1 | INTRODUCTION

This report aims to address the issue of delayed heart disease detection in the healthcare field. In many instances, a patient's risk of heart disease is detected when the situation has escalated, which effectively results in serious heart conditions and in some cases, heart failure. The goal is to develop a model that can predict whether a patient is at risk for heart disease or not, given updates in the patient's medical records of common vitals associated with heart disease prediction. Previous work reveals there have been attempts to build reasonable models for such a prediction. However, this project will expand on previous work by incorporating more complex machine learning models and critically evaluating models based on important diagnostics such as the confusion matrix, receiver-operating curve (ROC), and area under the ROC (AUC).

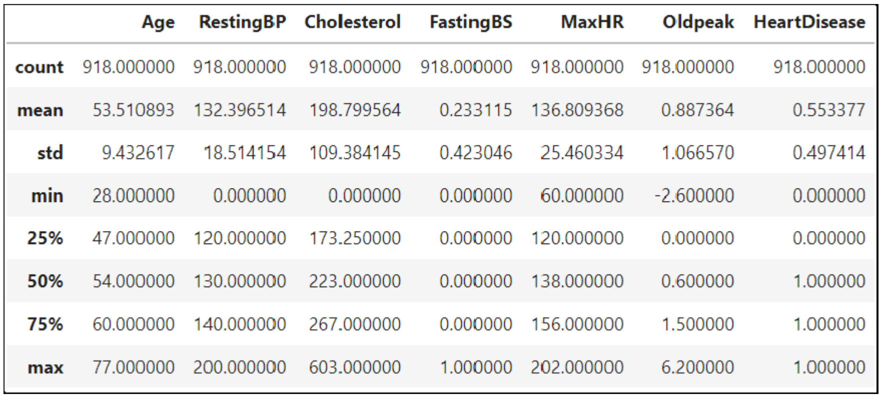
Patients may have medical tests conducted for various reasons, which may result in new values for vitals such as resting blood pressure, cholesterol level, and several other important measurements. Although these vitals may be updated in the patient's records, an automation system that can predict whether the new values result in a higher risk of heart disease to relay back to the family doctor currently does not exist. As a result, the patient may well be at a higher risk of heart disease, yet this will not be known until the patient exhibits signs of such heart complications. Therefore, it is imperative to construct a machine learning model that will predict whether a patient is at risk for heart disease given a set of vitals and to further implement a system that will notify a patient's family doctor if the patient's status for heart disease changes with updated parameters.

2 | DATA

2.1 Description

The data used to develop the model was obtained from the University of California Irvine (UCI) Machine Learning Repository. The data consists of 918 patients' medical records, which includes demographic information such as the patient's age (Age) and sex (Sex), along with measures of the patient's vitals at a given time. These measurements include the patient's chest pain type (ChestPainType), their resting blood pressure (RestingBP), their maximum heart rate (MaxHR) their cholesterol level (Cholesterol), their fasting blood pressure (FastingBP), the status of their resting electrocardiogram results (RestingECG), whether they have exercise angina (ExerciseAngina), the ST depression induced by exercise relative to rest (Oldpeak), the slope of the peak exercise ST segment (ST_Slope), and whether they have heart disease or not (heart disease).

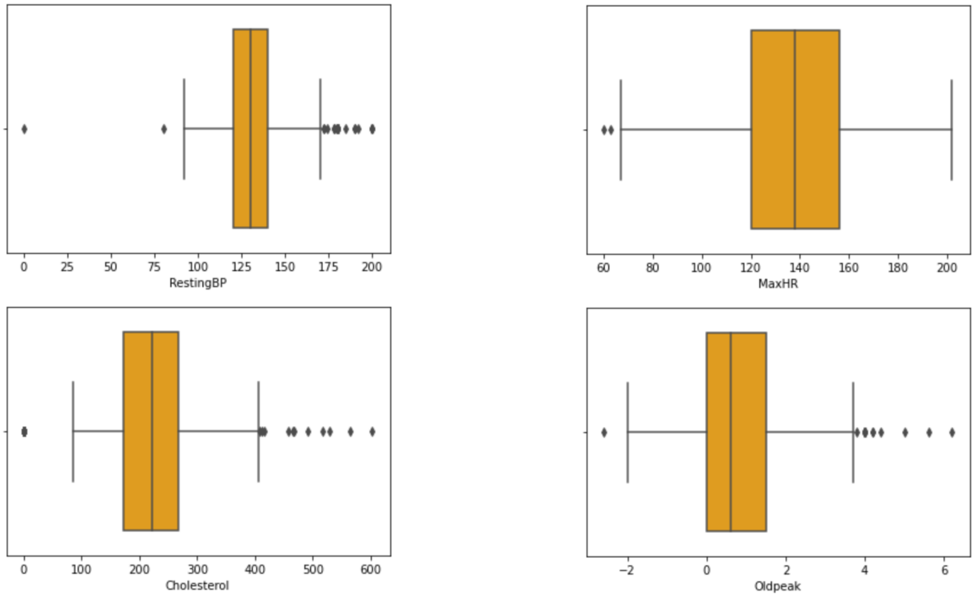
For non-categorical data, summary statistics can be viewed to gain a better understanding:



	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

2.2 Cleaning

The first step in cleaning the data is to check for null values. There are no null values in this dataset. Next, there is a check for duplicate data. There is no duplicate data. Finally, there is a check for outliers using boxplots.

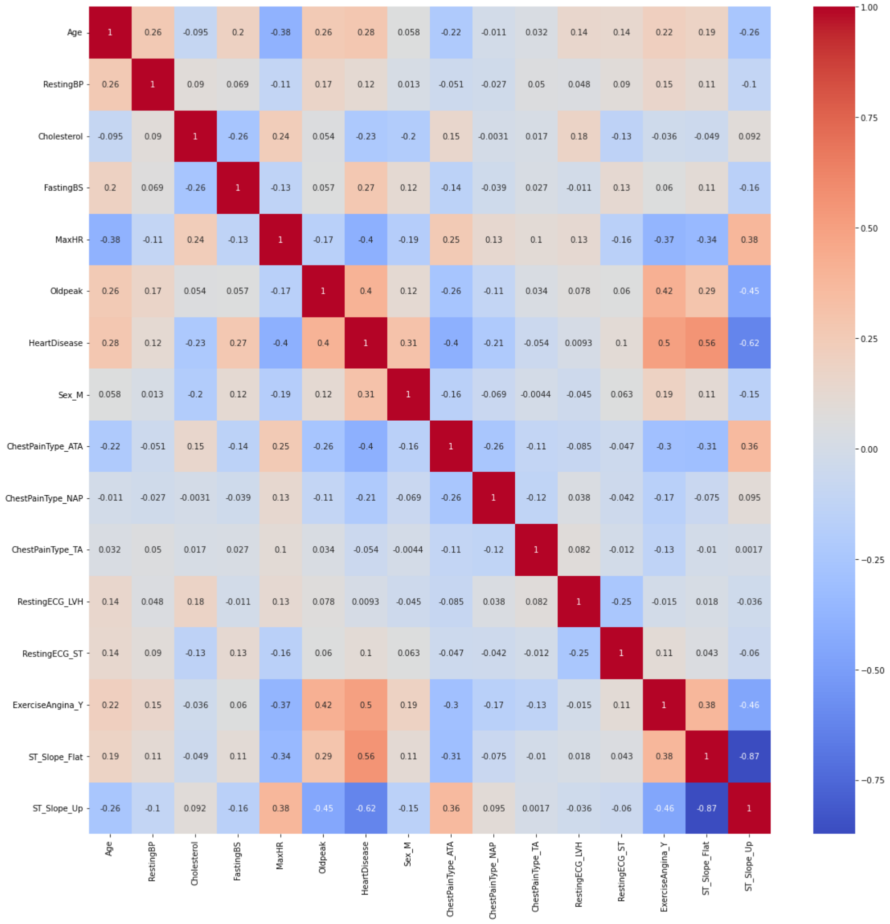


The following columns have outliers: RestingBP, Cholesterol, MaxHR, and Oldpeak. The outliers shown in the boxplots were treated using the InterQuartile Range (IQR) method. Quantile-based flooring and capping were used with a 10% percentile flooring and a 90% percentile capping. The data with outliers were simply removed from the dataset, rather than using a replacement method (ie. such as replace with mean, etc.). Given the problem being addressed is one related to healthcare and the outcome of this prediction model will be used to predict heart disease, it is important to keep the risk of misclassification low and therefore, replacement of outlier data was not considered.

The final step to data cleaning is introducing dummy variables for categorical data. The following variables are categorical: Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope. The base columns were dropped to provide a reference variable in the model. In essence, females will be the reference variable for Sex, atypical angina will serve as the reference variable for ChestPainType, etc. The data cleaning process is now complete and the data is ready to be used in model development.

2.3 Correlation Matrix

The following correlation matrix produced can be used for exploratory purposes. The correlation matrix provides initial insights into the types of patient vitals highly correlated with the risk of heart disease. These correlations can validate the significance of including certain independent variables in the machine learning models. For instance, it can be observed that ST_Slope_Flat is the most correlated with heart disease across all independent variables, with a correlation of 0.55. Furthermore, it can be observed that some patient vitals are negatively correlated with heart disease, such as various chest pain types and electrocardiogram results. These provide intuition with regard to how various independent variables will interact in the models in terms of predicting the risk of heart disease in a patient.



3 | MACHINE LEARNING MODELS

Machine learning is a type of artificial intelligence that uses statistical techniques to give computer programs the ability to learn from their past experiences and improve for future performances. Machine learning involves fitting models into previously observed data to make predictions and enhance decision-making. There are two types of machine learning: (1) supervised machine learning and (2) unsupervised machine learning. In supervised machine learning, a set of labeled data is proved for the system to learn and replicate; whereas, with unsupervised learning, the system is provided with a set of unlabeled data with unknown insights and is allowed to explore the data and draw inferences. This can be learned by clustering, density estimation, anomaly detection, and principal component analysis. There is also semi-supervised learning and reinforcement learning. For this model, supervised machine learning is the only type used.

The data was split into x training, x test, y training, and y test data. We split the data into train and test so that the training model can learn an effective mapping of inputs to outputs and the test model can effectively evaluate the model performance. For all the models, 30% was assigned to a test set and 70% to a train set. A random state of 42 was assigned to ensure that the train and test sets were evaluated in the same way as they are representatives of the main dataset. This will allow for a better performance mapping of the data and for making predictions with the final model.

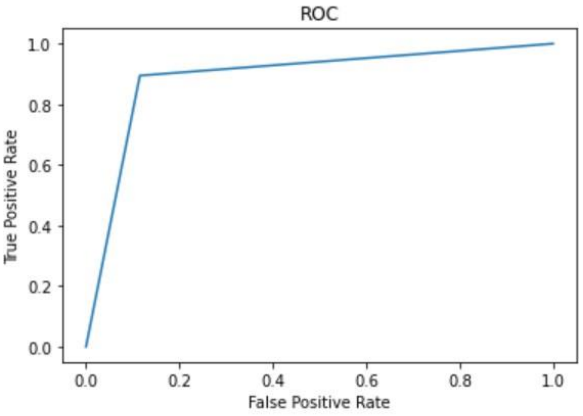
For the models, the machine learning models were scaled as the algorithm calculates the distance between data points such as KNN in order to make an assumption. Therefore, the train and test sets were normalized. Scaling is a technique that generalizes data points in order to reduce the distance between the data points. Variables are assigned weights according to their data points so if the distance is too high, the weights would be larger and this creates uncertainty in the final result of the model. An algorithm has to learn to identify common attributes which reduce processing time in the future and increase positive results. The two types of scaling are normalization and standardization. Normalization adjusts all the values on a common scale between the range of 0 to 1 while standardization adjusts the values to be centered about the mean where the mean of the data point will be zero and the standard deviation will be 1. The difference is that normalization uses a minimum and maximum value for the scaling while standardization uses the mean and standard deviation.

3.1 Naïve Bayes Classifier

Naïve Bayes is based on Bayes Theorem and assumes independence amongst predictors. As a result of this assumption, it performs better than other models such as logistic regression and requires less training set. The Gaussian naïve Bayes classifier was used in this specific model.

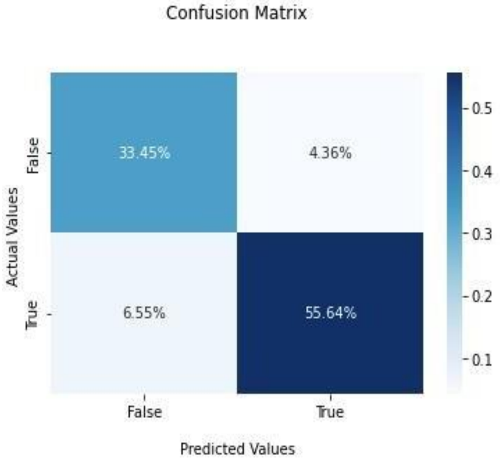
The outcome of the confusion matrix shows that the true positive rate is 55.64%, the true negative rate is 33.45%, the false negative rate (Type 1 error) is 6.55%, and the false positive rate (Type 2 error) is 4.36%. This means that the model correctly predicts the presence of heart disease 55.64% of the time, the model correctly predicts the absence of heart disease 33.45% of the time, the model incorrectly predicts the presence of heart disease 6.55% of the time, and the model incorrectly predicts the absence of heart disease 4.36% of the time.

The accuracy score is 0.8909 which means the model predicts 89.091% of the heart disease outcomes correctly. The precision score is 0.9373 which means 93.73% of the predictive positive identifications by the model are actually correct. The sensitivity is 0.8947, which means that 89.47% of actual positives by the model are correctly identified. The specificity score is 0.8847 which means that 88.47% of actual negatives are correctly identified by the model. The F1 score, which is the harmonic mean of precision and recall, is 0.9107. The ROC describes the relationship between the model's sensitivity versus specificity. The result shows the AUC to be 0.8897. The model shows an overlapping of the true positive rate and false-negative rate.



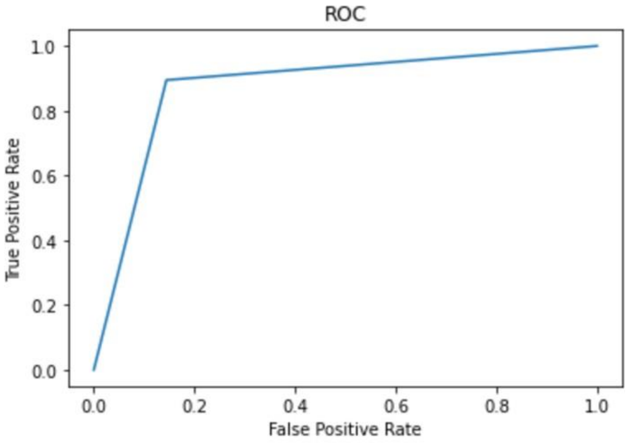
```
roc_auc_score:  0.8896761133603239
```

Confusion Matrix

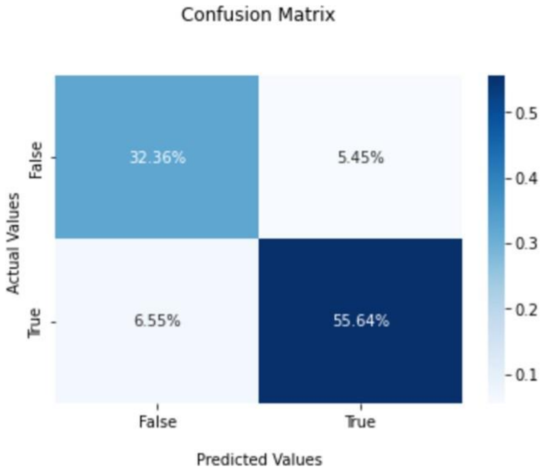


3.2 Logistic Regression

The data was also analyzed using logistic regression, as it is one of the most common machine learning models. Logistic regression is a model that is used to predict the probabilities of a binary outcome, by finding a linear curve solution to the data. While this model has its benefits, it has some serious real-world scenario drawbacks. Firstly, as mentioned, it can only find linear curve solutions, which may not be the case in many real-life scenarios. Secondly, it makes assumptions in terms of the distributions of the data. Finally, it assumes independent variables to be mutually independent, meaning that if there is a potential relationship between independent variables, the results are at risk of being skewed. So with this understanding of the advantages and disadvantages of logistic regression in place, it was important to see how successfully the model could predict the relationship between heart disease and the independent variables. The logistic regression shows high accuracy and F1 scores (0.88 and 0.90 respectively), which show the strength of the model in terms of precision and recall. This is further confirmed when looking at the ROC below, and the AUC score of 0.88, showing strong specificity and sensitivity. When looking at the confusion matrix, as stated earlier, the most important area to focus on is the false negative. While a value of 6.55% is fairly low, based on the context of the model in its relation to a healthcare-related issue, ideally this number should be lower. The confusion matrix shows high accurate values of true positives and true negatives of 55.64% and 32.36%, and a low false-positive rate of 5.45%, which further indicates the strength of this model.

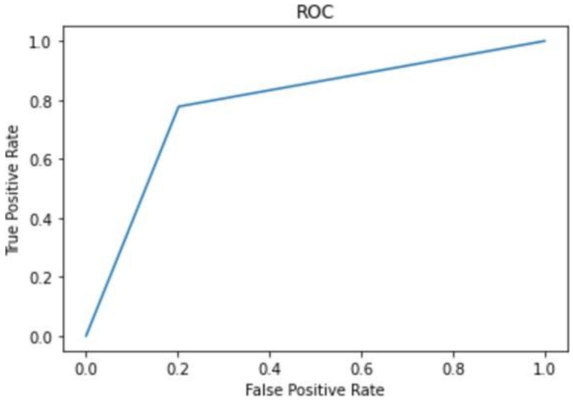


```
roc_auc_score:  0.8752530364372471
```

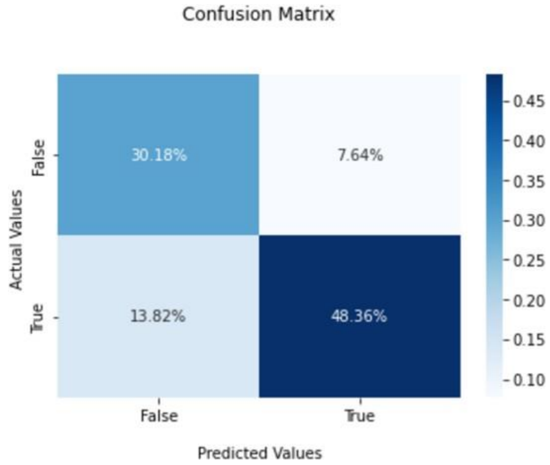


3.3 K-Nearest Neighbours

The next model explored was the K-Nearest Neighbours (KNN) algorithm. KNN works by analyzing the dataset, assuming the test datapoint to be similar, and then looking for K neighbors to determine a prediction. KNN differs from logistic regression because it supports non-linear solutions and is also a non-parametric model. The accuracy and F1 scores were much lower when compared with the previously discussed models (0.79 and 0.82), and the same can be said for the AUC score (0.79). When looking at the confusion matrix, it further clarifies the poorer quality of this model, especially with regard to false negatives, which at 13.82%, is more than double that of the logistic regression model. Additionally, the true positives and true negatives are lower than of previous models, with scores of 48.36% and 30.18% respectively. Furthermore, the false-positive rate of 7.64% is higher when compared to other models. Overall, the lower scores show the weakness of this model when evaluating the performance in terms of prediction, and it is likely due to the simpler nature of the KNN algorithm in the field of predictive analysis.

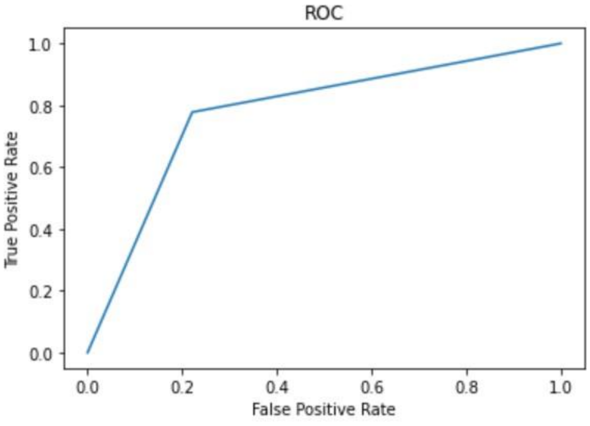


```
roc_auc_score:  0.7879273504273505
```



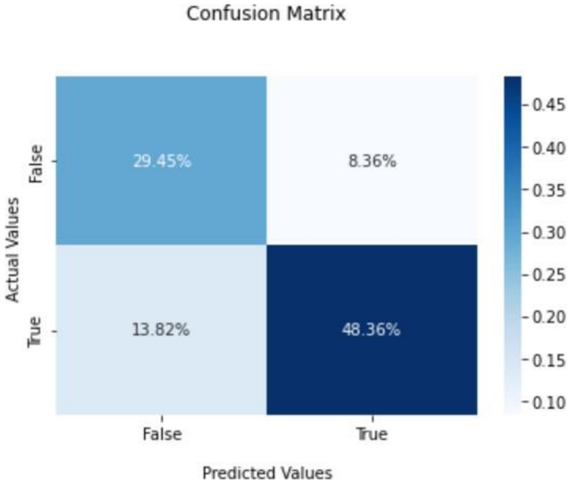
3.4 Decision Tree

Another conventional model incorporated into the analysis was the decision tree classification model. Like other models, the decision tree model uses various input variables in order to predict the value of a target variable, which in this case would be a patient's risk of heart disease. This model works by breaking data down into smaller subsets based on the outcome of a decision until a prediction can be made. The resulting output is a tree structure to specify consequences from a decision given an input. As previously mentioned, the false-negative rate was an important factor when considering which model to move forward with. It was important that this number be minimized as the consequences of a misdiagnosis when there is indeed a risk of heart disease are dire. Out of the models examined thus far, the decision tree classification model has the highest false-negative rate of 8.36%. Furthermore, it had the lowest specificity and accuracy scores of 29.45% and 77.82%. Therefore, this model is relatively weak and should not be considered.



```
roc_auc_score:  0.7783119658119658
```

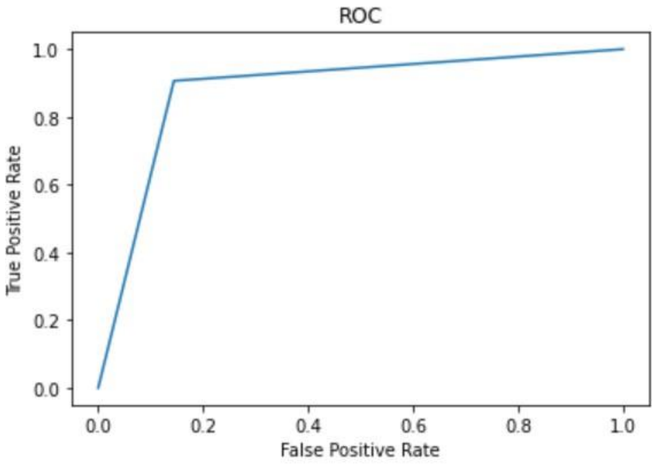
Confusion Matrix



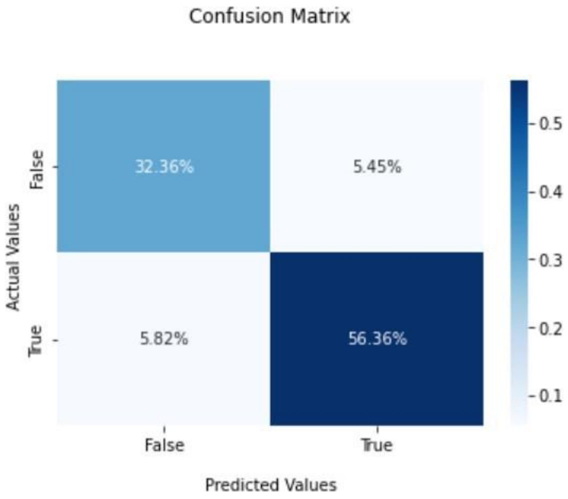
3.5 Random Forest

A random forest model is one in which the prediction is based on the individual prediction of many uncorrelated decision trees. The effectiveness of this model lies in the "uncorrelated" feature among a large number of trees. With this characteristic present in the forest, individual trees can mitigate each other's errors and provide a more robust model. In essence, it uses bagging and randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. The random forest model is initially fit on a training data set, and then the fitted model is used to predict the responses for the observations in the test data set in order to examine the model's accuracy.

The random forest model shows high accuracy and F1 scores (0.8873 and 0.9091 respectively), which indicate the strength of the model in terms of precision and recall. The AUC score of 0.881 is quite high and suggests this model be promising. The outcome of the confusion matrix showed that the true positive rate is 56.44%, the true negative rate is 32.36%, the false-negative rate is 5.82%, and the false-positive rate is 5.45%. Overall, this means the model incorrectly predicts the presence of heart disease 5.45% of the time, and the model incorrectly predicts the absence of heart disease 5.82% of the time. Compared to previous models explored, the random forest model appears to be the most reasonable, especially because the false-negative rate is very low and provides assurance that there will rarely be cases of heart disease that are not predicted.



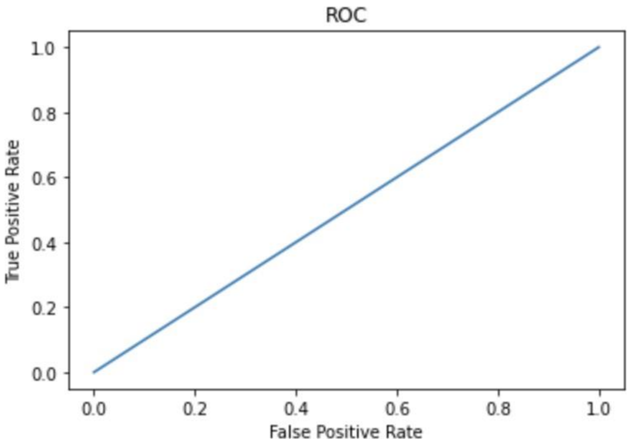
```
roc_auc_score:  0.8811009896536213
```

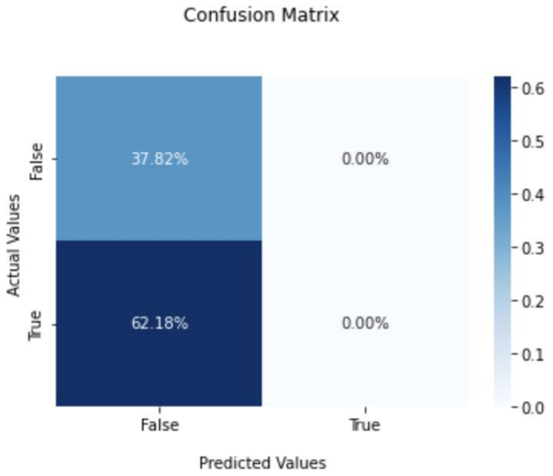


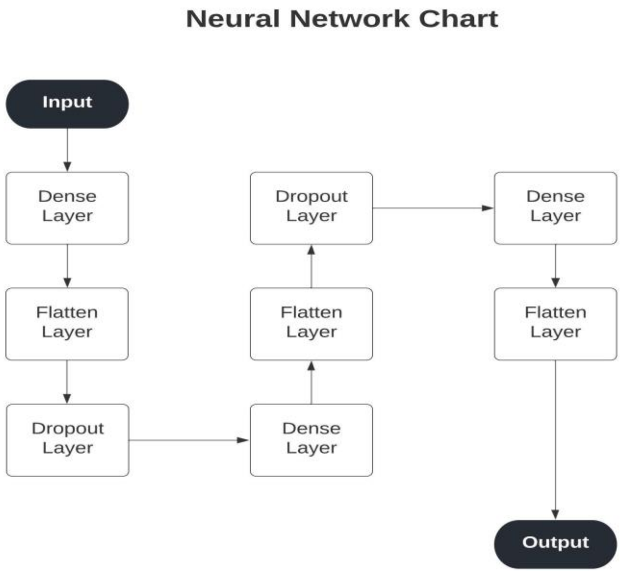
3.6 Neural Network

In order to build up the neural network model after configuring the data set, it is essential to first determine the layers of the neural network to set the network architecture. After this, it is important to train the neural network initially on a training data set. The fitted model is then used to predict the responses for the observations in the test data set to examine the accuracy of the model.

The accuracy score of 0.378 means that the model correctly predicts the status of heart disease 37.8% of the time. The outcome of the confusion matrix shows that the true positive rate is 0.00%, the true negative rate is 0.00%, the false-negative rate is 37.82%, and the false-positive rate is 62.18%. Therefore, the model incorrectly predicts the presence of heart disease 0.00% of the time, and the model incorrectly predicts the absence of heart disease 62.18% of the time. These values are concerning and it can be concluded that this model is not good at predicting the risk of heart disease in patients and should therefore not be considered or pursued any further.






Neural Network Chart

3.7 Summary of Model Outputs & Decision

Several models were examined to understand how to best predict the risk of heart disease given a patient's vitals. Ultimately, the random forest model was used as it carried the lowest false-negative rate. False-negative scores were considered important to the model as there are severe consequences to a patient's livelihood if there is inaction when there is heart disease present. Thus, it was of high importance that this number be minimized when choosing the model. Furthermore, the random forest also produced the highest sensitivity, which means it was the best at correctly predicting the risk of heart disease when there was one present. Additionally, this model resulted in a relatively high accuracy, which means it is a reasonable model with regard to correctly predicting an outcome for the dependent variable (HeartDisease).

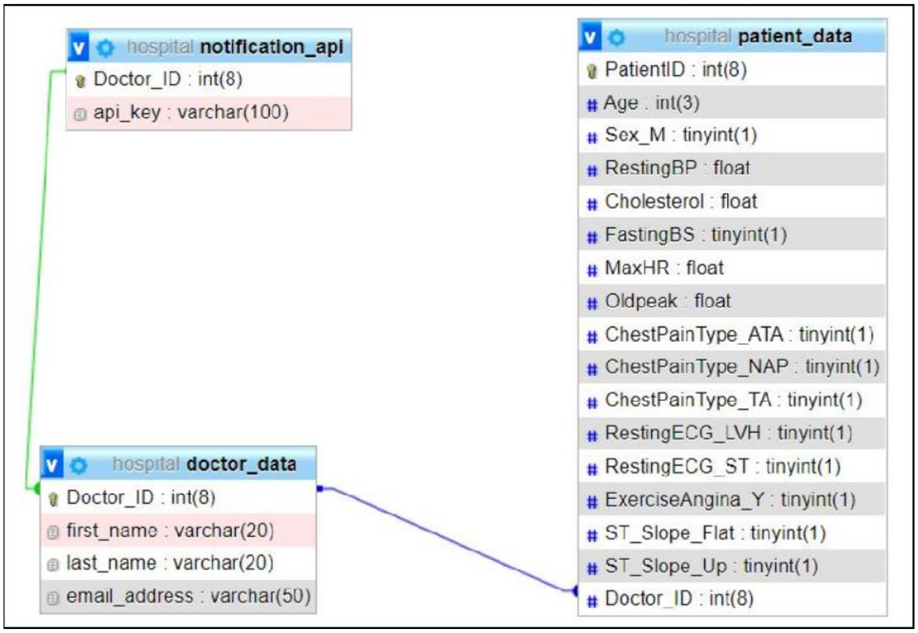
	Accuracy	Specificity	Sensitivity	False Negative
Naïve Bayes	89.09%	33.45%	55.64%	6.55%
Logistic Regression	88%	32.36%	55.64%	5.45%
KNN	78.55%	30.18%	48.36%	7.64%
Decision Tree	77.82%	29.45%	48.36%	13.82%
Random Forest	**88.73%**	**32.36%**	**56.36%**	**5.82%**
Neural Network	37.82%	37.82%	0%	62.18%

4 | DATABASE DESIGN & CONNECTION

A significant addition to this project made from the previous one is creating a SQL database that connects the current model with an updating database. This database provides many things to this model, but the most crucial aspect is the ease of information sharing and future scalability. When examining the best way to introduce the heart health prediction model, the most challenging part was finding accessible data. In addition to this difficulty finding data, research has shown that hospitals have been struggling to effectively create and share data with all aspects of the hospital (i.e. departments, stakeholders, etc.). This database will allow nurses or any other healthcare worker to update patients' records effectively and have those records be accessible to everyone in the hospital once they are updated.

The SQL database was created through a simple design. The database was created using SQL and is connected to using Python code. This connection allows specific Python codes to run through the database for manipulating and pulling values. The focus is to allow the swift recording of patient information and have this data run against the model in real-time. The database also contains the physician notification information, which is needed to send the notification if the new data reveals the patient is considered to be at risk of heart disease. The importance of this is seen when inputting the data that contains different health markers for the patient. The creation of this database is essential compared to just a spreadsheet that includes the patient data as this allows for the proper relations to be structured and scalability for the database that will enable it to continue to grow in the future.

As the data set grows and the hospital gets more information as time goes on, they will be able to update the model and add variables to it as they increase their data. They can also rerun the model on their new data to better understand the actual health implications of specific markers. The SQL database makes this process easier and will allow the hospital to collect and analyze its data instead of relying on physical documents and data gathered from other sources.



hospital notification_api
Doctor_ID : int(8)
api_key : varchar(100)

hospital doctor_data
Doctor_ID : int(8)
first_name : varchar(20)
last_name : varchar(20)
email_address : varchar(50)

hospital patient_data
PatientID : int(8)
Age : int(3)
Sex_M : tinyint(1)
RestingBP : float
Cholesterol : float
FastingBS : tinyint(1)
MaxHR : float
Oldpeak : float
ChestPainType_ATA : tinyint(1)
ChestPainType_NAP : tinyint(1)
ChestPainType_TA : tinyint(1)
RestingECG_LVH : tinyint(1)
RestingECG_ST : tinyint(1)
ExerciseAngina_Y : tinyint(1)
ST_Slope_Flat : tinyint(1)
ST_Slope_Up : tinyint(1)
Doctor_ID : int(8)

5 | SYSTEM

The system is designed in Python and in addition to the previous project, it now has an integrated SQL database, a web app with PywebIO library, as well as a notification system that sends an alert to the patient's family doctor using the PushBullet API and app.

The system is designed for healthcare representatives who are responsible for updating patients' test records using a website frontend. The representative logs in using their credentials and enters the patient ID for the patients whose medical records need updating. They enter the parameter(s) to update and change the value(s) accordingly. The system then takes these new values with previous ones, runs them into the machine learning model, and if the model predicts that there is a risk of heart disease, then the system retrieves the patient's family doctor’s information and sends out an alert to the doctor’s mobile to notify them that their patient has a risk of heart disease. If the model predicts that the patient is not at risk, then only the values are updated.

The system is a design, and a work in progress, on how to apply machine learning models in a real-world environment and we are continuously working on it to improve it in each iteration.