Heart Disease Diagnosis using Machine Learning

1nd Lauren Flanagan
Electrical and Computer Engineering
University of Western Ontario
London, ON, Canada
lflanag4@uwo.ca

2st Leah Krehling
Electrical and Computer Engineering
University of Western Ontario
London, ON, Canada
lkrehlin@uwo.ca

3rd Tareq Tayeh
Electrical and Computer Engineering
University of Western Ontario
London, ON, Canada
ttayeh@uwo.ca

***Abstract*—Heart disease is a range of illnesses that many people are diagnosed with every year, where it is the second leading cause of death in Canadians. With the modern resurgence of machine learning models, there is a lot of interest in using them for medical diagnosis, as the theoretical benefits could provide significant improvements to the healthcare industry. In this paper, four different classification models were compared, and then merged using ensemble learning methods into a single model. Our model achieved an accuracy of 86% and four false negatives.**

***Index Terms*—Artificial Intelligence, Machine learning, Supervised learning, Statistical learning, Predictive models, Naïve bayes, Neural networks, Logistic regression, Support vector machine, Ensemble learning.**

I. Introduction

The term heart disease consists of a group of conditions that affect the structure and functions of the heart. Diseases under this umbrella include blood vessel diseases, such as coronary artery diseases, heart rhythm problems (arrhythmias), congenital heart defects, heart failures, and other diseases related to the heart [1]. About 1 in 12 Canadians live with a diagnosed heart disease, totalling to 2.4 million adults. Every hour about 12 Canadians with a diagnosed heart disease die, making it the second leading cause of death in Canadians [2].

Disease analysis is an important aspect of medicine. The healthcare industry collects a huge amount of healthcare data from patients through clinic appointments, hospital visits, and medical studies. This data can be mined to discover hidden information about an illness that could offer a breakthrough in treatment, care, or diagnosis of patients. Accurate analysis of data, such as high blood pressure and high cholesterol, benefits early disease detection, which in turn could lower the number of patients dying of heart disease.

Automating the diagnosis process would allow a large data pool to be leveraged in an effort to find early signs of heart disease. This could inform the medical community of factors they had previously been unable to decisively link to the problem. In addition, it could create a system that can consider all of the affecting values together more efficiently and objectively than individual humans assessing a single patient. Overall, it is believed that machine learning in the medical field can help in providing contextual relevance, improving clinical reliability, helping physicians communicate objectively, reducing errors related to human fatigue, decreasing mortality rates, diminishing medical costs, and identifying diseases more readily [3]. A machine learning algorithm that can diagnose heart disease is the first step in providing these benefits.

In this paper, four different classification machine learning models were built to analyse the data and predict whether a patient has heart disease or not. The models tested were Logistic Regression, Support Vector Machine, Naïve Bayes and a Neural Network. Information on these models can be found in section II. The models were tuned, trained, and then compared using the metrics described in section II-F. Following this, the models were combined using ensemble learning methods and the results of all the models are discussed in section IX. The data set used in this paper was collected by the V.A. Medical Center of Long Beach and the Cleveland Clinic Foundation by Robert Detrano, M.D., Ph.D. [4] and is described in Data Set section IV.

This paper is organized as follows: section II details the evaluation metrics that were used for the models, as well as some basic information on how the models operate; Section III contains information on similar research in this field; Section IV contains information about the data set that was used; Section V discusses the pre-processing methods performed on the data; Section VI discusses the feature selection techniques performed on the data; Section VII contains information on the validation method used; Section VIII details the specifics of the final tuned models that were used; Section IX discusses all of the model's performances; Section X compares the performance between all models; Lastly, section XI contains the final conclusions derived from the research.

II. Background

This section will discuss the required background information about the machine learning models used and the comparison metrics applied to evaluate them.

A. Logistic Regression

Logistic regression is used to estimate the probability of a binary response based on a set of one or more independent variables. The logistic regression function seen in equation 1 takes an input in the range of negative to positive infinity and applies weights to map the input to an output range of 0 to 1. The output represents the probability that the dependent variable is true: as x increases the output will become closer to 1, and as x decreases the output will become closer to 0 [5]. Logistic regression uses a cost function to determine how

far it is from the global minimum. A solver is used to find weights to minimize the cost function [5].

$$f(x) = \frac{1}{1 + e^{-\theta x}} \tag{1}$$

B. Support Vector Machine

Support Vector Machine (SVM) is a discriminative classifier that finds a hyperplane that separates classes. The hyperplane is selected as the one that best separates the classes, with the largest margin between each class and the plane. SVM uses a kernel function to provide a way of using a linear classifier to solve a non-linear problem. The kernel function is applied to map the original linearly inseparable observations into a higher-dimensional space where the observations become linearly separable [6].

C. Neural Network

An artificial neural network is a computational model that was inspired by the biological neural networks in the human brain. The basic unit of computation in a neural network is the neuron, also called a node or unit. The neuron receives input from other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs. The node applies a function to the weighted sum of its inputs. The neuron then has a "firing rate" which is an activation function, which controls when the neuron triggers to pass along information. In a neural network the neurons are organized into layers and most of the connections in a network occur between neurons of different layers. The first layer is the input layer, the last is the output layer, any layers in between are referred to as "hidden" layers [7].

D. Naïve Bayes

Bayesian inference is a technique of statistical inference where Bayes' theorem updates the probability of a hypothesis when new evidence becomes available [8]. The direct application of Bayes theorem can become complex as it assumes dependency between all the input features, and only works effectively in calculating the conditional probabilities of the observations when the number of examples is large. In the case where the number of examples are relatively small, the theorem is simplified by assuming independence between the input features. This is referred to as Naïve Bayes [9]. In simple terms, the Naïve Bayes statistical classifier assumes that the absence or presence of a certain input feature is unrelated to the absence or presence of another input feature, maximizing the posterior probability in determining the class without using any Bayesian methods.

E. Ensemble Learning

Ensemble learning is used to improve one or both of accuracy and computation time. It combines multiple machine learning algorithms to obtain better learning outcomes than obtained from each machine learning algorithms on its own [10]. Each learner can either be trained on the complete data set, or on only a subset of the data. The outcome from each machine learning algorithm is combined using a voting scheme to produce the final output.

F. Comparison Metrics

Comparison metrics are used to test how well a model is performing, and to compare the performance between different models.

1) Confusion Matrix: A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix is designed to allow for easy identification of how and when the model is mistaking one class for another. The columns of the matrix are the predicted categories, and the rows are the actual categories.

The confusion matrix provides insight into the false positive and false negative tendencies of the classification model. A false positive is when the model predicts that the entry is of a category when it is not, and a false negative is when the model predicts that the entry is not of a category when it is. If the majority of the values are all along the diagonal of the matrix this indicates that the majority of the classifications predicted by the model are correct, while all values off of the diagonal are false positives and negatives [11].

2) Accuracy: The accuracy of the model is the ratio of correct predictions to total predictions. Thus, it is a percentage of how many of the predictions were correct. A high accuracy therefore corresponds to a model that makes a correct prediction in most cases. The formula for accuracy is equation 2.

$$accuracy = \frac{true\_predictions}{total\_predictions} \tag{2}$$

3) Recall: Recall is the ratio of the total number of correctly classified predictions for a class and the total number of entries in the tested data that were of that class. A high recall indicates the class is correctly being recognized by the model and there are a small number of false negatives [11]. The formula for recall is equation 3.

$$recall = \frac{true\_positives}{true\_positives + true\_negatives} \tag{3}$$

4) Precision: Precision is the ratio of the correctly classified positive predictions for a class and the total number of predictions the model made for that class. A high precision indicates an example labeled as positive is indeed positive and there are a small number of false positives [11]. The formula for precision is equation 4.

$$precision = \frac{true\_positives}{true\_positives + false\_positives} \tag{4}$$

5) F1-Score: The F1-Score is a measure that represents both the precision and recall of the model. This measure uses the harmonic mean in order to weigh the lower value more heavily. Due to this, the F1-Score will be closer to the lower

of the two values of precision and recall [11]. The formula for the F1-score is equation 5.

$$F1Score = \frac{2 \cdot recall \cdot precision}{recall + precision} \quad (5)$$

III. Related Work

Current research in using machine learning algorithms to predict heart disease is being done by several different researchers. Haq A., et al. proposed testing the following machine learning methods: Logistic Regression, SVM, Naïve Bayes, Artificial Neural Network, Decision Tree Classifier, and K-nearest Neighbour. These achieved results of 84%, 86%, 83%, 74%, 74%, and 76% accuracy respectively. To remove irrelevant features, they performed feature selection using three algorithms: Relief Feature Selection Algorithm, Minimal-Redundancy-Maximal-Relevance Feature Selection Algorithm, and Least Absolute Shrinkage Operator [12].

Researchers Kahramanli H., and Allahverdi N. [13] proposed a hybrid neural network that combines an artificial neural network with a fuzzy neural network. Using k-fold cross validation, they achieved an accuracy of 86.8%.

Researchers Soni J., et al. [14] surveyed different machine learning techniques used in heart disease predictions on 1000 different patient records. They discovered that Decision Trees and Bayesian Classifications had the highest accuracy of all techniques with a result of 90% and upwards, outperforming KNN, Neural Networks, and Classification based on clustering techniques where their results were all around 88% accuracy.

IV. Data Set

The data set used for our heart disease predictions was provided from "V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D." (Cleveland Database) [4]. The data set contained information on 303 patients with 76 attributes, but most published experiments used a subset of fourteen attributes. Hence, our analysis utilized this subset of processed data to predict heart disease presence in a patient.

Details of the attributes can be seen in table I. Attributes one through thirteen were the independent attributes, and attribute fourteen was the dependent attribute.

TABLE I
Attributes in the Data Set

#	Attribute	Description	Type
1	age	Age in Years	Integer
2	sex	Sex in Binary: '0 = Female, 1 = Male'	Binary
3	cp	Chest Pain : 1 = Typical angina, 2 = Atypical angina, 3 = Non-anginal pain, 4 = Asymptomatic	Integer / Categorical
4	trestbps	Resting Blood Pressure in mm Hg (millimeters of Mercury) on admission to the hospital	Integer / Continuous
5	chol	Serum Cholesterol in mg/dl (milligrams per deciliter)	Integer / Continuous
6	fbs	Fasting Blood Sugar: 0 = <120mg/dl, 1 = >120mg/dl	Binary
7	restecg	Resting electrocardiographic results: 0 = Normal, 1 = Having ST-T wave abnormality, 2= Showing probable or definite left ventricular hypertrophy by Estes' criteria	Integer / Categorical
8	thalach	Maximum heart rate achieved during thallium stress test	Integer / Continuous
9	exang	Exercise induced angina: 0 = No, 1 = Yes	Binary
10	oldpeak	ST depression induced by exercise relative to rest	Integer / Continuous
11	slope	Slope of peak exercise ST segment: 0 = Downsloping, 1 = Flat, 2 = Upsloping	Integer / Categorical
12	ca	Number of major vessels colored by fluoroscopy: 0-3 vessels	Integer / Categorical
13	thal	Thallium stress test result: 1 = Normal, 2 = Fixed Defect, 3 = Reversible Defect	Integer / Categorical
14	target	Heart disease present or not: 0 = No Heart Disease, 1 = Heart Disease	Binary

V. Data Pre-Processing

A. Data Editing and Cleansing

All attribute values from the processed data set were kept the same, except for 'thal and 'target'. 'thal' categorical values of 3/6/7 were shifted to 1/2/3 for easier data manipulation and interpretation, as it ensures the same numerical gap amount between each categorical value. 'Target' had five categorical values spanning from 0 through 4, but no details were released about what categorical values 2 through 4 could indicate other than heart disease is present, so their values were transformed to 1 for complete domain knowledge.

There were a couple of NaN values across the data set for attributes 'ca' and 'thal'. They were filled in with the attribute's mean value grouped by the 'target' attribute, rounded to the nearest whole number as they are categorical values. That ensured reasonable estimation of what the missing value could be rather than ignoring the entire patient's record.

B. Normalization

Afterwards, feature values were normalized. As the features had different ranges of values and types, normalization ensured certain features did not have a greater influence on the result due to their larger values. That was necessary for all the methods used to provide accurate predictions, except for Naïve Bayes, as the probability of each feature was calculated independently and automatically shifted all values to the same range of 0 through 1.

VI. Feature Selection

In order to optimize the model's heart disease prediction accuracy, certain features had to be selected from the thirteen available independent attributes. Eliminating the irrelevant features reduces the number of redundant and misleading data, reducing over fitting and the models' training time. The selection process was achieved via 3 methods: Intrinsic Discrepancy, Correlation Matrix, and Permutation Importance.

TABLE II
INTRINSIC DISCREPANCY RESULTS

Feature	Intrinsic Discrepancy
cp	0.601605
thal	0.593108
ca	0.470883
thalach	0.43874
oldpeak	0.422044
exang	0.369624
slope	0.316807
age	0.17894
sex	0.160989
restecg	0.0666806
chol	0.040521
trestbps	0.0306689
fbs	0.00126001

TABLE III
CORRELATION COEFFICIENTS OF FEATURES WITH 'TARGET'

Feature	Coefficient
age	0.2
sex	0.3
cp	0.4
trestbps	0.2
chol	0.1
fbs	0.0
restecg	0.2
thalach	-0.4
exang	0.4
oldpeak	0.4
slope	-0.3
ca	0.5
thal	0.5

TABLE IV
PERMUTATION IMPORTANCE RESULTS

Feature	Weight
ca	0.900 ± 0.806
thal	0.0567 ± 0.0267
oldpeak	0.0267 ± 0.0618
sex	0.0267 ± 0.0452
cp	0.0233 ± 0.0340
slope	0.0200 ± 0.0249
exang	0.0133 ± 0.0327
chol	0.0100 ± 0.0163
thalach	0.0033 ± 0.0133
restecg	0.0000 ± 0.0211
fbs	0.0000 ± 0.0000
age	-0.0067 ± 0.0267
trestbps	-0.0067 ± 0.0163

A. Intrinsic Discrepancy

Intrinsic Discrepancy measures the discrepancy between the "heart disease" and "no heart disease" distributions for each attribute via two probability distributions. It is based on information theory, and is defined as equation 6.

$$\delta\{p_1, p_2\} = min\{\int p_1(x) \log \frac{p1(x)}{p2(x)} dx, \int p_2(x) \log \frac{p2(x)}{p1(x)} dx\} \tag{6}$$

The results of the method against the data set are shown in figure II. A high intrinsic discrepancy value indicates that the feature could be valuable for inclusion in our machine learning models, as it has a lot of information gain about the presence of heart disease. As it can be seen from the results, fasting blood sugar, resting blood pressure, serum cholesterol, resting electrocardiographic results, sex, and age produced the lowest values, indicating the lowest information gain about the presence of heart disease.

B. Correlation Matrix

A correlation matrix is a matrix of correlation coefficients, which is a statistical measure that calculates the strength of the relationship between two relative variables. Values are between the range of -1.0 and 1.0, where -1.0 indicates a perfect negative correlation and 1.0 indicates a perfect positive correlation. 0.0 indicates there is no correlation between the variables. Due to visibility, only the correlation coefficients of the attributes with the 'target' can be seen in table III, as they are the only coefficients we are interested in from the matrix.

As it can be observed from the results, fasting blood sugar had a correlation of 0.0 with the presence of heart disease, meaning there was no correlation between the two. Serum cholesterol had a correlation of 0.1 with the presence of heart disease, where age, resting blood pressure, and resting electrocardiographic results had a correlation of 0.2 with the presence of heart disease. These small values indicated weak and insignificant correlations between the two variables, unlike the rest of the independent attributes with 'target'.

C. Permutation Importance

Permutation importance, or feature importance, measures which features have the biggest impact on predictions after a model has been fitted. It randomly re-orders a single column of the validation data, leaving the target and all other columns in place, and calculates the prediction accuracy of the now-shuffled data. The process is repeated with multiple shuffles to measure the amount of randomness in the calculation.

The feature importance of our data set can be seen in table IV. The features are displayed in descending order of their weight values, meaning the most important features rest towards the top of the list. The first number specifies how much the model's performance dropped with the random shuffle using the accuracy performance metric, and the second number specifies how much the model's performance varied from one reshuffle to the other. As it can be observed from the results, resting blood pressure, age, fasting blood sugar, and resting electrocardiographic results rest towards the bottom of the list, indicating the least importance features on the models.

D. Feature Selection Verdict

Upon conducting all three feature selection methods, it was decided that if an attribute satisfied all of the following three criteria, it was dropped: Intrinsic Discrepancy < 0.2, Correlation Coefficient with the target < 0.3, and Permutation

Importance < 0.02. It is important to note that defining a fixed threshold for each criteria was difficult. Threshold values were selected via a trial and error methodology on maximizing average prediction's accuracy, and ensured the attributes dropped had low information gain, and weak correlations and relationships with the presence of heart disease in a patient.

The attributes dropped were: age, fasting blood sugar, resting blood pressure, serum cholesterol, and resting electrocardiographic results.

VII. Validation Process

K-Fold cross-validation method was utilized to evaluate each machine learning model. The method first shuffles the data set randomly before splitting it into k groups, and for each unique group, that group was used as the test data set and the remaining groups were used as the training data. The model was then fit and evaluated accordingly. Each observation is assigned to a single group and remains in that group for the procedure duration. A k value of 5 was selected for this project, as it is a very common value in the field of applied machine learning and higher values would cause higher computational complexity, which was unnecessary with the small data set size in hand.

VIII. Implementations

The models were built in python 2.7, using Tensorflow and scikit-learn libraries.

A. Logistic Regression

The binary response of the logistic regression algorithm represents whether heart disease is present based on the independent input variables. The solver types that were considered include Newton-Conjugate Gradient (CG), Limited-memory Broyden-Fletcher-Goldfarb (LBFGS), Library for Large-Scale Linear Classification (liblinear), Stochastic Average Gradient (SAG), SAGA (a variant of Stochastic Average Gradient) [15]. The performance based on the solver type was investigated, and the model was run 5 times for each solver using the 5-fold validation.

B. Support Vector Machine

SVM is used to find a hyper-plane to separate when heart disease is present and when heart disease is not present in the patient. The performance of the SVM based on the kernel type was investigated, and the model was run 5 times for each kernel using 5-fold validation.

The different kernel types considered include Linear, Polynomial, Gaussian radial basis function, and Sigmoid [16]. The linear kernel takes the inner product of x and y and adds a constant shown in equation 7.

$$k(x, y) = x^T y + c \tag{7}$$

The polynomial kernel builds on the linear kernel to incorporate a slope term, and is of order d, shown in equation 8.

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{8}$$

For the rbf kernel, the sigma values affects the sensitivity to noise in the training data, shown in equation 9

$$k(x_i, x_j) = exp(-\Upsilon||x_i - x_j||)^2 \tag{9}$$

The sigmoid kernel comes from neural network theory. The alpha value represents slope and the c values if the y-intercept, shown in equation 10

$$k(x, y) = tanh(\alpha x^T y + c) \tag{10}$$

C. Neural Network

The network used in this paper was a fully connected feed forward network. This refers to all of the neurons of each layer being connected to each of the neurons in the next layer, and none of them being connected back to a previous layer. Various aspects of a neural network model can be tuned, and each tuned parameter is listed:

- Optimization Function: algorithm used to minimize or maximize error function dependent on the model's weights and biases
- Learning Rate: an algorithm which determines how much to modify the neuron weights to tune them to produce a favoured output
- Momentum: controls how the previous modification influences the current weight modification
- Weight Initialization: determines how the initial weights for the neurons are selected
- Activation Function: defines the output of that node given an input or set of inputs
- Weight Constraint: checks the size or magnitude of the weights and scales them so they are all below a predefined threshold
- Drop Out Weight: percentage of randomly selected neurons to ignore during each cycle of training
- Number of Neurons: the size of each layer which adds to the complexity of the network
- Batch Size: the number of patterns shown to the network before the weights are updated
- Epochs: the number of times that the entire training set is shown to the network during training

A range of values for each of these parameters was chosen using a grid search [17] method, with 3-fold validation employed. Due to the computationally heavy nature of this method the entire range of values was not tested, instead a few common values were tested for each and then finer tuning was done manually to improve the results incrementally. Though this is not as thorough tuning as testing all values using grid search and has limitations in results, the computational intensity of the grid search method made testing more values infeasible. From the results of the tuning it was determined that the model would be built and tested using the listed values: Optimization Function: Adam; Learning Rate: 0.001; Momentum: 0; Weight Initialization: normal; Activation Function: relu; Weight Constraint: None; Drop Out Weight:

0.2 or 20%; Number of Neurons: layer 1 has 16 neurons, layer 2 has 8; Batch Size: 10; Epochs: 300.

The Adam optimization function follows the method described in the original paper [18]. Adam is a first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments, it is also comparatively not computationally intensive and well suited to a variety of data sets [18].

For the output layer of this network a single neuron with a sigmoid activation function was used to create the binary classification output required of the problem. Sigmoid functions are optimal for this as it will pull the results of the model to either 0 or 1 to provide the class prediction made by the model.

D. Naïve Bayes

The Naïve Bayes theorem is as follows:

$$P(class \mid X1, X2, .., X8) = P(X1 \mid class) * P(X2 \mid class) * .. * P(X8 \mid class) * P(class) \quad (11)$$

where class = dependant attributes and X = independent attributes.

In other words, the probability of the class given the selected eight features is calculated by multiplying the probabilities of the individual features given the class with the probability of the class.

Input feature values were drawn from a Gaussian distribution. A Gaussian distribution is a continuous function that approximates the exact binomial distribution of events, and can be encapsulated by only using the mean and standard deviation. The Gaussian Probability Distribution Function can be calculated as:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (12)$$

where σ= standard deviation, μ = mean.

It has worked well in many real-life complex situations, as well as in previous publications of heart disease diagnosis.

E. Ensemble Learning

Voting schemes that were considered include: majority voting, where the most common prediction between all models is the accepted prediction; weighted voting where the importance of each model is defined and the prediction of heavier weighted algorithms has a greater influence on the accepted prediction; sample averaging where the average of each models prediction is the accepted prediction; and weighted average where the prediction of each model is multiplied by a weight and then averaged to determine the accepted prediction [10].

Ensemble learning was investigated using the majority voting and weighted voting schemes. It did not make sense to use the average voting or weighted average voting schemes because our output is a binary classification and the average would result in a non-binary output. Each voting scheme was explored to determine the best implementation and then a comparison of the two voting schemes was done.

Majority voting and weighted voting were tested with the learning outcome of all the considered algorithms: SVM, logistic regression, Naïve Bayes, and Neural Network. Both methods were run 5 times and the average results from each run were taken.

The most critical thing was to minimize the number of false negatives, so that patients with the disease are not dismissed. Therefore, for the weighted voting method, the weight of each algorithms was determined by the number of false negatives it predicted in each test. The steps to determine the weight were as follows: run each model; order the models from lowest number of false negatives to highest number of false negatives; assign the weights in this order so the model with the lowest number of false negatives receives the highest weighting; apply weighted voting process and record results of ensemble learning; repeat this process for 5 tests; average the results from all tests.

To determine the weight assigned to each model, several tests were done with different weight combinations and the averaged results were compared.

TABLE V
EVALUATION SCORES FOR LOGISTIC REGRESSION

Metrics	Solver Type				
	Newton-CG	**lbfgs**	**liblinear**	**sag**	**saga**
Accuracy (%)	88.33	85.83	81.25	83.13	79.90
Precision (%)	96.15	86.08	79.03	81.52	79.64
Recall (%)	80.64	81.62	76.81	78.79	75.60
F1-Score (%)	87.72	83.44	77.72	80.03	77.52
False Negatives	6.0	5.0	6.0	5.50	6.75
True Negatives	28.0	29.5	28.75	29.38	27.19
False Positives	1.0	3.5	5.25	4.62	5.31
True Positives	25.0	22.0	20.0	20.50	20.75

IX. RESULTS

For this data set anything under 75% accuracy is an unsatisfactory result and further tuning should be performed [11]. This is determined based on the related work that was reviewed in section III.

A. Logistic Regression

The average results of the accuracy, precision, recall, f1-score, along with the confusion matrix results, are presented in table V. Based on the results, lbfgs was selected as the best solver type. Since minimizing the number of false negatives was critical, the liblinear, Newton-cg, and saga solvers were eliminated from consideration. Then looking at the number of false positives, and the accuracy, the lbfgs solver preformed better than the sag solver. Therefore, the lbfgs solver was selected as the best solver type to use. Logistic regression with lbfgs resulted in 85% accuracy and 5 false negatives.

B. Support Vector Machine

The average results of the accuracy, precision, recall, f1-score and the confusion matrix are presented in table VI.

TABLE VI
EVALUATION SCORES FOR SVM

Metrics	Kernel Type			
	Linear	Poly	rbf	Sigmoid
Accuracy (%)	83.18	82.09	81.6	59.8
Precision (%)	83.70	83.21	81.20	56.57
Recall (%)	79.83	78.07	78.52	54.09
F1-Score (%)	81.20	79.98	79.52	54.25
False Negatives	5.8	6.3	6.05	13.17
True Negatives	28.4	28.23	27.7	22.62
False Positives	4.4	4.57	5.09	11.18
True Positives	22.0	21.50	22.75	15.63

Based on the results, the best preforming kernel was linear. By prioritizing the minimization of the number of false negatives, the sigmoid kernel was eliminated from consideration. The linear kernel performed better than the poly and rbf kernels in the number of false negatives, number of false positives and the overall accuracy. Therefore, the linear kernel was selected as the best kernel type to use. SVM with a linear kernel achieved an accuracy of 83% and 6 false negatives.

C. Neural Network

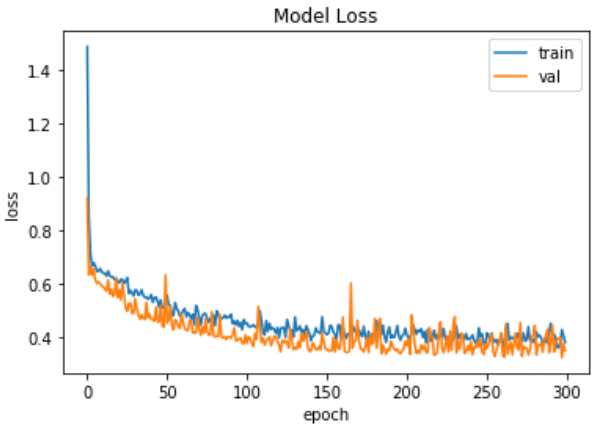

Fig. 1. Loss updating over the epochs

As a neural network model is repeatedly trained over epochs the weights are adjusted, and the accuracy and loss of the model is calculated for each epoch. The loss is a summation of the errors made for each example in the sets. For this model, the loss calculation was based on binary cross-entropy. When compared over epochs of training, the accuracy should increase over time and converge towards one. For loss, the lower it is the better the model, with the exception of an over fit model. There are four main problems with a model that can be identified from these curves, under fitting, over fitting, an unrepresentative training set, and an unrepresentative validation set [19].

The loss is seen in figure 1. The loss converges to 0 in both the training and validation data which indicates that the model is well fit to the data and the training and validation are close together which is also indicative of a model that is behaving desirably. The accuracy over time was also observed, and showed that the accuracy converged towards one with both the training and validation data.

TABLE VII
CONFUSION MATRIX OF THE NEURAL NETWORK

Comparison Metric	Results
Accuracy(%)	85.30
Precision(%)	86.0
Recall(%)	86.0
F1-Score(%)	85.0
False Negatives	8.20
True Negatives	26.50
False Positives	1.20
True Positives	25.0

The overall results received by the neural network can be seen in table VII, as well as the typical confusion matrix that the model was achieving. The neural network achieved an accuracy of 85% and 8 false negatives.

D. Naïve Bayes

The average results of the accuracy, precision, recall, f1-score and the confusion matrix are presented in table VIII. Naïve Bayes achieves an accuracy of 86% and 5 false negatives.

TABLE VIII
EVALUATION SCORES FOR NAÏVE BAYES

Comparison Metrics	Result
Accuracy (%)	85.72
Precision (%)	88.73
Recall (%)	87.21
F1-Score (%)	87.53
False Negatives	5.0
True Negatives	21.0
False Positives	6.0
True Positives	29.0

E. Ensemble Method

The results of the majority voting scheme are provided in table IX and show that a majority voting scheme does not improve the model. This is because the input from the worst preforming models has an equal input to the best preforming models. The results of the weighted voting scheme are shown in table X and show that assigning a high weight to the best preforming model, and low, equal weights to the other models results in the lowest number of false negatives.

X. COMPARISON OF RESULTS

Comparing the performance of each model, shown in table X, it can be seen that the accuracies are very similar between all models. Logistic regression performed the best with an accuracy of 85.83%. Logistic regression and naïve bayes both produced five false negatives, which was the best result across the individual models.

As seen in table IX, the ensemble method with a majority voting scheme resulted in a worse accuracy and a worse

TABLE IX
EVALUATION SCORES FOR ENSEMBLE METHOD

Comparison Metrics	Result
Accuracy (%)	83.83
Precision (%)	83.83
Recall (%)	80.71
F1-Score	82.05
False Negatives	5.40
True Negatives	28.40
False Positives	4.40
True Positives	21.50

TABLE X
EVALUATION SCORES FOR ENSEMBLE METHOD

Metrics	Weights			
	[4,3,2,1]	[4,2,1,1]	[4,1,1,1]	[6,1,1,1]
Accuracy (%)	82.29	84.92	85.25	86.23
Precision (%)	81.62	85.97	87.71	86.45
Recall (%)	79.50	80.78	81.02	85.32
F1-Score	80.50	83.20	84.08	85.80
False Negatives	5.8	5.4	5.0	4.4
True Negatives	27.6	29.0	27.6	27.2
False Positives	5.0	3.8	3.4	4.0
True Positives	22.6	22.8	24.4	25.4

number of false negatives than the best individual models. Table X shows that the ensemble method with a weighted voting scheme, where the weights are based on the number of false negatives, results in a lower number of false negatives and equal accuracy when compared to the best performing individual model. Although the accuracy did not improve from the individual models, it was critical to minimize false negatives so that patients with heart disease are not dismissed. The ensemble method still detects false negatives, however, it has reduced the number of false negatives in comparison to each individual model and therefore is a step in the right direction.

TABLE XI
FINAL COMPARISON OF THE MODELS

Model	LR	SVM	NN	NB
Accuracy(%)	85.83	83.18	85.30	85.72
False Negatives	5.0	5.8	8.2	5.0

XI. CONCLUSIONS

This work investigated and showed the potential of using machine learning models in the context of health care and disease diagnoses. After preforming data pre-processing, normalization, and feature selection on the data set used for heart disease predictions, k-fold cross validation was used to implement machine learning algorithms including Logistic Regression, SVM, Naïve Bayes, and a Neural Network. Between these models, the best accuracy achieved was 86%, with five false negatives. By implementing ensemble learning with a weighted voting scheme, the accuracy was 86% with four false negatives. This shows potential to lead to advances in the medical field. To improve upon these results, future steps include applying this work to a data set that includes more patients.

REFERENCES

[1] "Types of heart disease." [Online]. Available: https://www.heartandstroke.ca/heart/what-is-heart-disease/types-of-heart-disease

[2] P. H. A. of Canada, "Government of canada," Feb 2017. [Online]. Available: https://www.canada.ca/en/public-health/services/publications/diseases-conditions/heart-disease-canada.html

[3] "Artificial intelligence in medicine: Machine learning." [Online]. Available: https://www.ibm.com/watson-health/learn/artificial-intelligence-medicine

[4] R. Detrano, "Heart disease data set," 1988. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart Disease

[5] J. Kiggins and A. Gibson, "Deep learning a pratitioner's approach," *O'Reilly Media Inc.*, 2017.

[6] D. Team, "Kernel functions - introduction to svm kernel examples," Nov 2018. [Online]. Available: https://data-flair.training/blogs/svm-kernel-functions/

[7] D. Fumo, "A gentle introduction to neural networks series - part 1," Oct 2017. [Online]. Available: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc

[8] *Nature of Bayesian Inference*. John Wiley Sons, Ltd, 2011, ch. 1, pp. 1–75. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118033197.ch1

[9] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," *Proceedings of the 22nd international conference on Machine learning - ICML 05*, p. 529–536, Aug 2005.

[10] N. Demir, "Ensemble methods: Elegant techniques to produce improved machine learning results." [Online]. Available: https://www.toptal.com/machine-learning/ensemble-methods-machine-learning

[11] "Confusion matrix in machine learning," Feb 2018. [Online]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[12] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Information Systems*, vol. 2018, p. 1–21, Feb 2018.

[13] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert Systems with Applications*, vol. 35, no. 1-2, p. 82–89, 2008.

[14] J. Soni, U. Ansari, D. Sharma, and S. Soni, "Predictive data mining for medical diagnosis: An overview of heart disease prediction," *International Journal of Computer Applications*, vol. 17, no. 8, p. 43–48, 2011.

[15] J. Hale, "Don't sweat the solver stuff: Tips for better logistic regression models in scikit-learn," 9AD. [Online]. Available: https://towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451

[16] C. Souza, *Kernel Functions for Machine Learning Applications*, Mar 2010. [Online]. Available: http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

[17] "Tuning the hyper-parameters of an estimator." [Online]. Available: https://scikit-learn.org/stable/modules/grid$_search.html$

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[19] J. Brownlee, "How to use learning curves to diagnose machine learning model performance," Aug 2019. [Online]. Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/