Heart Disease Classification with K Nearest Neighbors Classifier

COGS 109 Final Project Report

Qixuan Ma, Yutong Luo, Dandan Liu

Introduction

Every 36 seconds, one person from the United States dies from cardiovascular disease, which sums up to about 655,000 deaths each year, accounting for 1 in every 4 deaths in this country[1]. Amongst the different types of heart disease, coronary heart disease is the most common, killing more than 350,000 people in 2017. The goal of this project is to find out the feasibility of determining the diagnosis of a patient from how similar their attributes are to other known patients. The problem we analyze is important because doctors usually diagnose patients based on their syndromes. If a model can help classify patient syndromes based on their health information with great performance, it can help improve the doctors' diagnosis process and save more lives. Our model is trying to achieve this goal of classification. To simulate this, we will perform a K Nearest Neighbor Classifier on a heart disease dataset[2] with K-Fold Cross Validation to assess its ability to correctly classify patients. After tuning our KNN model, we will also compare its results with some other classification models to conclude our model performance on our dataset.

The dataset we selected was uploaded to Rashik Rahman and is part of a larger heart attack database originally published in the UCI Machine Learning Repository[3]. This dataset is also cited in a few other published academic papers. The dataset contains 303 observations (with no missing values) each with 13 predictors (6 categorical, 6 continuous)[4]:

1. age – age in years
2. sex – sex (1 = male, 0 = female)
3. cp – chest pain type (0 = typical angina; 1 = atypical angina; 2 = non-anginal pain; 3 = asymptomatic)
4. trtbps – resting blood pressure in mmHg
5. chol – serum cholesterol in mg/dl
6. fbs – whether the patient's fasting blood sugar is higher than 120 mg/dl (1 = true, 0 = false)
7. restecg – resting electrocardiographic results (0 = normal; 1 = having ST-T; 2 = hypertrophy)
8. thalachh – maximum heart rate achieved
9. exng – whether the patient experienced exercise induced angina (1 = yes, 0 = no)
10. oldpeak – ST depression induced by exercise relative to rest

---

[1] CDC, Heart Disease Facts. https://www.cdc.gov/heartdisease/facts.htm

[2] Rahman, Rashik. https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

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

[4] Definition of the predictor labels as given by Hamid Reza Marateb and Sobhan Goudarzi in their 2015 paper "A noninvasive method for coronary artery diseases diagnosis using a clinically-interpretable fuzzy rule-based system". https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4468223/

11. slp (discrete) – slope of the peak exercise ST segment (1 = upsloping, 2 = flat, 3 = downsloping)
12. ca – number of major vessels colored by fluoroscopy
13. thal – thal rate (0 = normal, 1 = fixed defect, 2 = reversible defect)
14. num – outcome variable, diagnosis of heart disease (0 = <50% diameter narrowing, 1 = >50% diameter narrowing)

By building a K Nearest Neighbors classifier, we will attempt to predict the outcome variable to address the hypothesis that we can predict a patient's binary label (whether or not they are prone to heart attack) based on the similarities of their features to other patients with known diagnosis.

Methods

We are focusing on prediction. We want to predict if one patient will be prone to a heart attack or not based on one's physical condition. Our choice of data analysis is the K Nearest Neighbors classifier. We use this method due to two reasons. First, we want to classify patients based on their similarity in syndromes, and KNN will be an appropriate classifier to approach this goal. KNN will assign a label based on the labels of k nearest neighbors, which exactly meets our needs to classify based on feature similarity. Second, based on our data analysis, we do not see a linear correlation between features and output. As we can see in the correlation plot below, the maximum correlation between feature and output is -0.44, which does not show a strong linear relationship. KNN supports non-linear solutions and can handle our dataset well.

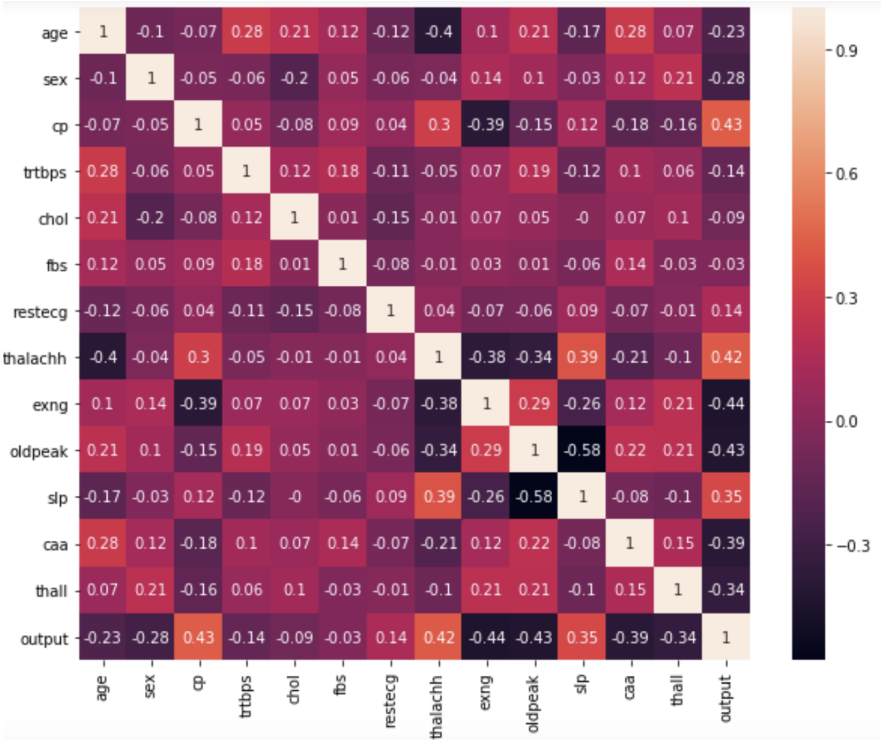

Figure 1. Correlation table for predictors in this dataset

We will include all features to fit our model. Because all the physical information is important for diagnosing heart attacks in the real world, we do want to lose some trivial details due to not including some features. This is a matter of life.

As mentioned above, the dataset contains 303 observations and 13 predictors, 6 of which are categorical variables. From there, we performed a hyperparameter search with GridSearchCV for the KNeighborsClassifier from the sklearn package with k = 10 to determine the best set of hyperparameters to use for this specific dataset. We found that 'auto' algorithm, 16 leaf size, power parameter value of 1, and weight function of 'distance' (weight points by the inverse of their distance).

We utilized K-fold Cross-Validation as part of our analysis, which follows the following procedure:

1. Shuffle the dataset randomly. By doing so we can randomly assign each observation to one of the folds.
2. Split the dataset into K folds
3. Then we repeat the following procedure on each fold for K rounds:
    a. Take one fold as testing data set
    b. Take the remaining folds as a training data set
    c. Fit a model on the training set and evaluate it on the testing set
    d. Record the model evaluations for both training and testing set

More details about our K-fold CV is in the next part, as we will illustrate this concept in the context of model selection.

Results

Following the logic in the Methods part, we used a 10-fold cross-validation. We choose K=10 here because it can properly split our data into train and test sets with a size of 272:31, and K = 10 is a value that has been found through experimentation to generally result in a model skill estimate with low bias a modest variance[5]. We obtain the average accuracy, f1 score, and area under ROC (AUC) for a range of k values from 1 to 50 to find the optimal value of k that will maximize the performance of the model (in this case performance is measured by the sum of accuracy, f1 score, and AUC). We choose the three evaluation metric for the following considerations:

1. Accuracy: it represents the percent of observations, both positive and negative, that are correctly classified. It is the most intuitive performance metric for a classification model, which is the easiest to be implemented in doctor-patient communications.
2. F1-score: it is the harmonic mean of precision and recall, and is commonly used in binary classification when we focus more on positive results. This fits our problem because those who are prone to heart attacks will be more of significance.

[5] https://machinelearningmastery.com/k-fold-cross-validation/

3. ROC(AUC): it measures the tradeoff between true positive rate (TPR) and false positive rate (FPR). With higher AUC, we can have a better model at distinguishing between patients with heart attacks and no disease.

To reduce the variability of the model results due to randomness, we ran the 10-fold cross validation 5 times to obtain a more replicable result. After plotting the performance of the classifier with different k values over 5 repetitions, we can see 2 peaks around k = 30 and k = 40 (Figure 2). Overall, our model performed best for k = 38, which we will use to evaluate the performance of the classifier in other metrics.

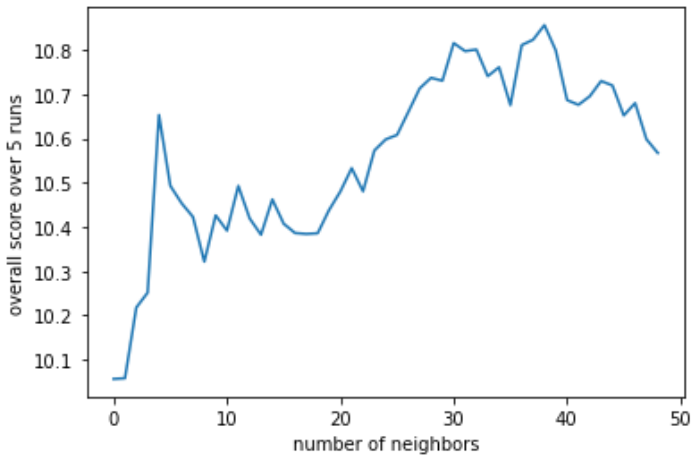

Figure 2. KNN classification performance for k values from 1 to 50

We then fitted the model using k = 38 and achieved a final accuracy of 70%, f1-score of 74% and ROC of 70% across the full dataset. We then plotted the confusion matrix of true labels versus predicted labels, as shown below.

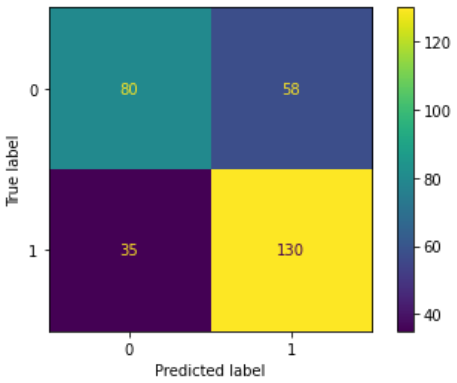

Figure 3. Confusion matrix of KNN classification

Based on our model evaluation metrics, the KNN Classifier does not yield satisfactory classification results. To better understand our model estimation, we also train different classification models and derive their metrics for comparison. We will further illustrate our model results and comparisons in the Discussion section.

Discussion and Conclusion

It is clear that our model does not do a good enough job of classifying patients with heart disease and patients that do not. On one hand, our three evaluation metrics all got around 70% and are not strong indicators for good performance. On the other hand, we have a relatively large number of False Positive and False Negative (Figure 3). These metrics are important to our question of diagnosing disease, because the former case will lead to people paying for unnecessary treatments, and the latter is more serious that patients with disease will not receive treatment. However, the number of people who do have heart disease but are predicted as not having heart disease (false negative) is lower than the number of people who do not have heart disease but are predicted to have heart disease (false positive), which is desirable because false positives could lead to preventative care for heart disease that are not severe enough to be life-threatening.

In this dataset, the threshold for a positive label (having heart disease) is determined to be >50% narrowing (stenosis) of at least one of the 3 major heart arteries, which is a conclusion made from animal experiments. However, according to P. J. Harris et al. in their 1980 paper "The Prognostic Significance of 50% Coronary Stenosis in Medically Treated Patients with Coronary Artery Disease", "coronary flow is generally not affected until the degree of stenosis approaches 75%" and for that reason, he recommended classifying patients according to the number of 75% or greater stenosed vessels[6]. Therefore, we hope that false positive results could push the patients in the right direction and stop or slow the progression of the stenosis. In future studies, it is worth considering the balance between false negatives and false positives. It is important for doctors to give correct diagnosis for their patients, especially for those who have the disease. Not recognizing their illness will prevent them from getting appropriate medical treatments, yielding serious results or even deaths.

	Precision	Recall	F1-Score
KNN	0.69	0.69	0.69
SVM	0.65	0.65	0.63
RF	0.83	0.83	0.83

[6] Harris PJ, Behar VS, Conley MJ, Harrell FE Jr, Lee KL, Peter RH, Kong Y, Rosati RA. The prognostic significance of 50% coronary stenosis in medically treated patients with coronary artery disease. Circulation. 1980 Aug;62(2):240-8. doi: 10.1161/01.cir.62.2.240. PMID: 7397965.

MLP	0.83	0.83	0.82
LOGREG	0.86	0.86	0.86
NB	0.82	0.82	0.82

Table 1. Comparison with sklearn models

Lastly, we would like to recenter our attention to the performance of our KNN model as opposed to some of the other classifiers from the sklearn library as well as more advanced approaches taken to classify this particular dataset. In Table 1 above, we selected a few of the models we tried from the sklearn library.

While our KNN model performed better than support vector machines, it fails short against all other models, including random forest, multilayer perceptron, logistic regression, and Naive Bayes. It is worth mentioning that other models were not tuned through hyperparameter search, so their optimal performance might be higher.

Compared with previous attempts to classify this dataset, the performance of our KNN model is even lower. In 2015, Hamid Reza Marateb and Sobhan Gourdarzi proposed a model which used a combination of multiple logistic regression and neuro-fuzzy classifier and achieved an accuracy of 84%[7]. A year earlier, Mahmoodabadi and Saniee Abadeh used an imperialist competitive algorithm based fuzzy expert system and achieved an accuracy of 94.92%.

This is sufficient to see that classifying a patient's prognosis based on the similarities between them and other already diagnosed patients is not very accurate, which could be attributed to a few reasons. First of all, the KNN classifier is based on the assumption that observations that are close together should have the same label. Our model does not take into account the "weights" of the different predictors, which could explain some of the performance discrepancies between our KNN and other sklearn library algorithms. This problem could be addressed by using a weighted KNN classifier. Moreover, as suggested by P. J. Harris et al., the number of stenosis arteries has an important effect on the progression of the disease as well. Measuring the degree of stenosis in each of the arteries could lead to higher accuracy across the board. Lastly, apart from biomedical features, it is known that other social and cultural factors such as income and race are significant indicators of whether a patient has heart disease[8]. Access to that information could also yield better results in similarity-based classification.

Appendix

The Python notebook and dataset for this project can be found in this Github repository: https://github.com/Harrison-Q-Ma/COGS109_Final_Project

[7] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4468223/

[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4541436/