Heart Disease Prediction Challenge

1st Zichong Wang
dept. Computer Science
Stevens Institute of Technology
Hoboken, USA
zichonglwang@gmail.com

***Abstract*—This project is forced on heart disease prevention. I build and compare the four different Machine Learning models: Logistic Regression, K-nearest neighbors, Decision Tree, and Support Vector Machines. The project data set dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them.**

***Index Terms*—heart disease, KNN, machine learning, logistic regression, decision tree, Support Vector Machines, EDA**

I. Introduction

According to the 2019 Global Health Estimates report released by WHO, heart disease has been the leading cause of death worldwide for the past 20 years. More people are dying from heart disease than ever before. Since 2000, heart disease deaths have increased by more than 2 million, rising to nearly 9 million in 2019. Heart disease now accounts for 16 percent of all causes of death. This dataset collected different indicators of physical health. I try to build Machine Learning (ML) models for heart disease prevention. Doctors can use the machine learning model results to make better judgments about their patients' conditions.

The target for this task is binary classification. Hence, this paper focuses on four ML models: Logistic Regression, K-nearest neighbors, Decision Tree, and Support Vector Machines. I will compare each model's advantages and disadvantages and explain the reason.

Overall, I evaluate holistic ML models and give recommendations for the models that should be used for different datasets.

II. Dataset & Problem Description

A. Properties of the dataset

The dataset contains a total of 1025 rows of data and 14 features. All data types are integers except ST depression induced by exercise relative to rest. The distribution of the data is shown in the figure below:

B. Miss value & Duplicate value & Balance

First, I check if missing values exist or duplicate data in the dataset. I use the Pandas library, and there is no missing value in the dataset. However, there exist 723 duplicate data in the dataset. The repetition rate of up to 70.47%. In other words, the dataset only has 303 unique data.

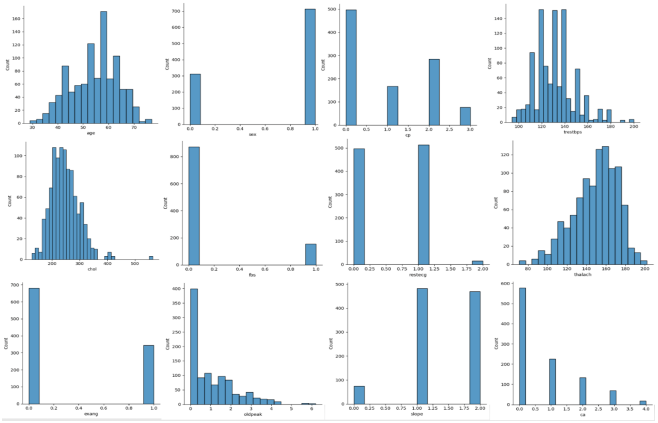

Figure 1: The distribution of the data

The dataset is balanced. There are 499 rows marked with 0(48.68%) and 526 rows marked with 1(51.32%) in the dataset. For the features, there are all balances except the fasting blood sugar(fbs).

C. Data Correlation

Second, I check the correlation. The top three correlations with the target are chest pain type(43.49%), maximum heart rate achieved(42.29%), and the slope of the peak exercise ST segment(34.55%).

I also check the correlation between different features. I need to focus on the correlation between various features because I need to consider whether I need to do a principal component analysis. If the correlation between two features is very high, I can consider merging them to reduce the feature dimension to improve the ML model's efficiency. The correlation between different features shown in the figure below:

III. Experiment Setup

A. Partition of training set and test set

There is very little unique data based on our data set, so I set up the unique and original data sets. Both data sets are divided by 80% training and 20% test sets.

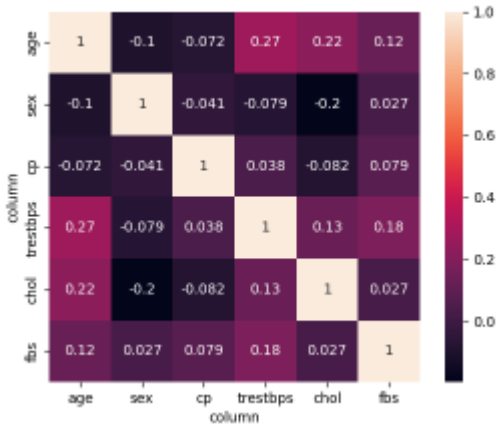

Figure 2: Correlation between different features 1

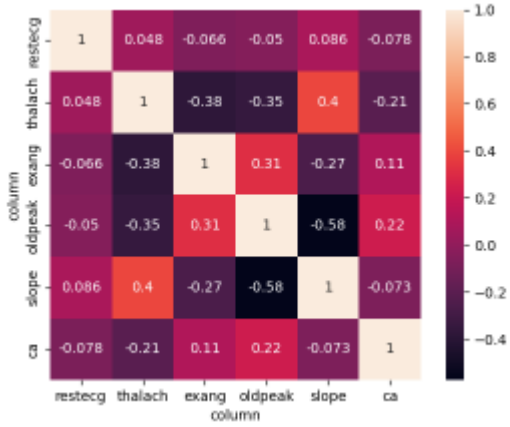

Figure 3: Correlation between different features 2

B. Exploratory Data Analysis

I do different Exploratory data analysis for different machine learning algorithms.

- For the Logistic Regression, I considered reducing the number of features to prevent overfitting. However, the datast sample size and the number of features is only thirteen. I did not cut the dataset.
- For the KNN algorithm, I normalize the data and plan the overall data to be between 0 and 1, thus eliminating the effect of different ranges of values for specific feature data. And I also test the accuracy of the different k values.
- For the CART decision tree, the model can effectively control overfitting by increasing the value of min samples split and min samples leaf or decreasing the value of max features and max-leaf nodes.
- I convert the feature to a standard normal distribution for the SVM learning. Normalizing the data will let SVM faster convergence during parameter optimization. It also prevents the classifier's performance from being affected by the significant difference in the size of different feature values in the dataset.

IV. Model Description

A. Logistic Regression

The first ML model I use the logistic regression. As we know, the essence of logistic regression is linear regression. The difference from linear regression is that logistic regression adds a Sigmod function mapping layer to the mapping of features to results. In other words, logistic regression first linearly sums the features and then uses the Sigmoid function to convert it into a probability solution. If the probability is greater than 0.5, it is classified as the first class. Otherwise, it is divided into the second class.

The advantage of using Logistic Regression: The data set I used has less features, and logistic regression would perform well. And logistic regression output category follows a Bernoulli binomial distribution which means logistic regression outputs have probabilistic claims. As the model is used, there will be a lot of new data in the future that can help the logistic regression model improve.

The disadvantage of using Logistic Regression: The data set I used has small number of data. Even if in the original data set, it just have 1025 rows. Therefore, logistic regression is very easy to overfitting. And logistic regression is a linear classifier, so it can't handle the case of correlation between features.

B. K-nearest neighbors

In the second ML model, I use the K-nearest neighbors(KNN). As we know, the essence of KNN is When the data and labels in the training set are known, the test data are input. The features of the test data are compared with the corresponding features in the training set. The top K data in the training set most similar to them are found, then the category corresponding to this test data is the one with the most occurrences among the number of K data.

The advantage of using KNN: First of all, KNN is not sensitive to outliers. And KNN is particularly suitable for multi-classification task. And KNN is too computationally intensive, especially when the number of features is enormous. Each data to be classified needs to calculate its distance to all known samples to get its K^{th} nearest neighbor. However, the data set I used has a small number of data. Therefore, it won't be lots of costs.

The disadvantage of using KNN: KNN will have low prediction accuracy for rare classes when the samples are unbalanced. When the training data set are unbalanced, such as when the sample size of one type is large, and the sample size of other types is small, it is possible that when a new test data is input, the sample of the large-capacity class predominates among the K^{th} neighbors of that sample. And the data set I used is not too balanced. It will decrease the accuracy. And KNN's dependence on the training data is enormous, and the fault tolerance of the training data is too poor. If there exist error data in the training data set, and there are close to the test point that needs to be classified, it will lead to inaccuracy of the predicted data.

C. Decision Tree

In the third ML model, I use the Decision Tree. As we know, decision tree is a tree structure, each internal node represents a judgment on an attribute, each branch represents the output of a judgment result, and finally each leaf node represents a classification result. The commonly used decision trees are ID3, C4.5, and CART. The classification effect of CART is generally better than other decision trees. The task is to classify this time. Hence, I use the CART decision tree.

The advantage of using Decision Tree: The CART decision tree does not require data pre-processing and normalization or handling missing values. CART decision trees can take both discrete and continuous values and are fault-tolerant for outliers.

The disadvantage of using Decision Tree: The cart decision tree algorithm is easy to overfitting, and the cart decision tree will cause a drastic change in the tree structure due to a bit of change in the training data set. And if the proportion of sample with one label in the training set is too large, the generated CART decision tree tends to be biased towards this label.

D. Support Vector Machines

In the last ML model, I use the support vector machines(SVM). The basic idea of SVM learning is to solve for the hyperplane that splits the training data set with the maximizes the geometric separation.

The advantage of using support vector machines: SVM can solve nonlinear problems, and SVM has no local minimum problem, and can handle high-dimensional data sets well.

The disadvantage of using support vector machines is that SVM is not very efficient when the size of the training data set is large. Because SVM solves support vectors with the help of convex quadratic programming and solving convex quadratic programming will involve the computation of a matrix of order m (m is the number of samples), the storage and analysis of this matrix will require high memory and cost huge computing time when the number of m is large. The SVM algorithm is an algorithm for solving binary classification problems and cannot handle multi-classification issues. But it does not affect this task.

V. Results & Discussion

A. Logistic Regression

The result of Logistic Regression. For original datasets, the training accuracy is 84.49%, and the test accuracy is 79.61%. For unique datasets, the training accuracy is 84.50%, and the test accuracy is 87.10%.

The accuracy of the test set of the unique dataset is higher than the accuracy of the training set of the unique dataset because the amount of data in the unique dataset is too small, which leads to overfitting.

B. K-nearest neighbors

First, I test the different k values to see how they affect accuracy. And also set original datasets and unique datasets. The result of K-nearest neighbors is shown in the figure below:

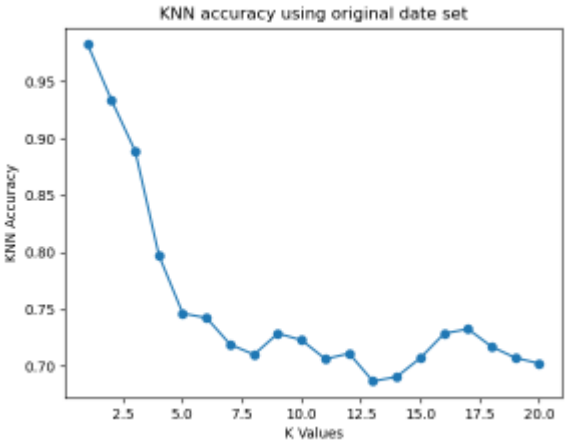

Figure 4: KNN accuracy using original date set

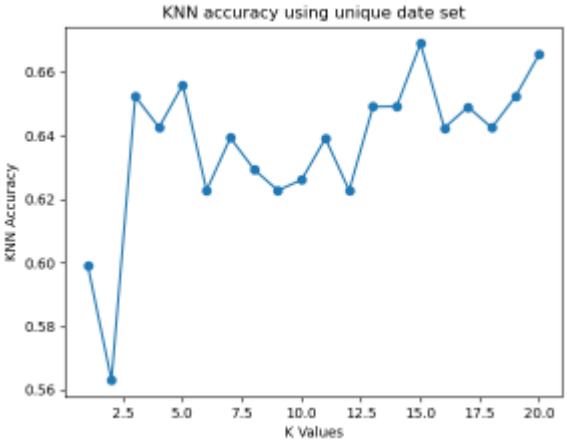

Figure 5: KNN accuracy using unique date set

I set k equal to 5. The test accuracy for the original dataset is 74.6%, the test accuracy for the broken original dataset is 74.7%, and the test accuracy for the normalized broken original dataset is 84.0%. The test accuracy for the unique dataset is 65.6%, the test accuracy for the broken unique dataset is 64.9%, and the test accuracy for the normalized broken unique dataset is 82.1%.

For the original dataset, the reason the k value increase is increase followed by test accuracy decrease is because there are so many duplicate data in the original data set. Hence, if the k value is large, then it is possible that the result just votes by the same point which means the bias is increased followed by the k values increase.

For the unique dataset, as the K value becomes smaller, the deviation will become smaller, and the variance will become larger, which is easy to overfitting and easily affected by noise. As the value of K increases, the deviation will become larger, and the variance will become smaller, which is easy to underfitting. Therefore, I set k equal to 5 to test normalizing the data.

After normalizing the data, the test accuracy of the model has improved significantly over that before normalized. The

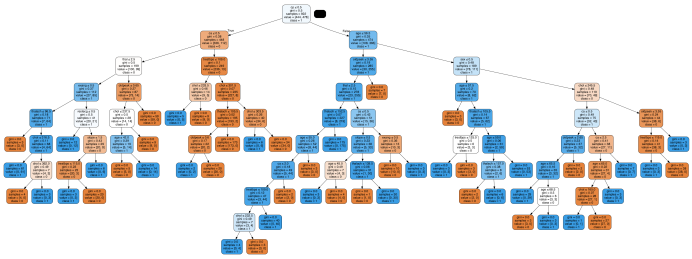

Figure 6: Decision Tree using original date set

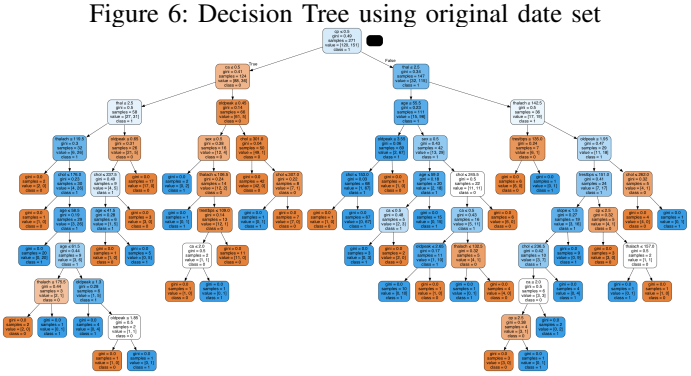

Figure 7: Decision Tree using unique date set

reason is because of model calculates Euclidean distance. Large values of features will affect the results more.

C. Decision Tree

For the Decision Tree, the test accuracy for the original dataset is 100% and the test accuracy for the unique dataset is 80.65%. The decision tree image shown in the figure below: The test accuracy for the original dataset is 100% because there are so many duplicate data in the original data set. Therefore, the data used for testing is also trained. Hence, the model can find the perfect path, resulting in a 100% prediction accuracy.

D. Support Vector Machines

For the Support Vector Machines, after normalizing the data, the test accuracy for the original dataset is 100% and the test accuracy for the unique dataset is 75.41%.

The reason that test accuracy for the original dataset is 100% is sane as decision tree. Because there are so many duplicate data in the original data set. Therefore, the data used for testing is also trained. Hence, the model can find the perfect position, resulting in a 100% prediction accuracy.

VI. Conclusion & Future Work

A. Conclusion

For this task, I use four different ML models to build hear disease prevention system. The result as Table 1 shows.

As the table shows, those four ML models have good accuracy in the original dataset. The CART Decision Tree and Support Vector Machines both get 100% accuracy. However, it is because might the test sample also be training. Hence, it is a fake result. And for K-nearest neighbors, it is a small variance dataset because the dataset is balanced and has lots of duplicate data. Hence, the accuracy is higher than Logistic Regression.

TABLE I
Test Accuracy

ML models	Original Dataset	Unique Dataset
Logistic Regression	79.61%	87.10%
K-nearest neighbors	84.0%	82.10%
CART Decision Tree	100.00%	80.65%
Support Vector Machines	100.00%	75.41%

The Logistic Regression has the highest test accuracy rate in the unique dataset. Logistic regression has good performance because the data set is balanced, and the size and features are small. KNN has the second-highest accuracy rate. Because I did Min-Max Normalization, each feature has the same weight when calculating the Euclidean distance. Also, it is a binary classification task, and the dataset is balanced. Therefore, KNN can also have good performance. CART Decision Tree has the third-highest accuracy rate. Because the CART decision tree ignores the correlation between the features. However, there is a correlation between these physical body indicators. Therefore, its performance is not as good as the previous two models. SVM should have good performance for small-sample, nonlinear binary classification tasks. There might have some issues in the normalization.

B. Future Work

In future work, based on the size of this dataset was small. The first thing needs to do is increase the number of the training sample. There are two ways, the first is to collect more patient data, but it needs more time. The other way is to build the shadow model. The shadow model can be used to generate the simulation data. However, the feasibility needs more study. There are non-negligible deviations between simulated and patient data.

The second thing is to implement ensemble learning. The basic idea of the ensemble learning algorithm is to combine multiple classifiers to achieve an ensemble classifier with a better prediction effect. In this task, I implement and compare a single ML model. Although the test accuracy is not bad, based on the sample size is small. If there are more data in the future, it is necessary to consider using ensemble learning.