PS2: Classification of Medical Treatment

Yi Lin

February 24, 2021

1 Introduction

The motivation for this project exploration is driven by the desire to improve the treatment prediction accuracy and to enhance awareness and knowledge on health. For clinicians and medical researchers, machine learning techniques are drawing substantial and considerable interest from a major hospital. In this project, the purpose is to address the need for capacity development in the medial areas alongside a practical guide to develop and evaluate a predictive logistic regression model for classifying medical treatment [1].

2 Data Cleaning

2.1 Pruning Outliers

1. Age. Outliers may cause much difference in both standard deviation values and mean, which directly/indirectly impact the model accuracy. A box plot of the dataset crossing numerical variable 'Age' showed that treatment might be varied by age. To remove such outliers of age, I pruned the dataset using IQR, leaving 10% values of both sides of the distribution. In other words, I remove patients who age are older than 70 and younger than 50.

2. Blood_Pressure. Since it is not possible and reasonable for people with negative values in blood_pressure, I chose to drop values of blood_pressure less than 0.

2.2 Pruning Missing Values

Notice that only 'family_history' has missing values. However, since missing values of 'family_history' have 2095 rows out of 6031 entire rows, it implies that directly dropping around 1/3 of the dataset is not reasonable. Therefore, I chose to fill missing values according to frequency. To impute missing values, I first checked *value_counts* to find the distribution, then got the rows with missing values, and finally filled those rows with randomly selected values (*np.random.choice*) based on the distribution.

3 Data Pre-processing

In the dataset, several categorical features cannot be applied to machine learning algorithms. Those categorical data should be converted as numbers. Since there are no ordinal variables, I don't need to use encoding methods on ordinal variables.

3.1 One-Hot Vectorization

To converted these categorical data (gender, blood_test, family_history, GeneC) into One-Hot vectors, I chose One-Hot Encoding to map each category to a vector that contains 1 and 0, denoting the presence of the attributes. After converted these categories, I broke the One-Hot encoded vectors as separated attributes. For example, 'gender' with [female, male] and one-hot vector is [1,0]. I broke 'gender' attribute into 2 additional columns: [female] and [male]. And therefore, the only [female] column would be 1, and [male] column would be 0. Finally, I created a new data frame as 'df_new' by combining numerical variables (treatment, age, blood_pressure, TestA, TestB, GeneD, GeneE, GeneF) and one-hot encoded vectors as separated features ('female', 'male', 'bt_negative', 'bt_positive', 'fh_False', 'fh_True', 'C_not active', 'C_active'). I also reset the data frame index as well.

4 Exploratory Data Analysis

Several visualization tools like displot, countplot, catplot, histogram in seaborn were be used to analyze and investigate all features' general trends concerning the target feature (treatment). I notice that medical treatment has an efficient impact on the age around 57 and 60. There is a clear turning point of treatment from 1 to 0 between age 63 and 64. This may imply that medical treatment may be recommended for patients with age less than 63 but not with age greater than 63. Also, the percentage of males and females in the dataset is 40% and 60%, respectively. From the statistics of gender-based treatment classification, an interesting found that medical treatment significantly affects females rather than males. Female with treatment is 2746 while male with treatment is 599.

Furthermore, to analyze the impact of statistical significance of special modifications (Test A-B, Gene C-F), I applied OLS to see the P-value of each attribute. Assume alpha is 0.05. Notice that TestA = 0.771 and GeneD = 0.260 are greater than 0.05, which implies statistically insignificant. The rest of the attributes, such as Test B, GeneE, GeneF, C_not active, C-active are smaller than 0.05, which are statistically significant.

To check for multicollinearity among special modifications, I applied the VIF score. Based on the Rule of Thumb, I found that VIF scores of all these are less than 5, which demonstrates that it is low multicollinearity which means lower correlation in TestA, TestB, GeneD, GeneE, GeneF, C_not active, C_active.

Moreover, I applied the Plot Correlation Matrix to visualize and explore the correlations among the features. I found that the features like gender(female, male), blood_pressure, age, TestA have relatively somewhat strong positive and negative correlation with treatment. Besides the correlation with treatment, I interestedly found that the correlation between TestA and age has a super strong positive relationship (corr = 0.937). This may imply that special modification TestA has an impact related to age and treatment.

To double-checked my work, I finally applied backward elimination as feature selection. I notice that the P-values of TestA, GeneD are greater than 0.05. So I decided to drop TestA and Gene D variables and applied the rest of the dataset into the model.

5 Model Training and Evalutaion

I utilized several simple methods with 60-20-20 on testing, training, and validation. I first split the training & testing set into 80-20 and then split the training & validation set into 60-20. Since it is a logistic regression model, performing hyper-parameter tuning is necessary. So, I tuned the regularization with three split sets. I applied 60% of the dataset as a training dataset into the fitting and put a validation set to calculate the predictive y-values.

In the sklearn's logistic regression, I conducted several parameters to adjust the regularization or not. The parameter like "penalty" allows us to choose a regularizer. The parameter like 'C' allows us to control the amount of regularization we like to have. The parameter like 'class_weight' allows us to adjust the weight to class frequencies. The parameter like 'solver' allows us to choose the algorithm to optimize the multi-class problems. In my model, I imputed penalty with 'None,' 'l1', and 'l2'; 'C' with 1 as default; 'class_weight' with 'balance' to automatically adjust weights inversely proportional to class frequencies in the input data; and 'solver' with 'liblinear' for our small dataset. Notice that when the penalty is 'None,' there is no regularization on logistic regression. Then I used predictive y values with the validation set to tune the hyper-parameter and to determine the best model by estimating the model's accuracy, precision, f1 score, false positive, false negative, and AUC score. And finally, I use a testing set to calculate the chosen model accuracy. There are three models:

- Model 1: Penalty is None
- Model 2: Penalty =l1, C=1, class_weight ='balanced', solver='liblinear'
- Model 3: Penalty =l2, C=1, class_weight ='balanced', solver='liblinear'

Table 1: Logisitic Regression Model Results (Regularization)

Regularization	Accuracy	Precision	F1 score	AUC	FP	FN
Model 1	0.74875	0.74659	0.78341	0.74012	0.614	0.393
Model 2	0.74792	0.75244	0.77971	0.74092	0.582	0.404
Model 3	0.74626	0.75104	0.77826	0.73924	0.582	0.403

It is an accuracy paradox to choose an optimal model since the model's accuracy is not the single criteria/metric for results quality. I also look through the results of F1, AUC, Precision, FP, FN as well. From the above table, I notice that the accuracy and F1 score of model 1 are higher than model 2 and 3. However, the precision of model 2 is somewhat higher than the other two models. The AUC score of model 2 is highest score than that of model 1 and 3. The false-positive rate of model 2 and 3 is much better than model 1 (0.614>0.582=0.582), and the false-negative rate of model 2 is also better than model 1 and 3 (0.404>0.403>0.393). To compare and contrast those metrics altogether, I would recommend the Logistic Regression model with regularization (penalty = l1) as the model to test its performance. By apply the testing set, I finally calculate that the accuracy of the optimal model is 74.9%. By looking through the coefficient of each feature with respect to the target variable (treatment), I notice that gender (female and male) has a 66% impact on the treatment; in other words, there is 66% that treatment is recommended changes with gender.

6 Conclusion

In this project, I conduct a study on classification model performance. To propose a prediction model for classifying whether a certain treatment is recommended for the patient or not. The dataset came from the DSCI552 class, and the data preparation was processed by python. The final cleaned-dataset contains 6031 rows and 14 features. I tested data by applying it into logistic regression with testing-training-validation three sets. To perform hyper-parameter tuning, I added the regularization to the logistic regression. To compare and contrast the model with different tuning regularization methods, I found that regularization can be used to avoid overfitting and slightly help to improve the model performance. Therefore, I would recommend a logistic regression model with regularization (penalty=l1) to build up the treatment prediction model. The accuracy of logistic regression with regularization is 74.9%, which shows that its performance has 74.9% of data fit the classification model.

References

[1] Sidey-Gibbons, J., Sidey-Gibbons, C. Machine learning in medicine: a practical introduction. *BMC Med Res Methodol* 19, 64 (2019). https://doi.org/10.1186/s12874-019-0681-4