Learning Systems

DA8008 (HT12) Halmstad University

Classification Model for Thyroid Disease

Xu Fei, Qiu Yinan

1. Introduction

This report is focus on constructing a model to tell if a particular set of measurements (test results) comes from a person who is normal, or suffers from being hypothyroid or hyperthyroid (i.e. 3 output categories).

The data for this project are 7200 observations representing patients. 5000 of them are training data, and the other 2200 sample are testing data, and there are 21 variables (there is no information on what these represent).

In this project, Fisher Index for each variable is calculated, variable is ranked according to Fisher Index. For this project, we construct two basic model, one is with KNN (K-Nearest Neighbor) classifier, and another one is the MLP (Multi-Layer Perceptron) classifier. And at last, we make a committee machine to improve the result, in the committee machine, we combine KNN classifier and MLP classifier together.

2. Methodology

2.1 Fisher Index

One way to estimate if a variable is useful for classification or not is to use the "Fisher index". The Fisher index (FI) is defined as:

$$FI(k) = \frac{(\mu_{k,1} - \mu_{k,2})^2}{(N_1 - 1)\sigma_{k,1}^2 + (N_2 - 1)\sigma_{k,2}^2}$$

Where the indices 1 and 2 refer to the two categories, respectively, and

$$\mu_{k,1} = \frac{1}{N_1} \sum_{x(n) \in c_1} x_k\,(n)$$

$$\mu_{k,2} = \frac{1}{N_2} \sum_{x(n) \in c_2} x_k\,(n)$$

$$\sigma_{k,1}{}^2 = \frac{1}{N_1 - 1} \sum_{x(n) \in c_1} [x_k(n) - \mu_{k,1}]^2$$

$$\sigma_{k,2}{}^2 = \frac{1}{N_2 - 1} \sum_{x(n) \in c_2} [x_k(n) - \mu_{k,2}]^2$$

Again, there is the caveat that:

High Fisher index =>Good for classification

2.2 KNN (K-Nearest Neighbor) classifier

In pattern recognition, the k-nearest neighbor algorithm (KNN) is a non-parametric method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor.

2.3 MLP (Multi-layer Perceptron) classifier

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

2.4 Committee Machine

A committee machine is a type of neural network using a divide and conquer strategy in which the responses of multiple neural networks (experts) are combined into a single response. The combined response of the committee machine is supposed to be superior to those of its constituent experts. Compare with ensembles of classifiers.

3. Data

We are given the file thyroidTrain.mat, which contains the matrices trainThyroidInput (5000 × 21), trainThyroidOutput (5000 × 3), and testThyroidInput (2200 × 21). The first matrix, trainThyroidInput, contains the input patterns for the training data. The second matrix, trainThyroidOutput, contains the outputs coded in a "1-out-of-3" fashion. That is, the outputs are coded as (1,0,0), (0,1,0), or (0,0,1). The third matrix, testThyroidInput, contains the inputs for the test data.

From Fig 3.1 we can see that the different variable show different contribution for the different class, for the class one, the 19th variable effect the most of this class, the third variable effect the second of the class two most, and the 19th variable effect the class three most.

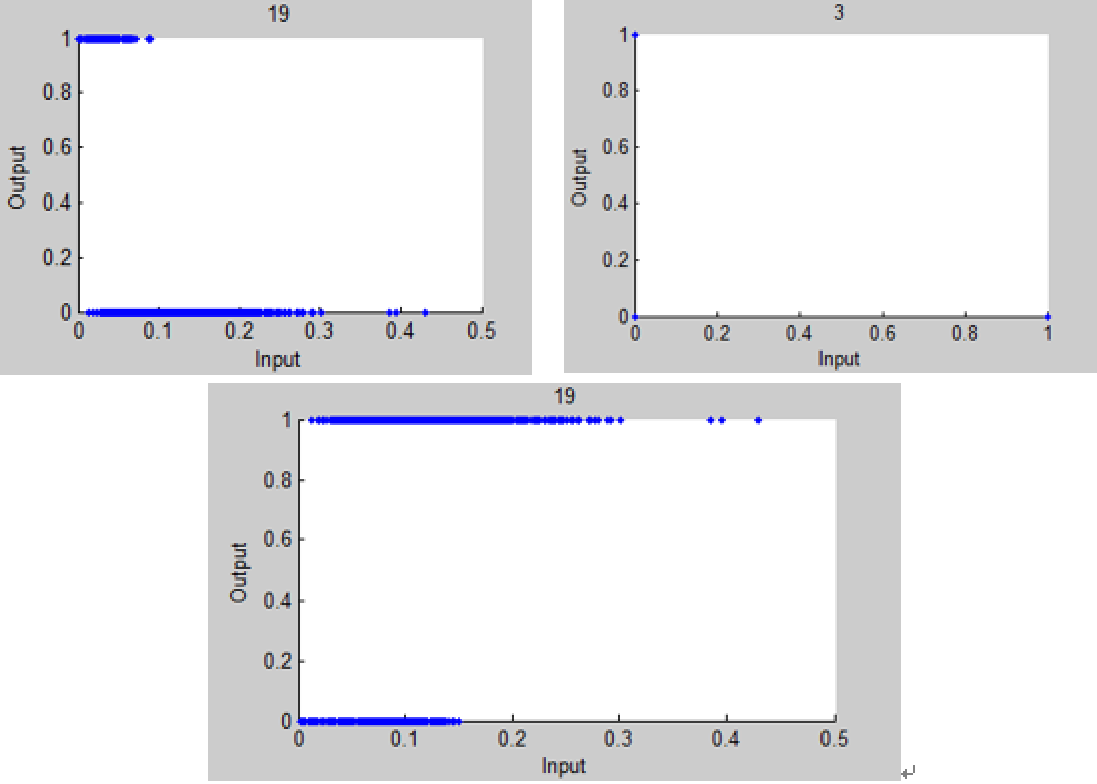

Fig 3.1 the relation between input and output

For this data, the output (trainThyroidOutput) is three dimension, that means we have three classes, we need to use Fisher Index to estimate if a variable is useful for classification or not. The result shows that from the least one the biggest one, the variables are: 17,21,19,18,10,3,20,2,16,1,7,13,8,6,4,12,14,11,5,9,15.

But the Fisher Index is used for two classes, here we should process the data first, I separate the class1 into class1 T and class1 F, that means the data is in class1 or not, then do the same measurement to the class2 and class3 also. So in the Fisher Index function, we can put the two class parameters.

Then we use the scatter plot to see if there are some relations between two variables, a simple way to find out if there is an approximate linear relationship between x_1 and x_2 is to compute Pearson's correlation coefficient, Fig 3.2 shows the result that we get use $Correff()$ function in Matlab, these four image are corresponding to the four pairs of the variables that with the highest Pearson’s correlation coefficient.

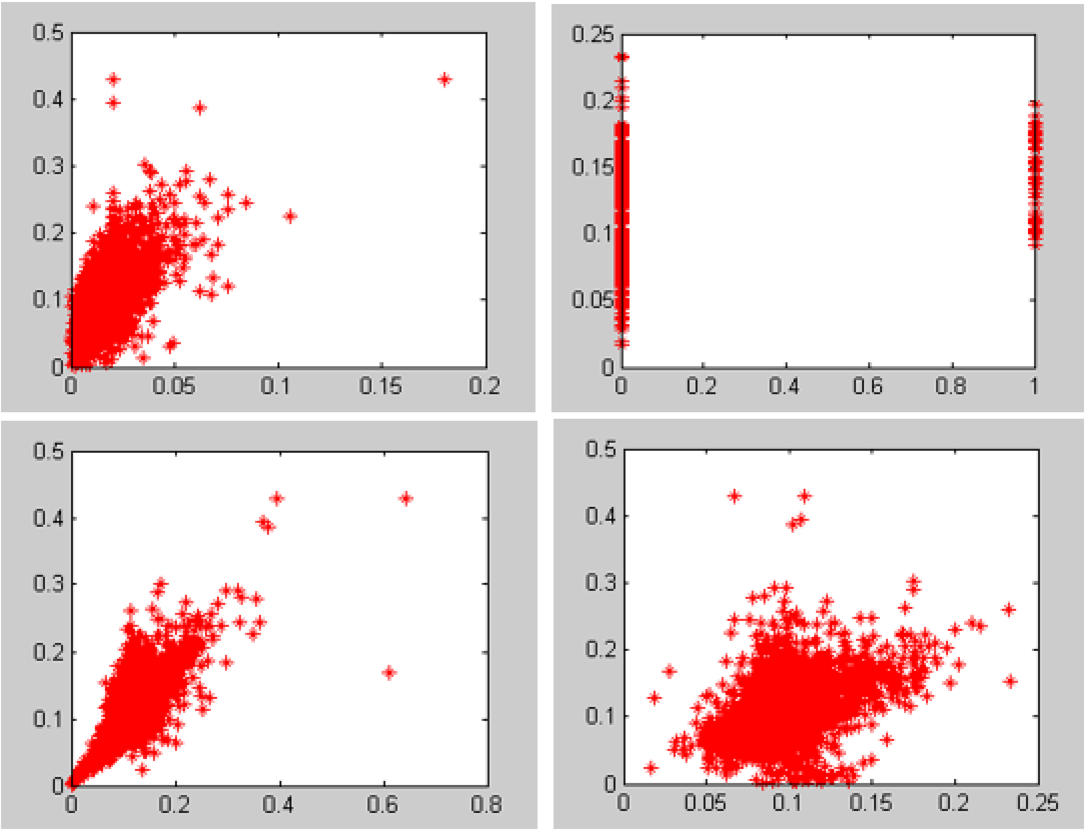

Fig 3.2 the relation between variables

4. Result

In this section, Result of Fisher Index, Gaussian linear classifier, logistic regression classifier, k-nearest neighbor classifier, multilayer perceptron learning method is presented.

4.1 KNN (K-nearest Neighbor) classifier

We have introduce the method of KNN classifier in section 2.2.

At first, I try to find the best K that means the best value of the neighbors. I try the value from 1-10, and use the kloss (kloss is the MSE that use cross-validation to calculate with the KNN model) to estimate the KNN classifier, the result is:

1	0.0754
2	0.0714
3	0.0612
4	0.0632
5	0.0620
6	0.0640
7	0.0614
8	0.0614
9	0.0636
10	0.0636

Then, I try to find the best set of variables that can fit the KNN classifier best, the features data have already sorted by the Fisher Index, I try from 1 to 21 features, Fig 4.1 shows the relation between the number of feature that I use with the kloss, and I also use kloss to estimate the model.

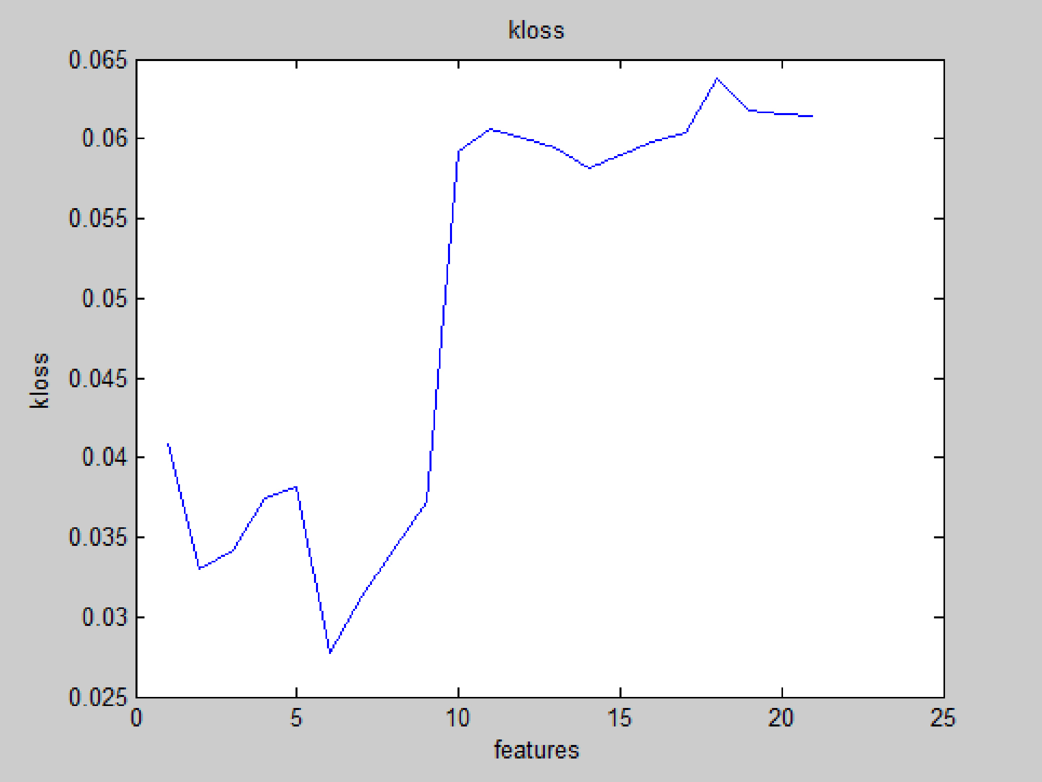

Fig 4.1 relation between features and kloss

From the result, we can clearly see that if we use 6 features that sorted by the Fisher Index with 6 highest value can get the least kloss, that means the best result we can get is to use 6 features.

So, the final model of KNN is 3NN and use six features as input.

Finally, we create a function to test the model, use the trainThyroidInput to get the corresponding output, and we compare it to the raw output (trainThyroidOutput), Fig 4.2 shows the origin class with variable 20th and 19th. Here I just plot out 500 observations to make the plot more clearly.

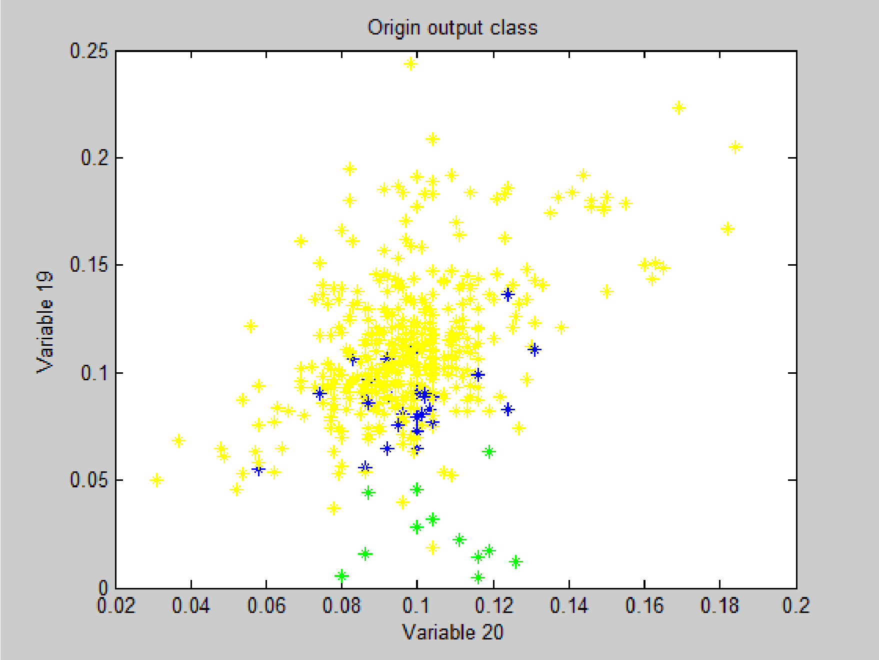

Fig 4.2 origin class with variable 20th and 19th

Fig 4.3 present the output class use all features, and also with the same variables.

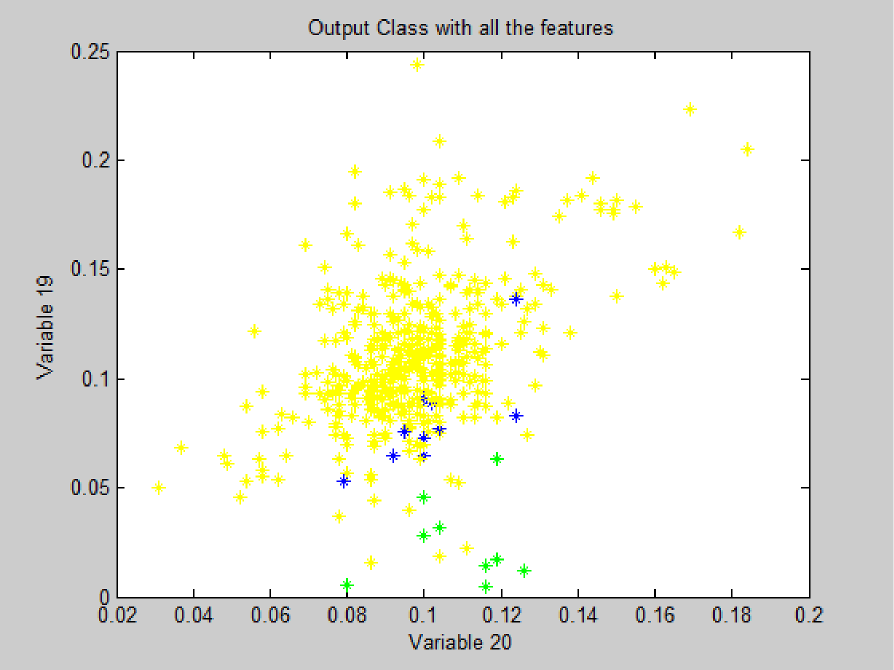

Fig 4.3 output (all features) class with variable 20th and 19th

Fig 4.4 present the output class use 6 features (the final model of KNN), and also with the same variables.

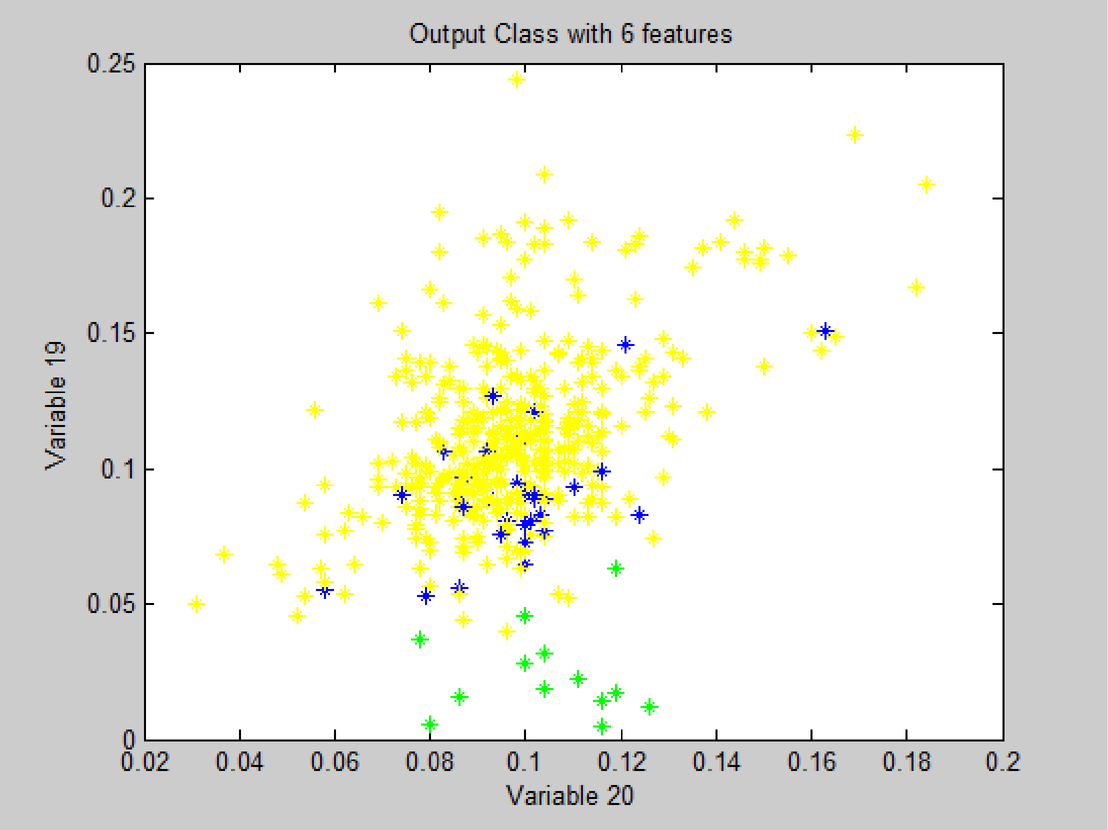

Fig 4.4 output (6 features) class with variable 20th and 19th

The MSE that we use 10-Fold to do the cross-validation is MSE = 0.0256.

4.2 MLP (Multi-layer Perceptron) classifier

We have introduce the method of KNN classifier in section 2.3.

At first, I try to find the best number of the hidden nodes that can the MLP classifier has the best performance. I try the nodes number from 4-10, and use the validation performance (it's same as MSE that calculate by the cross-validation with MLP model) to estimate the MLP classifier, the result is:

4	**0.0113**
5	**0.0152**
6	**0.0129**
7	**0.0141**
8	**0.0179**
9	**0.0181**
10	**0.0143**

Then, I try to find the best set of variables that can fit the MLP classifier best, the features data have also sorted by the Fisher Index, I try from 1 to 21 features, Fig 4.5 shows the relation between the number of feature that I use with the performance, and I use validation performance to estimate the model.

(In the requirement, we need to use the remaining features as the input for MLP classifier, but the result is really bad, because the Fisher Index of remaining features are too low, they can’t classify the input well. Later I will show an output plot of it.)

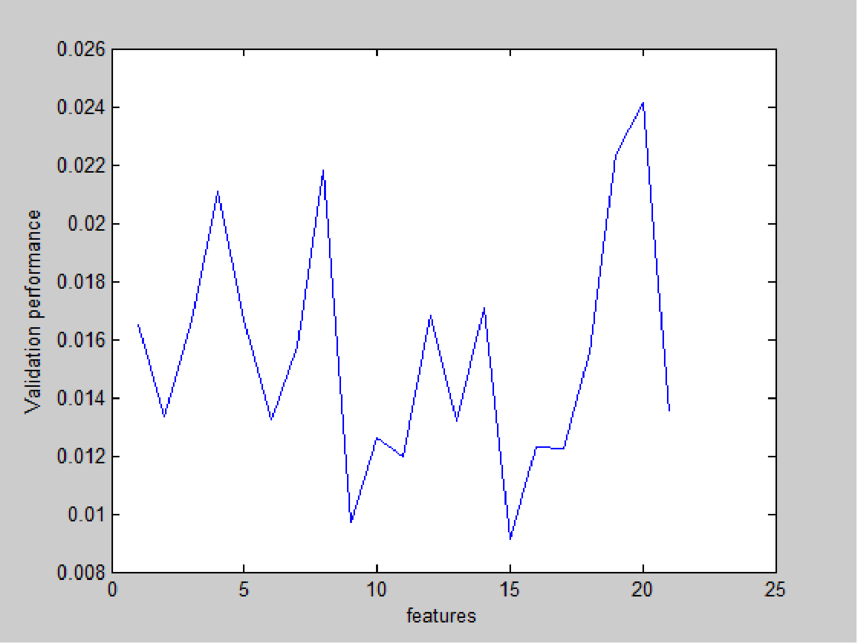

Fig 4.5 relation between features and validation performance

From the result, we can clearly see that if we use 15 features that sorted by the Fisher Index with 15 highest value can get the least validation performance (but it’s not stable because the construct of MLP classifier, I just always choose the best value).

Because the MLP classifier use BPNN as its construct, it’s not stable enough and have the local minimize extreme value problem, so I add another layer to the MLP classifier, I add a radbas which call Radial Basis Function as the first layer to make it more stable and overcome the local minimize extreme value problem. The first hidden layer is 4 hidden nodes, the second hidden layer is 4 hidden nodes too.

Finally, we create a function to test the model, the input and output is the same as the KNN classifier model. The plot result is shown in Fig 4.6.

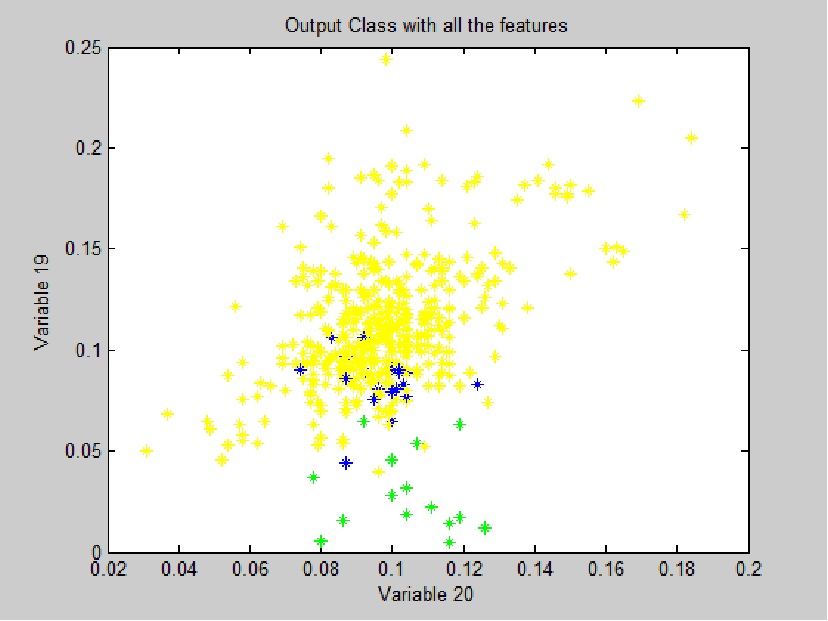

Fig 4.6 output (all features) class with variable 20th and 19th

Fig 4.7 present the output class use some features (the final model of MLP), and also with the same variables. It's almost the same as the origin output class that mean this model shows really good performance.

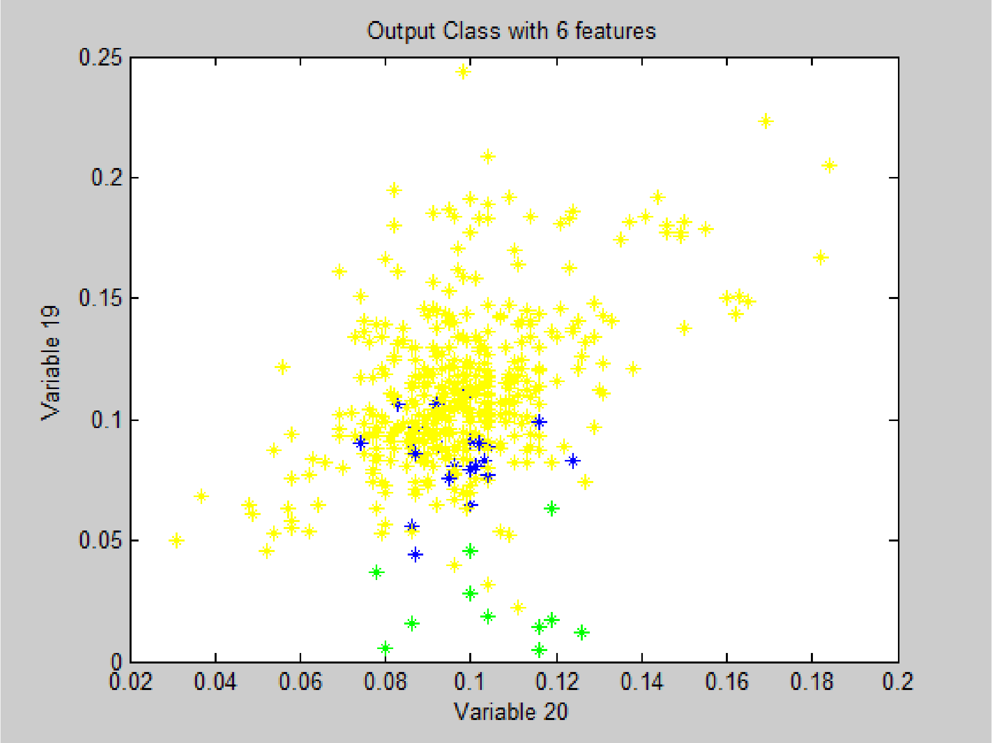

Fig 4.7 output (part of features) class with variable 20th and 19th

Fig 4.8 presents the performance of the final MLP classifier model. **The MSE that calculated by cross-validation is MSE=0.0073237**

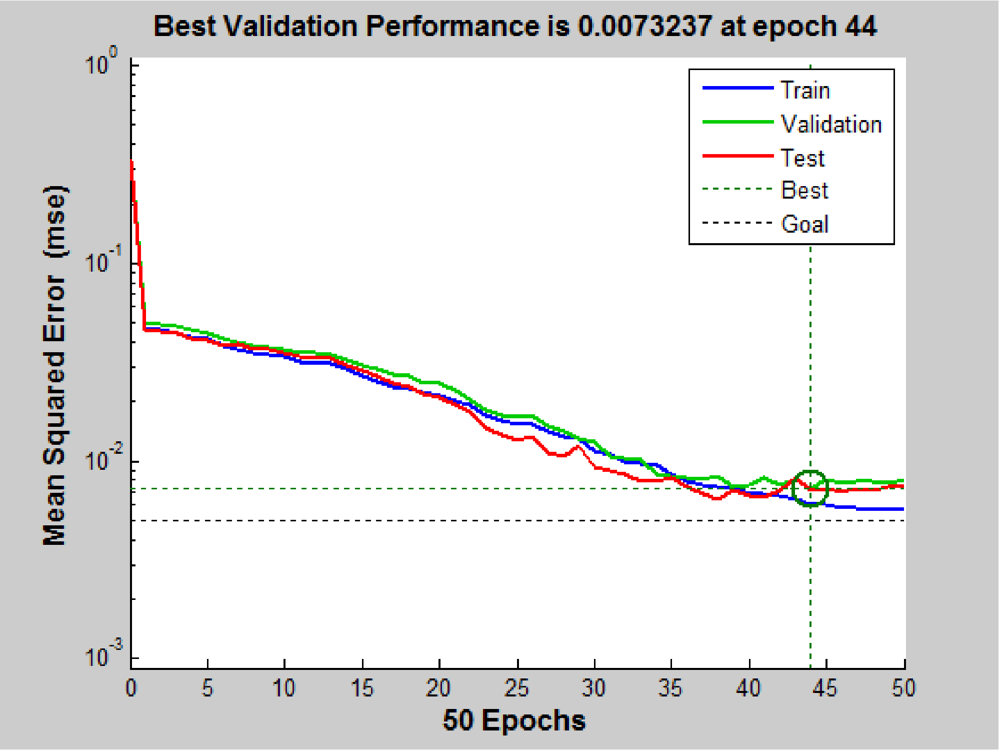

Fig 4.8 performance of the MLP classifier model

4.3 Committee Machine

At last, I used the committee machine that combine KNN classifier with the MLP classifier together to improve the result. Here, I used the MSE of each model as the weight to calculate the output of the committee machine. The function is:

$$\hat{y}_{wcom} = \sum_{k=1}^{K} \alpha_k \hat{y}_k(n)$$

$$with\ \alpha_k = \frac{MSE_k^{-1}}{\sum_{j=1}^{k} MSE_j^{-1}}$$

The final result of committee machine output is MSE=0.0073237, it the same as the MLP classifier because there's only two model, every time the output will be partial to the model that with smaller MSE.

4.4 Attachment

Fig 4.9 is the output class with variable 20th and 19th but calculated by the model that for the task (The MLP classifier use remaining features). We can see the result is really bad clearly.

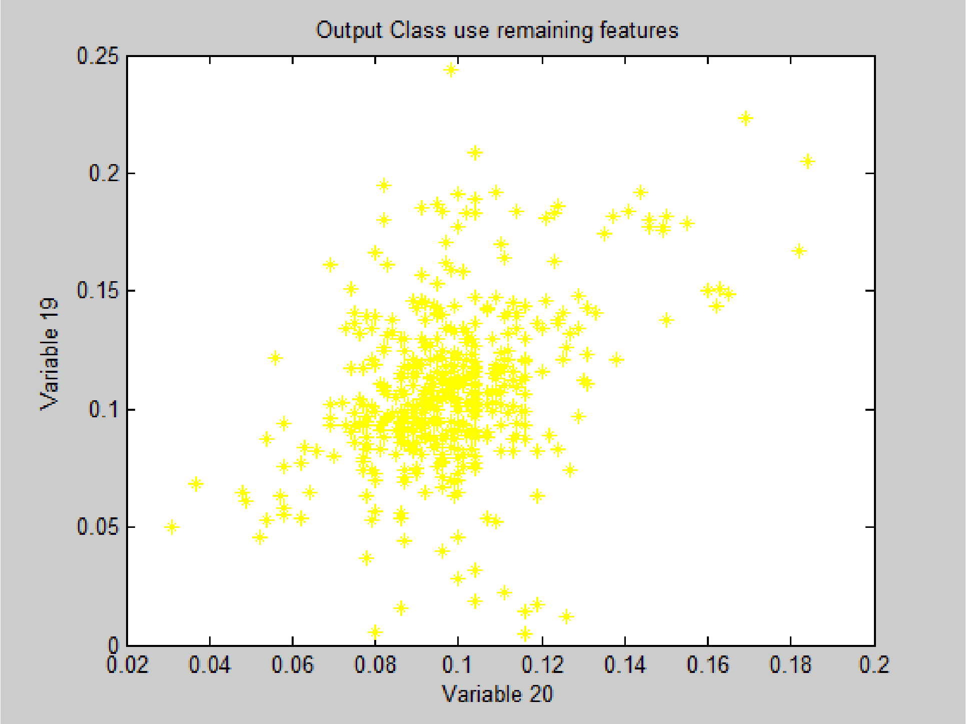

Fig 4.9 output (remaining features) class with variable 20th and 19th

5. Conclusion

For this project, we can get the conclusion from the result that committee machine should have the best result, but in this project, the committee machine only have two models, so the committee machine will be partial to the model with smaller MSE. Between the KNN classifier model and the MLP classifier model, the MLP classifier have better result, in my opinion, I think it's because the construct and the method of the KNN classifier, the method itself with more errors. For more development, I think I should do more test for the data, to move the outliers.