Heart Disease Detection Project Report

Group72 Member: Yangguang He, Xinlong Li, Ruixian Song

Abstract—Our project object is to detect whether patients have heart disease or not by given a number of features from patients. The motivation of our project is to save human resources in medical centers and improve accuracy of diagnosis. In our project we use different methods to detect heart disease such as Logistic Regression, SVM, Naïve Bayes, Random Forest and Artificial neural network. And among all these algorithms Random Forest gives us the best accuracy of 91.8%.

I. INTRODUCTION

Our problem is that we want to predict whether patients have heart disease by given some features of users. This is important to medical fields. If such a prediction is accurate enough, we can not only avoid wrong diagnosis but also save human resources. When a patient without a heart disease is diagnosed with heart disease, he will fall into unnecessary panic and when a patient with heart disease is not diagnosed with heart disease, he will miss the best chance to cure his disease. Such wrong diagnosis is painful to both patients and hospitals. With accurate predictions, we can solve the unnecessary trouble. Besides, if we can apply our machine learning tool into medical prediction, we will save human resource because we do not need complicated diagnosis process in hospitals. (though it is a very long way to go.) The input to our algorithm is 13 features with number values. We use several algorithms such as Logistic Regression, SVM, Naïve Bayes, Random Forest, Artificial Neural Network to output a binary number 1 or 0. 1 indicates the patient has heart disease and vice versa.

II. RELATED WORK

Before we did the experiments, we did research on how people explored heart disease prediction so that we can broaden our horizons and learn from them.

In 2011, Ujma Ansari [1] made use of Decision Tree model to predict heart disease and get a high accuracy of 99%, which inspires us to use a better version of Decision Tree and it is Random Forest. Unfortunately, the paper uses a dataset with 3000 instances but dose not provide a reference of how they get the data. The UCI website only provides 303 instances of dataset so we doubt where the author gets 3000 instances of dataset.

In 2012, Chaitrali S. Dangare [2] made the prediction by using three models and such models are Naïve Bayes, Decision Trees and Neural Network. We are using the same dataset as he did. The difference between his work and ours is that he added 2 more features into the dataset, which means there are 15 features of his work while there are 13 features in our dataset. Though there is no big difference between 13 features and 15 features in his work, what he did on dataset inspires us to make useful change to our dataset (Try normalization on dataset) to make our results comprehensive. However, during this paper there are only 3 models. More models need to be considered so that the results are comprehensive.

In 2017, Kaan Uyar and Ahmet İlhan[3] did the same experiment and used the same dataset as we did for projects. During their analysis, "Class distributions are interpreted as 54% absence and 46% presence of a heart disease". The dataset we download from Kaggle has 54% 1s and 46% 0s in the target column. From their analysis, we realize 1 indicates absence of heart disease and vice versa. To make it easily understood, we switched 1s and 0s in the target column so that 1 indicates presence of heart disease to show our confusion matrix[10] in our results.

After reviewing paper [4] and [5], we have learned that neural network has advantage of fault tolerance and it has the ability to work with inadequate knowledge as human beings. Therefore, in our project we decide to spend some time working on neural network to detect heart diseases.

III. DATASET AND FEATURES

Our dataset is based on UCI heart Disease Data Set [6] and we have 303 instances. According to UCI, "This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them."We guess too many features will bring too much noise so people has done feature extraction and reduce 76 features to 14 features. To better understand the meaning of the features, we have the responsibility to explain some of the attributes of original dataset from UCI as follows:

- age: age in years
- sex: sex (1 = male; 0 = female)
- cp: chest pain type
  -- Value 0: typical angina
  -- Value 1: atypical angina
  -- Value 2: non-anginal pain
  -- Value 3: asymptomatic
- trestbps: resting blood pressure (in mm Hg on admission to the hospital)
- chol: serum cholestoral in mg/dl
- fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- target: Heart disease (0 = no, 1 = yes)

Since the original dataset has missing values, we just downloaded a clean dataset from Kaggle[7]. We have split the dataset into 80% (242 instances) for training and 20%(61 instances) for test. We did normalization on our dataset to avoid overfitting. What we did to our dataset is to change 1s to 0s in target column and vice versa in order to make value 1 indicate the presence of heart disease and make value 0 indicate the absence of heart disease. Given such dataset we can do many interesting predicative tasks. For example, we can use these features to predict chest pain type. But the most important thing is that given the 13 attributes from a patient, we want to predict whether he has the heart disease or not because keeping healthy is very import to people.

IV. Methods

During this project, we have tried 5 algorithms for experiments and they are Logistic Regression SVM, Naïve Bayes, Random Forest and Neural Network.

A. Logistic Regression

Logistic Regression is a supervised learning that computes the probabilities for classification problems with two outcomes. It can also be extended to predict several classes. In Logistic Regression model, we apply the sigmoid function, which is

$$\sigma(\mathrm{z}) = \frac{1}{1+e^{-z}}.$$

This function successfully maps any number into the value between 0 and 1 and we can regard this value as the probability of predicting classes. For example, we have two classes and they are presence of heart disease and absence of disease. If we set the threshold as 0.5, applying the sigmoid function gives us a value of 0.7, which means the man has the 70% probability of having heart disease so we will predict that he has heart disease.

B. SVM (Support vector machine)

SVM aims to find a hyperplane in multiple dimensions (multiple features) that classifies the dataset. Here is a picture[8] of classification by SVM.

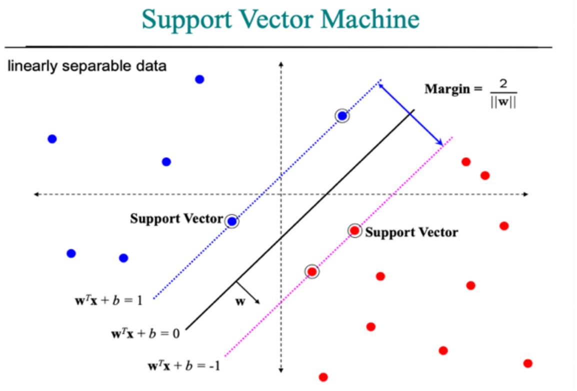

Fig. 1. Classification by SVM

The equation of the hyperplane form is

$$w^T x + b = 0$$

where w is a weight vector, x is input vector and b is a bias. The margin is the distance of closest points from the hyperplane and is calculated as

$$\frac{w}{||w||} * (x_+ - x_-) = \frac{w^T(x_+ - x_-)}{||w||} = \frac{2}{||w||} \qquad (8)$$

where $w^T x_+ + \mathrm{b} = +1$ and $w^T x_+ + \mathrm{b} = -1$. Our object is to maximize the margin $\frac{2}{||w||}$ or equivalently to minimize $||w||^2$. After adding loss function, the learning problem is to find a weight vector w that minimizes the cost function of (8)

$$||w||^2 + C\sum_{i}^{N} max\,(0{,}1 - y_i f(x_i))$$

And Gradient descent algorithm is able to minimize the cost function by iteratively updating the equation (8) of

$$w_{t+1} \leftarrow w_t - \eta_t \nabla_w C(w_t)$$

where η is the learning rate.

C. Naïve Bayes

Naïve Bayes assumes the independence between the features of the dataset and the Bayes Rule is

$$\mathrm{P(y\mid x)} = \frac{P(x|y)P(y)}{P(x)}$$

where P(y|x) is the probability of classification y given the data x. Applying Bayes theory, we can build a Naïve Bayes model to compute the probabilities from training data and then make predictions based on the features of the test data.

D. Random Forest

Random Forest is an ensemble learning method for classification and regression by constructing multiple decision trees in training and outputting the classification or prediction(regression). The goal of Random Forest is to combine weak leaning models into a strong and robust leaning model. From a tutorial[9] online, we learn that the algorithm of Random Forest can be summarized in 4 steps:

Step 1:Randomly draw M bootstrap samples from the training set with replacement.

Step 2: Grow a decision tree from the bootstrap samples. At each node: Randomly select K features without replacement and split the node by finding the best cut among the selected features that maximizes the information gain.

Step 3:Repeat the steps 1 and 2 T times to get T trees;

Step 4:Aggregate the predictions made by different trees via the majority vote.

E. Neural Network

From the picture we can see neural network has a collection of neurons and each neural node is connected with other neuron nodes through links. Each link has a weight as influence from one neuron node to another node. During the neural network, the input layer accepts the features of data as the weights and each neuron node in the first layer multiplies with its weight and results are summed up and transferred to the corresponding neuron node in the hidden layer and so on. Finally, each neuron node in the output layer will get the probability of its corresponding classification. During the training process, the backpropagation algorithm calculates the gradient of the error function and update the weights of each neurons.

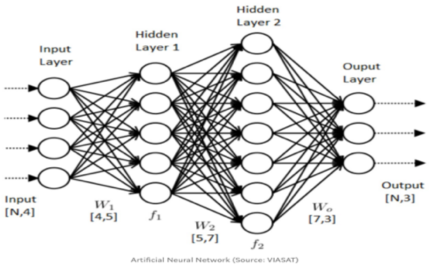

Fig. 2. Neural Network

V. Experiments/Results/Discussion

Since our project is a classification problem, we use accuracy, precision, recall and F1 score to evaluate the models. We would like to introduce the meaning of TP,FP,TN and FN. A true positive (TP) is a positive outcome predicted by the model correctly while a false positive (FP) is a positive outcome predicted by the model incorrectly. A true negative (TN) is a negative outcome predicted by the model correctly while a false negative (FN) is a negative outcome predicted by the model incorrectly.

We did not use cross-validation because our dataset is not very sufficient. We split the dataset into 80% for training and 20% for test. Here is the table of results of different methods and we will talk about each evaluation of methods in details.

TABLE I. Result Of Different Methods

Methods	Train accuracy	Test accuracy	precision	recall	F1 score
Logistic Regression	83.88%	85.25%	0.88	0.78	0.82
SVM	89.26%	86.89%	0.91	0.78	0.84
Naïve Bayes	83.47%	85.25%	0.88	0.78	0.82
Random Forest	100%	91.80%	0.92	0.89	0.91
Neural Network	83.88%	88.52%	0.92	0.81	0.86

A. *Logistic Regression*

Here is the confusion matrix of the Logistic Regression:

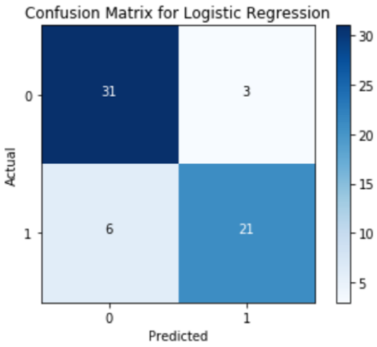

Fig. 3. Confusion Matrix for Logistic Regression

I used the L2 penalty, the square of the magnitude of coefficients, supported by Logistic Regression to avoid overfitting. The train accuracy is 83.88% and test accuracy is 85.25%. It performs well but not the best for us. The advantage of the Logistic Regression is that it does not need too much computational resources and it is highly interpretable. So it is easy and sufficient to apply Logistic Regression. However, the limitation of Logistic Regression is that it assumes linearity between the features of the dataset. In the real world, the data is rarely separable, neither as our dataset. That is why we cannot reach a very high accuracy of 90%.

B. *SVM*

Here is the confusion matrix for SVM:

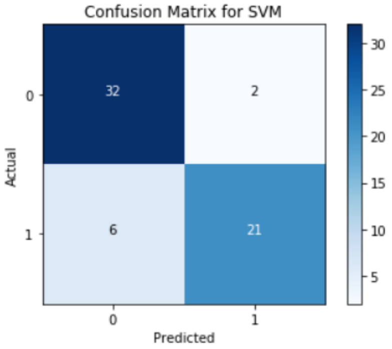

Fig. 4. Confusion Matrix for SVM

According to the tutorial of sklearn, for a small dataset it is better to use sklearn.svm.SVC(). The training accuracy is 89.26% and the test accuracy is 86.89%. The advantage of SVM is that it is very efficient with high dimensional spaces. The main disadvantage is that the SVM has many parameters that needs to be correctly chosen to achieve the best performance. For safety we just use the default parameters of SVM. And the test accuracy of 86.89%, which is better than Logistic Regression.

C. *Naive Bayes*

The confusion matrix for Naïve Bayes is

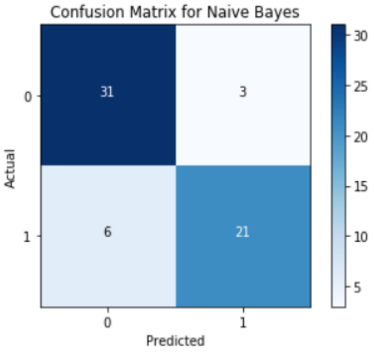

Fig. 5. Confusion Matrix for Naïve Bayes

The train accuracy is 83.47% and the test accuracy is 85.25%. The advantage of Naïve Bayes is that Naïve Bayes is able to make predications given a small amount of training data. The disadvantage of Naïve Bayes is that it assumes all features are mutually independent but in real life we can rarely get a dataset whose attributes are mutually independent and that might be why we cannot reach a very high accuracy of 90%.

D. Random Forest

The confusion matrix of Random Forest is:

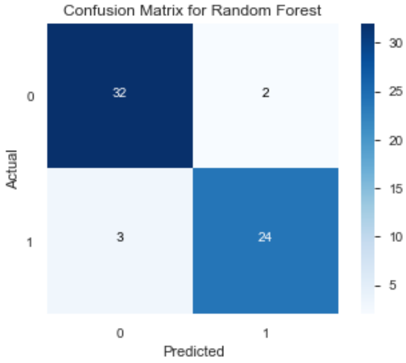

Fig. 6. Confusion Matrix for Random Forest

The train accuracy is 100% and the test accuracy is 91.80%. At the first beginning we use the default parameters (n_estimators=100, which means the number of trees in the forest is 100 and max_depth = None, which means the nodes are expanded until all leaves are pure or all leaves contain less than the minimum number of samples required to split an internal node). Though we get 100% test accuracy, we only get 85.25% test accuracy. We guess it might be overfitting. One reason might be the training data is not generalized during the training process so we decide to shuffle the dataset again and we tried the parameter random_state from 1 to 2000. When random_state is 1826, the test accuracy is 91.80%. Then we tried experiments on parameters of n_estimatros(from 10 to 300) and max_depth(from 10 to 300) and the best test accuracy is still 91.80%. This means with random_state=1825, the other default parameters are good enough to get the best test accuracy. For example, the number of trees in the forest is 100, which is appropriate. If the number of trees is small, it will cause underfitting because the model has not been optimized for the training data, let alone the test data. If the number of trees is too big, it will cause overfitting because the model become so complexed and sensitive to new data. The advantage of Random Forest is that it can deal with dataset with high features and balance the variance and it is not sensitive to the noise of the data. Among these 5 models, Random Forest outperforms any other models.

E. Neural Network

At the first beginning we tried to add 3-4 hidden layers in our neural network but it performs bad. The test accuracy is only 60%. Then we analyzed that the dataset is not big so we decide to make our network simple. At last we have only 1 hidden layer with 31 neuron nodes. For the optimization we use Adam instead of SGD (Stochastic Gradient Descent) because Adam is a combination of RMSprop and SGD with momentum and it takes advantage of momentum by moving average of the gradient. Our learning rate is 0.001, which is appropriate because the loss goes down in a normal speed. Since our dataset is not big, we just choose the batch size to be 200, which is enough for training. And we run 80 epochs to avoid overfitting. The train accuracy is 93.02% and the test accuracy is 88.52%, which is the second best. Here is the plot of Accuracy vs Epoch and Loss vs Epoch:

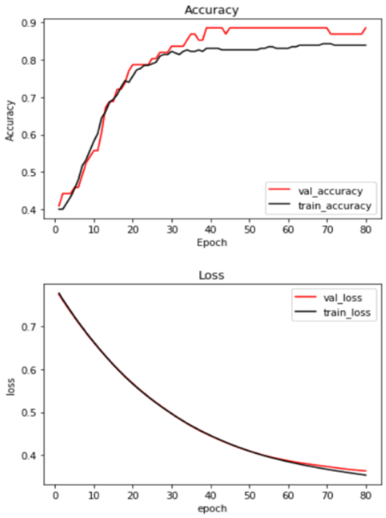

Fig. 7. Accuracy and Loss

As the epochs increase, the loss for test data is reaching 0.35 and test accuracy is reaching 89%. The advantage of the neural network is that neural network can deal with complicated datasets with high dimensional features (e.g. images) and make accurate predictions by building several hidden layers. However, when it comes to small dataset, the neural network does not perform well because it tends to become complicated.

VI. Conclusion/Future Work

We use some libraries[11] provided by Python to implement this project. After the experiments, the algorithm of Random Forest gives us the best test accuracy, which is 91.8%. The reason why it outperforms others is that it is not limited to the property of the dataset. Naïve Bayes requires the features to be mutually independent. Logistic Regress requires the features to be linearly separable. SVM requires the parameters to be appropriately set and the neural network requires a complicated and big dataset. Though we get a good result of 91.8% accuracy, that is not enough because it cannot guarantee that no wrong diagnosis happens. To improve accuracy, we hope to require more dataset because 300 instances of dataset are not sufficient to do an excellent job. In the future, to predict disease we want to try different diseases such as lung cancer by using image detection. In this way, the dataset becomes complicated and we can apply convolutional neural network to make accuracy predictions.

References

[1] Soni, Jyoti, et al. "Predictive data mining for medical diagnosis: An overview of heart disease prediction." *International Journal of Computer Applications* 17.8 (2011): 43-48.

[2] Dangare, Chaitrali S., and Sulabha S. Apte. "Improved study of heart disease prediction system using data mining classification

techniques." *International Journal of Computer Applications* 47.10 (2012): 44-48.

[3] Uyar, Kaan, and Ahmet İlhan. "Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks." *Procedia computer science* 120 (2017): 588-593.

[4] Kim, Jae Kwon, and Sanggil Kang. "Neural network-based coronary heart disease risk prediction using feature correlation analysis." *Journal of healthcare engineering* 2017 (2017).

[5] Baccouche, Asma, et al. "Ensemble Deep Learning Models for Heart Disease Classification: A Case Study from Mexico." *Information* 11.4 (2020): 207.

[6] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[7] https://www.kaggle.com/ronitf/heart-disease-uci

[8] https://www.robots.ox.ac.uk/~az/lectures/ml/lect2.pdf

[9] https://nthu-datalab.github.io/ml/labs/03_Decision-Trees_Random-Forest/03_Decision-Tree_Random-Forest.html

[10] https://www.kaggle.com/jprakashds/confusion-matrix-in-python-binary-class

[11] scikit-learn, keras, pandas and matplotlib

Individual contributions

Ruixian Song
He worked on looking for the topic and references before we start the project. He also worked on functions to plot the table of results.

Xinlong Li
He worked on first three models and they are Logistic Regression, SVM and Naïve Bayes.

Yangguang He
He worked on the last two models and they are Random Forest and neural network.

All of the members spent effort writing the final report.

Replies to critical reviews

Critical review from team 15:

What is the intuition behind the structure of the neural network?
Our response: At last we decide to make the architecture simple. The number of hidden layers is 1 and the hidden layer has 31 neurons. We make it simple because the dataset is not very big. Therefore, we do not have to make neural network complicated, otherwise it will be overfitting.

Results of neural network show possible over-fitting?
Is this monitored during the training process?
Without the accuracy and loss plot it is hard to the audience to see.
Our response: As the figure 7 shows in page 4, no overfitting appears. We have monitored the training process and use the results to plot the figure 7. Figure 7 is the plot of accuracy and loss and it is in our final report.

Critical review from team 68

Could you explain more about what kind of kernel in SVM do you use?: Please explain how do you decide which kind of kernel you used?
Our Response: We use RBF kernel in SVM because it works well in practice and it is relatively easy to calibrate compared to other kernels.

How do you decide the number of neurons in each hidden layer of your neural
network model?
Our Response: At first we have tried using several hidden layers but it did not perform well. Since the data set is not big, we tried to use just 1 hidden layer. And we tried the number of neurons from 10 to 40. When the number of neurons is 31, the neural network performs best.

Critical review from team 78
For logistical regression, why was the test accuracy higher than the training accuracy?
Our Response: Since the test data is not big, it might because the test data is easier to classify than the training data. If we have a larger dataset, such situation will not happen.

Why was relu used specifically and why wasn’ t batch normalization used or pooling?
Our Response: Relu is defines as Y = max(0, x). It is the most commonly used activation function in neural networks. Batch normalization can stabilize the learning process and reduce the number of training epochs to train deep networks. Pooling aims to reduce the computation in the network.

Since our architecture of neural network is not deep, we do not need these two techniques.

What specific heart disease was predicted or was it just any kind of heart disease?
Our Response: The dataset does not give a specific heart disease so we guess it might be any kind of heart disease.

Was there any specific feature that had a larger impact determining the condition of heart disease?
Our Response: Random forest has a function to compute the importance score of each feature. The features of "cp" (chest pain type), "ca" (number of major vessels colored by flourosopy)and "thal" (0= normal; 1 = fixed defect; 2 = reversable defect) have a larger impact determine the condition of heart disease than the others.