CARDIOVASCULAR DISEASE - PREVENTION AND HIGHLY RELATED FEATURES

I-Ling Yeh
School of Coumputing and Information
University of Pittsburgh
Pittsburgh, PA 15213
`ILY5@pitt.edu`

Shuo-Yuan Chang
School of Coumputing and Information
University of Pittsburgh
Pittsburgh, PA 15213
`SHC151@pitt.edu`

Yuke Liu
School of Coumputing and Information
University of Pittsburgh
Pittsburgh, PA 15213
`YUL233@pitt.edu`

April 20, 2020

ABSTRACT

The problem that we are trying to solve is figuring out which situation may easier for people to catch Cardiovascular Diseases. We will use logistic regression, Naive Bayes, K Nearest-Neighbors, Ensemble method, and random forest to train the model. From the models we analyzed, the predicted result becomes a reminder for us to reduce the rate of catching Cardiovascular Diseases.

1 Introduction

According to the CDC, approximately 647,000 have suffered from Cardiovascular Disease in America each year, and the death rate is up to 25% of each patient. This is a common disease among middle-aged people and elders which is closely related to our family members and maybe ourselves health situation in the future. To reduce the rate of catching Cardiovascular Diseases, we analyze the real-world patients' information to find the path for predicting the probability of having the disease.

We are trying to solve a problem is figuring out which situation may easier for people to catch Cardiovascular Diseases. More specifically, in the detailed physical index, which is closely related to Cardiovascular Disease. And in which living habit is most related to causing or reducing the rate of Cardiovascular Diseases. Also, we think gender and age also matters.

2 Data Pre-process

We had investigated the dataset with missing data, data types, the situations of collinearity, and non-normality. Depended on the data exploring, we had processed these problems to avoid affect the accuracy.

2.1 Data Understanding

The dataset that we decided to use is founded on Kaggle. This dataset is the Cardiovascular Disease dataset built up by Ulianova (2019)[1]. This dataset consists of 70,000 not-well-cleaned patient records. The columns in the dataset represent the patients' general information, detailed physical index, living habits, and whether they are Cardiovascular Disease patients. The general information contains age, weight, height, and gender. The physical index contains Systolic

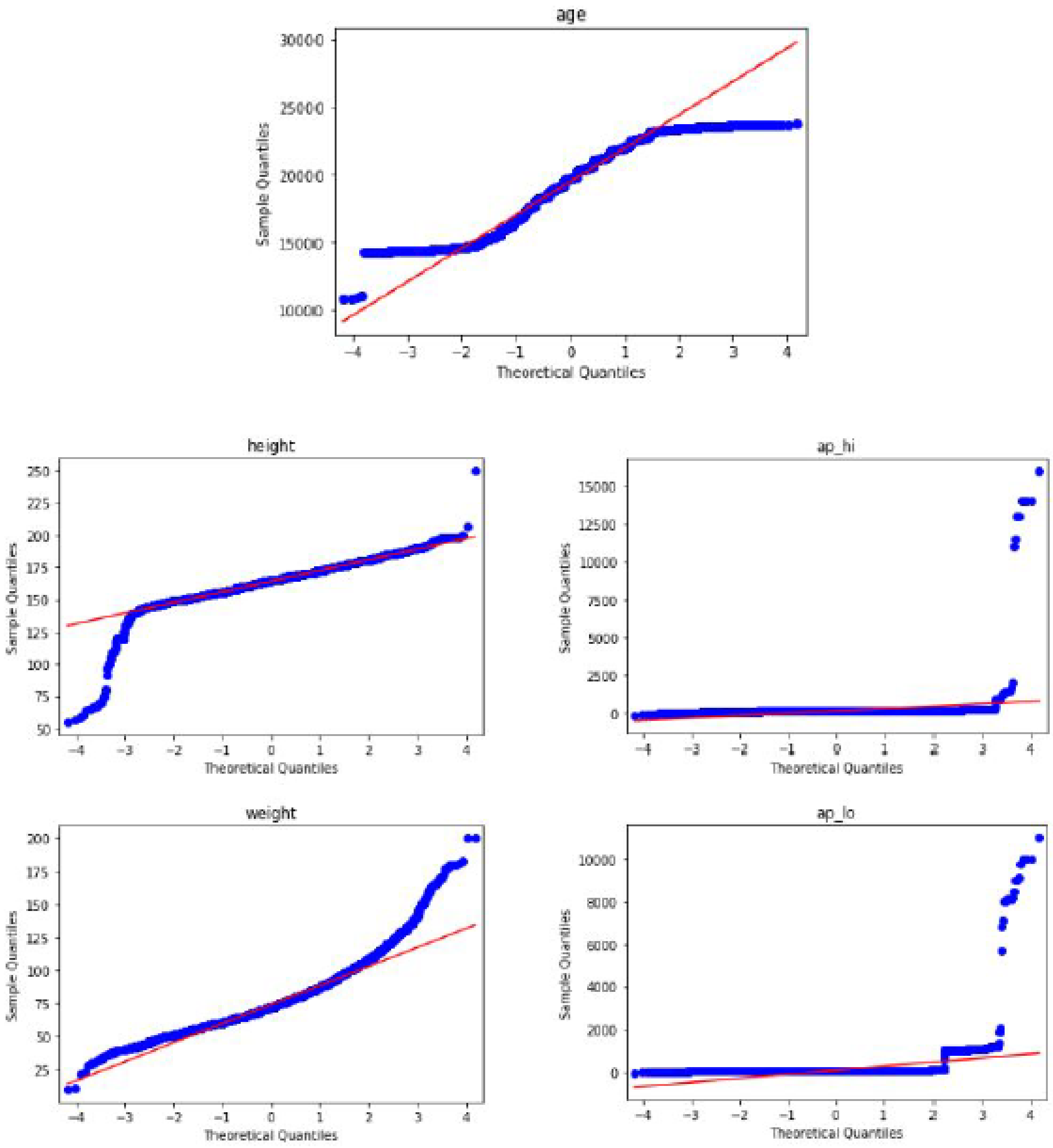

Figure 1: QQ Plot: The rad line is the P-value of expected; The blue is the P-value of observed

blood pressure, Diastolic blood pressure, Cholesterol, and Glucose. The living habits columns contain the data of daily smoking, alcohol intake, and physical activities. The last one is whether they are a Cardiovascular Disease patient.

To make sure features to be normalized and non-collinearity, we want to check the normalization and check the feature collinearity, which will affect the accuracy of the model. Five variables in this dataset are continuous: age, height, weight, ap_hi, ap_lo. Figure 1 shows these five variables' QQ plot. According to the figure, the P-value of observed is well-fitting the diagonal that is the P-value of expected. When the P-value of observed fit well the diagonal means the variable is normalization.

We summarize the strength of the linear relationship between two data samples through the Pearson Correlation. Figure 2 displays the Pearson Correlation of features. As can be seen, the relation between height and gender is the highest positive relationship, but it is not above 0.5 which indicates not a notable correlation. According to the figure, we cannot find the collinearity situation.

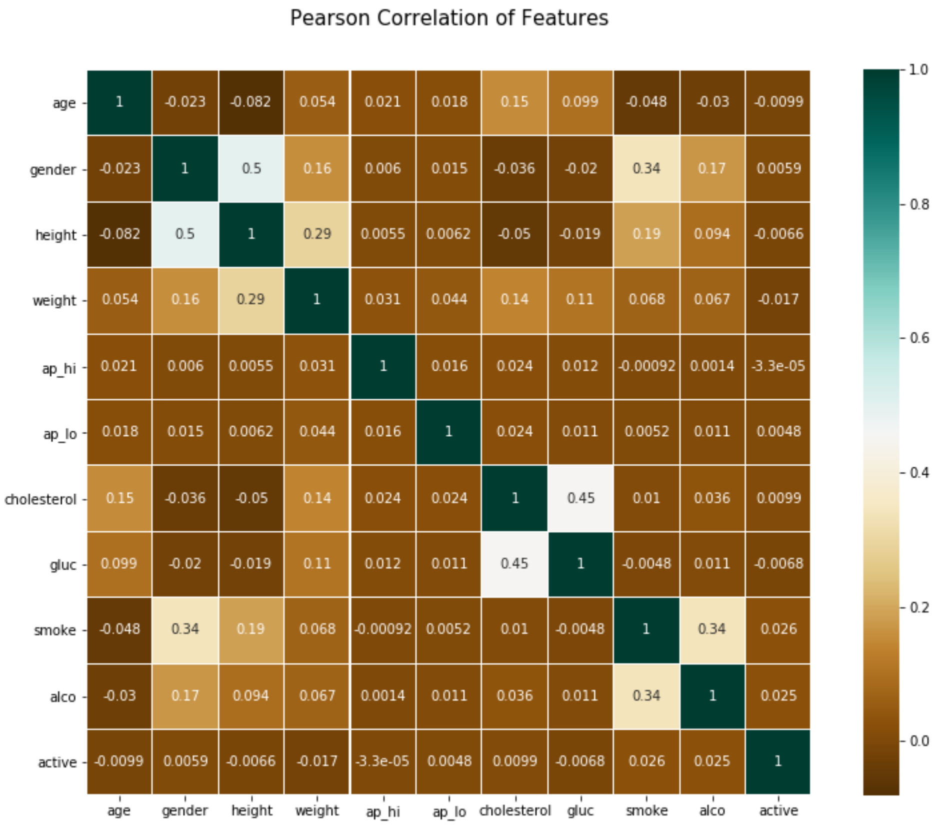

Figure 2: Pearson Correlation

2.2 Data Preparation

There are several variables which are categorical type, so we scale the feature to approach standardization. After feature scaling, we split the dataset into the training dataset and testing dataset. Before we training, we make sure that several people infected by cardiovascular and people don't are equal in training data and testing data. For the next step, we plan to add extra features to increase the model complexity to improve the accuracy of the model.

3 Modeling

Our target is a binary outcome and supervised learning, so different technology we can use to predict our outcome. The methods that we will use are Logistic Regression, Naive Bayes, K Nearest-Neighbors, Ensemble method, and Random Forest.

First, we choose Logistic Regression, because they are more simple that can be our evaluation pipeline. We train the Logistic Regression with a penalty that avoids overfitting. In the Logistic Regression, we get the 0.78 in training data and get the 0.78 in testing data.

Then, we choose K Nearest-Neighbors, because it is a simple model to classify into two groups. We train KNN with GridSearchCV to find the optimal number of neighbors, which is 37 neighbors. In KNN, we get the 0.75 in the training data and get 0.72 in testing data.

We continued the modeling with Naive Bayes. Different from other models, in this part we slightly changed our dataset. For all the values in the selected variables such as height, weight, age, Systolic blood pressure, Diastolic blood pressure, Cholesterol, and Glucose, we had divided these values into new columns as the binary classifiers. We get 0.82 in training data and 0.78 in testing data.

We can see Logistic Regression, Naive Bayes, and KNN all have lower accuracy. We want to increase the accuracy, so we choose the Ensemble method, which can produce a classifier with higher accuracy. We train XGBoost and LightGBM with GridSearchCV that tune hyperparameters. In XGBoost, we get 0.82 in training data and get 0.80 in testing data. In LightGBM, we get 0.83 in training data and get 0.80 in testing data.

In the above Ensemble methods, we can see there are a little bit overfitting in XGBoost and LightGBM. Eventually, we choose Random Forests, because it would be helpful to reduce variance, which reduces overfitting. We train Random Forests with GridSearchCV that find the optimal hyperparameters. In Random Forest, we get 0.81 in training data and get 0.80 in testing data.

4 Evaluation and Result

We have a target, which is a binary outcome, so this training is supervised learning. We tried to train many models. Figure 3 shows every model's AUC. Eventually, we train the Random Forest as our model, because it is the best accuracy and well fit between the training dataset and testing dataset. In the Random Forest with GridSearchCV, we get the 0.81 in training data and get 0.80 in testing data.

To explore what fact causes people to get Cardiovascular Disease, we use SHAP to find the variance important in the Random Forest model that we trained. Figure 4 shows the summary plot. As you can see, the ap_hi, which is Systolic blood pressure, is the most important variable to impact the prediction. For everyone, we need to focus on blood pressure to prevent getting Cardiovascular Disease. For doctors, this information can help them to identify the disease. For the patient of Cardiovascular Disease, they can control their blood pressure to avoid further deterioration.

5 Conclusion

Cardiovascular Disease leads to a higher death rate compared to other diseases. We predict if a patient has cardiovascular disease, and to explore what factors cause the disease. Our dataset, cardiovascular disease data, is from Kaggle, and it has 70000 observations and 13 variables. We check normalization and correlation and cleaning data to the training model. In the end, we got the Random Forest model as our best model. In our best model, we found the blood pressure is an important factor to affect our prediction. In the future, we can add some extra variables to increase model complexity and improve our model accuracy.

References

[1] Svetlana Ulianova. Cardiovascular Disease data (Version 1). In *https://www.kaggle.com/sulianova/cardiovascular-disease-dataset#cardio_train.csv*, Web. 2019.

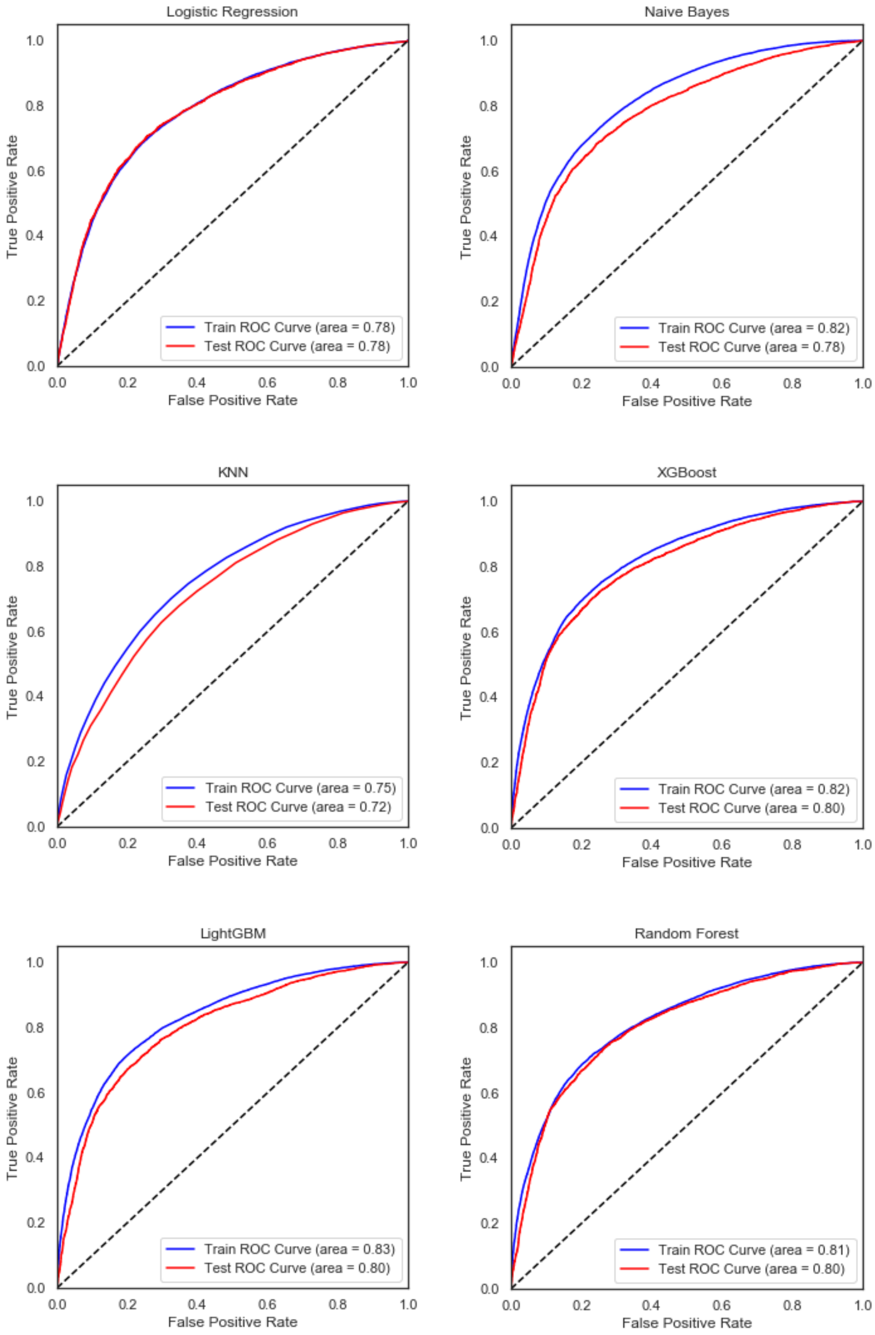
Logistic Regression
Train ROC Curve (area = 0.78)
Test ROC Curve (area = 0.78)
Naive Bayes
Train ROC Curve (area = 0.82)
Test ROC Curve (area = 0.78)
KNN
Train ROC Curve (area = 0.75)
Test ROC Curve (area = 0.72)
XGBoost
Train ROC Curve (area = 0.82)
Test ROC Curve (area = 0.80)
LightGBM
Train ROC Curve (area = 0.83)
Test ROC Curve (area = 0.80)
Random Forest
Train ROC Curve (area = 0.81)
Test ROC Curve (area = 0.80)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Figure 3: AUC

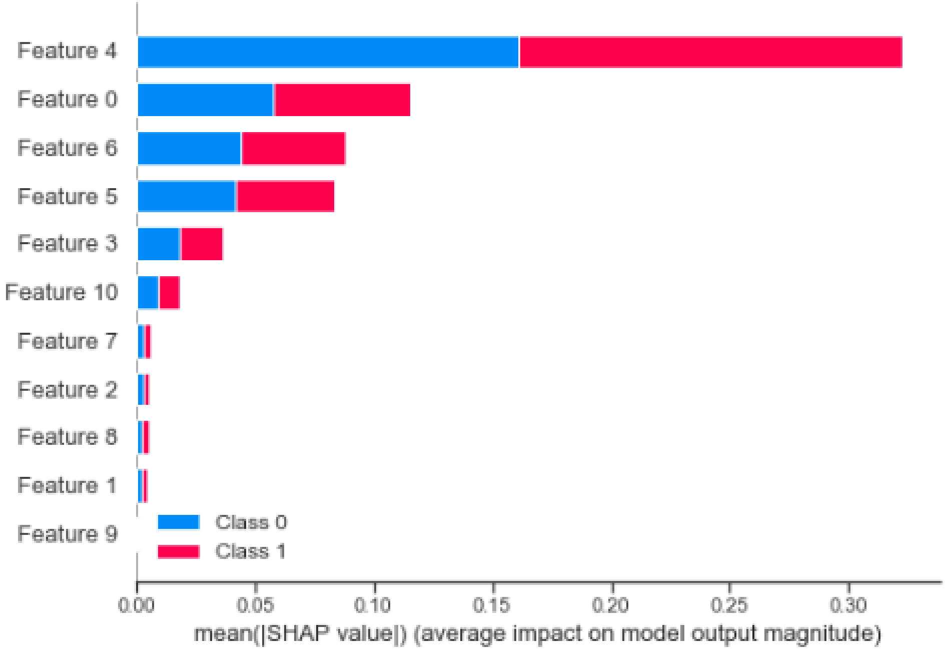

Figure 4: Summary Plot