

African Leadership University

COMPUTER SCIENCE CAPSTONE

Detection System for Non-Communicable Diseases using Machine Learning

Authors' full names: Samiratu Ntohsi

Supervisor's names: Dr. Tatenda Duncan Kavu

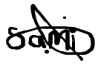
Year of Graduation: July 2021

Declaration

I certify that the work presented in this capstone is my own and that any work that has been performed by others is appropriately cited.

Date: 30 January 2021

Name: Samiratu Ntohsi

Sign: 

Acknowledgment

I would like to express my sincere gratitude to **God Almighty** for this great opportunity and the following people for their support and encouragement during working on this project.

- ★ My supervisor Dr. Tatenda Duncan Kavvu, for his guidance and support to me throughout this project.
- ★ The computer science department at ALU-Rwanda for their help and support.
- ★ My Mentor, Madelle Kangha, and Jumpstart Academy Africa make it possible for me to come to ALU.
- ★ My family for their prayers and encouragement.
- ★ My friends and classmates for their support throughout the program.

Abstract

Noncommunicable diseases, otherwise known as chronic diseases, remain a significant cause of deaths globally, responsible for about 70% of deaths. The burden keeps increasing significantly in low-income countries, most of which are African countries. One of the main challenges with NCD cases is the failure to detect them early. Delay in the detection of Noncommunicable diseases results in complicated cases that are expensive to treat. Furthermore, once an NCD reaches the detailed stage, prevention becomes difficult, and it is a challenge to slow the disease's development. Hence, the World Health Organization advises early detection to maximize the impact of non-communicable disease intervention strategies. This project aims to develop an automated system that uses machine learning to detect individuals at risk of Noncommunicable diseases. The focus for this project will be on using deep learning supervised machine learning techniques to build a web-based system for early detection of diabetes and heart disease.

The project will help improve early detection through access to screening NCDs services of the same quality as that which medical professionals will provide to the community.

Contents

Chapter 1: Project Proposal.....	7
1.1 Introduction and Motivation.....	7
1.2. Problem statement.....	7
1.3. Hypothesis.....	7
1.4. Project's Main Objectives.....	8
1.5. Specific Objectives.....	8
1.6. Project Scope.....	8
1.7 Technical Requirements.....	8
1.8 Project Timeline	8
Chapter 2: Literature Review.....	9
2.1 Introduction to Non-Communicable Diseases (NCDs).....	9
2.2. Applications of Machine Learning for NCDs Detection.....	9
2.3. Related work.....	10
2.4. Analysis of existing Diabetes and Heart Disease Detection Systems.....	10
2.5. Will Machine Learning improve the detection of NCDs?.....	11
2.6. Significance of Study.....	11
2.7 Conclusion	12
Chapter 3: Requirement Gathering, Analysis. & Methodology.....	13
3.1 Summary of Software Requirement Specification.....	13
3.2 Methodology.....	14
3.3 Functional Requirements.....	15
3.4 Non-functional Requirements.....	15
3.5 System Usability & Performance.....	15
3.6 Standard development tools used.....	15
3.7 Use case diagram.....	16
Chapter 4: System Design.....	17
4.1 Introduction.....	17
4.2 Data Exploration.....	17
4.3 Feature Engineering.....	18
4.4 Data Splitting.....	20
4.5 Model Development & Optimization.....	21
4.6 Model Fitting.....	22
4.7 System Architecture.....	23
4.8 User process flow.....	24
Chapter 5: Implementation & Testing.....	25
5.1 Introduction.....	25

5.2 Model Testing.....	25
5.3 Model performance analysis.....	27
5.4 Model Accuracy.....	28
5.5 Sample input and results	31
5.5 User Interface of the NCDs Detection System.....	32
Chapter 6: Conclusion & Recommendations.....	33
6.1 Conclusion	33
6.2 Recommendations.....	33
7.0 Bibliography.....	34

List of figures

Figure 1: Sample Structure of the Model.....	14
Figure 2: Use case diagram for the NCD detection web application.....	16
Figure 3: Kaggle heart disease data.....	17
Figure 4: UCI diabetes data.....	17
Figure 5: Extracting relevant features that contribute to diabetes.....	18
Figure 6: Extracting relevant features that contribute to heart disease.....	19
Figure 7: Data used for training model for diabetes detection.....	19
Figure 8: Data used for training model for heart disease detection.....	20
Figure 9: Data Splitting.....	20
Figure 10: Model for diabetes detection.....	21
Figure 11: Model for heart disease detection.....	22
Figure 12: Physical architecture of the NCDs detection system.....	23
Figure 13: User process flow.....	24
Figure 14: Accuracy, Precision, F1_score, Recall, & AUC of the models.....	26
Figure 15: Accuracy and area under the curve of diabetes and heart disease models..	27
Figure 16: Sample confusion matrix.....	28
Figure 17: Confusion matrix for diabetes detection mode'	29
Figure 18: Confusion matrix for heart disease detection Model.....	30
Figure 19: User interface for the NCDs Detection System.....	32

List of tables

Table 1: Parameters used during model fitting.....	22
Table 2: Performances of the models' Accuracy, Precision, F1_score, Recall, AUC metrics..	26
Table 3: Performance-based on accuracy and area under the curve of the models.....	29
Table 4: Summary of diabetes detection model confusion matrix.....	30
Table 5 Summary of heart disease detection model confusion matrix.....	30
Table 6: Sample inputs and results for diabetes detection model.....	31
Table 6: Sample inputs and outcomes for heart disease detection model	31

Chapter 1: Project

1.1 Introduction

Noncommunicable diseases are also known as chronic diseases. Individuals that have them cannot transmit it. They include Heart diseases, cancers, diabetes, and chronic respiratory diseases. They result from unhealthy habits like harmful use of alcohol, tobacco, unhealthy diets, and physical inactivity[1]. At the global level, NCDs are collectively responsible for 70% of all deaths. Low-and-middle-income have access to 3% of health workers and 1% of the world's financial resources[2]. Just as alarming is that the World Health Organization reveals that by 2030 the burden of disease for chronic conditions like cancer, cardiovascular and respiratory disease, and diabetes requiring specialist care will far surpass that of infectious diseases. According to the World Health Organization, the Covid-19 pandemic has increased NCDs' burden in most countries due to staff and funds' reallocation. Also, patients suffering from NCDs are more susceptible to contracting the virus[3]. Delay in the detection results in a more complicated NCD case, which is more expensive to treat. Hence, leaving us with the challenge of achieving high-impact NCDs intervention strategies through early detection given the limited resources. Machine learning models are widely used in the healthcare sector to predict and detect many diseases, including the detection of NCDs. Similar research by Hu M et al. uses machine learning to improve NCDs' intervention strategies in Bangladesh[4]. A. Dinh. et al. and S. Pasha, and P. Kadu et al. worked on similar projects for predicting NCDs. The results of these systems based on accuracy were good enough[5][6][7]. The methods proposed by these authors used traditional machine learning algorithms like Random forest, KNN, and Decision Trees. Therefore, for this project, the focus will be on deep learning algorithms. In the end, we will be comparing which machine learning technique performs better between deep learning and traditional machine learning approach.

1.2 Problem Statement

The failure to detect NCDs at an early stage is a challenge. The World health organization says the delay in the detection leads to further development of the disease. Diagnosis of NCDs late results in a complicated case that is expensive to treat. Also, low-and-middle-income countries face the challenge of limited access to health staff and qualified health personnel. For example, the doctor to patient ratio in Rwanda is 1 per 10,000, but WHO recommends 2.5 per 10,000 people[1]. Therefore, there is a need for an automated system that would help doctors carry out more diagnoses at the early stage.

1.3 Hypothesis

The use of machine learning techniques for detecting noncommunicable diseases will improve the early detection of NCDs.

1.4 Project's main objective

This project's main objective is to develop a web-based system that uses data and supervised machine learning techniques to automate the early detection of non-communicable diseases early to ensure a prompt treatment to prevent or slow down the development of the disease.

1.5 Specific Objectives

- Collect relevant data, prepare data, and use it for the training of traditional machine learning and deep learning models
- Achieve a disease detection accuracy of at least 95%
- Deploy the trained models to a web application that is user-friendly to facilitate user interactions.

1.6 Project Scope

- The project will focus on the detection of heart disease and diabetes.
- The system will be web-based only.
- Data will come from the UCI machine learning repository.
- The system will use the Keras model.

1.7 Technical requirements

Libraries for data preparation and training of the models are pandas, Numpy, Scikit-learn, Tensorflow. In addition, building and deployment of the website will require Streamlit Library. This open-source Python library makes it easy to create and share beautiful, custom web apps for machine learning and data science.

1.8 Project Timeline

Chapter 2: Literature review

2.1 Introduction to Non-Communicable Diseases (NCDs)

In this 21st century that we are in, NCDs are the primary cause of mortality. The WHO statistics show that they are collectively responsible for 70% of global deaths[1]. Some NCDs statistics from WHO worth knowing, which I have highlighted in the coming sentences that 15 million deaths from NCDs are premature between ages 30 - 60. Over 80% of these premature deaths happen in low-and-middle-income countries. Heart diseases/cardiovascular diseases contribute most to the NCDs death rates, the subsequent being cancers, followed by respiratory infections, and finally diabetes[2]. NCDs evolve from poor habits like physical inactivity, tobacco use, harmful alcohol use, and unhealthy diets[2]. NCDs are slow in progression; hence, prevention from further development is possible if detection happens early[8]. Several methods can assist in detecting NCDs, but we will be looking at how the new technology, machine learning, can serve the same purpose more efficiently.

2.1 Applications of Machine Learning for NCDs Detection.

IBM defines Artificial intelligence as a technology that gives computers the ability to mimic the learning, problem-solving, decision making, and perceptions of humans[9]. Machine learning is a subfield of Artificial Intelligence (AI) and, at the moment, is the primary form of AI. According to ScienceDirect, Machine Learning (ML) is a field of study that uses computer algorithms to convert empirical data into applicable models[10]. In the healthcare sector, AI uses automated diagnosis processes and automated treatment of patients with special needs. The adoption of AI in the healthcare sector has so many promising outcomes, like giving healthcare specialists time to focus on tasks that cannot be automated, improving the quality of services, and reducing costs[8]. A subset of machine learning called predictive modeling uses data and statistics to predict that model's machine learning outcome. Several researchers have carried out studies on using a predictive model to diagnose or predict NCDs, also known as Chronic Diseases (CD). According to G. Battineni et al., the adoption of predictive models by scientists for either diagnosis, forecasting, or prediction of NCDs has recently been an increase[8]. This section will review the literature on similar systems that exist for detecting NCDs or other diseases. For each scenario, close attention will be on the problem it addresses, the method used, and the results/ impact of the system. The literature review aims to ensure that the building's approach is not doing the same thing as the existing ones. Also, during the review of existing literature on similar systems, we will check if there are any gaps to cater to them in this system that we are building.

2.3 Related work

The inability to detect NCDs early enough is a challenge. H. Polat et al. confirms this in their study by saying that NCDs develop very slowly, making early detection and effective treatment the best approach for reducing death rates due to NCDs. Many researchers have carried out studies on the use of ML for forecasting or identifying individuals at risk of suffering from NCDs in the future. A. Dinh. et al., in their study titled 'A data-driven approach to predicting diabetes and cardiovascular diseases using machine learning, they present a system for detecting at-risk patients using survey data and laboratory results[5]. Another study by S. Pasha on diabetes and heart disease prediction using machine learning algorithms presents a system that improves predicting these diseases[6]. Another similar research is by P. Kadu and A. Buchade, which is about an approach that will inform the user about the probability that a disease will happen in the future. With this system, they hope to remove the burden of visiting the hospital regularly for checkups[7]. A more detailed analysis of similar methods is in the paragraphs below, intending to identify gaps.

2.4 Analysis of existing Diabetes and Heart Disease Detection Systems

Each of the similar systems reviewed is addressing a problem. P. Kadu and A. Buchade have built a system that individuals can use at their homes' comfort to predict NCDs' risk of NCDs in the future. Their main focus was to remove the burden of regularly visiting the hospital for health check-up routines[7]. To achieve their purpose, they used variables such as step count, hours of sleep, and weight obtained without going to the laboratory. Their approach uses clustering, an unsupervised machine learning technique. M. Hu et al. in Bangladesh proposed a system that helps to identify subjects at risk of NCDs in the future. They were addressing the fact that most individuals went undetected during NCD screening intervention strategies in Bangladesh. They used data from some of the intervention program surveys, used the Gradient Boosting algorithm, and predicted 98% of at-risk patients[4]. A. Dinh. et al. as well proposed a data-driven approach for identifying individuals at risk of NCDs in the future. Another purpose of their study was to identify critical variables that contribute more to the disease. Machine learning models used include Logistic Regression, Support Vector Machine, Random forest, and Gradient Boosting. The model performances were reasonably good, and based on the data, critical contributors to diabetes were waist size, age, self-reported weight, leg length, and sodium intake. The key contributors to cardiovascular disease were age, systolic blood pressure, self-reported weight, chest pain occurrence, and diastolic blood pressure[5]

2.5 Will Machine Learning improve the detection of NCDs?

The hypothesis for this research is that using machine learning techniques will improve the early detection of NCDs. G. Battineni et al. performed an analysis of 453 articles on applying machine learning predictive models for NCDs' diagnosis. Their study confirms that the adoption of machine learning in forecasting chronic diseases has been on the rise recently[8]. They also realized that popular ML techniques are; Support Vector Machines, Neural networks, and ensemble classification. A similar review by Y, Khan et al. on 35 articles on machine learning techniques used to detect/predict heart diseases reveals some research gaps in their various approaches. I will highlight two. The two gaps are; in the future, systems are not available for people who can use the application via the internet, and the second gap is that the number of attributes is way too many in one of the systems, which may result in higher computation time[11]. Finally, S. Weng et al. in 2017 presented a study on whether ML can improve NCDs risk prediction. The research focuses on heart disease (Cardiovascular diseases). This study confirms that machine learning will significantly improve detecting patients at risk of cardiovascular diseases. These individuals will then benefit from preventive treatments[12].

2.6 Significance of Study

Because similar systems exist, it is necessary to make it clear why this project is essential. The methods often used are traditional Machine learning techniques such as Support Vector Machines, Logistic Regression, and ensemble techniques such as Random Forest and gradient boosting from the reviewed literature. Therefore, we will focus on deep learning techniques- Keras models to be precise. Another feature we will add to the system that we did not find in these systems is giving users recommendations on preventive measures or treatments based on their results. Unlike these systems that exist, this project's resulting system will be hosted and accessed, and tested by users to the best of our knowledge. We hope that hosting it online will also inspire other young learners/students who are curious about the application of Machine learning in the medical field.

2.7 Conclusion

Gaps & technologies we will use in developing the NCD detection system

After reviewing the existing literature, we will consider the following relevant findings in our NCD detection system development.

- The variables said to be critical contributors to heart diseases, and diabetes will be used in our system. The key contributors to diabetes are waist size, age, self-reported weight, leg length, and sodium intake. Simultaneously, cardiovascular disease is age, systolic blood pressure, self-reported weight, chest pain occurrence, and diastolic blood pressure.

- Another gap we found was that the computation time would be high when there are too many variables. Therefore we will be implementing feature selection and will work with a maximum of 5 variables.
- The website will be hosted on Heroku, and users can access it over the internet. During the literature review, most of these systems are not available online, but it used just in the scenario for which it was designed.
- We will focus more on the application part and the user interaction on the web-based system.

Chapter 3: Requirement Gathering, Analysis. & Methodology

3.1 Summary of Software Requirement Specification

In the previous sections, we identified the inability to detect NCDs at an early stage. Two models will be developed to solve the problem, one for the prediction of diabetes and the other for the prediction of cardiovascular disease, and deploy them on a web application. It's going to be a user-centered system, and the user could be a doctor who wants to use it for their patients and individuals who want to know their diabetes or cardiovascular disease status. The user will provide their variables (diabetes: age, sex, weakness, delayed healing, obesity, sudden weight loss, Cardiovascular Disease: exercise, chest pain type, the stress of heart during exercise, maximum heart rate achieved, slope peak during training), and the system will detect whether they are at risk of the NCD in the future and then recommend precautions against the disease. It will be beneficial as often it could be challenging to detect NCDs at an early stage, but with machine learning, this is possible.

3.2 Methodology

Stage one: The first step will be data collection, and in this case, data will come from the UCI Machine Learning Repository. The early-stage [diabetes risk dataset](#) and the [heart disease dataset](#).

Stage two: Data preprocessing will involve data cleaning, missing values, normalization, and relevant feature selection. Below is an image showing the structure of the model for the detection of diabetes and cardiovascular disease.

Stage three: We will build our Keras model trained with the preprocessed data from stage two. The diagram below shows us the structure of the model.

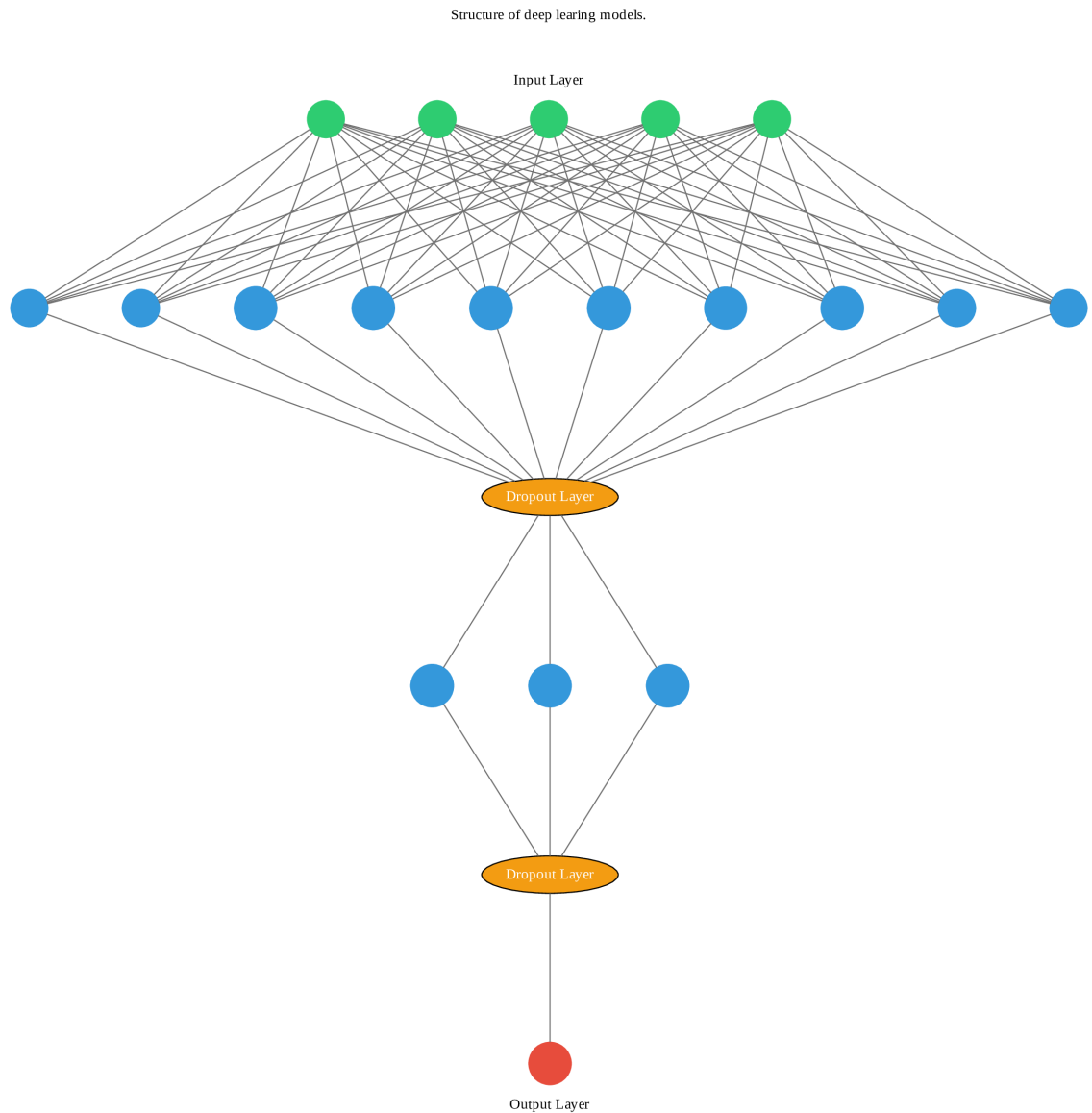


Figure 1: Sample Structure of the Model

In summary, the eight most relevant features that contribute more to the disease-in-question will be selected from the data and passed as input to the neurons in the model. Then there is a dropout technique in our layer for regularizing our deep neural network to reduce overfitting and minimize generalization error in our deep neural network. Then to a hidden layer of 3 neurons and another Dropout layer for regularization purposes. Finally, we pass it to the output layer, which is a single neuron.

Stage four: Model evaluation. After training and testing, the model will be evaluated using the accuracy score, Precision, Recall, F1_score, the area under the curve (AUC), and confusion

matrix. The confusion matrix will compare True Positives, True Negatives, False Positives, and False Negatives.

Stage five: Here, our models will be deployed to a web application to provide a user interface for user interaction.

3.3 Functional Requirements

About Page: The homepage will give a piece of brief information on what the web app does and provide two links: the diabetes detection page and the other for the Cardiovascular detection page.

Diabetes detection page: Users will be presented with a page where they will provide relevant variables for diabetes detection and get notified whether they are at risk or not.

Cardiovascular disease detection page Users will be presented with a page where they will provide relevant cardiovascular disease detection variables and get notified whether they are at risk or not.

3.4 Non-Functional Requirements

- The system will be accessible by anyone anywhere on desktop and mobile devices. Hence, the website will be very responsive on all screen sizes.
- Users should verify email within 5 mins.
- Users will be able to change their password and reset their password if they have forgotten it.
- Clear instructions and hints on the home page ensure that users get a seamless experience on the system.

3.5 System Usability & Performance

- The system should be easy to understand and use by users that access it.
- Once a user-provided their variable, they should get a response almost instantly.

3.6 Standard development tools used.

Hardware

- Operating system: Ubuntu 18.04, intel core i5
- RAM 8GB

Software tools

- Tensorflow: to provide us with the deep learning algorithm in this case Keras model is used.
- Google Colaboratory Notebooks: Environment for training our model.
- Scikit-learn: Provide us with the libraries needed for data preprocessing, model training, and evaluation.
- Visual studio code is the development environment for developing the website for the detection of NCDs.
- Chrome/Firefox: needed during development and debugging website
- Heroku: host our website as well as our database.

3.7 Programming Languages & Frameworks

- Python: For model development, training, and testing.
- Streamlit: Framework for building and deploying the model to a web application.

3.8 Use Case Diagram

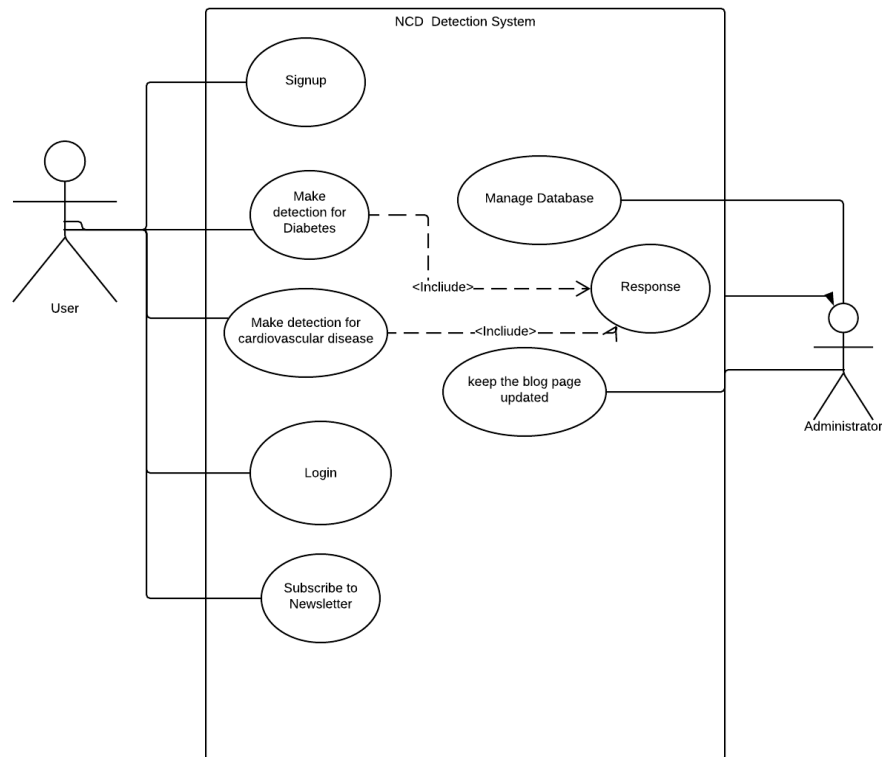


Figure 2: Use case diagram for the NCD detection web application.

Chapter 4: System Design

4.1 Introduction

This chapter explains how all the elements of the system for predicting NCDs have been implemented. The readers/experts will understand how the system's various modules were developed and integrated by reading this chapter. Also included in this chapter are process flow diagrams to help readers understand how the system will function.

4.2 Data Exploration

The datasets for heart disease and diabetes used for this project are available online in the UC Irvine machine learning repository. The diabetes dataset contains 768 data points collected using direct questionnaires from the patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh, and approved by a doctor. Below is a sample of the diabetes data.

The heart disease dataset contains 303 data points. The dataset was created by the Hungarian Institute of Cardiology, Budapest, University Hospital, Zurich, Switzerland, University Hospital, Basel, Switzerland, and V.A Medical Center, Long Beach and Cleveland Clinic Foundation. Below is a screenshot of the heart disease data.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Figure 3: Kaggle heart disease data

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
1	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive

Figure 4: UCI diabetes data

4.3 Feature Engineering

- Check for null values
- Identify data types
- Convert categorical to numeric values. For example, the diabetes dataset mainly had “Yes” and “No” values; they were transformed to “1” and “0,” respectively. In addition, the “Male” and “Female” were converted to “1” and “0” respectively and also “Positive” and “Negative.”
- The xgboost classifier algorithm is used to identify essential features in the dataset, and the first eight essential features are selected using the xgboost classifier for training the model.

Relevant feature Selection

```

from xgboost import XGBClassifier
from xgboost import plot_importance
# fit model to training data
xgb_model = XGBClassifier(random_state = 0 )
xgb_model.fit(Xdiab, Ydiab)
print("Feature Importances : ", xgb_model.feature_importances_)
# plot feature importance
plot_importance(xgb_model)
plt.show()

```

Feature Importances : [0.02468059 0.1104956 0.22259757 0.24462892 0.05272717 0.03362136
0.0257689 0.02738756 0.03067236 0.02826071 0.05165935 0.02664803
0.03390522 0.0117486 0.05558585 0.01961219]

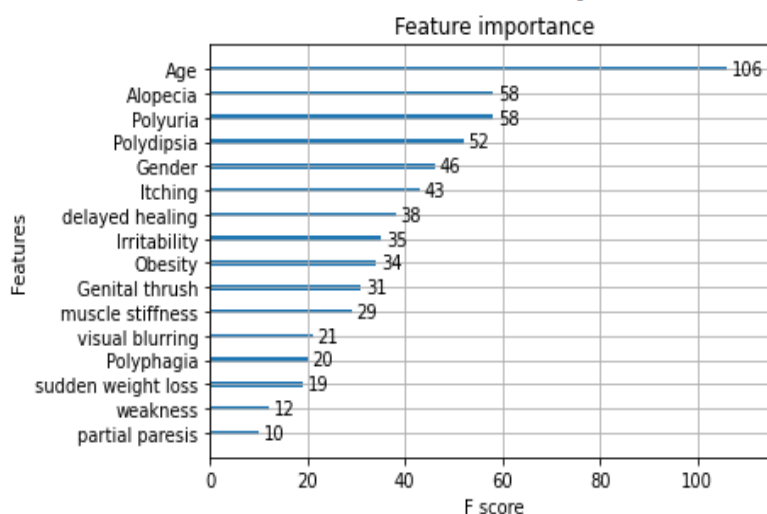


Figure 5: Implementation and Visualization of relevant features that contributes more to our diabetes prediction.

```
# fit model to training data
xgb = XGBClassifier(random_state = 0 )
xgb.fit(Xher, Yher)
print("Feature Importances : ", xgb_model.feature_importances_)
# plot feature importance
plot_importance(xgb)
plt.show()
```

```
Feature Importances : [0.02468059 0.1104956 0.22259757 0.24462892 0.05272717 0.03362136
0.0257689 0.02738756 0.03067236 0.02826071 0.05165935 0.02664803
0.03390522 0.0117486 0.05558585 0.01961219]
```

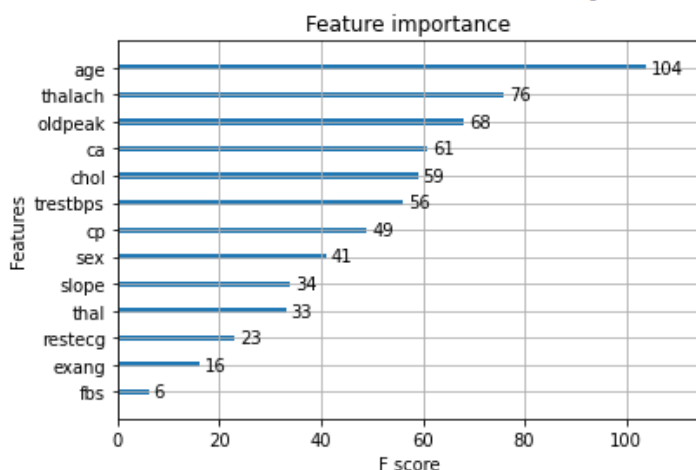


Figure 6: Implementation and Visualization of Extracting relevant features that contributes more to our heart disease prediction.

Processed Data

```
# Creating a new dataframe with the 8 most correlated features
diab_df = diabetes[['Age', 'Alopecia', 'Polyuria', 'Polydipsia', 'Gender', 'Itching', 'delayed healing', 'Irritability', 'class']]
diab_df.head()
```

	Age	Alopecia	Polyuria	Polydipsia	Gender	Itching	delayed healing	Irritability	class
0	0.324324	1	0	1	1	1	1	0	1
1	0.567568	1	0	0	1	0	0	0	1
2	0.337838	1	1	0	1	1	1	0	1
3	0.391892	0	0	0	1	1	1	0	1
4	0.594595	1	1	1	1	1	1	1	1

Figure 7: Data used for training model for diabetes detection

```
[42] heart_df = heart[['age' , 'thalach' , 'oldpeak', 'ca' , 'chol', 'trestbps', 'cp', 'sex', 'target']]
heart_df.head()
```

	age	thalach	oldpeak	ca	chol	trestbps	cp	sex	target
0	63	150	2.3	0	233	145	3	1	1
1	37	187	3.5	0	250	130	2	1	1
2	41	172	1.4	0	204	130	1	0	1
3	56	178	0.8	0	236	120	1	1	1
4	57	163	0.6	0	354	120	0	0	1

Figure 8: Data used for training model for heart disease detection

4.4 Splitting Data

The data is first separated into targets and features. Then, the preprocessed is split into training and test sets. The training set constitutes 80%, and the test set comprises 20% of the data set.

```
# split data into X and Y train
Yd = diab_df[['class']]
Xd = diab_df.drop(['class'],axis=1)
```

```
# Split data into train and test
Xd_train, Xd_test, yd_train, yd_test = train_test_split(Xd, Yd, test_size=0.2, random_state=43)
```

Figure 9: Data splitting

4.5 Model Development and Optimization

At this stage, the extracted features were used to train a deep learning model (Keras model). Two models were developed, one for diabetes prediction and another for heart disease prediction. For model Optimization, the dropout technique for regularizing Neural Network models is used. The dropout technique was proposed by Srivastava et al. in their 2014 [paper](#). The Stochastic Gradient Descent optimization algorithm was used in optimizing the diabetes prediction model. The SGD estimates the error gradient for the model's current state and then updates the model's weights using a backward-propagation of errors algorithm. This process is referred to as backward propagation. Below are screenshots of the models developed for diabetes and heart disease prediction.

```
25] from keras.optimizers import SGD
diab_model = Sequential()
diab_model.add(Dense(units=15, input_dim=8, kernel_initializer='uniform', activation='relu'))
diab_model.add(Dropout(0.3))
diab_model.add(Dense(units=3, input_dim=8, kernel_initializer='uniform', activation='relu'))
diab_model.add(Dropout(0.2))
diab_model.add(Dense(units=1, kernel_initializer='uniform', activation='sigmoid'))
print(diab_model.summary())
opt = SGD(lr=0.03, momentum=0.9)
diab_model.compile(loss = 'binary_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 15)	135
dropout (Dropout)	(None, 15)	0
dense_3 (Dense)	(None, 3)	48
dropout_1 (Dropout)	(None, 3)	0
dense_4 (Dense)	(None, 1)	4

=====
 Total params: 187
 Trainable params: 187
 Non-trainable params: 0

Figure 10: Model for diabetes detection

The heart prediction model uses the Adam optimization algorithm(an extension of the SGD algorithm) to update weights iteratively based on training data.

```
| heart_model = Sequential()
| heart_model.add(Dense(units=10, input_dim=8, kernel_initializer='uniform', activation='relu'))
| heart_model.add(Dropout(0.4))
| heart_model.add(Dense(units=3, input_dim=8, kernel_initializer='uniform', activation='relu'))
| heart_model.add(Dropout(0.4))
| heart_model.add(Dense(units=1, kernel_initializer='uniform', activation='sigmoid'))
| print(heart_model.summary())
| heart_model.compile(loss = 'binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Model: "sequential_4"

Layer (type)	Output Shape	Param #
dense_10 (Dense)	(None, 10)	90
dropout_4 (Dropout)	(None, 10)	0
dense_11 (Dense)	(None, 3)	33
dropout_5 (Dropout)	(None, 3)	0
dense_12 (Dense)	(None, 1)	4
Total params: 127		
Trainable params: 127		

Figure 11: Model for heart disease detection.

4.6 Model Fitting

Hyper-parameters determine how the neural network is trained and the network structure.

The table below gives a summary of the Hyper-parameters used during model fitting.

Heart Disease prediction model	Diabetes prediction model
validation_split=0.2	validation_split=0.2
epochs=300	epochs=200
batch_size=25	batch_size=5
verbose=2	verbose=2

Table 1: Hyper-parameters used during model fitting

4.7 NCD Detection System Physical Architecture

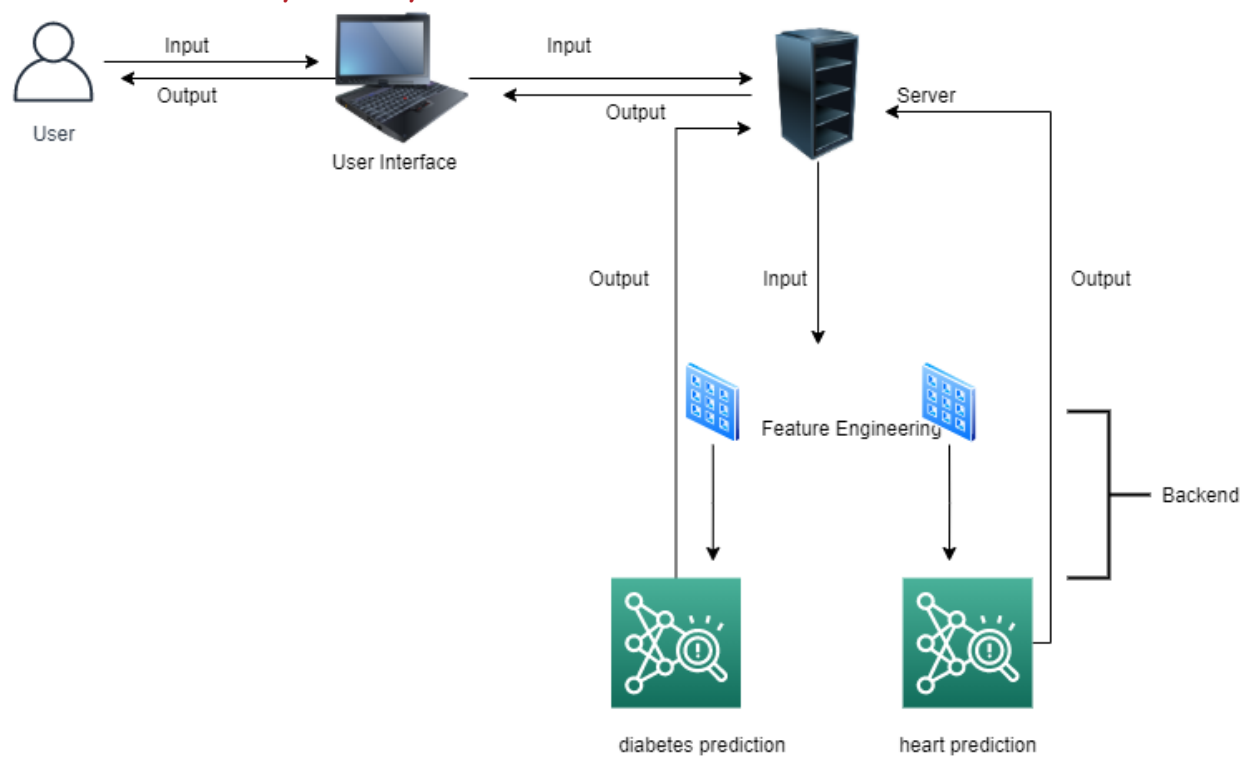


Figure 12: Physical architecture of the NCDs detection system

4.8 User Process Flow

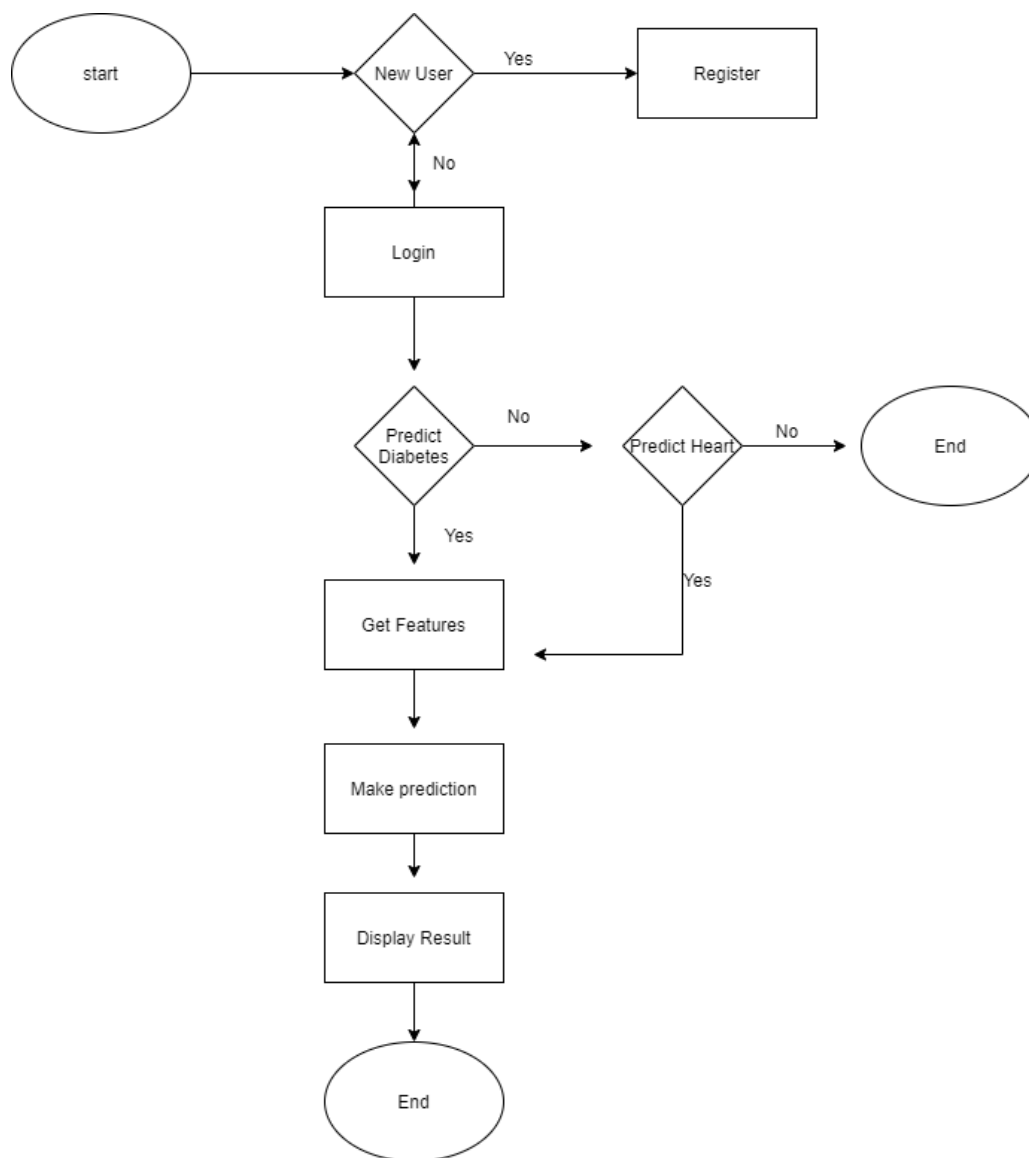


Figure 13: User process flow

Chapter 5: Evaluation and Results

5. 1 Introduction

This chapter will present the results obtained from testing the system. The test will be carried out of different inputs and model testing. For this project, the `train_test_split` function was used to separate the dataset into two parts which are the train set and test set. The train data set was used to train the model, and the test data set was used in evaluating the trained model. In total, 80% of the data set was used to extract features for heart disease and diabetes and train the deep learning model. The remaining 20% of the data which constituted the test set was used for model validation and evaluation.

5. 2 Model Testing

So far, the NCD detection system has two models, one for risk of diabetes detection and the other for detecting heart disease. Performance metrics such as accuracy, precision, `f1_score`, loss, recall, and area under the curve have been used.

- **Accuracy** gives the percentage of correct predictions for the test data set. The following formula provides the accuracy with the score;

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{Total prediction}}$$

- **Precision** tells us how accurate the model is when it says an individual is at risk of disease. The following formula gives precision;

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall** tells us how good our model is at predicting people who are actually at risk of disease. The following formula gives it;

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{false negatives}}$$

- **F1_score** is the harmonic mean of precision and recall and is a better measure accuracy score. It takes both the false positives and false negatives into account. The following formula gives it;

$$\text{F1_score} = 2 * \left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \right)$$

- **Area Under the Curve (AUC-ROC)** measures the ability of our model to distinguish between someone who is at risk of disease and someone who is not at risk of disease. The higher the value, the better the model's performance at distinguishing between an individual at risk of the disease and an individual who is not.

The table below summarizes the performance of diabetes and heart disease models based on Accuracy, Precision, F1_score, Recall, & AUC metrics.

Models	Accuracy	Precision	f1_score	Recall	AUC
Diabetes	0,94	0.96	0.93	0.91	0.98
Heart Disease	0.88	0.92	0.87	0.84	0.96

Table 2: Performances of the models' Accuracy, Precision, F1_score, Recall, & AUC metrics

Diabetes and Heart Disease

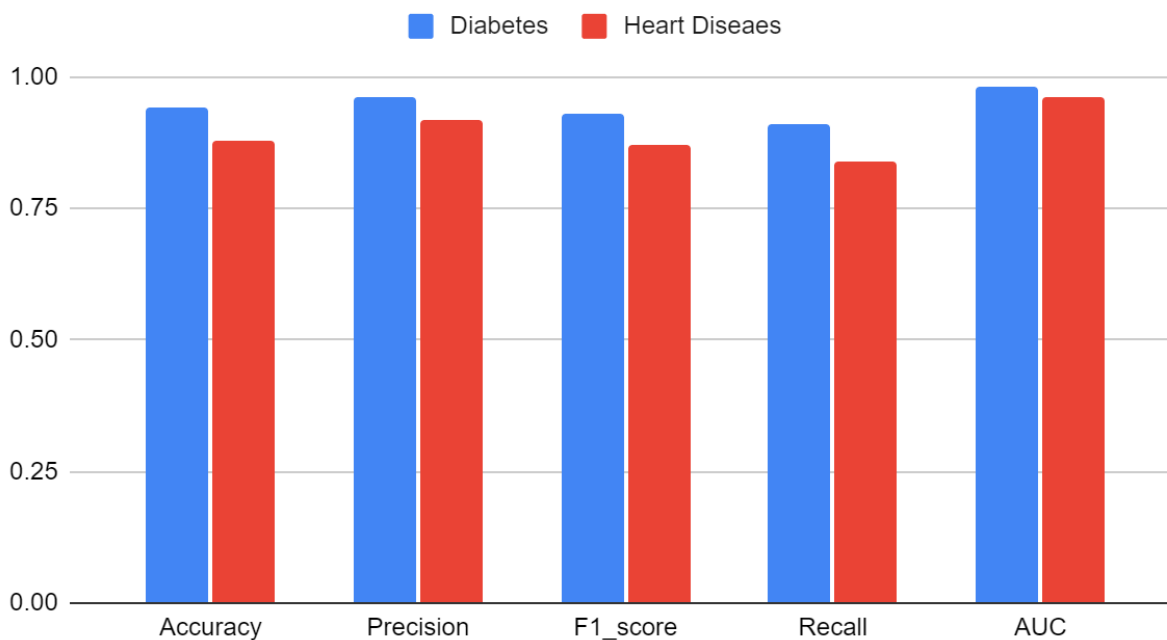


Figure 14: Accuracy, Precision, F1_score, Recall, & AUC of heart disease model vs. diabetes model

Below is a table summarizing the performances of diabetes and heart disease models based on Accuracy & AUC metrics in percentages.

	Accuracy	Area Under the curve
Heart Disease	88.52%	96.94%
Diabetes	94.23%	98.61%

Table 3: Performance-based on accuracy and area under the curve of the models

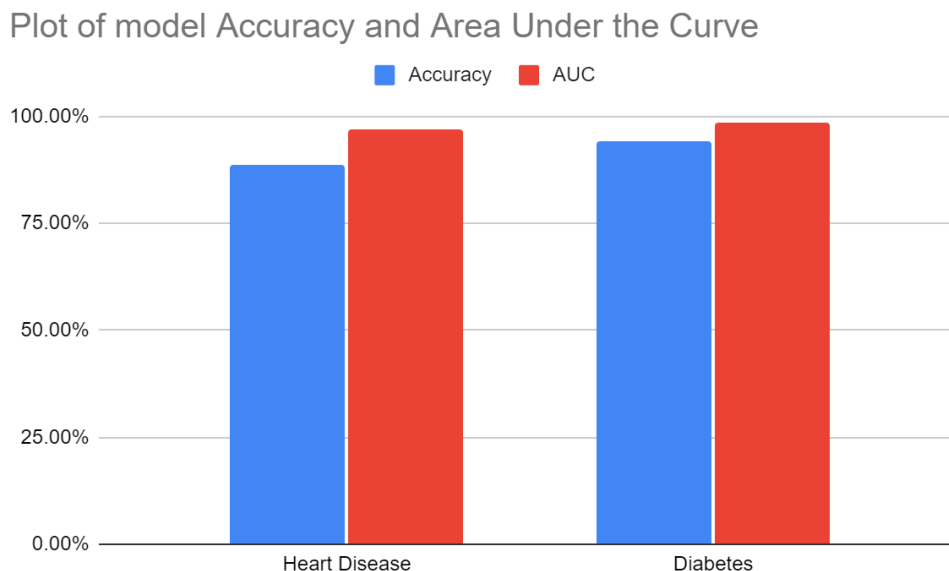


Figure 15: Accuracy and area under the curve of diabetes and heart disease models

5.3 Model performances analysis

The performance of the Diabetes and Heart, disease detection system depends on the size of the data you have. We can see this from the accuracy of the two models. The model for diabetes detection has a higher accuracy of 94.23%, while that for heart disease detection has a lower accuracy of 88.52%. This is because the diabetes data set has 768 data points, while the heart disease data set has 303 data points. The graphs plotted above clearly show how the heart diabetes model with more data performs well in Accuracy, Precision, F1_score, loss, Recall, and Area Under the Curve.

5.4 Model Accuracy: Confusion Matrix

In this section, the confusion matrix is used to help us see how confused our model is when making predictions. It will summarize the number of correct and incorrect predictions with count values.

	Negative	Positive
Negative	TN True Negative	FP False positive
Positive	FN False Negative	TP True Positive

Figure 16: Sample confusion matrix

5.5 Summary of confusion Matrix

True Positives gives the number of individuals the model predicted to be at risk of disease, and they are at risk of the disease.

True Negatives give the number of individuals our model predicts not to be at risk of the disease, and they are indeed not at risk of the disease.

False Positives give the number of individuals the model predicts to be at risk of the disease, but they are not at risk of the disease.

False Negatives gives the number of individuals the model predicts not to be at risk of the disease, but they are at risk.

Therefore, having a high False Negatives (FN) is considered more dangerous than having a high FP because of the rate of fatality, particularly in the medical domain.

Below is the confusion matrix for our diabetes and heart disease detection models.

```
[[36  3]
 [ 3 62]]
```

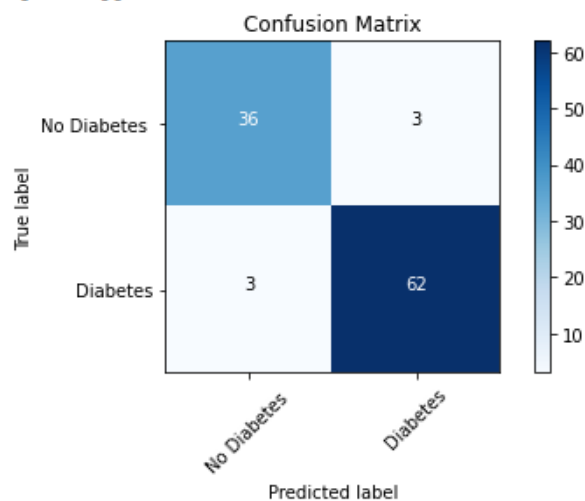


Figure 17: Confusion matrix for diabetes detection Model.

Above is the Confusion Matrix for the Diabetes detection model. The results are summarized in the table below.

Metrics	Count
True Positives	62
True Negatives	36
False Positives	3
False Negatives	3

Table 4: Summary of diabetes detection model confusion matrix

Below is the confusion matrix for our diabetes and heart disease detection models.

```
[[26  2]
 [ 5 28]]
```

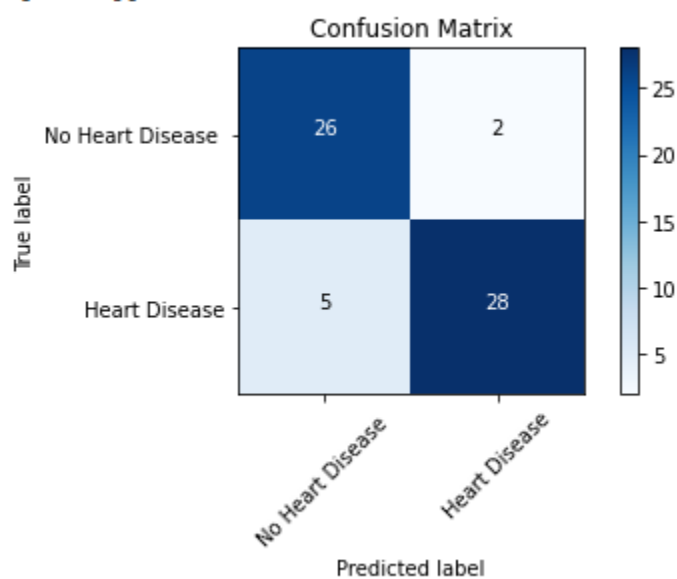


Figure 18: Confusion matrix for heart disease detection Model.

Above is the Confusion Matrix for the Heart disease detection model. The results are summarized in the table below.

Metrics	Count
True Positives	28
True Negatives	26
False Positives	2
False Negatives	5

Table 5: Summary of heart disease detection model confusion matrix

5.6 Sample Inputs and Results

The tables below show sample inputs given to the system and the results obtained.

Diabetes	At-Risk	Not at Risk
Age	62	28
Alopecia	Yes	No
Polyuria	Yes	No
Polydipsia	Yes	No
Gender	Male	Female
Itching	yes	Yes
Delayed Healing	No	Yes
Irritability	No	No
Prediction probability	0.99	0.28

Table 6: Sample inputs and results for diabetes detection model

Heart Disease	At-Risk	Not at Risk
Age	34	67
Maximum heart rate	130	100
Depression induced by exercise	2.5	1.3
Number of major vessels colored by fluoroscopy	2	1
Serum cholesterol in mg/dl	85	60
Resting blood pressure in mmHg	112	100
Chest Pain Types	3	1
Gender	Female	Male
Prediction probability	0.73	0.43

Table 7: Sample inputs and results for heart disease detection model

5.7 User Interface of the NCDs Detection System

Supported NCDs

☒ Diabetes

☐ Heart

☐ About

You are making prediction for Diabetes

Your age in Years

0.00 - +

Alopecia - Sudden loss of hair that starts with one or more circular

Yes ▾

Polyuria - Excessive Urination

Yes ▾

Polydipsia - Abnormal increase in thirst

Yes ▾

Sex

Male ▾

Itching - Dry itchy skin

Yes ▾

Delayed healing

Yes ▾

Irritability - Rapid changes in Mood

Yes ▾

Predict Diabetes

Figure 19: User interface for the NCDs Detection System

Chapter 6: Conclusion and Recommendations

6.1 Conclusion

The NCDs detection system can tell when you are at risk of diabetes/heart disease or not when provided with data. The output gotten from the system after giving it input is a probability between 0.0 to 1.0. If an individual's likelihood after providing input is more significant than 0.5, the individual is considered at risk of the disease. Otherwise, they are not at risk of the disease. The selected features identified to contribute to diabetes were age, alopecia, polyuria, polydipsia, gender, itching, delayed healing, and irritability. The identified features to contribute more to heart disease were age, maximum heart rate, depression induced by exercise, number of major vessels colored by fluoroscopy, serum cholesterol, resting blood pressure, chest pain type, and gender. These features were extracted using the xgboost algorithm. Although other existing studies used different data sets, age seems to be a major contributing factor to the occurrence of NCDs.

The system also tells us whether we are at risk of diabetes/heart disease or not and provides us the probability or confidence level. The greatest challenge faced was getting data of a specific group of people like Rwanda or any other African country, so we used data from an online repository. Although data online was used to develop the system, the size of the data was petite, which negatively affected the model's performance. The system was deployed using the streamlit library, which is a Python framework made for data scientists.

One limitation of this system is that the results rely on a good internet connection, which could be a problem for users living in low-bandwidth areas. As a future work, it would be interesting to investigate the direction of offline disease detection systems where internet connectivity is not required. .

6.2 Recommendations

- Doctors can use the heart disease detection model as a clinical decision support system.
- The diabetes model can be used by any individual who wishes to know their state of diabetes as the input does not require you to visit the laboratory.

Chapter 7: Bibliography

- [1] WHO, "Noncommunicable diseases," *who.int*, 2020.
https://www.who.int/health-topics/noncommunicable-diseases#tab=tab_1 (accessed Dec. 10, 2020).
- [2] J. J. Bigna and J. J. Noubiap, "The rising burden of non-communicable diseases in sub-Saharan Africa," *The Lancet Global Health*, vol. 7, no. 10. Elsevier Ltd, pp. e1295–e1296, Oct. 01, 2019, doi: 10.1016/S2214-109X(19)30370-5.
- [3] "The impact of the COVID-19 pandemic on noncommunicable disease resources and services: results of a rapid assessment."
<https://www.who.int/publications/i/item/ncds-covid-rapid-assessment> (accessed Feb. 17, 2021).
- [4] M. Hu, Y. Nohara, Y. Wakata, A. Ahmed, N. Nakashima, and M. Nakamura, "Machine Learning Based Prediction of Non-communicable Diseases to Improving Intervention Program in Bangladesh," *Eur. J. Biomed. Informatics*, vol. 14, no. 4, 2018, doi: 10.24105/ejbi.2018.14.4.5.
- [5] A. Dinh, S. Miertschin, A. Young, and S. D. Mohanty, "A data-driven approach to predicting diabetes and cardiovascular disease with machine learning," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, p. 211, Nov. 2019, doi: 10.1186/s12911-019-0918-5.
- [6] S. M. Pasha and S. Ankalaki, "Diabetes and heart disease prediction using machine learning algorithms," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 7, pp. 3247–3252, Jul. 2020, doi: 10.30534/ijeter/2020/60872020.
- [7] P. Kadu and A. Buchade, "Non communicable disease prediction system using machine learning," *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 1307–1311, 2019, Accessed: Nov. 26, 2020. [Online]. Available: www.ijstr.org.
- [8] G. Battineni, G. G. Sagaro, N. Chinatalapudi, and F. Amenta, "Applications of machine learning predictive models in the chronic disease diagnosis," *J. Pers. Med.*, vol. 10, no. 2, 2020, doi: 10.3390/jpm10020021.
- [9] "What is Artificial Intelligence (AI)? | IBM."
<https://www.ibm.com/cloud/learn/what-is-artificial-intelligence> (accessed Feb. 16, 2021).
- [10] "Machine Learning - an overview | ScienceDirect Topics."
<https://www.sciencedirect.com/topics/computer-science/machine-learning> (accessed Feb. 16, 2021).
- [11] Y. Khan, U. Qamar, N. Yousaf, and A. Khan, "Machine learning techniques for heart disease datasets: A survey," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1481, pp. 27–35, 2019, doi: 10.1145/3318299.3318343.
- [12] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve

cardiovascular risk prediction using routine clinical data?," *PLoS One*, vol. 12, no. 4, p. e0174944, Apr. 2017, doi: 10.1371/journal.pone.0174944.

- [13] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," 2011. Accessed: Feb. 17, 2021. [Online]. Available: <http://scikit-learn.sourceforge.net>.
- [14] W. McKinney, "Data Structures for Statistical Computing in Python," 2010.
- [15] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825. Nature Research, pp. 357–362, Sep. 17, 2020, doi: 10.1038/s41586-020-2649-2.
- [16] M. Abadi, "TensorFlow: learning functions at scale," *ACM SIGPLAN Not.*, vol. 51, no. 9, pp. 1–1, Dec. 2016, doi: 10.1145/3022670.2976746.