Mobile Applications for Cardiovascular Disease Detection

Boyu Shen
bshen27@wisc.edu

Shuhan Zhang
szhang633@wisc.edu

Yinghan Hu
yhu344@wisc.edu

Abstract

Cardiovascular diseases have been the major public health concern. Prediction of cardiovascular diseases can help early diagnose and control the process of this disease. We hope to find an optimal model that can be applied to mobile applications to help earlier detection for people globally. In this study, a predictive model will be developed to diagnose the risk of developing cardiovascular disease based on information offered by users. Our dataset consists of 70,000 records and 11 features. Several machine learning algorithms were used to build predictive models including Logistic Regression, k-Nearest Neighbors, Decision Tree and Random Forest algorithms. For algorithm selection, 5 x 2 Nested Cross Validation will be applied to choose the best algorithm based on average test accuracy. K-nearest neighbors and Decision Tree present to have close generalization performance which are higher than logistic regression and random forest. For hyperparameter tuning and feature selection, we integrated sequential feature selector into 10-fold cross-validation via grid search to get the best model setting. The result shows that the decision tree model with maximum depth at 7 and gini impurity reached the highest accuracy at 72.38% on validation dataset with 8 features except age, height and weight. Moreover, the result from model evaluation shows that the optimal decision tree model outperformed other proposed models. Therefore, we decided to apply it on mobile applications for cardiovascular disease detection.

1. Introduction

WHO's statistical data shows that around 17.9 million people die from cardiovascular diseases (CVDs) annually, accounts for 31% of all global deaths.[3] Therefore, CVDs have become the world's top leading cause of death. In daily lives, we can also hear that many of our elderly relatives are troubled by CVDs which made us worried. Heart attacks or strokes as main symptoms of cardiovascular diseases can cause irreversible damage to patients. Early treatment and medication can effectively reduce the number of deaths caused by cardiovascular diseases and control the process of disease development. Our motivation is to develop models can be applied to mobile phone apps to predict people's risk of getting cardiovascular disease, providing global people with easy access to assess their health condition toward cardiovascular disease.

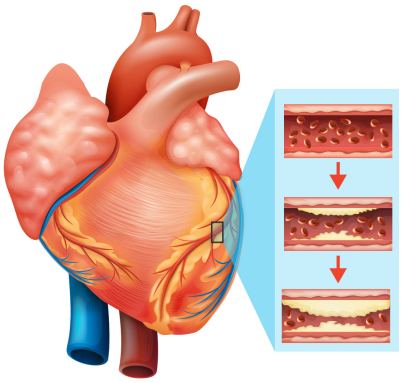

Figure 1. Effect of high cholesterol level on cardiovascular

Studies have shown that unhealthy diet, obesity, tobacco use, harmful use of alcohol and physical inactivity are highly associated with risk of getting cardiovascular diseases. Our dataset consists of 70,000 records of patients data and 11 input features. Input features can be separated into 3 types: objective features, examination features and subjective features. For objective features, it includes basic genetic information of patients including Age, height, weight and gender. Age is counted in days for high precision. The subjective features are self-assessments on unhealthy life habits including smoking, inactive physics and drinking. The objective features and subjective features are easy to collect and suitable for motivation of applying to apps on phones. However, examination features include level of cholesterol, level of glucose, diastolic blood pressure and systolic blood pressure need results of medical examination. Feature selection will be further discussed since there is a tradeoff between precision of model and original motivation. The target is labeled as presence of cardiovascular disease and absence of cardiovascular disease.

The whole dataset will be divided into 80% train dataset and 20% test dataset. Experiment was conducted to find the most precise method to predict the risk of CVDs on

patients, algorithms include logistic regression algorithm, k-nearest neighbors algorithm, decision tree algorithm and random forest algorithms are fitted with 5 x 2 nested cross validation. After choosing the best algorithm, two relatively well performed algorithms are selected to tune hyperparameters with grid search. For the evaluation part, we evaluated the model with best estimators on the independent test set, computed confusion matrix and F1 score.

2. Related Work

There are many professional and clinical studies conducted related to cardiovascular disease prediction. Aqeel Ahmed and Shaikh Abdul Hanan (2012) in their study combined 4 datasets of 920 records with 72 features to predict heart disease by using different data mining techniques[2]. Only 13 features were then considered significant and used further in their study, which are Age, Sex, P, Trstbps, Chol, Fbs, estecg, Thalach, Exang, OldPeak, Slope, Ca, Thal and Num. The study reported that Decision Tree and SVM estimators were the most effective in predicting heart disease, which yields a 92% accuracy. Besides, Mohammad Shafenoor Amin et al (2019) used 7 data mining algorithms including k-NN, Decision Tree, Naive Bayes, Logistic Regression, Vote, Support Vector Machine and Neural Network with 10 fold cross-validation and a selection of 12 features to create the classification model for the diagnosis of CVDs[5]. Similarly with Aqeel and Shaikh's study, except age and sex, other 10 features in this study are all professional clinical attributes such as hest pain type, maximum heart rate achieved and the slope of the peak exercise ST segment, which has to be collected from accurate medical examinations. Among the tested 7 algorithms, Vote, Naive Bayes and SVM outperformed the other algorithms based on their average generalization accuracy and average precision obtained from the experiments, and the best model has a generalization accuracy of 78.2%[1].

From a review of Cardiovascular Disease Prediction Using Data Mining Techniques, Kirmani made an assumption of Accuracy = Algorithms Used*Dataset used + Number of valid Attributes. Moreover, they stated that some algorithms presented the highest accuracy of diagnosis while taking a greater number of features into prediction. Therefore, although there are many great clinical trials conducted related to the prediction of cardiovascular disease which have fairly high prediction accuracy, we would expect to train a relatively parsimonious model which could also yield predictive and useful results based on easily accessible individual inputs for our intention of developing a convenient mobile application.

3. Proposed Method

3.1. k-Nearest Neighbors

We start with k-Nearest Neighbors algorithm. kNN algorithm is a micro-benchmark for comparison with other advanced algorithms. The kNN algorithm categorizes a data point depending on a plurality voting of its k neighbors. Distance function would be used to compute the distance so that we can find the k nearest neighbors around the target. We proposed to use this algorithm here as it could provide one way to do classification of whether a person having cardiovascular disease or not. There are two common ways to determine the distance, which are Euclidean distance function and Manhattan distance function as described below.

$$Euclidean : d(x^{[a]}, x^{[b]}) = \sqrt{\sum_{i=1}^{N}(x_i^{[a]} - x_i^{[b]})^2}$$

$$Manhattan : d(x^{[a]}, x^{[b]}) = \sum_{i=1}^{N} |(x_i^{[a]} - x_i^{[b]})|$$

where i $\neq j$,x$^{[a]}$ and $x^{[b]}$ are two different data points in an n-dimensional space.

Note that data point x stands for a patient sample with its target label as presence/absence of cardiovascular diseases. The algorithm of kNN picks the first k neighbors of query data point p. The predicted label of query point is determined by computing a plurality voting on selected k neighbors. The detailed of hyperparameter tunning is in experiment part.[7]

3.2. Decision Tree

From lecture, we know that decision tree algorithms can be considered as an iterative, top-down construction method for the hypothesis.It's a flowchart-like structure where each internal node indicates a test on one specific feature of whether a case happens or not, and each leaf node represents the class label or a regression output after passing all the way down from the tests. As this project is aimed at predicting the presence or absence of cardiovascular disease, we proposed to try decision tree method as it could give us some insight about how the feature could be used in predicting the disease.

The way how the tree grows depends on how the node splits. The tree determines a feature and the condition on this feature for the test in each internal node to generate the largest information gain. And, the tree stops growing when all child nodes are pure, containing only one type of class labels. Information gain mentioned above is a basic criterion for splitting the tree. The greater the information gain, the better the splitting is. It could be calculated through this

equation listed below.

$$Gain(D, x_j) = H(D) - \sum_{v \in Values(x_j)} H(D_v)\frac{|D_v|}{|D|}$$

Where D is the dataset at parent node, and D_v is the dataset at child node. $H(D)$ is commonly calculated through two method, one is Shannon Entropy and the other is Gini Impurity (Equation listed below).[8]

$$Entropy : H(p) = -\sum_i p_i \log_2(p_i)$$

$$Gini : 1 - \sum_i ((p_i)^2)$$

3.3. Random Forest

Since using Bagging as an ensemble methods could yield a less variant result than one individual estimator and is less prone to overfitting, we decided to try Random Forest algorithm after Decision Tree. Random Forest algorithm basically utilize the idea of Bagging and random feature subsets. Each node is given random feature subsets instead of the complete feature set. Based on bagging strategy, we use different bootstrap samples to fit each estimator,along with a random subset of features was selected at each node to choose the optimal split, which reduces the correlation between each tree estimators but increases the weakness of each estimator.

3.4. Logistic Regression

A binary class Logistic Regression model is chosen to fit apart from using the other three non-parametric algorithms. Probabilities are used to refer the likelihood of whether a label would be selected in the binary class Logistic Regression model. Below is the probability function.

$$Pr\left(y^{[i]} = c\right) = \frac{exp\left(\beta_c x^{[i]}\right)}{\sum_{k=1}^{K} exp\left(\beta_k x^{[i]}\right)}$$

where $y^{[i]}$ represents predicted result of the ith patient, c represents absence or presence of the cardiovascular diseases, β_c represents weights of the model of a specific class, $x^{[i]}$ represents a vector of features of patient i and k represents the number of classes to classify, which is 2 here.

We also applied L1 regularization of weights to the Logistic Regression model to prevent the model from overfitting. A regularization term with a coefficient λ is added to the loss function. This term is an L1 norm of the weights. The best setting of hyperparameters(λ) is determined by using Cross Validation methods.

4. Experiments

4.1. Dataset

The dataset is obtained from kaggle, consisting of 11 columns and 70,000 records of patients data with a binary label of presence or absence of cardiovascular diseases. The 11 input features include id, four objective features (Age, Height, Weight,gender), three subjective features(smoking, alcohol intake, physical activitity), four examination features (systolic blood pressure, Diastolic blood pressure, cholesterol, glucose). While some of the features are easy to collect, medical examination features are harder to measure. Our experiment takes this into consideration and performed feature selections, details in 4.5.

For the data cleaning part, we first checked whether our data is balanced. 50.03% of the targets are absence and 49.97% are presence which means our data does not have imbalanced problems. Then we deleted abnormal values of the features: height <130 cm and height >220 cm, weight <30 kg and height >180 kg, Systolic blood pressure >250 mmHg and Systolic blood pressure <70 mmHg, Diastolic blood pressure >200 mmHg and Diastolic blood pressure <40 mmHg. We also dropped duplicates in the data. Since the feature "id" is a simple sequence and may not provide helpful information to us, we dropped the column "id".

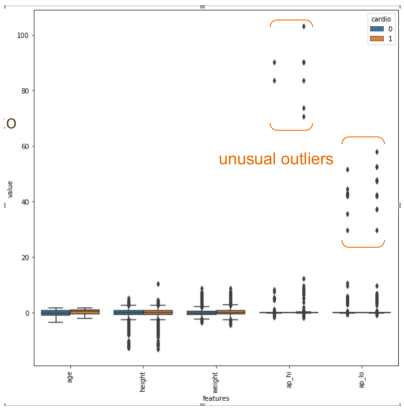

Figure 2. Box Plots of features

After data cleaning, we split the dataset into two separate parts - for training and testing. We apply stratified sampling to our dataset to maintain the original class proportion in resulting subsets, using the function train test split from scikit-learn. The training set consists of 80% of the images, while the test set contains the rest 20%.

4.2. Software

The softerware we used was python through Jupytor notebook with numpy, pandas Scikit, mlxtend and matplotlib packages. For data preprocess, pandas and numpy were used. For algorithm selection, feature selection and

model training, we used packages Scikit and mlxtend.[6] For visualization, package matplotlib was used.

4.3. Hardware

All the computations were executed on laptop CPUs.The computational hardware we used are group members' laptops, which are three Macbook pro, each with 16G memory.

4.4. Algorithm Selection

To choose the best algorithm, we used 5 x 2 Nested Cross Validation via *GridSearchCV* method that compares the four algorithms mentioned above, k-Nearest Neighbors, Logistic Regression, Decision Tree and Random Forest, with hyperparameter settings shown in the Table 1. After defining the setting for each algorithm classifier, we carry out the 2-fold cross validation in inner loop to select the best model setting for each algorithm classifier, and then we do the 5-fold cross validation in outer loop to generate the average test accuracy as a measure for the generalization performance of each algorithm classifier. [4]

Algorithms	Hyperparameter Values
Logistic Regression	Penalty: 12, Inverse of regularization strength (C): 10 to the power from -4 to 4
KNN	Number of neighbors: 10 to 150 by interval 10, Distance measure method(p): {1=Manhattan 2=Euclidean}
Decision Tree	Maximum depth: {1,2,3,4,5,6,7,8,9,10}, Criterion for splitting: {Gini, Entropy}
Random Forest	Number of estimators: {10, 100, 500, 1000, 10000}

Table 1. Hyper-parameters Settings of Algorithms

Based on the results listed in the table 2, we chose Decision Tree and k-Nearest Neighbors algorithm for further hyperparameter tuning and model selection, as they have higher generalization accuracy compared to Logistic Regression and Random Forest. One thing worth noticing is that kNN has larger variance than any other three algorithms depite its highest test accuracy, implying its instability.

4.5. Feature Selection

As mentioned in 4.1, some features require a lot more investment than others to measure, so we want to check whether those features have a crucial impact on our model and worth the investment. In 4.4, we finally chose Decision Tree and k-Nearest Neighbors, so we would proceed with these two models.

Algorithms	Outer Test Accuracy
kNN	72.11 ± 0.45
Decision Tree	72.09 ± 0.22
Random Forest	71.61 ± 0.31
Logistics Regression	71.60 ± 0.20

Table 2. Algorithm Selection Result

For Decision Tree model, we use a pipeline to combine these Decision Tree classifier and Sequential Feature Selector as one estimator, and pass it into *GridSearchCV* method to perform a 5-fold cross validation. This is to determine the best hyperparameter setting for Decision Tree hyperparameter maximum depth and Splitting Criterion for Decision Tree and best feature sets for Sequential Feature Selector. We searched the best maximum depth values within the range of integer 6 to 9 and splitting criterion from gini or entropy by taking reference from the inner loop cross validation result from nested cross validation in section 4.4. Because the step sizes of parameter lists we tried in Algorithm Selection are relatively large, we run the algorithm this time with narrower range and smaller step sizes. After parameters are selected, we test the model on the test dataset and thereby obtain the final test score.

The same procedure were carried out on k-Nearest Neighbor algorithm as well. The only two differences are that we tuned the hyperparameter value of number of neighbors and carried out a 10-fold cross validation for kNN.

5. Results and Discussion

After combining the sequential feature selector and kNN classifier together to do the 10 fold cross validation for model selection, it gives us the best hyperparameter setting that the number of neighbors is 43 and the best feature set is 10 features out of 11. The validation accuracy of the best setting model is 0.68492 and its generalization accuracy is 0.68268. For Decision Tree, after the similar procedure as done in kNN, we gain its best hyperparameter setting is 7 as the maximum depth while using 'gini' as the criterion for developing the tree, and the best feature set is 8 features out of 11. The validation accuracy for the best model setting is 0.723821, and its generalization accuracy is 0.7292. As this project is aimed at developing a mobile application to detect a typical type of medical disease, we then further examine our model by observing the learning curve, confusion matrix and calculating the F1 score.

5.1. k-Nearest Neighbors

The best kNN model reaches the generalization accuracy of 0.68268 on the test set. The *GridSearchCV* which carries

out 10-fold cross validation shows that the optimal choice of k is 43, and the optimal feature set is 10 original features excluding the feature Weight. Here we presents the learning curve, confusion matrix and the F1 score table of this optimal kNN model.

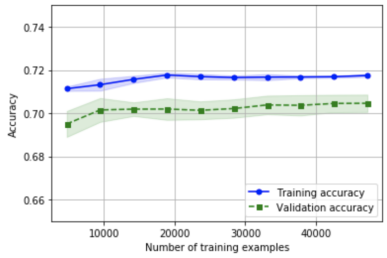

Figure 3. Learning curve of kNN model with k=43

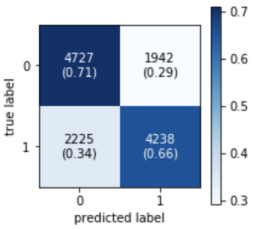

Figure 4. Confusion Matrix of kNN model with k=43

Precision	Recall	F1 Score
0.685761	0.655733	0.670411

Table 3. F1 score table for kNN model with k=43

From the learning curve, the gap between training accuracy and the test accuracy is around 1%, which is quite small and shows that our model doesn't suffer from overfitting (low variance). Also, as the desired accuracy is around 70% based on other related work, we can see from the curve that this model is generally not biased as its accuracy is around 70%. The generalization performance gets improved with larger training set size as expected, but one interesting thing about training accuracy is that it also gets slightly improved with larger training size. From the confusion matrix and F1 score table, the precision is calculated as 0.685761, the recall is calculated as 0.655733, and the F1 score is calculated as 0.670411.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2\frac{Recall \times Precision}{Recall + Precision}$$

5.2. Decision Tree

The best Decision Tree model reaches the generalization accuracy of 0.7292 on the test set. The *GridSearchCV* which carries out 5-fold cross validation shows that the optimal choice of maximum depth is 7 while using 'gini' as the criteria for information gain, and the optimal feature set is 8 original features excluding the feature Age, Height, Weight. Here presents the learning curve, confusion matrix and the F1 score table of this optimal Decision Tree model.

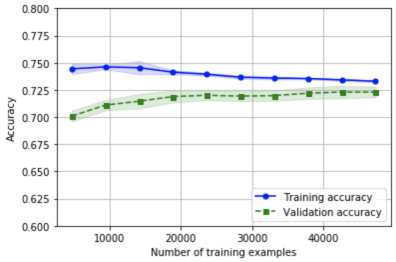

Figure 5. Learning curve of the optimal Decision Tree model

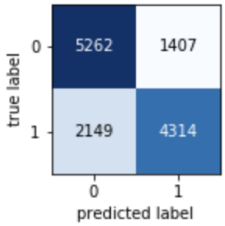

Figure 6. Confusion Matrix of of the optimal Decision Tree model

Precision	Recall	F1 Score
0.754064	0.667492	0.708142

Table 4. F1 score table for the optimal Decision Tree model

From the learning curve, the gap between training accuracy and the test accuracy decreases with the increasing size of training set and becomes quite small when training size is larger than 40000, which shows an improvement of generalization capacity. As our model deals with around 55000 training samples, it doesn't suffer from overfitting issue. Also, as the desired accuracy is around 70% based on the related work, we can see from the curve that this model is generally not biased as its accuracy is around 72.5%. From the confusion matrix and F1 score table, the precision is calculated as 0.754064, the recall is calculated as 0.667492, and the F1 score is calculated as 0.708142.

5.3. Disucssion

For the purpose of model selection and evaluation, we observed that Decision Tree model has higher F1 Score,

	Decision Tree	kNN
F1 score	0.708142	0.670411
Precision	0.754064	0.685761
Recall	0.667492	0.655733
Test Accuracy	0.7292	0.68268
Number of Features	8	10

Table 5. Comparison between Decision Tree and kNN

precision, recall, test accuracy than kNN. Higher precision indicates lower false positive rate, and higher recall also indicates lower false negative rate. In other words, Decision Tree model has a larger chance of correctly predicting the disease as well as a smaller chance to not detect the potential of this disease, which is really crucial to any medical disease's detection. Also, higher test accuracy shows better generalization capacity of Decision Tree model over kNN. Thus, we decided to apply Decision Tree into the final stage of developing the mobile application. One thing worth mentioning is that the generalization performance of the kNN model (0.68268) varies a lot from the result generated in nested cross validation (0.7211). One possible reason could be that we didn't try a wide range of hyperparameter values or other parameter candidates when doing hyperparameter tuning, due to the expensive Grid Search method and the limited time. So we may not find the true optimal hyperparameter setting for kNN.

For the purpose of feature selection, we finally decided to use 8 features which are Gender, Systolic blood pressure, Diastolic blood pressure, Cholesterol, Smoking, Alcohol intake, Glucose and Physical Activity. We noticed from both models that features like Glucose, Cholesterol, and Blood Pressure are important to the predicting results and cannot be excluded from consideration. Thus, although they may not be easy to collect , they are worth the efforts to obtain. Moreover, the smaller size of the feature set of Decision Tree could be useful for developing the mobile application as it requires less information from users so that it could improve user experience.

6. Conclusions

In this project, we first proposed four algorithms to detect Cardiovascular disease which are Random Forest, KNN, Decision Tree and Logistic Regression. After doing algorithm selection via nested cross validation and model selection via k-fold cross validation, we discovered the optimal Decision Tree model which has the best prediction capacity. We also showed that this model is powerful as it requires less user inputs but yields better predicting results. Thus, we believe the Decision Tree model is applicable in developing a mobile application which could predict the potential of having Cardiovascular. Referring back to the related work in section 2, we can see that our model doesn't have the same high prediction accuracy. This may result from the differences between the feature set being used, as professional and clinical indices used in their studies could be more informative in predicting cardiovascular disease but they are less accessible which may cost money and time. If we are going to continue this project, future improvement could be trying some other machine learning models such as Naive Bayes, Support Vector Machine etc. When doing algorithm selection and model selection, we could try using F1 score as the standard instead of accuracy. Also, we could search more thoroughly for the hyperparameter optimization of KNN model. In the end, we truly hope our model could be integrated into a useful mobile application to help people detect their potential of having Cardiovascular disease conveniently.

7. Acknowledgements

The dataset was downloaded from Kaggle at Cardiovascular Disease Detection. Part of the coding took reference from STAT451 Machine learning lecture.

References

[1] Ahmed, A., Hannan, S. A. (2012). Data Mining Techniques to Find Out Heart Diseases: An Overview. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 1(4), 18–23.

[2] Amin, M. S., Chiam, Y. K., Varathan, K. D. (2019). Identification of significant features and data mining techniques in predicting heart disease. Telematics and Informatics, 36(0736–5853), 82–93. https://doi.org/10.1016/j.tele.2018.11.007.

[3] Cardiovascular diseases (CVDs). (2017, May 17). World Health Organization.https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[4] Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, 10(7), 1895–1923. https://doi.org/10.1162/089976698300017197

[5] Kirmani M. M. Cardiovascular Disease Prediction Using Data Mining Techniques: A Review. Orient.J. Comp. Sci. and Technol;10(2).

[6] Raschka, S. (2014). Home - mlxtend. Http://Rasbt.Github.Io/Mlxtend/.

[7] Raschka, S. (2020). Lecture 02: KNN. STAT451 Lecture Notes.

[8] Raschka, S. (2020). Lecture 06: Decision Tree. STAT451 Lecture Notes.