Personalised Health Monitoring using Predictive Analytics

A Project Report
Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree

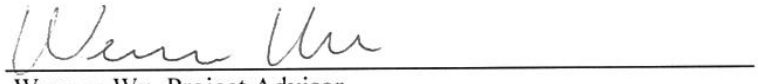
Master of Science in Software Engineering

By

Poojitha Amin, Nikhitha Reddy Anikireddypally, Suraj Ravindra Khurana,
Sneha Vadakkemadathil

December/2018

APPROVED



Wencen Wu, Project Advisor

Dan Harkey, Director, MS Software Engineering

Xiao Su, Department Chair, Computer Engineering

Personalised Health Monitoring using Predictive Analytics

Poojitha Amin[1], Nikhitha R. Anikireddypally[1], Suraj R. Khurana[1], Sneha Vadakkemadathil[1]

***Abstract*—Disease identification and diagnosis of ailments are at the forefront of machine learning research in healthcare. Physical health monitoring using wearable sensors plays a vital role in early detection of abnormalities in the physical health of people. According to a study by Centres for Disease Control and Prevention, about 9% of the population now have diabetes, and one in four are unaware of it. Also, in certain situations, if a person gets excessively fatigued, it may not show any disease immediately, but could be a harbinger of something to come. Today, the increasingly sedentary nature of many forms of recreation time and increasing urbanization results in decreased physical activity and thereby leads to a rise in health problems. A lot of wearables available today can provide important cues to people, however, these devices are not able to perform advanced predictions from the collected data about a disease condition or provide location-centric alerts to people. In this project, we propose real-time analytics approach on sensor data, to monitor the vital signs of a person, like heart rate, sleep patterns etc and notify the user in case of significant changes. The project aims to develop a model to perform predictions based on the captured physiological parameters. The outcome will be a personalized healthcare service which can significantly improve diagnostic accuracy, healthcare quality, and patients' quality of life.**

***Index Terms*— Real-time Health Monitoring, Fitbit API, Resampling, Ensemble Classifier, Heart Disease Prediction**

I. INTRODUCTION

Early detection of diseases and treating them on time can save lives. In general, people tend to ignore their symptoms and underestimate the possibility of having diseases of high mortality rate. Also, the percentage of people who take the annual check-up is relatively less because of the expensive medical services. To overcome such problems, we can use a personalized health monitoring system and make ourselves aware of the possible abnormalities through our vitals. The idea is to build a prediction system which can analyze the vitals of a person, detect any abnormalities, and alert the person about the possibility of developing a disease condition.

Wearable devices provide the first hope of solving these problems because they can be used to capture the vitals for longer periods and outside the laboratory or hospital. We plan to correlate the wearable-sensed health parameters with specific disease patterns to provide insights for an early diagnosis of a disease condition.

The vitals of a person are recorded using wearable sensors and the captured parameters are analysed to make predictions and alert the user. Also, upon availability, the captured data is combined with the previous health history of a person to make personalized predictions. The primary focus of our model is to detect abnormalities with an emphasis on identifying cardiovascular diseases, using time-series patterns with good accuracy. We would use machine learning classifiers to determine the severity of the risk, in terms of high and low. The system helps the user to proactively react to the onset of diseases.

During our research, we found that there are a lot of existing models for disease prediction in healthcare. However, most of them do not make use of data from wearables or activity trackers to perform predictions about a particular disease condition. Our model aims to utilize this data to perform advanced and personalized prognosis of cardiovascular problems. As millions of people use wearables to prevent diseases by maintaining a healthy lifestyle, this health monitoring system will help them to proactively identify and address possible disease conditions. From a user perspective, the process is made really simple. The sensor data is collected from wearable devices and users are notified in case of any abnormalities.

Any kind of abnormalities in a person's vitals can be responsible for the occurrence of heart disease. The paper discusses about different vitals and its correlation with the disease. This kind of analysis is further combined with the person's daily habits like smoking, drinking and exercising to provide a personalized prediction. The machine learning model will be integrated with an application to alert the user in case of an abnormality.

In this paper, we developed a prototype using data mining techniques with the help of historical heart disease dataset. The initial goal of the project was to understand the trends between different attributes and its correlation with the occurrence of heart disease. After this analysis we experimented with different classifier algorithms to predict possibility of heart disease. During our baseline experiments with the Cleveland heart disease dataset, we observed that Multi-layer Perceptron classifier provides the highest accuracy. However, this dataset has only limited number of samples which makes it challenging to build a stable model. Hence, we had to find another dataset with enough samples to build our final model. We used Framingham heart study dataset to build the final machine learning model which is an ensemble of three classifiers.

[1]Computer Engineering, San Jose State University, San Jose, CA

II. RELATED WORKS

Many approaches have been proposed to predict disease conditions and detect health anomalies. A research, intelligent heart disease prediction system using data mining techniques used Decision Trees, Naïve Bayes and Neural Networks for predicting the likelihood of patients getting a heart disease [3]. The study shows that 80% of heart disease patients are males of which 43% are between ages 56 and 63. Neural network model shows that attributes like "Old peak = 3.05 – 3.81", "Serum Cholesterol >= 382.37", "Chest Pain Type = 2", "CA=0" favour the no heart disease predictable state [3]. However, the limitation is that the model only used categorical data. In a study, about supervised sequence learning for cardiovascular prediction [1], a strong relationship has been established between chronic heart diseases and vitals. In this study, they trained and validated a semi-supervised, multi-task LSTM on 57,675 person-weeks of data from off-the-shelf wearable heart rate sensors, showing high accuracy at detecting multiple medical conditions, including diabetes (0.8451), high cholesterol (0.7441), high blood pressure (0.8086), and sleep apnea (0.8298) [1].

Another study evaluates the potential of behavioural risk factors, especially Fitbit-assessed behaviour, to predict readmission for postsurgical cancer inpatients [8]. They analysed the physical activity data from a Fitbit tracker worn by patients during their in-hospital recovery, and built a machine learning model that can identify with an 88.3% accuracy which patients were readmitted to the hospital within 30 days of discharge [8]. Another research paper proposed a machine learning model to predict chronic obstructive pulmonary disease (COPD) using physiological time series patterns [6]. A logistic regression classifier was used to predict the outcome of the binary classification problem. The outcomes are grouped into one of the two classes - "low risk" or "high risk" based on one or more predictor variables. An average accuracy of 79.27% was achieved by the logistic regression model in classifying COPD categories [6].

Additionally, there has been a strong relation established between sleep and health issues. A research conducted at the University of Tokyo used Fitbit HR Charge to collect time series sleep data and proposed a personalized approach for detecting unusual sleep patterns [9]. In a study to predict sleep efficiency from wearable devices [4], a robust automated system was built to analyse sleep quality. The system detects low activity periods and annotates the sleep periods and divides the activity time into segments. Sleep efficiency, which is the ratio of sleep duration minus wake after sleep onset was used as the sleep quality benchmark. Multiple models were used to evaluate the prediction. Adaptive Boosting which gave an F1 score of 68%, Random Forest produced a score of 75% and Support Vector Machines with a score of 70% [4]. It was built with an ability to identify behavioural changes specific to an individual. The paper also discussed deep learning methods to screen potential sleep problems. The deep learning models: Convolutional Neural Network, Simple Recurrent Neural Network, Multi-level Perceptron, each performing with a good F1 score of 80% and above in predicting sleep efficiency [4].

Recent research studies in the field of healthcare analytics have used Fast Fourier Transformation (FFT) to process time-series medical data to produce better accuracy of prediction. Fourier transformation is a robust tool for classification and is faster than the standard Discrete Fourier Transformation [7]. This system uses ensemble frameworks with FFT to outperform basic classifiers.

Another research used heart related data collected from wearable devices to classify and inform whether a person is stressed, depressed or caught by influenza [10]. They used Heart rate variance (HRV) which includes time-domain parameters, frequency-domain parameters and non-linear parameters, to develop a machine learning model for classification [10]. There have been studies conducted on time series clinical data using customized data mining techniques with frequency measurement and feature selection [2]. They used frequency measurement of features and Partial Swarm Optimization to improve accuracy and efficiency of the model. Another paper talks about the considerations and challenges in integrating activity trackers in predictive healthcare analytics [5]. This study emphasis the need to ensure person's' privacy and importance of making sure that electronic systems do not crowd out the decision making by physicians.

III. DATA DESCRIPTION

The dataset used for training the machine learning model is a subset of the Framingham Heart Study dataset. The Framingham Heart Study is a continuing cardiovascular cohort study among a population of residents in the city of Framingham, Massachusetts. The dataset includes laboratory, clinic, and questionnaire data on 4,240 participants. 3658 of the 4240 records are complete samples, and the rest of the records have missing values for certain attributes. The following attributes and risk factors are provided in the dataset:

1. Age – Age in years
2. Male – 0 = Female; 1 = Male
3. Education – Education level of the person
4. CurrentSmoker – 0 = non-smoker; 1 = smoker
5. CigsPerDay - Number of cigarettes smoked per day
6. BPMeds – 0 = Not on Blood Pressure medication; 1 = On Blood Pressure medication
7. PrevalentStroke – Prevalent stroke condition

8. PrevalentHyp – Prevalent high blood pressure
9. Diabetes – 0 = No; 1 = Yes
10. TotChol – Total cholestrol mg/dL
11. SysBP – Systolic blood pressure (mmHg)
12. DiaBP – Diastolic blood pressure (mmHg)
13. BMI – Body Mass Index
14. HeartRate – Heart rate, Beats/Minute
15. Glucose - Blood glucose level (mg/dL)
16. TenYearCHD - Response variable, risk of coronary heart disease (CHD)

IV. TECHNOLOGY DESCRIPTIONS

A. Client Technologies

Client technologies used for this project can be broadly divided into two sections, Web technologies and Fitbit technology, as explained in the following sections.

1) Web Technologies: To enable client interaction for the users, we built a web application using React and Redux frameworks. This web application enables users to register and connect their profile and Fitbit details with the model. The application communicates with the backend servers to store the vital information and then, displays the returned prediction from the model.

- *React*: It is a JavaScript library, which helps in building interactive user interfaces. This framework is extremely useful in building single page applications. Some features like stateful components, one-way data binding, virtual DOM makes the user interface very efficient.

- *Redux:* It is another library of JavaScript for maintaining the application state. It is an open-source library and, is compatible with React Framework. It helps in storing the state of an application at a central location. This enables every react component to have a privilege of accessing the state information. Also, it helps in time-travel debugging, making it easy to test any logic.

B. Fitbit

Fitbit provides an Application Programming Interface (API) for the developers to make use of the Fitbit data. Fitbit API provides information about the vitals like frequency, intensity, pattern and difference in the start and end time of movements to determine the health metrics. We used the Fitbit API to gather the heart rate data. In addition to the heart rate, the following details can also be extracted using the API.

- Calories consumed
- No of Steps taken
- Total Distance Travelled
- Total Floors climbed
- Inactive Minutes
- Light Active time
- Moderate Active Time
- Heavy Activity Time
- Activity Calories
- Total Sleep Time
- Total time Being Awake
- Number of awakenings from Sleep
- Resting period in Minutes

However, current scope of this project only requires heart rate and activity related information.

C. Server Technologies

A backend server enables effective communication between user interface and heart disease prediction model. It helps in storing the vital information into the database. Then, the prediction model does a heart disease check based on the vital information and user information. Server helps in sending back the CHD prediction to the user interface.

- *NodeJS* **:** It is a server-side scripting language. It is based on an event-driven architecture. It performs asynchronous I/O calls, which optimizes the throughput and scalability in web applications. It runs on a single thread by performing asynchronous calls and thereby allows many concurrent connections at no cost.

D. Machine Learning Technologies

Used the following machine learning libraries and packages in the implementation of the machine learning components in the project.

1) Machine Learning Libraries:

- *Scikit-learn*: Machine learning library for classification algorithms, grid search, resampling and classification metrics.
- *Imbalanced-learn:* Package offering a number of re-sampling techniques commonly used in datasets showing strong between-class imbalance.
- *Matplotlib & Seaborn:* Python 2D plotting library for data visualization.

- *SciPy-Stat:* Statistical functions library used to establish the relationship between the attributes and the probability of the heart disease.

2) Machine Learning Algorithms:

- *Random Forest Classifier*: This is an Ensemble method of classifying data, through the construction of multiple decision trees.
- *Gradient Boosting Classifier*: This is an Ensemble of prediction models that is known to be very effective and works by minimizing the loss of the model with the help of gradient descent.
- *Logistic Regression Classifier*: This classification model is used to predict the probability of the classes where the output variable is a binary variable, simply put as either 1 or 0.
- *Voting Classifier*: This is an ensemble method that works by combining the predictions of multiple base estimators using average predicted probabilities. This technique provides an improve model which is highly robust.

E. Data Tier Technologies

The data tier comprises of the data storage and the data access layer.

1)Database Technologies: Database helps in managing user credentials, user profile and user vital information. It is updated with the time-series data, received from the Fitbit. The vital information along with other user details is loaded into the prediction model to identify the possibility of a heart disease

- *MongoDB* - MongoDB is an open source document-oriented database, that uses JSON-like documents. It enables dynamic schemas, and also, allows insertion of data without a defined schema. It also helps in easy replication of database for enabling high availability. It improves read performance by performing integrated caching.

V. PROJECT ARCHITECTURE

The architecture consists of five consecutive phases - data collection, data pre-processing, data transformation, model construction and user notification.

A. Data Collection

The input to the system consists of real-time data collected from Fitbit device as well as lifestyle-related information about the person. Real-time data collected from the Fitbit device form a continuous streaming data. Collecting the lifestyle-related information is a one-time process during the user registration phase. The system will have an interface to fetch Fitbit data upon user authentication and authorization.

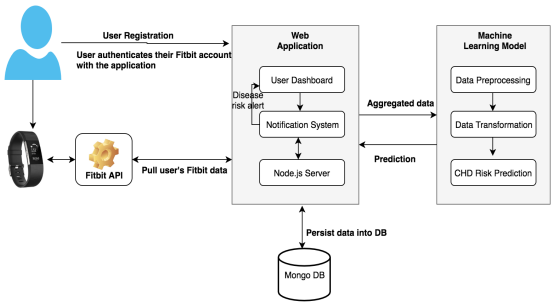


Fig. 1: Architecture Diagram

B. Data Pre-processing

The data pre-processing phase consists of a series of steps, and deals with methods to handle

- streaming time-series data,
- data quality issues,
- noisy data inputs,
- filtering of data records, and
- normalization

Data quality issues can include missing attribute values, duplicate records and inconsistent formats. Duplicate samples are removed and records with missing attributes are either replaced with the mean value or removed from the dataset. Noisy data inputs form the outliers and are handled during the pre-processing phase. As a final step, data records are normalized so that they contribute equally to the analysis.

C. Data Transformation

Data transformation phase starts with data aggregation and generalization. The time-series data collected by the application through Fitbit is aggregated on a daily basis. This phase also involves converting any categorical attributes present in the input into discrete. This phase is crucial as it directly impacts the performance and efficiency of the machine learning model. Feature selection and extraction techniques are also considered during this phase prior to model development.

After the significant data transformation steps, a predictive model is developed. The transformed data is split into two subsets. One of the subsets is used as training data for the model. The training subset is fed into different machine

learning classification algorithms to generate predictions. The accuracy of predictions is evaluated using statistical metrics. The algorithm which gives the best accuracy with better separation of the classes is chosen to build the final machine learning model.

D. Model Evaluation

The model is validated using a validation dataset during the development phase. The validation dataset is a subset of the collected data. It will be replaced with the actual real-time health data of a person when the system is live. The performance of the machine learning model is evaluated during this phase and necessary modifications to the hyper-parameters of the algorithm are made to improve the performance. The model is trained on different folds of data and cross validated to evaluate consistency. The output from this phase is fed back to the data transformation phase to re-train the model. This will help improve the prediction accuracy of the system. The final model evaluation is performed on a test dataset.

E. User notification

This phase deals with sending notifications to the user in case of any abnormalities in his/her vitals. Challenges in implementing this module include filtering out false alarms during the period in which a user is exercising or playing games. This module will be triggered only when the predicted value from the machine learning model indicates a risk of heart disease. In such a scenario, the application will send an alert to the user with an appropriate message. This phase also deals with defining notification types, scheduling the notifications, tracking them for delivery and responding to user-selected actions.

VI. SYSTEM DESIGN

A. Machine Learning Block Diagram

Figure 2 shows the system design for machine learning module. Our ML design includes all the ML phases - acquiring, processing and modelling data, model optimization and evaluation. ML routines include experimenting in a loop and testing. Fine tuning is also performed as part of ML routine. These kind of routines helps in improving the algorithms and thereby, achieve better results. The model is then prepared for deployment and the results are consumed by the users to know about their heart disease possibility.

B. Server Design

Server plays an important role in the communication between client and the prediction model. Figure 3 shows the sequence diagram displaying the flow of requests and responses.

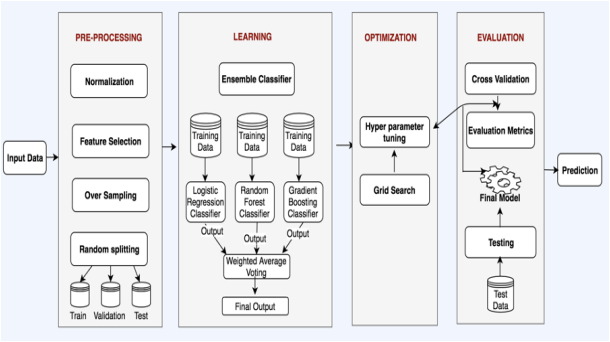


Fig. 2: Block diagram of ML design

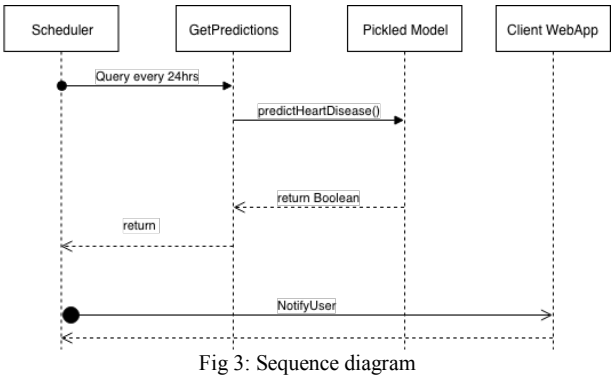


Fig 3: Sequence diagram

VII. IMPLEMENTATION

We have divided our project implementation into three segments namely Client implementation, Server implementation and Machine learning model implementation. The following sections explain each of these segments in detail.

A. Client Implementation

1) UI Implementation: A single page web application is built using React and Redux frameworks. The page is dynamic in nature because of virtual DOM in React. The web application contains 3 important components:

1. Login
2. Register
3. Dashboard

The web application allows the user to login, if he/she has an existing account. The user credentials are taken in a secure way. The authenticity of credentials is checked by using the

user's hash or encrypted details. Once, the user successfully, logs in, he/she can view the main dashboard, which displays the prediction.

If the user is new to the application, he/she needs to fill in some important details to register. The registration portal includes user's personal profile, username, password, lifestyle habits and Fitbit Credentials. These details help in connecting with the Fitbit API to fetch the required vitals. Along with user's vitals, user's lifestyle habits are also taken into consideration.

The main dashboard contains important information about user's condition. It displays an alert based on user's heart disease condition. The result is updated every day in the portal, based on the vital information. When an abnormality or a heart disease is predicted, user is alerted about this. Alert does not confirm the heart disease, it is only a prediction for the user to get it diagnosed and take precautions.

The web application will be hosted on Amazon Web Services, which will enable cloud access to the application.

B. Server Implementation

1) Nodejs Server Implementation: Backend server implementation plays a vital role in the performance of a web application. This application's backend is hosted on a Nodejs server. Server functionalities include forwarding requests from UI to database and also, forwarding the response from Prediction Model to the UI. This server is hosted on Amazon Web Services to enable scalability. All the requests from UI hit the load balancer initially. These requests are distributed to different web servers based on the load on each instance. Then, the server communicates with the database to perform write and read operations.

Once the server writes any new user data to the Database, Prediction Model is triggered. It fetches the user profile from Database, other vitals from Fitbit API and then, runs the model. The response from the model is again sent back using server and the load balancer.

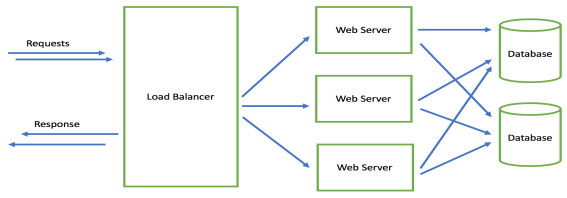


Fig. 4 : Server Implementation

C. Fitbit Integration

We performed some analysis on the existing Crowd-Source Data of Fitbit. Figure 5 shows analysis results for heart rate for one subject. This data analysis is used as a reference to understand the trends. Also, we are fetching the live stats of vitals of each user through Fitbit API. The following steps are used to extract the data using Fitbit API.

- Create and register an application with Fitbit (dev.fitbit.com) to obtain the OAuth 2.0 Client ID and the Client Secret ID.
- Import the fitbit packages and use the Client ID and API obtained from the previous step in generating the access tokens.
  - Step 1 - Import the fitbit library
  - Step 2 - Set the Client ID and Client Secret
  - Step 3 - Call the browser_authorize function to grant the access
  - Step 4 - Derive the access and refresh tokens
- Heart Rate API

  The below API returns the heart rate data and the details level is set to 5 seconds.

  auth2_client.intraday_time_series('activities/heart', date, detail_level='5sec')

- Steps API

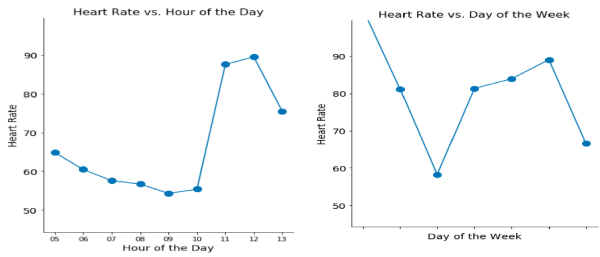


Fig. 5: Heart rate analysis on crowd-sourced data

D. Machine Learning Model Implementation

1) Exploratory Data Analysis: We performed exploratory data analysis to understand the underlying structure of the dataset. We analysed univariate distribution of each attributes in the raw dataset. We also used bivariate visualisation to see the relationship between the response variable (CHD risk) and different predictors. Below histograms and boxplots show bivariate visualisation of categorical and quantitative features respectively.

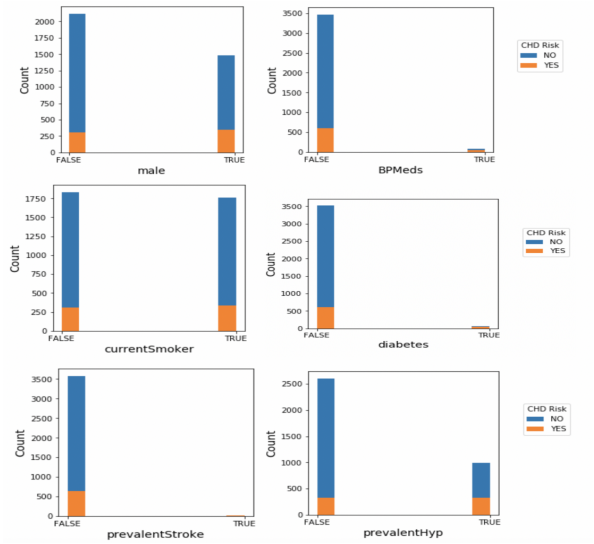


Fig. 6: Bivariate Histograms of Categorical Features

Based on the histograms in figure 6 and boxplots in figure 7, gender, represented by the variable 'male' seems to be a good predictor. Similarly, the participants with a CHD risk seems to have a higher median of age. Attributes like sysBP, currentSmoker and diabetes also shows a slight relation to the response variable.

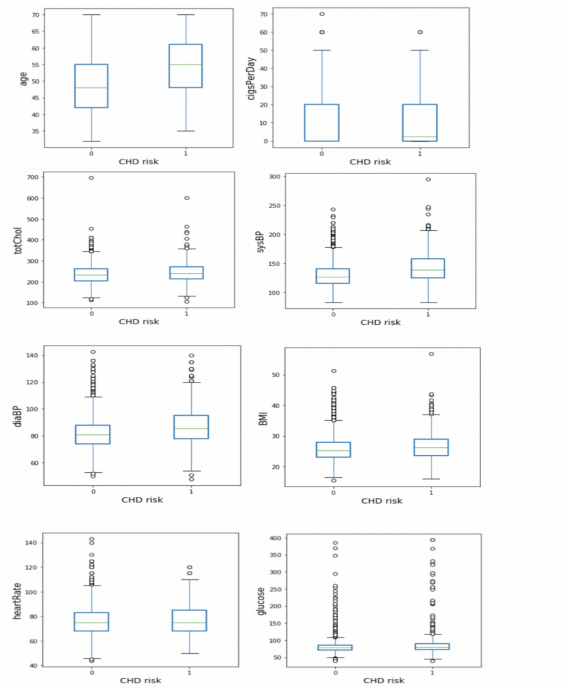


Fig. 8: Bivariate Boxplots of Quantitative Features

Figure 7 shows the variation of heart rate on subjects with and without a cardiovascular disease condition. People with a disease condition appears to have a higher average heart rate.

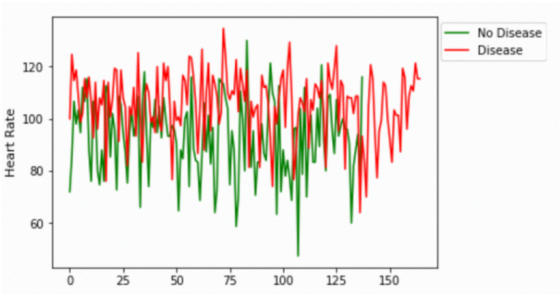


Fig. 9: Variation of heart rate on subjects with and without heart disease

2) Supervised Classification Experiments: The goal of this study is to classify heart disease condition, and in order to achieve this we experimented with several supervised classification approaches such as KNN, MLP classifier, Support Vector Machines, Gradient Boosting, and Logistic Regression. Below are the results from some of the models tried.

Logistic Regression Classifier: Logistic regression is extensively used for building prediction models for a binary outcome such as disease identification. A logistic regression equation is written as,

$$P = e^{\beta_0+\beta_1 X_1} \div (1 + e^{\beta_0+\beta_1 X_1})$$

If we plug in all the selected features, we form the following equation:

$$logit(p) = \beta_0 + \beta_1 * sex + \beta_2 * age + \beta_3 * cigsPerDay$$

$$+\beta_4 * totChol + \beta_5 * BP + \beta_6 * heartRate$$

$$+\beta_7 * BMI + \beta_8 * glucose$$

TABLE I. Threshold vs. Sensitivity and Specificity for Logistic Regression

Threshold	Sensitivity	Specificity
0.1	0.84	0.50
0.2	0.54	0.85
0.3	0.29	0.95
0.4	0.21	0.99

The confusion matrix in Figure 8 shows correct predictions and incorrect ones. We can see that the model is highly specific than sensitive. We also experimented with lowering the thresholds in order to increase the sensitivity as shown in Table I. However, we couldn't strike the right balance between specificity and sensitivity.

	precision	recall	f1-score	support
0	0.86	1.00	0.93	715
1	0.88	0.16	0.27	133
avg / total	0.87	0.86	0.82	848

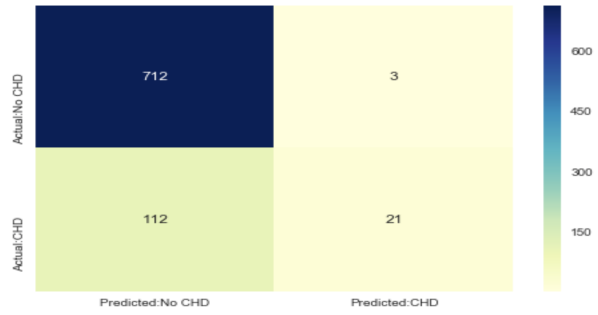


Fig. 9 : Logistic Regression Confusion Matrix

Random Forest Classifier: Random Forest is an ensemble classifier that generates the output by aggregating the results from the base classifiers. We chose to experiment with random forest because of its following features:

1. Weighted random forest is known to balance the error in imbalanced data.
2. It also estimates the feature importance.

We observed some improvement in the recall values, though not satisfactory enough.

	precision	recall	f1-score	support
0	0.90	0.86	0.88	715
1	0.39	0.48	0.43	133
avg / total	0.82	0.80	0.81	848

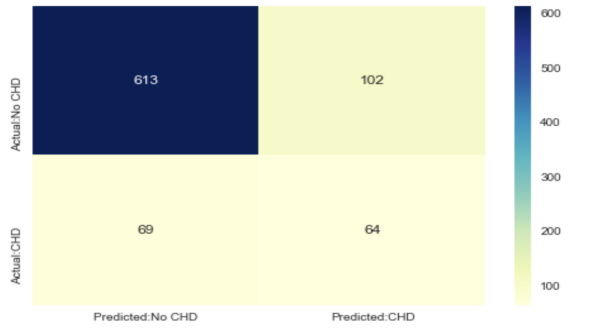


Fig. 10: Random Forest Confusion Matrix

Gradient Boosting Classifier: It works on the principle of ensemble of models. We experimented with Gradient Boosting, because of its following features:

1. All the results predicted right are assigned a lower weight.
2. All the results which are not classified accurate, are given higher weights.
3. It is a principled method designed to handle the class imbalance based on incorrectly classified examples by constructing successive training sets.

	precision	recall	f1-score	support
0	0.87	0.99	0.93	715
1	0.81	0.19	0.30	133
avg / total	0.86	0.87	0.83	848

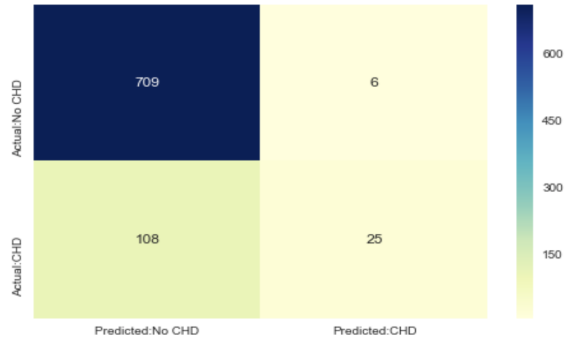


Fig. 11: Gradient Boosting Confusion Matrix

3) Anomaly Detection: We also experimented with other techniques dedicated to imbalanced datasets such as anomaly detection. Anomaly detection is the detection of rare events and in our case, positive disease condition is the rare event. This way the minor class is treated as an outlier, thus separating and classifying the positive samples.

We used the scikit-learn's Isolation Forest algorithm for outlier detection. Random feature selection method is adopted by Isolation Forest Algorithm to detect abnormalities. After feature selection, the maximum and minimum feature values are identified to perform a split. Usually, a dataset contains less disease condition data compared to healthy data. Hence, it is easier to remove the unhealthy data from the healthy records and isolate them. However, another disadvantage is separating healthy data as it requires more conditions. Therefore, it is better to calculate an anomaly score for every small observation and then, we can come to a conclusion based on that score.

	precision	recall	f1-score	support
0	0.88	0.95	0.91	715
1	0.52	0.30	0.38	133
avg / total	0.82	0.85	0.83	848

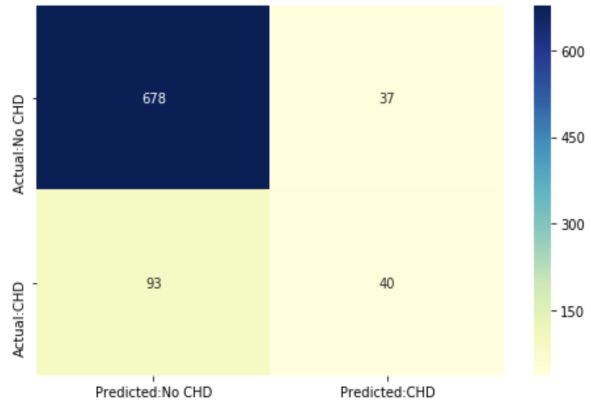


Fig. 12: Isolation Forest Prediction Confusion Matrix

4) Resampling: We have a class-imbalanced dataset as shown in Figure 12. It consists samples mostly of one class, and from our above experiments, it is clear that the classification models on our heart disease dataset performance is going to be minimal. To tackle such a problem, we tried some of the resampling techniques.

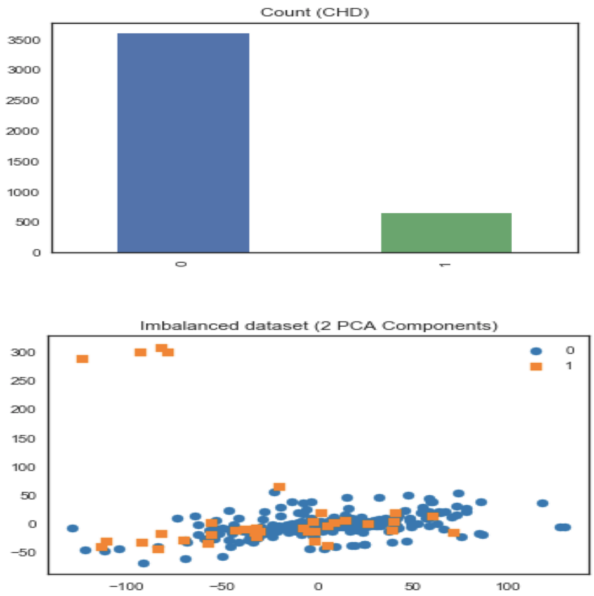


Fig. 13: Imbalanced class count and data distribution

SMOTE: According to N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer [11] , Synthetic Minority Oversampling Technique (SMOTE) is an oversampling method, it takes one additional integer parameter k (number of neighbours). It adds to the minor class new synthesized objects.

Random Oversampling: According to H. He, E.A. Garcia [12], Random oversampling (ROS, also known as bootstrap oversampling) takes no additional input parameters. It adds to the minor class new $(m - 1)|C1(S)|$ objects. Each of them is drawn from uniform distribution on C1 (S).

Random Undersampling: According to H. He, E.A. Garcia [12] (Random Undersampling (RUS) is an undersampling method, it takes no additional parameters. It chooses random subset of C0(S) with $|C0(S)|m-1$ elements and drops it from the dataset. All subsets of C0(S) have equal probabilities to be chosen.

TABLE II : Recall values for different resampling techniques

Resampling Technique	Positive Class Recall
Under Sampling	73%
Over Sampling	56%
SMOTE	61%

5) Model Refinement:

Hyperparameter Tuning using Grid Search: We used Grid searching to obtain optimal parameters for a given model. We methodically built and evaluated our model for each combination of algorithm parameters specified in the grid. The larger the grid, the more execution time, and hence we observed sufficient amount of growth in execution time as well. Some of the hyperparameters values tuned are:

- *Logistic Regression:*
  1. penalty: Specified the norm used in the regularization.
  2. dual: Primal formulation.
  3. max_iter: Number of iterations taken to converge.

  The best results were achieved for the following combination of parameters
  => {'C': 1.0, 'dual': False, 'max_iter': 200}

- *Random Forest:*
  1. n_estimators: The number of trees in the forest.
  2. criterion: The function to measure the quality of a split.
  3. min_samples_split: The minimum number of samples required to split an internal node.
  4. bootstrap: Method for sampling data points.

  The best results were achieved for the following combination of parameters
  => {'bootstrap': True, 'criterion':entropy, 'min_samples_split': 5, 'n_estimators': 180}

- *Gradient Boosting Classifier:*
  1. n_estimators: Number of trees in the model.
  2. loss: The function that will be minimized while finding the best split.
  3. learning_rate: The size of the iterative step.
  4. min_samples_leaf: Method for sampling data points.
  5. max_features: The number of attributes used while deciding the optimum split.

  The best results were achieved for the following combination of parameters
  => {'loss': exponential, 'n_estimators': 1000, 'min_samples_leaf': 2, 'learning_rate': 0.005, max_features: 5}

E. Data Tier Implementation

MongoDB is used as a database for this application. The database is hosted on Amazon Web Services. A MongoDB cluster is created to enable master-slave configuration. This kind of approach helps in having a reliable database access. Even if the master database is corrupted or lost, we always have data in the slave database. Also, MongoDB is eventually consistent by default. This helps in having a secure, available, consistent and scalable database.

We have chosen a document-oriented database to store the user profile. User Profile includes personal details, Fitbit connection details and lifestyle details. User credentials are stored in a secure way in the Database. All the credentials are encrypted and the hash value is stored in the Database. Also, MongoDB creates an object id for every new user. This ID is used as an unique identity for every user profile. Data collected through user's Fitbit account will be aggregated and saved on the database. This data will be used to call the machine learning module to generate predictions.

VIII. TESTING AND VERIFICATION

A. Machine Learning Models Evaluation

Identification of heart disease and no heart disease conditions, is a binary classification problem. One category represents the overwhelming majority of records. This is a classic imbalanced dataset classification problem, where the rate of disease in public is lower compared to the healthy people, i.e. the positive class records is clearly outnumbered by the negative class records. In these cases, accuracy is never a good metric to measure the performance of the model.

1) Confusion Matrix: The performance of a classification model is determined by using a confusion matrix. We trained our model using a training data set consisting of 3392 records and a test data set of 848 records. Our result from the ensemble approach in the confusion matrix form is as below.

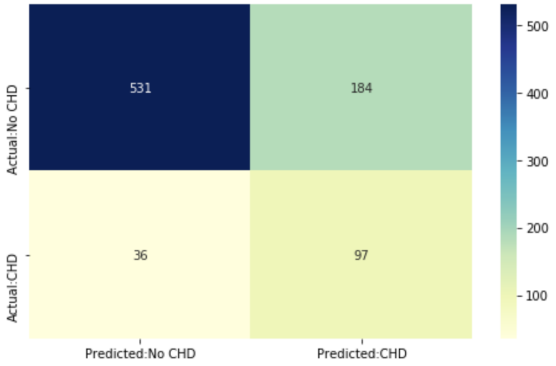


Fig. 14: Confusion matrix of Ensemble model

2) Sensitivity (Recall) and Specificity: We achieved an accuracy of 87% with the Logistic Regression model. However, further analysis of data identifies majority of the data belongs to one class and hence the events of interest have been misclassified.

Intuitively we conclude that, the identification of the no heart disease data points in our problem is not helpful and we should focus on identifying the positive heart disease cases. The best metric to maximize in our scenario is the Recall. Recall or Sensitivity is the ability to identify a heart disease condition in a population of individuals affected with heart disease.

For our data set, we have 848 records, of which 133 (16%) actually are likely to get heart diseases in the future.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\Rightarrow \frac{\textit{Heart disease correctly identified}}{\textit{Heart disease correctly identified} * \textit{Heart disease incorrectly labeled as not heart disease}}$$

$$\Rightarrow \frac{97}{97+36} = 73\%$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

$$\Rightarrow \frac{\textit{No heart disease correctly identified}}{\textit{No heart disease correctly identified} * \textit{No heart disease incorrectly labeled as heart disease}}$$

$$\Rightarrow \frac{531}{531+184} = 74\%$$

In other words, our model will identify 73% of people with the heart disease, but 26% of healthy people will incorrectly be assumed as a heart disease condition. Our measure of accuracy will be a combination of good sensitivity and specificity.

3) K-folds Cross Validation: In order to measure the prediction score of the ensemble model, we performed a cross validation test for the detection of disease and no-disease class. We used K-Folds Cross Validation library. Our data was split into 5 different folds or sets. We used k-1 subsets to train our data and used the last fold as our test data. Even in the cross-validation setup, the ensemble model achieved good recall values.

4) Area Under Receiver Operating Characteristic Curve: ROC curve is another way of visualizing the performance of a binary classifier. The area under the resulting ROC curve (AUC) gives a complete picture of the performance of a trained classifier on a given dataset. TP (sensitivity) can then be plotted against FP (1 – specificity) For each threshold the True Positive rate is plotted against the False Positive rate. The output is a Receiver Operating Characteristic curve as given below:

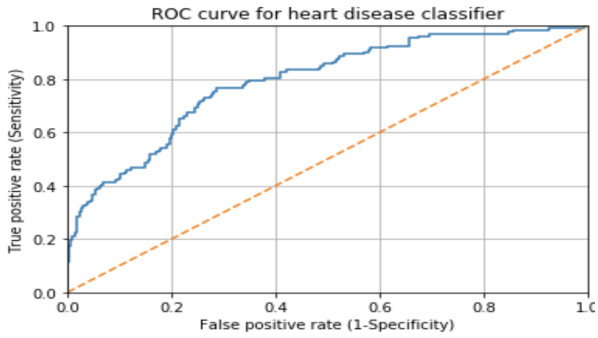


Fig. 15: ROC curve

roc_auc_score from scikit-learn metrics library, computes the area under the ROC curve from prediction scores. Both test data set and the predicted probabilities for the positive class are considered to fetch the true outcomes. These values are loaded into AUC. AUC score ranges between 0.0 and 1.0. In our case, the returned AUC score is ≃80%.

B. Database Testing

1) Schema validation: The forms of the web application and the database scheme are well-validated. An end-to-end testing is performed to ensure the correctness of the data and data fields. There are certain optional fields in the form and hence, the schema of that object changes based on the user entry. The database schema validation is performed for various use cases.

2) Duplicate data checks: Duplicate data checks are performed for username, email id and Fitbit credentials. No two users can have same values for these three fields. Insert and update functionalities are checked to ensure that the database does not allow duplicate entries.

3) Login and User Security: All the user credentials are encrypted and stored in the database. The database is tested to ensure correctness of the login attempts by user. An end-to-end testing from application to database is performed for a secure way of logging in to the application.

IX. CLIENT APPLICATION

The end users are provided with a flexible mobile and web application called Stay-Fit. This application allows the user to register and create a user profile.

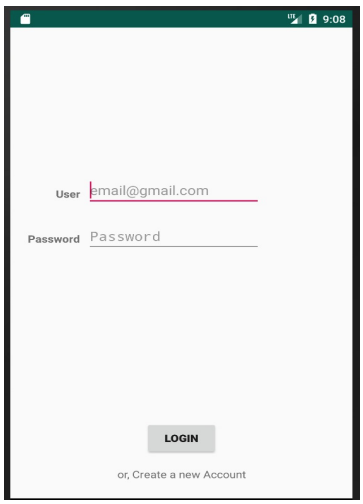


Fig.16: User Login

The user profile screen collects personal and lifestyle information, along with some previous medical history. Personal information includes Gender and Age. Lifestyle includes smoking or drinking habits. Previous medical history includes some important vitals which are not available through wearables. The future scope of the project is to collect more vitals from daily wearables to make it more user friendly.

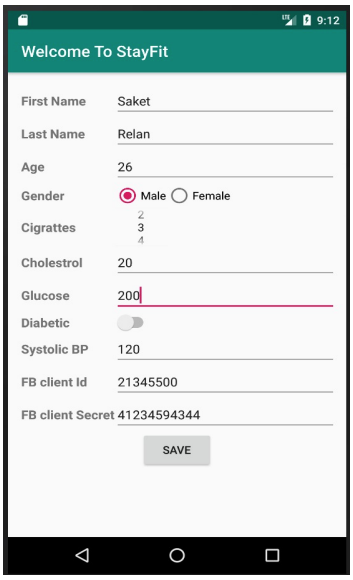


Fig. 17: User Profile

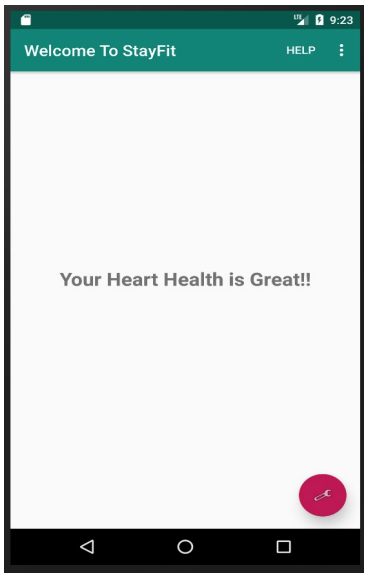


Fig. 18: Prediction Outcome

X. CONCLUSION

We obtained an accuracy of 87% with the Logistic Regression model. However, relying on accuracy alone in an imbalanced class scenario can be misleading. Hence, we use the resampling technique to overcome the class imbalance issue and built an ensemble model to improve the recall values. Our ensemble model using the combination of logistic regression, gradient boosting, and random forest provides an improvement of 40%-70% improvement in the recall values. We get an AUC score of 80%. The model evaluation results of our ensemble classifier show that our approach performs better than many other existing techniques.

REFERENCES

[1] B. Ballinger, J. Hsieh, A. Singh, N. Sohoni, J. Wang, G. Tison, G. Marcus, J. Sanchez, C. Maguire, J. Olgin and M. Pletcher, "DeepHeart: Semi-Supervised Sequence Learning for Cardiovascular Risk Prediction", Arxiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1802.02511.

[2] P. Radha and R. Divya, "Multiple time series clinical data with frequency measurement and feature selection," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), 2016.

[3] S. Palaniappan and R. Awang, "Intelligent heart disease prediction system using data mining techniques," 2008 IEEE/ACS International Conference on Computer Systems and Applications, 2008.

[4] A. Sathyanarayana, J. Srivastava, and L. Fernandez-Luque, "The Science of Sweet Dreams: Predicting Sleep Efficiency from Wearable Device Data," Computer, vol. 50, no. 3, pp. 30–38, 2017.

[5] R. Amarasingham, R. E. Patzer, M. Huesch, N. Q. Nguyen, and B. Xie, "Implementing Electronic Health Care Predictive Analytics: Considerations And Challenges," Health Affairs, vol. 33, no. 7, pp. 1148–1154, Jan. 2014.

[6] Y. Xie, S. J. Redmond, M. S. Mohktar, T. Shany, J. Basilakis, M. Hession, and N. H. Lovell, "Prediction of chronic obstructive pulmonary disease exacerbation using physiological time series patterns," 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013

[7] J. Zhang, R. L. Lafta, X. Tao, Y. Li, F. Chen, Y. Luo, and X. Zhu, "Coupling a Fast Fourier Transformation With a Machine Learning Ensemble Model to Support Recommendations for Heart Disease Patients in a Telehealth Environment," IEEE Access, vol. 5, pp. 10674–10685, 2017.

[8] S. Bae, A. K. Dey, and C. A. Low, "Using passively collected sedentary behavior to predict hospital readmission," Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp 16, 2016.

[9] Z. Liang, M. A. C. Martell, and T. Nishimura, “A Personalized Approach for Detecting Unusual Sleep from Time Series Sleep-Tracking Data,” 2016 IEEE International Conference on Healthcare Informatics (ICHI), 2016.

[10] Wijaya, A., Prihatmanto, A.,& Wijaya, R., (2016). Shesop Healthcare: Stress and influenza classification using support vector machine kernel DOI: 10.13140/RG.2.1.2449.0486

[11] N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

[12] Garcia. E.A and He. H, "Learning from imbalanced data", in IEEE Transactions on Knowledge and Data Engineering, 2009, pp. 1263-1284.