ព្រះរាជាណាចក្រកម្ពុជា
ជាតិ សាសនា ព្រះមហាក្សត្រ

Project: Diagnosis Disease and Recommendation with ML

GROUP: I3-AMS2(D)

Name of Students	ID of Students	SCORE.
Sorn Sreynatt	e20210292	……
Ty Kana	e20210275	……
Vey SreyPich	e20210708	……
Vorn Seavmey	e20211478	……

Lecturer : Professor: Phann Raksmey

Academic Year: 2024 -2025

Content:

Abstract

This study presents a machine learning-based system for precise disease diagnosis and personalized treatment recommendations, utilizing datasets from Kaggle and other sources. Employing Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) classifiers enhances diagnostic accuracy, complemented by Neural Networks for advanced pattern recognition. Methodologically, rigorous data preprocessing ensures data quality, including handling missing values and standardizing formats. Model training incorporates cross-validation for robust performance validation.

Hyperparameter optimization via GridSearch fine-tunes model parameters for optimal diagnostic efficacy. Evaluation metrics such as accuracy, precision, recall, and F1-score gauge model performance comprehensively. The system features a user-friendly Streamlit interface, enabling symptom-based disease prediction and personalized health recommendations, thus enhancing accessibility to healthcare insights.

This integration of advanced machine learning techniques aims to revolutionize diagnostic accuracy and treatment outcomes, improving patient care delivery. Future developments will focus on expanding datasets, refining algorithms, and integrating real-time patient data to further enhance diagnostic precision and personalized healthcare delivery.

Introduction

"It is health that is real wealth and not pieces of gold and silver." These words by Mahatma Gandhi underscore the fundamental truth that health is the cornerstone of human well-being and prosperity. In our modern age, despite advances in technology and medicine, disparities in healthcare access persist, posing significant challenges to global health equity.

Access to quality healthcare remains uneven across regions and populations, emphasizing the urgent need for innovative solutions that can democratize healthcare services. Early detection of diseases and personalized treatment recommendations are pivotal in improving health outcomes and reducing healthcare costs. This paper proposes a transformative approach to address these challenges using machine learning technology.

Imagine a future where predictive analytics and machine learning algorithms enable individuals to anticipate health issues before symptoms manifest. This proactive approach empowers individuals to make informed decisions about their health, leading to better management of chronic conditions and prevention of diseases.

By leveraging datasets from Kaggle and other repositories, this study introduces a machine learning-based system designed to democratize healthcare access. The system automates disease diagnosis and offers personalized treatment recommendations based on individual symptoms and medical history. This innovative approach not only enhances diagnostic accuracy but also empowers individuals with timely medical insights, fostering a proactive approach to healthcare management.

Through the integration of technology and health awareness, this study aims to bridge gaps in healthcare delivery, making quality healthcare more accessible and efficient worldwide. By harnessing the power of machine learning, we can realize Gandhi's vision of health as true wealth, where every individual can lead a healthier and more fulfilling life.

2.1 Research Background

Cambodia, as a developing country, has invested heavily in various sectors such as infrastructure, agriculture, education, and the medical sector. Despite the government's significant efforts and expenditures to bolster these areas, challenges persist that hinder comprehensive support for the entire population. These issues are particularly pronounced in the medical sector, where several key problems can be identified:

1**. Lack of Resources**: Despite the presence of medical schools in Cambodia, these institutions often lack the resources and expertise to address certain rare diseases. This results in a shortage of specialized medical professionals who can diagnose and treat these conditions effectively. Additionally, while some hospitals in urban areas are equipped with advanced medical devices, these resources are still insufficient to meet the needs of the entire country. The high cost of medical equipment further exacerbates this issue, limiting accessibility and availability for the broader population.

2. **Awareness and Health Education**: Although most people understand the importance of health, there are significant gaps in health awareness and education. Several factors contribute to this issue:

- **Financial Constraints**: Many Cambodians have low incomes and prioritize spending on essential needs such as electricity, water, food, and education for their children. This financial strain often leaves little room for regular health check-ups. Doctors recommend that individuals undergo health check-ups at least once or twice a year reference, but it is estimated that 80 to 90 percent of Cambodians do not adhere to this guideline. People tend to seek medical attention only when they start experiencing symptoms or serious health issues, which is not the optimal approach for maintaining good health.

- **Lack of Health Education**: There is a widespread lack of awareness about health education in Cambodia. Even those who are aware of health guidelines sometimes neglect them. For instance, many people consume junk food, foods high in cholesterol, excessive sugar, and alcohol, and some continue to smoke. These lifestyle choices contribute to various health problems, exacerbating the overall healthcare challenges in the country.

3. **Geographical Disparities**: The distribution of medical resources and healthcare services is uneven, with urban areas generally having better access to medical facilities and professionals than rural areas. This disparity means that people in remote and rural regions often face significant obstacles in accessing necessary healthcare services, leading to delayed treatment and poorer health outcomes.

4. **Healthcare Infrastructure:** While the government has made strides in improving healthcare infrastructure, many facilities still lack modern equipment and adequate staffing. This limitation affects the quality of care that can be provided, particularly in public hospitals and clinics.

5. **Training and Retention of Medical Professionals**: There is a need for ongoing training and professional development for medical personnel to keep up with advancements in medical science and technology. Additionally, retaining skilled medical professionals is a challenge, as many seek better opportunities abroad, further depleting the local talent pool.

6**. Economic Disparities in Healthcare:** There is a significant gap between rich and poor in terms of access to healthcare. Wealthier individuals can afford private healthcare services, which tend to be of higher quality and more accessible. In contrast, lower and middle-class citizens often struggle with limited access to healthcare services due to financial constraints. Public healthcare services, while more affordable, are frequently under-resourced and overburdened, leading to longer wait times and reduced quality of care. This disparity exacerbates health inequities and highlights the need for more inclusive healthcare policies. Reference

According to Open Development Cambodia , healthcare spending accounted for 485 million USD in 2019, which is approximately 6% of the country's GDP. Recognizing the importance of good health services, the government has committed to strengthening healthcare services and quality, especially in rural areas, by working with many development partners.

2.2 The Role of AI in Healthcare

Artificial Intelligence (AI) has made significant strides in various fields over the past few years, including healthcare. AI's ability to analyze vast amounts of data, recognize patterns, and make predictions positions it as a powerful tool for addressing many of the challenges faced by the Cambodian healthcare system. AI can assist in optimizing resource allocation, improving diagnostic accuracy, enhancing health education, and bridging the gap between urban and rural healthcare services.

AI technologies, such as machine learning, natural language processing, and computer vision, have been increasingly applied in medical contexts. These technologies can analyze medical records, research papers, and other data sources to provide insights that might not be

immediately apparent to human practitioners. For instance, AI can identify patterns in patient data that indicate early signs of disease, suggest personalized treatment plans, and even predict potential health risks based on genetic and lifestyle factors. Health Care Transformer

2.3 Introducing AI-Based Diagnosis

To address the aforementioned problems, this project focuses on using machine learning for disease diagnosis. The machine learning model will predict diseases based on patient-reported symptoms, ranking the top 10 possible diseases out of 41 common diseases. The system will also provide additional information such as disease descriptions, precautions, doctor information, recommended workouts, and dietary advice. This holistic approach aims to enhance healthcare accessibility and quality, particularly in resource-limited settings.

By leveraging datasets from platforms like Kaggle and other medical repositories, a machine learning-based system can be developed to automate disease diagnosis and offer personalized treatment recommendations. This system can enhance diagnostic accuracy, reduce the burden on overworked healthcare professionals, and provide patients with quick and reliable health assessments.

For example, AI algorithms can be trained to recognize patterns in imaging data, such as X-rays and MRIs, to detect conditions like pneumonia, tumors, and other anomalies. Similarly, natural language processing can be used to analyze doctors' notes and patient histories to identify correlations and recommend appropriate interventions.

2.4 Benefits of AI in Healthcare

The integration of AI into Cambodia's healthcare system offers several key benefits:

1. **Enhanced Diagnostic Accuracy**: AI systems can process and analyze medical data with high precision, reducing the likelihood of misdiagnoses and ensuring that patients receive accurate and timely information about their health.

2. **Resource Optimization**: AI can help prioritize medical resources and personnel, ensuring that patients with the most urgent needs receive attention first. This is particularly important in settings where healthcare resources are limited.

3. **Increased Accessibility**: AI-powered tools can be deployed in rural and remote areas, providing quality healthcare services to populations that might otherwise lack access to specialized medical care.

4. **Personalized Medicine**: AI can analyze individual patient data to provide tailored treatment recommendations, taking into account unique genetic, environmental, and lifestyle factors.

5. **Improved Health Education**: AI can be used to develop educational programs and materials that enhance public awareness about health and wellness, encouraging preventative care and healthier lifestyles.

Literature Review

Intelligent Disease Diagnosis Using Machine Learning: An Overview

The field of intelligent disease diagnosis using machine learning has seen substantial advancements, aiming to address the constraints of medical resource availability and improve preliminary disease detection. This literature review synthesizes findings from three notable studies in this domain.

1. Comparative Study of Machine Learning Algorithms for Multi-Disease Prediction" by Bharati et al.

Bharati et al. (2020) conducted a comparative study on the efficacy of various machine learning algorithms for multi-disease prediction based on patient symptoms. The study employed decision trees, random forests, and support vector machines (SVMs), comparing their performance in terms of accuracy, precision, and recall. The researchers utilized a comprehensive dataset comprising patient symptoms and corresponding diagnoses, demonstrating that random forests outperformed other algorithms, achieving an accuracy of 92%. The study emphasized the importance of feature selection and data preprocessing in enhancing model performance. Reference

2. Intelligent Disease Prediagnosis Only Based on Symptoms" by Luo et al.

Luo et al. (2021) explored the development of an intelligent disease prediagnosis system based solely on patient-reported symptoms. This study aimed to alleviate the strain on medical resources by providing a preliminary diagnosis that could guide patients towards appropriate medical treatment. The authors employed neural networks and SVMs to categorize diseases into main categories, subtypes, and specific diseases. Their hierarchical approach to disease identification demonstrated promising results, with the neural network model achieving superior accuracy compared to the SVM. The system's practical application in medical triage and its

potential to assist in areas with limited medical resources were highlighted as significant contributions. Reference

3. Machine Learning Models for Disease Diagnosis: A Comparative Analysis by Zhang et al.

Zhang et al. (2022) conducted a comparative analysis of different machine learning models for disease diagnosis using symptom-based data. The study included logistic regression, k-nearest neighbors (KNN), and deep learning models, evaluating their performance on a dataset encompassing various common diseases.

significantly outperforming traditional machine learning models. The authors attributed the success of the CNN to its ability to capture complex patterns in the data, suggesting its potential for real-world applications in automated disease diagnosis. Reference

Integration and Comparative Insights

The comparative insights from these studies underscore the varying effectiveness of different machine learning algorithms in disease diagnosis. Bharati et al. and Zhang et al. both highlighted the superiority of ensemble and deep learning models over traditional algorithms like decision trees and logistic regression. Luo et al.'s hierarchical approach using neural networks further adds to the evidence that advanced neural architectures can provide high accuracy in symptom-based disease prediction.

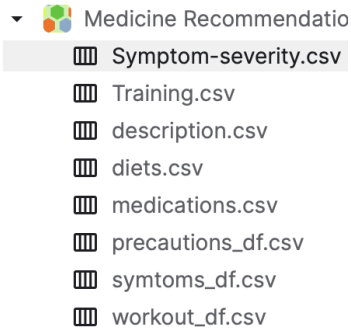
Data Preparation

3.1 Resource of Data

The step input data into the place and prepare it to use in our machine learning training . this step divided into 2 processes:

- Data exploration: it is used to understand the nature of the data that we have to work with. we can understand the characteristics, format and quality of the data. When we understand the feature,we find Correlation, general trends, and outliers.
- Data pre-processing: is preprocessing of data for its analysis.

For data we select data from kaggle. for data we use it have some file in this data it have cleanData description diets doctor precaution symptoms training workout.



3.2 Data Cleaning

The process of cleaning and converting raw data into a usable format. It is the process of cleaning the data, selecting the variable to use and transforming the data in a proper format to make it more suitable for analysis on the next step.

From the dataset above we have 20 feature or variable such as:

	prognosis	abdominal_pain	acidity	altered_sensorium	anxiety	back_pain	blackheads	bladder_discomfort	blister	bloody_stool	...	vomiting
0	Fungal infection	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
1	Fungal infection	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
2	Fungal infection	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
3	Fungal infection	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
4	Fungal infection	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

5 rows × 87 columns

- Unnamed
- Disease
- Symptom-1
- Symptom-2
- Symptom-3
- Symptom4

We already know about the features that we use to find Average accuracy but for the step we have change data from category to numerical to study the model.

To change data from category to numerical to study the model we use ‘ `result_df.columns’`

and here is the results

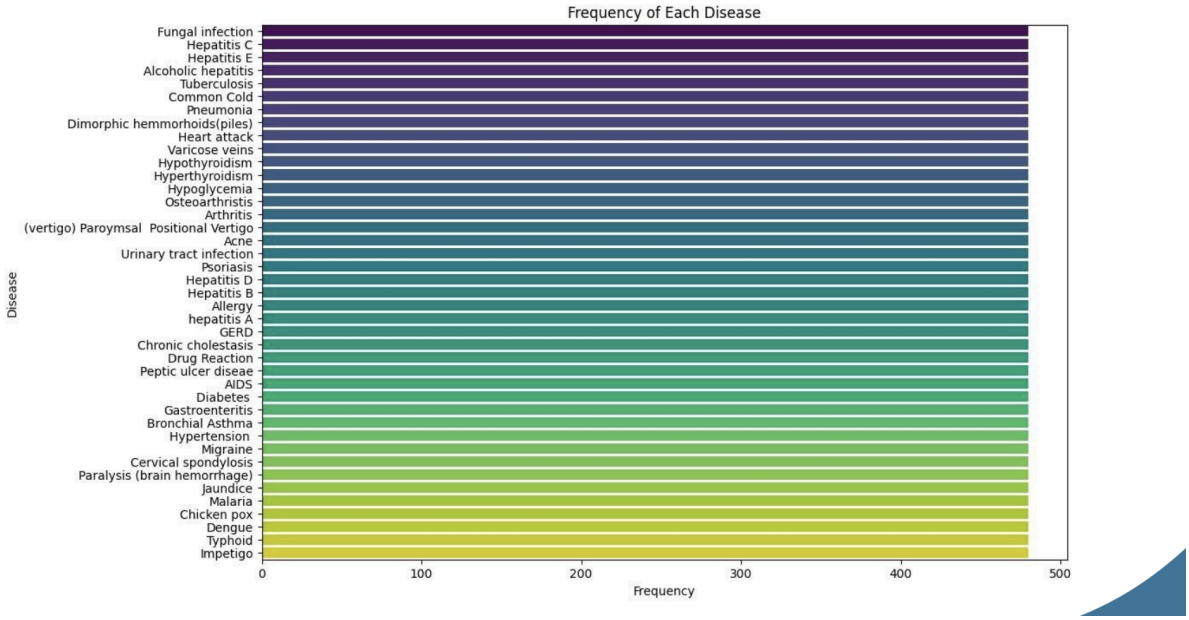
```
Index([' prognosis', ' abdominal_pain', ' acidity', ' altered_sensorium',
       ' anxiety', ' back_pain', ' blackheads', ' bladder_discomfort',
       ' blister', ' bloody_stool', ' blurred_and_distorted_vision',
       ' breathlessness', ' bruising', ' burning_micturition', ' chest_pain',
       ' chills', ' cold_hands_and_feets', ' constipation',
       ' continuous_feel_of_urine', ' continuous_sneezing', ' cough',
       ' cramps', ' dark_urine', ' dehydration', ' diarrhoea',
       ' dischromic _patches', ' distention_of_abdomen', ' dizziness',
       ' excessive_hunger', ' extra_marital_contacts', ' family_history',
       ' fatigue', ' foul_smell_of urine', ' headache', ' high_fever',
       ' hip_joint_pain', ' indigestion', ' irregular_sugar_level',
       ' irritation_in_anus', ' joint_pain', ' knee_pain',
       ' lack_of_concentration', ' lethargy', ' loss_of_appetite',
       ' loss_of_balance', ' mood_swings', ' movement_stiffness',
       ' muscle_wasting', ' muscle_weakness', ' nausea', ' neck_pain',
       ' nodal_skin_eruptions', ' obesity', ' pain_during_bowel_movements',
       ' pain_in_anal_region', ' painful_walking', ' passage_of_gases',
       ' patches_in_throat', ' pus_filled_pimples', ' red_sore_around_nose',
       ' restlessness', ' scurring', ' shivering', ' silver_like_dusting',
       ' skin_peeling', ' skin_rash', ' small_dents_in_nails',
       ' spinning_movements', ' spotting_ urination', ' stiff_neck',
       ' stomach_pain', ' sunken_eyes', ' sweating', ' swelling_joints',
       ' swelling_of_stomach', ' swollen_legs', ' ulcers_on_tongue',
       ' vomiting', ' watering_from_eyes', ' weakness_in_limbs',
       ' weakness_of_one_body_side', ' weight_gain', ' weight_loss',
       ' yellow_crust_ooze', ' yellowing_of_eyes', ' yellowish_skin',
       'itching'],
      dtype='object')
```

```
... array(['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',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo'], dtype=object)
```

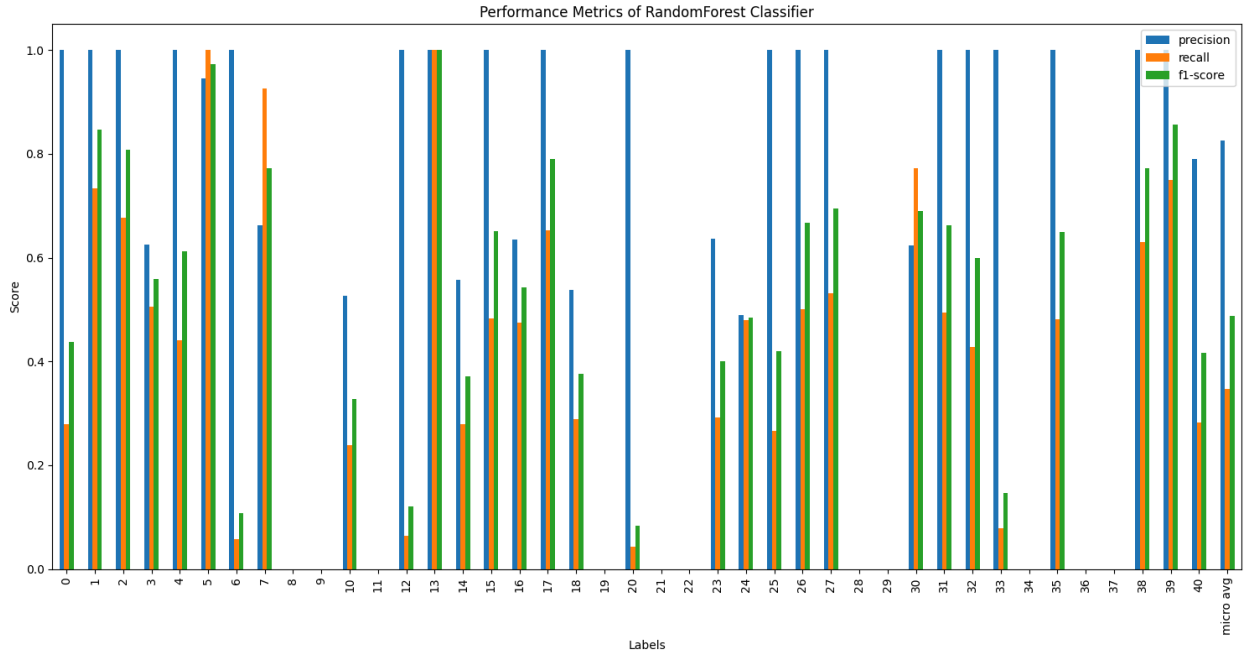
Using one hot encoder to extract the symptoms out

After that we should check to see if it is biased or not . so we use “ result_df['prognosis'].value_counts()” this code to know about it . Here are the results.

```
prognosis
Fungal infection                          480
Hepatitis C                               480
Hepatitis E                               480
Alcoholic hepatitis                       480
Tuberculosis                              480
Common Cold                               480
Pneumonia                                 480
Dimorphic hemmorhoids(piles)              480
Heart attack                              480
Varicose veins                            480
Hypothyroidism                            480
Hyperthyroidism                           480
Hypoglycemia                              480
Osteoarthristis                           480
Arthritis                                 480
(vertigo) Paroymsal  Positional Vertigo   480
Acne                                      480
Urinary tract infection                   480
Psoriasis                                 480
Hepatitis D                               480
Hepatitis B                               480
Allergy                                   480
hepatitis A                               480
GERD                                      480
...
Chicken pox                               480
Dengue                                    480
Typhoid                                   480
Impetigo                                  480
Name: count, dtype: int64
```



After we checked imbalance data we saw the original data are imbalanced .
and we should be testing the original data with some model. Here are some results that we had testing .



finally we have know about the Average Accuracy by using

“
```
train_index, test_index in kf.split(X):

   X_train, X_test = X.iloc[train_index], X.iloc[test_index]

   y_train, y_test = y_encoder[train_index], y_encoder[test_index]”
```

We get Average Accuracy: 0.33648373983739843.

Find the average accuracy saw that it is low average accuracy so we chose to change the data set to using training data . Here is a data set for using 5 rows and 133 columns.

In this training data have 20 features.

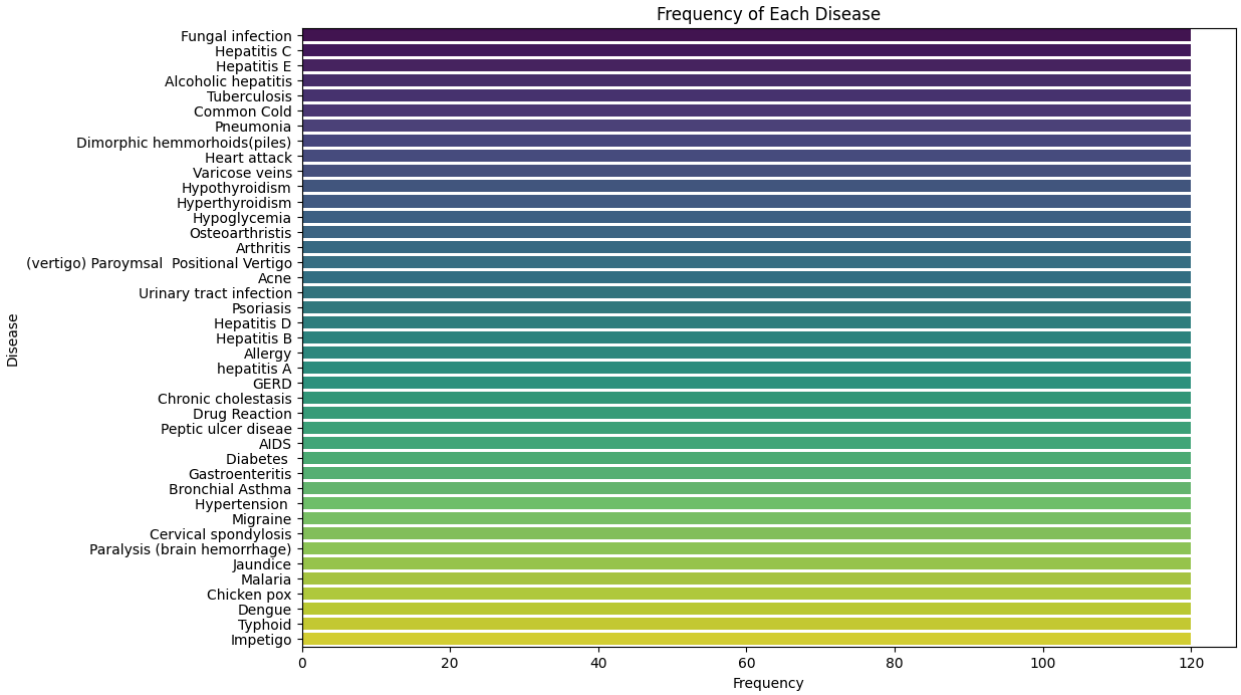
	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blacl
0	1	1	1	0	0	0	0	0	0	0	...	
1	0	1	1	0	0	0	0	0	0	0	...	
2	1	0	1	0	0	0	0	0	0	0	...	
3	1	1	0	0	0	0	0	0	0	0	...	
4	1	1	1	0	0	0	0	0	0	0	...	

5 rows × 133 columns

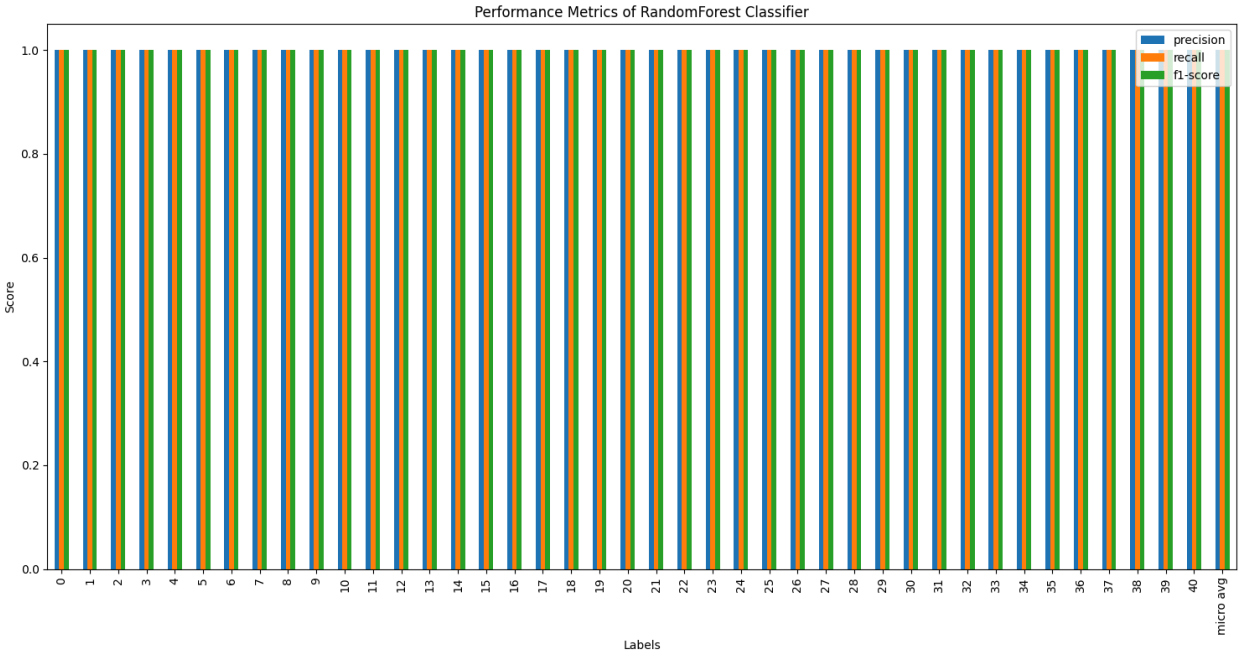
We already know about the features that we use to find Average accuracy but for step we have change data from category to numerical to study the model.

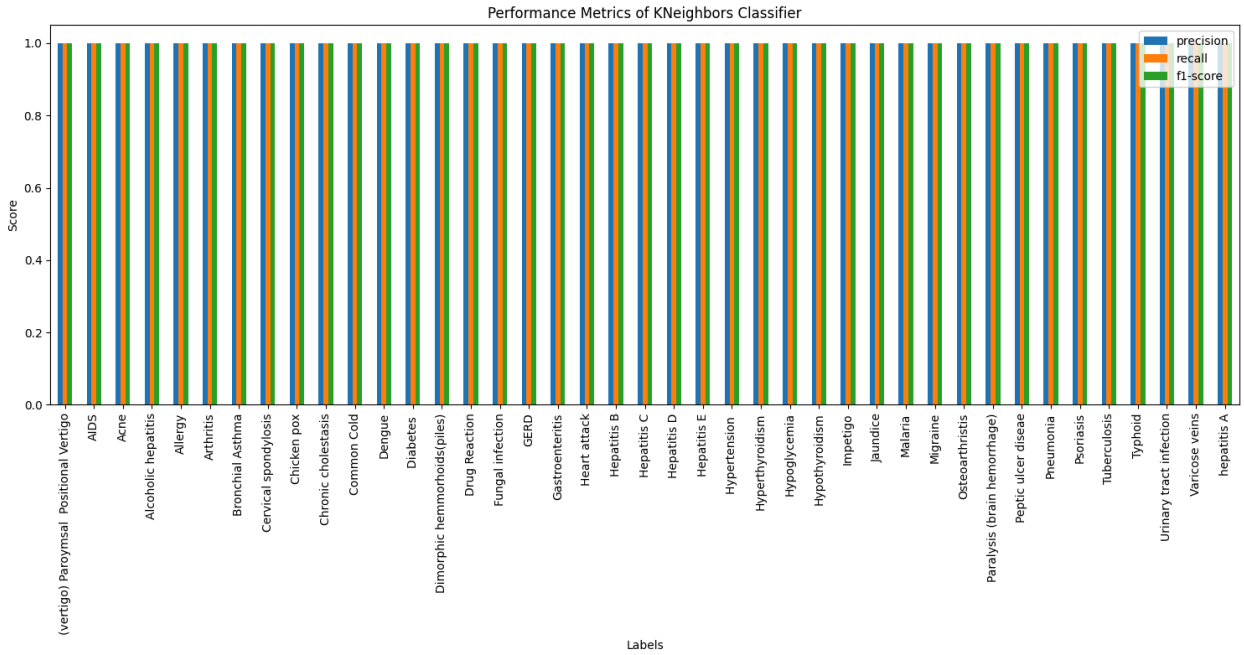
```
array(['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', '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',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo'], dtype=object)
```

We must check imbalance data same ask original data .



Here we are testing training data with some modes l. Here are some results that we had testing .





After we checked imbalance testing already we saw we get 1.0 of average accuracy of training data.

So Training data it's imbalanced. it has Average accuracy 1.0

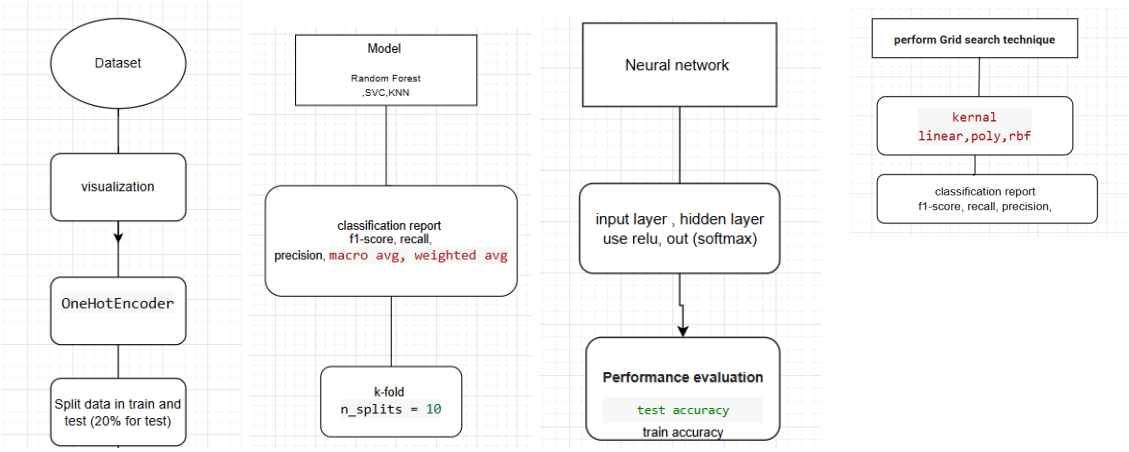
The average accuracy is ok for a machine so we chose to use this data .

METHODOLOGY

4.1 Roadmap for model training

To create a roadmap for our project on diagnosing diseases and providing recommendations using machine learning, we will follow a structured and detailed approach. Our method includes several critical steps to ensure that our work is clear and comprehensible to readers. First, we will begin with a dataset that includes information on various diseases. We will apply label encoding to convert categorical data into numerical values, making it suitable for machine learning algorithms. Next, we will split the dataset into training and testing sets to enable accurate model evaluation. We will explore and choose from multiple models, including K-Nearest Neighbors (KNN), Random Forest, Neural Networks, and Support Vector Machines (SVM). For the SVM, we will employ grid search techniques to optimize hyperparameters and improve model performance. We will evaluate our models using classification reports to measure key metrics such as precision, recall, and F1-score. Additionally, we will implement K-fold cross-validation to ensure the robustness and reliability of our models, helping to prevent overfitting. By

following this comprehensive roadmap, we aim to build an effective system for disease diagnosis and recommendation using machine learning.



4.2 Model Selection

For our project on diagnosing diseases and providing recommendations using machine learning, we have chosen to study and compare several models: Random Forest, Support Vector Classifier (SVC), K-Nearest Neighbors (KNN), and Neural Networks. These models were selected for their diverse strengths and applicability to our classification task. Additionally, we will use GridSearchCV to optimize the hyperparameters of our models, particularly for the SVC, to enhance their performance. By exploring these models and employing grid search techniques, we aim to identify the most effective model for accurately diagnosing diseases and providing relevant recommendations. This systematic approach will ensure that our project leverages the strengths of different machine learning algorithms to achieve robust and reliable results.

Data splitting

Train-test split is a fundamental technique in machine learning for evaluating the performance of a model. The primary purpose of this method is to assess how well the model generalizes to unseen data. The dataset is divided into two subsets: the training set and the testing set. Typically, the training set comprises a larger portion of the data (e.g., 70-80%), which is used to train the model. The remaining data (e.g., 20-30%) forms the testing set, which is used to evaluate the

model's performance. By separating the data into these two sets, we can get an unbiased estimate of how the model will perform on new, unseen data, ensuring that it has not simply memorized the training data but can generalize well to other data points.

4.3 Enhancement Technique

Random forests (RF)

Random Forest (RF) is an ensemble learning method used for classification and regression tasks. It creates multiple decision trees during training and combines their predictions to improve accuracy and reduce overfitting. RF uses random subsets of data (bagging) and features for each tree, enhancing diversity. It's robust, handles large datasets well, and provides feature importance estimates. RF is widely used in finance, healthcare, and other fields for its accuracy and versatility.

Let N be the number of decision trees in the forest.

For each tree i in N:

- Randomly sample a subset of the training data with replacement (bootstrap sample).
- Randomly select a subset of features.
- Build a decision tree using the sampled data and features.

For classification, the final prediction is given by the majority class among all trees:

$$\hat{y}_{\text{RF}} = \text{argmax}_c \sum_{i=1}^{N} 1(\hat{y}_i = c)$$

$\hat{y}_{\text{RF}}$ The Random Forest prediction. For classification, it's the class with the most votes among all trees.

$1(\hat{y}_i = c)$: The indicator function. It returns 1 if the prediction of the i $(\hat{y}_i)$ tree matches class c, and 0 otherwise.

For regression, the final prediction is the average of predictions from all trees:

$$\hat{y}_{\text{RF}} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i$$

Where N is the number of decision trees in the Random Forest

$\hat{y}_i$ the prediction of tree i

MultiOutputClassifier for SVC

The MultiOutputClassifier is a strategy for extending classifiers that do not natively support multi-output classification to handle multiple target variables. In the context of a Support Vector Classifier (SVC), it fits one classifier per target variable, effectively creating a separate SVC for each output.

Mathematically, if we have m target variables and we are using an SVC, the MultiOutputClassifier will train m independent SVC models.

X : is the input feature matrix with n samples and p features.

Y : is the output matrix with n samples and m target variables.

yj : is the j-th target variable column vector from Y.

For each target variable j (where $j = 1, 2, \ldots, m$):

Train an SVC fj on X to predict yj : $f_j(X) = y_j$

where fj is the SVC model for the j-th output variable.

The overall prediction for the MultiOutputClassifier is the concatenation of the predictions from each SVC:

$\hat{Y} = [f_1(X), f_2(X), \ldots, f_m(X)]$

In essence, the MultiOutputClassifier transforms the multi-output classification problem into m single-output classification problems, each of which is solved using a separate SVC. The final prediction for each sample is a vector containing the predictions of all m SVC models.

K-nearest neighbor (KNN)

The K-Nearest Neighbor (KNN) classifier is a nonparametric, instance-based learning algorithm used for classification and regression. This algorithm relies on the concept of nearest neighbors to make predictions. It classifies new instances based on the similarity measure, typically a distance metric.

Euclidean Distance Formula

The Euclidean distance between two points

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \text{ and } \mathbf{y} = (y_1, y_2, \ldots, y_n)$$

in an n-dimensional space is calculated as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Classification Process

1. **Determine the Value of K**: Select the number of nearest neighbors to use (the value of K).
2. **Compute Distances**: Calculate the distance between the new instance and all the instances in the training dataset using the Euclidean distance formula.
3. **Identify Nearest Neighbors**: Identify the K instances in the training dataset that are closest to the new instance.
4. **Make a Prediction**: For classification, the new instance is assigned to the class that is most common among the K nearest neighbors (majority voting).

4.4 Neural Network Integration

Neural network is a computational model inspired by the human brain's structure and function. It consists of layers of interconnected neurons that process data. The layers include an input layer, one or more hidden layers, and an output layer. Each neuron in a layer processes input data by applying a weighted sum and an activation function to produce an output.

Concepts

- Layers: Composed of an input layer, hidden layers, and an output layer.
- Neurons: Units that process input data using weights, biases, and activation functions.
- Weights and Biases: Parameters that are learned during training to minimize the error.
- Activation Functions: Functions like sigmoid, tanh, or ReLU that introduce non-linearity to the model.

Mathematical Representation

1. **Weighted Sum**: For neuron j in layer l:

$$z_j^l = \sum_{i-1}^{n} w_{ij}^l a_i^{l-1} + b_j^l$$ where

w_{ij}^l is the weight between neuron i in layer l-1 and neuron j in layer l.

a_i^{l-1} is the activation of neuron i in layer l-1.

b_j^l is the bias of neuron j in layer l.

ActivationFunction:

The activation of neuron j in layer l is:

$$a_j^l = \sigma(z_j^l)$$

where σ sigmaσ is the activation function, such as sigmoid

$(\sigma(z) = \frac{1}{1+e^{-z}})$, tanh $(\sigma(z) = \tanh(z))$

ReLU $(\sigma(z) = \max(0, z))$.

Workflow with Formulas

Input Layer : The input layer is the first layer of the neural network that receives the raw input data. Each neuron in this layer represents one feature of the input data.

Hidden Layer: Hidden layers are intermediate layers between the input and output layers where the network learns to detect features and patterns. Each neuron in a hidden layer performs a weighted sum of the inputs from the previous layer, adds a bias term, and applies an activation function.

Output Layer : The output layer is the final layer of the neural network that produces the output prediction. The number of neurons in this layer corresponds to the number of desired output values.

The loss function

used in neural networks quantifies the model's performance by measuring the difference between predicted and actual values.

Here are the formulas for some common loss functions used in neural networks:

Mean Squared Error (MSE):

$$L_{\text{MSE}} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

Mean Squared Logarithmic Error (MSLE):

$$L_{\text{MSLE}} = \frac{1}{m}\sum_{i=1}^{m}(\log(\hat{y}_i + 1) - \log(y_i + 1))^2$$

Binary Cross-Entropy Loss:

$$L_{\text{BCE}} = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Categorical Cross-Entropy Loss:

$$L_{\text{CCE}} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

Backpropagation:

Calculate Gradients: Compute the gradient of the loss with respect to weights and biases.

$$\frac{\partial L}{\partial w_{ij}^l} = \delta_j^l a_i^{l-1}$$

where δ_j^l is the error term for neuron j in layer l, calculated as:

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} \cdot \sigma'(z_j^l)$$

and $\sigma'(z_j^l)$ is the derivative of the activation function.

Update Weights and Biases

$$w_{ij}^l = w_{ij}^l - \eta \frac{\partial L}{\partial w_{ij}^l}$$

$$b_j^l = b_j^l - \eta \frac{\partial L}{\partial b_j^l}$$

where η is the learning rate.

Iterate: Repeat the forward propagation, loss calculation, and backpropagation steps for multiple epochs until the loss converges.

4.5 Hyperparameter Tuning

Grid search technique to find the best parameter for SVC

Grid search is essentially a methodical way of systematically searching through a specified parameter grid to find the combination of hyperparameters that yield the best performance for a machine learning model

Parameter Grid Definition:

Let P be the set of hyperparameter combinations to explore. Each combination is denoted as p, where p is a tuple representing a specific configuration of hyperparameters:

Each hyperparameter combination p consists of values for individual hyperparameters:

$$p = (C, \text{kernel}, \gamma)$$

Grid Search Optimization Objective:

Define an optimization objective to maximize the model's performance. This can be formulated as:

$$p^* = \text{argmax}_{p \in P} \, \text{CV}(p)$$

where p* is the optimal hyperparameter combination that maximizes cross-validated performance.

Let CV(p) represent the cross-validated performance (e.g., accuracy, F1 score) of the model trained with hyperparameters p.

4.6 Evaluation Metrics

Evaluation metrics are measures used to assess the performance of machine learning models. They help quantify how well a model is performing in terms of its predictions compared to the actual ground truth.

Accuracy measures the proportion of correctly classified instances out of all instances.

Precision measures the proportion of true positive predictions among all positive predictions made by the model.

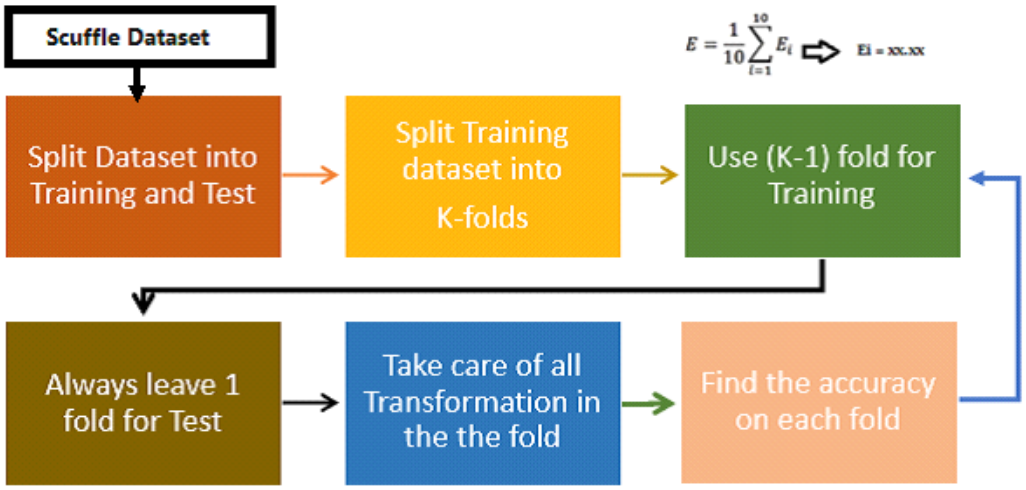
Recall measures the proportion of true positive predictions among all actual positive instances in the data.

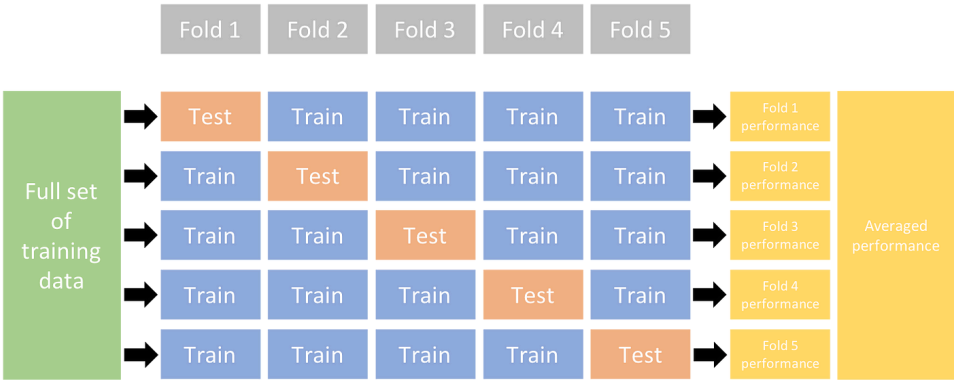
F1 Score is the harmonic mean of precision and recall. It provides a balanced measure between precision and recall.

		Actual class: Positive	Actual class: Negative	
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)	**Precision:** TP / (TP + FP)
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative	**Negative Predictive Value:** TN / (TN+FN)
		Recall or Sensitivity: TP / (TP + FN)	**Specificity:** TN / (TN + FP)	**Accuracy:** TP + TN / (TP + TN + FP + FN)

K-fold Cross-Validation

Is a technique used to assess the performance and generalization of machine learning models. It involves partitioning the dataset into k subsets/folds, training the model k times, each time using a different fold as the validation set and the remaining folds as the training set. The performance metrics are then averaged over the k iterations to obtain a more robust evaluation of the model's performance.



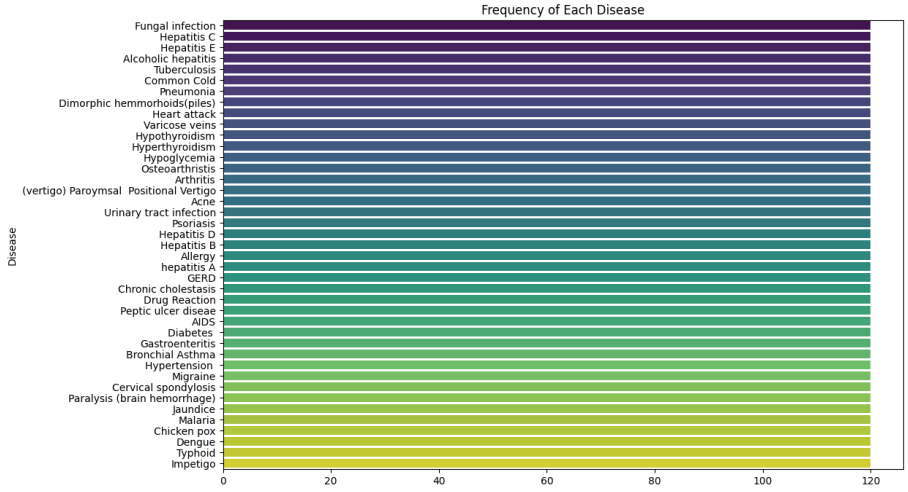


4.7 Result

We are using a dataset from training because we do not have any knowledge about medical and about *Symptoms of diseases* .

Using Training.csv data

Plotting the frequency of each disease



The target variable is `y_encoder`, which is the encoded version of the 'prognosis' column with feature 132 .

Split the Data:train 80% and test 20%

`X_train = (3936, 132)`

`X_test = (984, 132)`

`y_train = (3936, 41)`

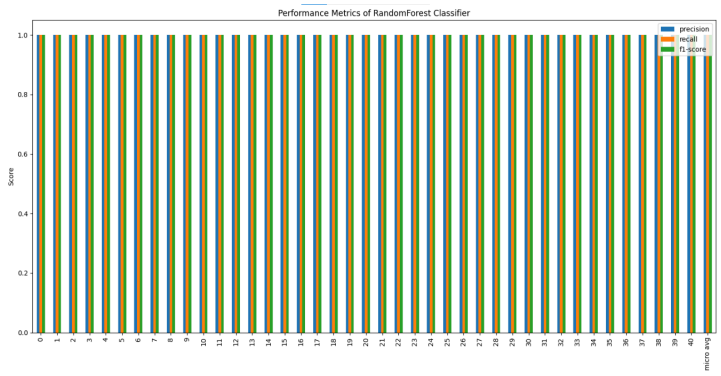
`y_test = (984, 41)`

Random forest

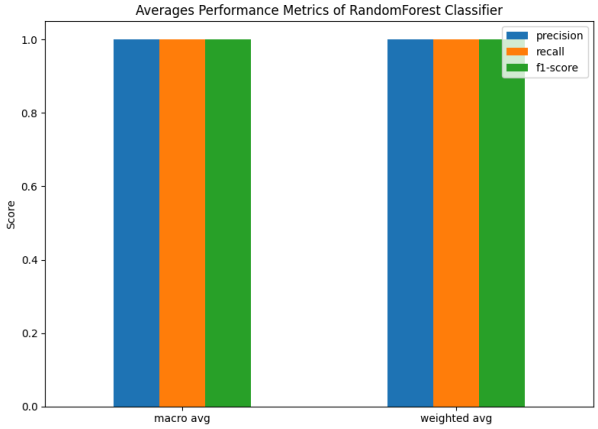
scikit-learn (sklearn):

rc = RandomForestClassifier(random_state=42)

Classification Report:Precision, recall, accuracy,F1 Score,



Plot the macro and weight averages separately



	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.0	1.0	1.0	18.0
AIDS	1.0	1.0	1.0	30.0
Acne	1.0	1.0	1.0	24.0
Alcoholic hepatitis	1.0	1.0	1.0	25.0
Allergy	1.0	1.0	1.0	24.0
Arthritis	1.0	1.0	1.0	23.0
Bronchial Asthma	1.0	1.0	1.0	33.0
Cervical spondylosis	1.0	1.0	1.0	23.0
Chicken pox	1.0	1.0	1.0	21.0
Chronic cholestasis	1.0	1.0	1.0	15.0
Common Cold	1.0	1.0	1.0	23.0
Dengue	1.0	1.0	1.0	26.0
Diabetes	1.0	1.0	1.0	21.0
Dimorphic hemmorhoids(piles)	1.0	1.0	1.0	29.0
Drug Reaction	1.0	1.0	1.0	24.0
Fungal infection	1.0	1.0	1.0	19.0
GERD	1.0	1.0	1.0	28.0
Gastroenteritis	1.0	1.0	1.0	25.0
Heart attack	1.0	1.0	1.0	23.0

Hepatitis B	1.0	1.0	1.0	27.0
Hepatitis C	1.0	1.0	1.0	26.0
Hepatitis D	1.0	1.0	1.0	23.0
Hepatitis E	1.0	1.0	1.0	29.0
Hypertension	1.0	1.0	1.0	25.0
Hyperthyroidism	1.0	1.0	1.0	24.0
Hypoglycemia	1.0	1.0	1.0	26.0
Hypothyroidism	1.0	1.0	1.0	21.0
Impetigo	1.0	1.0	1.0	24.0
Jaundice	1.0	1.0	1.0	19.0
Malaria	1.0	1.0	1.0	22.0
Migraine	1.0	1.0	1.0	25.0
Osteoarthristis	1.0	1.0	1.0	22.0
Paralysis (brain hemorrhage)	1.0	1.0	1.0	24.0
Peptic ulcer diseae	1.0	1.0	1.0	17.0
Pneumonia	1.0	1.0	1.0	28.0
Psoriasis	1.0	1.0	1.0	22.0
Tuberculosis	1.0	1.0	1.0	25.0
Typhoid	1.0	1.0	1.0	19.0
Urinary tract infection	1.0	1.0	1.0	26.0
Varicose veins	1.0	1.0	1.0	22.0
hepatitis A	1.0	1.0	1.0	34.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	984.0
weighted avg	1.0	1.0	1.0	984.0

kf = KFold(n_splits=10, shuffle=True, random_state=42)

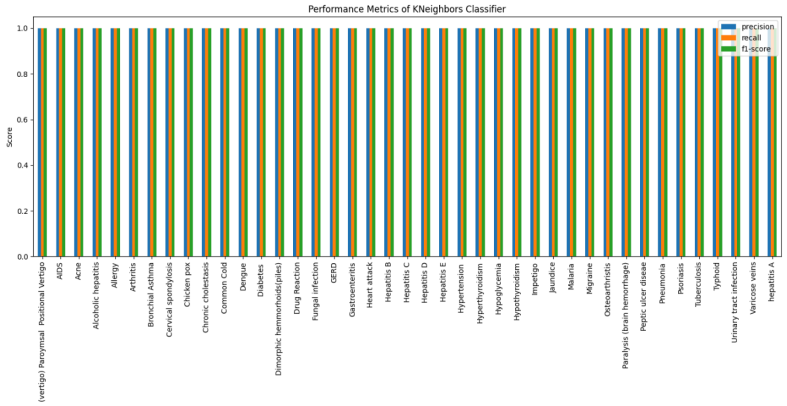
K-nearest neighbor (KNN)

scikit-learn (sklearn):

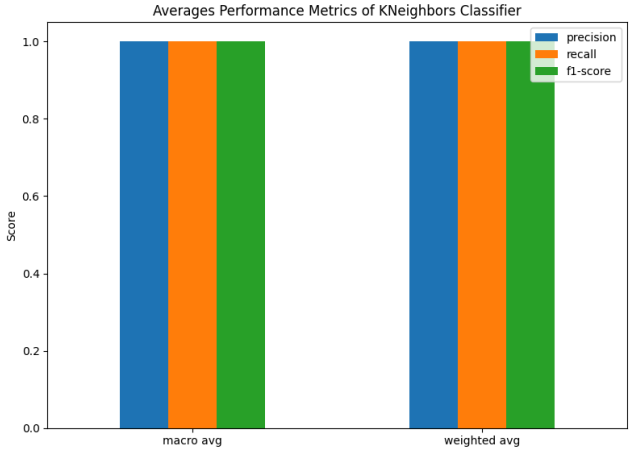
knn.fit(X_train, y_train)

adjust the number of neighbors k = n_neighbors =5

Classification Report:Precision, recall, accuracy,F1 Score,



Plot the macro and weight averages separately



	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.0	1.0	1.0	18.0
AIDS	1.0	1.0	1.0	30.0
Acne	1.0	1.0	1.0	24.0
Alcoholic hepatitis	1.0	1.0	1.0	25.0
Allergy	1.0	1.0	1.0	24.0

Arthritis	1.0	1.0	1.0	23.0
Bronchial Asthma	1.0	1.0	1.0	33.0
Cervical spondylosis	1.0	1.0	1.0	23.0
Chicken pox	1.0	1.0	1.0	21.0
Chronic cholestasis	1.0	1.0	1.0	15.0
Common Cold	1.0	1.0	1.0	23.0
Dengue	1.0	1.0	1.0	26.0
Diabetes	1.0	1.0	1.0	21.0
Dimorphic hemmorhoids(piles)	1.0	1.0	1.0	29.0
Drug Reaction	1.0	1.0	1.0	24.0
Fungal infection	1.0	1.0	1.0	19.0
GERD	1.0	1.0	1.0	28.0
Gastroenteritis	1.0	1.0	1.0	25.0
Heart attack	1.0	1.0	1.0	23.0
Hepatitis B	1.0	1.0	1.0	27.0
Hepatitis C	1.0	1.0	1.0	26.0
Hepatitis D	1.0	1.0	1.0	23.0
Hepatitis E	1.0	1.0	1.0	29.0
Hypertension	1.0	1.0	1.0	25.0
Hyperthyroidism	1.0	1.0	1.0	24.0
Hypoglycemia	1.0	1.0	1.0	26.0
Hypothyroidism	1.0	1.0	1.0	21.0
Impetigo	1.0	1.0	1.0	24.0
Jaundice	1.0	1.0	1.0	19.0
Malaria	1.0	1.0	1.0	22.0
Migraine	1.0	1.0	1.0	25.0
Osteoarthristis	1.0	1.0	1.0	22.0
Paralysis (brain hemorrhage)	1.0	1.0	1.0	24.0

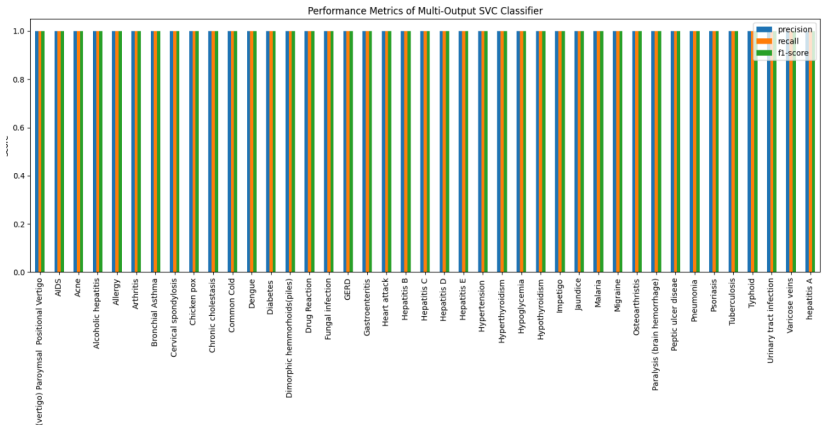
Peptic ulcer diseae	1.0	1.0	1.0	17.0
Pneumonia	1.0	1.0	1.0	28.0
Psoriasis	1.0	1.0	1.0	22.0
Tuberculosis	1.0	1.0	1.0	25.0
Typhoid	1.0	1.0	1.0	19.0
Urinary tract infection	1.0	1.0	1.0	26.0
Varicose veins	1.0	1.0	1.0	22.0
hepatitis A	1.0	1.0	1.0	34.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	984.0
weighted avg	1.0	1.0	1.0	984.0

Multi-Output Support Vector Classifier (SVC)

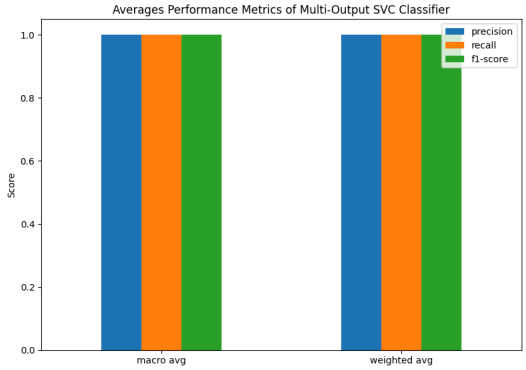
scikit-learn (sklearn):

multi_target_svc = MultiOutputClassifier(SVC(random_state=42))

Classification Report: Precision, recall, accuracy,F1 Score,



Plot the macro and weight averages separately



	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.0	1.0	1.0	18.0
AIDS	1.0	1.0	1.0	30.0
Acne	1.0	1.0	1.0	24.0
Alcoholic hepatitis	1.0	1.0	1.0	25.0
Allergy	1.0	1.0	1.0	24.0
Arthritis	1.0	1.0	1.0	23.0
Bronchial Asthma	1.0	1.0	1.0	33.0
Cervical spondylosis	1.0	1.0	1.0	23.0
Chicken pox	1.0	1.0	1.0	21.0
Chronic cholestasis	1.0	1.0	1.0	15.0
Common Cold	1.0	1.0	1.0	23.0
Dengue	1.0	1.0	1.0	26.0
Diabetes	1.0	1.0	1.0	21.0
Dimorphic hemmorhoids(piles)	1.0	1.0	1.0	29.0
Drug Reaction	1.0	1.0	1.0	24.0
Fungal infection	1.0	1.0	1.0	19.0
GERD	1.0	1.0	1.0	28.0
Gastroenteritis	1.0	1.0	1.0	25.0
Heart attack	1.0	1.0	1.0	23.0
Hepatitis B	1.0	1.0	1.0	27.0
Hepatitis C	1.0	1.0	1.0	26.0

Hepatitis D	**1.0**	**1.0**	**1.0**	**23.0**
Hepatitis E	**1.0**	**1.0**	**1.0**	**29.0**
Hypertension	**1.0**	**1.0**	**1.0**	**25.0**
Hyperthyroidism	**1.0**	**1.0**	**1.0**	**24.0**
Hypoglycemia	**1.0**	**1.0**	**1.0**	**26.0**
Hypothyroidism	**1.0**	**1.0**	**1.0**	**21.0**
Impetigo	**1.0**	**1.0**	**1.0**	**24.0**
Jaundice	**1.0**	**1.0**	**1.0**	**19.0**
Malaria	**1.0**	**1.0**	**1.0**	**22.0**
Migraine	**1.0**	**1.0**	**1.0**	**25.0**
Osteoarthristis	**1.0**	**1.0**	**1.0**	**22.0**
Paralysis (brain hemorrhage)	**1.0**	**1.0**	**1.0**	**24.0**
Peptic ulcer diseae	**1.0**	**1.0**	**1.0**	**17.0**
Pneumonia	**1.0**	**1.0**	**1.0**	**28.0**
Psoriasis	**1.0**	**1.0**	**1.0**	**22.0**
Tuberculosis	**1.0**	**1.0**	**1.0**	**25.0**
Typhoid	**1.0**	**1.0**	**1.0**	**19.0**
Urinary tract infection	**1.0**	**1.0**	**1.0**	**26.0**
Varicose veins	**1.0**	**1.0**	**1.0**	**22.0**
hepatitis A	**1.0**	**1.0**	**1.0**	**34.0**
accuracy	**1.0**	**1.0**	**1.0**	**1.0**
macro avg	**1.0**	**1.0**	**1.0**	**984.0**
weighted avg	**1.0**	**1.0**	**1.0**	**984.0**

cv_scores = cross_val_score(multi_target_svc, X_train, y_train, cv=kf, scoring='accuracy')

k-fold Cross-Validation: n_splits = 10

Neural network

using Pytorch Library

`torch`: This imports the PyTorch library, which is used for deep learning and neural network computations.

`torch.nn as nn`: This imports the neural network module from PyTorch, which includes classes for building neural network architectures.

`torch.optim as optim`: This imports the optimization module from PyTorch, which includes various optimization algorithms for training neural networks.

Split the Data:train 80% and test 20%

An instance of the `NeuralNetwork` class is created as `model`.

The loss function is defined as `nn.CrossEntropyLoss()`, suitable for multi-class classification tasks.

The optimizer is defined as `optim.SGD(model.parameters(), lr=0.1)`, using stochastic gradient descent (SGD) with a learning rate of 0.1.

neural network architecture :

Input Layer: 132 to 512 features.

Hidden Layer 1: 512 to 256 features. (relu)

Hidden Layer 2: 256 to 128 features. (relu)

Output Layer: 128 to 41 class probabilities. (softmax)

lr =0.1

num_epochs = 1000,

batch_size = 200

Model evaluation : train and test accuracy

GridSearch

hyperparameter tuning using GridSearchCV with scikit-learn

`param_grid`: This dictionary defines the hyperparameter grid for GridSearchCV. It specifies different values to try for the hyperparameters 'C' (regularization parameter), 'gamma' (kernel coefficient), 'kernel' (kernel type), and 'degree' (polynomial degree for 'poly' kernel).

gridsearch = GridSearchCV(estimator=model_SVM, param_grid=param_grid)
gridsearch.fit(X_train, y_train)

Split the Data:train 80% and test 20%

Model Selection: SVC

`best_params = gridsearch.best_params_`: After GridSearchCV completes, this line retrieves the best hyperparameters found during the search.

param_grid = { 'C': [0.1, 1, 10, 100],

'gamma': [1, 0.1, 0.01, 0.001],

'kernel': ['rbf', 'poly', 'linear'],

'degree': [2, 3, 4, 5]

Best parameters found: {'C': 0.1, 'degree': 2, 'gamma': 1, 'kernel': 'rbf'}

Classification Report:Precision, recall, accuracy,F1 Score.

Accuracy: 1.0

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	18
1	1.00	1.00	1.00	30
2	1.00	1.00	1.00	24
3	1.00	1.00	1.00	25
4	1.00	1.00	1.00	24
5	1.00	1.00	1.00	23
6	1.00	1.00	1.00	33
7	1.00	1.00	1.00	23

8	1.00	1.00	1.00	21
9	1.00	1.00	1.00	15
10	1.00	1.00	1.00	23
11	1.00	1.00	1.00	26
12	1.00	1.00	1.00	21
13	1.00	1.00	1.00	29
14	1.00	1.00	1.00	24
15	1.00	1.00	1.00	19
16	1.00	1.00	1.00	28
17	1.00	1.00	1.00	25
18	1.00	1.00	1.00	23
19	1.00	1.00	1.00	27
20	1.00	1.00	1.00	26
21	1.00	1.00	1.00	23
22	1.00	1.00	1.00	29
23	1.00	1.00	1.00	25
24	1.00	1.00	1.00	24
25	1.00	1.00	1.00	26
26	1.00	1.00	1.00	21
27	1.00	1.00	1.00	24
28	1.00	1.00	1.00	19
29	1.00	1.00	1.00	22
30	1.00	1.00	1.00	25
31	1.00	1.00	1.00	22
32	1.00	1.00	1.00	24
33	1.00	1.00	1.00	17
34	1.00	1.00	1.00	28

```
          35       1.00      1.00      1.00        22
          36       1.00      1.00      1.00        25
          37       1.00      1.00      1.00        19
          38       1.00      1.00      1.00        26
          39       1.00      1.00      1.00        22
          40       1.00      1.00      1.00        34

    accuracy                           1.00       984
   macro avg       1.00      1.00      1.00       984
weighted avg       1.00      1.00      1.00       984
```

Table of the result in our model above

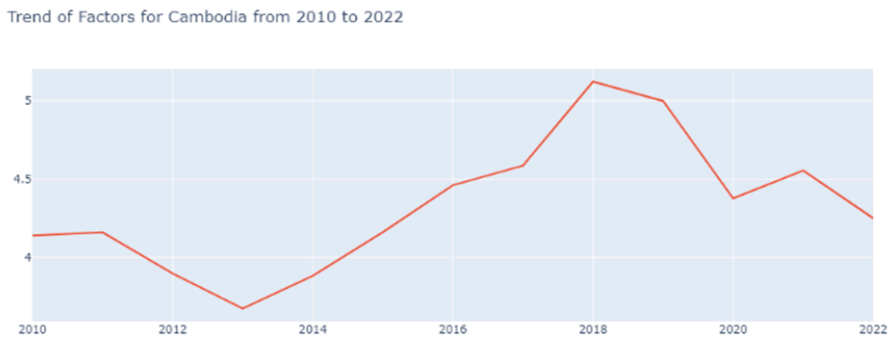
Model	Precision	recall	f1-score	Acuracy	k-fold(n_split=10)
Ramdom Forest	1	1	1	1	1
SVC	1	1	1	1	1
KNN	1	1	1	1	1

Model	acuracy	Hyperparameter
Neural network	0.83	lr = 0.1 (132,512,256,128,41) , batch_size=200, optimizer = optim.SGD , number_epoch =1000
Grid Search for SVC	1	{'C': 0.1, 'degree': 2, 'gamma': 1, 'kernel': 'rbf'}

All our models did really well, except for the neural network which scored 0.83 while the others got a perfect 1. They all used the same preprocessed data, so it's surprising that the neural network didn't do as well. This could be because of how it was set up or trained differently from the rest. Even though they started with the same data, the neural network's performance was a bit different, showing there might be ways to make it work better.

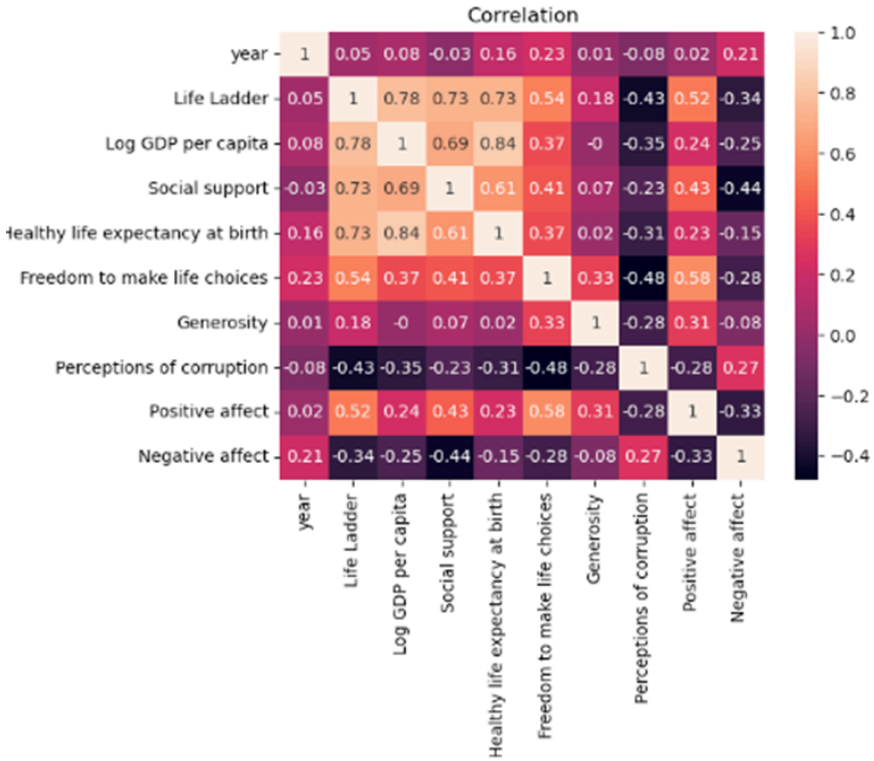
6. Impact and Conclusion

6.1. Cambodia's happiness score



In the last study of the happiness score in Cambodia , we noticed that the trend has been going up and down over time , showing that there are problems affecting people's happiness. To find out what is causing these changes , we looked at different factors, including logged GDP per capita social support, healthy life expectancy , freedom to make life choices, generosity and perception of corruption.

After that we study the correlation to see which factors affect the score happiness the most.



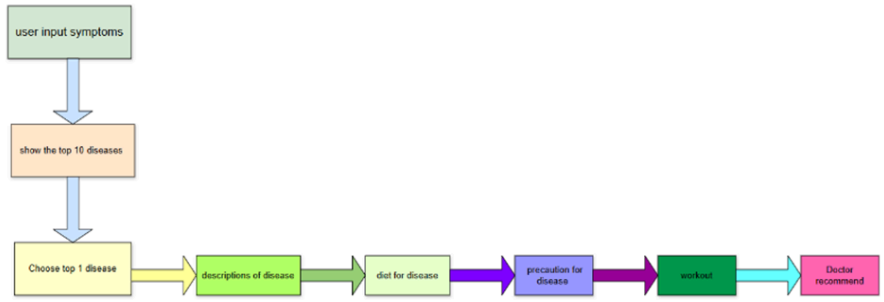
According to the graph , GDP, social support and healthy life expectancy are the strongest correlation among the factors.

After doing the research and discussion we found the final solution which is disease diagnose system.

The system helps with early detection and proper management of disease, leading to better health for the people by using modern medical tools and technology, the system will identify disease early and allow for timely treatment that can prevent serious health problems and reduce the burden on individuals and their families. However one of the concerns that users may have is that how does our system work ?

6.1 . Application Overview

To address the healthcare issue, we propose the implementation of a disease diagnose system.



How it works:

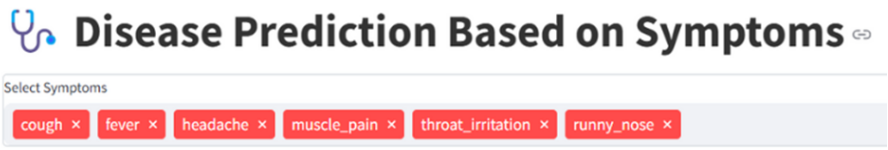
1. Symptom Input: Individuals can input their symptoms into the system using a user-friendly interface.

2. Prediction: The system analyzes the symptoms using advanced AI and machine learning algorithms to predict potential diseases and show a dashboard which include the top 10 diseases.

3. Recommendations: Provides users with potential diagnoses and recommendations and information such as description of disease, diet, precaution, workout and most importantly is the suggestion of doctor who can cure the disease suited in Cambodia.

6.3. Demo of disease diagnose system:

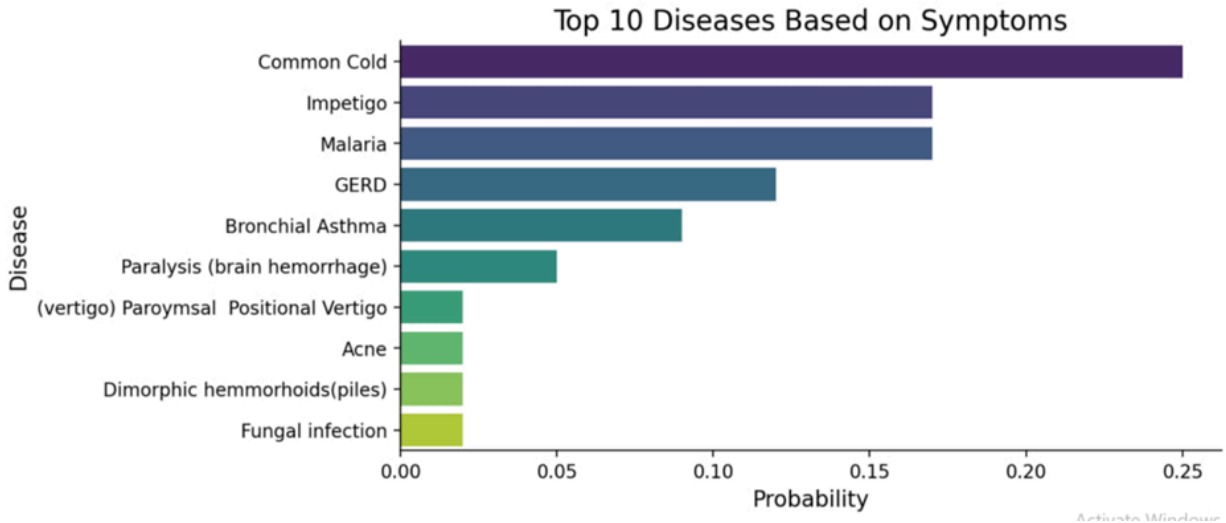
1. User input symptoms



The interface includes a section labeled “select symptoms” where multiple symptoms are displayed as red tags with white text.

At the very first step user can use the symptoms and input their illness symptoms to the systems. The symptoms listed are "cough," "fever," "headache," "muscle pain," "throat irritation," and "runny nose." Each symptom is accompanied by a small "x" icon, indicating that the user can remove the symptom from the selection. The interface is designed to allow users to input their symptoms for disease prediction purposes.

2. Prediction



a bar chart titled "Top 10 Diseases Based on Symptoms." It displays a ranked list of diseases along the y-axis and their corresponding probabilities along the x-axis. The diseases are listed from top to bottom as follows:

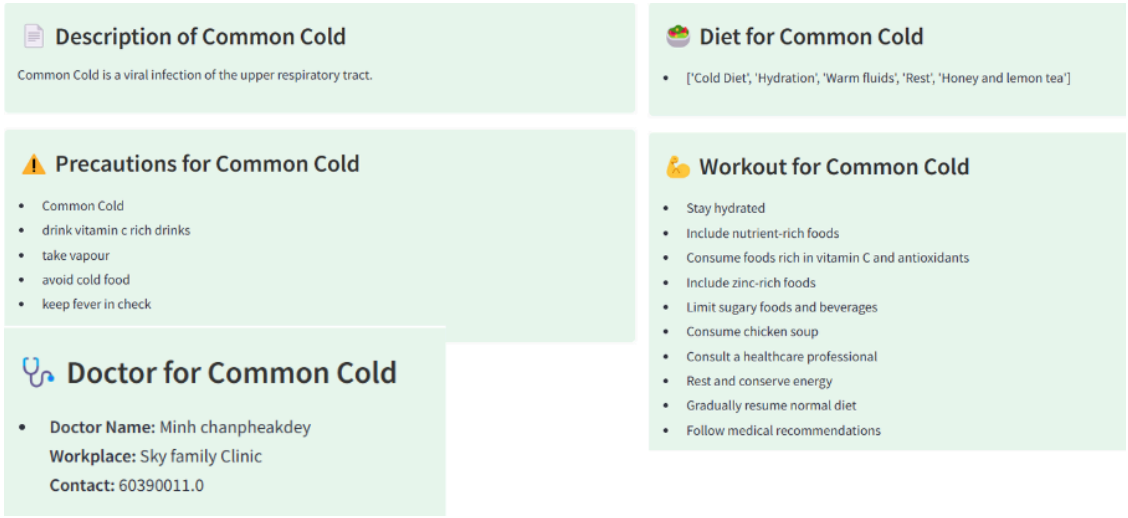
1. Common Cold
2. Impetigo

3. Malaria
4. GERD
5. Bronchial Asthma
6. Paralysis (brain hemorrhage)
7. (vertigo) Paroxysmal Positional Vertigo
8. Acne
9. Dimorphic Hemorrhoids (piles)
10. Fungal infection

The probabilities are shown as horizontal bars, with the Common Cold having the highest probability and Fungal infection having the lowest among the top ten. The colors of the bars range from purple for the highest probability to green for the lowest, providing a visual gradient of disease likelihood based on the symptoms.

3. Recommendation :

after prediction will have show the description of the disease and recommendation below the dashboard



There are sections of a webpage providing information about the common cold. It is divided into four quadrants, each with a different focus related to managing the common cold such as description of common cold , diet for common cold , precaution for common cold, workout for common cold and last but not least is the doctor recommendation in Cambodia of specific disease.

Each section is visually distinguished and uses icons to represent the type of information provided such as description of common cold which educate the user about what the disease is, including its cause, symptoms, and general characteristics. On the left side is diet for common cold offers dietary recommendation that can help alleviate symptoms and support the immune system. Below description is precaution lists preventive measures to avoid catching or spreading the disease. Next to that is the workout section that provides guidelines on physical activity and general health practice during a disease. Lastly is doctor recommendation in Cambodia for specific disease which offers localized medical advice and recommendations for healthcare providers in Cambodia for specific disease including the common cold.

Having these sections ensures that users get a holistic view of managing the common cold, from understanding the disease to taking practical steps for treatment and prevention. It empowers individuals with knowledge and resources to handle the illness effectively, promoting better health outcomes.

6.2. future work and potential enhancement :

Despite the progress made, more work is necessary to enhance the system's capabilities such that

Enhanced AI and Machine Learning Models:

- Development of more sophisticated AI algorithms that can learn from a wider array of data sources, including genetic, environmental, lifestyle, and social determinants of health, to improve diagnostic accuracy

Blockchain for Data Security and Integrity:

- Implementing blockchain technology to ensure the security, integrity, and traceability of patient data, enhancing patient trust and compliance with data protection regulations

AI-Driven Personalized Treatment Recommendations:

- Using AI to not only diagnose diseases but also to recommend personalized treatment plans based on a patient's unique genetic makeup, lifestyle, and health history

Collaboration with Research and Development:

- Fostering closer collaboration between diagnostic system developers and medical researchers to continuously integrate the latest scientific discoveries and clinical insights into diagnostic tools

Continuous Learning Systems:

- Designing diagnostic systems that continuously learn and adapt from new data, including patient outcomes and emerging medical research, to keep improving their accuracy and relevance

6.3. conclusion :

To conclude everything that has been stated so far, the implementation of a disease diagnosis system in Cambodia presents a transformative solution to address the critical healthcare issues that contribute to low happiness scores. This system leverages advanced technologies like AI and machine learning to provide early, accurate, and accessible disease detection, which can significantly improve individual and public health outcomes.

By integrating a disease diagnosis system, Cambodia can create a more resilient and responsive healthcare system, ultimately fostering a healthier, more productive, and happier society. This innovation addresses immediate health concerns while also laying the foundation for long-term improvements in public health, economic stability, and overall quality of life.

Reference

1.https://docs.ultralytics.com/guides/kfold-cross-validation/

2.https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/

3.https://www.analyticsvidhya.com/blog/2020/12/decluttering-the-performance-measures-of-classification-models/

4.https://medium.com/@rdhawan201455/knn-k-nearest-neighbour-algorithm-maths-behind-it-and-how-to-find-the-best-value-for-k-6ff5b0955e3d

5.https://en.wikipedia.org/wiki/Random_forest

6.https://www.analyticsvidhya.com/blog/2021/04/estimation-of-neurons-and-forward-propagation-in-neural-net/

7.https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/

8.https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html