DISEASE PREDICTION BASED ON WEB APPLICATION USING STREAMLIT

Alvon Danilo Sukardi
Computer Science Department
(School of Computer Science)
Bina Nusantara University
Jakarta, Indonesia 11480
alvon.sukardi@binus.ac.id

Gerry William Nanlohy
Computer Science Department
(School of Computer Science)
Bina Nusantara University
Jakarta, Indonesia 11480
gerry.nanlohy@binus.ac.id

Michelle Christian Bell
Computer Science Department
(School of Computer Science)
Bina Nusantara University
Jakarta, Indonesia 11480
michelle.bell@binus.ac.id

***Abstract* — The most crucial aspect of any person's existence is their health. Weekly or monthly check up of one's health is most important for the prevention and also to stay healthy. Healthcare is one of the most important aspects of human life. Nowadays, so many are not willing to go to hospital, due to work overload and negligence of their health. The doctors and nurses are putting up maximum efforts to save people's lives without even considering their own loves. There are also some villages which lack medical facilities. Nowadays, the individual is not having that much time to go for health check-up. Recently, due to covid-19, no one is willing to go to hospital for health checkup due to the fear of spreading virus. In this situation, technology plays and important role. The domain we used here is Machine Learning, it is the technique by which machines can learn from past experiences like a human being and make it efficient in future. ML is the domain which is widely used nowadays, and it is the most efficient domain in health care. We will develop a GUI to get the symptoms from the user. The models used in this paper are K-Nearest Neighbor and Random Forest Classifier. The output is the disease, the accuracy of model, its definition and the treatment of the particular disease based on the symptoms given by the individual. As we all know the saying which tells that prevention of the disease at an early stage is much better than the cure which we take after we get affected by the disease'. This paper shows detailed explanation of how to find the diseases from symptoms, so that the individual can contact the respective doctor and stay healthy at an early stage.**

Keywords — diseases, disease prediction, machine learning, Random Forest Classification, KNN algorithm (K-Nearest Neighbors Algorithm).

I. INTRODUCTION

A disease is a condition that affects the individual functioning of body totally. Diseases if neglected will lead to the death of an individual. Diseases can be identified by the symptoms of the body of an individual. Health is the most important in every human's life. Weekly or monthly check up of one's health is most important for the prevention and also to stay healthy. Nowadays, the individual is not having that much time to go for health check-up. Recently, due to covid-19, no one are willing to go to hospital for health check-up due to the fear of spreading virus. In this situation, technology plays and important role[1].

The popularity of disease prediction models with various types of algorithms shows the reliability of machine learning applications in disease prediction problems such as using common machine learning techniques such as the K-Nearest Neighbor, the Random Forest Classifier, Naive Bayes and Decision Tree which are used to predict disease and the results will be used by individuals to contact their respective doctors and stay healthy early on[1].

Disease prediction is one of the topics that is quite difficult to implement in the form of machine learning models because there are several factors that influence its manufacture, such as uncertainty and mismatch of symptoms for the predicted disease, the similarity of symptoms that make predictions for the disease uncertain, the presence of disease complications experienced in one prediction work and various other factors[2]. These many elements make data analysis and data extraction processes in applying machine learning to this topic as a challenge. As a result, the modeling method used is more limited so that the implementation of machine learning models is much easier, faster but still prioritizes the main factor, there are accuracy in predicting diseases based on inputted symptoms[3].

The author implements a machine learning model with the K-Nearest Neighbor model and the Random Forest Regression model used in this study to see the symptom data provided in the dataset directly, which is then separated into training and test data, and then K-Nearest Neighbor and Random Forest Classifier was conducted to determine how accurate the machine learning model was for disease prediction[4][5]. According to the authors, the new machine learning model is much more precise than the old machine learning model. The machine learning model that has been built and formed can be tried by readers through a streamlit website that has been provided by the author.

The following is a list of the remaining papers that will be discussed next: The tools and methods used to develop this model are explained in Chapter 2, the findings of the analysis acquired from the model are discussed in Chapter 3, the outcomes of the model analysis are reviewed in Chapter 4, and the conclusions are explored in Chapter 5.

II. Related Works

1. Designing Disease Prediction Model Using Machine Learning Approach (Dhiraj Dahiwade, Prof. Gajanan Patle, Prof. Ektaa Meshram, 2019)*Maintaining the Integrity of the Specifications*

Many diseases have arisen throughout history as a result of the environment and habits of the majority of people. A model that can predict the type of disease at an early stage before the disease progresses further in this technologically advanced age is very much needed. The goal of this research is to develop a model that can predict disease patterns from known datasets. KNN (K-Nearest Neighbor) and CSS were used as comparison algorithms to create the model (Convolutional Neural Network). The conclusion of this research is that the CSS algorithm is more accurate than the KNN algorithm, with a score of 84.5 percent.

2. Comparative Study of Classifier for Chronic Kidney Disease prediction using Naive Bayes, KNN and Random Forest (Devika R, Sai Vaishnavi Avilala, V. Subramaniyaswamy, 2019)

CKD (*chronic kidney disease*) is a very common disease that affects more than 15% of Indians [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5969474/]. This disease weakens a person's body, causing anemia, nerve damage, and weak bones. Early detection is required to prevent the disease from progressing further. This study compares several machine learning algorithms for predicting CKD disease, including Naive Bayes, K-Nearest Neighbor, and Random Forest Classifier. According to the findings of this study, the Random Forest Classifier algorithm has the shortest execution time while producing the most accurate results when compared to the Naive Bayes and K-Nearest Neighbor algorithms.

3. Predicting Disease Risks Using Feature Selection Based on Random Forest and Support Vector Machine (Jing Yang, Dengju Yao1, Xiaojuan Zhan, and Xiaorong Zhan, 2019)

Prediction of disease is critical in biomedicine and bioinformatics. To aid in disease prediction efficiency, a model was developed using SVM (Support Vector Machine) and Random Forest to select factors that can be used as disease predictors. The results show that the model algorithm developed in this study can improve classification accuracy.

III. Tools, Framework, And Method

To classify diseases based on their symptoms, the author uses a machine learning method including K-Nearest Neighbor algorithm and the Random Forest Classifier. The author also creates a web interface so that users can see how the product works. To accomplish this, the author employs the following tools and algorithms are used to develop this product:

A. Python

Python is one of many programming languages that are widely used by programmers. The popularity of this programming language is not without reason, python is known as a programming language that is very close to human language (high-level programming language), making it easier for developers and even average people to understand. Furthermore, Python is well-known as one of the languages widely used in the application of machine learning, data analysis, and deep learning.

The author also uses the python programming language in this study because the library required to make stock predictions with random forest regression has already been provided by python. The following libraries were used in this study :

1. Streamlit

Streamlit is an open source python library that is used to create a web that can be modified using the functions it provides[6]. This library is intended specifically for machine learning and data science purposes, as most machine learning and data science libraries can be used in streamlit to quickly build, deploy, or host responsive web sites[7].

2. Pandas

Pandas is a Python-licensed open source library that is widely used to analyze data so that developers can gain more insight from it[8]. Pandas has many tools for analyzing data, one of which is the ability to provide statistical functions based on the analyzed data[9].

3. Seaborn

Seaborn is a Python data visualization library that can visualize complex functions as well as provide very informative statistical plotting results[10].

4. Matplotlib

Matplotlib is a Python-based data visualization library[11]. Matplotlib, unlike Seaborn, can only perform basic data visualization.

5. Plotly

Plotly is an open source library that uses the Python programming language to perform highly interactive and high-quality data visualization[12]. Because of its interactivity, users can directly modify the plot's data visualization.

6. Scikit-learn

Scikit-learn is a Python library for making predictions based on data analysis. This library was created by combining the NumPy, SciPy, and matplotlib libraries[13]. Furthermore, scikit-learn includes a wide variety of libraries related to machine learning algorithms for both supervised and unsupervised learning[14].

7. NumPy

NumPy is a Python library that is widely used for scientific computing and offers many functions related to arrays, mathematics, and logic[15]. A NumPy array is a numerical representation of data that can be used to perform numerical computations effectively and efficiently[16].

B. *Visual Studio Code*

Visual Studio Code is a code editor that supports a variety of programming languages, including html, CSS, and JavaScript. Furthermore, this tool is compatible with all PC operating systems, including Windows, macOS, and Linux. To use Python in Visual Studio Code, users must first download the python extension, which is one of the most popular extensions in the program [https://code.visualstudio.com/docs]. Before starting to code the program itself, users must first create their own folder or workspace in Visual Studio Code[17].

The authors of this study use these tools to accommodate projects and build disease prediction programs in combination with other methods. When using Visual Studio Code to build projects, the author achieves many advantages, including the ability to connect directly to GitHub, countless extensions that make it easier for writers to write programs, a cleaner project workspace, and much more.

C. *Jupyter Notebook*

Jupyter Notebook is a web-based notebook application that is mainly used to compute a language. Jupyter notebook supports over 40 different programming languages, including Python. This notebook is also commonly used for big data computing[18]. This is the main reason that drives the authors to use these tools for data processing tasks like exploratory data analysis, data cleaning, data transformation, and even the development of their own machine learning models. The.ipynb file (python notebook) that can be downloaded when using this notebook can provide a more interactive report because it can provide a more interesting explanation than using ordinary comments[19].

D. *GitHub*

GitHub is a hosting service that allows you to manage software versions. Furthermore, GitHub is widely used due to its collaboration features, which enable many developers to collaborate in a GitHub environment with many other supporting features wherever and whenever they are[20]. The author uses this tool because the author wants all of the project members to collaborate in one environment so that project processing development takes less time and is more efficient.

E. *Random Forest Classification*

Random Forest is a machine learning algorithm that utilizes averaging to improve accuracy and resolve overfitting, and it also uses several decision trees for each sample, which is divided into sub-samples[21].

Because the author's dataset contains thousands of rows of data, the author requires a machine learning algorithm that can manage very large amounts of data. The Random Forest algorithm is well suited to overcoming these issues because it has been shown to provide better accuracy results when dealing with large datasets than traditional decision tree algorithms[22].

The author uses this algorithm to classify certain diseases based on their symptoms. The algorithm will divide the dataset into sub-samples in several trees randomly[5], depending on the number of estimators and max depths entered by the user. Then a vote will be taken for each class, which will later be combined to find the highest vote. The highest vote will be the final prediction.

F. *K-Nearest Neighbor*

K-Nearest Neighbor, or KNN for short, is a machine learning algorithm that is commonly used for classification, but it may also be used for regression cases. It works by providing a training dataset with data points grouped into a specific class, so that data testing can predict that class[23].

It works by determining the number of neighbors to be used for classification. The distance from each data point in the dataset is then calculated. Finally, to determine the class prediction, determine K from data that has not seen during training[4].

This algorithm, like Random Forest, is used by the author to classify diseases based on their symptoms. Furthermore, the authors include this algorithm to compare with random forest to determine which classification algorithms produce the best accuracy.

IV. Result And Discussion

A. *Design Overview*

The following figure will show the system design that the author has made:

As shown in figure 1 below, there are three main pages in the website, which is Home, Data Processing / EDA (*Exploratory Data Analysis*), and About Us.

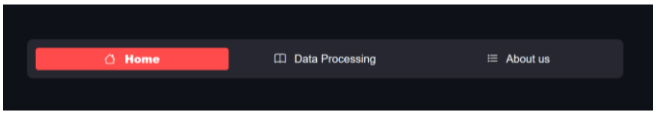
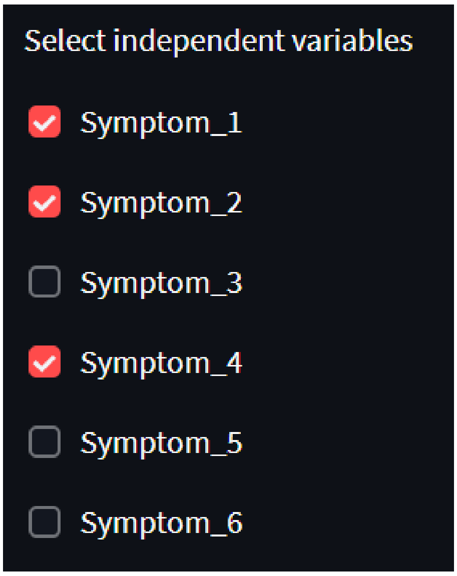


Fig1. Navigation Bar of The Website

Before predicting the disease experienced by the user, the user must enter the number of disease symptoms as one of the desired parameters by checking the symptom box as shown in figure 2 below. The large number of selected disease symptoms will later be used as an independent variable to select the number of disease symptoms that are inputted in the selection column. In addition, the large number of symptoms that are checked will not affect the results in the selection column.



Select independent variables

☑ Symptom_1

☑ Symptom_2

☐ Symptom_3

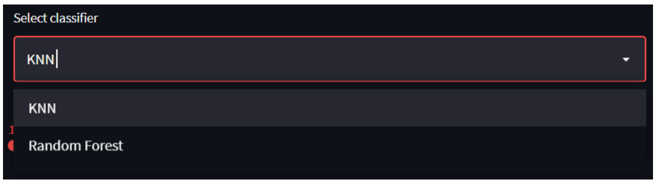
☑ Symptom_4

☐ Symptom_5

☐ Symptom_6

Fig 2. Home (Independent Variables Checklist Box Input)

Furthermore, the user can choose between two model algorithms as shown in figure 3. The first is K-Nearest Neighbor which is a model to find the closest value from the predicted value to determine the prediction result and the second is the Random Forest Classification algorithm which is a classification algorithm to find the highest voting value from each tree exists in the forest or collection of trees to determine the prediction results. Based on the accuracy of the model, the Random Forest Classification algorithm gives slightly better results with the default parameters for these two algorithms.

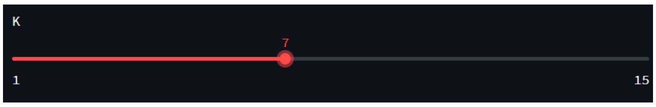


Select classifier

KNN|

Fig 3. Home (Classifier Input)

In the K Nearest Neighbor method, the K value parameter as shown in figure 4 is the number of nearest neighbors used by the algorithm in the classification and regression processes. The value of K is set with an odd value to avoid any similarity in distance that can appear in the KNN process being run.

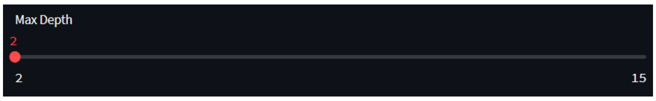


K

7

Fig 4. Home (K Input)

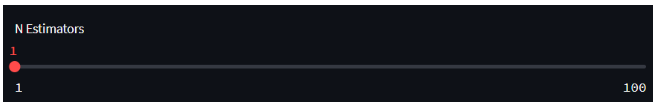
In the Random Forest Classification method, in figure 5 Max Depth parameter represents the depth of each tree in the forest. The deeper the tree, the more splits it has and it captures more information about the data. Then for N Estimators parameter as shown in figure 6 represents the number of trees in the forest. Usually, the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training process considerably, therefore we do a parameter search to find the sweet spot.



Max Depth

2

Fig 5. Home (Max Depth Input)



N Estimators

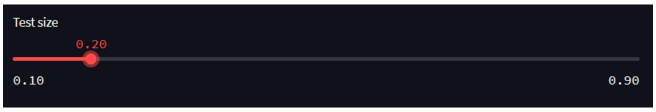
1

Fig 6. Home (N Estimators Input)

Next parameter is test size as shown in figure 7 in the train-test split data, procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Nevertheless, common split percentages include:

- Train: 80%, Test: 20%

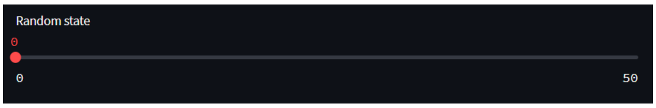
- Train: 67%, Test: 33%
- Train: 50%, Test: 50%


Test size

0.20

Fig 7. Home (Test Size Input)

The last parameter is random state as shown in figure 8 in the train-test split data to controls the shuffling process. With random_state =None, we get different train and test sets across different executions and the shuffling process is out of control. And with random state=0 , we get the same train and test sets across different executions.


Random state

0

Fig 8. Home (Random State Input)

After the user has set all the desired inputs and parameters, there will be a notification verification of information about the classifier method and parameters used as shown in figure 9 below. Every time the user makes changes to the classifier method and this notification parameter will update itself.

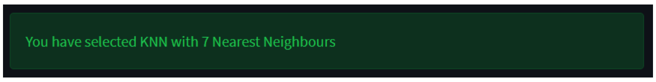

You have selected KNN with 7 Nearest Neighbours

Fig 9. Home (Information Verification Announce)

Then the user can input and select the symptoms of the disease in the symptom selection column as shown in figure 10 and figure 11. The lack of inputted disease symptoms will not affect the disease prediction results for the maximum number of disease symptoms that can be selected in the symptom selection column as shown in figure 12.

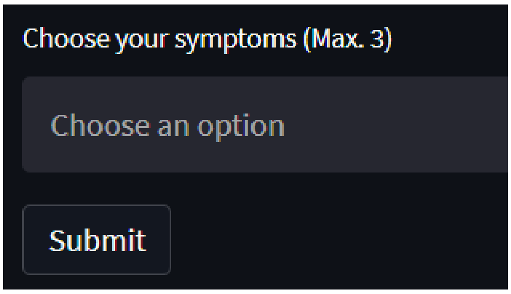

Choose your symptoms (Max. 3)

Fig 10. Home (Choose the Symptoms)

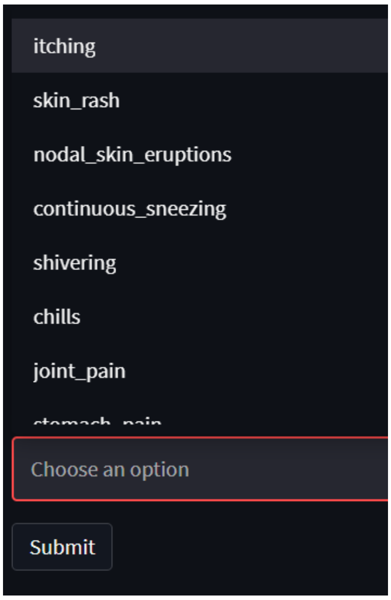
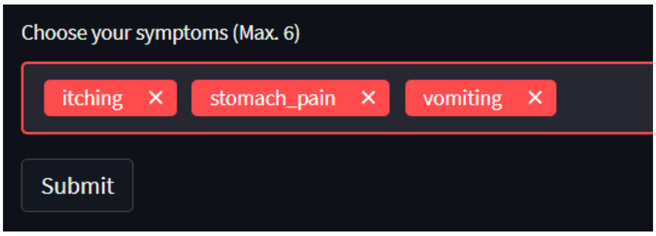


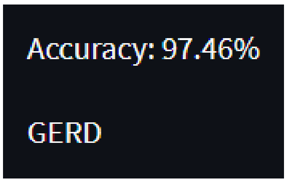
Fig 11. Home (Choices of Symptoms)


Choose your symptoms (Max. 6)

itching ✕ stomach_pain ✕ vomiting ✕

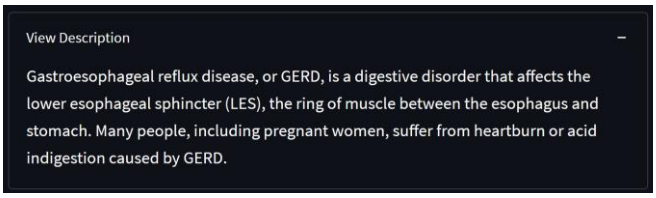
Fig 12. Home (Choices of Symptoms Appearance)

According to figure 13, after entering the symptoms of the disease by pressing submit button the user will get a prediction of what disease is experienced for the symptoms that have been inputted previously. Users can see the accuracy of machine learning model predictions based on the machine learning method used. Not only that, to complete information about the disease that has been predicted, the user can find out the description and how to precautions the disease that has been predicted as shown in figure 14 and figure 15.


Accuracy: 97.46%

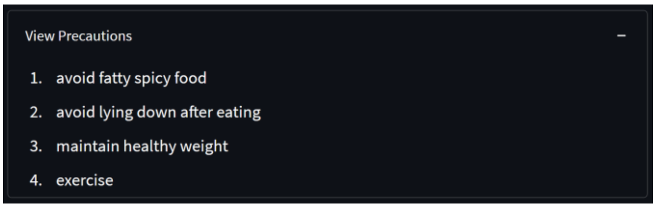
GERD

Fig 13. Home (Disease Prediction Results)


View Description –

Gastroesophageal reflux disease, or GERD, is a digestive disorder that affects the lower esophageal sphincter (LES), the ring of muscle between the esophagus and stomach. Many people, including pregnant women, suffer from heartburn or acid indigestion caused by GERD.

Fig 14. Home (Description of the Disease Prediction Results)



View Precautions –

1. avoid fatty spicy food
2. avoid lying down after eating
3. maintain healthy weight
4. exercise

Fig 15. Home (Precautions of the Prediction Results)

Not only that, but the author also adds data visualization so users can also see machine learning model data visualizations depending on the user choosing what machine learning algorithm is used, whether K-Nearest Neighbor algorithm or the Random Forest Classification algorithm. If the user used K-Nearest Neighbor algorithm as classifier, then the visualization data will be in the form of a three-dimensional chart as shown in figure 16 that has unique symbols accompanied by different colors to represent each disease that can be seen in the figure 17. Furthermore, if the user used Random Forest Classification algorithm as classifier, then the visualization data will be in the form of a tree diagram which is technically a graphviz as shown in figure 18.

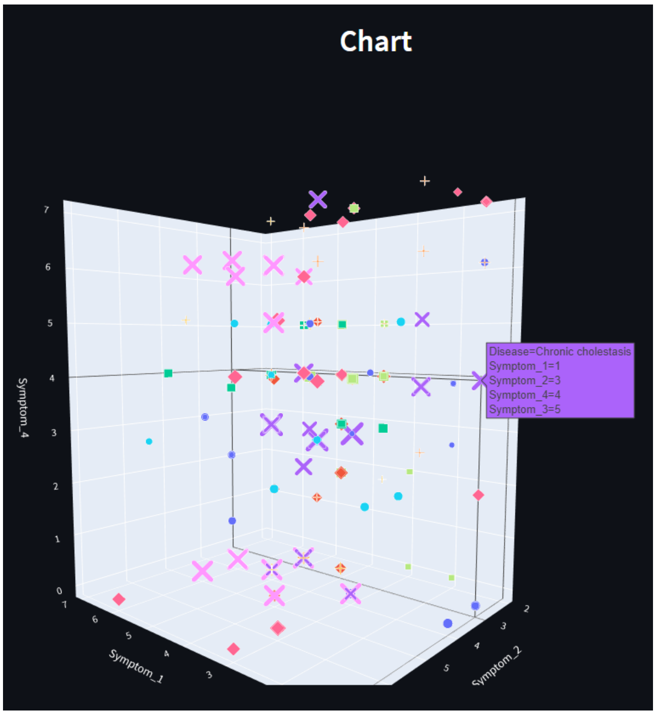
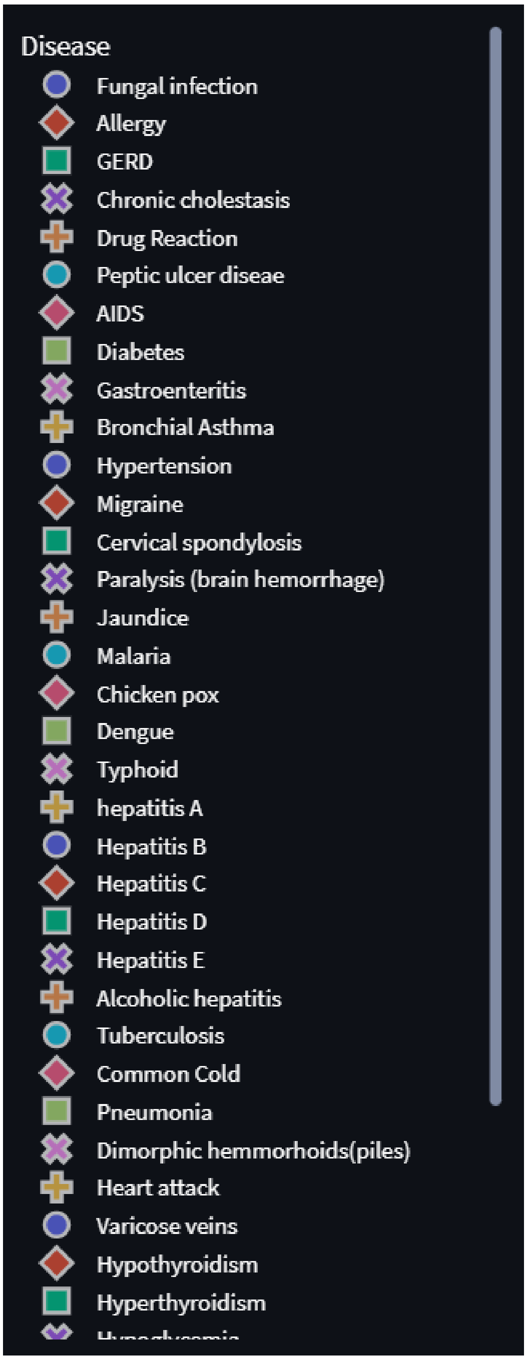


Fig 16. Home (Chart of the Disease Prediction Results)



Disease

● Fungal infection
◆ Allergy
■ GERD
✖ Chronic cholestasis
✚ Drug Reaction
● Peptic ulcer diseae
◆ AIDS
■ Diabetes
✖ Gastroenteritis
✚ Bronchial Asthma
● Hypertension
◆ Migraine
■ Cervical spondylosis
✖ Paralysis (brain hemorrhage)
✚ Jaundice
● Malaria
◆ Chicken pox
■ Dengue
✖ Typhoid
✚ hepatitis A
● Hepatitis B
◆ Hepatitis C
■ Hepatitis D
✖ Hepatitis E
✚ Alcoholic hepatitis
● Tuberculosis
◆ Common Cold
■ Pneumonia
✖ Dimorphic hemmorhoids(piles)
✚ Heart attack
● Varicose veins
◆ Hypothyroidism
■ Hyperthyroidism
✖ Hypoglycemia

Fig 17. Home (Legend Symbols of the Chart)

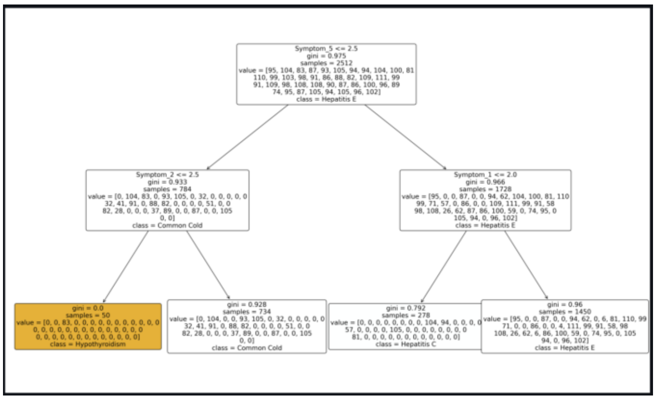
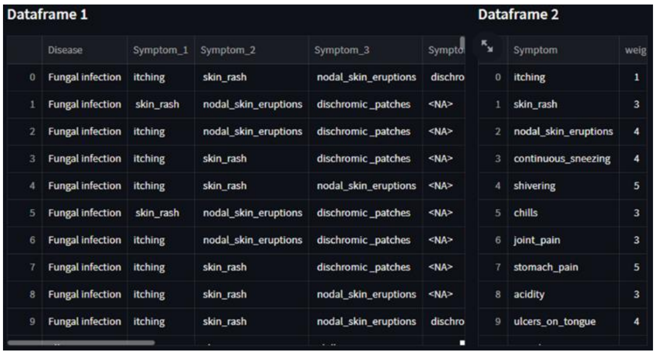


Fig 18. Home (GraphViz)

B. *Data Analysis and Result*

A good model can be obtained by going through several stages of data processing, including exploratory data analysis, data cleaning, data transformation, data splitting, and model building. Some of the datasets used by the author in this study are as follows :



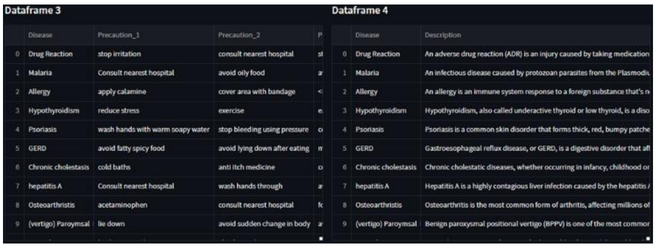
Dataframe 1

	Disease	Symptom_1	Symptom_2	Symptom_3	Sympto
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischro
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	<NA>
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	<NA>
3	Fungal infection	itching	skin_rash	dischromic _patches	<NA>
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	<NA>
5	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	<NA>
6	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	<NA>
7	Fungal infection	itching	skin_rash	dischromic _patches	<NA>
8	Fungal infection	itching	skin_rash	nodal_skin_eruptions	<NA>
9	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischro

Dataframe 2

	Symptom	weig
0	itching	1
1	skin_rash	3
2	nodal_skin_eruptions	4
3	continuous_sneezing	4
4	shivering	5
5	chills	3
6	joint_pain	3
7	stomach_pain	5
8	acidity	3
9	ulcers_on_tongue	4

Fig 19. Dataframe 1 & Dataframe 2



Dataframe 3

	Disease	Precaution_1	Precaution_2	P
0	Drug Reaction	stop irritation	consult nearest hospital	st
1	Malaria	Consult nearest hospital	avoid oily food	a
2	Allergy	apply calamine	cover area with bandage	<
3	Hypothyroidism	reduce stress	exercise	e
4	Psoriasis	wash hands with warm soapy water	stop bleeding using pressure	c
5	GERD	avoid fatty spicy food	avoid lying down after eating	m
6	Chronic cholestasis	cold baths	anti itch medicine	c
7	hepatitis A	Consult nearest hospital	wash hands through	a
8	Osteoarthristis	acetaminophen	consult nearest hospital	fc
9	(vertigo) Paroymsal	lie down	avoid sudden change in body	a

Dataframe 4

	Disease	Description
0	Drug Reaction	An adverse drug reaction (ADR) is an injury caused by taking medication
1	Malaria	An infectious disease caused by protozoan parasites from the Plasmodiu
2	Allergy	An allergy is an immune system response to a foreign substance that's n
3	Hypothyroidism	Hypothyroidism, also called underactive thyroid or low thyroid, is a diso
4	Psoriasis	Psoriasis is a common skin disorder that forms thick, red, bumpy patche
5	GERD	Gastroesophageal reflux disease, or GERD, is a digestive disorder that af
6	Chronic cholestasis	Chronic cholestatic diseases, whether occurring in infancy, childhood or
7	hepatitis A	Hepatitis A is a highly contagious liver infection caused by the hepatitis /
8	Osteoarthristis	Osteoarthritis is the most common form of arthritis, affecting millions of
9	(vertigo) Paroymsal	Benign paroxysmal positional vertigo (BPPV) is one of the most commor

Fig 20. Dataframe 3 & Dataframe 4

The author of this product [24] employs a dataset from [25] that consists of four dataframes, there are:

1. Dataset on disease and its symptoms (Figure 19).
2. Dataset on symptoms and their impact on the body every two days (Figure 19).
3. Dataset on disease and prevention (Figure 20).
4. Dataset on disease and description of the disease (Figure 20).

After understanding each dataframe, the author will analyze all of them. However, because the author is developing an application to predict the disease, the author will only examine dataframe 1 and dataframe 2. While dataframe 3 and dataframe 4 will be used by the author to provide a disease description and treatment after the disease has been predicted.

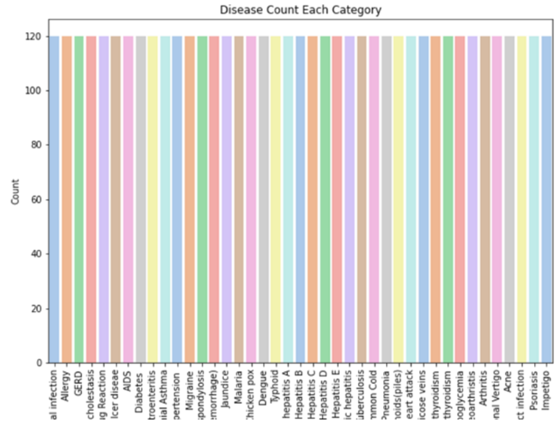


Fig 21. Disease Count in Each Category

According to the bar chart as shown in figure 21 above, there are 41 different types of disease, and for each type of disease, there are 120 diseases with the same name but different symptoms and data patterns. Furthermore, because all diseases have the same number, the data distribution in the dataset looks normal.

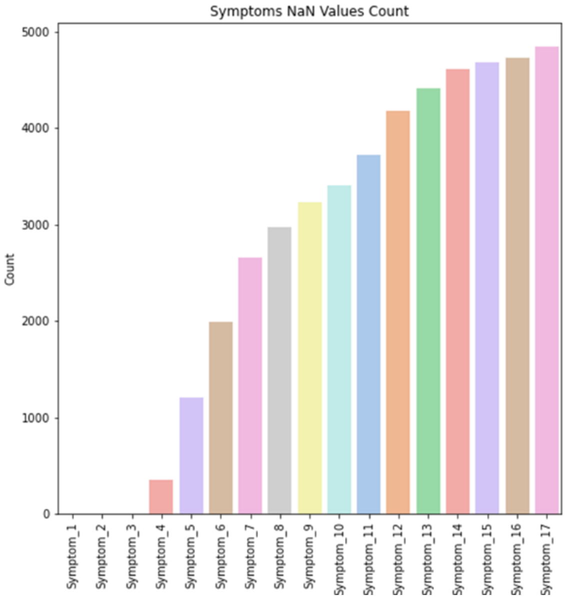
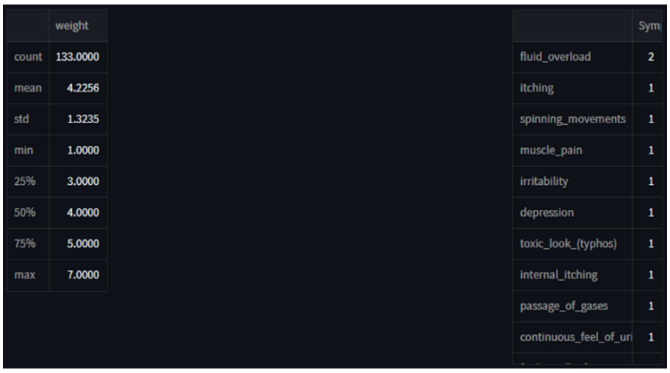


Fig 22. Symptoms NaN Values

The author can conclude from the above chart as shown in figure 22 that the empty disease symptom begins with the third symptom and increases in value with the next symptom. This is due to the fact that in the majority of cases, only two to three major symptoms are mentioned by patients. As a result, if there are more disease symptoms, there will be more empty column values.



	weight
count	133.0000
mean	4.2256
std	1.3235
min	1.0000
25%	3.0000
50%	4.0000
75%	5.0000
max	7.0000

	Sym
fluid_overload	2
itching	1
spinning_movements	1
muscle_pain	1
irritability	1
depression	1
toxic_look_(typhos)	1
internal_itching	1
passage_of_gases	1
continuous_feel_of_uri	1

Fig 23. Weight & Symptoms

A summary value of the weight column in dataframe 2 is shown in the left table in figure 23 above. Because the standard deviation value is much smaller than the mean value, it can be concluded that the data is less varied.

Furthermore, the left table contains a count value of 133, indicating that there are 133 different disease symptoms in the dataset. However, as shown in the right table in figure 23, one disease symptom, "fluid overload," has the same value. As a result, the total number of symptoms of the disease is 132. This same value will be deleted later, one of which will be during data cleaning.

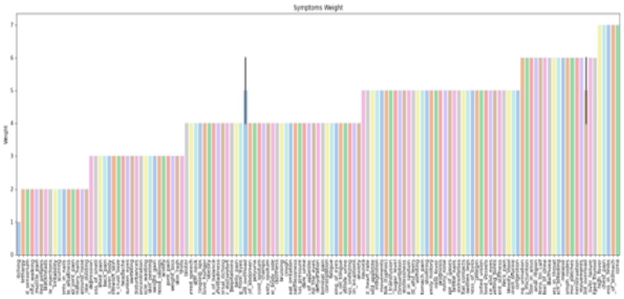


Fig 24. Symptoms Weight

The chart above as shown in figure 24 above, shows that the weight data in the dataset ranges from 1 to 7, with most of them falling between 4 and 5. While only a few symptoms are worth 1 or 7, The weight here is the value used to calculate the impact of symptoms. every two days to the body the greater the impact on the body, the higher the value.

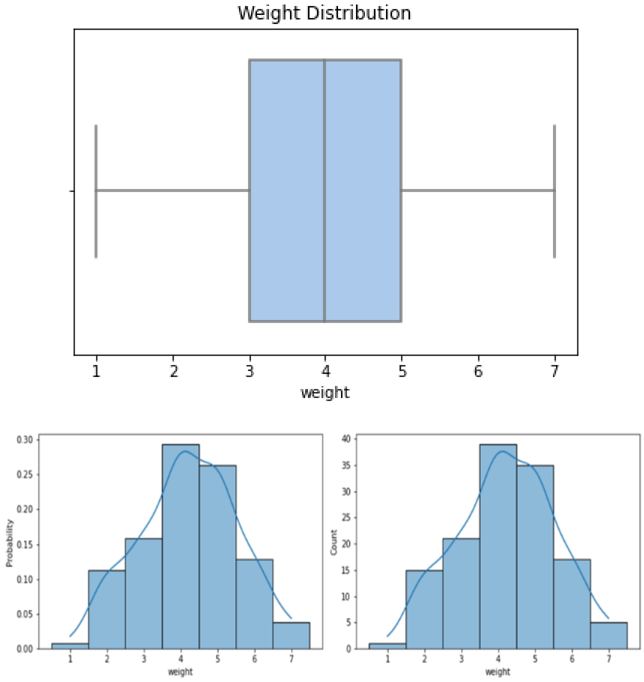
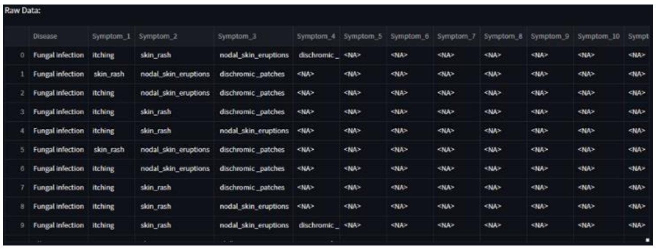


Fig 25. Weight Distribution

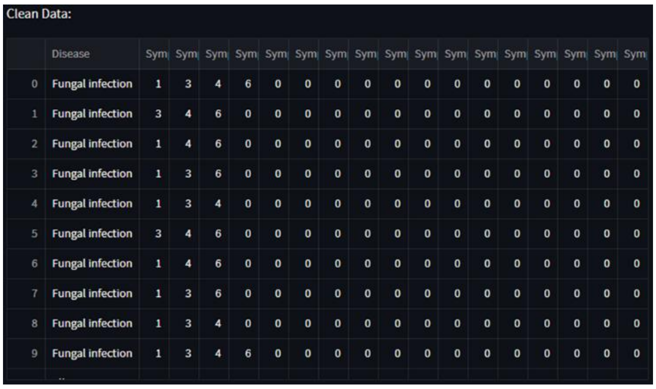
The boxplot on the leftmost chart as shown in figure 25 above, shows that the data distribution is very normal (normal distribution), as the author stated, and there are no outliers because the data is very balanced between min, max, Q1, Q2, and Q3. As previously explained, the values that appear the most in the histogram in figure 25, are weight values equal to 4 and 5, which causes the probability of the weight to be the highest.



Raw Data:

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Symptom_9	Symptom_10	Sympt
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
3	Fungal infection	itching	skin_rash	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
5	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
6	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
7	Fungal infection	itching	skin_rash	dischromic _patches	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
8	Fungal infection	itching	skin_rash	nodal_skin_eruptions	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>
9	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>

Fig 26. Raw Data



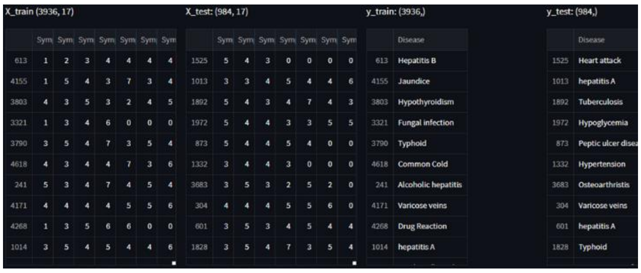
Clean Data:

	Disease	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym	Sym
0	Fungal infection	1	3	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0
1	Fungal infection	3	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	Fungal infection	1	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Fungal infection	1	3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Fungal infection	1	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Fungal infection	3	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Fungal infection	1	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Fungal infection	1	3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Fungal infection	1	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	Fungal infection	1	3	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig 27. Clean Data

There following list are four major stages in data cleaning and data transformation as shown in figure 26 and figure 27, including:

1. Delete duplicate data.
2. The first step is to remove the duplicate data. As a result, the author will delete one of the "fluid overload" data points to improve the model's accuracy.
3. Replace the value of NaN with a value of 0.
4. Replace the symptom string that has a different value with the actual value, then replace it with the weight symptom. It turns out that there are some incorrect values in the column, specifically extra spaces (for example, 'dischromic _patches,' 'foul smell of urine,' and'spotting_ urination'). As a result, the author removes the existing spaces and converts them to numbers using the weights in dataset 2.



X_train (3936, 17)

	Sym	Sym	Sym	Sym	Sym	Sym	Sym
613	1	2	3	4	4	4	4
4155	1	5	4	3	7	3	4
3803	4	3	5	3	2	4	5
3321	1	3	4	6	0	0	0
3790	3	5	4	7	3	5	4
4618	4	3	4	4	7	3	6
241	5	3	4	7	4	5	4
4171	4	4	4	4	5	5	6
4268	1	3	5	6	6	0	0
1014	3	5	4	5	4	4	6

X_test: (984, 17)

	Sym	Sym	Sym	Sym	Sym	Sym	Sym
1525	5	4	3	0	0	0	0
1013	3	3	4	5	4	4	6
1892	5	4	3	4	7	4	3
1972	5	4	4	3	3	5	5
873	5	4	4	5	4	0	0
1332	3	4	4	3	0	0	0
3683	3	5	3	2	5	2	0
304	4	4	4	5	5	6	0
601	3	5	3	4	5	4	4
1828	3	5	4	7	3	5	4

y_train: (3936,)

	Disease
613	Hepatitis B
4155	Jaundice
3803	Hypothyroidism
3321	Fungal infection
3790	Typhoid
4618	Common Cold
241	Alcoholic hepatitis
4171	Varicose veins
4268	Drug Reaction
1014	hepatitis A

y_test: (984,)

	Disease
1525	Heart attack
1013	hepatitis A
1892	Tuberculosis
1972	Hypoglycemia
873	Peptic ulcer disea
1332	Hypertension
3683	Osteoarthristis
304	Varicose veins
601	hepatitis A
1828	Typhoid

Fig 28. Split Data

Training and data testing are required to build the model as shown in figure 28 above. Training is used to train the model using existing data, whereas testing is used to see how the model performs with data that has never been seen or trained before. The data will be divided on a 4:1 scale, with the training set receiving 80% and the testing set receiving 20%.

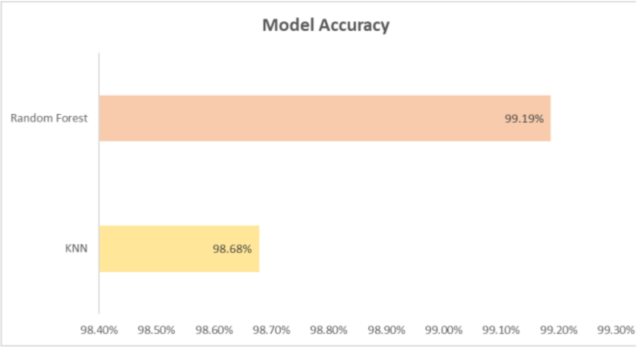


Fig 29. Model Accuracy

KNN and Random Forest Classification were the model algorithms investigated. Based on the model's accuracy in figure 29, the random forest algorithm produces slightly better results with the default parameters for these two algorithms.

V. Conclusion

Many people have been concerned about their health since the pandemic. Many people suffer from mental disorders because of having too many thoughts. Our products exist to alleviate these issues. In this research, the author has provided the benefits of the product as well as its explanation. However, our product's weakness is that the input of symptoms is very pattern oriented. To provide the best predictive results, the symptoms provided by the user must be in sequence and follow the pattern of symptoms in the training data.

If the symptom data entered corresponds to a specific disease, but the sequence is randomized or different from the original, the prediction results will be incorrect. Furthermore, the dataset has a small data pattern, which means that the model's accuracy can reach 98 percent or higher.

In the future, the author hopes that this product can be improved in terms of the amount of data in the dataset, data patterns in the dataset, disease types, and so on. Thus, this product can provide significant assistance for health workers.

References

[1] R. C. Mallela, R. L. Bhavani, and B. Ankayarkanni, “Disease Prediction Using Machine Learning Techniques,” *Proc. 5th Int. Conf. Trends Electron. Informatics, ICOEI 2021*, pp. 962–966, Jun. 2021, doi: 10.1109/ICOEI51242.2021.9453078.

[2] A. Mosavi, S. F. Ardabili, and S. Shamshirband, “Demand Prediction with Machine Learning Models ; State of the Art and a Systematic Review of Advances,” no. May, pp. 1–21, 2019, doi: 10.20944/preprints201905.0175.v1.

[3] A. Mosavi, S. F. Ardabili, and S. Shamshirband, “Demand Prediction with Machine Learning Models; State of the Art and a Systematic Review of Advances,” May 2019, doi: 10.20944/PREPRINTS201905.0175.V1.

[4] P. Cunningham and S. J. Delany, “K-Nearest Neighbour Classifiers-A Tutorial,” *ACM Comput. Surv.*, vol. 54, no. 6, 2021, doi: 10.1145/3459665.

[5] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, “RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12343 LNCS, pp. 503–515, 2020, doi: 10.1007/978-3-030-62008-0_35.

[6] “Streamlit documentation.” [Online]. Available: https://docs.streamlit.io/.

[7] H. Dani, “Review on Frameworks Used for Deployment of Machine Learning Model,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 2, pp. 211–215, 2022, doi: 10.22214/ijraset.2022.40222.

[8] “pandas documentation — pandas 1.4.2 documentation.” [Online]. Available: https://pandas.pydata.org/docs/.

[9] W. McKinney, “pandas: a Foundational Python Library for Data Analysis and Statistics,” *Python High Perform. Sci. Comput.*, no. January 2011, pp. 1–9, 2011.

[10] M. Waskom, “seaborn: statistical data visualization,” *J. Open Source Softw.*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/JOSS.03021.

[11] “Complex and semantic figure composition — Matplotlib 3.5.2 documentation.” Accessed: Jun. 18, 2022. [Online]. Available: https://matplotlib.org/stable/tutorials/provisional/mosaic.html.

[12] “Plotly Python Graphing Library.” [Online]. Available: https://plotly.com/python/.

[13] “scikit-learn: machine learning in Python — scikit-learn 1.1.1 documentation.” [Online]. Available: https://scikit-learn.org/stable/.

[14] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. January, pp. 2825–2830, 2011.

[15] “NumPy documentation — NumPy v1.22 Manual.” [Online]. Available: https://numpy.org/doc/stable/.

[16] S. Van der Walt and M. Aivazis, “The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering,” *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011, [Online]. Available: http://aip.scitation.org/doi/abs/10.1109/MCSE.2011.37.

[17] J. K. Rask, F. P. Madsen, N. Battle, H. D. Macedo, H. Daniel Macedo, and P. G. Larsen, “Visual Studio Code VDM Support,” no. December, pp. 1–20, 2020, [Online]. Available: https://pypl.github.io/IDE.html.

[18] “Project Jupyter | Home.” [Online]. Available: https://jupyter.org/.

[19] “(PDF) PENGGUNAAN PIRANTI LUNAK JUPYTER NOTEBOOK DALAM UPAYA MENSOSIALISASIKAN OPEN SCIENCE.” [Online]. Available: https://www.researchgate.net/publication/326132474_PENGGUNAAN_PIRANTI_LUNAK_JUPYTER_NOTEBOOK_DALAM_UPAYA_MENSOSIALISASIKAN_OPEN_SCIENCE.

[20] “Hello World - GitHub Docs.” [Online]. Available: https://docs.github.com/en/get-started/quickstart/hello-world.

[21] “sklearn.ensemble.RandomForestClassifier — scikit-learn 1.1.1 documentation.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[22] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, “Random forests and decision trees,” *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 5, pp. 272–278, 2012.

[23] K. Taunk, “2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019,” *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. Iciccs, pp. 1255–1260, 2019.

[24] “Streamlit.” [Online]. Available: https://share.streamlit.io/alvon17/disease-prediction-web-app/application.py.

[25] “Disease Symptom Prediction | Kaggle.” [Online]. Available: https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset.