SP-24-X-DATA MINING

DISEASE PREDICTION SYSTEM: CANCER AND DIABETES

FINAL REPORT

CS 4850 - Section 01 – Spring 2024

Professor Sharon Perry

Website: https://sp-24-x-data-mining.github.io/

Github link: https://github.com/sp-24-x-data-mining/sp-24-x-data-mining.github.io

March 2024

Number of lines in source code:	**423**
Number of Project Components:	**14**

Team Members:

Name	Role	Cell Number	Email
Alexus Glass	Developer	4049601998	lushgaloreinc2@gmail.com
Samuel Futral	Documentation	7067162064	samuelfutral@gmail.com
Kokou Adje	Developer	4703583315	adjelaime@gmail.com
Keegan Begley	Lead	6783667862	kbegley@students.kennesaw.edu
Sharon Perry	Project Advisor	7703293895	sperry46@kennesaw.edu

Contents

1.0 Abstract

The data mining project is a web application designed with a machine learning-powered platform implemented to help with cancer and diabetics prediction. This project aims to assist individuals and healthcare professionals to predict the probability of cancer and diabetics using data based on health information. The project is a combination of machine learning models and algorithms, data mining, and web development using Python programming language.

The objective of this project is to design machine learning models capable of predicting cancer and diabetes using patient's health indicator information provided such as age, blood pressure, breast density, glucose level, etc. Secondly, develop easy user-friendly interface web application to gather input data from the user, and display the result in a real time manner. Finally, implement a strong data mining and processing system to guarantee data and prediction accuracy.

The application begins with the login into the system and process input user data and then preprocesses this data to be analyzed for machine learning. Logistic Regression generates cancer and diabetics predictions using trained classification models and the result is displayed to the user.

This project aims to provide a good and rapid predictive tool which is a complement for traditional cancer and diabetics diagnostic methods. In addition, the scope of this project is for health care providers, healthcare professionals, and serves as educational resources for people who want to learn more about the relationship between healthcare, intelligence artificial, data mining in the prediction of cancer and diabetics.

1.2 Project Goals

First, the objective of this project is to design machine learning models capable of predicting cancer and diabetes using patient's health indicator information provided such as age, blood pressure, breast density, glucose level, etc. Secondly, develop easy user-friendly interface web application to gather input data from the user, and display the result in a real time manner. Finally, implement a strong data mining and processing to guarantee data and prediction accuracy.

The goal of this document is to define the goal of SRS documentation and specify the nonfunctional and functional requirements for this project, and then present some user interface (UI), data flow diagram.

1.3 Definitions and Acronyms

Element	Description
UI	User Interface
OS	Operating systems
GPU	Graphics Processing Unit
GB	Gigabytes
DRAM	Dynamic Random-Access Memory
WCAG	Web Content Accessibility Guidelines
HIPAA	Health Insurance Portability and Accountability Act

BMI	Body Mass Index
HTTPS	Hypertext Transfer Protocol Secure
Data mining	Extraction of different types (pattern, correlation, insight) of data from large dataset
Machine learning	Part of artificial Intelligence to develop models which can learn from extracted data to make accurate prediction

2.0 Analysis and requirements

The implementation of this project was impacted by some design constraints such as environment, user characteristic, and system requirements.

2.1 Environment

Environmental constraints or resource constraints regroup hardware condition and limitations, software dependencies, and OS compatibility. The system needs to be compatible with laptops, desktops, current and previous OS. Another constraint is software dependencies where the system must be compatible with modern browsers as Google Chrome, Microsoft Edge, Mozilla Firefox, Opera. The software libraries (example of Python libraries used in machine learning) must also have a compatibility with the chosen IDE (PyCharm).

The table below resumes some limitations for the environment.

CPU	RAM(GB)	OS(GB)	External Disk	OS type	Browser
10	128	200	500	Windows	Google Chrome
14	1000	200	1000	Linus	Microsoft Edge

2.2 User Characteristics

User characteristic is the capability of user to interact friendly with the system. For this project, the application should provide accommodation for the users according to the degree of their technical knowledge. The navigation in the system should be easy.

The application should be accessible for the user with disabilities in compliance with accessibility standards (WCAG). For accessibility requirements, the application should consider text size, navigation with keyboard, screen readability, and color contrast.

The system must be able to ensure the input data accuracy by controlling the user input and handling errors and validating the input.

2.3 System

The implementation of this application has some system constraints such as: integration with other systems which are externs, performance, security, privacy. The application must use machine learning to predict cancer or diabetics with the possibility to extract data from public databases. This integration with the external system (e.g. connection with public database) should be possible although data format compatibility complexity. This project involves big data

manipulation, and the system should be able to handle those data processing. For security purposes, the application should comply with data security and privacy regulations (HIPAA) by encrypting protocols and securing the transmission methods. The machine learning models chosen to train to predict the accurate result should endure serious testing and validation processes.

2.4 Functional Requirements

Functional requirements, also called behavioral requirements, are the requirements of the system's external observable behavior and focus on specific features, interactions, input, output. For this project, the functional requirements are its functionalities that help users to input data, process those data with the model of machine learning, and then make accurate prediction.

2.4.1 Authentication

To use the system, the users need to provide login information to have access to either cancer prediction page or diabetics prediction page, and then display the result. After a successful login, the cancer page or diabetic page is displayed according to the choice of the user.

2.4.2 Enter Cancer Information

For diabetics prediction page, the user needs to input the following information: glucose level, blood pressure, BMI, and age, and then display the result by clicking on the predict button.

Cancer prediction information

Input	Description
Age	Significant factor for risk of cancer. Some cancers are dangerous in a specific age category.
Tumor size	Measured in cm, it is key factor for cancer diagnostic
Menopause	Menopause status has a huge impact of risk of cancer, example breast cancer
Tumor type	No cancer, invasive cancer, and on-invasive cancer and impact the tumor behavior
Breast Density	It is a risk for cancer detection and is either low density or high density

2.4.3 Enter Diabetic information

On the cancer page, the user must fill the following information: age, tumor size, menopause, Tumor type, breast density.

Input	Description
Glucose Level	The key factor in diabetic diagnostic is the glucose level and high level is a red flag
Blood pressure	High blood pression means high risk of diabetic
BMI	It is a value obtained from height and weight. Higher BMI means risk of diabetics
Age	Significant factor for risk of diabetic.

2.4.4 Data processing

Data processing is the generation of a result through data collection, data preprocessing, model training, and the result presentation to the user.

2.4.4.1 Data Collection

The purpose of data collection is to gather information from the user about cancer and diabetes. The user put in the form designed for receive information such as age, tumor size, menopause, tumor type, and breast density for cancer and glucose level, blood pressure, BMI, age for diabetic.

2.4.4.2 Data Preprocessing

This step requires data cleaning, data transforming, data preparing to make sure of its relevance, quality, and most compatibility with machine learning algorithm. For this step, the following are handling: missing value, outlier detection, feature scaling, encoding categorical variables, data splitting. For data preprocessing tasks, some Python libraries such as NumPy, Pandas, Scikit-learn are used. Once data is clean and compatible, then follow the steps of the model training.

2.4.4.3 Model Training

The cleaned data from data processing are formatted for the training. To train the models, Logistic Regression algorithm is used on the training data (X_train, y_train) and learn patterns. During the training processing, the models are fit to the training data (from dataset) to learn from the input and the target labels corresponding.

2.4.4.4 Display Result

The result is displayed to the user the outcome based to its input data. Although there are many ways to display the results such as chart and graph, color coding, interactive visual, we chose for our project to give just feedback to the user. For that reason, in the case of cancer, the result will be the cancer is benign or malignant and diabetic or non-diabetic for diabetic prediction.

4.0 Non-Functional Requirements

4.1 Security

The security of data in this project needs to be guaranteed. The data must be encrypted because it is a healthcare dataset and during the transmission of data, HTTPS protocols will be used to keep the confidentiality. The access to the software is secured by authentication (login and password); also, access to the database requires a role-based access control (RBAC). For data privacy, the application should comply with data security and privacy regulations (HIPAA) by encrypting protocols and securing the transmission methods.

4.2 Capacity

The capacity requirement for this project consists of handling the load of information, data, and resources dedicated to the system to meet the performance desired. The volume of data, the hardware, the storage capacity, and the database capacity need to be evaluated. For example, we need to evaluate the amount of information or data which will be processed by the system without impact on its performance.

4.3 Usability

The usability requirement involves the design of user interface and how ease to interact with the system. The user interface must be user-friendly, intuitive and must be available and usable on different OS. The application should make the navigation between pages clear and easy.

4.4 Other

The system should be extensible and reliable. The system must allow the implementation of or integration of new features, new functionality without decreasing either its performance or capacity. The system must implement fault tolerance by handling potential errors during the system utilization.

5.0 Design constraints

The design conception and implementation of this project was impacted by some design constraints we will describe in the following section.

5.1 User authentication

The use of the system requires authentication where the user must provide their login information. The user authentication is implemented in Python with Flask-login extension which secures the data and use session.

5.2 Data processing

Data processing is a key aspect of software design for software projects which handle large amounts of data, in our case, the data from diverse public health databases. For our web application, data processing involves analysis of user's input and then prepare them for the machine learning models for classification tasks. Panda, a python extension is used for data preprocessing and handle missing values from the extraction to prepare accurate and usable data for machine learning models.

5.3 Use of machine learning

In this step, scikit-learn library is used to demonstrate how the models trained for classification tasks are incorporated in the application. This library also implements the functions for the prediction tasks for cancer and diabetes.

5.4 User interfaces

The conception of the user interfaces needs to comply with the WCAG guidelines for usability and accessibility.

6.0 System architecture

This part of SDD document gives an overview of the system architecture and the description is detailed in the following session for the web application.

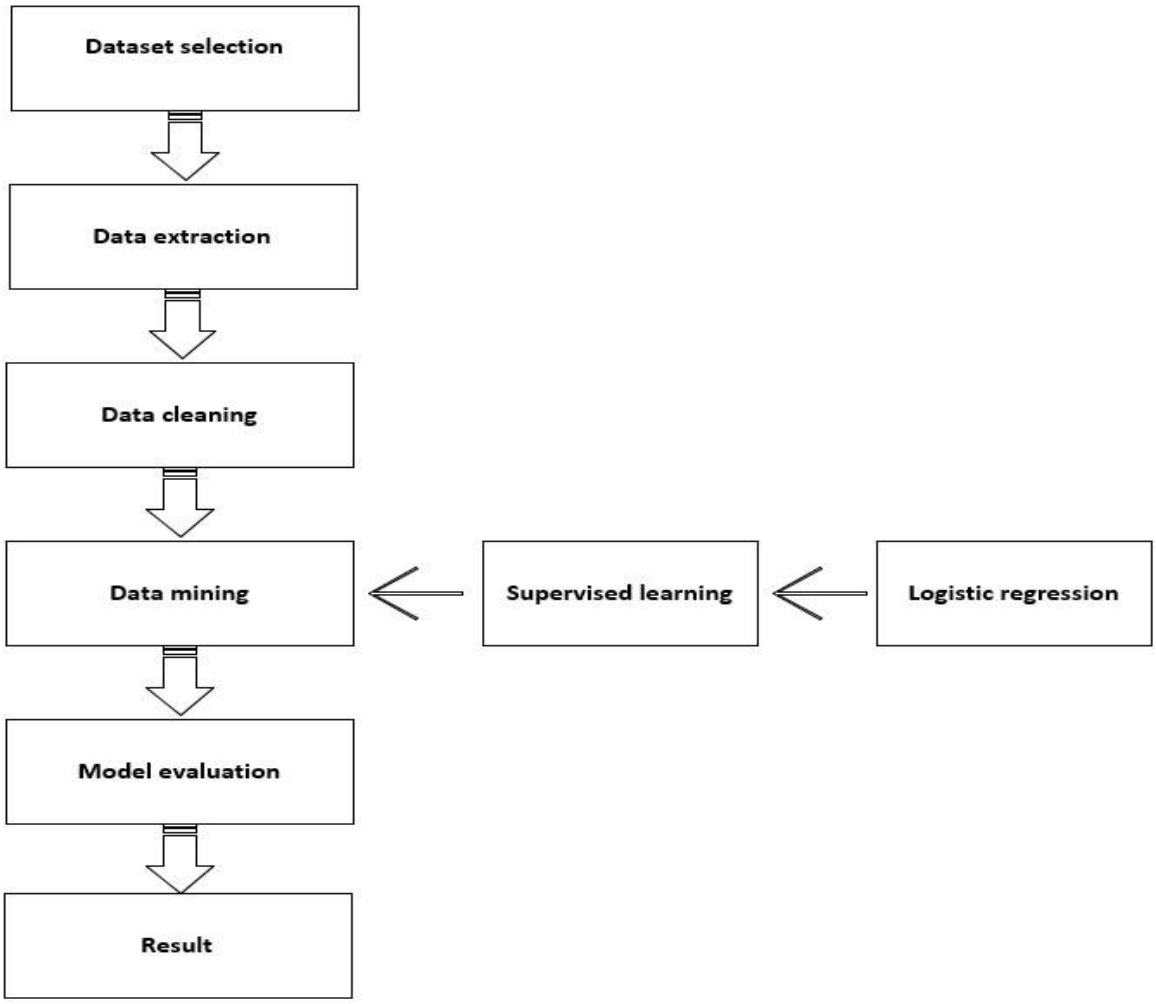
6.1 Front-end elements

Front-end elements allow interaction between the client side and the user. These elements are : HTML (HyperText Markup Language), CSS Stylesheets, JavaScript.

6.2 Back-end element

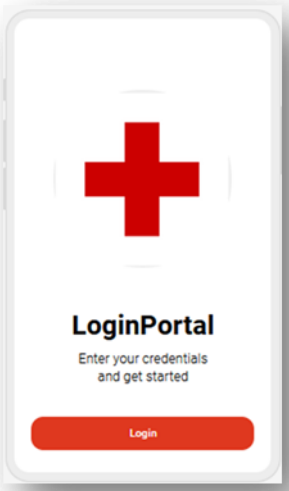
Back-end elements allow interaction between the client side and the server side. The elements used are the following: Flask Framework (provide URL mapping, request handling, routing), Machine Learning Models (Logistic Regression in our case for prediction and classification tasks, scikit-learn to train models), database connectivity (SQLAlchemy ORM connector is used to facilitate the communication between public databases and the software)

6.3 Architecture Diagram

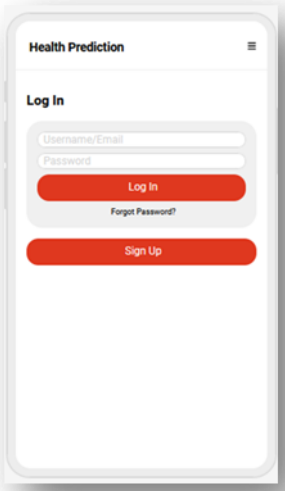


6.4 Design drawings

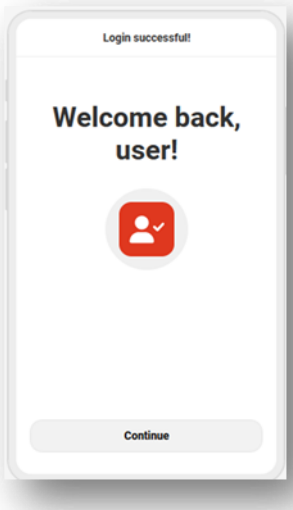
The following are the descriptions of how the software works.



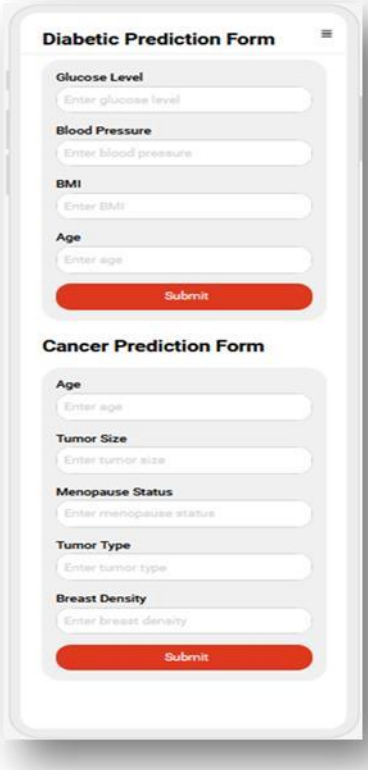
Login portal will be first page that user interacts with, will bring them to the log in/sign up page.

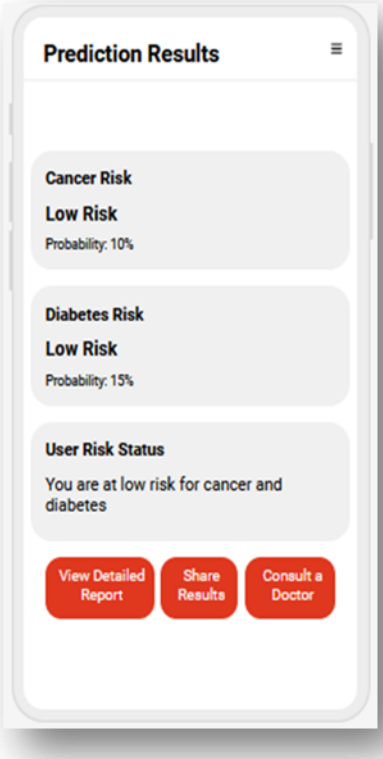


User will input account info, or if it is their first time they will have the option to sign up.



Log in success screen if the user logs in successfully.





Prediction forms for the user to input their data to get their me analysis performed. One for car and one for diabetes.	The results page will show after the forms are submitted and compared to machine learning model that was fed research and studies.

7.0 Version control description

In our machine learning project, we utilized Git as the version control system, it was chosen for its robustness and flexibility, which is ideal for managing the iterative changes common in machine learning development. The project was organized within a single Git repository, which included all related files such as datasets, model scripts, and documentation, facilitating easy access and management. We adopted a Feature Branch workflow, where the main branch stored stable versions of the project, and development of new features was carried out in separate branches, such as feature-logistic-regression-model and feature-data-preprocessing. These were merged back into the main branch upon completion and testing.

Commit practices were strictly enforced; we required clear and simple, yet descriptive messages and encouraged small changes to simplify both of the troubleshooting and code reviews. Each pull request underwent peer review by at least one other team member and had to pass automated tests using GitHub Actions before merging. By doing this, we ensured code quality and functionality was up to our project standards. We used semantic versioning to track different stages of our project, with tags like v1.0 for the first deployment and then the further tags for newer additions.

The use and integration of tools like GitHub Issues and actions for task tracking and the automated testing helped make things easier and more streamlined. Challenges such as managing the dependencies and ensuring that the consistencys were addressed by using a docker to contain and isolate the development environment and conducting thorough reviews of dependency changes. Security measures included securing repository access by managing permissions via GitHub teams to restrict merging abilities.

Throughout this project, we learned the important role of continuous integration in maintaining code quality. Our very strict and rigorous code reviews and automated testing were instrumental in minimizing bugs and enhancing the reliability of our machine learning model. For our future projects, we intend to explore and use more advanced branching strategies to better manage our developmental work, and thus improving our version control practices and final project outcomes.

8.0 Challenges, assumptions, and risk assessments

The implementation of this project involves some challenges, assumptions, and risk assessments.

8.1 Challenges

The first challenge is data availability and quality. Medical data access can be difficult because of privacy and regulatory exigences and restriction. For example, we need to adhere to HIPAA regulations. It is also difficult to adapt the data from datasets to our format. The second challenge is the machine learning models accuracy; to have a good prediction, it is crucial to have accurate data to train those models. In addition, there is user interface and experience challenge we face because the software must be accessible for all kinds of users including users with disabilities and healthcare professionals.

8.2 Assumptions

There are divers assumptions for this cancer and diabetic prediction web application. One assumption is about data access where we assume that data is available for the training of machine learning models. We also assume that healthcare providers have the technologies required for this application. For example, we assume that the users (healthcare providers and patients) have computers to use this application. We assume also that we have regulatory approval for meeting the necessary regulatory requirements to use this software in healthcare.

9.0 Test plan and Test report

The Test Plan will cover testing of the requirements and the app. Functional requirements will be listed along with a column to capture whether the test passed or failed.

Sample Test Report is included below:

SAMPLE TEST REPORT	Pass	Fail
Authentication	yes	
Enter cancer Information	yes	
Enter diabetic Information	yes	
Data processing	yes	
Data Preprocessing,	yes	
Model Training,	yes	
Display Result	yes	

10.0 Analysis & Development

The use of machine learning in healthcare has created brand new options for early disease detection, which is incredibly important for the effective treatment and management of certain ailments and diseases. This project was created with the goal of developing a predictive model that will be able to estimate and predict the likelihood of diabetes and cancer, based on the provided data of the user or patient. The conditions that were chosen were due to their high prevalence and impact on public health across the globe. The early detection can be used to improve patient outcomes and focus more on preventative healthcare.

The foundation of any ML model is data. For this project, we sourced data from two established databases: the Pima Indians Diabetes Database for diabetes prediction and the UCI Breast Cancer Wisconsin (Diagnostic) dataset for cancer prediction. Both datasets are well-regarded in the health informatics community for their comprehensive data points and historical use in similar studies.

These datasets allow us to gather very important information on these two deadly diseases as well as the important symptoms and their correlating disease percentages. The datasets are vast and should contain enough data to build the initial learning model. For the future, it will help to include more datasets and health data to improve our model and increase the likelihood of the detection of cancer and diabetes.

The input data that we need from the users is based off the many health factors that are correlated with these diseases. For diabetes there are things such as glucose level and blood pressure, while

for cancer, there are questions such as age, and tumor size. These are the most important factors in dealing with cancer and diabetes. The study and datasets that were used reflect these with the data that they contain and are exactly why we chose them in the first place. The user or doctor of our application will be able to put in their stats and will hopefully be able to get their prediction results accurately and quickly.

Upon obtaining the datasets, we undertook a rigorous data cleaning process. This involved handling missing values, removing duplicates, and standardizing features. We encoded categorical variables to numerical values to facilitate processing by our machine learning algorithms. Our preprocessing efforts also included normalizing the data to ensure that the model's performance was not biased towards variables with higher magnitude.

To create our model, we decided to go with python as it is the preferred language to use for data science and machine learning on this scale. Its abilities are perfect for this project, and the language itself will be able to do everything that we need from it. Our team is well versed in the usage of python and are extremely familiar with it. The code will be created and complied using the Microsoft visual studio IDE, as it is very capable of working with python and reflects our project's needs.

For the development of our model, the most important step was to identify the most relevant features that contribute to the outcomes of diabetes and cancer. After this was done, we split the data into two different sets. There is a training set as well as the testing set. For this project, we maintained a 80-20 split, which means that we trained our model off of 80 percent of the data sets, and then tested the models with the remaining 20 percent. By doing this, we can create a very robust learning model as well as test out its effectiveness in its job.

For the training of our machine learning model, we trained separate logistic regression models for each disease, both cancer and diabetes. Training involved adjusting the weights of the features through an iterative process to minimize prediction errors. We utilized regularization techniques to prevent overfitting, ensuring that our models generalized well to new data. These are very important steps that were taken to ensure the quality and effectiveness of the model. It will allow for better results when it is used.

Once the model was created and developed, it needed to be evaluated on its performance and effectiveness. To do this, there were different metrics that were used to evaluate our machine learning model. The most important is the accuracy of the predictions. By using different metrics in the evaluation phase, we were able to find new insights and information into our models ability to correctly predict the diseases, as well as its ability to differentiate between the different classes.

While the model on its own may be impressive, there are other important parts of this project, including, but not limited to, the user interface. The user interface is incredibly important as it allows the user to access the model and use it for predictive healthcare needs. For our project, we wanted to create a very simple user interface that would be very streamlined and easy to understand for whomever the user may be.

The application for the machine learning model will contain a login in portal which will require a username and password. This project is intended to be used by the patient and the doctors, while creating this, and in both cases, this meant the possibility of needing security for important health

care data. By including security and a login stage, we can safely manage any healthcare data without any issues or negligence.

Upon logging into the application, there is a very simple form that will be displayed to the user. Here is where the relevant information regarding the patients' health will be inputted. The forms contain the many different health factors that were chosen for this project. There will be cancer forms, as well as a form for diabetes. In both cases, the health information will be inputted to their respective machine learning model that we have created and tested.

After the necessary information is inputted, it can be submitted. This will then bring the user to the results page. The results page displays the percentages and likelihood of the user having either cancer or diabetes. The machine learning health prediction model will run the inputs and display the outputs in this section of the application.

As spoken about earlier, there are very important security measures that must be taken when dealing with a patient's healthcare data. To adhere to the high standards that exist in the healthcare industry, we employed the use of data encryption to protect the user's data, so that nobody will be able to access it. The only important part is when the data is in transit. For our application, there will not be any storage abilities, so we do not need to worry about the longterm storage of patient health data. Each use of the application will be a one-time deal, and will not store any information, so the inputs will ne to be reinputted to the application every time for the patient. We deemed this as being okay as the patients will have new data about their health every time they go in for a new appointment.

11.0 Summary

Cancer and Diabetic Prediction Web Application project provides a strong tool for healthcare providers and professionals to predict and detect early cancer and diabetic from patients. The prediction is done with artificial intelligence by machine learning models. This project has many requirements and designs. A version control is integrated to control the version of the application. A test plan and report are written for the project. This project is implemented without some challenges and assumptions. We are grateful for the hard work of the team for the time dedicated

APPENDIX – Project Plan

1. Platform:

For this project, we will use a hardware and software platform. The right hardware needs to be chosen to properly run the machine learning program and must impact the quality and the performance of the models. The hardware requirements are the following for a computer to operate this application:

- Adequate processor for the execution of instructions.
- Good storage and memory to store a large amount of data from the processor.
- Graphics Processing Units (GPUs) to handle graphical data.
- For future utilization, high-performance computing resources like cloud infrastructure can be used for the software.

For the software infrastructure, the followings are required:

- Python: is important for the implementation of this project.

Large Language Model (LLM): a kind of AI trained to recognize and generate text.
Web frameworks: Flask, Django which offer some features for the application.

2. **Collaboration Tools:**

Communication	—	Cellphones (Call/Text) / Microsoft Teams / GroupMe
Collaboration	—	Discord (Mandatory unless another tool is authorized by Perry)
Version Control	—	GitHub

3. **Deliverables:**

a. Team/Project Selection document (Individual Assignment)
b. Weekly Activity Reports (WARs – Individual Assignment)
c. Peer Reviews (Individual Assignment)
d. Project Plan (Group Assignment)
e. Project Requirements And Design
f. Present Prototype for Peer Review (Group Assignment)
g. Website (Group Assignment)
h. Video Demo (Group Assignment)
i. C-Day Application/Submission (Group applies to C-Day but each member submits individual bonus points documentation in Individual Assignments)
j. Final Report Package (Group Assignment)
k. iOS and Android compatible mobile time travel apps

4. **Project Schedule and Task Planning (GANTT CHART)**

02/21/2024

- Research dataset
- Create graphs
- Data analysis

02/28/2024

- Data mining
- Organize Data Table
- Data analysis
- Start on System Design

03/06/2024

- System Design & architecture
- Research
- Python Coding & SQL 03/13/2024
- System Design & Architecture

- Python Coding & SQL

03/20/2024

- Documentation & Report
- Python Coding & System structure 03/27/2024
- Documentation Report 04/03/2024
- Documentation & Code implementation 04/10/2024
- Finish Final report draft 04/17/2024
- Make sure the project is precise, clean and easy to understand. 04/21/2024
- Turn in the project.

5. Meeting Schedule

The team will be meeting on Mondays and Wednesdays after class at 8pm. The length of the meetings will be an hour typically, with the option of additional time if needed. During these meetings we will discuss our current focus, allocate tasks, and update project schedules based on progress status.

6. Statement of Participation

Project ID: SP-24

STATEMENT OF PARTICIPATION:

By signing below, I Kokou Adje acknowledge that I will participate in all meetings, communications, deliverables, and other tasks necessary to complete the project. If I do not, I understand that Professor Perry will meet with me to remedy the situation.

Kokou Adje 3/7/2024

_____ _____

Team Member Date

By signing below, I Hailey Walker acknowledge that I will participate in all meetings, communications, deliverables, and other tasks necessary to complete the project. If I do not, I understand that Professor Perry will meet with me to remedy the situation.

Alexus Glass 3/7/2024

_____ _____

Team Member Date

By signing below, I William Stigall acknowledge that I will participate in all meetings, communications, deliverables and other tasks necessary to complete the project. If I do not, I understand that Professor Perry will meet with me to remedy the situation.

Samuel Futral 3/7/2024

_____ _____

Team Member Date

By signing below, I William Stigall acknowledge that I will participate in all meetings, communications, deliverables, and other tasks necessary to complete the project. If I do not, I understand that Professor Perry will meet with me to remedy the situation.

Keegan Begley 3/7/2024

_____ _____

Team Member Date

APPENDIX – Gant Chartt

Project Name: SP-24-X-Data Mining
Report Date: 04/48/2024

					Milestone #1				*Milestone #2*				*Milestone #3*		C-Day	
Deliverable	**Tasks**	**Complete%**	**Current Status Memo**	**Assigned To**	**02/21**	**02/28**	**03/06**	**03/13**	**03/20**	**03/27**	**04/03**	**04/10**	**04/17**	**04/24**	**05/01**	**05/08**
Requirements	Research dataset	0%	In progress	Alexus	10											
	Create graphs	0%		Kokou	10											
				Keegan	10											
	Data analysis	0%		Samuel	10											
Project design	Data mining	0%		Alexus		5										
	Organize Data Table	0%		Kokou		10										
	Data analysis	0%		Samuel&Keegan		5										
	Start on System Design	0%		Alexus		10										
Development	System Design & architecture	0%		Alexus & Kokou			10									
	Research	0%		Samuel&Keegan			10									
	Python Coding & SQL	0%		Alexus & Kokou			40									
Final report	Documentation & Report	0%		Samuel					8							
	Code implementation	0%		Alexus & Kokou					8	8	8					
	Finish Final report draft	0%		Samuel&Keegan								10				
	Project's review	0%		Alexus&Samuel&Kokou&Keegan								10	5			
	Turn in the project	0%		Alexus										5		
			Total work hours	**192**	**40**	**30**	**60**	**0**	**16**	**8**	**8**	**20**	**5**	**5**	**0**	**0**

* formally define how you will develop this project including source code management

Legend

Planned	192
Delayed	
Number	Work: man hours