Heart Failure Prediction

GitHub repo

高自在	曾駿馳	顏永明	王柏鈞
N26102131	F74089046	P76101136	N26102050

1. Introduction

> 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 5CVD 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. [1]

The motivation for starting this project is to demonstrate knowledge of artificial learning and data analysis and expound upon model building and training skills that we have learned throughout the course. The main obstacle is data preprocessing which includes missing data imputation, feature selection, and feature scaling.

2. System framework

In this project, our task is to create a model to assess the likelihood of a possible heart disease event. It is a binary classification task in supervised learning. Some common classification models such as decision trees and support vector machines can be applied. In addition to the common models, there are advanced models such as gradient boosting classifiers and deep neural networks. We will construct some of these models and analyze the differences between them.

There are three problems for data preprocessing, as we mentioned above: missing data imputation, feature selection, and feature scaling.

For missing data imputation, Mean/Median/Mode imputation is a common technique to deal with it. KNN imputation is also a considerable option [3].

For feature selection, we will evaluate the correlation between each feature with heatmap and drop unnecessary features if there are any. Besides feature selection, feature extraction is also a technique used for dimensionality reduction, which can be considered as another solution [3].

For feature scaling, standard mean/std scaler and min;max normalization are two standard methods. In addition, Gauss rank transformation, a novel standardization technique with better performance in deep neural networks training, is worth trying in this project [4].

3. Expected results

We expect our model's prediction will have over 60% recall score, which is a best score to look at, since false negatives are unacceptable in medical applications.

Table 1 is the detail of the dataset features [1].

4. Exploratory Data Analysis

The dataset includes 11 features with a total amount of 918 observations. We will do a 90/10 split on it to manually obtain training set and testing set later on. Training set and testing set will contain 825 and 92 observations.

4.1. Data Features

To accomplish our task, we separate data features into numeric features, categorical features and the target feature:

Numeric Features

- Age
- RestingBP
- Cholesterol
- MaxHR

Feature	Description	Value
Age	age of the patient	years
Sex	sex of the patient	M:Male, F:Female
ChestPainType	chest pain type	TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic
RestingBP	resting blood pressure	mm Hg
Cholesterol	serum cholesterol	mm/dl
FastinBS	fasting blood sugar	1: if FastingBS > 120 mg/dl, 0: otherwise
RestingECG	resting electrocardiogram results	Normal: Normal, ST: having ST-T wave abnormality, LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria
MaxHR	maximum heart rate achieved	Numeric value between 60 and 202
ExerciseAngina	exercise-induced angina	Y: Yes, N: No
Oldpeak	oldpeak=ST	Numeric value measured in depression
ST_Slope	the slope of the peak exercise ST segment	Up: upsloping, Flat: flat, Down: downsloping
HeartDisease	output class	1: heart disease, 0: Normal

Table 1. Dataset features [1]

- Oldpeak

Categorical Features

- Sex
- ChestPainType
- FastingBS
- RestingECG
- ExerciseAngin
- ST_Slope

Target Feature

- HeartDisease

4.2. Check Missing Values

First, let's take a glimpse of the dataset. Table 2 shows that there is no null value in the dataset. However, we find that the minimum values of RestingBP and Cholesterol are 0, which is not normal. We assume that they are the missing values filled with 0s incorrectly.

Table 3 shows that there is only 1 row missing value at RestingBP. We simply remove this row.

Figure 1 shows that there are so many missing values at Cholesterol. We replace 0s with NaN by now, will deal with them later.

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

Table 2. Missing values

	Age	Sex	ChestPainType	RestingBP
449	55	M	NAP	0

Table 3. Missing values

4.3. Categorical Features Encoding

For further analysis, we want to encode the categorical features. Before encoding, we separate categorical features into nominal and ordinal features. Nominal variable comprises a finite set of discrete values with no relationship between values. Ordinal variable comprises a finite set of discrete values with a ranked ordering between values.

	Age	Sex	ChestPainType	RestingBP	Cholesterol
293	65	M	ASY	115	0
294	32	M	TA	95	0
295	61	M	ASY	105	0
296	50	M	ASY	145	0
297	57	M	ASY	110	0
...	...	...	...	...	...
514	43	M	ASY	122	0
515	63	M	NAP	130	0
518	48	M	NAP	102	0
535	56	M	ASY	130	0
536	62	M	NAP	133	0

Figure 1. Cholesterol with 0

4.3.1 Nominal Features

- Sex
- ExerciseAngina

Nominal features in this dataset are both binary, so we can simply assign 0 and 1 as Table 4.

Sex		
M	$\longrightarrow$	0
F	$\longrightarrow$	1
ExerciseAngina		
N	$\longrightarrow$	0
Y	$\longrightarrow$	1

Table 4. Encoding Nominal Features

4.3.2 Ordinal Features

- ChestPainType
- RestingECG
- ST_Slope

Ordinal features are encoded as Table 5.

4.4. Relation Observation

After encoding, now we can show the relation between each feature.

Numeric features

Now we want to see the correlation between each numerical features

ChestPainType		
ASY	$\longrightarrow$	0
NAP	$\longrightarrow$	1
ATA	$\longrightarrow$	2
TA	$\longrightarrow$	3
RestingECG		
Normal	$\longrightarrow$	0
ST	$\longrightarrow$	1
LVH	$\longrightarrow$	2
ST_Slope		
Down	$\longrightarrow$	0
Flat	$\longrightarrow$	1
Up	$\longrightarrow$	2

Table 5. Encoding Nominal Features

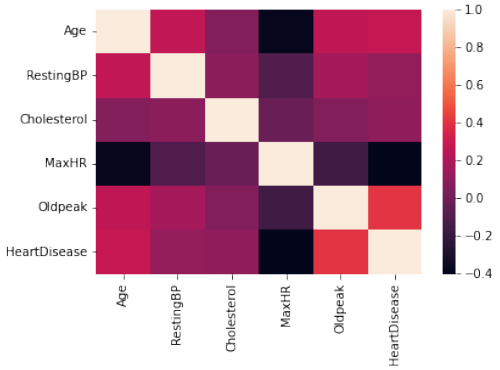


Figure 2. Heat Map

As shown in Figure 2, the correlation between each features are not high, so we don't have to worry about multicollinearity. We can see that OldPeak and MaxHR have relatively high correlation (around ±0.4) with HeartDisease

Categorical Features

	name1	name2	chi_value	chi_p	chi_DF	chi_LLP
0	ST_Slope	HeartDisease	35.988984	1.531410e-08	2	17.994492
0	ChestPainType	HeartDisease	32.254736	4.624851e-07	3	14.586651
0	ExerciseAngina	HeartDisease	15.593001	7.854479e-05	1	9.451841
0	Sex	HeartDisease	7.688718	5.556708e-03	1	5.192749
0	FastingBS	HeartDisease	4.966882	2.583719e-02	1	3.655941
0	RestingECG	HeartDisease	2.990942	2.241430e-01	2	1.495471

Figure 3. Chi-Square test

To see the relation between categorical features and heartdisease, we randomly choose 100 samples from dataset to have a Pearson Chi-Square Independent

Test. In Figure 3, it shows that ST_Slope, ChestPainType and ExerciseAngina are top 3 features that are most likely related to HeartDisease.

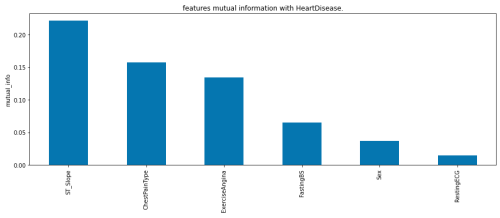


Figure 4. Feature Mutual Information

We use features mutual information to enhance our conjecture in Figure 4.

5. Preprocessing

Before training model, we utilize some methods to preprocess data. The following subsections are introduction about different methods.

5.1. Encoding Categorical Features

We encode the categorical features same as we did in EDA.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	0	2	140	289.0	0	0	172	0	0.0	2	0
1	49	1	1	160	180.0	0	0	156	0	1.0	1	1
2	37	0	2	130	283.0	0	1	98	0	0.0	2	0
3	48	1	0	138	214.0	0	0	108	1	1.5	1	1
4	54	0	1	150	195.0	0	0	122	0	0.0	2	0

Figure 5. Encode Categorical Feature

5.2. Spliting Data

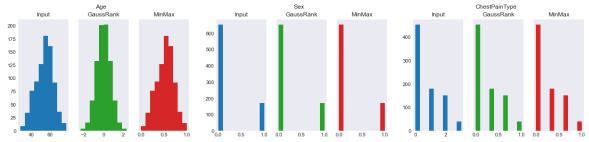
We split our dataset before further preprocessing to avoid any data leakage.

5.3. Missing Data Imputation

After splitting, we can now study how to impute missing values of Cholesterol. We perform the imputation only on the training set to avoid any data leakage as we mentioned above. We choose KNNImputer as our imputer. KNNImputer is an imputation for completing missing values using k-Nearest Neighbors [5].

5.4. Feature Scaling

Standard mean/std scalar and min-max normalization are two common methods for feature scaling. However, there is a new technique called Gauss rank transformation. It is believed that Gauss rank transformation has better performance in deep neural networks training [2]. It should be noted that Gauss rank transformation can only be performed on numeric features, so we perform min-max normalization on categorical features. Figure 6 shows the transformation in different types of features



(a) Numeric (b) Nominal (c) Ordinal

Figure 6. Transformation Comparison
Blue:Original, Green:GaussRank, Red:MinMax

6. Model

6.1. Model Structure

Inspired by residual neural network, we stack 4 fully connection layers that stay same dimension and add previous input every 2 layers. Then we reduce dimension with 2 fully connection layers. Figure 7 shows our model structure.

Model Training

We use Adam(adaptive moment estimation) optimizer to train our model. Since our task is a binary classification problem, binary cross-entropy with logits is the first loss function that comes up with our minds.

Model Performance

Shown in Figure 8, our classification model reaches 89% F1 score with 90% Recall score and 88% Accuracy score.

Model Explanation

By SHAP(SHapley Additive exPlanations), we calculate shap values to analyze the importance of each features. In Figure 9, we can see that Oldpeak and ST_Slope are top 2 contributors to the prediction, which matches our conjecture in EDA.

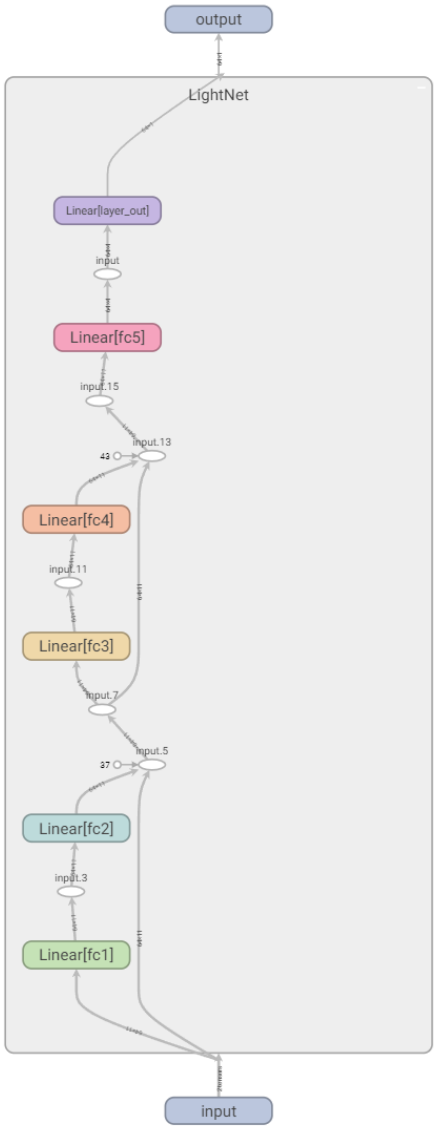
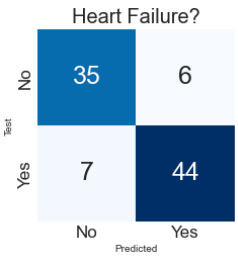


Figure 7. Model Structure



(a) Confusion matrix

Accuracy	0.88
Precision	0.88
Recall	0.90
F1	0.89
ROC AUC	0.88

(b) Result

Figure 8. Model Performance

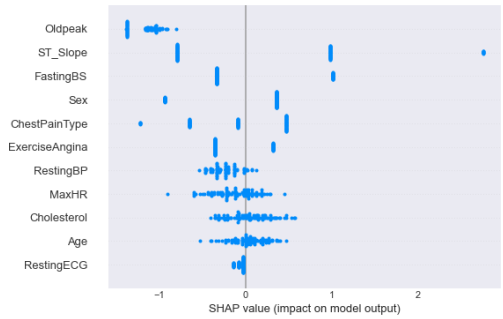


Figure 9. Shap Values

7. Conclusion

We compare different classification models with our model in Table 6. It turns out that our model is slightly better than other models except KNN classifier, which has lower Recall score than ours. It is worth noting that Recall score is an important metric to look at for our task since false negative needs to be avoided in medical applications.

	ours	Logistic Regression	Random Forest	SVC	KNN	Ada Boost
Accuracy	0.88	0.86	0.86	0.88	0.90	0.88
Precision	0.88	0.90	0.91	0.92	0.96	0.93
Recall	0.90	0.84	0.82	0.86	0.86	0.84
F1	0.89	0.87	0.87	0.89	0.91	0.89
ROC AUC	0.88	0.86	0.86	0.88	0.91	0.88

Table 6. Compare to different models

References

[1] fedesoriano. Heart failure prediction dataset. https://www.kaggle.com/fedesoriano/heart-failure-prediction, Sept. 2021. Accessed: 2021-12-16. 1, 2

[2] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 4

[3] JacksonKwong. Hf prediction(98% rec 85% acc 89% f1) with lgbm. https://www.kaggle.com/jacksonkwong/hf-prediction-98-rec-85-acc-89-f1-with-lgbm#3.-Data-Modeling, Dec. 2021. Accessed: 2021-12-16. 1

[4] Jiwei Liu. Gauss rank transformation is 100x faster with rapids and cupy. https://developer.nvidia.com/blog/gauss-rank-transformation-is-100x-faster-with-rapids-and-cupy/, June 2021. Accessed: 2021-12-16. 1

[5] Olga Troyanskaya, Mike Cantor, Gavin Sherlock, Trevor Hastie, Rob Tibshirani, David Botstein, and Russ Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17:520–525, 07 2001. 4