Classification models for SPECT Heart data

Mingyi Wei, Jingyu Xu

Southern University of Science and Technology of China, Shenzhen

Abstract

SPECT is an instrument for functional and metabolic imaging in vivo with the help of a single photon nuclide labeled drug. With advances in radiation medicine, doctors can use certain features of SPECT images to determine whether a person's heart is healthy or not. Based on those features, we try to separate abnormal patients from healthy patients. Dimension reduction and feature selection play essential roles in these mass-spectrometric data with high-dimensional features. Classification models, based on feature selection techniques, are applied to the SPECT heart dataset. Precision accuracy are considered finally to compare different methods, and ensemble method is used to improve the model.

Keywords: heart dataset, ensemble, classification, feature selection.

1 Introduction

SPECT imaging is used as a diagnostic tool for myocardial perfusion. During cardiac SPECT study patient is injected with radioactive agent, (Tl-201), that during decay emits single photon of 150 [keV] energy. During the study the detectors are located around the patient body and rotated. Using high-level reconstruction algorithms one 3D image, from a set of 2D planar views at different angles, is created.

Cardiac SPECT images represent LV myocardial muscle perfusion that is proportional to distribution of radioactive counts within the myocardium. Typical 2D-image resolution [1] is 64x64, all the images are black and white, 8 bits-per-pixel with 256 shades of gray. Brighter areas on the image correspond to well perfused areas of myocardium. When part of myocardium is not visible an ischemia is suspected.

The UCI SPECT Heart Data Set, which describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images [2]. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature pattern was created for each patient.

We selected training sets and test sets from the UCI SPECT Heart Data Set, with 80 samples in the training Set and 187 samples in the test Set, each containing 45 variables. The first variable y is a 0-1 binary variable, indicating an unhealthy or healthy heart. The remaining 44 variables are features extracted from SPECT images, both have integer values from the 0 to 100.

the histogram of variable y is as follows:

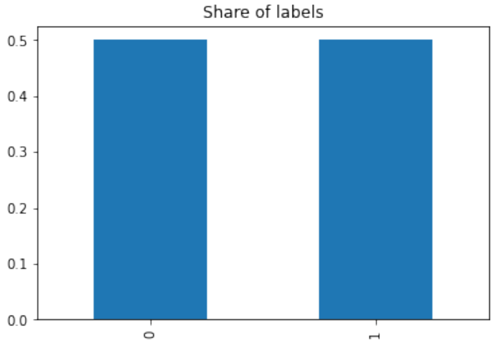


Figure 1: share of labels

2 Proposed models

2.1 Feature selection methods

2.1.1 LASSO regression

We consider a classical logistic regression model first,

$$logit(p) = \log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p, \tag{1}$$

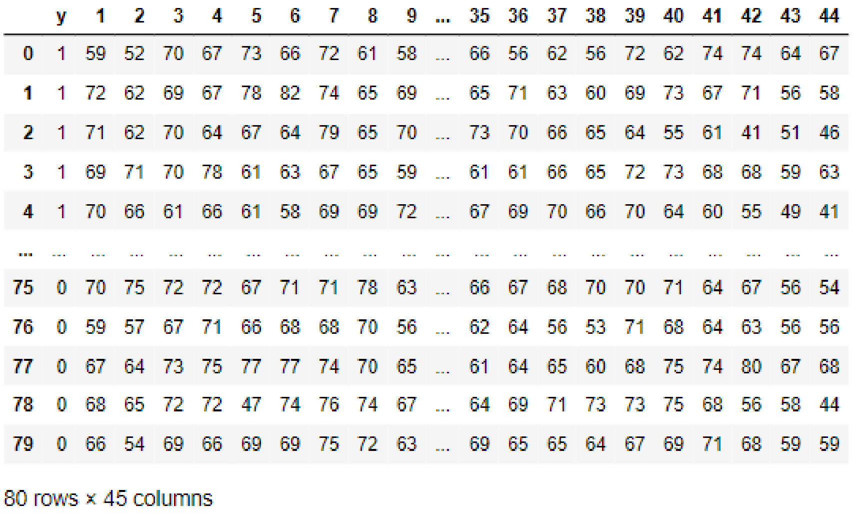
where y is the variable of 0 or 1, 1 represents person who gets cancer and 0 means he/she is healthy.

There are many methods to select essential variables. However, it is difficult to invert a matrix $X^\top X$ as $n < p$. So we use LASSO [3] to penalize the explanatory variable coefficients by adding constraints to the loss function, which screens variables greatly. The LASSO estimation can be expressed as

$$\beta^{\text{lasso}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{2}$$

By equation (2), the basic LASSO model estimation is built.

Next we use the lasso regression to train the model, and get the 7 essential variables, partial data is shown in the Figure 2.



	y	1	2	3	4	5	6	7	8	9	...	35	36	37	38	39	40	41	42	43	44
0	1	59	52	70	67	73	66	72	61	58	...	66	56	62	56	72	62	74	74	64	67
1	1	72	62	69	67	78	82	74	65	69	...	65	71	63	60	69	73	67	71	56	58
2	1	71	62	70	64	67	64	79	65	70	...	73	70	66	65	64	55	61	41	51	46
3	1	69	71	70	78	61	63	67	65	59	...	61	61	66	65	72	73	68	68	59	63
4	1	70	66	61	66	61	58	69	69	72	...	67	69	70	66	70	64	60	55	49	41
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
75	0	70	75	72	72	67	71	71	78	63	...	66	67	68	70	70	71	64	67	56	54
76	0	59	57	67	71	66	68	68	70	56	...	62	64	56	53	71	68	64	63	56	56
77	0	67	64	73	75	77	77	74	70	65	...	61	64	65	60	68	75	74	80	67	68
78	0	68	65	72	72	47	74	76	74	67	...	64	69	71	73	73	75	68	56	58	44
79	0	66	54	69	66	69	69	75	72	63	...	69	65	65	64	67	69	71	68	59	59

80 rows × 45 columns

Figure 2: partial data with selected features

2.1.2 SCAD (smoothly clipped absolute deviation)

The smoothly clipped absolute deviation (SCAD) penalty, introduced by Fan and Li (2001), was designed to encourage sparse solutions to the least squares problem, while also allowing for large values of β. The SCAD penalty is part of a larger family known as “folded concave penalties” . The SCAD penalty looks like this:

$$\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X\beta})'(\boldsymbol{y}-\boldsymbol{X\beta}) + n\sum_{j=1}^{d} p_\lambda(|\beta_j|) \tag{3}$$

$$\text{Where } p_\lambda(|\beta_j|) = \lambda^2 - (|\beta_j - \lambda|)^2 I(|\theta| < \lambda) \tag{4}$$

When $\beta_{j0} \neq 0$, The penalty function can be approximated with a second-order Taylor expansion.

$$p_\lambda \approx p_\lambda(|\beta_{j0}|) + \frac{1}{2}\{p_\lambda{}'(|\beta_{j0}|)/|\beta_{j0}|\ \}(\beta_j{}^2 - \beta_{j0}{}^2) \tag{5}$$

Thus there is the following iterative formula:

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=i}^{n} y_i x_{ij} - \sum_{i=1}^{n} x_{ij}\frac{exp[\alpha+\beta' x_i]}{1+exp[\alpha+\beta' x_i]} + n \times p_\lambda{}'(|\beta_{j0}|)/|\beta_{j0}|\beta_j \tag{6}$$

$$\frac{\partial^2 l}{\partial {\beta_j}^2} = -\sum_{i=1}^{n}\left[\frac{(x_{ij})^2 exp[\alpha+\beta' x_i]}{[1+exp(\alpha+\beta' x_i)]^2}\right] + n \times p_\lambda{}'(|\beta_{j0}|)/|\beta_{j0}| \tag{7}$$

$$\therefore {\beta_j}^{(m)} = {\beta_j}^{(m-1)} - \left[\frac{\partial^2 l(\beta)}{\partial {\beta_j}^2}\right]^{-1}\frac{\partial l(\beta)}{\partial \beta_j} \tag{8}$$

This implies that the approximate SCAD solution is :

$$\hat{\beta}_{\text{SCAD}} = \left(X^\top X + \left(\frac{p'_\lambda(|\beta_0|)}{2|\beta_0|}\right) I\right)^{-1} X^\top Y. \tag{9}$$

2.1.3 MCP (maximum convex penalty)

The penalty of MPC is:

$$for \quad \gamma \geq 1,\ p_\gamma(x;\lambda) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma} & ,if|x| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2 & ,if|x| \geq \gamma\lambda \end{cases} \tag{10}$$

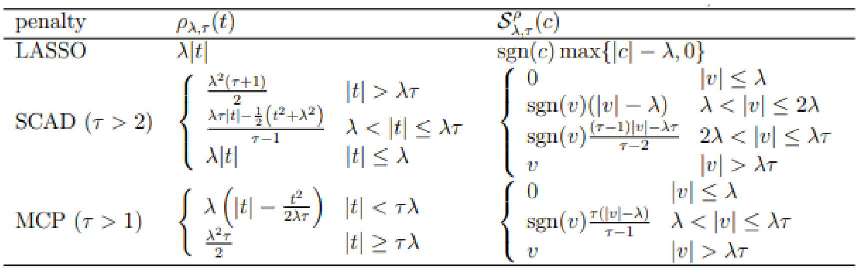
the derivative is:

$$p'_\gamma(x;\lambda) = \begin{cases} (\lambda - \frac{|x|}{\gamma})sign(x) & ,if|x| \leq \gamma\lambda \\ 0 & ,if|x| \geq \gamma\lambda \end{cases} \tag{11}$$

the MPC is to minimize:

$$\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X\beta})'(\boldsymbol{y} - \boldsymbol{X\beta}) + n\sum_{j=1}^{d} p_\lambda(|\beta_j|) \tag{12}$$

where the penalty function $p_\lambda(|\beta_j|)$, is as above.



penalty	$\rho_{\lambda,\tau}(t)$	$\mathcal{S}^{\rho}_{\lambda,\tau}(c)$
LASSO	$\lambda\|t\|$	$\text{sgn}(c)\max\{\|c\| - \lambda, 0\}$
SCAD ($\tau > 2$)	$\begin{cases} \frac{\lambda^2(\tau+1)}{2} & \|t\| > \lambda\tau \\ \frac{\lambda\tau\|t\| - \frac{1}{2}(t^2+\lambda^2)}{\tau-1} & \lambda < \|t\| \leq \lambda\tau \\ \lambda\|t\| & \|t\| \leq \lambda \end{cases}$	$\begin{cases} 0 & \|v\| \leq \lambda \\ \text{sgn}(v)(\|v\| - \lambda) & \lambda < \|v\| \leq 2\lambda \\ \text{sgn}(v)\frac{(\tau-1)\|v\| - \lambda\tau}{\tau-2} & 2\lambda < \|v\| \leq \lambda\tau \\ v & \|v\| > \lambda\tau \end{cases}$
MCP ($\tau > 1$)	$\begin{cases} \lambda\left(\|t\| - \frac{t^2}{2\lambda\tau}\right) & \|t\| < \tau\lambda \\ \frac{\lambda^2\tau}{2} & \|t\| \geq \tau\lambda \end{cases}$	$\begin{cases} 0 & \|v\| \leq \lambda \\ \text{sgn}(v)\frac{\tau(\|v\| - \lambda)}{\tau-1} & \lambda < \|v\| \leq \lambda\tau \\ v & \|v\| > \lambda\tau \end{cases}$

Figure 3: the penalty function

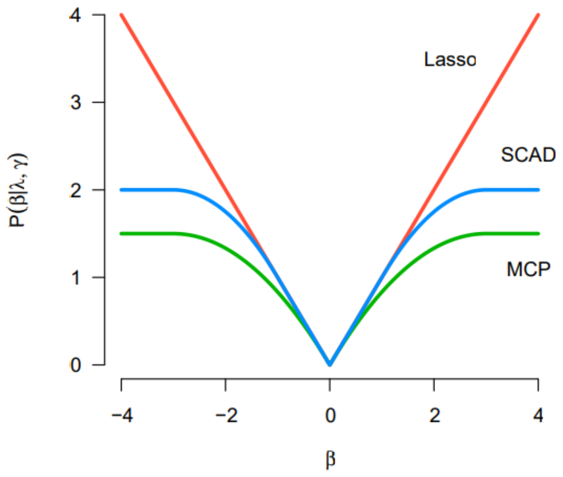


Figure 4: the penalty of the above three methods

The above two pictures are the penalty of three methods. After comparing the three methods on this dataset, we find that logistic based on lasso performs best, so we will use lasso to do variable selection in the third part of our report.

2.2 Decision Tree

The algorithm of decision tree learning [4] is usually to recursively select the optimal feature and segment the training data according to this feature. The selection of this feature mainly changes through the change of maximum depth. The specific steps are as follows:

(1) Construct the root node (i.e. the first layer in the decision tree), put all the training data into it, and divide the data set into subsets according to the principle of optimal classification through the optimal features.

(2) Assign the subset to the next level node.

(3) If the subset has not been classified correctly, select the new optimal feature and continue the segmentation until all the training data are classified, that is, a decision tree is generated.

2.3 Random Forest

Random forest [5] is an algorithm that integrates a variety of different and intersecting decision trees through integrated learning, and its accuracy is much higher than that of a single decision tree. The approximate process of random forest is as follows:

(1) Select n samples from the sample set so that the training samples of each tree have intersection.

(2) K features are randomly selected from all features. For the selected n samples, these features are used to establish a decision tree.

(3) Repeat the above two steps to generate all decision trees to form a random forest.

(4) After the decision of each tree, the new data is finally integrated to determine which class to allocate.

2.4 Logistic

Logistic regression uses the sigmoid function:

$$g(z) = \frac{1}{1+e^{-z}} \tag{13}$$

Mapping the input data to the [0,1] interval [6] can get a predicted value, and then mapping the value to the above function can complete the conversion from value to probability, that is, the classification task.

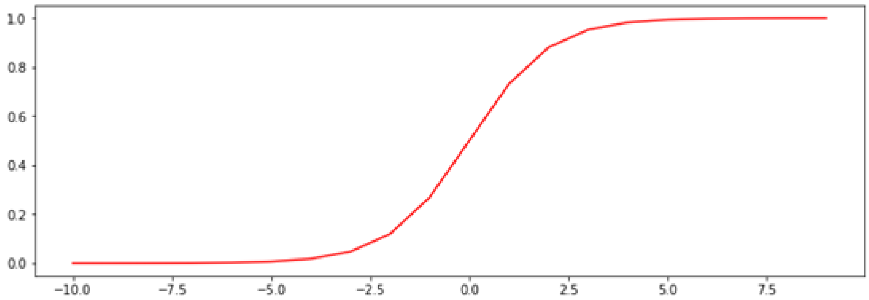


Figure 5: the sigmoid function

It is assumed that the dividing line between the two categories is:

$$\theta + \theta_1 x_1 + ... + \theta_n x_n = \sum_{i=1}^{n} \theta_i x_i = \theta^T x \tag{14}$$

Then the prediction function can be expressed by:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{15}$$

Thus, the classification task can be converted into a probability formula:

$$P(y = 1|x, \theta) = h_\theta(x) \tag{16}$$

$$P(y = 0|x, \theta) = 1 - h_\theta(x) \tag{17}$$

2.5 KNN

KNN algorithm is different from the above algorithm. It does not need training. When new samples are input, k nearest samples are directly found in the data. If most of these samples belong to a certain category, the sample is determined as this category. The specific steps are as follows:

(1) Calculate the distance between the tested sample and each data (generally use the Euclidean distance).

(2) Select the k points with the smallest distance (the K value needs to be specified, and the change of K value affects the classification results).

(3) Return the category with the highest frequency among the k points as the category of the test sample.

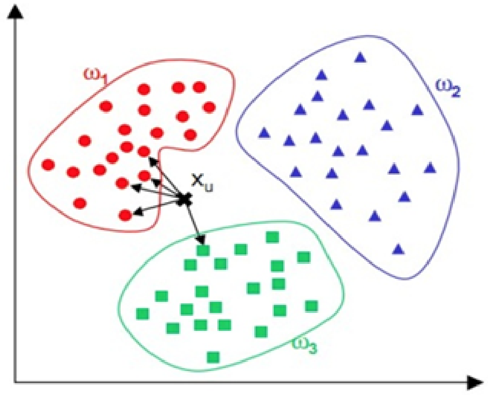


Figure 6: the KNN algorithm

It can be seen from the above figure that the principle of KNN algorithm is relatively simple, but it requires multiple distance calculation and classification judgment. The process is cumbersome and the accuracy is low, which will be reflected in the results.

2.6 SVM

The idea of support vector machine [7] is based on the linear division of low dimension. It mainly maps the points in low dimension space to high dimension space to make them linearly separable, and then re maps back to low dimension space to obtain the segmentation line, and then obtains the corresponding judgment rate.

The hyperplane can divide the samples into two categories. In order to improve the accuracy, the support vector machine method maximizes the distance of the support vector (Gap in the figure below), so as to obtain the optimal classification hyperplane.

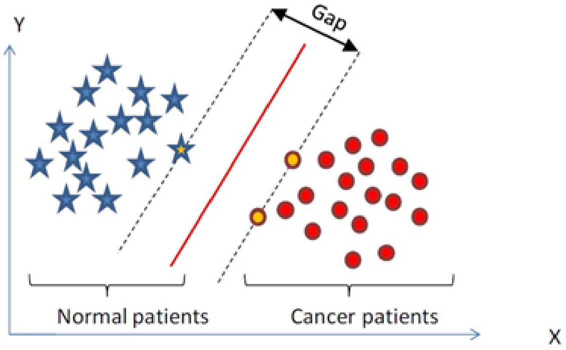


Figure 7: support vector machine

2.7 MLP

We also use an artificial neural network method called Mlp to classify the dataset. Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons.

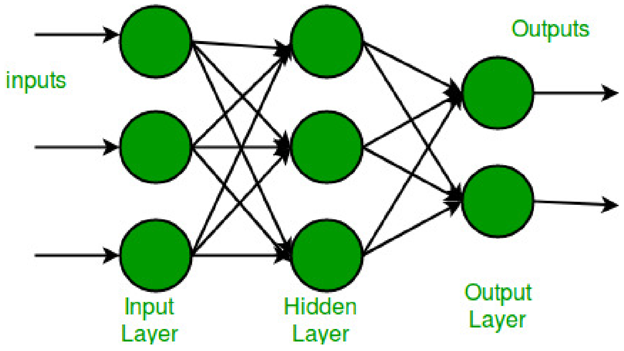


Figure 8: Multi-layer perception

In the multi-layer perceptron diagram above, we can see that there are three inputs and thus three input nodes and the hidden layer has three nodes. The output layer gives two outputs, therefore there are two output nodes. The nodes in the input layer take input and forward it for further process, in the diagram above the nodes in the input layer forwards their output to each of the three nodes in the hidden layer, and in the same way, the hidden layer processes the information and passes it to the output layer. The result will be showed in the next part.

3 Results

After dividing the training set and test set, we first choose seven models for training and prediction. They are: gbm, Naive bayes, Random forest, Logistic, KNN, neural network(MLP) and SVM.

3.1 Classify without dimension reduction

We first apply different models to the original data(without dimension reduction), The prediction effect of the seven models is shown in the Table 1 below:

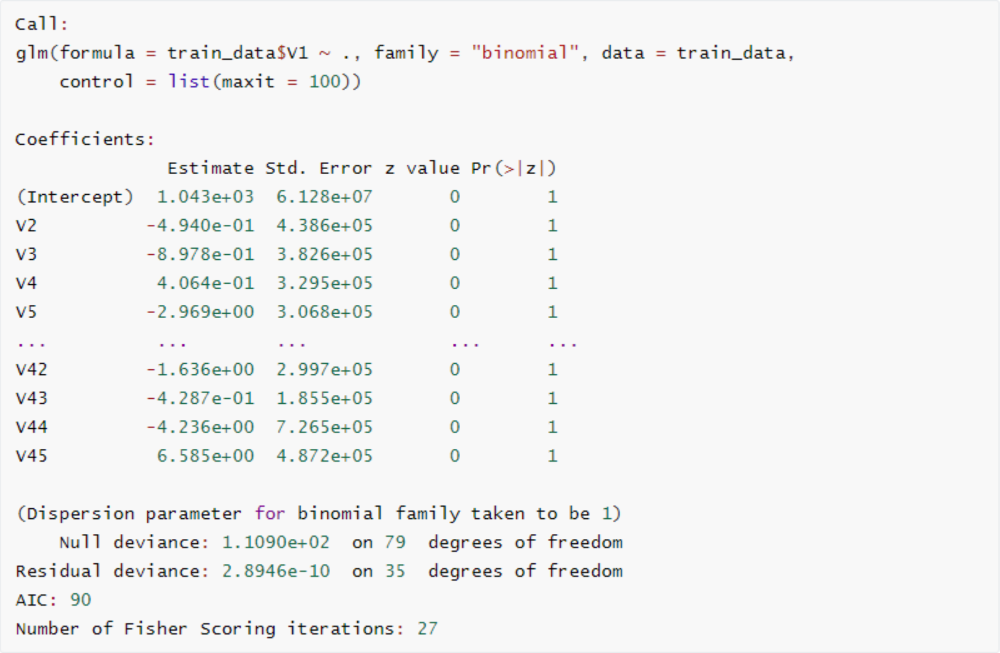
Table 1: AUC Scores for seven Methods without dimension reduction

Classification Methods	Accuracy
SVM	0.771
KNN	0.758
Random Forest	0.778
Mlp-nn	0.688
Logistic	0.631
Naive bayes	0.804
gbm	0.775

We can find that the effect of Naive bayes is the best without dimensionality reduction, which has an AUC: 0.804. Because our data is very sparse, the sample size is only 80. The saturated model is not so well without any dimensionality reduction (this will be explained in 3.2). Therefore, We use lasso method to select variables, and run the model again with the selected variables.

3.2 Why we should do dimension reduction?

In our previous work, all variables in the data set were used to construct the classification model. The results of the Logistic model are shown in the following table:



```
Call:
glm(formula = train_data$V1 ~ ., family = "binomial", data = train_data,
    control = list(maxit = 100))

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.043e+03  6.128e+07       0        1
V2          -4.940e-01  4.386e+05       0        1
V3          -8.978e-01  3.826e+05       0        1
V4           4.064e-01  3.295e+05       0        1
V5          -2.969e+00  3.068e+05       0        1
...           ...        ...          ...      ...
V42         -1.636e+00  2.997e+05       0        1
V43         -4.287e-01  1.855e+05       0        1
V44         -4.236e+00  7.265e+05       0        1
V45          6.585e+00  4.872e+05       0        1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1.1090e+02  on 79  degrees of freedom
Residual deviance: 2.8946e-10  on 35  degrees of freedom
AIC: 90
Number of Fisher Scoring iterations: 27
```

Figure 9: the logistic regression

From the output of the Logistic model, we can see that the model is not convergent. The more likely reason is that there is significant multicollinearity between variables. Therefore, to verify our conjecture, we visualized the correlation coefficients between 44 variables. The correlation coefficient visualization is shown below.

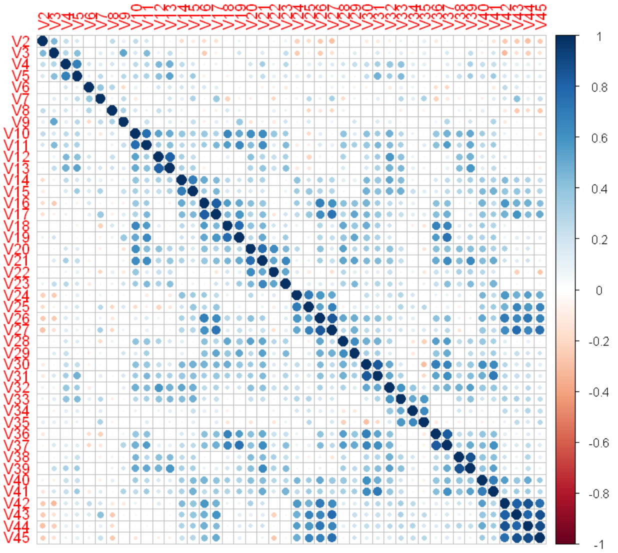


Figure 10: the corr of different factors

It can be seen from the image that there is indeed a relatively high linear correlation between many adjacent variables, which is represented by many small squares with darker colors on the diagonal in the image. Therefore, in order to obtain a more reasonable model, we need to carry out variable selection.

3.3 Dimension reduction

We use the method proposed in section 2.1.1: lasso to achieve dimension reduction and variable selections. We consider 5 folds cross-validation to select the best model. Figure 11 shows the path of tuning parameter so that we can get the best tuning parameter $\lambda = 0.09454452$ and 7 selected variables.

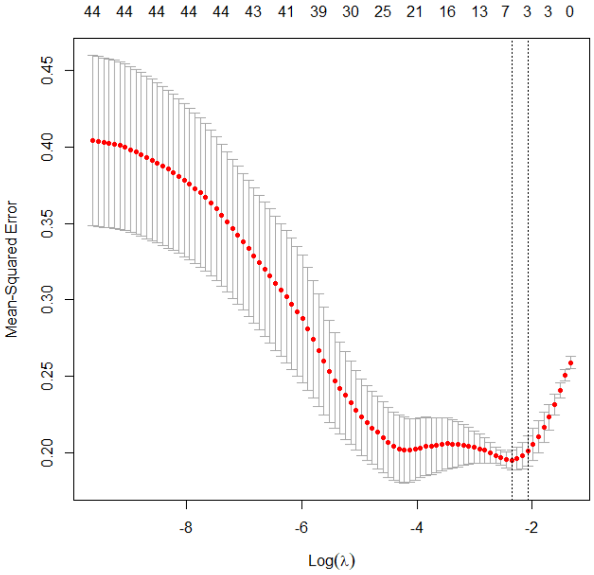


Figure 11: the best tuning parameter of lasso

Then we have finished the variable selections. Next, we will use classification methods to estimate whether he was diagnosed.

Firstly, because the decision tree algorithm can intuitively display the classification process, this paper first uses this method to train the data. The 0 and 1 representing not diagnosed and diagnosed respectively are converted to N and P, making the decision tree clearer.

Specify the maximum depth of 1 first. Figure 12 shows that the generated decision tree is successfully classified, but the AUC value is really low at this time, only 0.561.

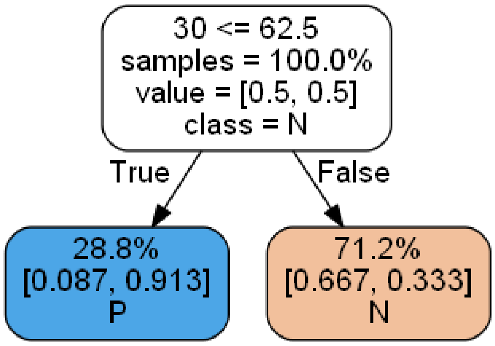


Figure 12: Decision tree, max depth=1

Because too few branches will lead to large prediction deviation, reassign the maximum depth to 3, re-obtain the decision tree classifier, train and predict the corresponding AUC value and decision tree image as Figure 13. The ROC-AUC score attains 0.692.

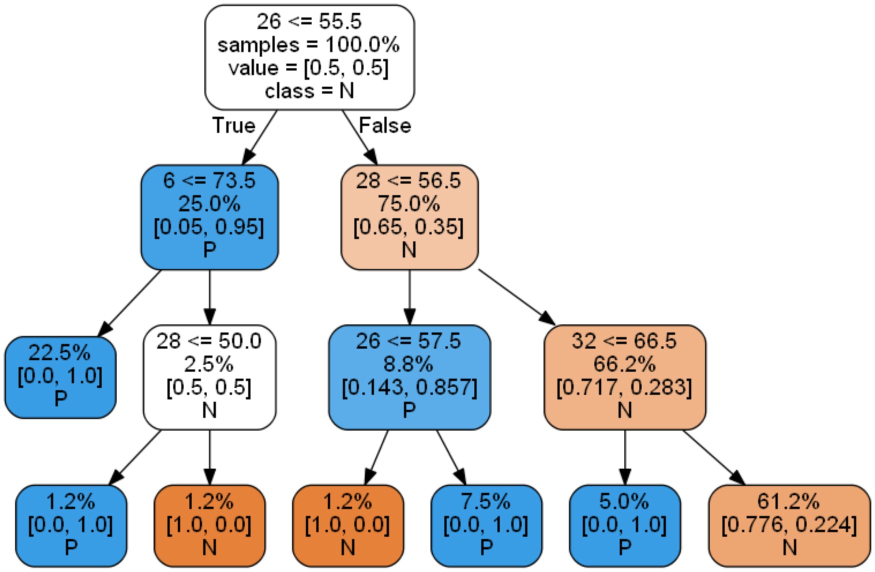


Figure 13: The decision tree, max depth=3

We consider decision tree as an intuitive method, but the accuracy is not so good when applied to the heart dataset. So we re-run the other classification models, The accuary is shown in Table 2.

Table 2: Results for different Classification Methods

Classification Methods	Accuracy
SVM	0.690
KNN	0.734
Naive bayes	0.810
Random Forest	0.785
Mlp-nn	0.650
gbm	0.786
Logistic	0.731

We can find several methods have a higher AUC than the saturated model, especially logistic and gbm.

3.4 Ensemble

In section 3.2, we apply several classification methods to the dataset which has been reduced dimension. In this part, we further use ensembles to imporve our model.

Ensemble strategy consists of two methods: Boosting and Bagging [8]. If prediction errors are relatively uncorrelated, we may choose Bagging to improve the accuracy; Otherwise, boosting would be a great choice. Thus, Checking whether they are relatively correlated is our first order of business, Figure 14 shows their relationships:

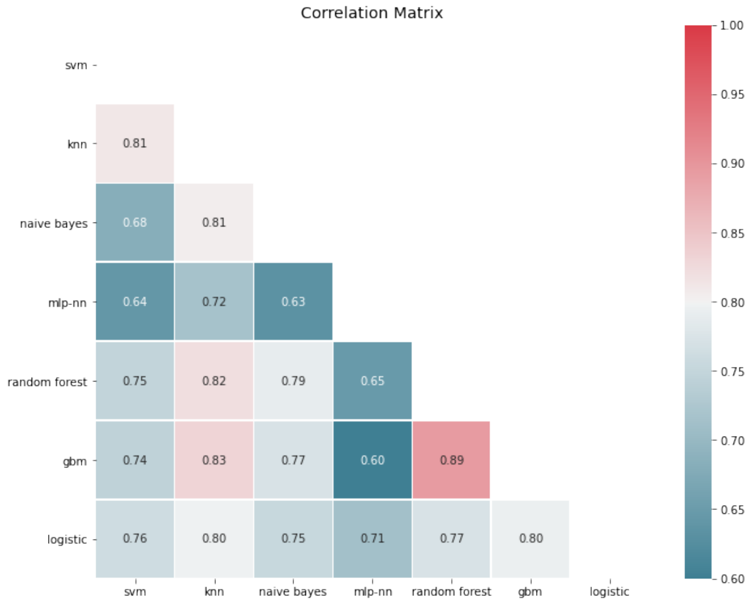


Figure 14: The correlation matrix of the above classification methods

As is shown in Figure 14, we can find that different methods have high correlation, which is to be expected for models that perform well, since it is typically the outliers that are hard to get right. Yet most correlations are in the 70-80% span, so there is decent room for improvement.

We use the boosting method to train the model, and find that the ensemble performs better than most of other single method, which has a high AUC score: 0.803. The ROC curve is shown in Figure 15.

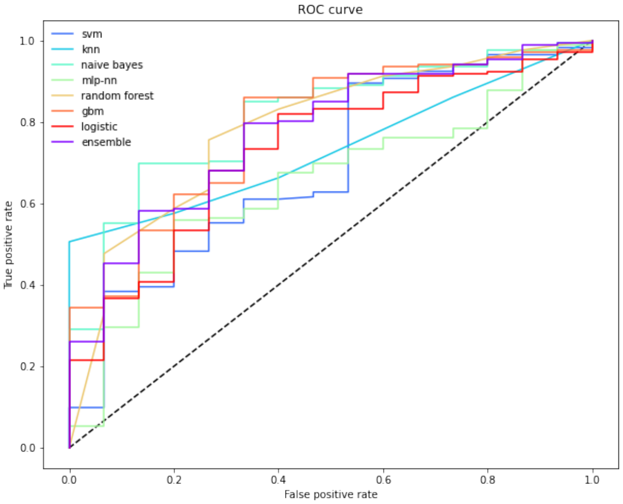


Figure 15: The ROC curve for several classification methods and ensemble

Finally, we create the ensemble of several methods and get an result on this dataset, hope this will help separate heart attack patients from healthy patients!

4 Conclusion

We used the logistic regression and several methods to fit the dataset, but because of multicollinearity, fitted probabilities of logistic regression numerically 0 or 1 occurred. So we can conclude that: the saturated model is not so good. There are a bunch of non-important features. In order to solve this problem, we applied three variable selection methods to develop our model. We compared them and finally used the lasso regression. We selected seven most important factors.

The best model in our mind is the boosting model which ensembles KNN, Naive bayes, random forest and gbm. The final AUC is 0.803, which is a relatively great model.

References

[1] Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M. Goodenday, L.S. ”Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis” Artificial Intelligence in Medicine, vol. 23:2, pp 149-169, Oct 2001

[2] Cios, K.J., Wedding, D.K. Liu, N. CLIP3: cover learning using integer programming. Kybernetes, 26:4-5, pp 513-536, 1997

[3] Tibshirani R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B, Statistical Methodology. 58(1): 267–288.

[4] Podgorelec, Vili Kokol, Peter Stiglic, Bruno Rozman, Ivan. (2002). Decision Trees: An Overview and Their Use in Medicine. Journal of medical systems. 26. 445-63. 10.1023/A:1016409317640. Feature Space. Journal of the Royal Statistical Society, Ser. B, 70: 849–911. 544-549.

[5] Breiman, L.Random Forests. Machine Learning 45, 5–32 (2001).

[6] Logistic Model. In: Encyclopedia of Entomology. Springer, Dordrecht. (2004)

[7] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning20.3 (1995): 273-297.

[8] Trevor Hastie, Robert Tibshirani, Jerome Friedman(2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Ed.), Springer, New York, 337-388