University of British Columbia

Project Report - STAT 538A - Fall 2020

Who Got Hepatitis

Zhipeng Zhu

December 8, 2020

Introduction

Hospitals usually collect data from various laboratory testing of visitors to build a personal profile or help diagnosis. In this report, I get a set of laboratory values regarding hepatitis patients. I will construct a model to predict whether a visitor gets hepatitis based on the laboratory values observed.

There will be two approaches to be tried in this project. The first is to apply logistic regressions to create a multiclass classifier. The second is to use naive Bayes classifier to predict the classes. In this project, both approaches will evaluate all the attributes/features of the dataset and generate a predictive model based on training data. Testing data will evaluate both models from two approaches, and the accuracy will be calculated for comparison.

Data

The dataset comes from UCI Machine Learning Repository. Here is an overview of the dataset after some filtering.

- Total size: 582 cases of laboratory values.
- Response: 4 categories. (Healthy; Hepatitis; Fibrosis*; Cirrhosis*)
- Predictor: 12 attributes. (Age; Sex; ALB; ALP; ALT; AST; BIL; CHE; CHOL; CREA; GGT; PROT)

The dataset includes laboratory values from 225 female visitors and 357 male visitors. The third and the fourth categories, "Fibrosis" and "Cirrhosis" are two severe stages of Hepatitis. Boxplots can show an overall distribution of attributes by categories.

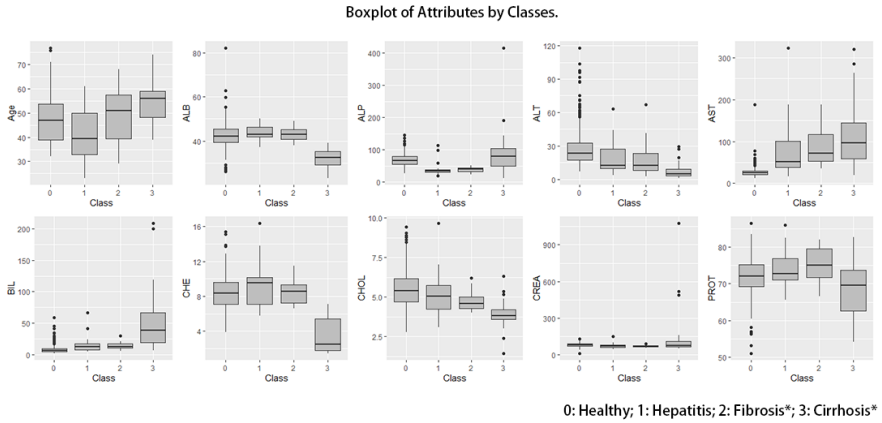

Figure 1: Distribution of attributes by categories

According to the boxplots, values of attributes vary from category to category. Most of the attributes show a significant difference between different categories. Therefore, all the attributes will be used in the model to predict the response.

Model

Multiclass Logistic Regression

Usually, logistic regression is used to predict a binary response. In my project, as the dataset includes more than two categories as a response, a simple logistic regression can not satisfy the predictive target. Therefore, multiclass logistic regression is the proper method to deal with the data. The basic regression function for data input as $\{(X_i, y_i)\}$ where $y_i \in \{1, 2, \ldots, k\}$ is.

$$h_\theta(x_i) = \begin{bmatrix} p(y_i = 1 \mid x_i; \theta) \\ p(y_i = 2 \mid x_i; \theta) \\ \vdots \\ p(y_i = k \mid x_i; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ \vdots \\ e^{\theta_k^T x_i} \end{bmatrix} \tag{1}$$

In the function, θ_i is the parameters of the model. The $\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x_i}}$ is to normalize the probabilities and regulize the sum as 1. Thus, the probability that the input data X_i belongs to category/class j is

$$p(y_i = j \mid x_i; \theta) = \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}} \tag{2}$$

The matrix of parameters in the regression can be denoted as

$$\theta = \begin{bmatrix} \theta_1^T \\ \theta_2^T \\ \vdots \\ \theta_k^T \end{bmatrix} \tag{3}$$

We define the likelihood function as

$$L(\theta) = -\frac{1}{m} \left[\sum_{i=1}^{m} \sum_{j=1}^{k} 1\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{l=1}^{k} e^{\theta_l^T x_i}} \right] \tag{4}$$

By using the gradient method, we optimize the log-likelihood by θ and get the estimation of parameters. The generated model with estimated parameters will be used to predict the response when the newly observed data passes in.

In this project, I will utilize the "sklearn", a popular Python package for machine learning, to realize the multiclass logistic regression. The "sklearn"

package provides a function called "linear_model.LogisticRegression" for general logistic regression. It supports the multiclass logistic regression using "one-vs-rest" approach in default. In classification, OvR approach compares one specific class with others and consider all other classes as one group. We are also able to choose the other approach, "multinomial" ("many-ve-many"), for classification. Such MvM approach compares many classes versus the other many ones. It does much more comparisons than OvR does and select a best choice for the classification.

Naive Bayes Classifier

In the naive Bayes classifier, the attributes are assumed to be independent of each other. Each attribute is assumed to follow the normal distribution. For jth attribute in ith class, we have

$$att_{ij} \sim N(\mu_{ij}, \sigma_{ij}) \tag{5}$$

where μ_{ij} and σ_{ij} are the sample mean and sample deviation calculated from the specific attribute of a particular class from the training data.

Then, for a specific class, we have a unique collection of normal distributions from its attributes. This collection represents the characteristic of a class and distinguishes it from other classes.

After collecting all the characteristics of classes, the model is ready to recognize the newly observed data. Each case of observed data will be put into the collections of normal distributions from attributes in classes. By the probability density function of normal distribution

$$p_{ij} = \frac{1}{\sqrt{2\pi} * s} * \exp\left(\frac{(att_j - \mu_{ij})^2}{2 * \sigma_{ij}^2}\right) \tag{6}$$

The probability that the jth attribute in a single row data follows the normal distribution of corresponding attribute in ith class is calculated. The probability that such a single row data belong to ith class is the joint probability of all attributes.

$$P_i = \prod_{j=1}^{n} p_{ij} \tag{7}$$

The naive Bayes classifier selects the class with maximum probability as the label of the single row data. In other words, the joint probabilities determine the predicted outcome of input data.

Train and Test

The total dataset has 582 rows, and we need to split it into the training set and testing set. The proportion between the training set and testing set is designed as $2 : 1$. Each set collects the data rows that randomly selected from each

category in the original dataset. Therefore we can assume that the training set and the testing are similarly representatives for the population.

With the same method of data splitting, models from two approaches are trained and tested. Specifically, we have three models in total. For the regression approach, there are two models, OvR model and MvM model. The model from naive Bayes classifier does the same job as well.

There are several aspects to compare these three models. The most important one is to check the predictive accuracy. Accuracy is calculated as

$$Accuracy = \frac{\#\ of\ successful\ predictions}{\#\ of\ observations} \tag{8}$$

The other one is to check the processing time needed by each model. All procedures counted includes "data importing", "data splitting", "model training", "model testing" and "accuracy reporting". Seconds consumed by each model will be recorded by "time" package in Python.

Results

The data of accuracies and time consuming is recorded in the following table.

Model	Accuracy (%)	Time(s)
OvR Multiclass Logistic Regression	95.36	0.025
MvM Multiclass Logistic Regression	96.39	0.192
Naive Bayes Classifier	93.81	0.012

Among the three models, MvM multiclass logistic regression shows the best accuracy than the others. However, it takes significantly more time in computing. The naive Bayes classifier runs much faster than the others, though it scores worst accuracy.

Conclusion and discussion

Generally, all three models can handle the job of classification for hepatitis patients. The difference between their accuracies and running times have several reasons.

The naive Bayes classifier does not require any gradient computing and directly calculate the probabilities. It works more simply than regressions. However, it loses some accuracy due to the assumption of independence of attributes. This may be significant towards the final predictive model, and regression models count the correlations.

The regression models can provide more precise predictions but are more sensitive to the training data. Outliers will result in difficulties in regression models. When the sample size is small, such an effect from outliers may become more significant. In this project, the size of the training data is quite large as 388 and thus has efficiently reduced the effect of outliers.

Among the regression models, the OvR model takes less time than the MvM one does. MvM models have to deal with more comparisons between different combinations of classes. Strictly, MvM models trade the time efficiency for the prediction's performance.

In conclusion, three models have their specialties depending on the data and the targeted performance. With a small size of training data, naive Bayes classifier will be the absolute choice. When training dataset is large, regression models are better approaches towards the prediction. Furthermore, when the number of classes becomes larger, the MvM model will achieve higher accuracy if we have strong computing power. Besides, OvR model will be a good balance between efficiency and performance.

Bibliography

[1] Dua, D and Graff, C. (2019). UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`. Irvine, CA: University of California, School of Information and Computer Science.

Appendix A: Code

GitHub Page: STAT-538A-Project