Research Project

Tushar Gautam

Submitted: 6 May 2022

Abstract

The code for the project can be found here.

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, and in developing nations it's crucial to have accurate and cheap automated systems for detecting such diseases. Very simple models like Bayesian Logistic Models and Bayesian Probit Models can be really quick in setting up but lack in accuracy because they can't handle complicated data well. There can be an improvement if the non-linearlity in the data is reduced by removing the selected features.

Motivation and Dataset Details

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of five CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Following are the attributes of the dataset with details regarding each one:

1. **Age:** age of the patient [years]
2. **Sex:** sex of the patient [M: Male, F: Female]
3. **ChestPainType:** chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. **RestingBP:** resting blood pressure [mm Hg]
5. **Cholesterol:** serum cholesterol [mm/dl]
6. **FastingBS:** fasting blood sugar [1: if FastingBS ¿ 120 mg/dl, 0: otherwise]

7. **RestingECG:** resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of ¿ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. **MaxHR:** maximum heart rate achieved [Numeric value between 60 and 202]
9. **ExerciseAngina:** exercise-induced angina [Y: Yes, N: No]
10. **Oldpeak:** oldpeak = ST [Numeric value measured in depression]
11. **ST_Slope:** the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. **HeartDisease:** output class [1: heart disease, 0: Normal]

The task is the analyse all 11 features (1 to 11) and predict the probability of a heart disease. This is a massively non linear problem, hence the Machine Learning methods are suitable for such tasks.

Data Cleaning and Preprocessing

Before cleaning and Pre-processing data, I split it into training and testing sets with 80-20 ratio.

With so many features, we have to make sure that the data is in the correct format. The very first thing to do is to check for missing values. The following code checks for *nan* values.

```
missing = df.isnull().sum()
print(missing)
Age 0
Sex 0
ChestPainType 0
RestingBP 0
Cholesterol 0
FastingBS 0
RestingECG 0
MaxHR 0
ExerciseAngina 0
Oldpeak 0
ST_Slope 0
HeartDisease 0
dtype: int64
```

We can see that the data is quite clean with no missing values. The next task is to convert the categorical values to numerical. Now we can see that all the features are numerical however they're ranging from single digit values to 100s. We have to bring all of these to one scale and for that I used standard scaling on all the features. After this we have data properly scaled, and can proceed to algorithms.

Algorithms till Mid Term

For Mid Term report, I worked on really simple models like Logistic Model with Newton Raphson updates, Probit Model with L-BFGS updates and Bayes Classifier. The overview of those results are discussed below.

Bayes Classifier

I used a basic Bayes Classifier, with all features in the dataset and got an accuracy of 50%. However when I remove the three features (Sex, Oldpeak, ST_Slope), I got an accuracy of 54%.

Logistic Model with Newton Raphson updates

The very first thing I did was to use complete data as it is. All features can be important and I wanted to give the Machine Learning model a chance to learn all the features. Running this test was not successful as I only got the accuracy of 40%. This is understandable because the dataset is massively non linear and we don't have enough parameters in our model to fit the dataset.

To reduce the non linearity, I tried removing three features (Sex, Oldpeak, ST_Slope) which I thought might not be very important. After this, I got an accuracy of 42%. There's an improvement but still a long way to go.

Code can be found here.

Probit Model with L-BFGS updates

I tried the exact same tests with Probit Model and got very similar results. With all features, I got the accuracy of 40.4% and when 3 columns are removed, I got the accuracy of 44.34%. This is showing some improvement, but still I believe more advanced models can help improve the results.

Code can be found here.

More advanced Algorithms

Hamiltonian Monte Carlo with Leapfrog

For this test, I varied ϵ from $\{0.005, 0.1, 0.2, 0.5\}$ and L from $\{10, 20, 50\}$. The results are shown in Table 1. From the table, we can see that this test has been a failure and my reasoning for that is the high dimensionality of the dataset. I believe that we need a more powerful algorithm to fit such non linearity. To test is hypothesis, I tried Bayesian Neural Network next.

ϵ	L	Accuracy	Likelihood	Acceptance Rate
0.005	10	51.74	-0.69	0
0.005	20	51.74	-0.69	0.
0.005	50	51.74	-0.69	0
0.01	10	51.74	-0.69	0.
0.01	20	51.74	-0.69	0.
0.01	50	51.74	-0.69	0.
0.02	10	51.74	-0.69	0.
0.02	20	51.74	-0.69	0.
0.02	50	51.74	-0.69	0.
0.05	10	51.74	-0.69	0
0.05	20	51.74	-0.69	0
0.05	50	51.74	-0.69	0

Table 1: Results Table.

Bayesian Neural Network

From the previous results, we can see that the problem is highly non-linear. Hence a BNN is the perfect tool to handle such cases. For this I used 3 layered BNN with 50 neurons in each layer. I tested both relu and tanh activations and got the following results.

For ReLU, I got the testing accuracy of 77.9 % and plots for predictive likelihoods are shown in Figure 1 and 2

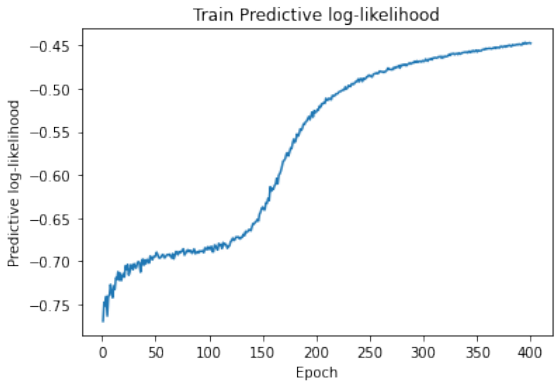

Figure 1: ReLU Training log likelihood.

Similarly for Tanh, I got the testing accuracy of 77.3 % and plots for predictive likelihoods are shown in Figure 3 and 4

Variational logistic regression with EM updates

I tried a lot of different variations, but the best results I got was in this test. After doing this I got the test accuracy of 81% and predictive likelihood 0.72. In this test, I got the mean and civariance matrices as follows:

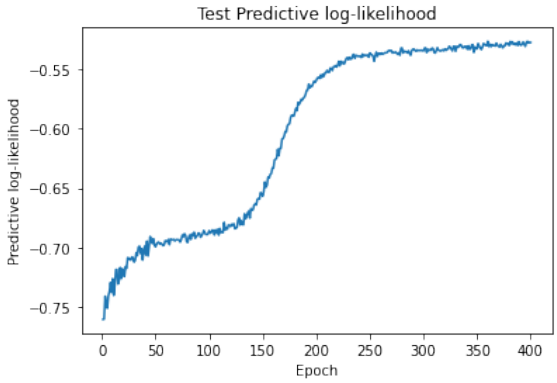

Figure 2: ReLU Testing log likelihood.

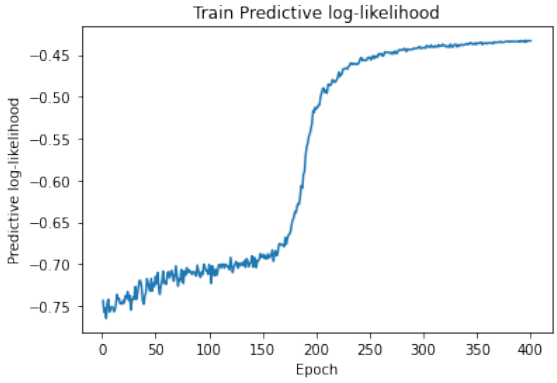

Figure 3: ReLU Training log likelihood.

$$\mu = \begin{pmatrix} 0.23 & 0.75 & 0.02 & -0.46 & 0.40 & 0.11 & -0.33 & 0.86 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 9.53e^{-03} & -3.64e^{-04} & -1.82e^{-03} & 5.20e^{-04} & -1.18e^{-03} & -1.93e^{-03} & 2.77e^{-03} & -1.56e^{-04} \\ -3.91e^{-04} & 8.58e^{-03} & 1.21e^{-04} & 4.97e^{-04} & -7.05e^{-04} & -8.73e^{-04} & 8.69e^{-04} & -1.74e^{-03} \\ -1.82e^{-03} & 1.21e^{-04} & 7.69e^{-03} & -1.37e^{-03} & -5.17e^{-04} & -1.71e^{-04} & 1.67e^{-04} & -9.29e^{-04} \\ 5.20e^{-04} & 4.97e^{-04} & -1.37e^{-03} & 8.87e^{-03} & 2.05e^{-03} & -1.14e^{-03} & -1.81e^{-03} & -8.83e^{-04} \\ -1.18e^{-03} & -7.05e^{-04} & -5.17e^{-04} & 2.05e^{-03} & 8.22e^{-03} & -1.39e^{-04} & -4.34e^{-04} & 4.59e^{-04} \\ -1.93e^{-03} & -8.73e^{-04} & -1.71e^{-04} & -1.14e^{-03} & -1.39e^{-04} & 7.69e^{-03} & -1.08e^{-03} & 8.11e^{-05} \\ 2.779e^{-03} & 8.69e^{-04} & 1.678e^{-04} & -1.81e^{-03} & -4.34e^{-04} & -1.08e^{-03} & 9.92e^{-03} & 2.71e^{-03} \\ -1.56e^{-04} & -1.74e^{-03} & -9.29e^{-04} & -8.83e^{-04} & 4.59e^{-04} & 8.11e^{-05} & 2.71e^{-03} & 8.70e^{-03} \end{pmatrix}$$

Conclusion

1. Normalisation is really important for good results.
2. Simple algorithms at times can't do well on a highly non linear dataset.

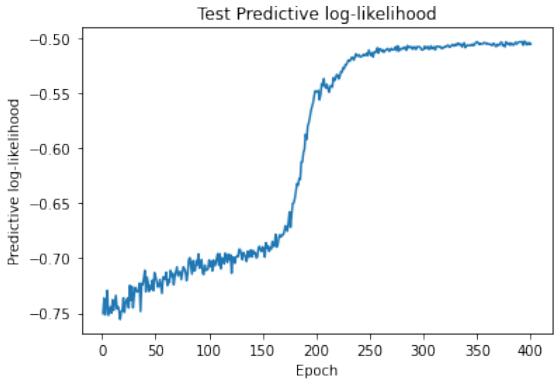

Figure 4: ReLU Testing log likelihood.

3. BNN is a powerful algorithm that can be used widely in all sorts of problems.

4. Variational logistic regression with EM updates was the most effective algorithm for my problem dataset.

References

1. fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

2. Genkin, Alexander, David D. Lewis, and David Madigan. "Large-scale Bayesian logistic regression for text categorization." technometrics 49, no. 3 (2007): 291-304.

3. Jaakkola, Tommi S., and Michael I. Jordan. "A variational approach to Bayesian logistic regression models and their extensions." In Sixth International Workshop on Artificial Intelligence and Statistics, pp. 283-294. PMLR, 1997.

4. Meltzer, Eric B., William T. Barry, Thomas A. D'Amico, Robert D. Davis, Shu S. Lin, Mark W. Onaitis, Lake D. Morrison, Thomas A. Sporn, Mark P. Steele, and Paul W. Noble. "Bayesian probit regression model for the diagnosis of pulmonary fibrosis: proof-of-principle." BMC medical genomics 4, no. 1 (2011): 1-13.