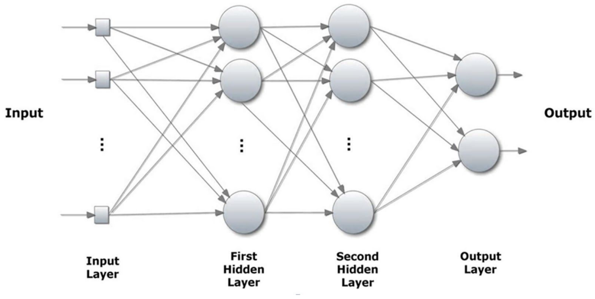

Figure 3. neural network structure

Input: 964 training samples. 208 test samples.
x: 37 features, including Hospital id, waiting time, Email(0/1), SMS(0/1), Age, Gender(0/1), message(in total 31).
y:Label(0/1).
Output: 208 prediction labels in 0/1.
Results:
training accuracy: 0.94, test accuracy: 0.89

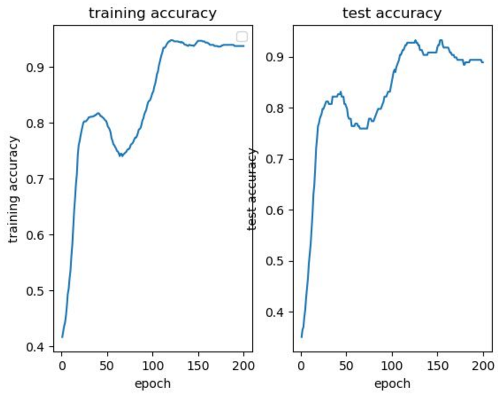

Figure 4. accuracy curve

6. Conclusions

In this project, we apply four machine learning methods on a patient dataset, predicting whether they will show up or not. Also, we try to analyse what features have more determinant effect. We clean the raw dataset to the form we want, implement four prediction models, finetune the parameters and evaluate their performances. We hope the project will help Medumo Inc. explore deeper in data collecting and mining, generating superior model with more high quality training data.

7. Timeline and Roles

Task	Deadline	Lead
SVM & Neural Network	12/01/18	Zhonghao Guo/Jialiang Shi
Logistic Regression & Decision Tree/Random Forest	12/01/18	Tianzhi Wu/Zihao Yuan
Finetuning and Evaluation	12/08/18	All
Report and Presentation	12/11/18	All

References

1) Schuldt, Christian, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE, 2004.
2) C. J. C. Burges. A tutorial on support vector machine for pattern recognition. Data Min. Knowl. Disc. 2 (1998) 121.
3) P. Xu, A. K. Chan. Support vector machine for multi-class signal classification with unbalanced samples. Proceedings of the International Joint Conference on Neural Networks 2003. Portland, pp.1116-1119, 2003.
4) Working with missing data, pandas 0.23.4 documentation.
5) Donald B. Rubin. Inference and missing data. Biometrika, Volume 63, Issue 3, 1 December 1976, Pages 581–592.
6) Bernhard Scholkopf, Alexander J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press Cambridge, MA, USA.
7) S. B. Kotsiantis, D. Kanellopoulos and P. E. Pintelas, Data Preprocessing for Supervised Learning. International Journal of Computer Science Volume 1, 2006.