HeartScan: An Application Utilizing Machine Learning Algorithms to Predict Heart Disease

Dev Suri

May 17, 2020

1 Abstract

This paper details a project that uses an investigation of the trends and correlations between an individual's health information and their presence of heart disease, based on data taken from the Hungarian Institute of Cardiology, University Hospitals in Switzerland, etc., to create a model that predicts future developments of heart disease. This can be done through machine learning algorithms to create a model for finding such trends and correlations and an android app to gather an individual's health information. This project allows any individual to get a keener sense of where they stand in terms of their risk of developing heart-related illnesses in the future.

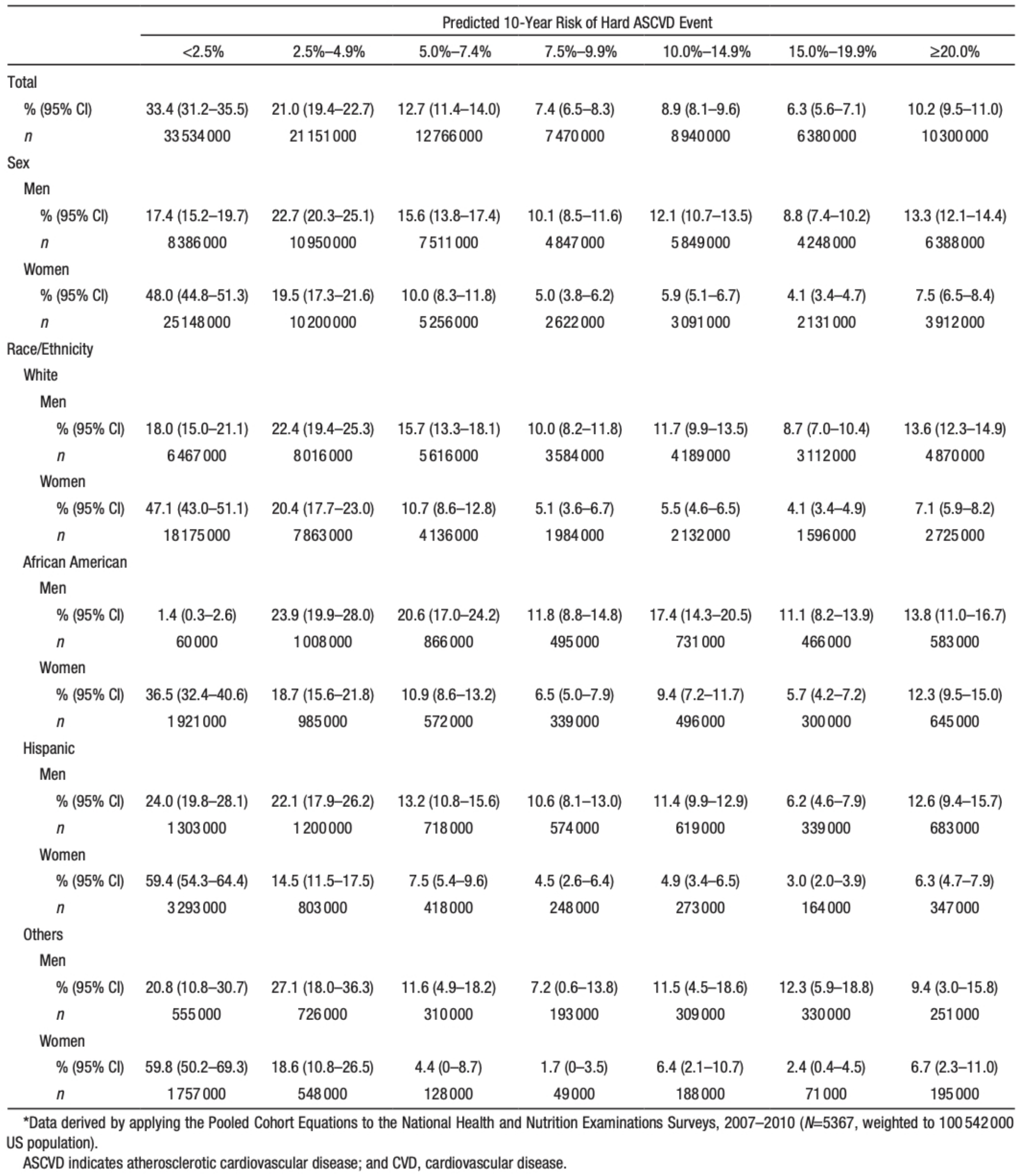
2 Introduction

As it stands today, heart disease is the number one leading cause of death in the United States, surpassing even cancer. Commonly used existing methods of predicting the risk of heart disease developing in individuals do not utilize machine learning. This project employs machine learning algorithms and techniques to improve on the currently existing models.

Section 1 provides context and background on how existing methods of prediction work. Following it, section 2 discusses the machine learning algorithms and data processing techniques that were used to develop an improved model for predicting future heart disease. Then, section 3 discusses the process by which the model was created and implemented into an android application. Finally, section 4 details future work to be done and potential improvements to the model.

3 Background

Currently, the prevailing method of determining an individual's risk of heart disease involves using guidelines from the American Heart Association's Guideline on the Assessment of Cardiovascular Risk. While the AHA's guidelines are by no means deficient or incomprehensive, some of their guidelines involve distributions of probability separated through strict cut-offs that take into account as factors, age and gender (See Figure 1).



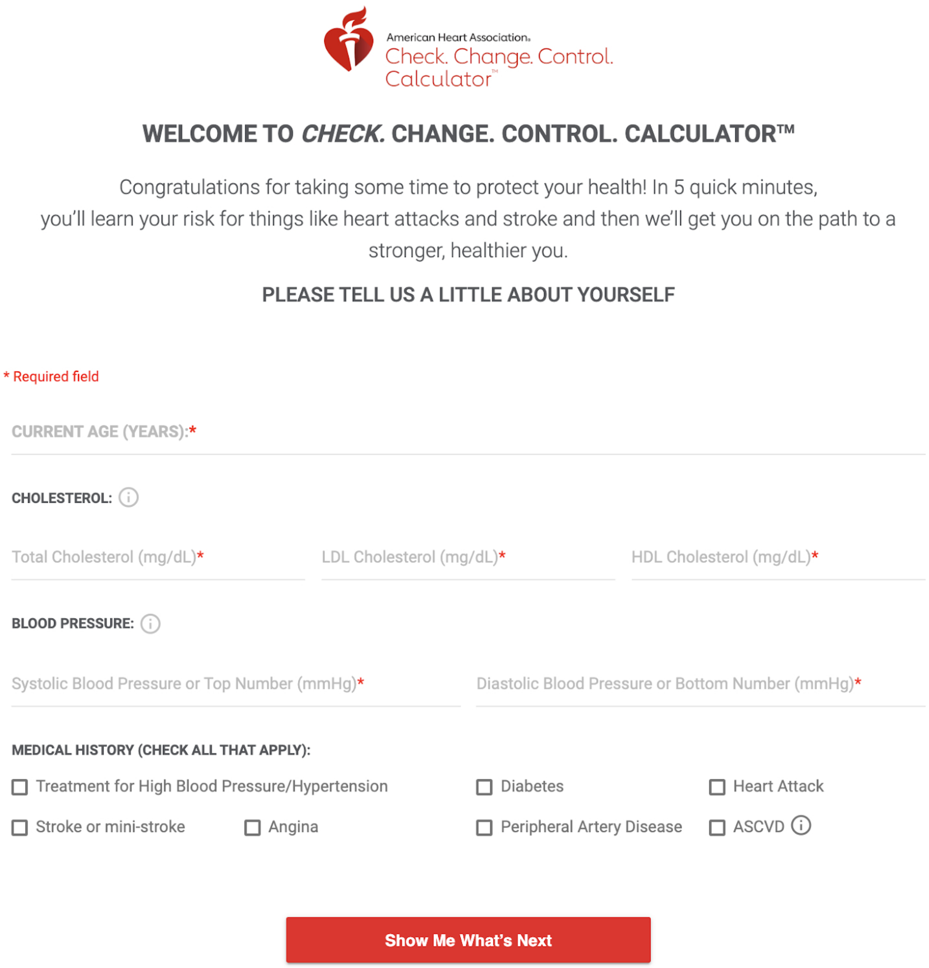
	Predicted 10-Year Risk of Hard ASCVD Event						
	<2.5%	2.5%–4.9%	5.0%–7.4%	7.5%–9.9%	10.0%–14.9%	15.0%–19.9%	≥20.0%
Total							
% (95% CI)	33.4 (31.2–35.5)	21.0 (19.4–22.7)	12.7 (11.4–14.0)	7.4 (6.5–8.3)	8.9 (8.1–9.6)	6.3 (5.6–7.1)	10.2 (9.5–11.0)
n	33 534 000	21 151 000	12 766 000	7 470 000	8 940 000	6 380 000	10 300 000
Sex							
Men							
% (95% CI)	17.4 (15.2–19.7)	22.7 (20.3–25.1)	15.6 (13.8–17.4)	10.1 (8.5–11.6)	12.1 (10.7–13.5)	8.8 (7.4–10.2)	13.3 (12.1–14.4)
n	8 386 000	10 950 000	7 511 000	4 847 000	5 849 000	4 248 000	6 388 000
Women							
% (95% CI)	48.0 (44.8–51.3)	19.5 (17.3–21.6)	10.0 (8.3–11.8)	5.0 (3.8–6.2)	5.9 (5.1–6.7)	4.1 (3.4–4.7)	7.5 (6.5–8.4)
n	25 148 000	10 200 000	5 256 000	2 622 000	3 091 000	2 131 000	3 912 000
Race/Ethnicity							
White							
Men							
% (95% CI)	18.0 (15.0–21.1)	22.4 (19.4–25.3)	15.7 (13.3–18.1)	10.0 (8.2–11.8)	11.7 (9.9–13.5)	8.7 (7.0–10.4)	13.6 (12.3–14.9)
n	6 467 000	8 016 000	5 616 000	3 584 000	4 189 000	3 112 000	4 870 000
Women							
% (95% CI)	47.1 (43.0–51.1)	20.4 (17.7–23.0)	10.7 (8.6–12.8)	5.1 (3.6–6.7)	5.5 (4.6–6.5)	4.1 (3.4–4.9)	7.1 (5.9–8.2)
n	18 175 000	7 863 000	4 136 000	1 984 000	2 132 000	1 596 000	2 725 000
African American							
Men							
% (95% CI)	1.4 (0.3–2.6)	23.9 (19.9–28.0)	20.6 (17.0–24.2)	11.8 (8.8–14.8)	17.4 (14.3–20.5)	11.1 (8.2–13.9)	13.8 (11.0–16.7)
n	60 000	1 008 000	866 000	495 000	731 000	466 000	583 000
Women							
% (95% CI)	36.5 (32.4–40.6)	18.7 (15.6–21.8)	10.9 (8.6–13.2)	6.5 (5.0–7.9)	9.4 (7.2–11.7)	5.7 (4.2–7.2)	12.3 (9.5–15.0)
n	1 921 000	985 000	572 000	339 000	496 000	300 000	645 000
Hispanic							
Men							
% (95% CI)	24.0 (19.8–28.1)	22.1 (17.9–26.2)	13.2 (10.8–15.6)	10.6 (8.1–13.0)	11.4 (9.9–12.9)	6.2 (4.6–7.9)	12.6 (9.4–15.7)
n	1 303 000	1 200 000	718 000	574 000	619 000	339 000	683 000
Women							
% (95% CI)	59.4 (54.3–64.4)	14.5 (11.5–17.5)	7.5 (5.4–9.6)	4.5 (2.6–6.4)	4.9 (3.4–6.5)	3.0 (2.0–3.9)	6.3 (4.7–7.9)
n	3 293 000	803 000	418 000	248 000	273 000	164 000	347 000
Others							
Men							
% (95% CI)	20.8 (10.8–30.7)	27.1 (18.0–36.3)	11.6 (4.9–18.2)	7.2 (0.6–13.8)	11.5 (4.5–18.6)	12.3 (5.9–18.8)	9.4 (3.0–15.8)
n	555 000	726 000	310 000	193 000	309 000	330 000	251 000
Women							
% (95% CI)	59.8 (50.2–69.3)	18.6 (10.8–26.5)	4.4 (0–8.7)	1.7 (0–3.5)	6.4 (2.1–10.7)	2.4 (0.4–4.5)	6.7 (2.3–11.0)
n	1 757 000	548 000	128 000	49 000	188 000	71 000	195 000

*Data derived by applying the Pooled Cohort Equations to the National Health and Nutrition Examinations Surveys, 2007–2010 (*N*=5367, weighted to 100 542 000 US population).

ASCVD indicates atherosclerotic cardiovascular disease; and CVD, cardiovascular disease.

Figure 1: The diagnostic method of the American Heart Association that predicts the chance of a major heart incident occurring in the next ten years in an individual using linear cutoff values.

They also have an online application called "Check. Change. Control. Calculator", however, this calculator only takes into account age, blood pressure, and cholesterol levels. They do not have a more complex model that can simultaneously take into account a greater number of factors (See Figure 2).



American Heart Association.
Check. Change. Control.
Calculator™

WELCOME TO *CHECK.* CHANGE. CONTROL. CALCULATOR™

Congratulations for taking some time to protect your health! In 5 quick minutes, you'll learn your risk for things like heart attacks and stroke and then we'll get you on the path to a stronger, healthier you.

PLEASE TELL US A LITTLE ABOUT YOURSELF

* Required field

CURRENT AGE (YEARS):*

CHOLESTEROL: ⓘ

Total Cholesterol (mg/dL)*   LDL Cholesterol (mg/dL)*   HDL Cholesterol (mg/dL)*

BLOOD PRESSURE: ⓘ

Systolic Blood Pressure or Top Number (mmHg)*   Diastolic Blood Pressure or Bottom Number (mmHg)*

MEDICAL HISTORY (CHECK ALL THAT APPLY):

□ Treatment for High Blood Pressure/Hypertension   □ Diabetes   □ Heart Attack

□ Stroke or mini-stroke   □ Angina   □ Peripheral Artery Disease   □ ASCVD ⓘ

Figure 2: The features the American Heart Association's Check. Change. Control. Calculator takes into account.

4 Development and Techniques

The objective of this project was to build upon the insufficiencies laid out in the background section in existing heart disease prediction models and utilize machine learning techniques to create a more advanced model. Considering this, I have separated the process of creating this project into the steps that follow.

4.1 Technologies Used

All machine learning and model creation was done in python 3, using a package called Scikit Learn, a versatile and fast machine learning library. The IDE used was google colaboratory, an online IDE with access to cloud GPU's and TPU's which allows for machine learning calculations to be computed quickly and efficiently. For the android application, android studio was used to code the app, and the language used was Java.

4.2 Data Collection and Processing

To begin, I decided to use a dataset consisting of various health and vital measurements for patients collected from several hospitals and cardiology institutions to create my model for predicting heart disease. The dataset contains vast information about past patients - their resting heart rates, blood pressures, cholesterol levels, presence of certain heart conditions, chest pain, age, gender, and most importantly, whether they had heart disease. I chose this dataset for two reasons, one being that datasets containing information on the heart-related statistics I needed to create this model are rare, and two, that even in most of these datasets, the proportion of patients who have heart disease is low compared to those who do not. In order to have successful training for my model on this dataset, I needed data that had a high proportion of individuals with heart disease as well. Ultimately, due to the lack of availability of such data, and the considerations I mentioned earlier, I settled on using a dataset from the UCI Maching Learning Repository, which can be found here. The dataset contains statistics about past patients with heart disease collected from the following institutions by the following doctors:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.

2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Although the dataset is rather small, with about only 303 total entries, the proportion of patients that have heart disease to those that do not is high, i.e. 54 percent of all patients in the dataset have heart disease. This makes it suitable to train a model on. See Figure 2.5 for a series of histograms depicting the features and fields of the dataset. In total there are 14 fields, 13 inputs and 1 output, the output being whether the patient had heart disease or not. This output is represented in a binary fashion with 1 indicating presence and 0 indicating absence. Other fields in the dataset may deviate from this, however, and use 1-4 for indicating various types of presence as opposed to solely 1.

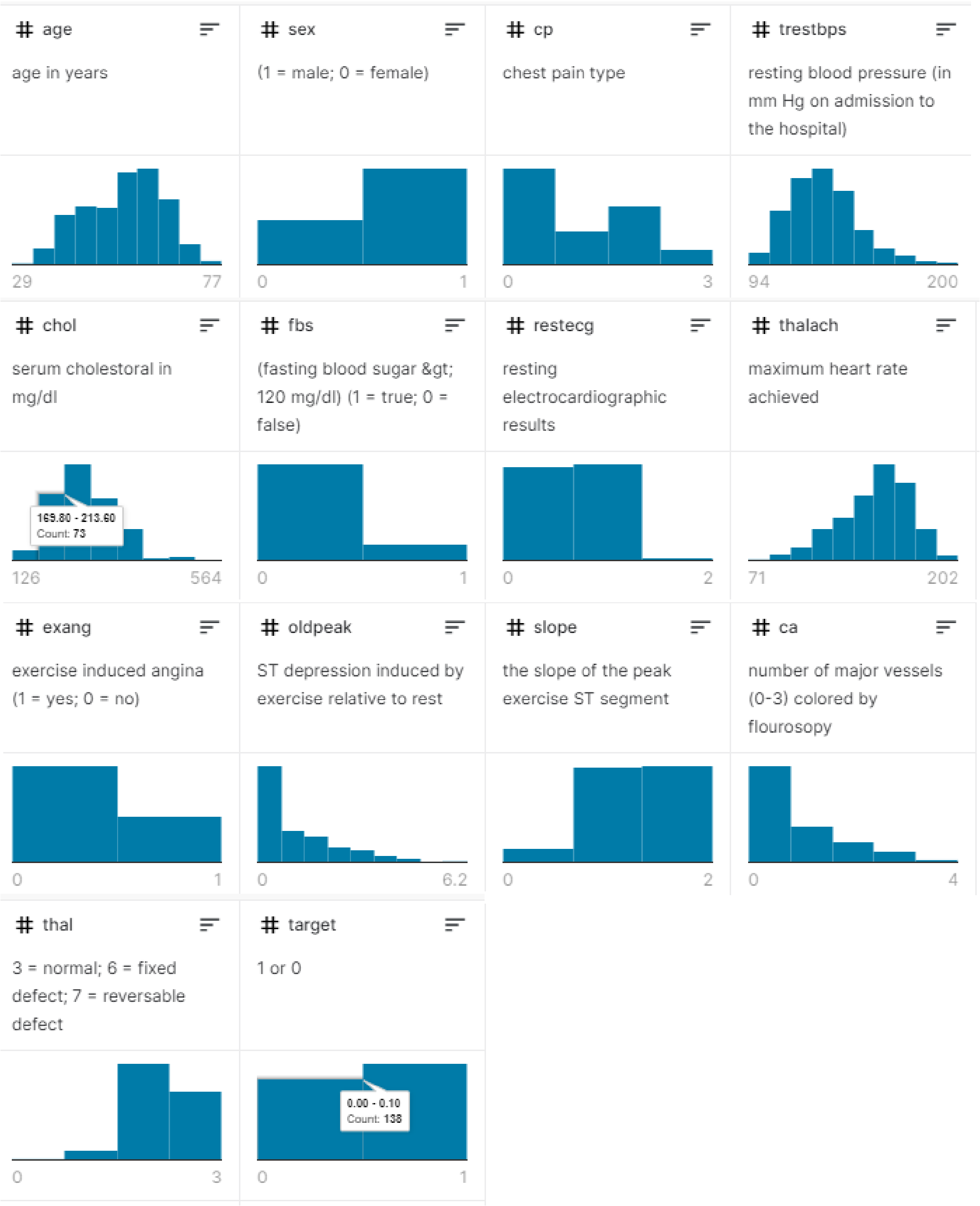


Figure 3: Histograms showcasing all fields of the dataset.

I first created a correlation matrix to analyze the dataset to see if there were any obvious correlations between variables (see Figure 3). A few variables that were comparatively strong in indicating presence of heart disease (r=0.40) were whether the patient had thalassemia (a blood disorder causing reduced production of hemoglobin) and chest pain; however, an r-value of 0.40 was not sufficient enough for these fields to be given special consideration in the creation of my model, so I proceeded to prepare the dataset for training on models.

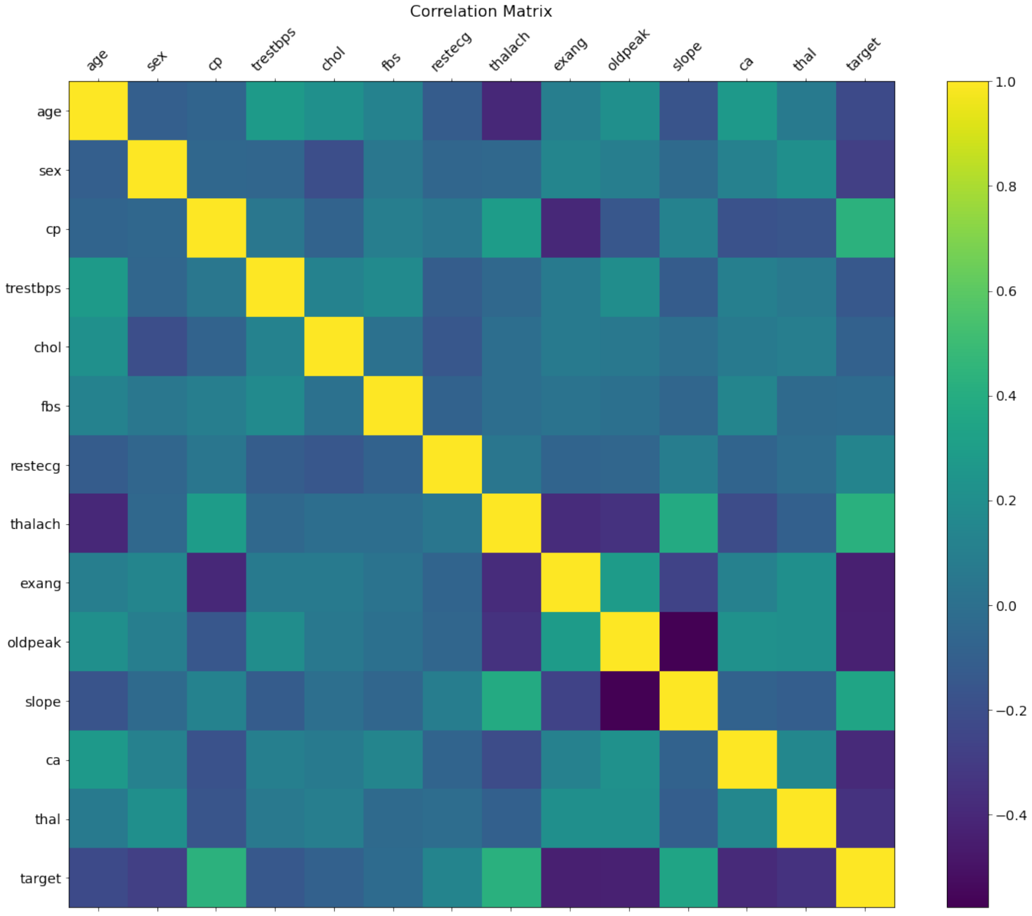


Figure 4: The correlations between all 14 fields displayed using a color matrix.

Certain fields in the dataset, such as age, can vary quite a bit. Age, for example, in this dataset varies from 29 to 77. This difference in age can affect ML (machine learning) model performance, so I used a method from the Scikit Learn API called StandardScaler. StandardScaler scales all data to within two standard deviations of the mean, and ensures the data is a value represented by a number in the range [-1,1]. Using StandardScaler thus creates less deviation in the data and makes it easier for the model to train on it. This technique was used on all fields that were not binary or represented by integers from 0 to 4.

The dataset was also split into a ratio of 80% training set to 20% test set. All models were trained on the training set and accuracy tests for the models were done on the test set.

4.3 Application of Machine Learning

With the dataset processing finished, I created various models in python for training on the dataset.

4.3.1 Support Vector Machine

The first model created is a support vector machine (SVM). A support vector machine attempts to create a hyper-plane in N dimensions, where N is the number of features, that classifies as many points as possible correctly by separating them using the hyper-plane. The SVM Model I have created uses a sigmoidal function (sigmoid) kernel to train on the training set. The SVM's accuracies with the sigmoid kernel on the training set and test set were 0.805 and 0.820, respectively.

4.3.2 Naive Bayes Classifier Using Gaussian Methods

The second model created is a naive bayes classifier. The Naive Bayes Classifier (NBC) relies on Bayes' Theorem of conditional probability. Given a vector of x features, Naive Bayes calculates the probability of the vector belonging to a certain class, in this case 0 or 1, heart disease absence and presence, respectively. NBC assumes that features are independent of each other, that is, that one input field has no affect on the other. This is obviously not true with the dataset, for example, maximum heart rate is primarily correlated with age. The older one gets, the lower maximum heart rate they can sustain. But, it is necessary that this assumption is made for the NBC to be able to train on the dataset. Fortunately, as shown earlier by the correlation matrix, there is not such a strong correlation between any two fields that would nullify the use of Bayesian methods. There are several different types of Naive Bayes Classifier's one can use in the scikit learn package, such as Multinomial, Complement, and Bernoulli, however, I chose to use a Gaussian NBC. This just means that the model uses the Gaussian Method of calculating likelihood in its probability determination calculations. The accuracy of my Naive Bayes Classifier was 0.835 and 0.852 on the training and test set, respectively.

4.3.3 XGBoost

XGBoost stands for extreme gradient boosting. It is a model that uses the gradient descent algorithm to minimize error. Simple gradient boosting was used as opposed to stochastic and regularized gradient boosting. The accuracy of this model, on the train and test set, 0.992 and 0.852, respectively.

4.3.4 K-Neighbors Classifier

K-Nearest Neighbors is a machine learning algorithm that attempts to classify data based on neighboring data. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors, where k is a positive integer, typically small. For example, if k = 1, then the object is simply assigned to the class of that single nearest neighbor, and if k=10, the highest frequency of its ten nearest neighbors classifications is used to determine the class the point belongs to. The K-Neighbors Classifier Model had accuracies on the train and test sets of 0.860 and 0.836, respectively.

4.3.5 Random Forest Classifier

Random Forest Classifiers are models constructed of many individual decision trees. Decision Trees by themselves tend to overfit to data, however, by putting many of them into a random forest model, the overfitting problem is negated. The decision trees are randomly created from subsets of the data, and then vote on particular features. The random forest classifier then aggregates the votes from individual, unique decision trees to determine the final classification of the input data object. My Random Forest Classifier obtained accuracies of 0.886 and 0.879, respectively, on the train and test sets.

4.3.6 Binomial Logistic Regression

The last individual model I used was logistic regression. Since the output of my models must have two levels, 0 and 1, my logistic regression model is better described as a binomial logistic regression model. Interestingly, my binomial logistic regression model obtained an accuracy of 1 (perfect) on the training set, but only 0.836 on the test set.

4.4 Optimizations and Results

4.4.1 Using Kernels to Improve SVM Accuracy

A radial basis function (rbf) kernel was selected for use as opposed to a sigmoidal function kernel after testing differences in accuracies on the test set after training with various kernels on the training set. The SVM model accuracies improved from their original 0.805 and 0.820 on the training set and test set, respectively, to 0.876 and 0.869 (See Figure 4).

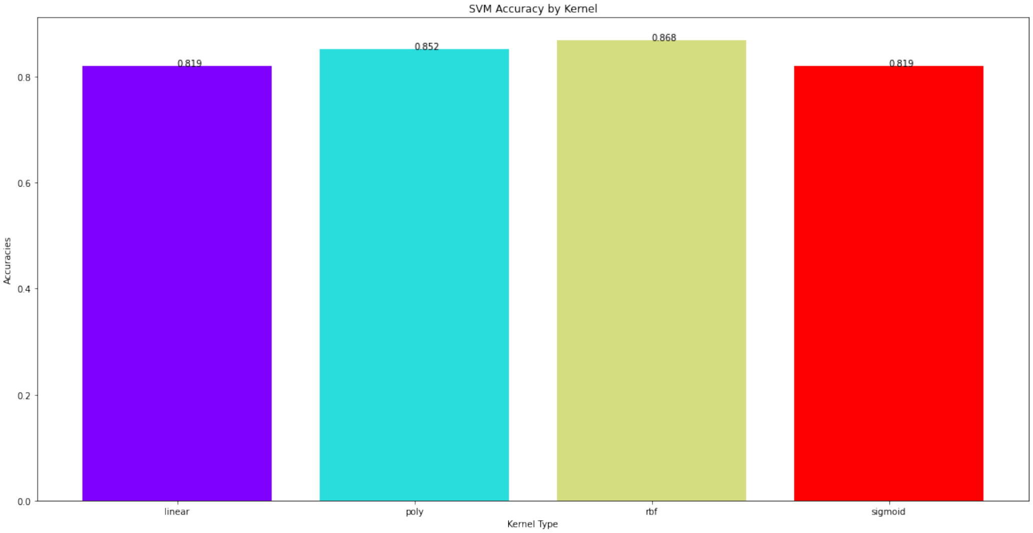


Figure 5: The accuracies achieved on test set using different kernels.

4.4.2 Optimizing Random Forest

Random Forest Classifier can be optimized by changing the number of decision trees to be created in the forest, and the number of same-featured samples needed to split a decision trees into two. Using matplotlib to plot these difference in accuracies based on looping through variations of configurations of these numbers (See Figure x), the new accuracies obtained for Random Forest Classifier were 0.913 and 0.902 on the train and test sets, respectively (See Figure 5).

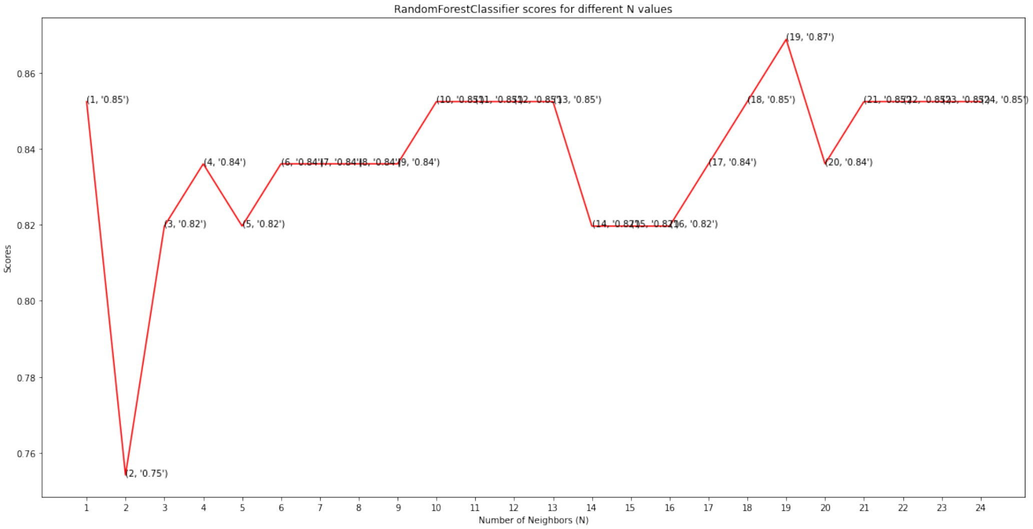


Figure 6: Results of accuracies achieved on test set by varying n=number of estimators.

4.4.3 Voting Classifier

Ensembled models are models that use many smaller, individual models in their calculations. They are essentially several smaller models put together in order to achieve one singular model that delivers superior accuracies to all its component models. I used scikit learn to implement one particular type of ensembled model known as a Voting Classifier. The voting classifier takes into account all individual model outputs as votes (See Figure 6), and classifies the input as a certain output based on whichever vote type is the majority. This is known as ”hard” voting. There is also another variation which provides more weight to certain individual models, known as ”soft” voting. This weightage can be specified. In my case, the model I use implements hard voting. This model obtained the highest accuracy out of all models I made, with an accuracy of 0.934 and 0.920 on the training and test sets, respectively. Since this was by far my superior model, I decided to go ahead and move forward with this. This model was then implemented in my android application.

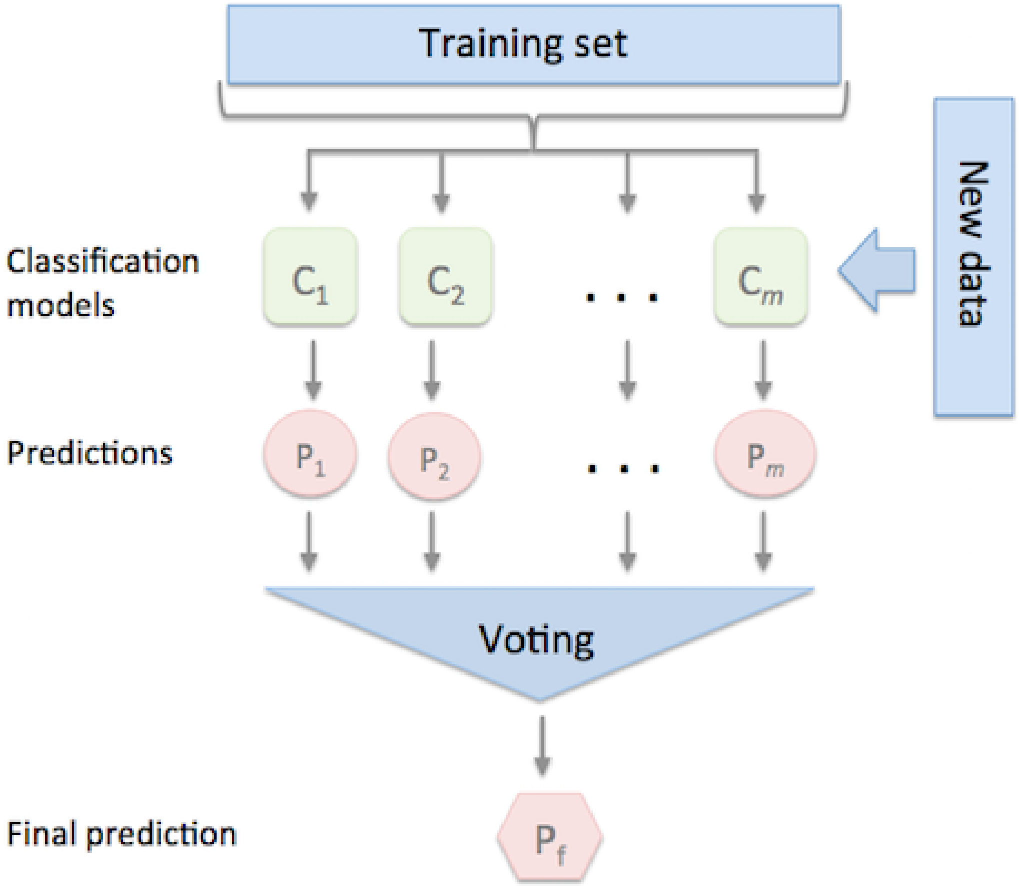


Figure 7: A diagram explaining the structure of voting classifiers. In this case, the C models are all the individual models I created, the Pm are all their predictions, and Pf is the final output determined.

4.5 Android Implementation of the Model

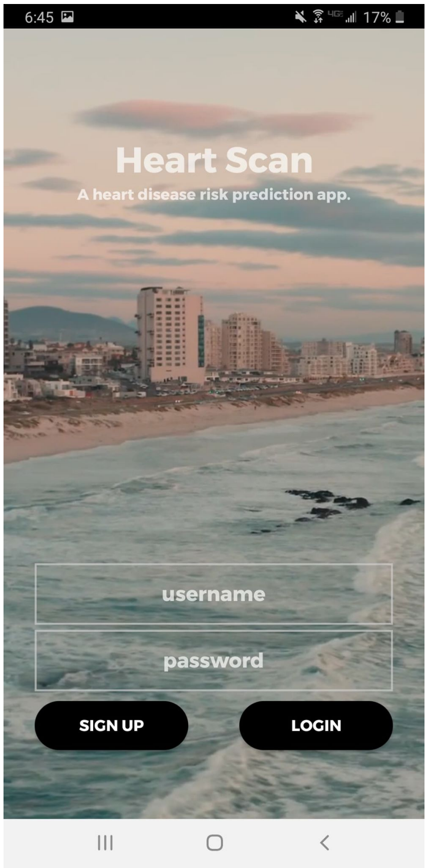
The implementation of an android application of the model was relatively simple. First, the voting classifier model was embedded in a python file. This file was then configured to run in the android app using a package called Chaquopy. Chaquopy allows for python files to be run in android studio.

The app itself was split into five activities: signup, login, Load previous data, Risk Assessment Quiz, Risk Quiz Results. The signup and login activities used VideoViews to create animated backgrounds of nature and cities for aesthetic appeal. Whenever a user signs up, their login info is stored in the cloud using FireBase Cloud Store, a Google data cloud storage service. Upon logging in, users are given the choice to load previous data, or run a new heart disease risk assessment (input their data and have the ML model analyze it). The Risk Assessment Quiz Activity quizzes the user on a series of questions that gathers the data the model requires to run its assessment. Finally, the Risk Quiz Results activity shows the user their risk assessment and allows the user to save their data to their account. See Figure 7 and 8 for pictures of the app. The application has a particular feature called "Autofill". The model asks a lot of questions to users that are not typically known by an individual. Autofill, using the dataset the model was trained on, takes into account a users age and gender and automatically inputs a value similar to what is expected by the model for that user. This allows users who do not know certain vital statistics about themselves to still be able to use the model. The application also builds on the simple binary risk assessment of high and low delivered by the voting classifier. By outputting pre-rounding threshold values, the voting classifier outputs a decimal value that approximates risk from [0,1.0]. This is then scaled to [0, 100] and delivered as the user's risk of developing heart disease in the future.

5 Conclusion and Future Work

The android application housing the risk assessment model is called HeartScan. HeartScan allows any individual to input their health statistics, and using the machine learning techniques described earlier, provides them with an assessment of their risk of developing heart disease in the future. HeartScan helps make the model more accessible to people busy in their daily lives who would not be likely to go to a cardiologist to get diagnosed for any heart conditions.

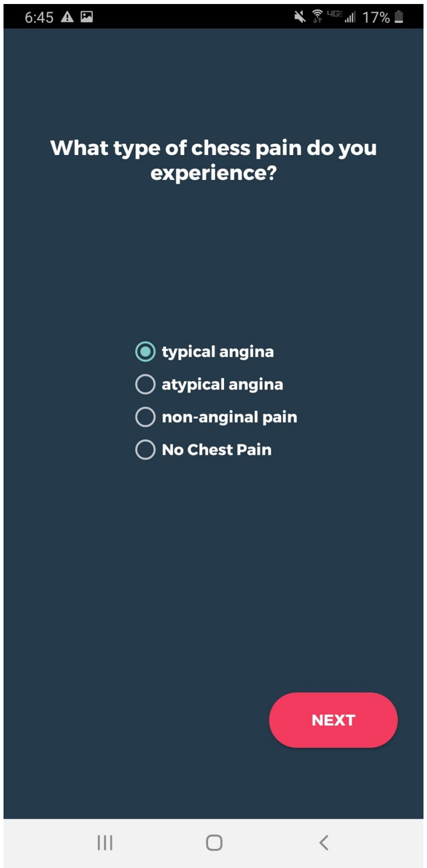
Although HeartScan is complete, there are ways in which a general project of similar function can be improved. For one, the dataset which the model



Heart Scan

A heart disease risk prediction app.

Figure 8: Login Activity



What type of chess pain do you experience?

● typical angina
○ atypical angina
○ non-anginal pain
○ No Chest Pain

Figure 9: Risk Assessment Quiz Activity

trains on has a sample size N=303 only. In a medical context, this is rather small. A larger sample could help to produce a more accurate model. Moreover, a lot of the health questions asked by the model contain information that, even with autofill, could be hard for an individual to find. Implementing a guide or help tool in the app that shows users how they could potentially obtain such information may be helpful in allowing more users to be able to run the risk assessment successfully.

References

[1] Farooq, U., *Tools to Run Python on Android*. Retrieved from https://towardsdatascience.com/tools-to-run-python-on-android-9060663972b4 (2019, March 28).

[2] American Heart Association, *Check. Change. Control. Calculator.*. Addison Wesley, Massachusetts, 2nd Edition, (2016, March 18).

[3] Goff, D. C., Lloyd-Jones, D. M., Bennett, G., Coady, S., D'Agostino, R. B., Gibbons, R., … Wilson, P. W. F., *ACC/AHA Guideline on the Assessment of Cardiovascular Risk.*. doi: 10.1161/01.cir.0000437741.48606.98, (2013).