Homework A1 Report: Predicting Heart Disease Presence - Generative v.s. Discriminative Models

Honglin (Leo) Wang
UNI: hw3124
The Fu Foundation SEAS, Columbia University, New York, NY 10034 USA
hw3124@columbia.edu

Abstract

Cardiovascular disease remains a leading cause of mortality globally. Approaches using Machine Learning (ML) to assist in predicting the presence of heart disease has gained attention over the past years. This report focuses on prediction by 2 categories: generative and discriminative ML models about heart disease based on UCI Heart Disease Dataset. Data cleaning or preprocessing, feature engineering are made upon 14 major attributes to build train and test datasets. Naïve Bayesian Classifiers (NB) with Gaussian likelihood and Bernoulli probability distribution are used to build 3 NB models, linear regression models regularized with L2 (Ridge), L1 (Lasso) penalties are also built, with hyperparameters tuned by grid search. Models are evaluated on test set, with 5 main metrics, confusion matrices and ROC curves illustrated. In the end, sensitivity of model performance on hyperparameters are analyzed and regression coefficients in Lasso are interpreted, helping in making intuitive and reasonable predictions.

1 Introduction

Heart disease (also known as Cardiovascular Disease, CVD) stands as a general term for disorders affecting the heart vessels especially coronary arteries, including myocardial infarction, heart failure, arrhythmia, etc. Additionally, Coronary Artery Disease (CAD) is one of the most common subtypes: the reduced blood flow to the heart muscle could cause angina, heart attacks and so. CVD remains as one of the leading factors of mortality worldwide.

Traditional clinical diagnosis involves imaging, auscultation, medical history and non-invasive tests. However, such diagnosis can be very costly. Approaches using Machine Learning (ML) combining some of the non-invasive test results are gaining attention for their high applicability, low cost, and potential of usage as an assistance in clinical diagnosis.

This report aims to present, given the UCI Heart Disease dataset as a benchmark, how 2 main categories of ML models: generative and discriminative models predict whether a patient has heart disease based on medical attributes. The representative models are chosen for both categories: Gaussian and Bernoulli Naïve Bayesian (NB) models for generative models and Linear Regression (LR) for discriminative models. L1 (Lasso) and L2 (Ridge) regularization are used in LR models to prevent overfitting of the training data. Comparison between simple models from both categories are made upon model evaluation on test dataset. How the hyperparameters, e.g. the regularization factor α may affect the model performance are also analyzed.

2 Dataset and Metrics

UCI Heart Disease Dataset is published in American Journal of Cardiology presented in a paper by R. Detrano et al. from UCI. It contains 920 cases in total and 76 attributes are attached to each case. The data were collected from 4 different locations including Cleveland Clinic Foundation, Hungarian Institute of Cardiology, V. A. Medical Center in Long Beach, University Hospital of Zurich in Switzerland, and thus stored as 4 separate datasets, each has slightly different form. Amongst most cases, only given a few of the attributes are given and the rest are NA (not a number, or invalid) values. This possibly led most published experiments on this dataset refer to only a subset of 14 of these attributes.

The author has tried to read the raw .data files and decide which ones of the 76 attributes to use in predicting whether a case has heart disease or not. However, this attempt has failed due to different forms and corrupted rows in the raw files. Therefore, this report only focuses on the 14 attributes that are mostly referred in other published experiments. The 14 attributes are described in **Table 1**.

Table 1 Attributes used to Predict Heart Disease Presence

Column	Type	Description
age	int	Age in years (rounded).
sex	binary	Sex: 0 = female, 1 = male.
cp	category	Chest pain type: 1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic.
trestbps	int	Resting blood pressure in mmHg.
chol	int	Serum cholestoral in mg/dL (rounded).
fbs	binary	Fasting blood sugar > 120 mg/dL: 1 = true, 0 = false.
restecg	category	Resting electrocardiographic results: 0 = normal, 1 = ST-T wave abnormality, 2 = left ventricular hypertrophy.
thalach	int	Maximum heart rate achieved (in exercise, rounded).
exang	binary	Exercise induced angina: 1 = yes, 0 = no.
oldpeak	float	ST depression induced by exercise relative to rest.
slope	category	The slope of the peak exercise ST segment: 1 = upsloping, 2 = flat, 3 = downsloping.
ca	category	Number of major vessels colored by fluoroscopy, ranges from 0 to 3.
thal	category	Thalassemia: 3 = normal, 6 = fixed defect, 7 = reversable defect.
num	binary	Diagnosis of heart disease (have disease or not): 0 = no, $\geqslant 1$ = yes.

Metrics used to evaluate model performance are Accurary, Precision, Recall, F1 score, AUC (area under ROC curve). F1 score is dependent and actually the harmonic mean of Precision and Recall. For Linear Regression models, the predicted probability are taken by clipping values outside the interval [0,1] and 1-MSE (Mean Squared Error) are included as an additional metric. Usually -MSE is taken. We used 1-MSE to map it to the almost same level of the other metrics for our models.

3 Model Selection

All codes and experiments are implemented under the environments as follows: Python 3.12.7, Numpy 1.26.4, Pandas 2.2.2, Sci-kit Learn 1.5.1, Seaborn 0.13.2 and Matplotlib 3.9.2.

3.1 Data Cleaning

Data cleaning is applied after all 920 entries are read from the preprocessed dataset files rather than the raw data. By checking each of the 14 columns, we found few noisy values. There are 0 values in the columns *chol* and *trestbps* which are impossible for living human kind. These 0 values are assigned to NA before we handle NA values. For the *num* attribute, the values range from 0 to 4 while any value above 0 indicates the determined presence of heart disease, hence 1-4 are all reset to 1.

Amounts, proportion (ratio to total 920) of NA values and the method applied to handle them for all 14 attributes are shown in **Table 2**.

Table 2 NAs Counts and Method Applied to Handle NAs

Column	#NA	Ratio(%)	Method
trestbps	60	6.5	Flagging + Imputing
chol	202	**22.0**	Flagging + Imputing
fbs	90	9.8	Flagging + Imputing + Rounding
restecg	2	0.2	Dropping
thalach	55	6.0	Flagging + Imputing
exang	55	6.0	Flagging + Imputing + Rounding
oldpeak	62	6.7	Flagging + Imputin
slope	309	**33.6**	Flagging + Imputing + Rounding
ca	611	**66.4**	Flagging + Imputing + Rounding
thal	486	**52.8**	Flagging + Imputing + Rounding

The dropping method means to drop any case having NA in the corresponding column. We can see that every column except the *restecg* has a proportion greater than or equal to 6%. This large proportion could affect the model's final performance. Hence we decided to create a NA flag for each of them (flagging) and use the iterative imputer to impute values (imputing). For the five imputed columns having the data type categorical or binary, we rounded and clipped the values imputed to the nearest factor level (rounding). The Pearson correlation coefficient matrix is shown in **Fig. 1**, where the coefficients for original dataset are shown above the diagonal line, and coefficients when all NAs are dropped are shown below.

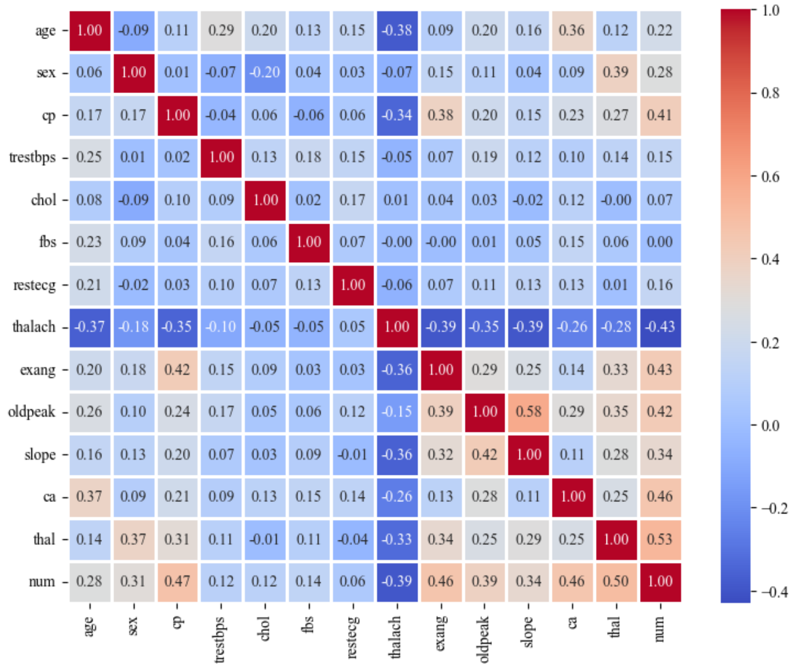

Fig. 1 Pearson Correlation Coefficient Matrix: Above diagonal: Origina; Below diagonal: Dropped NAs.

We can see there are big changes for the coefficients in **Fig. 1** after NAs are dropped simplistically, this impact is huge and we're losing a lot of information when training models. Thus the iterative imputation is worthwhile. For iterative imputed columns we used the mean values as an initial guess and ran 10 iterations of imputation. Usually the mean and mode values are imputed, while iterative imputer imputes the predicted values out of other columns by means of linear regression. This will have a slight impact on our results and also slightly differs from (better than) only using means or models because our model could learn from the NA flags and determine whether the imputed values should be used or not. We can check the distribution of each imputed column and see whether it's twisted. The results for the columns *chol* and *oldpeak* are shown in **Fig. 2**. The *chol* value distribution is more tackled because there are too many NAs, and this is still better than randomly guessing a mode, because it reasons the natural correlations amongst these attributes and more smoothly renders a new distribution (the peak would be higher if simple imputer was used).

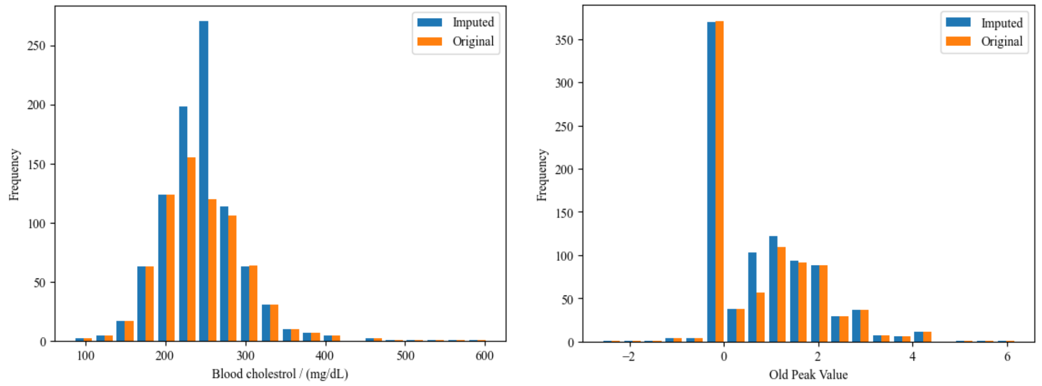

Fig. 2 Distribution of *chol* (left) and *oldpeak* (right) Values

And for *restecg*, i.e. the resting electrocardiographic results, there is only 0.2% of the values are missing, this won't make a big difference in later evaluation, so we've decided to drop the NAs. Plus, the results are not costly to collect from a patient and we can require them to get tested. Finally, we arrived at a dataset with 918 entries.

3.2 Feature Engineering

We use the most common tactic to encode all of these features as inputs of models. For binary columns, we are keeping them unchanged, because 0/1 values are meaningful for showing which class the case are in, i.e. whether sex is male or female. These values are exclusive of each others. For categorical columns, i.e. more than 1 class, we have decided to use the one-hot encoding. In this case, most classes are uncorrelated or not strongly correlated, thus one-hot encoding are enough to reflect on their distributions. And for the numeric features, they are already meaningful. However, their mean and variance tells little about anomaly in health conditions: if you're common, then it's common that you're healthy. So we decided to use standard scaler to normalize these columns. The value normalized reflects how far one person's attribute deviates from the common one, thus could provide more useful knowledge about the presence of heart disease.

Furthermore, we focused on the numeric columns *age*, *chol* and *trestbps*. Researches have shown that all of aging, higher blood cholesterol and higher resting heart rate are key factors in heart disease diagnosis. Thus we decided to use these columns to fit our goal *num* by taking each's logarithm by Wolfram Mathematica. The result obtained is quite interesting: the coefficient of *chol* diminishes and *trestbps* is negative: $num \sim age^{0.59} trestbps^{-0.36}$. Hence, we decided to calculate this and add it as a new feature.

This new feature is not totally dependent on *age* nor *trestbps* by multiplication, and definitely not linearly dependent, thus we may feed it in Naïve Bayesian Decision Models.

3.3 Model Training

The dataset after feature engineering is further split into train and test sets, with 20% in the test. We have used the grid search with 10-fold cross validation on train set to determine the best hyperparameters for all of the models.

Generative Models The generative models used are Gaussian Naïve Bayes (Gaussian NB), Bernoulli Naïve Bayes (Bernoulli NB) and a mixed version of both (MNB).

The Gaussian NB uses a hyperparameter *var_smoothing* (σ^2) that determines the union distribution of all categories or values in one column, and it takes all inputs as numeric ones. The Bernoulli NB uses a hyperparameter *alpha* (α) as to smooth the dataset to avoid 0 category cardinals and reduce the bias, but it takes all positive input values as 1 and negative as 0, so it's working better on features that are already binary or one-hot encoded. Albeit, Bernoulli NB makes sense on the normalized numeric features: it takes 1 if the value is above the average and 0 if not. To combine the advantages of both and discard their shortcomings, a combined version of both MNB is used following the tactic that, Gaussian NB is applied on numeric features and Bernoulli NB on the rest, the predicted probability (likelihood) is multiplied together and then normalized since Gaussian NB returns the probability density.

According to the grid search cross validation results, the best hyperparameters are: $\sigma^2 = 0.05$ for Gaussian NB, $\alpha = 1$ for Bernoulli NB, and a combined $\sigma^2 = 1$ and $\alpha = 1$ for MNB.

Discriminative Models The discriminative models used are regular linear regression model (LR) with no hyperparameters, Lasso (L1 regularized LR) and Ridge (L2 regularized LR).

The LR models have more interpretability than NBs in terms of direct regression coefficient matched with columns. Hyperparameters α for Lasso and Ridge are grid searched and the best is chosen holistically considering all metrics aforementioned in section 2.

According to the results, $\alpha = 0.01$ is chosen for Lasso and $\alpha = 50$ is for Ridge.

All models are again trained on the train set with their best hyperparameters.

4 Model Evaluation

All of the 6 models are evaluated on the previously split test set with 10-fold cross validation. The performance score is then taken the average of cross validation. Their performance in units of % over the 5 main metrics (except 1-MSE) are shown in **Table 3** with maximum displayed in bold text for each metric.

Table 3 Models' Performance

Category	Model	Accuracy	Precision	Recall	F1 score	AUC
Generative	Gaussian NB	80.35	**88.58**	78.64	82.98	**90.98**
	Bernoulli NB	83.25	86.35	86.74	86.36	90.15
	MNB	**83.77**	85.87	88.48	**86.95**	88.92
Discriminative	LR	79.91	82.21	85.76	83.76	88.84
	Ridge	82.13	82.88	**90.23**	86.17	90.66
	Lasso	77.72	79.95	85.76	82.35	88.91

Specifically seen from the table above, Gaussian NB has the lowest Accuracy but highest Precision amongst all NBs, which suggests it tends to overpredict positives with high confidence, but its high AUC manifests it discriminates the sign of heart disease well with appropriate threshold. For Bernoulli NBs, its Precision and Recall are more balanced and this leads to a higher F1 score. Even though it is based on totally binary inputs, it still behaves well, indicating that binary transformations (above/below average) of the numeric features aligned better with true decision boundary, and could be used as new features. The MNB has the highest Recall and F1 amongst all NBs, showing that it identifies even more positives.

The regular LR is slightly worse than most of the models, but its Recall is relatively strong, and it has a reasonable AUC. This might show it tends to overfit the data and could not distinguish well which attribute matters. Meanwhile, Ridge has a strong overall performance, which arguably demonstrating that regularization improves generalization and reduces the variance. The Lasso model has the lowest Accuracy, Precision and F1 score. Though sparsity in coefficients are induced, some features that are informative may have been discarded.

On one hand, generative models generally perform slightly better on Precision than discriminative, potentially due to the relatively small dataset and NB's robustness with fewer parameters. On the other hand, discriminative models with a proper regularization outperform generative ones in aspects of Recall and are competitive in F1 score and AUC.

Confusion Matrix The confusion matrices of all 6 models are shown in **Fig. 3**. It is noticeable that the LR and MNB shares the same confusion matrix. But according to their cross validation tests, MNB shows more generalizability.

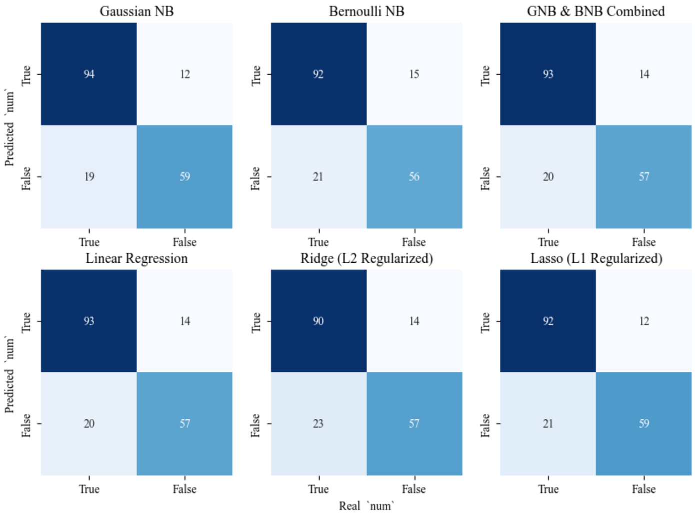

Fig. 3 Confusion Matrices of 6 Models

ROC Curve The ROC curves of all 6 models are shown in **Fig. 4**. The diagonal dashed line shows the worst ROC curve a random classifier could achieve (AUC = 0.50).

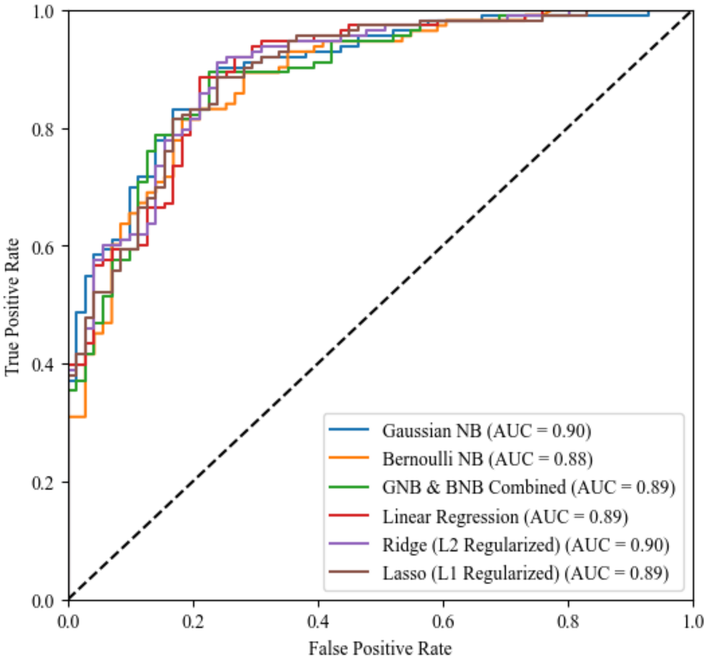

Fig. 4 ROC Curves of 6 Models

5 Discussion

5.1 Sensitivity of Model Performance on Hyperparameters

To further investigate the sensitivity of model performance on hyperparameters, we trained models over different grids of hyperparameters in the range that convergence is confirmed. Then the scores' average and standard deviation are taken in a 10-fold cross validation over the test set. To better illustrate this, only Gaussian and Bernoulli NBs, Ridge and Lasso are chosen because they all have exactly 1 hyperparameter. The MNB has 2 hyperparameters and the its result cannot compare directly to others, thus not considered. **Fig. 5.** shows how hyperparameters affect the models' performance, where the line is the averaged score taken over 10-fold cross validations, and the strip is one standard deviation from the mean.

We can conclude from **Fig. 5**. For Gaussian NB, bad performance coexists with lower variance. This is caused by the spikes generated in the Gaussian NB probability density distribution, the fundamental properties of continuous numeric features are discretized in this way and the probability likelihood could go through sharp change with an oscillation in input. For Bernoulli NB, the performance functions behaves like piecewise functions of smoothing factor alpha, this is because that our dataset is relatively small. Besides, its performance crashes when the alpha is large since it's averaging every class, blurring all the possibilities and underfitting. This indicates we should carefully choose the smoothing factor and vary with different dataset sizes.

For Ridge, the alpha or regularization coefficient has less impact over model performance, and the curve resonates with our previous choice of $\alpha = 50$ in Model Training section: this choice balances over almost all metrics. The same scenario shows up for Lasso. But bad performance happens when the coefficients get pretty large. If the alpha is pretty large, the penalty of having nonzero regression coefficients gets bigger, thus the model will try to zero out all the coefficients, and prone to make prediction $num = 0$ which means negative diagnosis result of heart disease. This is similar to a random guesser that always predicts a patient is not sick at all. This equivalence explains why the every metric is dropping in Ridge and approaching a constant (which is the performance of a no-sickness teller) in Lasso.

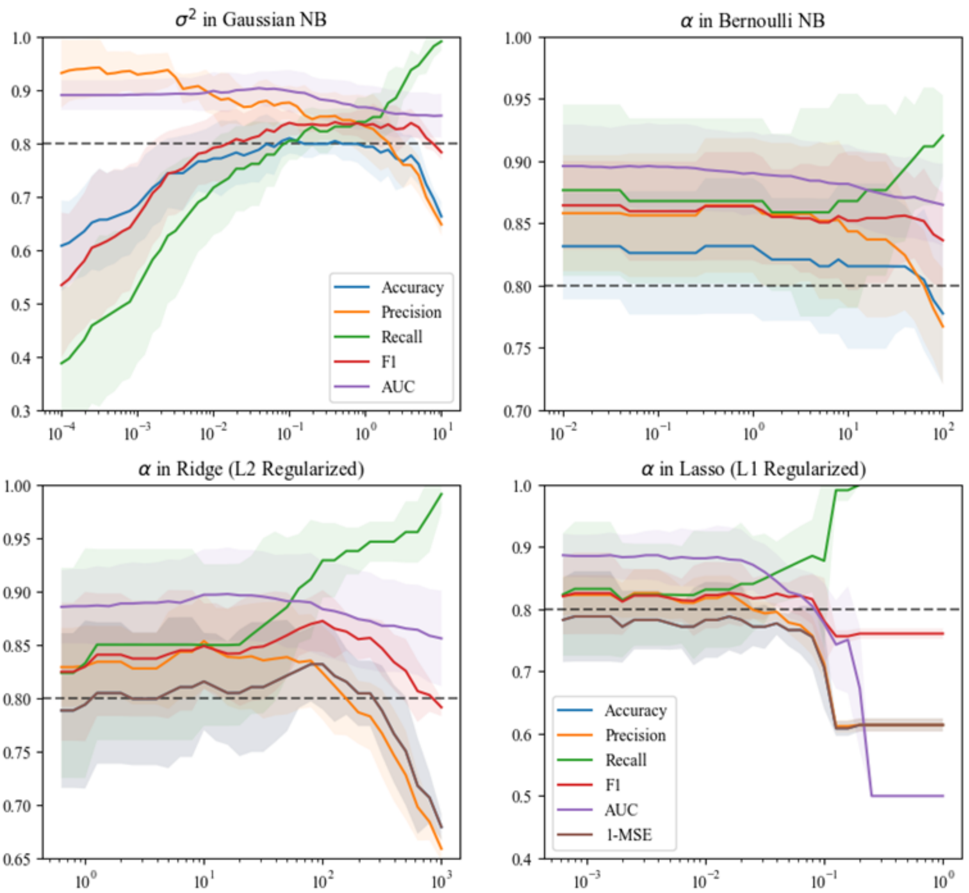

Fig. 5 Model Performance as Functions of Hyperparameters

5.2 Interpretation of Lasso Coefficients

Lasso zeroes out more non-essential coefficients with higher alpha. We have chosen a relatively small alpha for Lasso model when trained, thus many coefficients are still reserved and little sparsity is introduced. Albeit, we can sort the coefficients and focus on the ones with highest magnitudes. All coefficients **multiplied by 100** matched with their column name are listed in **Table 4**. Some of the column is in the form *name* = *value* due to one-hot encoding. @*name* is the NA flag of the column *name*.

Table 4 Lasso Coefficients

Column	Coef.	Column	Coef.	Column	Coef.	Column	Coef.
age	3.078	cp = 1	0	slope = 0	-6.448	@slope	-5.006
trestbps	0.394	cp = 2	-2.807	slope = 1	3.734	@trestbps	0
chol	0.105	cp = 3	0	slope = 2	0	@chol	**10.265**
thalach	-2.805	cp = 4	**22.962**	thal = 3	-0.667	@thalach	0
oldpeak	5.126	restecg = 0	0	thal = 6	0	@oldpeak	0
$\text{age}^{0.59}\text{trestbps}^{-0.38}$	0	restecg = 1	0	thal = 7	6.095	@thal	0
sex	**13.902**	restecg = 2	0	ca	**11.402**	@exang	0
fbs	0	exang	**11.189**	@ca	0.871	@fbs	5.100

We have picked and highlighted 5 coefficients with maximal magnitude. The highest is the one hot encode indicator of *cp* = 4, which means the asymptomatic chest pain is most associated with heart disease. In comparison, *cp* = 2 which is the indicator of atypical angina in chest, is weakly negatively correlated. *Sex* also has a high coefficient, somehow indicating males are more prone to heart disease than females. The column *exang* has a high coefficient as well, showing that the angina induced by exercise is abnormal and reveals the health condition of coronary arteries and blood vessels to some extent.

Some of the results are quite intuitive. For instance, the slightly negative and positive coefficients in *thal* = 3 (normal) and 7 (reversable defect) coincides with intuition that normal in Thalassemia test result reduces the possibility one may have heart disease. The high coefficient of *ca* also reflects the fact that the more number of major vessels colored by fluoroscopy, which suggests blockages or reduced blood flow, the more probable the patient has heart disease.

To our surprise, the *chol* column has low coefficient and its NA flag has a high one. In commonsense, higher blood cholesterol means high risk of heart attack, but several studies (e.g. Dena Zeraatkar *et al*. *Annals of Internal Medicine*, 2019) in recent years have also shown that cholesterols can be categorized into "good" and "bad" ones. One is prone to heart disease only if "bad" cholesterol accumulates in blood.

AI Tool Usage Disclosure

AI Tools Used ChatGPT (OpenAI, GPT-4 Configuration)

AI Contribution ChatGPT are used to suggested the outline of report structure, and provided explanation in the class extension of BaseEstimator.

Personal Contribution The author has written almost all parts of his code on his own. No AI like Copilot was used in code writing. Some parts are inspired by the online documentation and forum posts about Scikit Learn and other package libraries. This report, except for its structure, is all written, typed manually. It cost about 9 hours in total in writing.