

UNIVERSITY OF THE PHILIPPINES MANILA  
COLLEGE OF ARTS AND SCIENCES  
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

CHITS DIABETES DATA WAREHOUSE WITH BIG  
DATA ANALYTICS

A special problem in partial fulfillment  
of the requirements for the degree of  
**Bachelor of Science in Computer Science**

Submitted by:

Reuben Joseph B. Cabrera

June 2018

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

## ACCEPTANCE SHEET

The Special Problem entitled “CHITS Diabetes Data Warehouse with Big Data Analytics” prepared and submitted by Reuben Joseph B. Cabrera in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

---

**Ma. Sheila A. Magboo, M.Sc.**  
Adviser

### EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. ( <i>cand.</i> )	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, Ph.D. ( <i>cand.</i> )	_____	_____
4. Perlita E. Gasmen, M.Sc. ( <i>cand.</i> )	_____	_____
5. Marvin John C. Ignacio, M.Sc. ( <i>cand.</i> )	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
7. Geoffrey A. Solano, Ph.D. ( <i>cand.</i> )	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

<hr/> <b>Ma. Sheila A. Magboo, M.Sc.</b> Unit Head Mathematical and Computing Sciences Unit Department of Physical Sciences and Mathematics	<hr/> <b>Marcelina B. Lirazan, Ph.D.</b> Chair Department of Physical Sciences and Mathematics
---	---

---

**Leonardo R. Estacio Jr., Ph.D.**  
Dean  
College of Arts and Sciences

## **Abstract**

Electronic medical records have made data collection efficient allowing health professionals and researchers to analyze patient records faster. CHITS, the first EMR in the Philippines has collected millions of data across different areas in Luzon however, CHITS does not contain any data analysis module. This project aims to make use of the large amount of data available in CHITS by developing a diabetes recommender system with big data analytics. The system recommends treatments by finding similar patients by analyzing patient profiles through the use of a similarity metric called Cosine Similarity.

*Keywords:* Recommender System, Big Data Analytics, Cosine Similarity

# Contents

<b>Acceptance Sheet</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>I. Introduction</b>	<b>1</b>
A Statement of the Problem . . . . .	4
B Objectives of the Study . . . . .	4
C Significance of the Project . . . . .	5
D Scope and Limitations . . . . .	5
<b>II. Review of Related Literature</b>	<b>7</b>
<b>III. Theoretical Framework</b>	<b>12</b>
A Diabetes . . . . .	12
B CHITS . . . . .	17
C Big Data . . . . .	18
D Recommender System . . . . .	19
E Collaborative Filtering . . . . .	19
F Python . . . . .	20
G Finnish Diabetes Risk Score . . . . .	20
<b>IV. Design and Implementation</b>	<b>21</b>
<b>V. Results</b>	<b>26</b>
<b>VI. Discussions</b>	<b>35</b>
<b>VII. Conclusions</b>	<b>39</b>
<b>VIII. Recommendations</b>	<b>41</b>

<b>IX.</b>	<b>Bibliography</b>	<b>43</b>
<b>X.</b>	<b>Appendix</b>	<b>47</b>
	A Source Code . . . . .	47
<b>XI.</b>	<b>Acknowledgement</b>	<b>68</b>

# List of Figures

1	Context Diagram	21
2	Use Case Diagram for Health Professional/Researcher	22
3	Homepage of the System	26
4	Patient Profile Form	27
5	Recommender System Results	28
6	Recommender System Results	28
7	Top 5 Recommendations	29
8	Score Results	29
9	Search page with Datatable	30
10	Select Field: Age Group	30
11	Select Field: DM Type	31
12	Select Field: Sex	31
13	Datatable Results	32
14	Datatable Results when using search	32
15	Datatable Results with pagination	33
16	Dashboard Results	33
17	Dashboard Results	34
18	Dashboard Results	34

# I. Introduction

Diabetes is a chronic disease caused by the deficiency in production of insulin by the pancreas, or by the ineffectiveness of the insulin produced. Deficiency of insulin results in increased concentrations of glucose in the blood, which in turn damage many of the bodys systems particularly the blood vessels and nerves. [1]

There are two major types of diabetes, type 1 and type 2 diabetes. Type 1 diabetes occurs when the immune system attacks and destroys the insulin. Type 2 diabetes, the most common form of diabetes is caused by several factors such as lifestyle and genes. [2] Individuals with diabetes frequently experience symptoms such as frequent urination, blurry vision, increased hunger, extreme unexplained fatigue, frequent skin,bladder or gum infections and many more. Diabetes can be acquired through heredity. People who have close relatives who have history with diabetes are more likely to develop as well. Obese/Not physically active individuals are also likely to develop type 2 diabetes. Extra weight sometimes causes insulin resistance and is common in people with type 2 diabetes. [3]

Diabetes caused 57,000 deaths annually therefore ranked 4th most deadly disease in the Philippines. Health professionals predicted that globally, the number of people with diabetes would reach 320 million by 2025, however it has already been recorded that the number has reached over 400 million. According to the International Diabetes Federation, there were 3.5 million cases of diabetes in the Philippines in 2015 excluding the estimated 1.84 million cases that are presumed to be undiagnosed. [4] Last year, it was recorded that there are at least 6 million Filipinos diagnosed to have diabetes. According to Dr. Augusto Litonjua, President of the Philippine Center for Diabetes Education Foundation, this record could double or grow even further by 2040 because of undiagnosed diabetes cases. [5] Due to the increasing number of diabetes cases, Philippines is now being seen as as a diabetes hot spot by the Philippine Society of Endocrinology, Diabetes, and Metabolism (PSEDM). The cause of the growing number of diabetes cases were due to lifestyle among Filipinos. [6]

The number of diabetes cases increases due to the unhealthy lifestyle of many Filipinos particularly unhealthy eating patterns. Filipinos consume mainly rice, bread and other foods with high level of sugar content. [7] Filipinos are also fond of consuming calorie-dense food from fast food chains which may lead to obesity and a higher risk for diabetes.[5]

Reports show that undiagnosed cases are due to the following reasons: Education, Economic Standing, Cultural Issues, Folk Medicine Belief.

Due to lack of education, diabetes symptoms and treatment are not well taught in many countries (including Philippines). Filipinos lack awareness of what causes diabetes. Some may have financial issues that they are unable to visit or consult their doctor on a regular basis to see if they are diagnosed properly. Some may also resort to Folk Medicine Belief as it is cheaper but, it doesn't properly manage the diagnosis. [8]

The cost of medical treatments also keeps on increasing. One of the responsibilities of our government is to provide an effective healthcare system that should improve the healthcare management in the Philippines. One of the existing healthcare systems in the Philippines is an electronic medical record System called Community Health information Tracking System or CHITS. It was developed by the National Telehealth Center to further improve health information management at the rural health unit level. CHITS allows health personnel to be able to gather data and generate reports as needed for further analysis. 142 barangays were the first to use CHITS. The said system is also a platform for the health workers by allowing them to explore eHealth applications such a telemedicine and eLearning. [9, 10]

The eHealth program is not only limited to the health workers but to also researchers as the eLearning program is a training for anyone who is interested in doing research on eHealth. Rural Communities have difficulties in gaining access of health professional care. eMedicine allows patients to services from specialized doctors stationed in cities by allowing them to be referred directly to the clinical



specialist or to the Philippine General Hospital(PGH). As an eRecord, CHITS can easily work with other information system that would allow better information sharing to other platforms that would need for health information records that can be used for further analysis. Monitoring of health performance is done through the eSurveillance program by using mobile technology and web-based system allowing them to achieve goals of the DOH priority programs. The program gives guidance to the community on social protection, basic education, civil registration and disaster management through mReports. .[9]

Healthcare data generated everyday from various sources grows exponentially. Application of Big Data Analytics allows us to make use of the vast amount of data by analyzing large and complex healthcare data. Healthcare quality is further improved when researchers are able to generate insights that could help explain relationships in the given data. Examples of how big data analytics is applied in healthcare are predicting diseases when they will be likely to occur or providing a patient centric treatment which means that the medication is calibrated and focus on the patient's conditions. [11, 12]

Health Recommendation Systems(HRS) is a specialization of a recommender system driven by individualized health data such as a personal health record. The main goal of an HRS is to supply the user with medical information which is meant to be highly relevant to the medical development of the patient. [13]

Recommender System are powered by an algorithm called Collaborative Filtering. This type of algorithm takes into account the user's preferences and recommends items that are similar to the user's interest. [11] The idea behind collaborative filtering is combining the users preference and item ratings which are critical factors in providing the best recommendations to the user. some of these examples are tv shows (Netflix) which sees what genre the user is interested in and recommends movies/tv shows that are highly rated and with the same genre.

In the context of health recommender systems, item ratings do not exist however, another way to provide a recommendation based upon a health record of a

patient is through the use of a similarity score. The idea of this type of algorithm is to compare the new user to existing users in the database and find which user is the most similar and the system recommends treatments(diet plans,exercise regimen, etc) to the new user. [14]

## **A Statement of the Problem**

Large volumes of multi-structured healthcare data are difficult to store, manage, and becomes useless when data is not analyzed. Healthcare data comes from many different sources such as physician notes, medical history, electronic health record and many more. It is important to properly organize these data for more efficient analysis and to generate accurate medical recommendations.

CHITS contains large amounts of patient data however, the system does not contain a data analytics module that will allow healthcare professionals/providers to have a decision support tool provides information about non medical aspects such as diet plans, exercise regimen or even for diabetes management.

To address these problems, it is essential to use Big Data Analytics to be able to process large amounts of data and gain insights from the analysis. Through data mining, the system can provide recommendations that are calibrated to the patients attributes therefore providing the best recommended treatment. This will also allow healthcare professionals to gain insight which will help them in understanding the relationships between factors concerning the increase of diabetes cases allowing them to provide an holistic assessment of the patient.

## **B Objectives of the Study**

This research will create a Diabetes Recommender System in CHITS with the following functionalities:

1. Allows local physician/researcher to
  - (a) Input Patient Case information

- (b) View Recommendation output
- (c) Be able to do advance search queries
- (d) View medical record of patients
- (e) View statistical trends/dashboard
- (f) View Information regarding Diabetes
- (g) Allow extraction of patient record for anonymized data using SQL commands

## **C Significance of the Project**

Big Data analytics is essential in improving quality of healthcare. Patient data contains vast information that when analyzed properly, may give valuable insights that helps in maintaining diabetic patients. The recommender system will be useful to as a guide to decision making process of a physician and helpful to improving diabetes research.

The system will allow CHITS to have a proper data warehouse and a data analytics component for Diabetes. The system will provide the health professional a guide/recommendation on how to treat the patient based on the data provided. This will allow a faster and better judgment for the professional as the system gives a guidance based on previous cases that are similar to the current case. The system minimizes cost as it makes information-retrieval faster compared to the traditional paper method. Researchers/Health professionals would be able to run analytic queries and view important relationships and trends on the increase of diabetes cases. Effective analysis lessens the frequency of the patient visiting the health professional therefore saving time and money.

## **D Scope and Limitations**

1. The recommendation model will be based on the CHITS dataset locally.
2. The system provides recommendation treatments to Diabetes cases only

3. The basis for comparison will be based on the following features
  - (a) Weight
  - (b) Height
  - (c) Waist Circumference
  - (d) Hip Circumference
  - (e) Age
  - (f) Gender
  - (g) Hba1C result
  - (h) FBS result
4. The recommendations given by the system are based on the treatments provided by a doctor that is recorded in CHITS
5. The system will not save patient consultation in this module but rather in the original CHITS module
6. extraction of anonymized data is limited to deletion of name
7. The type of recommender system is limited to collaborative filtering by using similarity metrics.
8. The type of parameter to be measured is limited to similarity score as it is applicable to health care data
9. The system is to be deployed upon separate workstation
10. Efficiency of data mining is dependent on machine workstation locally installed.

## II. Review of Related Literature

There has been a lot of health recommender system that has been developed by using Big Data Analytics. Most recommender systems use a technique called Collaborative Filtering. Collaborative Filtering is a data mining technique to predict a user's opinion about an item or a service based on the known preferences of a large group of users.[14] Systems that utilize big data analytics must also use frameworks such as Apache Spark or Apache Mahout to be able to process large amounts of data in a short amount of time. Most of the systems have a similar flow on how they develop the recommender engine. The following systems share a common goal: to improve quality and efficiency to healthcare.

A health recommender system developed by the authors in[11] focuses on diabetes. The data analysis was done by utilizing Spark framework. The system used Naive Bayes algorithm to classify the patients by Pre-diabetic, Diabetes, and no Diabetes. To develop the hybrid recommender system, they used Content filtering, Rule and Case based filtering. Rule based filtering is used to filter profiles based on user queries. Case based filtering is used to extract similar profiles based on patient health history. To further improve the case filtering approach, they used the K-Nearest Neighbor Algorithm. The process is done by comparing the score of the new case and existing cases and nearest score will be given as the suggestion. The study presented on how the recommender system can be trusted by building recommender systems that suggest treatments from reliable sources(ex. Health on Net authority).

A study was done by Chaitanya and Ashmin Kaul and Saurav Verma on healthcare prediction systems using big data. Their study discusses three algorithms:(1) Collaborative and Recommendation Engine(CARE); (2) Collaborative Health Care System Model (COHESY) and (3) Hierarchical Association Rule Mining Model (HARM). CARE utilized collaborative filtering for analysis predictions of the disease. The process begins by selecting a patient to be tested. A cluster based on matching of diseases was also formed and collaborative filtering process

was applied to predict the patient's next visit. COHESY is based on a collaborative filtering mechanism for analyzing and recommending exercises that can improve a person's health. the algorithm suggests activities by modifying the value of any health parameter to which the algorithm is being tested. HARM, on the other hand is based on the concept of associative mining through which the patient's future symptoms can be predicted against the current or the past symptoms. The study claims that these algorithms can be used more efficiently and generate more promising results with larger datasets.[15]

The system built by [16] was to predict the personalized mortality prediction by comparing the present case patient and a past patient recorded in an EMR by using a patient similarity metric. They hypothesize that analyzing similar patients can give a personalized and precise prediction. Idea of their hypothesis was the amount of predictive utility contributed by a past patient should be directly proportional to the degree of similarity between the past and index patient. To compare similar patients, they utilize a patient similarity metric called cosine similarity which calculates the dot product between the two patients which gives an cosine of the angle as the result. The result of this algorithm ranges from -1 to 1 which signifies minimum and maximum similarity respectively. They used this algorithm to separate the dataset to the most similar patients to train the dataset using predictive models namely death counting, logistic regression and decision tree. The idea behind using cosine similarity first to find similar patients was to improve mortality prediction rather than train the dataset with the whole population. [16]

The aim of this recommender system was to build a friend recommendation system by using lifestyle of users to suggest friend as required and system should be used for health application. The system was divided into 5 modules. (1) Data Collection, (2) Data Analysis, (3) Indexing, (4) Friend-Matching Module and (5) Recommendation Module. The method to build the recommendation was to use the knowledge of probability distribution and calculate lifestyle vectors, find similar users to user I by applying cosine similarity then sort the results. The

proposed method added a feature which applies by recommending based on blood group which can be used in healthcare applications. [17]

A data-driven recommender system was built by state author that is used for therapy decision support to assist physicians or other health professionals during clinical decision making. Their approach was to apply methods from data mining and machine learning to extract knowledge from the data to achieve more individual recommendations, learn from experience and reveal unknown patterns in the dataset. They also describe the use of similarity metrics for rating prediction and recommendation computation which are capable of processing sparse and inhomogeneous data vectors as compared to the use of collaborative filtering or hybrid approaches. [18]

The system built by author is an ICF based automated system that stores and monitors health status of patient with spinal cord injuries in order to recommend rehabilitation methods for patients based on his/her status. The system uses similarity measures to implement the recommender system namely cosine, correlation, spearman, hamming, and jaccard. The system consists of three phases (1) preprocessing phase in which the medical description of each patient status is converted into its equivalent ICF code. (2) Similarity phase which aims to get the similarity between two vectors by applying the similarity measures then getting the top score for the (3) recommend phase which gets the data of the patient with the highest score and the data returned to the system will be the recommended rehabilitation method for the current patient. The results of this study show that the cosine similarity measure gives the most accurate result. [19]

A recommender system was built for health personnel by recommending courses that would further stimulate their education. The goal was to recommend courses based on the user's course registration history and user's interesting area. The steps to build the recommendation system were divided into three steps. (1) To select and to provide the correct data information source as input to the recommender system. Next, was to build the recommendations. And last (3) was to persist the

recommendations in an appropriate data collection. To calculate similar users, they implemented the cosine-similarity measures which calculates the distance between two points(vector). to implement the algorithm, they used Python, Pandas which is an open-source library which provides easy-to-use data structures and data analysis tools in Python and SciPy which is a scientific computing tool for Python. The similarity of the current course to any other course is stored in a similarity matrix which contains the ID of the course and the similarity score. The output of the system was a list of users and their five recommended courses.[20]

The purpose of this study is to survey the existing patient centric social networks and its functions, recommendation techniques and evaluation methods used. A health recommender system is described as a special type of recommender system that enables patients to find other patients similar to them and connects them by how social network connects people and share information such as experiences or treatments or may recommend caregivers to patients. The techniques applied in a health recommendation system uses the following. (1) Patient Data Similarity which is based on the distance among continuous data items found in the patient profile by suitable weighting. (2) Patient Similarity using Conditions and Treatments which finds similarity based on terms of patient conditions and treatments. (3) Content Matching which is based on the content of the users on similar topics then they might be interested in getting to know each other. This is computed using the cosine similarity measure. This type of system can be used to help patients find a solution same to their condition or symptoms. [21]

The CARE system by Nitesh V. Chawla, PhD and Darcy A. Davis, PhD was to serve as a data-driven computational aid for physician assessing the disease risks facing their patients. In their study, healthcare has transformed from a disease-centered model towards a patient-centered model. The difference between the two model is in the disease-model, the health professional decision making is centered around clinical expertise and data from medical evidence and tests while the patient-centered model, patients actively participate on their own care and re-



ceive services focused on individual needs and preferences, informed by advice and oversight from their healthcare providers. They also utilized collaborative filtering specifically using inverse frequency and vector similarity. They also explained the challenge involved using collaborative filtering as diseases do not have a rating system as to what books or movies have. the turn-around for this problem is that to compare similar cases between patients.[14]

The android application made by the authors in [22] used a different algorithm called Pearson Correlation for matching patient with similarities with other patients in the database. The algorithm compares the patients and rating them of values ranging from -1 to 1 with a score of 1 means that the two patients are perfectly correlated. The closest patient is the one which has the score that is closest to 1. Once the closest patient is found, the system then recommends the same treatment(diet,exercise,medication) to the new user. The overall system includes a management module and reminding function which are also essential components to improve healthcare. The application can also be extended to other diseases to provide other kinds of healthcare guidelines recommendations.

### III. Theoretical Framework

#### A Diabetes

Diabetes is a metabolic disease in which patients have abnormally high blood glucose(BG).[23] This condition influences the human body by diminishing the insulin which conveys glucose into the platelets. This expands the sugar level in the body causing difficulties like stroke, heart disease, blindness, kidney failure and death.[24]

##### 1. Types

There are two main types of Diabetes:

- (a) Type 1 - a disease in which the pancreas produces little or no insulin. This results into glucose building up in the blood instead of being used for energy. Individuals with Type1 Diabetes need to inject insulin as prescribed.[23]
- (b) Type 2 - a disease in which the pancreas does not produce enough insulin, or the human body does not properly use the insulin it makes.[23]
- (c) Gestational Diabetes - is a type of diabetes that occurs in pregnant women.[24, 25]

##### 2. Symptoms:[26]

- (a) Loss of weight
- (b) Obscured vision
- (c) Skin Infections
- (d) Frequent urination
- (e) Fatigue
- (f) Increased Thirst and Hunger
- (g) Blurred Visions

(h) Nausea

(i) Irritability

3. Factors:[\[27\]](#)

(a) Type 1

i. Family History

ii. Diseases of the Pancreas

iii. Infection/Illness

(b) Type 2

i. Obesity

ii. Impaired Glucose Tolerance

iii. Insulin Resistance

iv. Ethnic Background

v. Gestational Diabetes

vi. Sedentary Lifestyle

vii. Family History

viii. Polycystic ovary Syndrome

ix. Age

(c) Gestational

i. Obesity

ii. Glucose Intolerance

iii. Family History

iv. Age

v. Ethnic Background

4. Diagnosis

There are different medical test to identify and diagnose diabetes:[\[24\]](#)

(a) Fasting Blood Glucose Test (FBS)

- (b) Post Prandial Blood Sugar Test(PPBS)
- (c) Random Blood Sugar Level (RBS)
- (d) Oral Sugar Tolerance Test
- (e) Glycosylated hemoglobin(HbA1c)
- (f) Urine Test
- (g) There is also a protocol for type-2 diabetes called Self-Monitoring of Blood Glucose (SMBG) that is advised by healthcare providers to help the patient achieve proper blood glucose targets.[23] blood glucose is affected by many different factors, including diet, starting blood glucose level, physical activity, mental stress, illness, sleep patterns and many more. These factors has a different impact on a person's blood glucose. SMBG requires T2D patients to control these factors regularly and accurately especially diet.[23]
- (h) Diet Management

The purpose of a diet plan for diabetes patients is to control the blood sugar, manage weight and control risk factors for heart disease.

A diabetes diet is based on eating three meals a day at regular times. This allows the body to make use of the insulin it produces or get through a medication. Sometimes, a doctor may recommend a registered dietitian to further improved the diet plan. Dietitians may talk the patient into improving eating habits by chooisng the right portion size that suit the needs of the patient's size(by means of height and weight) and level of activity.[28]

Recommended diet:

- i. Healthy carbohydrates - Focus on healty carbohydrates such as fruits,vegetables, whole grains, legumes and low-fat dairy products
- ii. Fiber-rich foods - Fiber moderates how your body digests and helps control blood sugar levels.

- iii. Heart-healthy fish. Eat heart-healthy fish at least twice a week
  - Fish can be a good alternative to high-fat meat. Fish such as salmon, tuna are rich in omega-3 fatty acids that promote heart health by lowering blood fats called triglycerides.
- iv. Good fats - Foods containing monounsaturated and polyunsaturated fats can help lower your cholesterol levels.

Foods to avoid:

- i. Saturated fats - High-fat dairy products and animal proteins.
- ii. Trans fats- Types of fats are found in processed snacks, baked goods, shortening and stick margarines.
- iii. Cholesterol - include high-fat dairy products and high-fat animal proteins. Aim for no more than 200 mg of cholesterol a day. -
- iv. Sodium - Aim for less than 2,300 mg of sodium a day. For patients with hypertension, aim for less than 1,500 mg a day.

(i) Exercise Management

Diabetes prevention is crucial to maintaining health and prevent complications. Exercise is one proven way to help manage diabetes.

According to The American College of Sports Medicine and the American Diabetes Association, exercise

- i. plays a vital role in preventing and controlling blood sugar levels
- ii. can prevent/delay type 2 diabetes
- iii. can prevent diabetes during pregnancy.

Exercise is essential because it improves insulin sensitivity by helping the cells of the body use available insulin. It also stimulates a separate mechanism to allow the cells to use glucose for energy thus allowing regulation of blood glucose levels.

- i. Aerobic Exercise:

helps the body use insulin more effectively. Examples include:

- A. Swimming
- B. Rowing
- C. Cycling
- D. Basketball
- E. Dancing
- F. Skating
- G. Hiking and many more.

The amount of time recommended(ages 18-64) for aerobic activity is

- A. 30 minutes of daily moderate physical aerobic activity.
- B. at least 5 times a week.

ii. Strength Training:

helps lower blood sugar levels and increase insulin sensitivity. Examples include:

- A. weight machines
- B. Lifting free weights
- C. Resistance bands
- D. Exercises that use body weight such as sit-ups, squats, planks, and push-ups and many more.

The amount of time recommended for strength training is

- A. at least twice a week, in addition to the recommended amount of aerobic activity.

iii. Monitoring blood glucose levels when exercising

Monitoring blood glucose levels before and after exercise is essential. This indicates how well the body is responding to the exercise regimen and may help avoid blood sugar fluctuations which can be dangerous.

Testing of blood sugar should be done 30 minutes before exercise.

If they are:

- A. Lower than 100 milligrams per deciliter (mg/dL) - blood sugar may be too low to exercise. Low blood sugar is known as hypoglycemia.
- B. Between 100 and 250 mg/dL - this is the optimal range which is safe for most people to begin exercising.
- C. 250 mg/dL or higher - blood sugar may be too high to exercise. Carry out a urine test for ketones (which indicate more insulin is needed to control blood sugar)

During exercise, blood sugar should be tested every 30 minutes. Exercise should be stopped if the patient is experiencing blood sugar falls below 70mg/dl, weakness, tingling, and confusion.

Rechecking of blood sugar should be done several times. Physical activity can lower blood glucose for up to 24 hours.[\[29\]](#)

## **B CHITS**

Community Health Information Tracking System(CHITS) is the first electronic medical record system in the Philippines. It was developed to improve health information management at the RHU level by enabling users to gather data and generate reports needed for decision making. There are currently a total of 169 health stations across the Philippines using CHITS. For NCR, 63 health centers in Quezon City, 9 in Navotas, 29 in Taguig and 10 in Pasay City

CHITS started as a research on computer-based information system by the University of the Philippines Manila in 2004 and was taken over by UP Manila National Telehealth Center for its development and implementation in 2006.

CHITS is a capacity-building program which instills relevant health information systems components among health workers. CHITS runs on free and open source software. As of 2011, the second version of CHITS uses the Open Medical Record

System as a platform. It allowed the developers/organizations to modify or adopt the code without developing the system from scratch. This also allows CHITS software developers can study and modify the CHITS-openMRS freely.[30, 9]

Key Features of CHITS as an EMR:

1. user friendly interface which presents to the health worker a signified view of the various DOH program data fields.
2. codifies their patient's clinical conditions and the care the health workers provided
3. tracks patient health status
4. remind them of services of patients seeking or needing follow up care

## C Big Data

Big Data is defined as the exponential growth and availability of huge amount of data. It is believed and proven that with bigger data, leads to better and more accurate analysis and could lead to confident decision making which allows better judgment which results to higher operation efficiency, reduced risk and cost reduction.[31]

There are seven properties that makes Big Data different from other data.[32, 31, 33]

1. Volume - Large quantity of data (in terms of petabytes or more). Storing of data was an issue prior to advance and affordable storage devices.
2. Velocity - The pace at which the volume of data grows. It poses a problem with respect to time and efficiency therefore, there is a need to deal with the huge amount of data in real time
3. Variety - Data is growing in variety of formats. Examples would include structured,instructed, relational, non-relational, and different data sources such as videos,images, reports, etc.



4. Variability - Data is constantly changing.
5. Visualization - using graphs/charts that is more effective in conveying meaning/information compared to a spreadsheet.
6. Veracity - represents both the credibility of the data source and the suitability of the data for the target audience.
7. Value - after addressing the issues of the different properties mentioned above, an organization must gain some value from the data (insights, predictions etc).

## D Recommender System

Recommender Systems in general suggest items to purchase that takes into consideration the user's preferences. The system takes into account items similar to your past purchases or videos watched by other similar users. The prediction for which item to suggest is based upon similarity measures. Many techniques are used for recommendation task such as collaborative recommender systems, content-based recommender systems, knowledge-based recommender systems and various other techniques.[22]

## E Collaborative Filtering

Collaborative Filtering is the most common technique used in most recommender systems. The technique works by suggesting products/items to the user basing on user or item similarity.[11]

The method considers the ratings of other users in the system to make personalized predictions on an active user's preferences. The computation for rating prediction and generating recommendations are based on the similarity metrics which are capable of processing sparse and inhomogeneous data vectors.[34]

As mentioned in [14]. Healthcare data do not have ratings systems. The alternative to this solution is to utilize similarity metrics. To give a recommendation

to a patient, the system must compare similar cases between patients.

The following algorithm is used to compute for similarity:

1. Cosine Similarity

$$\text{Formula: } \cos(\theta) = \frac{AB}{\|A\|^2\|B\|^2} = \sum_i^n \frac{A_i B_i}{\sqrt{\sum_i^n A_i^2 \sum_i^n B_i^2}}$$

The algorithm returns the output similarity of numerical vectors A and B which ranges from 0 - 1 in which a value close to 0 indicates no correlation and a value close to 1 indicates the two items/person are similar.[20]

## F Python

Python is an interpreted, object-oriented, high-level Programming language used in many different applications such as data analysis, machine learning, web development and many more. [35]

Python includes libraries to help in data analysis:

1. Pandas - An open source, BSD licesend libraty providing high performance, easy-to-use data structures and data analysis for Python [36]
2. scikit-learn - a Python library for machine learning with simple and efficient tools of data mining and analysis which is built on Numpy,SciPy and matplotlib. [37]

## G Finnish Diabetes Risk Score

The Finnish Diabetes Risk Score is a questionnaire for assessing risk for type 2 Diabetes. The questionnaire consist of 8 questions and each question has a corresponding criteria that a person should encircle. Each answer corresponds to a number of points and is totalled after the person answers all the questions. the total points is matched to a particular risk assessment. For example, having a total score of below 7 states that the person has a low risk while a score higher than 20 implies a very high risk of type 2 diabetes.[38]

## IV. Design and Implementation

1. The diagram below shows the interaction between the CHITS personnel and the recommender system. The user inputs the patient case information by using the web application then the web app sends the data to a web server where the server feeds the data to the recommender model which will send back to recommendation back to the web server and back to the web application and be presented to the user through the user interface of the web application.

The system will be developed using Python. The recommender model and analytics will be implemented using Pandas and scikit-learn. The web application will be developed using Flask, a python web framework.

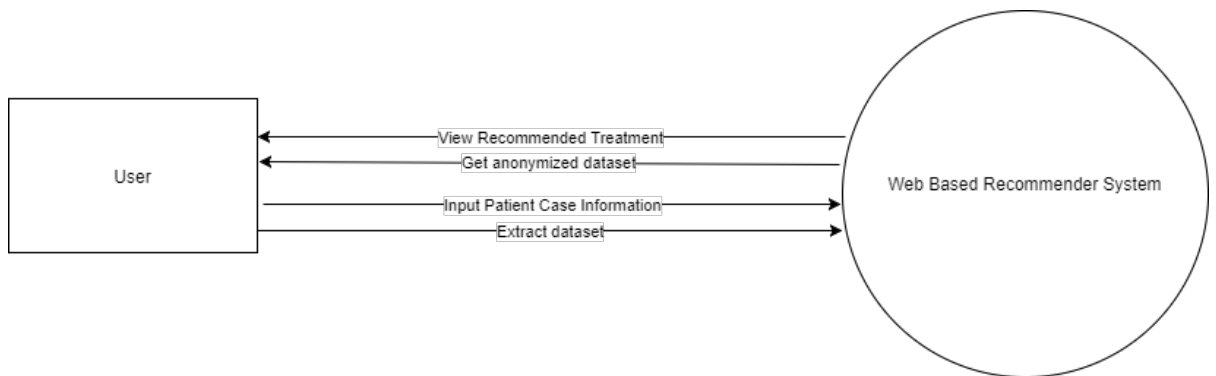


Figure 1: Context Diagram

2. The figure below shows the overall view of the system functionalities which focuses on the CHITS personnel. They can input patient case information to the system and view the recommendation output through the system's user interface. The system also allows the user to do search queries for specific information and be able to analyze statistical trends on diabetes through a dashboard.

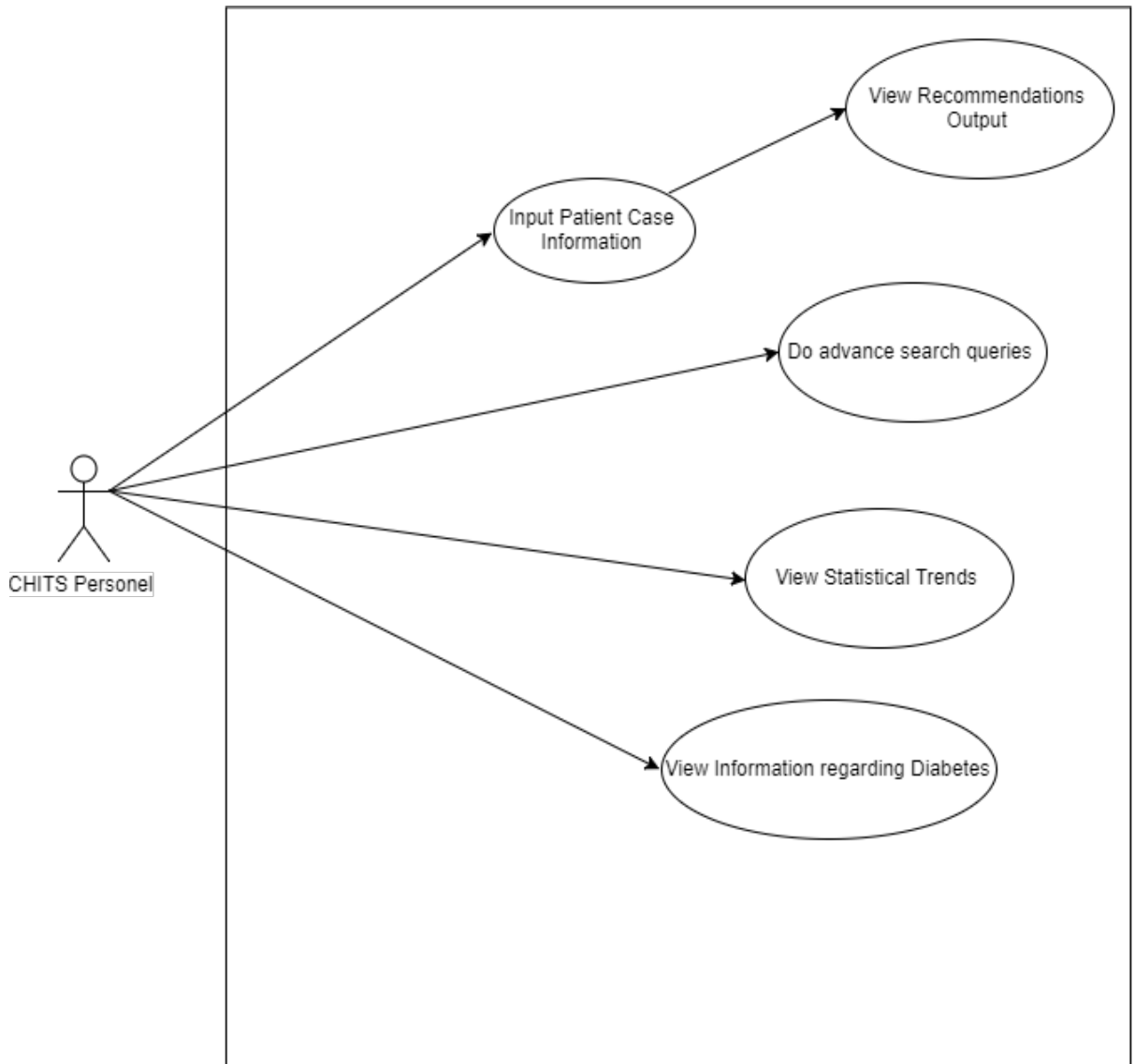


Figure 2: Use Case Diagram for Health Professional/Researcher

### 3. Building the Recommendation Model

- (a) Select Diabetes cases from CHITS dataset
- (b) Get all records(family history, social history, treatment, medication, lab reports, etc) of the patients identified as diabetic patients.
- (c) Save data to Diabetes to Diabetes Health Recommender System Database as a CSV file.
- (d) Load CSV to Pandas and load the data as a dataframe. This will allow to insert the vectors into the cosine similarity algorithm easier.
- (e) Make a new dataframe by dropping the following columns: person\_id, weight, height, encounter\_id, birthdate, CN, HN, DN, DIAG\_DM, DIAG\_NONDM, TPN, and gender as these parameters are not needed for computation
- (f) Make a separate dataframe using the values inputted from the system. The system automatically computes the BMI of the inputted data then uses it as a parameter in the separate dataframe as BMI is needed for the Finnish Risk Score Assessment.  
  
Both the recommender and the separate dataframe(inputted data) must contain data for waist circumference,hip circumference, age, gender, Hba1c result, fbs result, and BMI.
- (g) Apply Finnish Risk Score Assessment as weights to newly created dataframe and recommender dataframe. The scores can be found in [\[38\]](#)
- (h) Append recommender dataframe with the newly created dataframe
- (i) Normalize Dataframe by using the pre-processing package found in sklearn library.
  - i. Get all the values from the newly created dataframe and store it inside a list variable
  - ii. get a min-max object from pre-processing package using MinMaxScaler() function.

- iii. use the function of the min-max object, `fit_transform(values)` to normalize the values
- iv. create a new recommender dataframe with the normalized values
- (j) Get last row of the recommender dataframe and remove last row from the recommender dataframe
- (k) Compute for each row of the recommender dataframe the Cosine Similarity(from `sklearn.metrics.pairwise`) with the acquired dataframe.

The computation for cosine similarity is as follows

- i. Cosine Similarity takes two numerical vectors A and B. Each vector containing data for Waist Circumference, Hip Circumference, Age, Gender, Hba1c Result, FBS result and BMI
  - ii. Vector A contains the data of each parameter for one record in the dataset. Vector B contains the inputted data from the system. Both vectors must contain the same number of parameters, otherwise the algorithm wont work.
  - iii. the algorithm computes the angle between these two vectors by computing the dot product of the vectors divided by the magnitude of these two vectors
- (l) Store each result in a 2 dimension List in a format of `[index,result]`
  - (m) Sort the List according to result
  - (n) Return the top 5 results of the list

4. Minimum System Architecture:

- (a) at least 4 GB RAM
- (b) AMD A10/Intel I3 or better
- (c) Windows/Linux/MacOS
- (d) Python 2.7

## V. Results

Below is the user interface of the Diabetes Recommender System which includes the home page, treatment page, search page, and dashboard page.

The main functionality of the system is the treatment page where the health professional can input the patient profile and get the recommended treatment in the results page.

Researchers can also search for patient profiles in the search component of the system.

The data analytics component of the system is presented through a dashboard and shown using graphs.



Figure 3: Homepage of the System

This is the homepage of the system which includes what the system is all about and includes facts about diabetes in the Philippines.



The screenshot shows the CHITS Patient Profile Form. On the left is a navigation menu with options: HOME, FAQ, TREATMENT, SEARCH, STATISTICS, LINKS, and CHITS. The main form area is titled 'Patient Information' and contains the following fields: Weight(kg), Height(cm), Waist Circumference(cm), Hip Circumference(cm), Age, and Sex (Male/Female). Below these is a 'LAB RESULTS' section with a red warning box that says 'Default value is 0.00 if result is unavailable'. The lab results show HbA1C(%) and Fasting Blood Sugar(mg/dL) both at 0.00. On the right side, there is a sidebar with a blue header that says 'Information: Click to view information for each data.' and a list of parameters: Weight, Height, Waist and Hip Circumference, Age, Gender, HbA1C, and FBS, each with an information icon.

Figure 4: Patient Profile Form

fig 4 is the form in which the health professional must input the patient's profile then clicking the submit button in order to get the recommended treatments. The page also includes facts about each parameter and explains how it contributes as a diabetes factor.

The parameters included in the form are

1. Weight
2. Height
3. Waist Circumference
4. Hip Circumference
5. Age
6. Gender
7. Hba1C Lab Result
8. FBS lab Result

To find similar patients, the system compares patient profiles using the Cosine Similarity algorithm. The system also uses the Finnish Risk Score Assessment to add weights to the parameters before being computed.

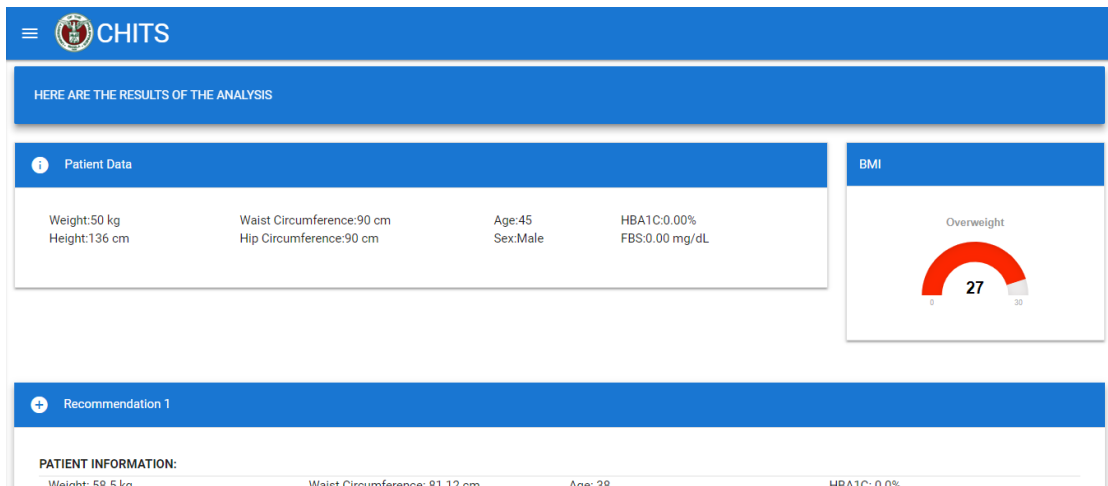


Figure 5: Recommender System Results

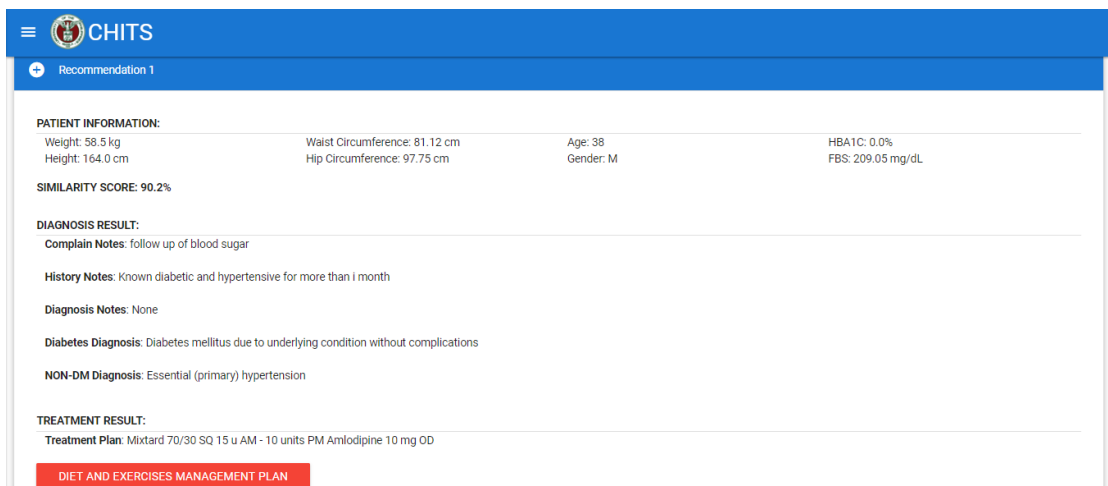


Figure 6: Recommender System Results

fig 5 and fig 6 shows the recommended treatment after the system finds the similar patients to the new patient. The page includes the information inputted by the health professional as well as the gauge of the body mass index of the patient. The system shows the top 5 recommendation and displays all information of that patient including profile, diagnosis notes, and treatments.

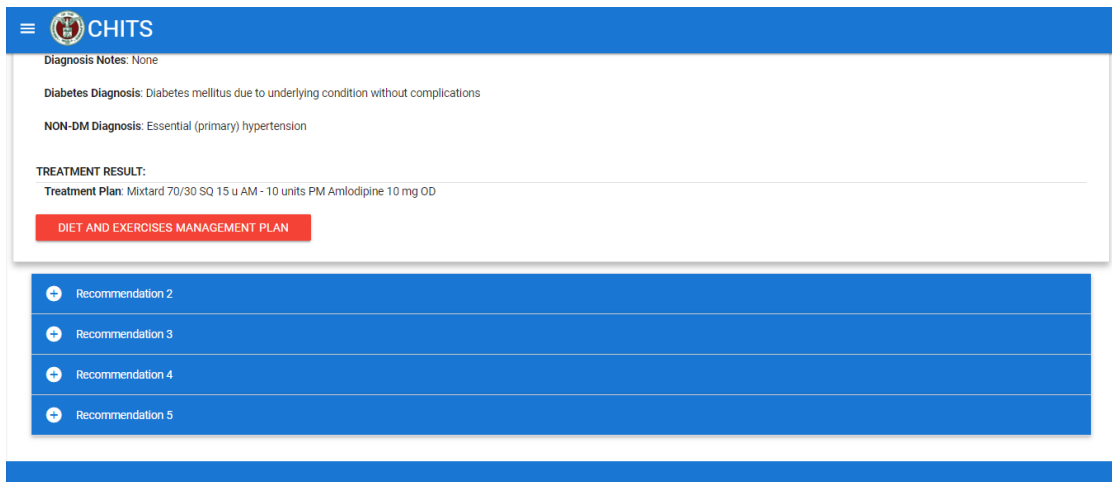


Figure 7: Top 5 Recommendations

The other recommendations can be clicked upon by the user (same as the first recommendation) to view other patient profiles and treatments.

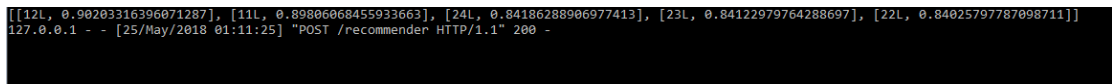


Figure 8: Score Results

fig 8 show the following score results from the algorithm. The scores shown in the command prompt are the top five similar patients. the results show which row the patient record is in the dataset followed by the similarity score

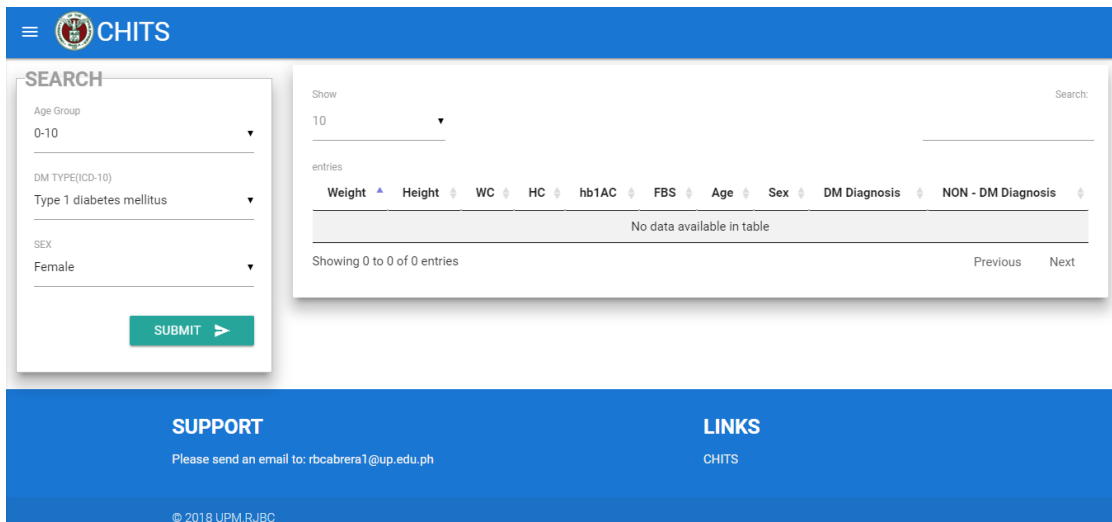


Figure 9: Search page with Datatable

Users can search for patient profiles here by querying their sex, DM type(based on ICD10) and age group. A datatable is included to enable users to sort any parameter or search any text query. Users may also limit the search results by the select field on the upper left side of the data table

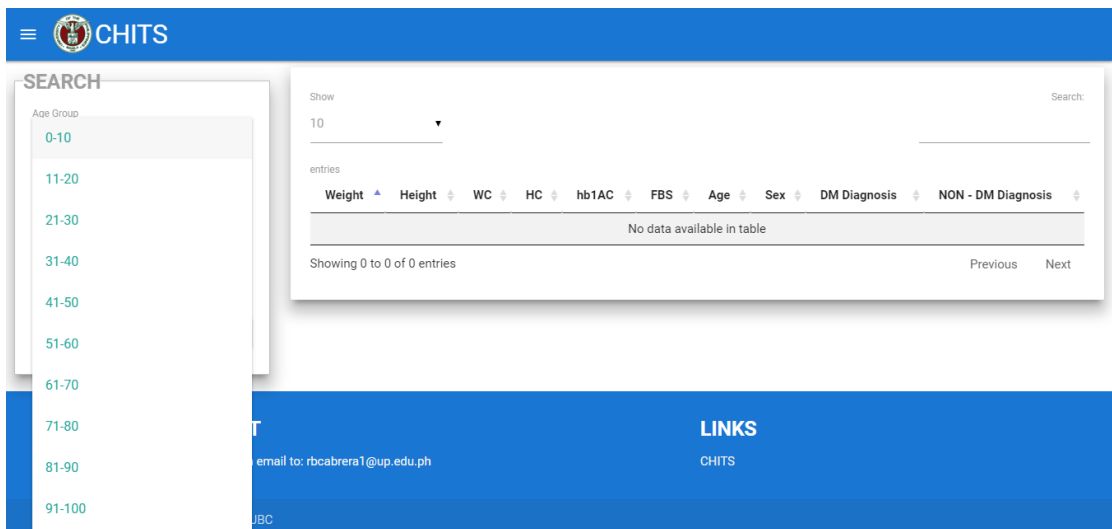


Figure 10: Select Field: Age Group

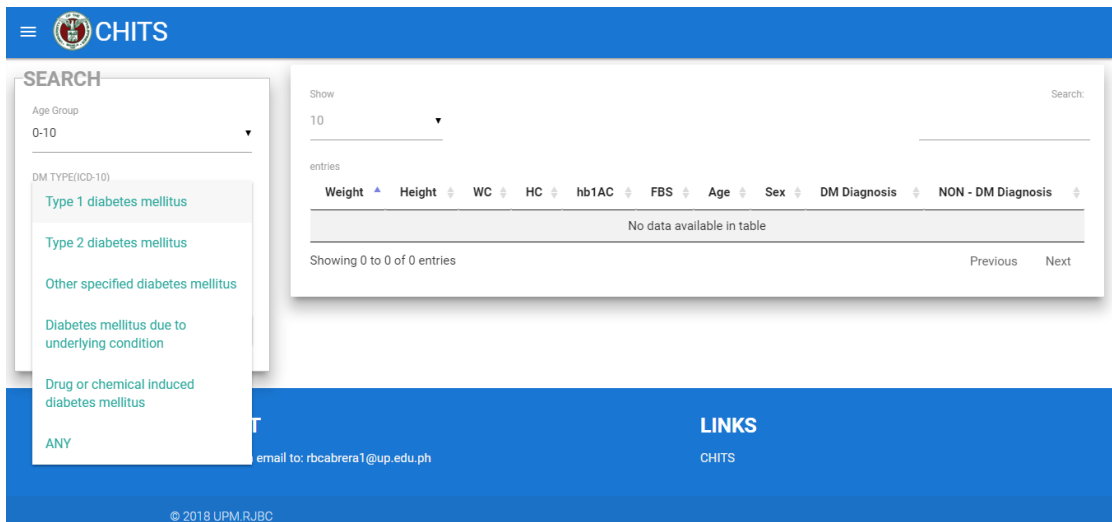


Figure 11: Select Field: DM Type

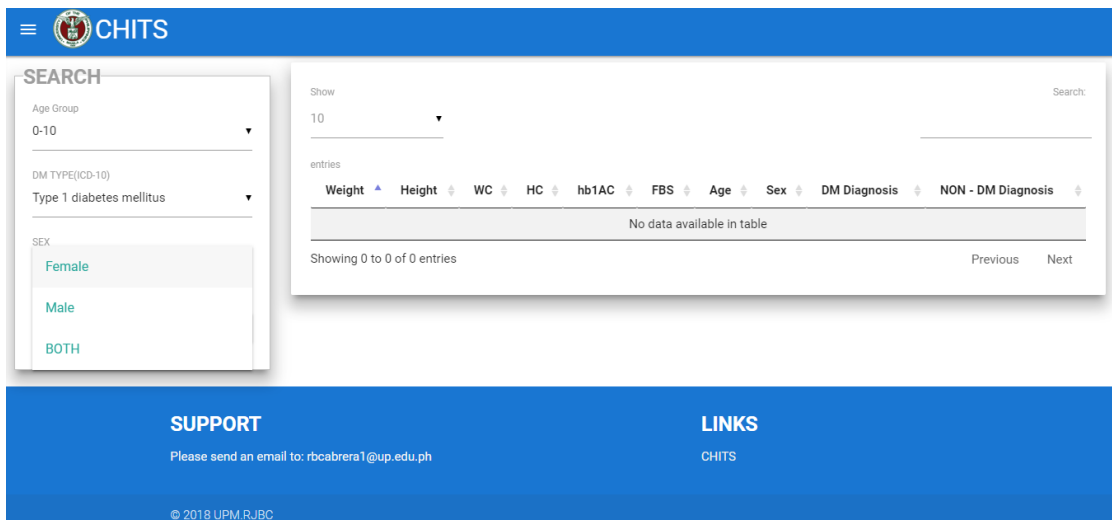


Figure 12: Select Field: Sex

In fig 10, 11 and 12, the user can choose wit parameters he/she is interested in. After selecting which options in the parameters to search, clicking the submit button will show the search results in the datatable.

**SEARCH**

Age Group  
51-60

DM TYPE(ICD-10)  
ANY

SEX  
BOTH

SUBMIT

Show 10 entries

Weight	Height	WC	HC	hb1AC	FBS	Age	Sex	DM Diagnosis	NON - DM Diagnosis
45.1	144.0	88.8	120.8	9.68%	86.51	52	F	Type II diabetes mellitus, non-insulin-requiring	None
52.6	150.0	86.21	100.04	9.56%	96.3	58	F	Type II diabetes mellitus, non-insulin-requiring	HYPERTENSION, HPN (ESSENTIAL HYPERTENSION) (MALIGNANT HYPERTENSION)
58.5	157.0	91.0	91.0	0.0%	195.4	53	F	Type II diabetes mellitus, non-insulin-requiring	None
61.0	152.0	100.97	97.28	9.28%	149.57	54	F	Type II diabetes mellitus, non-insulin-requiring	None

Figure 13: Datatable Results

The datatable shows the result of the queries. The data can be sorted by clicking on the column header. This can be sorted in descending or ascending order.

≡ **CHITS**

52.0	125.0	83.04	90.01	10.28%	143.0	51	F	to underlying condition with hyperglycemia	None
43.0	163.0	78.33	87.17	6.72%	144.59	51	F	Diabetes mellitus due to underlying condition with diabetic cataract	Other and unspecified genitourinary tract infection in pregnancy
45.0	133.0	77.0	86.0	6.68%	111.6	51	F	Diabetes mellitus due to underlying condition with diabetic cataract	None

Showing 1 to 10 of 118 entries Previous 1 2 3 4 5 ... 12 Next

**SUPPORT**

Please send an email to: [rbcabrera1@up.edu.ph](mailto:rbcabrera1@up.edu.ph)

**LINKS**

[CHITS](#)

© 2018 UPM,RJBC

Figure 14: Datatable Results when using search

The datatable automatically limits the results while the user types on the search field.

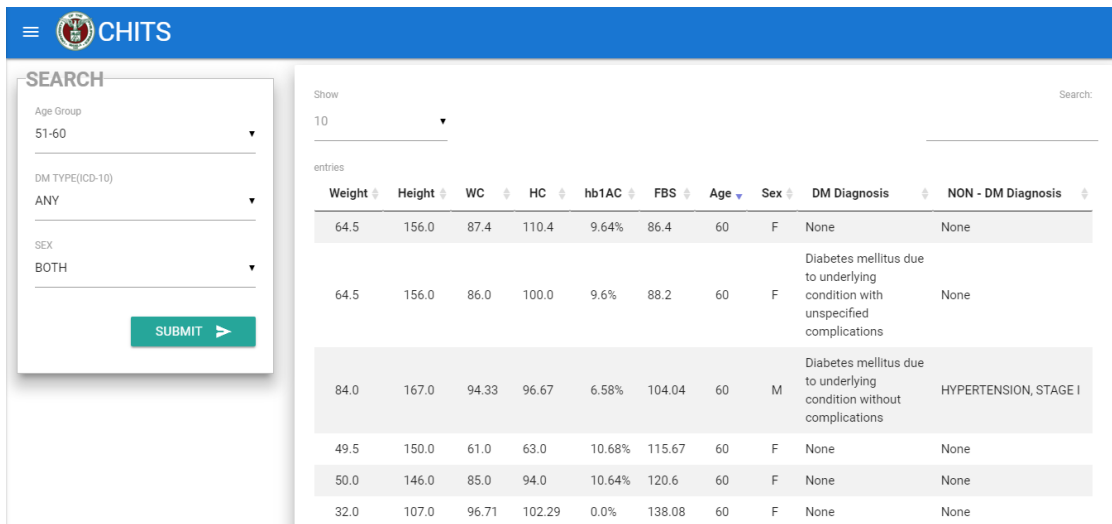


Figure 15: Datatable Results with pagination

The datatable automatically adds a pagination when the results of the search query exceeds the number of results to be shown on the table.

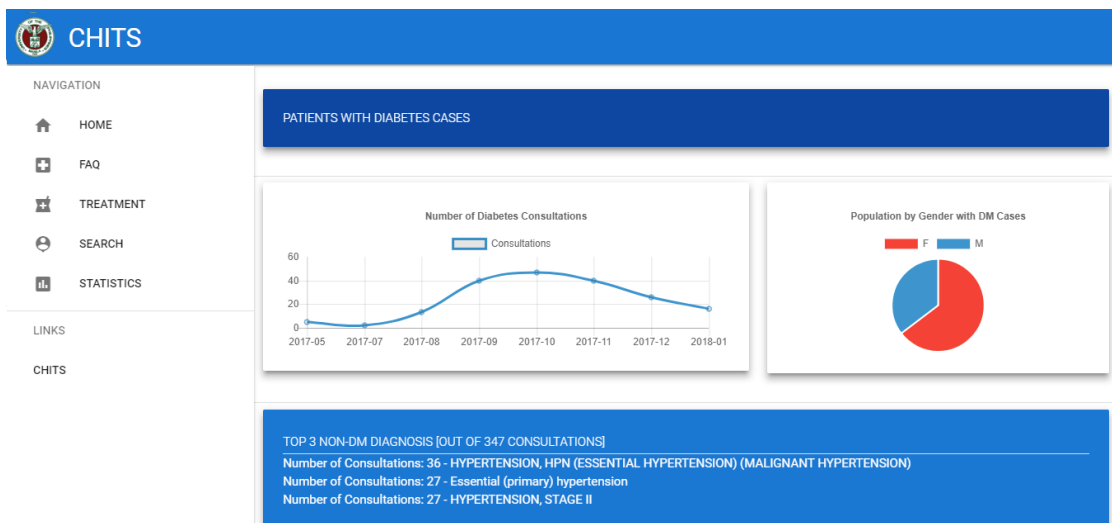


Figure 16: Dashboard Results



Figure 17: Dashboard Results

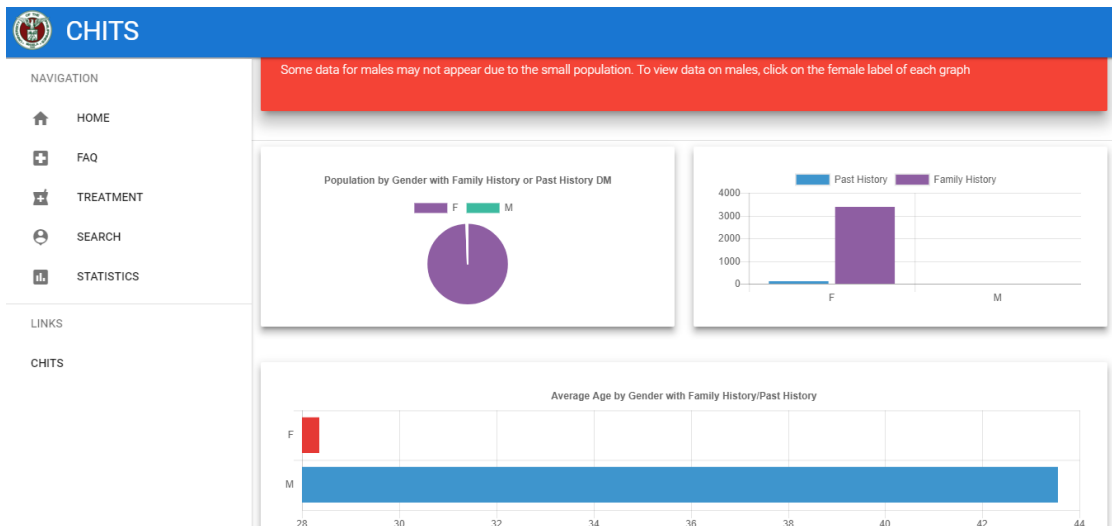


Figure 18: Dashboard Results

The dashboard shows different statistics on the diabetes data. Some graphs show the number of cases per diagnosis, top recorded FBS result, specific age groups of patients, the number of cases per month, the number of patients with past history or family history.



## VI. Discussions

The Diabetes Recommendation System is an web application that gives treatment recommendations to the diabetes cases. The system includes a recommendation component and a data analytics component through the use of a dashboard and a data table.

The advantages of the system is it gives out complete information(except for patient's sensitive data) that maybe useful to the health professional such as the patient profile and history. This gives the health professional the complete idea of why a certain treatment is given as compared by just giving the diabetes diagnosis and treatment. This is due to some diseases being related to diabetes.

The system also views the patient profile in the search component and it allows the users to sort the data which is a great component for data analysis as it allows the user to understand the patient profile easily.

Another advantage of the system is that the system utilizes the data collection module of CHITS by presenting charts that tell the conditions of the patients which will allow health organizations or health care stations to further improve their health care modules/systems.

The disadvantages of the system is that it may give out results that are not the best treatment in some cases. This is due because some patients are not diagnosed with diabetes during that encounter period therefore, some treatments may include also non diabetes treatments. There are also some cases in which the better treatment(by a health professional) is not the first recommendation by the system. This is because the similarity is based on the patient profile and the treatment is not included in the similarity computation. There are also cases wherein the the diabetes diagnosis is not indicated but a treatment for diabetes is included. This is due to a data entry problem involved in CHITS.

In comparison to other recommendation system, the system uses cosine similarity to find similar patients while other recommendation systems used other sim-

ilarity metrics such as Pearson Correlation. Most of the data needed for the similarity computation requires numerical variables while other systems used mostly categorical variables.

A lot of problems were encountered during the development of the system. The first problem I encountered was the system's core functionality was supposed to be built using a big data framework, Apache Spark but issues came up when building the python script as I was unable to run it as a python script but was able to successfully run the code in a Jupyter notebook. There is an instruction on how to run the code as a python script but was still unable to successfully run it.

While testing the algorithm, some potential problems can be noticed. These are the following:

1. The algorithm computes similar columns therefore requiring the dataset to be transposed first.
2. the dataset must be transformed into a Coordinate Matrix to be transposed which leads to more confusion if it actually returns the right result when DIMSUM is applied.
3. The algorithm computes all similarities between columns which is not needed for the system therefore wasting computational time and effort.

The solution to this problem was to use a python library called scikit-learn which contains an implementation of cosine similarity which is also the basis of DIMSUM. the function also solves the problem presented with DIMSUM as it enables the computation to be done by row, data was not needed to be transformed into another object, and reducing the time and effort for computing the needed result from the dataset.

Another problem encountered in developing the system was the data extraction from CHITS. Studying the system was not an easy task as it was designed in a

different way. CHITS was designed to present their records by column which means for example, I have two rows with a record of the same person, same encounter id but different concept id with means that I have to build the dataset to present the records by row. The main obstacle in data extraction was a data entry problem as the fields needed for the system is either null or not formatted in a proper way. Examples are some patient entry does not have a record of the height or weight, most patient do not have any past or family history recorded, food intake and physical activity was not recorded, lab results such as FBS was not recorded in the proper field but rather was saved in a different field. Another problem was finding the diabetic patient themselves since the dataset is dependent on their records.

To strategy to build the recommender system data set was to find the treatments that are specifically for diabetes such as Metformin. Finding the diagnosis is not that helpful since not all diagnosis was given a treatment therefore will become useless for the recommendation system. After finding the those treatments, you get the patient profile needed(weight, height, waist circumference,gender,etc) and save it as a CSV file. I also saved a another csv file which contains the distinct encounter id and person id of that record which will be used later for pre-processing. This process is done to different health stations to obtain the different datasets which will become combined as a single recommender data set.

To build per health station dataset, a python script was built to make a dataframe which would present the data by row as mentioned before. The strategy here was by using pandas library, loading the two datasets gathered from one health stations and using the distinct encounter id and person id with the dataset containing their information and looping to properly place the data in their respective columns to that specific person during that encounter.

Once the different datasets was built, another python script was built to append all the datasets into one csv file. I tested the datasets for potential errors and the results were not very convincing since almost all patients were similar

with the minimum result 0.7 which means they are likely similar but with not convincing explanation. the problem was due to the high value of the weight and height parameters were covering the other parameters needed so the algorithm was mostly relying on those two parameters for the result and was "disregarding" the others. To solve this issue, normalizing the dataset before it was being used in the system which will level all the parameters equally. This was done using sklearn preprocessing. The values of the data was ranging from 0 - 1 therefore allowing the algorithm to function properly and including the other parameters.

as mentioned before, data entry was the main obstacle to the recommender as the system relies on the dataset that it uses as basis. the solution was to apply linear interpolation to fill up the null values of numeric parameters. Pandas as a function to apply it a dataframe and it was able to at least fill up most of the null values with estimated values in order to make the system more functional.

Despite the problems stated above, the system was successfully built and is able to give out recommendations that are properly computed and presented by giving out the top 5 recommendations.

## VII. Conclusions

The Diabetes Recommendation System is a web application that allows health professionals to get treatment recommendations which could be a useful tool to help the doctor analyze the profile of the patient and help in deciding the proper treatment to improve the patient's condition. A successful treatment does not only take care of the patient's well being but also reducing additional costs to both the patient and the hospital.

Researchers can also use the system for data analysis that enables to learn more about the condition of diabetic Filipinos and the current trends in diabetes cases. This will allow them to improve their health programs that they provide therefore also improving the healthcare quality of the patients.

The system was developed by gathering data from CHITS, creating python scripts to build the dataset, building the user-interface of the system, pre-processing the data for the system, and implementing the cosine similarity algorithm and other processes of the system.

To gather data from CHITS, studying the how CHITS works was the first step then after understanding the core of the CHITS system, different data mining techniques were applied. Text mining was applied to get lab results and treatments needed for the system. A python script was also built that made creating the dataset faster.

Building the user-interface of the system, The Materializecss framework made it as user-friendly and as responsive as possible. The charts are properly organized to properly tell the story of data, what it means and how valuable it could be to a researcher. Doctors can also input data using the forms provided by the system and view the top 5 recommendations containing the information of the patient profile including history notes, non-diabetes diagnosis, diabetes diagnosis and the provided treatment.

Pre-processing the data allows the system to calculate how similar the patients

are which is the most crucial component of the system as it is depend on the result of the similarity metric to give out the recommendation.

With the potential of CHITS as an electronic medical record system, the Diabetes Recommendation System, can be a potential healthcare application that will allow improving quality health care in the Philippines through data analytics and treatment recommendations.

## VIII. Recommendations

### 1. Dataset

To improve the performance of the system, it is important to continue adding more data to the data set by collecting more datasets from different health centers that uses CHITS as the recommendations is dependent on how many records exist the data set. The recommendations stated below are further notes on how make the dataset more useful for the system.

Currently, the data set contains only 7 parameters to be used by the algorithm as the some factors related to diabetes cannot be mined. A possible way to improve the system is to provide more parameters that the can be implemented in the algorithm. Categorical variables such as physical activity or alcohol drinking frequency. Numerical variables such as lab results could further improve finding similar patients.

The data set was also interpolated due to many null values in important fields for computation. To further improve the calculation for similarity, the data as much as possible must not contain null values for fields which are required by the algorithm.

The data set only contains medical treatments mined from CHITS, to further improve the recommendation the system provides, the data set must also contain non-medical treatments such as diet plans or exercise regimens provided by a health professional

A crucial part of how to improve the dataset is to have a health professional review the dataset or better, review the patient's history to show that the treatment provided by the previous doctor is a good treatment therefore eliminating any treatment that are not suitable by an expert's judgment. In this manner, the system will be more reliable.

To enhance the performance of finding similar patients, other similarity metrics could be implemented in the system.

## 2. Web Application

If the system can be hosted in a server hosted through HTTPS, it can be used as a desktop or mobile application by implementing a service worker to transform the web application to a progressive web application. This allows the user to install the application in their device(mobile,tablet,desktop) instead of accessing the system through a browser.

The data table in the search component of the system could not be properly viewed in a mobile platform. Some tweaks are required in order to arrange the way it presents data without losing it's functionality

The data analysis component of the system could also be improved by adding more statistical information that can be found in the dataset such as correlation of diabetes with non-diabetes diagnosis(ex. Hypertension).

The system could also improve the recommendation quality if it includes a data collection component that allows the doctor to confirm if the recommended treatment improved the condition of the patient. This allows removing treatments that are not effective. The system can be greatly improved and more reliable if this step is followed.



## IX. Bibliography

- [1] WHO, “Diabetes mellitus.” <http://www.who.int/mediacentre/factsheets/fs138/en/>.
- [2] NIDDK, “Symptoms causes of diabetes..” <https://www.niddk.nih.gov/health-information/diabetes/overview/symptoms-causes>.
- [3] “General diabetes facts and information.” [http://www.joslin.org/info/general\\_diabetes\\_facts\\_and\\_information.html](http://www.joslin.org/info/general_diabetes_facts_and_information.html).
- [4] imoney, “10 fatal diseases in the philippines you should be aware of.” <http://www.imoney.ph/articles/10-fatal-diseases-philippines/>.
- [5] “6m pinoys have diabetes.” <http://newsinfo.inquirer.net/805812/6m-pinoys-have-diabetes>.
- [6] “Philippines seen as diabetes hotspot’.” <http://www.sunstar.com.ph/manila/local-news/2016/03/23/philippines-seen-diabetes-hotspot-464154>.
- [7] K. S. Cabrera, “Filipinos eating patterns blamed for diabetes’.” <http://www.thestandard.com.ph/lifestyle/155855/filipinos-eating-patterns-blamed-for-diabetes.html>.
- [8] T. Joe, “The problem of diabetes in the philippines.” <http://diabeticfilipino.com/the-problem-of-diabetes-in-the-philippines/>.
- [9] “Chits.” <https://telehealth.ph/project-chits/>.
- [10] U. Digital and U. M. Technology WorldTagged Community Health Information Tracking System, “Chits: Community health information tracking system.” <http://www.glamourmoments.net/chits-community-healthh-information-tracking-system//>.

- [11] J. Archenaa and E. Anita, "Health recommender system using big data analytics," *Journal of Management Science and Business Intelligence*, 2017.
- [12] S. Prasad, "Applying big data analytics to improve patient-centric care." <http://www.mtbeurope.info/content/ft1501001.htm>.
- [13] M. Wiesner and D. Pfeifer, "Health recommender systemsl concepts, requirements, technical basics and challenges," 2014.
- [14] P. Nitesh V. Chawla and P. Darcy A. Davis, "Bringing big data to personalized healthcare: A patient-centered framework," *J Gen Intern Med*, p. 660665, 2013.
- [15] S. V. Chaitanya Kaul, Ashmin Kaul, "Comparitive study on healthcare prediction systems using big data," 2015.
- [16] J. A. D. Joon Lee, David M. Maslove, "Personalized mortality prediction driven by electronic medical data and a patient similarity metric," 2015.
- [17] P. A. V. D. Tofik R. Kacchi, "Recommender system based on lifestyle of users," *International Journal of Advanced Research in Computer Engineering Technology (IJARCET) Volume 5, Issue 5*, 2016.
- [18] D. K. J. S. S. A. H. M. a. S. Z. Felix Grer, Stefanie Beckert, "Therapy decision support based on recommender system methods," *Journal of Healthcare Engineering Volume 2017 (2017), Article ID 8659460*, 2017.
- [19] G. B. T. C. S. C. N. M. Dario Antonelli, Elena Baralis, "Analysis of diabetic patients through their examination history," *Expert Systems with Applications 40 (2013)*, 2013.
- [20] M. B. W. O. M. L. N. R. V. A. O. Bruno bia, Marcia Sampaio Lima, "An approach for course recomendation applied to continuous education on health area," 2017.

- [21] M. R. P. Swarupananda Bissoyi, Brojo Kishore Mishra, "Recommender systems in a patient centric social network a survey," *International conference on Signal Processing, Communication, Power and Embedded System (SCOPE)-2016*, 2016.
- [22] S. B. S. S. P. Sanket Bankhele, Ashish Mhaske, "A diabetic healthcare recommendation system," *International Journal of Computer Applications*, vol. 167, 2017.
- [23] P. J. S. R. P. M. P. Yan Luo, Charles Ling, "Glucoguide: An intelligent type-2 diabetes solution using data mining and mobile computing," *Data Mining in Biomedical Informatics and Healthcare*, 2014.
- [24] V. V. V. and A. C., "Prediction and diagnosis of diabetes mellitus -a machine learning approach," 2015.
- [25] M. Melissa Conrad Stoppler, "Gestational diabetes." [https://www.medicinenet.com/gestational\\_diabetes/article.htm](https://www.medicinenet.com/gestational_diabetes/article.htm).
- [26] Healthline, "How diabetes affects women: Symptoms, risks, and more." <https://www.healthline.com/health/diabetes/symptoms-in-women#symptoms>.
- [27] "What increases my risk of diabetes?." <https://www.webmd.com/diabetes/guide/risk-factors-for-diabetes#1>.
- [28] M. C. Staff, "Diabetes diet: Create your healthy-eating plan." <https://www.mayoclinic.org/diseases-conditions/diabetes/in-depth/diabetes-diet/art-20044295/>.
- [29] J. Leonard, "Managing diabetes with physical activity and exercise." <https://www.medicalnewstoday.com/info/diabetes/exercise.php>.
- [30] R. G. F. P. G. S. A. J. P. A. M.-I. R.-S. A. D. F. S. J. G. F. Arturo M. Ongkeko, Jr. and P. H. Fernandez-Marcelo, "Community health information and track-

ing system (chits): Lessons from eight years implementation of a pioneer electronic medical record system in the philippines,” 2016.

- [31] A. G. Shoro and T. R. Soomro, “Big data analysis: Ap spark perspective,” *Global Journal of Computer Science and Technology: C Software Data Engineering*, 2015.
- [32] D. A. Sathi, *Big Data Analytics: Disruptive Technologies for Changing the Game*. 3695 W. Quail Heights Court Boise: MC Proess Online, LLC, 2012.
- [33] A. DeVan. <https://www.impactradius.com/blog/7-vs-big-data/>.
- [34] D. K. J. S. S. A. H. M. Felix Grer, Stefanie Beckert and S. Zaunseder, “Therapy decision support based on recommender system methods,” *Journal of Healthcare Engineering*, vol. 2017, 2017.
- [35] “What is python? executive summary.” <https://www.python.org/doc/essays/blurb/>.
- [36] “Pandas data analysis library.” <http://pandas.pydata.org/>.
- [37] “scikit-learn machine learning in python.” <http://scikit-learn.org/stable/index.html>.
- [38] H. A. Mohsen Janghorbani and M. Amini, “Evaluation of the finnish diabetes risk score (findrisc) as a screening tool for the metabolic syndrome,” 2013.

# X. Appendix

## A Source Code

```
app.py

from flask import Flask, render_template, request, flash
from forms import CaseForm
from search_forms import SearchForm

# how to import external .py file
from diagnosis_dashboard import get_ICD10_list,
    get_dm_cases_counts_mmyy, get_dm_cases_by_gender_count,
    get_dm_cases_counts, get_age_group,
    get_dm_cases_average_age_gender, \
get_history_cases_by_gender_count, get_average_age_gender,
    get_age_group_past_fam_history,
    get_count_ph_fh_by_gender, \
get_ICD10_type2, get_ICD10_type1,
    get_ICD10_dm_underlying_cond, get_ICD10_otherDM,
    get_ICD10_Drug_chemical_induced, \
get_ICD10_type2_withgender, get_ICD10_type1_withgender,
    get_ICD10_dm_underlying_cond_withgender,
    get_ICD10_otherDM_withgender,
    get_ICD10_Drug_chemical_induced_withgender

# from cs_pandas import calculate_CS, get_data, get_treatments
from recommend import calculate_CS, get_data,
    get_max_FBS_value, get_max_HBA1C_value,
    get_nonDM_count, get_bmi_label, get_data_df_count,
    get_max_fbs_value_column, get_max_hba1c_value_column
from query import get_search_data

app = Flask(__name__)
app.config['SECRET_KEY'] = 'reuben158'

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/faq')
def faq():
    return render_template('faq.html')

@app.route('/recommender', methods = ['POST', 'GET'])
def recommender():
    form = CaseForm()

    if form.validate_on_submit():
        bmi = (form.weight.data / (form.height.data / 100) ** 2)
        bmi = str(round(bmi, 2))
        bmi_res = get_bmi_label(bmi)

        # res = calculate_CS(form.weight.data, form.height.data,
        # form.wc.data, form.hc.data, form.sex.data, form.age.
        # data, form.hba1c.data, form.fbs.data)
        res = calculate_CS(bmi, form.wc.data, form.hc.data, form.
            sex.data, form.age.data, form.hba1c.data, form.fbs.
            data)
        print res
        list_treatments = get_data(i[0] for i in res)
        for i in res:
            i[1] = round(i[1], 4) * 100

        if form.sex.data == '2':
            sex_label = 'Female'
        else:
            sex_label = 'Male'

        return render_template('result.html', list_treatments =
            list_treatments, form = form, sex_label = sex_label,
            bmi = bmi, bmi_res = bmi_res, res = res)

    return render_template('recommender.html', form = form)

@app.route('/search', methods = ['POST', 'GET'])
def search():
    s_form = SearchForm()

    if s_form.validate_on_submit():
        search_df = get_search_data(s_form.search_sex.data, s_form.
            search_dm_type.data, s_form.search_age_group.data)

        return render_template('search.html', s_form = s_form,
            search_df = search_df)

    return render_template('search.html', s_form = s_form)

@app.route('/statistics')
def dashboard():
    # send value to template
    cases_gender = get_dm_cases_by_gender_count()
    dm_cases_count = get_dm_cases_counts()
    dm_cases_year_month = get_dm_cases_counts_mmyy()
    # print cases_gender
    dm_cases_ave_age = get_dm_cases_average_age_gender()
    dm_cases_age_grp = get_age_group()
    dm_cases_specific_count = get_ICD10_list()

    dm_history_gender = get_history_cases_by_gender_count()
    dm_history_gender_age_ave = get_average_age_gender()
    dm_history_gender_age_grp = get_age_group_past_fam_history
    ()
    dm_history_gender_count_by_gender =
        get_count_ph_fh_by_gender()

    fbs_max_value = get_max_FBS_value()
    HBA1C_max_value = get_max_HBA1C_value()
    non_dm_max_value = get_nonDM_count()

    ICD10_type2 = get_ICD10_type2()
    ICD10_type1 = get_ICD10_type1()
    ICD10_underlying_cond = get_ICD10_dm_underlying_cond()
    ICD10_otherDM = get_ICD10_otherDM()
    ICD10_Drug_Chemical_induced =
        get_ICD10_Drug_chemical_induced()

    ICD10_type2_withgender = get_ICD10_type2_withgender()
    ICD10_type1_withgender = get_ICD10_type1_withgender()
    ICD10_dm_underlying_cond_withgender =
        get_ICD10_dm_underlying_cond_withgender()
    ICD10_otherDM_withgender =
        get_ICD10_otherDM_withgender()
    ICD10_Drug_chemical_induced_withgender =
        get_ICD10_Drug_chemical_induced_withgender()

    dataset_count = get_data_df_count()
    highest_fbs_profile = get_max_fbs_value_column()
    highest_hba1c_profile = get_max_hba1c_value_column()

    return render_template('dashboard.html',
        dm_cases_specific_count = dm_cases_specific_count,
        dm_cases_year_month = dm_cases_year_month,
        cases_gender = cases_gender,
        dm_cases_count = dm_cases_count,
        dm_cases_ave_age = dm_cases_ave_age,
        dm_cases_age_grp = dm_cases_age_grp,
        dm_history_gender = dm_history_gender,
        dm_history_gender_age_ave = dm_history_gender_age_ave,
        dm_history_gender_age_grp = dm_history_gender_age_grp,
        dm_history_gender_count_by_gender =
            dm_history_gender_count_by_gender,
        fbs_max_value = fbs_max_value,
        HBA1C_max_value = HBA1C_max_value,
        non_dm_max_value = non_dm_max_value,
        ICD10_type2 = ICD10_type2,
        ICD10_type1 = ICD10_type1,
        ICD10_underlying_cond = ICD10_underlying_cond,
        ICD10_otherDM = ICD10_otherDM,
        ICD10_Drug_Chemical_induced =
            ICD10_Drug_Chemical_induced,
        ICD10_type2_withgender = ICD10_type2_withgender,
        ICD10_type1_withgender = ICD10_type1_withgender,
        ICD10_dm_underlying_cond_withgender =
            ICD10_dm_underlying_cond_withgender,
        ICD10_otherDM_withgender = ICD10_otherDM_withgender,
        ICD10_Drug_chemical_induced_withgender =
            ICD10_Drug_chemical_induced_withgender,
        dataset_count = dataset_count,
        highest_fbs_profile = highest_fbs_profile,
        highest_hba1c_profile = highest_hba1c_profile)

if __name__ == '__main__':
    app.run(debug=True)
```

```

diagnosis_dashboard.py

import pandas as pd
import glob
import datetime as dt

def create_dataframe(path_name):
    path = r'DS/' + path_name # use your path
    allFiles = glob.glob(path + "/*.csv")
    frame = pd.DataFrame()
    list_ = []
    for file_ in allFiles:
        df = pd.read_csv(file_, index_col=None, header=0)
        list_.append(df)
    frame = pd.concat(list_)
    return frame

frame = create_dataframe('Diabetes_Cases')

def change_time_format(df):
    df['date_created'] = pd.to_datetime(df['date_created']).dt
        .strftime("%Y-%m")

def get_age(df):
    df['birthdate'] = pd.to_datetime(df['birthdate'])
    now = dt.date.today()
    df['age'] = ((now - df['birthdate'])/365).dt.days

change_time_format(frame)
get_age(frame)

#calculate age group https://stackoverflow.com/questions
#35550486/data-grouping-in-pandas
frame['count'] = 1
frame['age_group'] = pd.cut(frame.age,
    [0,10,20,30,40,50,60,70,80,90,100])
age_grp = frame.pivot_table('count', index='age_group',
    columns='gender', aggfunc='sum').fillna(0)

#shape[0] = row count
ICD10_count_list = []
ICD10_count_list.append(['Type 2 diabetes mellitus', frame[
    frame['value_text'].str.match('Type 2') | frame[
    'value_text'].str.match('Type II diabetes')][
    'value_text'].shape[0])
ICD10_count_list.append(['Type 1 diabetes mellitus', frame[
    frame['value_text'].str.match('Type 1') | frame[
    'value_text'].str.match('Type I diabetes')][
    'value_text'].shape[0])
ICD10_count_list.append(['Diabetes mellitus due to underlying
    condition', frame[frame['value_text'].str.match('
    Diabetes mellitus')][
    'value_text'].shape[0])
ICD10_count_list.append(['Other specified diabetes mellitus',
    frame[frame['value_text'].str.match('Other specified
    diabetes mellitus')][
    'value_text'].shape[0])
ICD10_count_list.append(['Drug or chemical induced diabetes
    mellitus', frame[frame['value_text'].str.match('Drug or
    chemical induced diabetes mellitus')][
    'value_text'].shape[0])

#get Functions here
def get_ICD10_list():
    return ICD10_count_list

def get_ICD10_type2():
    return frame[frame['value_text'].str.match('Type 2') |
        frame['value_text'].str.match('Type II diabetes')].
        groupby('value_text')['value_text'].count()

def get_ICD10_type1():
    return frame[frame['value_text'].str.match('Type 1') |
        frame['value_text'].str.match('Type I diabetes')].
        groupby('value_text')['value_text'].count()

def get_ICD10_dm_underlying_cond():
    return frame[frame['value_text'].str.match('Diabetes
        mellitus')].groupby('value_text')['value_text'].count()

def get_ICD10_otherDM():
    return frame[frame['value_text'].str.match('Other specified
        diabetes mellitus')].groupby('value_text')['
        value_text'].count()

def get_ICD10_Drug_chemical_induced():
    return frame[frame['value_text'].str.match('Drug or
        chemical induced diabetes mellitus')].groupby('
        value_text')['value_text'].count()

def get_dm_cases_counts_mmyy():
    v = frame['date_created'].value_counts().sort_index()
    v_2017 = v[v.index.str[:4].astype(int) >= 2017]
    return v_2017

def get_dm_cases_counts():
    return frame.groupby(['value_text'])['value_text'].count().
        sort_values(ascending = False)

def get_dm_cases_by_gender_count():
    return frame.groupby(['gender'])['gender'].count()

def get_dm_cases_average_age_gender():
    return frame.groupby(['gender'])['age'].mean().round()

def get_age_group():
    return age_grp

def get_ICD10_type2_withgender():
    return frame[frame['value_text'].str.match('Type 2') |
        frame['value_text'].str.match('Type II diabetes')]\
        .groupby(['value_text', 'gender'])['value_text'].count()\
        .unstack().fillna(0).stack()

def get_ICD10_type1_withgender():
    return frame[frame['value_text'].str.match('Type 1') |
        frame['value_text'].str.match('Type I diabetes')]\
        .groupby(['value_text', 'gender'])['value_text'].count()\
        .unstack().fillna(0).stack()

def get_ICD10_dm_underlying_cond_withgender():
    return frame[frame['value_text'].str.match('Diabetes
        mellitus')]\
        .groupby(['value_text', 'gender'])['value_text'].count()\
        .unstack().fillna(0).stack()

def get_ICD10_otherDM_withgender():
    return frame[frame['value_text'].str.match('Other specified
        diabetes mellitus')]\
        .groupby(['value_text', 'gender'])['value_text'].count()\
        .unstack().fillna(0).stack()

def get_ICD10_Drug_chemical_induced_withgender():
    return frame[frame['value_text'].str.match('Drug or
        chemical induced diabetes mellitus')]\
        .groupby(['value_text', 'gender'])['value_text'].count()\
        .unstack().fillna(0).stack()

#get data from past and family history dataset
past_fam_history_df = create_dataframe('DM_PH_FH_2')

#change_time_format(past_fam_history_df)
get_age(past_fam_history_df)

past_fam_history_df['count'] = 1
past_fam_history_df['age_group'] = pd.cut(past_fam_history_df
    .age, [0,10,20,30,40,50,60,70,80,90,100])
past_fam_history_age_grp = past_fam_history_df.pivot_table('
    count', index='age_group', columns='gender', aggfunc
    ='sum').fillna(0)

def get_history_cases_by_gender_count():
    return past_fam_history_df.groupby(['gender'])['gender'].
        count()

def get_average_age_gender():
    return past_fam_history_df.groupby(['gender'])['age'].mean
        ().round(decimals=2)

#dataframe
def get_age_group_past_fam_history():
    return past_fam_history_age_grp

def get_count_ph_fh_by_gender():
    return past_fam_history_df.groupby(['gender', 'value_coded'])
        ['gender'].count()

```

```

forms.py

from flask_wtf import FlaskForm

from wtforms import StringField, IntegerField, TextAreaField,
    SubmitField, RadioField, SelectField, DecimalField
from wtforms.fields.html5 import TelField

from wtforms.validators import InputRequired, Length, AnyOf
    ,NumberRange

class CaseForm(FlaskForm):

    age = IntegerField('Age', validators=[InputRequired('age
        is required'), NumberRange(min=0, max=100)])

    height = DecimalField('Height(cm)', places=2, rounding=
        None, validators=[InputRequired('height is required')
        , NumberRange(min=0, max=200)])
    weight = DecimalField('Weight(kg)', places=2, rounding=
        None, validators=[InputRequired('weight is required

query.py

import pandas as pd
import numpy as np

from recommend import get_data_df

search_df = get_data_df()

def get_search_data(sex, dm_type, age_group):
    print sex, dm_type, age_group
    df = get_dm_type_data(dm_type)

    sex_query = get_sex_data(sex)
    age_group_query = get_age_group_data(age_group)
    return df.query(sex_query + " and " + age_group_query)

def get_dm_type_data(dm_type_value):
    if dm_type_value == '1':
        return search_df[search_df['DIAG_DM'].str.match('Type 1')
            | search_df['DIAG_DM'].str.match('Type I diabetes')]
    elif dm_type_value == '2':
        return search_df[search_df['DIAG_DM'].str.match('Type 2')
            | search_df['DIAG_DM'].str.match('Type II diabetes')]
    elif dm_type_value == '3':
        return search_df[search_df['DIAG_DM'].str.match('Diabetes
            mellitus')]
    elif dm_type_value == '4':
        return search_df[search_df['DIAG_DM'].str.match('Other
            specified diabetes mellitus')]
    elif dm_type_value == '5':
        return search_df[search_df['DIAG_DM'].str.match('Drug or
            chemical induced diabetes mellitus')]
    else:
        return search_df

def get_sex_data(sex_value):
    if sex_value == "1":
        return 'gender == "F"'
    elif sex_value == "2":
        return 'gender == "M"'
    else:

recommend.py

import numpy as np; import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn import preprocessing

def get_data_df():
    return df

def get_data_df_count():
    return len(df)

#def calculate_CS(weight,height,wc,hc,sex,age,hba1c,fbfs):
def calculate_CS(bmi,wc,hc,sex,age,hba1c,fbfs):

    aList = []
    #female

```

```

), NumberRange(min=0, max=200))

wc = DecimalField('First Name', places=2, rounding=
    None, validators=[InputRequired('Waist
    Circumference is required'), NumberRange(min=0,
    max=200)])
hc = DecimalField('HipCircum.(cm)', places=2, rounding
    =None, validators=[InputRequired('Hip
    Circumference is required'), NumberRange(min=0,
    max=200)])

hba1c = DecimalField('HBA1C', places=2,default=0.0,
    rounding=None, validators=[InputRequired('HBA1C
    Result is required'), NumberRange(min=0, max
    =100)])
fbfs = DecimalField('fbfs', places=2, default=0.0,rounding
    =None, validators=[InputRequired('FBS Result is
    required'), NumberRange(min=0, max=500)])

sex = RadioField('Sex', choices = [('1',' Male') ,(2','
    Female')])

submit = SubmitField("Send")

return '(gender == "F" or gender == "M")'

def get_age_group_data(grp_value):

    if grp_value == "0":
        query1 = 'age>=0 '
        query2 = 'age<=10 '

    elif grp_value == "1":
        query1 = 'age>=11 '
        query2 = 'age<=20 '

    elif grp_value == "2":
        query1 = 'age>=21 '
        query2 = 'age<=30 '

    elif grp_value == "3":
        query1 = 'age>=31 '
        query2 = 'age<=40 '

    elif grp_value == "4":
        query1 = 'age>=41 '
        query2 = 'age<=50 '

    elif grp_value == "5":
        query1 = 'age>=51 '
        query2 = 'age<=60 '

    elif grp_value == "6":
        query1 = 'age>=61 '
        query2 = 'age<=70 '

    elif grp_value == "7":
        query1 = 'age>=71 '
        query2 = 'age<=80 '

    elif grp_value == "8":
        query1 = 'age>=81 '
        query2 = 'age<=90 '

    elif grp_value == "9":
        query1 = 'age>=91 '
        query2 = 'age<=100'

    elif grp_value == "ANY":
        query1 = 'age>=0 '
        query2 = 'age<=100'

    return query1 + " and " +query2

if sex == '2':
    df_flask = pd.DataFrame([[wc,hc,0,1,hba1c,fbfs,age,bmi]],
        columns=['wc','hc','isMale','isFemale','HBA1C','FBS
        ','age','BMI'])
    # df_flask = pd.DataFrame([[weight,height,wc,hc,0,1,hba1c,
        fbfs,age]],columns=['weight','height ','wc','hc','isMale
        ','isFemale','HBA1C','FBS','age'])
    #male
    else:
        df_flask = pd.DataFrame([[wc,hc,1,0,hba1c,fbfs,age,bmi]],
            columns=['wc','hc','isMale','isFemale','HBA1C','FBS
            ','age','BMI'])
        # df_flask = pd.DataFrame([[weight,height,wc,hc,1,0,hba1c,
            fbfs,age]],columns=['weight','height ','wc','hc','isMale
            ','isFemale','HBA1C','FBS','age'])

#apply Finnish Risk Assesment to new DF
df_flask['age'] = df_flask.apply(apply_FRA_age,axis = 1)
df_flask['wc'] = df_flask.apply(apply_FRA_wc,axis = 1)

```

```

df_flask ['FBS'] = df_flask .apply(apply_FRA_FBS,axis =
1)
df_flask ['BMI'] = df_flask .apply(apply_FRA_BMI,axis =
1)

frames = [rec_df, df_flask ]
c_df = pd.concat(frames)

#normalize values from 0 to 1
df_val = c_df.values
min_max_scaler = preprocessing.MinMaxScaler()
df_val_scaled = min_max_scaler.fit_transform(df_val)
df_scaled = pd.DataFrame(df_val_scaled)

#get last row.
df_input = df_scaled .iloc [[ df_scaled .shape[0]-1]]

#drop last row.
df_sc = df_scaled .drop(df_scaled .index[df_scaled .shape[0]-1])

for i,row in df_sc .iterrows():
    result = cosine.similarity (df_sc .iloc [[ i ]], df_input) [0][0]
    #print result
    aList .append([i,result ])

aList = sorted(aList,key=lambda l:l[1], reverse=True)
z = aList [0:5]
return z

# return [i [0] for i in z]

def get_data(row_num):
# data_list = pd.Series()
data_list = []
for i in row_num:
# .squeeze to convert dataframe to a pandas Series
data_list .append(df .iloc [[ i ]].squeeze())

return data_list

#for dashboard purposes
def get_max_FBS_value():
return rec_df ['FBS'] .max()/5

def get_max_HBA1C_value():
return rec_df ['HBA1C'] .max()

def get_nonDM_count():
non_dm = df .groupby(['DIAG_NONDM'])['DIAG_NONDM'] .
count().sort_values(ascending=False)
return non_dm .drop(non_dm .index[0]) .head(3)

def get_max_fbs_value_column():
row = df ['FBS'] .idxmax()

return df .iloc [[row]]

def get_max_hba1c_value_column():
row = df ['HBA1C'] .idxmax()
return df .iloc [[row]]

#apply Finnish Risk Factor Assesment:
def apply_FRA_age(row):
if row['age'] > 45:
return row['age'] * 0

elif row['age'] >= 45 and row['age'] <= 54:
return row['age'] * 2

elif row['age'] >= 55 and row['age'] <= 64:
return row['age'] * 3
else:
return row['age'] * 4

def apply_FRA_wc(row):
if row['isFemale'] == 1:
if row['wc'] < 80:
return row['wc'] * 0
elif row['wc'] >= 80 and row['wc'] <=88:
return row['wc'] * 3
else:
return row['wc'] * 4

else:
if row['wc'] < 94:
return row['wc'] * 0
elif row['wc'] >= 94 and row['wc'] <=102:
return row['wc'] * 3
else:
return row['wc'] * 4

def apply_FRA_BMI(row):
if isinstance(row['BMI'], str):
row['BMI'] = float(row['BMI'])
if row['BMI'] < 25:
return row['BMI'] * 0
elif row['BMI'] >= 25 and row['BMI'] <= 30:
return row['BMI'] * 1
else:
return row['BMI'] * 3

def apply_FRA_FBS(row):
if row['FBS'] >= 126:
return row['FBS'] * 5
return row['FBS'] * 0

def get_bmi_label(res):
res = (float(res))
if res < 18.5:
return 'Underweight'
elif res >=18.5 and res <=24.9:
return 'Normal'
elif res >=25 and res <=29.9:
return 'Overweight'
else:
return "Obesed"

# df = pd.read_csv('DS/RS_DS/final_dataset_3.csv')
df = pd.read_csv('DS/RS_DS/interpolated_data_withBMI.csv')
rec_df = df .drop(['person_id', 'weight', 'height', 'encounter_id',
', birthdate', 'CN', 'HN', 'DN', 'DIAG_DM',
'DIAG_NONDM', 'TPN', 'gender'], axis=1)

# apply FINISH RISK ASSESMENT POINTS AS WEIGHTS.
rec_df ['age'] = rec_df .apply(apply_FRA_age,axis = 1)
rec_df ['wc'] = rec_df .apply(apply_FRA_wc,axis = 1)
rec_df ['FBS'] = rec_df .apply(apply_FRA_FBS,axis = 1)
rec_df ['BMI'] = rec_df .apply(apply_FRA_BMI,axis = 1)

# print get_max_fbs_value_column()
# df_val = rec_df .values
# min_max_scaler = preprocessing.MinMaxScaler()
# df_val_scaled = min_max_scaler.fit_transform(df_val)
# df_scaled = pd.DataFrame(df_val_scaled)

# g = cosine.similarity (df_scaled)

# print pd.DataFrame(g)
# print rec_df .columns.values
# np.savetxt("foo.csv", g, delimiter=",")
# print type(rec_df .iloc [[0]]['weight'])

search_sex = SelectField('Sex', choices = [(1, 'Female'),
(2, 'Male'), (3, 'BOTH')])

search_dm_type = SelectField('DM Type', choices = [(1, '
Type 1 diabetes mellitus'), (2, ' Type 2 diabetes
mellitus'),
(3, 'Other specified diabetes mellitus'),(4,' Diabetes
mellitus due to underlying condition'),(5,' Drug or
chemical induced diabetes mellitus'),
(6,' ANY')])

search_age_group = SelectField('age group', choices =
[(0, '0-10'), (1, '11-20'), (2, '21-30'),
(3, '31-40'), (4, '41-50'), (5, '51-60'),(6,

```



```

'61-70'), ('7', '71-80'), ('8', '81-90'), ('9',
'91-100'), ('ANY', 'ANY'))]]

.navbar.html
<div class="navbar-fixed">
  <nav>
    <div class="nav-wrapper blue darken-2">

      <a href="#" data-activates="mobile-demo" class="
      button-collapse show-on-large"><i class="
      material-icons">menu</i></a>
       <a href="/" id
      = "sp_name" class="brand-logo">CHITS</a>
      <!-- <ul class="right hide-on-med-and-down">
      <li><a href="/"><i class="material-icons left">
      home</i></a></li>
      <li><a href="/recommender"><i class="material-
      icons left">local_pharmacy</i> TREATMENT
      </a></li>
      <li><a href="/search"> <i class="material-icons
      left">person_pin</i></a></li>
      <li><a href="/statistics"> <i class="material-icons
      left">insert_chart</i> STATISTICS</a></li>
      </ul> ---
      </div>
      </nav>
    </div>

    <ul class="side-nav" id="mobile-demo">
      <li>
        <div class="user-view">
          <div class="background">

            dashboard.html
            {% extends 'dashboard_layout.html' %}
            {% block body %}
            <br>

            <div class="row">
            <div class="col s12 l12 m12">
            <div class="card-panel blue darken-4 z-depth-4
            animated bounceInLeft">
            <span class="light white-text">PATIENTS WITH
            DIABETES CONSULTATIONS
            </span>
            </div>
            </div>
            </div>
            <div class="divider"> </div>

            <div class="row">
            <div class="col s12 l12 m12">
            <div class="card-panel z-depth-4"><canvas id="
            year_months_graph" width="150" height="50"></
            canvas></div>
            </div>
            </div>
            <div class="divider"> </div>

            <div class="row">
            <div class="col s12 l3 m12">
            <div class="card-panel z-depth-4"><canvas id="pie-
            chart" height="220"></canvas></div>
            </div>
            <div class="col s12 l9 m12">
            <div class="card-panel blue darken-2 z-depth-4
            animated zoomInDown">
            <h6 class="light white-text">Top 3 Non-DM
            Diagnosis [Out of {{dataset_count}}
            Consultations] </h6>
            <div class="divider"></div>
            {% for i,value in non_dm_max_value.iteritems()
            %}
            <span class="white-text"> Number of

```

```

submit_search = SubmitField("SEARCH")


</div>
<br>
<!--
<a href="#">
  <span class="name white-text">
    Diabetes Recommender System</
    span>
  </a>
  <a href="#">
    <span class="School white-text">UPM
    - CHITS</span>
  </a> ---
  <br><br>
</div>
</li>
<li><a class="subheader">NAVIGATION</a></li>
</li>
<li><a href="/"><i class="material-icons left">
home</i> HOME</a></li>
<li><a href="/faq"><i class="material-icons
left">local_hospital</i>FAQ</a></li>
<li><a href="/recommender"> <i class="material-
icons left">local_pharmacy</i> TREATMENT
</a></li>
<li><a href="/search"> <i class="material-icons
left">person_pin</i></a></li>
<li><a href="/statistics"> <i class="material-icons
left">insert_chart</i> STATISTICS</a></li>
</li>
<div class="divider"></div>
<li><a class="subheader">LINKS</a></li>
<li><a href="https://telehealth.ph/
project-chits">CHITS</a></li>
</ul>

Consultations: {{value}} - {{i}} </span
> <br>
{% endfor %}
</div>
</div>
</div>
<div class="divider"></div>
<div class="row">
<div class="col s12 l6 m12">
<div class="card-panel red z-depth-4 center animated
zoomInDown">
<span class="light white-text"> Highest Recorded
FBS result</span>
<div class="divider"></div>
<h4 class="white-text">{{fbs_max_value}} mg/
dL</h4>
<a class="waves-effect waves-light btn modal-
trigger" href="#modal_fbs">Click here to
view profile</a>
</div>
</div>
<div id="modal_fbs" class="modal">
<div class="modal-content teal">
<h4 class="white-text">Patient Profile</h4>
<div class="row">
  {% for index, row in highest_fbs_profile.iterrows
  () %}
  <p class="white-text">Weight: {{row['weight
  ']]} kg </p>
  <p class="white-text">Height: {{row['height
  ']]} cm</p>
  <p class="white-text">Waist Circumference:
  {{row['wc']}}</p>
  <p class="white-text">Hip Circumference: {{
  row['hc']}}</p>
  <p class="white-text">Age: {{row['age']}}</
  p>
  <p class="white-text">Gender: {{row['gender
  ']]}}</p>
  <p class="white-text">Hba1C Result: {{row['
  HBA1C']}}%</p>
  <p class="white-text">FBS Result: {{row['
  FBS']}} mg/dL</p>

```



```

        text: 'Number of Consultations per diagnosis [Total
        Consultations: ' + dm_specific_cases_total +']'
    }
}
});

</div class="divider"> </div>
<br>
<div class="row">
<div class="col s12 l12 m12">
<div class="card-panel red z-depth-5 animated
bounceInLeft">
<span class="white-text"> PATIENTS WITH
FAMILY HISTORY OR PAST HISTORY
</span>

<p class="white-text light">Some data for males
may not appear due to the small population. To
view data on males, click on the female label of
each graph</p>
</div>
</div>
<div class="divider"> </div>

<div class="row">
<div class="col s12 l6 m12">
<div class="card-panel z-depth-4"><canvas id="
dm_ph_fh_gender" width="400" height="150"></
canvas></div>
</div>

<div class="col s12 l6 m12">
<div class="card-panel z-depth-4"><canvas id="
dm_history_gender_count_by_gender_ph_fh_bar-chart
-grouped" width="800" height="300"></canvas
></div>
</div>

</div>

<div class="row">

<div class="col s12 l12 m12">
<div class="card-panel z-depth-4"><canvas id="bar-
chart-horizontal" width="800" height="150"></
canvas></div>
</div>

<div class="row">
<div class="col s12 l12 m12">
<div class="card-panel z-depth-4"><canvas id="
dm_ph_fh_age_group_bar-chart-grouped" width
="800" height="200"></canvas></div>
</div>
</div>

{% endblock %}

{% block script %}

<script type="text/javascript">

var dm_specific_cases = [];
var dm_specific_values = [];
var dm_specific_cases_total = 0
{% for i in dm_cases_specific_count %}
    dm_specific_cases.push('{{i[0]}}');
    dm_specific_values.push('{{i[1]}}');
    dm_specific_cases_total = dm_specific_cases_total +
    {{i[1]}};
{% endfor %}
dm_specific_cases_total = dm_specific_cases_total.toString
();

new Chart(document.getElementById("specific-dm-cases-
pie-chart"), {
type: 'pie',
data: {
labels: dm_specific_cases,
datasets: [{
label: "Population",
backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f",
"#e8c3b9", "#c45850"],
data: dm_specific_values
}]
},
options: {
title: {
display: true,
}
}
});

text: 'Number of Consultations per diagnosis [Total
Consultations: ' + dm_specific_cases_total +']'
}
});

var dm_cases_ave_age_labels = [];
var dm_cases_ave_age_values = []
{% for i,value in dm_cases_ave_age.iteritems() %}
    dm_cases_ave_age_labels.push('{{i}}');
    dm_cases_ave_age_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("dm_cases_bar-chart
-horizontal"), {
type: 'horizontalBar',
data: {
labels: dm_cases_ave_age_labels,
datasets: [
{
label: "Average Age by Gender",
backgroundColor: ["#e53935", "#1565c0"],
data: dm_cases_ave_age_values
}
]
},
options: {
legend: { display: false },
title: {
display: true,
text: 'Average Age by Gender with DM
Consultations'
}
}
});

/*year_months_graph*/
var year_months = [];
var data_year_months = []
{% for i,value in dm_cases_year_month.iteritems() %}
    year_months.push('{{i}}');
    data_year_months.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("year_months_graph
"), {
type: 'line',
data: {
labels: year_months,
datasets: [{
data: data_year_months,
label: "Consultations",
borderColor: "#3e95cd",
fill: false
}],
},
options: {
title: {
display: true,
text: 'Number of Diabetes Consultations'
}
}
});

/*cases_count*/

var dm_cases = [];
var dm_values = [];
{% for i,value in dm_cases_count.iteritems() %}
    dm_cases.push('{{i}}');
    dm_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("dm_cases_graph"),
{
type: 'horizontalBar',
data: {
labels: dm_cases,
datasets: [
{
label: "Number of cases",
backgroundColor: ["#3e95cd", "#8e5ea2", "#3
cba9f", "#e8c3b9", "#c45850", "#3e95cd",
"#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850",
"#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#
c45850", "#3e95cd", "#8e5ea2", "#3cba9f",
"#e8c3b9", "#c45850", "#3e95cd"],
data: dm_values
}
]
},
options: {
legend: { display: false },
}
});

```

```

        title : {
            display: true,
            text: 'Number of Consultations per Specific
                Diagnosis'
        }
    });

    //gender_count

    var dm_cases_gender= [];
    var dm_values_gender = [];

    {% for i,value in cases_gender.iteritems() %}
        dm_cases_gender.push('{{i}}');
        dm_values_gender.push('{{value}}');
    {% endfor %}

    new Chart(document.getElementById("pie-chart"), {
        type: 'pie',
        data: {
            labels: dm_cases_gender,
            datasets: [{
                label: "Gender",
                backgroundColor: ["#f44336", "#3e95cd"],
                data: dm_values_gender
            }
        ],
        options: {
            title: {
                display: true,
                text: 'Population by Gender with DM
                    Consultations'
            }
        }
    });

    var dm_cases_female= [];
    var dm_cases_male = [];
    var dm_cases_age_group = [];

    {% for a_group,value in dm_cases_age_grp.iterrows() %}
        dm_cases_female.push('{{ value['F'] }}')
        dm_cases_male.push('{{ value['M'] }}')
        dm_cases_age_group.push('{{ a_group }}')
    {% endfor %}

    new Chart(document.getElementById("age_group_bar-chart
        -grouped"), {
        type: 'bar',
        data: {
            labels: dm_cases_age_group,
            datasets: [
                {
                    label: "Female",
                    backgroundColor: "#e53935",
                    data: dm_cases_female
                }, {
                    label: "Male",
                    backgroundColor: "#0d47a1",
                    data: dm_cases_male
                }
            ]
        },
        options: {
            title: {
                display: true,
                text: 'Age Group by Gender with DM Consultations'
            }
        }
    });

    var dm_history_sex_labels= [];
    var dm_history_sex_values = [];
    {% for i,value in dm_history_gender.iteritems() %}
        dm_history_sex_labels.push('{{i}}');
        dm_history_sex_values.push('{{value}}');
    {% endfor %}

    new Chart(document.getElementById("dm_ph_fh_gender"),
        {
            type: 'pie',
            data: {
                labels: dm_history_sex_labels,
                datasets: [{
                    label: "Average Age",
                    backgroundColor: ["#8e5ea2", "#3cba9f"],
                    data: dm_history_sex_values
                }
            ]
        },
        options: {
            title: {
                display: true,
                text: 'Population by Gender with Family History or
                    Past History DM'
            }
        }
    });

    var dm_history_gender_count_by_gender_ph_fh_labels= [];
    var dm_history_gender_count_by_gender_past_history_values
        = [];;
    var dm_history_gender_count_by_gender_family_history_values
        = [];;

    {% for i,value in dm_history_gender_count_by_gender.
        iteritems() %}
        dm_history_gender_count_by_gender_ph_fh_labels.push
            ('{{i[0]}}');

        if ('{{i[1]}}' == '1')
            dm_history_gender_count_by_gender_past_history_values
                .push('{{value}}');
        else
            dm_history_gender_count_by_gender_family_history_values
                .push('{{value}}');

        /*console.log('{{i[0]}}');
        console.log('{{i[1]}}' == '1');
        console.log('{{value}}');*/
    {% endfor %}

    new Chart(document.getElementById("
        dm_history_gender_count_by_gender_ph_fh-bar-chart
        -grouped"), {
        type: 'bar',
        data: {
            labels:
                dm_history_gender_count_by_gender_ph_fh_labels
                    .slice(1, 3),
            datasets: [
                {
                    label: "Past History",
                    backgroundColor: "#3e95cd",
                    data:
                        dm_history_gender_count_by_gender_past_history_values
                }, {
                    label: "Family History",
                    backgroundColor: "#8e5ea2",
                    data:
                        dm_history_gender_count_by_gender_family_history_values
                }
            ]
        },
    });

    var dm_ph_fh_cases_female= [];
    var dm_ph_fh_cases_male = [];
    var dm_ph_fh_cases_age_group = [];

    {% for a_group,value in dm_history_gender_age_grp.iterrows
        () %}
        dm_ph_fh_cases_female.push('{{ value['F'] }}')
        dm_ph_fh_cases_male.push('{{ value['M'] }}')
        dm_ph_fh_cases_age_group.push('{{ a_group }}')
    {% endfor %}

    new Chart(document.getElementById("
        dm_ph_fh_age_group_bar-chart-grouped"), {
        type: 'bar',
        data: {
            labels: dm_ph_fh_cases_age_group,
            datasets: [
                {
                    label: "Female",
                    backgroundColor: "#e53935",
                    data: dm_ph_fh_cases_female
                }, {
                    label: "Male",
                    backgroundColor: "#0d47a1",
                    data: dm_ph_fh_cases_male
                }
            ]
        },
        options: {
            title: {
                display: true,
                text: 'Age Group by Gender with Family History
                    or Past History DM'
            }
        }
    });

```

```

    }
  });

var dm_history_ave_age_labels= [];
var dm_history_ave_age_values = [];
{% for i,value in dm_history_gender_age_ave.iteritems() %}
  dm_history_ave_age_labels.push('{{i}}');
  dm_history_ave_age_values.push('{{value}}');
{% endfor %}
new Chart(document.getElementById("bar-chart-horizontal"), {
  type: 'horizontalBar',
  data: {
    labels: dm_history_ave_age_labels,
    datasets: [
      {
        label: "Population (millions)",
        backgroundColor: ["#e53935", "#3e95cd"],
        data: dm_history_ave_age_values
      }
    ]
  },
  options: {
    legend: { display: false },
    title: {
      display: true,
      text: 'Average Age by Gender with Family History /Past History'
    }
  }
});

var ICD10_type2_labels = [];
var ICD10_type2_values = [];

{% for i,value in ICD10_type2.iteritems() %}
  ICD10_type2_labels.push('{{i}}');
  ICD10_type2_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("doughnut-chart_icd_type2"), {
  type: 'doughnut',
  data: {
    labels: ICD10_type2_labels,
    datasets: [
      {
        label: "Number of Consultations",
        backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850"],
        data: ICD10_type2_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Type 2 diabetes mellitus'
    }
  }
});

var ICD10_type1_labels = [];
var ICD10_type1_values = [];

{% for i,value in ICD10_type1.iteritems() %}
  ICD10_type1_labels.push('{{i}}');
  ICD10_type1_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("doughnut-chart_icd_type1"), {
  type: 'doughnut',
  data: {
    labels: ICD10_type1_labels,
    datasets: [
      {
        label: "Number of Consultations",
        backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850"],
        data: ICD10_type1_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Type 1 diabetes mellitus'
    }
  }
});

var ICD10_underlying_cond_labels = [];
var ICD10_underlying_cond_values = [];

{% for i,value in ICD10_underlying_cond.iteritems() %}
  ICD10_underlying_cond_labels.push('{{i}}');
  ICD10_underlying_cond_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("bar-chart-horizontal_idc_underlying_condition"), {
  type: 'horizontalBar',
  data: {
    labels: ICD10_underlying_cond_labels,
    datasets: [
      {
        label: "Number of Consultations",
        backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850", "#d81b60", "#4a148c", "#0d47a1", "#006064", "#fdd835", "#ff6f00", "#f4511e", "#00e676"],
        data: ICD10_underlying_cond_values
      }
    ]
  },
  options: {
    legend: { display: false },
    title: {
      display: true,
      text: 'Diabetes mellitus due to underlying condition'
    }
  }
});

var ICD10_otherDM_labels = [];
var ICD10_otherDM_values = [];

{% for i,value in ICD10_otherDM.iteritems() %}
  ICD10_otherDM_labels.push('{{i}}');
  ICD10_otherDM_values.push('{{value}}');
{% endfor %}

new Chart(document.getElementById("doughnut-chart_icd_otherDM"), {
  type: 'doughnut',
  data: {
    labels: ICD10_otherDM_labels,
    datasets: [
      {
        label: "Number of Consultations",
        backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850"],
        data: ICD10_otherDM_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Other specified diabetes mellitus'
    }
  }
});

var ICD10_Drug_Chemical_induced_labels = [];
var ICD10_Drug_Chemical_induced_values = [];

{% for i,value in ICD10_Drug_Chemical_induced.iteritems() %}
  ICD10_Drug_Chemical_induced_labels.push('{{i}}');
  ICD10_Drug_Chemical_induced_values.push('{{value}}');
{% endfor %}

```

```

{% endfor %}

new Chart(document.getElementById("doughnut-chart-icd10-Drug-Chemical-induced"), {
  type: 'doughnut',
  data: {
    labels: ICD10_Drug_Chemical_induced_labels,
    datasets: [
      {
        label: "Number of Consultations",
        backgroundColor: ["#3e95cd", "#8e5ea2", "#3cba9f", "#e8c3b9", "#c45850"],
        data: ICD10_Drug_Chemical_induced_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Drug or chemical induced diabetes mellitus'
    }
  }
});

/* unique array JS*/
function onlyUnique(value, index, self) {
  return self.indexOf(value) === index;
}

var E08_M_values = []
var E08_F_values = []
var E08_labels = []

{% for i,value in ICD10_dm_underlying_cond_withgender.iteritems() %}
E08_labels.push('{{i[0]}}')
if('{{i[1]}}' == "M"){
  E08_M_values.push('{{value}}')
}
else{
  E08_F_values.push('{{value}}')
}

{% endfor %}

E08_labels = E08_labels.filter ( onlyUnique );

new Chart(document.getElementById("bar-chart-grouped-E08"), {
  type: 'horizontalBar',
  data: {
    labels: E08_labels ,
    datasets: [
      {
        label: "Male",
        backgroundColor: "#3e95cd",
        data:E08_M_values
      }, {
        label: "Female",
        backgroundColor: "#8e5ea2",
        data: E08_F_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Diabetes mellitus due to underlying condition'
    }
  }
});

var E09_M_values = []
var E09_F_values = []
var E09_labels = []

{% for i,value in ICD10_Drug_chemical_induced_withgender.iteritems() %}
E09_labels.push('{{i[0]}}')
if('{{i[1]}}' == "M"){
  E09_M_values.push('{{value}}')
}
else{
  E09_F_values.push('{{value}}')
}

{% endfor %}

E09_labels = E09_labels.filter ( onlyUnique );

new Chart(document.getElementById("bar-chart-grouped-E09"), {
  type: 'horizontalBar',
  data: {
    labels: E09_labels,
    datasets: [
      {
        label: "Male",
        backgroundColor: "#3e95cd",
        data: E09_M_values
      }, {
        label: "Female",
        backgroundColor: "#8e5ea2",
        data: E09_F_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Drug or chemical induced diabetes mellitus'
    }
  }
});

var E10_M_values = []
var E10_F_values = []
var E10_labels = []

{% for i,value in ICD10_type1_withgender.iteritems() %}
E10_labels.push('{{i[0]}}')
if('{{i[1]}}' == "M"){
  E10_M_values.push('{{value}}')
}
else{
  E10_F_values.push('{{value}}')
}

{% endfor %}

E10_labels = E10_labels.filter ( onlyUnique );

new Chart(document.getElementById("bar-chart-grouped-E10"), {
  type: 'horizontalBar',
  data: {
    labels: E10_labels,
    datasets: [
      {
        label: "Male",
        backgroundColor: "#3e95cd",
        data: E10_M_values
      }, {
        label: "Female",
        backgroundColor: "#8e5ea2",
        data: E10_F_values
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Type 1 diabetes mellitus'
    },
    scales: {
      yAxes: [{
        scaleLabel: {
          display: true
        }
      }
    ],
    xAxes: [{
      ticks: {
        min: 0,
        stepSize: 1,
        max: 10
      },
      scaleLabel: {
        display: true
      }
    }
  ]
});

var E11_M_values = []
var E11_F_values = []
var E11_labels = []

{% for i,value in ICD10_type2_withgender.iteritems() %}
E11_labels.push('{{i[0]}}')
if('{{i[1]}}' == "M"){

```

```

        E11_M_values.push('{{value}}')
    }
    else {
        E11_F_values.push('{{value}}')
    }
}

{% endfor %}

E11_labels = E11_labels.filter ( onlyUnique );

new Chart(document.getElementById("bar-chart-
grouped_E11"), {
type: 'horizontalBar',
data: {
labels: E11_labels,
datasets: [
{
label: "Male",
backgroundColor: "#3e95cd",
data: E11_M_values
}, {
label: "Female",
backgroundColor: "#8e5ea2",
data: E11_F_values
}
]
},
options: {
title: {
display: true,
text: 'Type 2 diabetes mellitus'
}
}
});

var E13_M_values =[]
var E13_F_values =[]
var E13_labels =[]

{% for i,value in ICD10_otherDM.withgender.iteritems()
%}
E13_labels.push('{{i[0]}}')
if('{{i[1]}}' == "M"){
E13_M_values.push('{{value}}')
}
else {
E13_F_values.push('{{value}}')
}
}

{% endfor %}

```

dashboard\_layout.html

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
<meta name="viewport" content="width=device-width
, initial-scale=1, maximum-scale=1.0"/>
<!-- <link href="https://fonts.googleapis.com/icon?
family=Material+Icons" rel="stylesheet" > -->
<link rel="stylesheet" type="text/css" href=" {{
url_for('static',filename='css/materialize.css')
}}">
<link rel="stylesheet" type="text/css" href=" {{
url_for('static',filename='css/style.css') }}">
<link rel="stylesheet" type="text/css" href=" {{
url_for('static',filename='css/icon.css') }}">
<link rel="stylesheet" type="text/css" href=" {{
url_for('static',filename='css/animate.css')
}}">
<link rel="stylesheet" type="text/css" href=" {{ url_for('
static',filename='css/info-circle.css') }}">
<style type="text/css">
body {
display: flex;
min-height: 100vh;
flex-direction: column;
}
main {
flex: 1 0 auto;

```

```

E13_labels = E13_labels.filter ( onlyUnique );

new Chart(document.getElementById("bar-chart-
grouped_E13"), {
type: 'horizontalBar',
data: {
labels: E13_labels,
datasets: [
{
label: "Male",
backgroundColor: "#3e95cd",
data: E13_M_values
}, {
label: "Female",
backgroundColor: "#8e5ea2",
data: E13_F_values
}
]
},
options: {
title: {
display: true,
text: 'Other specified diabetes mellitus'
},
scales: {
yAxes: [{
scaleLabel: {
display: true
}
}],
xAxes: [{
ticks: {
min: 0,
stepSize: 1,
max: 10
},
scaleLabel: {
display: true
}
}],
}
}
});

```

```

</script>
{% endblock %}

```

```

}
main, footer {
padding-left: 300px;
}
@media only screen and (max-width : 992px) {
header, main, footer {
padding-left: 0;
}
/* nav */
}
}
</style>
<title>CHITS Diabetes Recommender System</title>
</head>
<body>
<header>
<div class="navbar-fixed">
<nav>
<div class="nav-wrapper blue darken-2">
<a href="#" data-activates="mobile-demo" class="
button-collapse"><i class="material-icons">
menu</i></a>

<a href="/" id="sp_name" class="brand-logo">&

```

```

      nbsp CHITS</a>
    </div>
  </div>
</nav>
</div>
<ul class="side-nav fixed" id="mobile-demo">
  <!-- <li>
    <div class="user-view">
      <div class="background blue darken-2">
        </div>
      <br>
    </div>
  </li> -->
  <li><a class="subheader">NAVIGATION</a></li>
  <li><a href="#"><i class="material-icons left">
    home</i> HOME</a></li>
  <li><a href="/faq"><i class="material-icons left">
    local.hospital</i>FAQ</a></li>
  <li><a href="/recommender"> <i class="material-
    icons left">local.pharmacy</i> TREATMENT
    </a></li>
  <li><a href="/search"> <i class="material-icons
    left">person.pin</i>SEARCH</a></li>
  <li><a href="/statistics"><i class="material-icons
    left">insert.chart</i> STATISTICS</a></li>
    <div class="divider"></div>
    <li><a class="subheader">LINKS</a></li>
    <li><a href="https://telehealth.ph/project-chits/">CHITS</a></li>
  </ul>

</header>

<!-- <div class="container"> -->
<main>
  {% block body%}

  {% endblock %}
</div>
-->
</main>
<footer class="page-footer blue darken-2">
  <div class="container">
    <div class="row">
      <div class="col l6 s12">
        <h5 class="white-text">SUPPORT</h5>
        <p class="grey-text text-lighten-4">
          Please send an email to: <a class="grey-
            text text-lighten-4" href="mailto:
            rbcabrera1@up.edu.ph">rbcabrera1@up
            .edu.ph</a></p>
      </div>

      <div class="col l4 offset -12 s12">
        <h5 class="white-text">Links</h5>
        <ul>
          <li><a class="grey-text text-lighten-3"
            href="https://telehealth.ph/project-
            chits/" target="_blank">CHITS</a>
          ></li>
        </ul>
      </div>
    </div>
  </div>

  faq.html
  {% extends 'layout.html' %}
  {% block body %}
  <div class="container">
    <div class="section">
      <h5 class="light">How does the system recommend?</div>
      <div class="divider"></div>
      <br>
    </div>
  </div>

```

```

<div>Icons made by <a href="https://www.
  flaticon.com/authors/wissawa-khamsriwath
  " title="Wissawa Khamsriwath">Wissawa
  Khamsriwath</a> from <a href="https://
  www.flaticon.com/" title="Flaticon">www.
  flaticon.com</a> is licensed by <a href="
  http://creativecommons.org/licenses/by
  /3.0/" title="Creative Commons BY 3.0"
  target="_blank">CC 3.0 BY</a></div>
</div>
<div class="footer-copyright">
  <div class="container">
    2018 UPM.RJBC
  </div>
</div>
</footer>
<script src="https://code.jquery.com/jquery-2.1.1.min.js
"></script>
<script type="text/javascript">
  $(document).ready(function(){
    $('. slider '). slider ();
  });
  $(document).ready(function(){
    $('. parallax '). parallax ();
  });
  $(document).ready(function() {
    $(' select '). material_select ();
  });
  $(document).ready(function(){
    $('. collapsible '). collapsible ();
  });
  $(document).ready(function(){
    // the "href" attribute of the modal trigger must specify
    the modal ID that wants to be triggered
    $('. modal '). modal ();
  });
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.
  js/2.7.1/Chart.min.js"></script>
<script defer src="https://use.fontawesome.com/releases/
  v5.0.8/js/all.js" integrity="sha384-
  SIE9911GASHoBfWbelyBPLsUlWY1GwNDJ03JSJO04KZ33K2bwfV9YBauFm
  " crossorigin="anonymous"></script>
  {% block script%}
  <script type="text/javascript" src="{ { url_for(' static ',
    filename = 'js/graphs.js' ) }}"></script>
  {% endblock %}
  <script type="text/javascript" src="{ { url_for(' static ',
    filename = 'js/materialize.js ' ) }}"></script>
  <script type="text/javascript" src="{ { url_for(' static ',
    filename = 'js/layout.js ' ) }}"></script>
  <script type="text/javascript" src="{ { url_for(' static ',
    filename = 'js/init.js ' ) }}"></script>
</body>
</html>
<div class="col l4 m12 s12 hide-on-med-and-down
  animated slideInLeft"> </div>
<div class="col l8 m10 s10">
  <p class="light animated fadeIn">
    The system analyzes the patient's profiles and
    attempts to find a patient similar to current
    patient. The system compares the patient's
    weight, height, waist circumference, hip
    circumference, age, gender, HbA1c Result, and
    FBS result.
  </p>

```



```

</p>
  <p class="light animated fadeIn">
    The algorithm used to find similar patients is
    called Cosine Similarity. <a href="http://
    mlwiki.org/index.php/Cosine.Similarity"
    target="_blank">Click here for more
    information on how the algorithm works</a>
  </p>

  <p class="light animated fadeIn">
    Once the system finds similar patients, it provides
    the top 5 nearest patients and gives out patient
    profile information and treatment as
    recommendation.
  </p>

  <p class="red-text light animated fadeIn">
    <strong>WARNING:</strong> The recommendation
    given by the system does not guarantee
    improving the patient's condition. It is still
    recommended to consult a doctor for proper
    medication
  </p>

  <a href="/recommender" class="waves-effect waves-
  light btn">GET RECOMMENDATION</a>
</div>
</div>

<div class="section">
  <h5 class="light">DASHBOARD DATA</h5>
  <div class="divider"></div>
  <br>
  <div class="row">
  <div class="col l8 m12 s12">
    <p class="light animated fadeIn">

    The data for the analysis was collected from CHITS (
    Community Health Information Tracking System
    ). The data extraction and analysis was done
    using Python and SQL.
  </p>

  <a href="/statistics" class="waves-effect waves-light
  red btn">DATA ANALYSIS</a>

</div>

<div class="col l4 m12 s12 hide-on-med-and-down
  animated slideInRight"> </div>
</div>
</div>

<div class="section">
  <h5 class="light">Multiple Platforms</h5>
  <div class="divider"></div>
  <br>
  <div class="row">
  <div class="col l4 m12 s12 hide-on-med-and-down
  animated slideInLeft"></div>
  <div class="col l8 m12 s12 animated fadeIn"> The
  system may be used in all types of platforms from
  mobile, tablets and desktop/laptop computers. </
  div>

</div>
</div>
</div>

{% endblock %}

</div>
</li>
<!--
  <li>
    
    <div class="caption center-align">
      <h3>Healthier Diet</h3>
      <h5 class="light grey-text text-lighten-3">Here'
      s our small slogan.</h5>
    </div>
  </li> -->
</ul>
</div>

<!-- Icon Section -->
<div class="row container">
  <div class="col m4">

  <div class="icon-block">
    <h2 class="center red-text"><i class="
    material-icons">local_hospital</i></h2>

    <h6 class="center" style="font-size:85%;">
    Diabetes Recommender System</h6>
    <div class="divider"></div>
    <p class="light">
      The system allows patients to improve their
      lifestyle by recommending medical/
      non-medical treatments.<br> <br>
      Can be used as a decision support
      system by Health Professionals or as
      a guide to patients

  </p>
</div>

</div>

<div class="col m4">
  <div class="icon-block">
    <h2 class="center light-blue-text"><i class
    ="material-icons">tag_faces</i></h2>
    <h6 class="center">Patient-Centered</h6>
    <div class="divider"></div>
    <p class="light">The system analyzes your
    profile to recommend you the best
  </p>
</div>

```

```

        regimen that suits your needs. </p>
    </div>
</div>
<div class="col m4">
<div class="icon-block">
    <h2 class="center orange-text"><i class="
        material-icons">trending-up</i></h2>
    <h6 class="center">Research and Development
    </h6>
    <div class="divider"></div>
    <p class="light">Researchers may view
        trends in diabetes cases in the
        Philippines for further research
        and analysis. This may enable them to
        develop proper healthcare managment
        plan to improve healthcare systems</p>
    </div>
</div>
</div>
</div>

```

```

    </div>
</div>
</div>
<div class="parallax">
    
    </div>
</div>
<div class="row">
<div class="col s12 m12 l12">
    <canvas id="line-chart" width="300" height
        ="80"></canvas>
</div>
</div>

```

```
<br>
```

```
<br> <br>
```

```

<div class="parallax-container" style="height:300px;">
<div class="section no-pad-bot">
<div class="container">
<div class="row center">
<h1 class="center white-text" style="font
    -weight: 100;">State of the
    Philippines</h1>
<br><br>
<h5 class="header col s12 light white-text
    ">Diabetes rank 4<sup>th</sup> in
    the Top 10 diseases in the Philippines
    </h5>

```

```
{% endblock %}
```

```
layout.html
```

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8"/>
<meta name="viewport" content="width=device-width
    , initial-scale=1, maximum-scale=1.0"/>
<link rel="stylesheet" type="text/css" href="{ {
    url_for('static', filename='css/materialize.css')
    }}">
<link rel="stylesheet" type="text/css" href="{ {
    url_for('static', filename='css/style.css') }}">
<link rel="stylesheet" type="text/css" href="{ {
    url_for('static', filename='css/icon.css') }}">
<link rel="stylesheet" type="text/css" href="{ {
    url_for('static', filename='css/animate.css')
    }}">
<link rel="stylesheet" type="text/css" href="{ { url_for('
    static', filename='css/info-circle.css') }}">
<link rel="stylesheet" type="text/css" href="{ { url_for('
    static', filename='css/jquery.dataTables.min.css') }}">
<!-- <link rel="stylesheet" type="text/css" href="{ {
    url_for('static', filename='css/jquery-gauge.css') }}">
    -->
<style type="text/css">
    body {
    display: flex;
    min-height: 100vh;
    flex-direction: column;
    }
    main {
    flex: 1 0 auto;
    }
    span{
    margin: 0em;
    }
</style>

```

```

<!-- <div class="container"> -->
    <main>
    {% block body%}
    {% endblock %}
</div>
<!-- </div>
-->
</main>
<footer class="page-footer blue darken-2">
<div class="container">
<div class="row">
<div class="col l6 s12">
<h5 class="white-text">SUPPORT</h5>
<p class="grey-text text-lighten-4">
    Please send an email to: <a class="grey
    -text text-lighten-4" href="mailto:
    rbcabrera1@up.edu.ph">rbcabrera1@up
    .edu.ph</a></p>
</div>
<div class="col l4 offset-l2 s12">
<h5 class="white-text">Links</h5>
<ul>
<li><a class="grey-text text-lighten-3"
    href="https://telehealth.ph/project-
    chits/" target="_blank">CHITS</a
    ></li>
</ul>
</div>
</div>
<!-- <div>Icons made by <a href="https://
    www.flaticon.com/authors/wissawa-
    khamsriwath" title="Wissawa Khamsriwath
    ">Wissawa Khamsriwath</a> from <a href
    ="https://www.flaticon.com/" title="
    Flaticon">www.flaticon.com</a> is licensed
    by <a href="http://creativecommons.org/
    licenses/by/3.0/" title="Creative Commons
    BY 3.0" target="_blank">CC 3.0 BY</a
    ></div>
<div>Icons made by <a href="http://www.
    freepik.com" title="Freepik">Freepik</a>
    from <a href="https://www.flaticon.com/"
    title="Flaticon">www.flaticon.com</a> is
    licensed by <a href="http://
    creativecommons.org/licenses/by/3.0/" title
    ="Creative Commons BY 3.0" target="
    _blank">CC 3.0 BY</a></div>

```

```

<title>CHITS Diabetes Recommender System</title>
</head>
<body>
{% include '_navbar.html' %}

```



```

<label for="first_name">Hip Circumference(cm)
  </label>

  {% for error in form.hc.errors %}
    <h6 style="color:red;">{{ error }}</h6>
  {% endfor %}

</div>
</div>
<div class="row">
<div class="input-field col s12 m6 l6">
  {{ form.age }}
  <label for="first_name">Age</label>

  {% for error in form.age.errors %}
    <h6 style="color:red;">{{ error }}</h6>
  {% endfor %}

</div>

  <div style="padding: 20px;">
    <label style="font-size: 120%;">Sex:</label>
    <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">
      {% for subfield in form.sex %}
        <tr>
          <td>{{ subfield (class="with-gap") }}</td>
          <td>{{ subfield.label }}</td>
        </tr>
      {% endfor %}

      {% for error in form.sex.errors %}
        <h6 style="color:red;">{{ error }}</h6>
      {% endfor %}
    </div>
  </div>

<div class="divider"></div>

  <div class="row">
    <div class="col s12 m12 l4"> <h5 class="grey-text">Lab Results</h5> </div>
  <div class="col s12 m12 l8">
    <div class="card-panel red">
      <span class="white-text"><span>Default value is 0.00 if result is unavailable</span>
    </span>
  </div>
</div>

  <div class="row">
    <div class="input-field col s12 m12 l6">
      {{ form.hba1c }}
      <label class="active" for="first_name">HBA1C (%)</label>

      {% for error in form.hba1c.errors %}
        <h6 style="color:red;">{{ error }}</h6>
      {% endfor %}
    </div>

    <div class="input-field col s12 m12 l6">
      {{ form.fbs }}
      <label class="active" for="first_name">Fasting Blood Sugar(mg/dL) </label>

      {% for error in form.fbs.errors %}
        <h6 style="color:red;">{{ error }}</h6>
      {% endfor %}
    </div>
  </div>

<div class="right">
  <button class="btn waves-effect waves-light" type="submit" name="action">Submit
  <i class="material-icons right">send</i>
</div>
</button>
</div>
</fieldset>
</form>
<br>

<div class="col s12 m12 l5">
  <div class="card-panel blue darken-3 animated slideInRight">
    <span class="white-text">Information: Click to view information for each data.
  </span>
</div>

  <ul class="collapsible popout animated slideInRight" data-collapsible="accordion">
    <li>
      <div class="collapsible-header"><i class="fas fa-weight"></i></div>
      <div class="collapsible-body"><p class="light">The more fatty tissue you have, the more resistant your cells become to insulin.</p>
      <a href="https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444">Source</a>
    </li>
    <li>
      <div class="collapsible-header"><i class="fas fa-plus"></i></div>
      <div class="collapsible-body"><p class="light">Obese people are more likely to develop type 2 diabetes. Extra weight sometimes causes insulin resistance and is common to people with type 2 DM. BMI is used to see if you are at risk of type 2 diabetes. It is computed as BMI = kg/m<sup>2</sup></sup> </p>
      <a href="https://www.niddk.nih.gov/health-information/diabetes/overview/symptoms-causes">Source</a>
    </li>
    <li>
      <div class="collapsible-header"><i class="fas fa-heartbeat"></i></div>
      <div class="collapsible-body"><p class="light">The waist-to-hip ratio (WHR) is a quick measure of fat distribution that may help indicate a person's overall health. People who carry more weight around their middle than their hips may be at a higher risk of developing certain health conditions. According to the World Health Organization (WHO), having a WHR of over 1.0 may increase the risk of developing conditions that relate to being overweight, including heart disease and type 2 diabetes. </p>
      <a href="https://www.medicalnewstoday.com/articles/319439.php">Source</a>
    </li>
    <li>
      <div class="collapsible-header"><i class="fas fa-sort-numeric-up"></i></div>
      <div class="collapsible-body"><p class="light">Your risk increases as you get older. This may be because you tend to exercise less, lose muscle mass and gain weight as you age </p>
      <a href="https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444">Source</a>
    </li>
    <li>
      <div class="collapsible-header"><i class="fas fa-male"></i></div>
      <div class="collapsible-body"><p class="light">Sex-related differences in lifestyle may lead to differences in the risk of developing diabetes mellitus </p>
      <a href="http://www.who.int/bulletin/volumes/91/9/12-113415/en/">Source</a>
    </li>
  </ul>

```

```

</li>
  <div class="collapsible-header"><i class="fas fa-sort-numeric-up"></i> &nbsp;HbA1C</div>
  <div class="collapsible-body"><p class="light">This blood test indicates your average blood sugar level for the past two to three months. It measures the percentage of blood sugar attached to hemoglobin, the oxygen-carrying protein in red blood cells. The higher your blood sugar levels, the more hemoglobin you'll have with sugar attached. An A1C level of 6.5 percent or higher on two separate tests indicates that you have diabetes. An A1C between 5.7 and 6.4 percent indicates prediabetes. Below 5.7 is considered normal.mellitus </p><a href="https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451">Source</a></div>
</li>
</div>
</div>
</div>
<div class="collapsible-header"><i class="fas fa-sort-numeric-up"></i> &nbsp;FBS</div>
  <div class="collapsible-body"><p class="light">A blood sample will be taken
  </div>
</div>
  <div class="collapsible-body">
    <div style="height: 130px; id="g1"></div>
    <script type="text/javascript">
      window.onload = function () {
        var g1 = new JustGage({
          id: "g1",
          value: {{bmi}},
          min: 0,
          max: 30,
          title: " {{bmi_res}}",
          relativeGaugeSize: true
        });
      }
    </script>
  </div>
</li>
</ul>
</div>
</div>
<div class="row">
  <div class="col s12 m12 l12">
    <div class="card-panel blue darken-2 z-depth-4 animated slideInRight">
      <span class="white-text"> HERE ARE THE RESULTS OF THE ANALYSIS
    </span>
  </div>
</div>
<div class="col s12 m12 l12">
  <ul class="collapsible">
    <li>
      <div class="collapsible-header active blue darken-2 white-text"><i class="material-icons">info</i>>Patient Data</div>
      <div class="collapsible-body">
        <div class="row">
          <div class="col s12 l3 m12">
            <span>Weight:{{form.weight.data}} kg</span><br>
            <span>Height:{{form.height.data}} cm</span>
          </div>
          <div class="col s12 l4 m12">
            <span>Waist Circumference:{{form.wc.data}} cm</span><br>
            <span>Hip Circumference:{{form.hc.data}} cm</span>
          </div>
          <div class="col s12 l2 m12">
            <span>Age:{{form.age.data}}</span><br>
            <span>Sex:{{sex_label}}</span>
          </div>
          <div class="col s12 l3 m12">
            <span>HbA1C:{{form.hba1c.data}}%</span><br>
            <span>FBS:{{form.fbs.data}} mg/dL</span>
          </div>
        </div>
      </div>
    </li>
  </ul>
</div>
  <div class="col s12 l3 m12">
    <ul class="collapsible" data-collapsible="accordion">
      <li>
        <div class="collapsible-header active blue darken-2 white-text">
          <i class="material-icons">add_circle</i>
          Recommendation {{loop.index0+1}}
        </div>
        <div class="collapsible-body">
          <h6>Patient Information:</h6>
          <div class="divider"></div>
          <div class="row">
            <div class="col s12 l3 m3">
              <span>Weight: {{t['weight']}} kg</span>
            </div>
            <div class="col s12 l3 m3">
              <span>Height: {{t['height']}} cm</span>
            </div>
            <div class="col s12 l3 m3">
              <span>Waist Circumference: {{t['wc']}} cm</span>
            </div>
            <div class="col s12 l3 m3">
              <span>Hip Circumference: {{t['hc']}} cm</span>
            </div>
          </div>
        </div>
      </li>
    </ul>
  </div>
</div>
<div class="collapsible-header active white-text blue

```

```

        <div class="col s12 l3 m3">
        <span>Age: {{t['age']}}</span>
        <br>
        <span>Gender: {{t['gender']}}</span>
    </div>

    <div class="col s12 l3 m3">
    <span>HBA1C: {{t['HBA1C']}}%</span>
    <br>
    <span>FBS: {{t['FBS']}} mg/dL</span>
    </div>
</div>

<div>
<h6>Similarity Score: {{res[loop.index0
]1}}</h6>
</div>
<br>

<div class="row">
    <h6>Diagnosis Result:</h6>
    <div class="divider"></div>
    <div class="col s12 l12 m12">
        <span><b>Complain Notes</b>: {{t[
CN']}}</span><br><br>
        <span><b>History Notes</b>: {{t[
HN']}}</span><br><br>
        <span><b>Diagnosis Notes</b>:
        {{t['DN']}}</span><br><br>
        <span><b>Diabetes Diagnosis</
b>: {{t['DIAG_DM']}}</
span><br><br>
        <span><b>NON-DM Diagnosis
</b>: {{t['DIAG_NONDM
']}}</span><br><br>
    </div>
</div>

    <div class="row">
    <h6>Treatment Result:</h6>
    <div class="divider"></div>
    <div class="col s12 l12 m12">
    <span><b>Treatment Plan</b>: {{t[
TPN']}}</span>
    </div>
</div>

</div>

<a href="https://www.diabetes.ca/diabetes-
and-you/healthy-living-resources/
diet-nutrition/basic-meal-planning"
target="_blank" class="waves-effect
waves-light red btn">Diet and
Exercises Management Plan</a>
</li>
</ul>
</div>

{% endfor %}
</div>

{% endblock %}

search.html
{% extends 'layout.html' %}
{% block body %}

<div class="row">
    <form class="col s12 m3 l3 animated slideInLeft"
    method="POST" action="{{ url_for('search')
}}">
    {{ s_form.csrf_token }}
    <fieldset class="z-depth-5">
<legend><h5 class="grey-text">SEARCH</h5></
legend>
    <div class="row">
        <div class="input-field col s12 m12 l12">
        {{ s_form.search_age_group }}
        <label>Age Group</label>
        </div>

        <div class="input-field col s12 m12 l12">
        {{ s_form.search_dm_type }}
        <label>DM TYPE(ICD-10)</label>
        </div>

        <div class="input-field col s12 m12 l12">
        {{ s_form.search_sex }}
        <label>SEX</label>
        </div>

        <div class="input-field col s12 m12 l12">
        </div>
    </div>
    <div class="right">
    <button class="btn waves-effect waves-light" type="
submit" name="action">Submit
    <i class="material-icons right">send</i>
    </button>
</div>
</div>
</div>

</fieldset>

</form>

<div class="col s12 m9 l9">
<div class="card-panel z-depth-5 animated
slideInRight">
<table id="myTable" class="bordered responsive
-table striped">
<thead>
<tr>
<th>Weight</th>
<th>Height</th>
<th>WC</th>
<th>HC</th>
<th>hb1AC</th>
<th>FBS</th>
<th>Age</th>
<th>Sex</th>
<th>DM Diagnosis</th>
<th>NON - DM Diagnosis</th>
</tr>
</thead>
<tbody>
{% if search_df is defined%}
{% for i,value in search_df.iterrows() %}
<tr>
<td style="padding-left:25px">{{value['
weight']}}</td>
<td style="padding-left:25px">{{value['
height']}}</td>
<td style="padding-left:20px">{{value['wc
']}}</td>
<td align="center">{{value['hc']}}</td>
<td style="padding-left:25px;align="center"
">{{value['HBA1C']}}%</td>
<td align="center">{{value['FBS']}}</td>
<td style="padding-left:25px" align="center"
">{{value['age']}}</td>
<td style="padding-left:30px" align="center"
">{{value['gender']}}</td>
<td>{{value['DIAG_DM']}}</td>
<td>{{value['DIAG_NONDM']}}</td>
</tr>
</tbody>
</table>
</div>
</div>

```

```

        </tr>
        {% endfor %}
    {% endif %}
</tbody>
</table>
</div>
</div>
    {% endblock %}

dataset_extract.sql
//GET DATASET FOR RECOMMENDER SYSTEM
SELECT obs.person_id, obs.concept_id, obs.encounter_id, obs.
value_numeric, person.gender, person.birthdate, obs.
value_text, obs.date_created
FROM obs
INNER JOIN
( SELECT concept_id, encounter_id, person_id, value_text
FROM obs
WHERE concept_id = 44 AND (
value_text LIKE '%metformin%' OR
value_text LIKE '%insulin%' OR
value_text LIKE '%Glibenclamide%' OR
value_text LIKE '%glipizide%' OR
value_text LIKE '%Rosiglitazone%' OR
value_text LIKE '%glibenclamide%' OR
value_text LIKE '%glimpiridee%' OR
value_text LIKE '%acarbose%'
)
OR ( value_text LIKE '%diabetes%' AND concept_id = 84)
) sub
ON obs.person_id = sub.person_id AND obs.encounter_id =
sub.encounter_id
INNER JOIN person ON person.person_id = obs.person_id
WHERE obs.concept_id = 21 OR obs.concept_id = 23 OR obs.
concept_id = 26 OR obs.concept_id = 27
OR obs.concept_id = 44 OR obs.concept_id = 43 OR obs.
concept_id = 84 OR obs.concept_id = 40
OR obs.concept_id = 41 ORDER BY obs.person_id

// matching
SELECT DISTINCT obs.person_id, obs.encounter_id, person.
birthdate, person.gender
FROM obs
INNER JOIN

Dataset_Generator.py
# coding: utf-8
# # IMPORT PANDAS AND ENTER HC Name
#
# In[1]:
import pandas as pd
# In[2]:
def create_dataset ( file ):
# # LOAD CSV TO DATAFRAME
# In[3]:
dist_id = pd.read_csv(file+"/" +file+"_dist_pid_eid.csv")
data = pd.read_csv(file+"/" +file+"_raw_ds.csv")
# # CREATE DATASET
# In[4]:
columns = ['person_id', 'encounter_id', 'weight', 'height',
'wc', 'hc', 'isMale', 'isFemale', 'birthdate', 'age',
'CN', 'HN', 'DN', 'DIAG_DM', 'DIAG_NONDM', 'TPN']
n_df = pd.DataFrame(columns=columns)
for index_dist_id, row_dist_id in dist_id.iterrows():
# print index_dist_id, row_dist_id ['person_id'],
row_dist_id ['encounter_id']
n_df.set_value(index_dist_id, 'person_id', row_dist_id
['person_id'])
n_df.set_value(index_dist_id, 'encounter_id',
row_dist_id ['encounter_id'])
n_df.set_value(index_dist_id, 'birthdate', row_dist_id
['birthdate'])

( SELECT concept_id, encounter_id, person_id, value_text
FROM obs
WHERE concept_id = 44 AND (
value_text LIKE '%metformin%' OR
value_text LIKE '%insulin%' OR
value_text LIKE '%Glibenclamide%' OR
value_text LIKE '%glipizide%' OR
value_text LIKE '%Rosiglitazone%' OR
value_text LIKE '%glibenclamide%' OR
value_text LIKE '%glimpiridee%' OR
value_text LIKE '%acarbose%'
)
OR ( value_text LIKE '%diabetes%' AND concept_id = 84)
) sub
ON obs.person_id = sub.person_id AND obs.encounter_id =
sub.encounter_id
INNER JOIN person ON person.person_id = obs.person_id
WHERE obs.concept_id = 21 OR obs.concept_id = 23 OR obs.
concept_id = 26 OR obs.concept_id = 27
OR obs.concept_id = 44 OR obs.concept_id = 43 OR obs.
concept_id = 84 OR obs.concept_id = 40
OR obs.concept_id = 41 ORDER BY obs.person_id

//GET UNIQUE IDS with PAST HISTORY OR FAMILY
HISTORY
SELECT DISTINCT obs.person_id, obs.concept_id, person.
gender, person.birthdate, obs.concept_id, obs.value_coded,
obs.voided
FROM obs
INNER JOIN person ON obs.person_id = person.person_id
WHERE concept_id = 465 AND (value_coded = 1 OR
value_coded = 2 ) AND obs.voided = 0 ORDER BY obs
.person_id
;

if row_dist_id ['gender'] == 'M':
n_df.set_value(index_dist_id, 'isMale', 1)
n_df.set_value(index_dist_id, 'isFemale', 0)
else:
n_df.set_value(index_dist_id, 'isMale', 0)
n_df.set_value(index_dist_id, 'isFemale', 1)

for index_data, row_data in data.iterrows():
if row_dist_id ['person_id'] == row_data['
person_id'] and row_dist_id ['encounter_id']
== row_data['encounter_id']:
if row_data['concept_id'] == 21:
n_df.set_value(index_dist_id, 'weight',
row_data['value_text'])
elif row_data['concept_id'] == 23:
n_df.set_value(index_dist_id, 'height',
row_data['value_text'])
elif row_data['concept_id'] == 26:
n_df.set_value(index_dist_id, 'wc',
row_data['value_text'])
elif row_data['concept_id'] == 27:
n_df.set_value(index_dist_id, 'hc',
row_data['value_text'])
elif row_data['concept_id'] == 40:
n_df.set_value(index_dist_id, 'CN',
row_data['value_text'])
elif row_data['concept_id'] == 41:
n_df.set_value(index_dist_id, 'HN',
row_data['value_text'])
elif row_data['concept_id'] == 43:
n_df.set_value(index_dist_id, 'DN',
row_data['value_text'])
elif row_data['concept_id'] == 44:
n_df.set_value(index_dist_id, 'TPN',
row_data['value_text'])
elif row_data['concept_id'] == 84:
if "diabetes" in row_data['value_text'].
lower():
n_df.set_value(index_dist_id,
DIAG_DM, row_data['value_text'])
else:
n_df.set_value(index_dist_id,
DIAG_NONDM, row_data['
value_text'])

```

```

n_df['birthdate'] = pd.to_datetime(n_df['birthdate'],
    errors='coerce')
n_df['birthdate'].dt.strftime('%m/%d/%Y')

# In[8]:

n_df['age'] = pd.to_datetime(n_df['birthdate'])
now = dt.date.today()
n_df['age'] = ((now - n_df['birthdate']/365).dt.days

# In[9]:

n_df

# In[10]:

n_df.to_csv( file + "/" + file + "_dataset.csv", index=False)

file = "g"
while file != 'exit':
    file = raw_input("ENTER FILE:")
    create_dataset( file )

path =r'A_DATASET' # use your path
allFiles = glob.glob(path + "/*.csv")
frame = pd.DataFrame()
list_ = []
for file_ in allFiles:
    df = pd.read_csv(file_, index_col=None, header=0)
    list_.append(df)
frame = pd.concat(list_)

# # Save DF as CSV
#

# In[6]:

#index = False to avoid index column in csv file
frame.to_csv(" final_dataset .csv", index=False)

$(document).ready(function() {
    $('select').material_select();
});

$(document).ready(function(){
    $('ul.tabs').tabs();
});

h3,h5,h6 {
    text-transform: uppercase;
    font-weight: 900;

    @media (max-width: 400px) {
        font-size: 22px;
    }
    @media (min-width: 1200px) {
        font-size: 75px;
    }
}

i.icon-grey {
    color: grey;
}

span{
    font-weight: light
}

/*@media only screen and (min-width: 1200px) {
    tbody {
        display: block;
        height: 500px;
        overflow: auto;
    }
    thead, tbody tr {
        display: table;

```

```

# In[5]:

n_df

# In[6]:

n_df['weight'].fillna(0, inplace=True)
n_df['height'].fillna(0, inplace=True)
n_df['wc'].fillna(0, inplace=True)
n_df['hc'].fillna(0, inplace=True)

# In[7]:

import datetime as dt

Dataset_Concat.py

# coding: utf-8

# # The glob module finds all the pathnames matching a
# specified pattern according to the rules used by the
# Unix shell,

# In[1]:

import pandas as pd
import glob

# # append all df to a list. concat list to create single DF

# In[2]:

init.js

(function($){
    $(function(){

        $('button-collapse').sideNav();

    }); // end of document ready
})(jQuery); // end of jQuery name space

style.css

/* Custom Stylesheet */
/**
 * Use this file to override Materialize files so you can
 * update
 * the core Materialize files in the future
 *
 * Made By MaterializeCSS.com
 */

.icon-block {
    padding: 0 15px;
}

.icon-block .material-icons {
    font-size: inherit;
}

#sp_name{
    text-transform: uppercase;

    margin: 0 0 1rem;

    @media (max-width: 400px) {
        font-size: 22px;
    }
    @media (min-width: 1200px) {
        font-size: 75px;
    }
}

```



```

width:100%;
table-layout:fixed;
}
thead {
width: calc( 100% - 1em )
}
table {
width:100%;
}

graphs.js
try{
new Chart(document.getElementById("line-chart"), {
type: 'line',
data: {
labels: [2015,2016,2040],
datasets: [{
data: [3.5,6,12],
label: "Philippines",
borderColor: "#3e95cd",
fill : false
}
}
}
}
}

}*/
/*for table
*/
};
options: {
title: {
display: true,
text: 'Diabetes Cases (in millions)'}
}
});
catch(err){
console.log("Error")
}
}

```

## **XI. Acknowledgement**

I would like to thank God first for everything. Without you, none of these would be possible. I thank you for helping me achieve my goals all these years.

I would like to thank my parents for their continuous support in doing my SP. To my father for giving me lots of advices during my college life and to keep on fighting. To my mom for not giving up on me even at my lowest point. I thank you both for everything since my 1st year in UP Mindanao until today here at UP Manila for all the sacrifices that you have done for me.

Thank you to my older sister Roxy for accompanying me. Thank you for the endless support and especially cooking our dinner. Thank you to my younger sister, Janee for calling every night.

Thank you to my family for the countless support, for feeding us, sa paghatid at sundo sa amin every weekend. Thank you sa lahat.

To my two best friends in the world, Ryco and Renzon para sa kalokohan sa group chat. GG = gising na gising tuloy ako kapag gumagawa ng SP.

Thank you to Alyssa Gail for being supportive throughout the development of my SP.

Thank you to my adviser, Ma'am Sheila Magboo for the the many advices and consultations to complete my SP. Thank you for introducing CHITS, if it wasn't for you, I would have to go find data sets on my own. I thank you for the learning opportunity with CHITS and to my new learnings in developing my SP. Thank you for giving me the opportunity to show my skills. Thank for the motivation finish my SP. Thank you for believing in me during the proposal and defense.

Special mention to Sir Marvin Ignacio even though I am not your advisee, I was still able to approach you for consultations on machine learning and I was able to improve my SP because of your help. Thank you sir Marvin. Learning

Thank you to Albert Francis E. Domingo,MD for the countless advice an consultations about Diabetes. I gained lots of knowledge in developing my SP because of the information you provided.

I would like to thank the CHITS personnel for their help in developing my SP. To sir Art Ongkeko for managing my CHITS account. To Sir Elvin, for providing the datasets. To sir Jojo, for the many times I visited Telehealth and asked a lot of questions regarding about the CHITS system.

Thank you to my cousin, MM Yunque for giving me the oppurtunity to be an intern at PLDT Global to learn about Big Data Analytics and to Sir Jolo Balbin for being my mentor

To the Dota Boys, boss Janssen, boss Enzo at boss Marron, I thank you for the memories for four years. Thank you for bonding natin sa PGH court, SSWC, Pedro Gil computer shop, sa Radiant side, sa Dire Side, sa Rosh Pit, sa Mid Lane at sa UPM. Thank you sa lahat ng support both ingame and in the real world.

Thank you to the three UP families I have been with, First, to my UPMin friends, thank you for my first year in UP. We made so much memories for a year.

Next sa 2013 UPM Computer Science batch. Thank you for the memories we had for years. Maraming salamat din po sa tulong niyo sa mga classes natin. Special mention po kila Jed, RS, Reniel, Neil, Ivan, Kier, Zoleta, Francis, Jeffrey, Andrei, Jerome, Sir Berwin. Salamat sa mga kalokohan niyo.

Last sa 2014 UPM Computer Science batch, Thank you for everything.. Thank you for sa support sa mga MP at projects. Special Mention sa ETL group(Benge, Kenneth, Kharl, Red, Delwyn, Sigfreed), sa VR Group(Red, Michael, Kenneth) at sa Braderhood(Ivan, Neil, Zoleta, Michael, Sigfreed, Kenneth). Thank you sa effort, teamwork at support sa mga projects natin.

Thank you to ma'am Eden sa pag asikaso ng mga kailangan gawin ever since paglipat ko. Thank you to all my ComSci profs who have given their best to teach. Without you, I wouldn't be where I am today.

Thank you to my laptop, ASUS X550DP AMD A10, AMD Radeon 8670M + 8650G, 8GB DDR3 RAM, sa dalawang beses na pagpalit ng battery, dalawang motherboard, isang charger at isang HDD pero nagawa nating tapusin ang SP.