UCI Heart Disease Prediction Using Supervised Learning Techniques

Jingyi Mao

Department of Mathematics,
Wilfrid Laurier University,
Waterloo, Ontario, Canada
Email: maox0490@mylaurier.ca

April 8, 2019

Abstract: Heart disease is a common circulatory disease. The circulatory system is composed of the heart, blood vessels and the neurohumoral tissue that regulates blood circulation. Circulatory disorders are also known as cardiovascular diseases. Since heart disease is one of the significant health problems nowadays, it attracts researchers looking for understanding the crucial factors that causing a heart disease based on the current health condition. In this project, we are going to explore and compare the performance of five different supervised learning techniques, including Logistic Regression, K-Nearest Neighbour (KNN), Naïve Bayes, Decision Tree, and Support Vector Machine (SVM) to automate the heart disease prediction systems using the UCI heart disease dataset. The analysis aims to help the physician for heart disease detection based on the characteristics of patients such as gender, age, blood pressure, serum cholesterol, etc.

Table of Contents

1. Introduction

1.1. **Background.**

Supervised learning algorithms is a machine learning task that establishes a mathematical model based on a collection of data that contains both the explanatory variables and the desired outputs[1]. It suggests a function from labelled training data consisting of a set of training examples. Each training example has one or more input variables and the desired output. In the mathematical model, the training data is usually represented by a matrix which consist of all the input variables in each column. Through the learning from given training data, supervised learning algorithms can predict the output based on the new inputs by iterative optimize an objective function[2]. An optimal function will help the algorithm to correctly determine the class label that were not part of the training data set.

Supervised learning algorithms include classification and regression[3]. Classification methods are used when the outputs are limited to a specific set of values, and regression algorithms are used when the responses are numeric values. The similarity learning is an area of supervised learning closely related to regression and classification. However, the intention is to learn from training examples using a similarity function that measures how similar or related two objects are.

According to Public Health Agency of Canada, heart disease is the 2^{nd} leading cause of death among Canadians, and about 1 in 12 Canadian adults live with diagnosed heart disease[4]. Since heart disease is mostly diagnosed based on the physicians knowledge and experience, this requires a massive amount of time and work before the final decision can be made. Therefore, an effective and efficient automated heart disease prediction system is not only beneficial in the healthcare sector for heart disease detection but also reduces workload for the specialist. In this project, we used five popular supervised learning algorithms to design a decision support system for heart disease; meanwhile, it reduces the costs tremendously for medical tests for the patients.

1.2. **Data Source.**

In the age of big data, lots of data are publicly available, especially in the healthcare sector. However, not all the data are mined to help people discovering the hidden patterns and making decision. Advanced data mining techniques are used for knowledge discovery in the database and medical research, particularly in heart disease prediction.

UCI Machine Learning Repository[5] provides the dataset we used in this project, and it is publicly available at *"https://archive.ics.uci.edu/ml/datasets/Heart+Disease"* that you can download from the website. In this dataset, we were given 303 patients' information with their corresponding labels of whether the heart disease is presence or absence. For

each patient, we have their age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels, and thalassemia information.

1.3. **Data Dictionary.**

Before starting our analysis and applying supervised learning algorithms, we firstly need to take a close look at our data and understanding the meaning and representations of each variables. Table 1 shows a summary of all the variables, descriptions, valid values, and attribute types within the dataset.

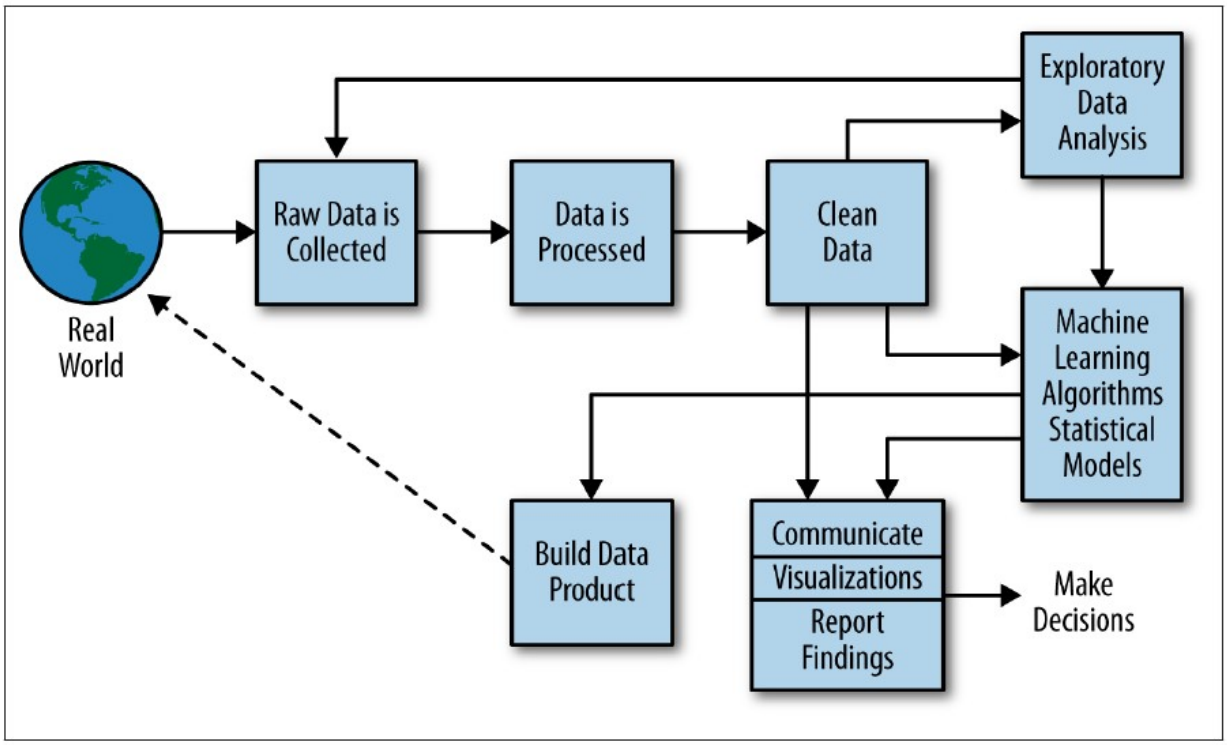
TABLE 1. Data Dictionary

Variable	**Description**	**Valid Values**	**Attribute Type**
age	Age in years	29 - 77	Numeric
sex	Sex	1 = male, 0 = female	Binary
cp	Chest pain type	0 = typical angina, 1 = atypical angina, 2 = non-anginal pain, 3 = asymptomatic	Nominal
trestbps	Resting blood pressure in mm Hg on admission to the hospital	94 - 200	Numeric
chol	Serum cholestoral in mg/dl	126 - 564	Numeric
fbs	Fasting blood sugar > 120 mg/dl	1 = true, 0 = false	Binary
restecg	Resting electrocardiographic results	0 = normal, 1 = having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria	Nominal
thalach	Maximum heart rate achieved	71 - 202	Numeric
exang	Exercise induced angina	1 = yes, 0 = no	Binary
oldpeak	ST depression induced by exercise relative to rest	0 - 6.2	Numeric
slope	The slope of the peak exercise ST segment	0 = upsloping, 1 = flat, 2 = downsloping	Nominal
ca	Number of major vessels	0, 1, 2, 3, 4	Nominal
thal	Thalassemia	0, 1, 2, 3	Nominal
target	Diagnosis of heart disease	1 = presence, 0 = absence	Binary

2. Data Prepossessing

From the data dictionary, we can observe that there are three different kinds of attribute types in our dataset. They are numeric, nominal, and binary. Since we have both qualitative and quantitative data, we can not treat them equally, and this requires certain data prepossessing steps before we can actually fit in the data to machine learning models. Figure 1 shows the general process of data science according to Rachel Schutt and Cathy ONeil[6]. In this section, we will conduct the basic explanatory data analysis(section 2.1), creating dummy variables(section 2.2), normalization(section 2.3), and train test split(section 2.4) for our data.

Figure 1. The data science process



2.1. **Explanatory data analysis.**

The Explanatory Data Analysis(EDA) entails making plots and building intuition form the dataset. EDA helps out a lot, as well as trial and error and iteration. In statistics, EDA is an approach to analyzing data and summarize their main characteristics. Then a statistical model can be questioned whether it can be used or not. However, EDA is primarily for seeing what the data can tell us beyond the formal modelling or hypothesis testing task. In Table 2, it shows the summary statistics for our UCI heart disease dataset.

TABLE 2. Summary Statistics

Attributes	**count**	**mean**	**std**	**min**	**25%**	**median**	**75%**	**max**
age	303	54.37	9.08	29	47.5	55	61	77
sex	303	0.68	0.47	0	0	1	1	1
cp	303	0.97	1.03	0	0	1	2	3
trestbps	303	131.62	17.54	94	120	130	140	200
chol	303	246.26	51.83	126	211	240	274.5	564
fbs	303	0.15	0.36	0	0	0	0	1
restecg	303	0.53	0.53	0	0	1	1	2
thalach	303	149.65	22.91	71	133.5	153	166	202
exang	303	0.33	0.47	0	0	0	1	1
oldpeak	303	1.04	1.16	0	0	0.8	1.6	6.2
slope	303	1.4	0.62	0	1	1	2	2
ca	303	0.73	1.02	0	0	0	1	4
thal	303	2.31	0.61	0	2	2	3	3
target	303	0.54	0.5	0	0	1	1	1

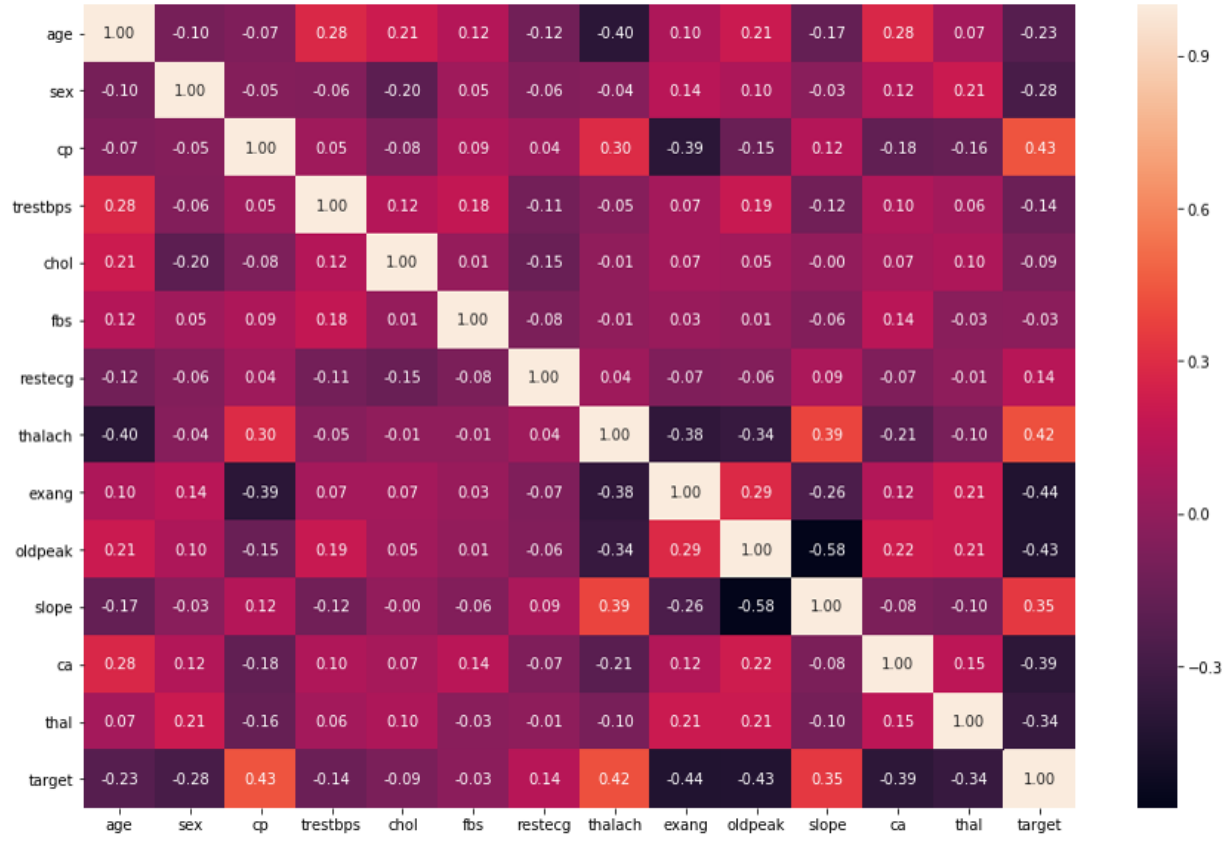
Next, we look at the correlations between each of our variables. All correlation values between the data are listed in Figure 2. As a result of this listing, it is aimed to ensure that these properties are used in different places by performing different operations. Thus, the p-value process determines a hypothesis and a hypothesis thesis is presented between each characteristic according to this hypothesis. In this process, after determining the class property as a hypothesis, the relations between all the other properties are checked. This results in a different number for each property. What is important here is that these numbers are not close to 1 or -1. If the number is close to 1, that means two variables are positively high correlated; in contrast, -1 means two variables are negatively high correlated. For both cases, they are very similar to each other in some sense. In a simple example, if variable A is increasing, then variable B is increasing; or if A is increasing, then B is decreasing. These are particularly inadequate, and it may have a negative impact while building a model. Therefore, one of the highly correlated inputs is usually get removed.

As we can conclude from Figure 2, none of the pairs in our dataset are highly correlated or closed to 1 or -1. The most highly correlated variables are the slope of the peak exercise and ST depression induced by exercise relative to rest which is -0.58. However, this is acceptable, and we are competent to move on with further analysis.

2.2. **Creating dummy variables.**

Because of over half of our attributes are categorical variables, we can not just ignore them. Therefore, a dummy variable is needed which takes the value 0 or 1 to indicate the absence or presence of any categorical effect that may be suspected to shift the outcome. In other words, the dummy variables perform like a “power switch” that turn various parameters on and off

FIGURE 2. Correlation between variables



in an equation. Another advantage of the dummy variable is that even for the nominal data, we can treat it statistically like an interval-level variable. For example, if we take an average of a 0, 1 variables, the result is the proportion of all the 1's in the distribution. Figure 3 shows the sample output for the first six rows from our dataset after creating the dummy variables.

FIGURE 3. First six rows after creating dummy variables

	age	sex	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	ca	...	cp_1	cp_2	cp_3	thal_0	thal_1	thal_2	thal_3	slope_0	slope_1	slope_2
0	63	1	145	233	1	0	150	0	2.3	0	...	0	0	1	0	1	0	0	1	0	0
1	37	1	130	250	0	1	187	0	3.5	0	...	0	1	0	0	0	1	0	1	0	0
2	41	0	130	204	0	0	172	0	1.4	0	...	1	0	0	0	0	1	0	0	0	1
3	56	1	120	236	0	1	178	0	0.8	0	...	1	0	0	0	0	1	0	0	0	1
4	57	0	120	354	0	1	163	1	0.6	0	...	0	0	0	0	0	1	0	0	0	1
5	57	1	140	192	0	1	148	0	0.4	0	...	0	0	0	0	1	0	0	0	1	0

2.3. **Normalization.**

In this section, we introduce the normalization techniques that can present all of the attributes on the same scale. The reason we need to normalize our data is that for some of the algorithm such as KNN which we will talk about later, requires the distance calculation. Therefore, we need to guarantee that all the variables are measured in the trick. For example, it makes no sense if we calculate the distance between one measured in centimetres and another one measured in meters, and this will result in an inaccurate measurement of the distance between variables and therefore misleading the algorithms when we build the model.

There are quite few standardize and normalize techniques such as min-max normalization, Z-score, decimal scaling, and so on. However, we are not going to go through all the techniques, and we used the most common one, the min-max normalization method in this project. The formula is defined as

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{2.1}$$

where x represent the current data point, and x' is the normalized data. The formula (2.1) tells that we take the minimum and maximum values from each attribute as our two tails and all other values are scaled to between those values to form our distribution which will then returns all the variables in the same scale.

Figure 4 shows the sample output after normalization from the first six rows of our dataset. We can compare it with Figure 3 which has not applied the normalization techniques, and we now have all the input values between 0 and 1.

FIGURE 4. Normalized data

	age	sex	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	ca	...	cp_1	cp_2	cp_3	thal_0	thal_1	thal_2	thal_3	slope_0	slope_
0	0.708333	1.0	0.481132	0.244292	1.0	0.0	0.603053	0.0	0.370968	0.0	...	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.
1	0.166667	1.0	0.339623	0.283105	0.0	0.5	0.885496	0.0	0.564516	0.0	...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.
2	0.250000	0.0	0.339623	0.178082	0.0	0.0	0.770992	0.0	0.225806	0.0	...	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.
3	0.562500	1.0	0.245283	0.251142	0.0	0.5	0.816794	0.0	0.129032	0.0	...	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.
4	0.583333	0.0	0.245283	0.520548	0.0	0.5	0.702290	1.0	0.096774	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.
5	0.583333	1.0	0.433962	0.150685	0.0	0.5	0.587786	0.0	0.064516	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.

2.4. **Train test split.**

Training and testing on the same dataset is a methodological mistake because a model that would have a perfect score for the current dataset; however, it may fail to predict an unknown dataset. And this is known as overfitting. To avoid overfitting, it is common practice when performing a supervised learning experiment to store part of the existing data as a test set. In this project, we randomly spilt 80% of our data as the training data, and another 20%

as the test dataset. More specifically, 242 records for training data, and 61 records for the testing data.

2.4.1. *k fold Cross Validation.*

When evaluating different settings for parameters, there is still a risk of overfitting on the test set because the parameters can be squeezed until the estimator performs optimally. In this situation, knowledge about the test set can leak into the model and evaluation metrics would not report on generalization performance. To solve this issue, another part of the dataset can be held out as a "validation set": training proceeds on the training set, once the evaluation is done on the validation set, and the experiment seems to be successful, then the final evaluation can be done on the test set.

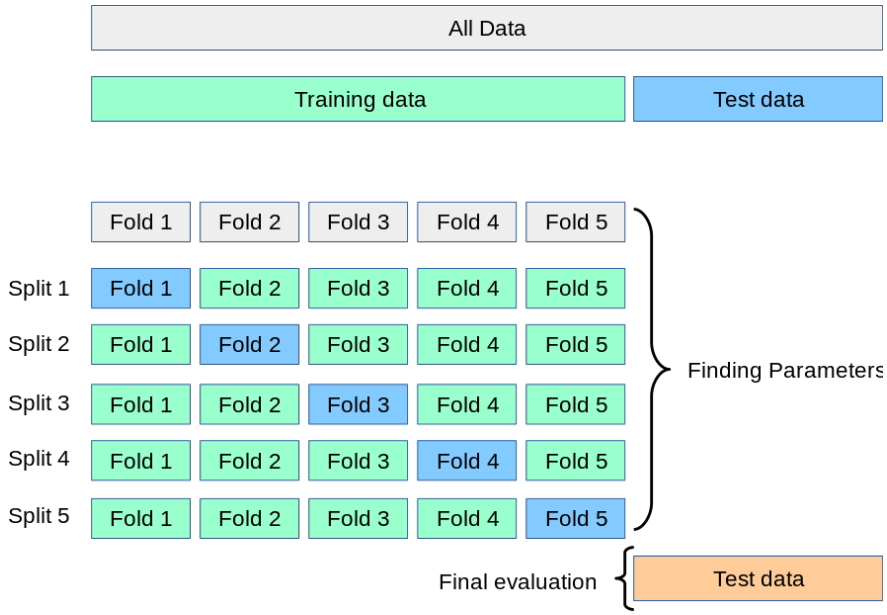
Still, by partitioning the available data into three sets, we thoroughly limit the number of samples which can be used for training the model, and the outcomes can depend on a particular random choice for the pair of training and validation sets.

A solution to this problem is called cross-validation (CV), which implies the test set still carried out the final evaluation, although the validation set is no longer needed when doing a CV. In the basic approach for k-fold CV, the training set is split into k smaller sets. The following steps are followed for each of the k folds:

- using $k - 1$ of the folds as training data
- the resulting model is validated on the remaining part of the data

Figure 5 illustrate an example when we have 5 fold cross-validation[7]:

FIGURE 5. Example for 5 fold Cross-Validation



The performance reported by k-fold cross-validation is the average of all the values computed in the loop (i.e. the average of accuracy for all iterations). This approach can be computationally expensive without waste too much data, which is a significant success in problems such as inverse inference where the number of samples is not large enough.

3. Methodology

3.1. **Evaluating a Classifier's Performance.**

3.1.1. *Confusion Matrix.*

A confusion matrix is obtained to calculate the accuracy of classification. A confusion matrix shows how many instances have been assigned to each class. Since we only have two classes for our dataset, and therefore we have a 2x2 confusion matrix, which is defined as following:

		Actual class	
		Positive	Negative
Class from classifier	Positive	TP	FN
	Negative	FP	TN

The True Positive (TP) denotes the number of records classified as true while they were actually true. False Negative (FN) denotes the number of records classified as false while they were actually true. False Positive (FP) denotes the number of records classified as true while they were actually false. Lastly, True Negative (TN) denotes the number of records classified as false while they were actually false.

Moreover, the method we used for model evaluation and comparison in this project is the accuracy rate, which is the percentage of test set that are correctly classified. Mathematically,

$$Accuracy = \frac{TP + TN}{Total}$$

And we want the accuracy rate as higher as possible.

3.2. **Logistic Regression.**

Logistic Regression is a variation of Linear Regression, useful when the observed output variable y is categorical. It creates a formula that predicts the probability of the class label as a function of the explanatory variables. Specifically, a logistic regression model has the form

$$log(\frac{p(x)}{1 - p(x)}) = \eta(x) \tag{3.1}$$

where $\eta(x)$ is a function describing systematic dependence on the explanatory variables.

$$\eta(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d \tag{3.2}$$

Logistic regression fits a special S-shaped curve by taking the linear regression and transforming the numeric estimate into a probability. To achieve this, a sigmoid function can be used and define as following:

$$\sigma(x) = \frac{1}{1 + \exp^{-\eta(x)}}$$

Basically, all the inputs squash the value between 0 and 1, and very negative inputs end up close to 0, and very positive inputs end up close to 1. It steadily increases around the input 0.

Now, we are ready to apply the logistic regression technique to train the model. Without cross-validation, we get 91.80% accuracy. And Table 3 presents the confusion matrix for the test data. As discussed early in section 2.4.1, this result might not be reliable since there is still a risk of overfitting on the test data. Therefore, 10-fold cross validation is also constructed for model evaluation purpose.

TABLE 3. Confusion Matrix for Logistic Regression

	Presence	**Absence**
Presence	28	4
Absence	1	28

By using 10 fold CV, the accuracy 0.88, 0.8, 0.72, 0.68, 0.88, 0.76, 0.91, 0.87, 0.91, and 0.83 are reached respectively, which gives an average accuracy of 82.42%.

3.3. **K-Nearest Neighbour (KNN).**

KNN was first popularized by Cover and Hart in 1967, who investigated the properties of 1-NN, the rule which uses only one neighbour[8]. The basic idea for KNN algorithm is very simple and straightforward, which assuming the data we are trying to classify similar to its neighbour; that is, the probability of its classes is roughly equal to its neighbour. Put it into formula, we have

$$p(y|x) \simeq p(y|x') \tag{3.3}$$

To determine the similarity, we used Euclidean distance which is calculated as

$$d = \sqrt{\sum_{i=1}^{p} (x_i - x_i')^2} \tag{3.4}$$

The smaller the distance is, more similar to the neighbour.

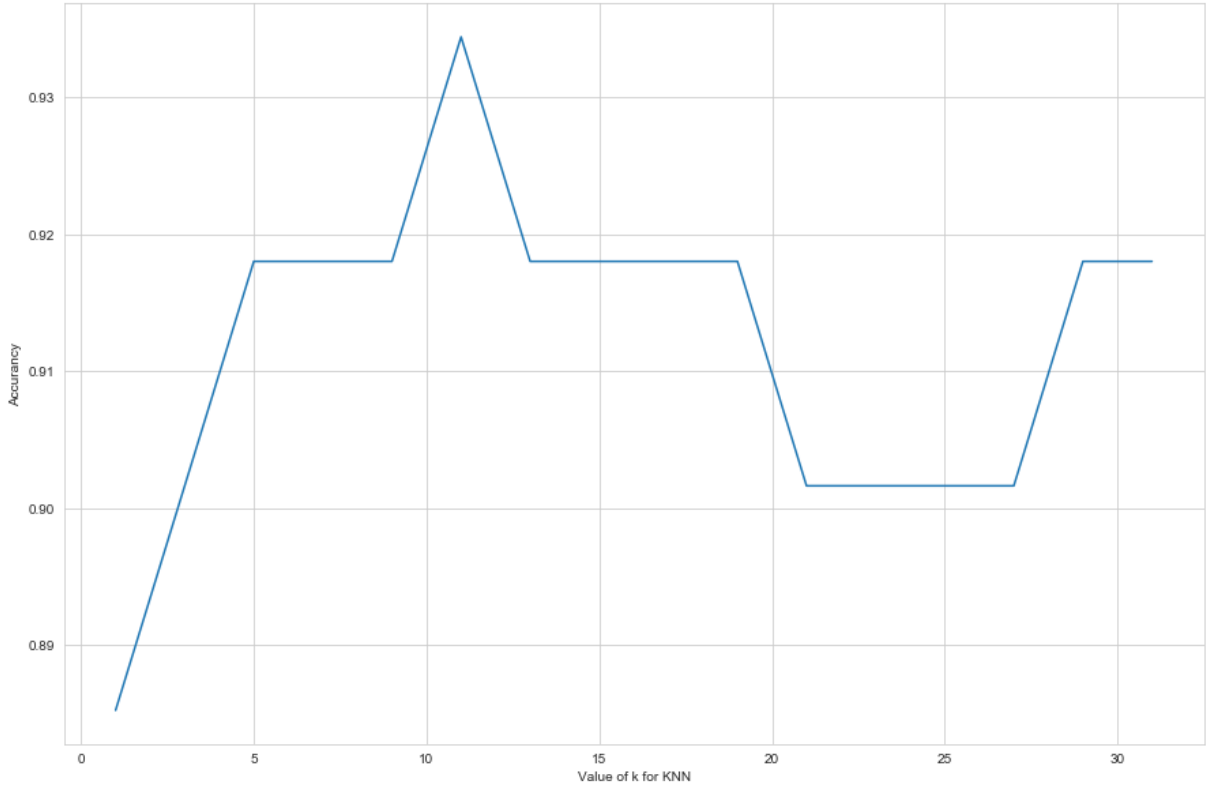
Moreover, the value of k cannot be even because it creates confusion for the algorithm to determine the class label when we have the same amount of the data point from each class.

For our dataset, we tested the model from k = 1 to k = 31. Figure 6 demonstrates the value of k versus the accuracy rate. Clearly, when k = 11, the model gives the best accuracy rate.

TABLE 4. Confusion Matrix for KNN with K = 11

	Presence	**Absence**
Presence	29	3
Absence	1	28

FIGURE 6. Accuracy for KNN with Different K

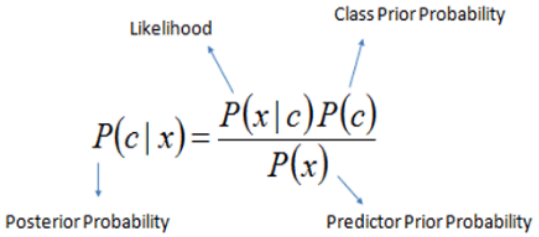


By using 10 fold CV, the accuracy for each test given by 0.8, 0.76, 0.76, 0.8, 0.92, 0.84, 0.87, 0.78, 0.78, and 0.87 respectively, which gives an average accuracy of 81.84%.

3.4. **Naïve Bayes.**

Naïve Bayes classifier is based on Bayes theorem. This classifier algorithm uses conditional independence, means it assumes that an attribute value on a given class is independent of the values of other attributes.

Let $X = \{X_1, X_2, \cdots, X_n\}$ be a set of n attributes. Then the conditional probability of class label can be express as



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Applying the Naïve Bayes method, we get the confusion matrix as following

TABLE 5. Confusion Matrix for Naïve Bayes

	Presence	**Absence**
Presence	31	1
Absence	12	17

By using 10 fold CV, the accuracy for each test given by 0.44, 0.68, 0.64, 0.68, 0.72, 0.64, 0.74, 0.57, 0.7, and 0.78 respectively, which gives an average accuracy of 65.83%.

3.5. **Decision Tree.**

The decision tree are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. There are two steps in this techniques building a tree and applying the tree to the dataset.

The confusion matrix for Decision Tree is observed as

TABLE 6. Confusion Matrix for Decision Tree

	Presence	**Absence**
Presence	22	10
Absence	1	28

By using 10 fold CV, the accuracy for each test given by 0.76, 0.76, 0.6, 0.76, 0.76, 0.68, 0.74, 0.57, 0.78, and 0.78 respectively, which gives an average accuracy of 71.90%.

3.6. **Support Vector Machine.**

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

The confusion matrix for SVM is from the test set is given by

Table 7. Confusion Matrix for SVM

	Presence	**Absence**
Presence	28	4
Absence	1	28

By using 10 fold CV, the accuracy for each test given by 0.92, 0.8, 0.72, 0.76, 0.84, 0.76, 0.87, 0.91, 0.83, and 0.91 respectively, which gives an average accuracy of 83.22%.

4. Model Comparison

In summary, we can compare the classifier performance based on both accuracy rate with 10 fold cross-validation and one without cross-validation. Table 8 shows the accuracy for all of five supervised learning techniques we used in this project.

Table 8. Summary

Algorithms	**Accuracy without CV**	**10-fold CV Accuracy**
Logistic	91.80%	82.42%
KNN	93.44%	81.84%
Naïve Bayes	78.69%	65.83%
Decision Tree	88.52%	71.90%
SVM	91.80%	83.22%

5. Conclusion

The overall objective of our work is to predict more accurately the presence of heart disease. In this paper, five different supervised learning techniques are conducted, they are Logistic Regression, K-Nearest Neighbour, Naïve Bayes, Decision Tree, and Support Vector Machine. The analysis shows that the SVM provides highest accuracy using the 10 fold cross-validation, and KNN with k = 11 returns the best performance without cross-validation. Because of

overfitting for test data has higher chance to happen without cross-validation; therefore, the SVM is recommended with a 83.22% accuracy overall.

For the future work, the system can be further expanded. For example, more algorithms such as Neural Networks, CNN, and random forest can be used to compare the performance for the prediction. In addition, more input attributes such as obesity and smoking can be added to the model to get more accurate results. Lastly, understanding the crucial factors that causing a heart disease based on the current health condition can be investigated to help people prevent the heart disease.

References

[1] Russell, S. J., and Norvig, P. (2010), *Artificial intelligence: A modern approach*, Pearson, ISBN 9780136042594.

[2] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012), *Foundations of machine learning*, Cambridge (EE. UU.): The MIT Press, ISBN 9780262018258.

[3] Alpaydin, E. (2010), *Introduction to machine learning*, Cambridge, Massachusetts: MIT Press, ISBN 9780262012430.

[4] Public Health Agency of Canada. (2017). Heart Disease in Canada. Retrieved from https://www.canada.ca/en/public-health/services/publications/diseases-conditions/heart-disease-canada.html

[5] Dua, D. and Graff, C. (2019), *UCI Machine Learning Repository*, [Online]. https://archive.ics.uci.edu/ml/datasets/Heart+Disease.

[6] O'Neil, C., Schutt, R. (2013, October). *Doing Data Science: Straight Talk from the Frontline*, ISBN 9781449358655, pp.41.

[7] 3.1. Cross-validation: Evaluating estimator performance. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/cross_validation.html

[8] Cover TM, Hart PE (1967). *Nearest neighbor pattern classification.* IEEE Transactions on Information Theory. 13 (1): 2127.

[9] Weka 3: Data Mining Software in Java, [Online]. https://www.cs.waikato.ac.nz/ml/weka/.