

DISEASE CLASSIFICATION USING MACHINE LEARNING

by

RUDHREESH KUMAAR R B R	20BLC1103
BEERELLY MANASA REDDY	19MIA1009
NAMITHA	19MIA1031
HARSHITHA. K	19MIA1026

A project report submitted to

Dr. VETRIVELAN. P

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE3506 – ESSENTIALS OF DATA ANALYTICS

in

**B.Tech. ELECTRONICS AND COMMUNICATION
ENGINEERING**

Vandalur – Kelambakkam Road

Chennai – 600127

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report entitled “**DISEASE CLASSIFICATION USING MACHINE LEARNING**” is a bonafide work of **RUDHREESH KUMAAR R B R – 20BLC1103, Namitha – 19MIA1031, Harshitha – 19MIA1026 and Manasa Reddy – 19MIA1009** who carried out the Project work under my supervision and guidance for **CSE3506-Essentials of Data Analytics**.

Dr. VETRIVELAN.P

Professor & ACOE,

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

This work focuses on the prognosis of diseases based on symptoms using machine learning techniques. Typically, patients consult a general physician and then are referred to a specialist, which can be a time-consuming and costly process. The existing method involves a preliminary examination by the general physician to diagnose the symptoms, which is not always accurate. As a result, the need for a more efficient and accurate diagnosis method arises. In this work, the authors propose a machine learning-based approach to diagnose diseases based on symptoms. The proposed method takes input from the patient about a few symptoms and then asks the patient to respond with yes or no for a few critical symptoms. The method then provides a prognosis based on the symptoms reported. The results obtained are promising, and the proposed method provides an acceptable level of accuracy for medical purposes. This work shows that accurate disease prognosis can be achieved using machine learning techniques based on symptom data. In conclusion, the authors have demonstrated the potential of machine learning-based approaches for the prognosis of diseases based on symptoms, which could reduce the time and cost associated with traditional diagnosis methods. They have also shown that a high level of accuracy can be achieved with symptom data, which can help in making informed medical decisions.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Vetrivelan. P**, Professor & ACOE, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

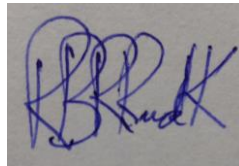
We are extremely grateful to **Dr. Susan Elias**, Dean of School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.



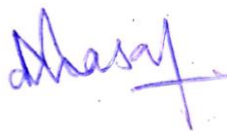
NAMITHA



RUDHREESH KUMAAR R B R



HARSHITHA K



MANASA REDDY

TABLE OF CONTENTS

SL. NO.	TITLE	PAGE NO.
	ABSTRACT	3
	ACKNOWLEDGEMENT	4
1	INTRODUCTION	7-9
1.1	OBJECTIVES	8
1.2	BENEFITS	8-9
1.3	FEATURES	9
2	BLOCK DIAGRAM AND ITS COMPONENTS	10-13
2.1	BLOCK DIAGRAM	10
2.2	LIBRARIES USED	11
2.3	ML MODELS USED	11-12
2.4	SOFTWARE SPECIFICATIONS	12

3	SYSTEM IMPLEMENTATION AND ANALYSIS	13-25
3.1	DATA REPROCESSING	13-14
3.2	PREDICTION MODELS	15-22
3.3	PROGNOSIS	23-24
3.4	RESULTS AND INFERENCE	25
4	CONCLUSION AND FUTURE WORK	26-27
	CONCLUSION	26
	FUTURE WORK	26-27
5	APPENDIX	28-31
	REFERENCES	28
	BIO-DATA	29-30
	SHAREABLE LINKS	31

CHAPTER 1

INTRODUCTION

In today's digital age, access to health information is more prevalent than ever before. As a result, more and more people are turning to online resources to learn about different diseases, diagnoses, and treatments. However, the sheer amount of medical information available can be overwhelming and confusing for laymen, especially when it comes to understanding the complex medical vocabulary used to describe diseases and symptoms. This can lead to misinterpretation of information, incorrect self-diagnosis, and delayed treatment, potentially resulting in serious health consequences.

To address this issue, a recommendation system that can help doctors and patients quickly and accurately identify diseases based on symptoms is needed. This system should be user-friendly, easily accessible, and capable of adapting to the unique requirements of the healthcare industry. Machine learning algorithms, which have proven to be highly effective in a variety of fields, can be used to develop such a system.

The Disease Prediction System using Symptoms and Machine Learning algorithms is a project that seeks to provide a user-friendly, accurate, and reliable solution for early detection of diseases based on symptoms entered by the user. The project uses a dataset consisting of two main columns, "Disease" and "Symptoms," which have been preprocessed to facilitate easy classification of data. The dataset is used to train the system using four different machine learning algorithms: Decision Tree, Random Forest, Naive Bayes, and K-Nearest Neighbour, all of which have been shown to produce an accuracy rate of over 90%.

The system is designed to be accessible and easy to use, with a graphical user interface (GUI) that enables users to enter symptoms and receive personalised disease predictions based on their input. The system saves the results of each user with their name for future preferences. The project's success demonstrates the

potential of machine learning and artificial intelligence in the healthcare industry and provides a framework for future developments in this field.

1.1 OBJECTIVES

- To develop a machine learning model that predicts the disease based on the symptoms entered by the user.
- To provide a user-friendly interface (GUI) for easy interaction with the system.
- To implement various machine learning algorithms like Decision Tree, Random Forest, Naive Bayes, and K-Nearest Neighbor for disease prediction and compare their performance.
- To store the user's data and results in a database for future preferences.
- To provide a recommendation system for doctors and medicine using review mining and save time.

1.2 BENEFITS

Early Detection of Diseases: The project enables early detection of diseases based on the symptoms entered by the user, which can lead to early treatment and better outcomes.

Accuracy: The machine learning algorithms used in this project provide high accuracy in disease prediction, making it a reliable tool for healthcare professionals.

Time-Saving: The system saves time for healthcare professionals by providing disease recommendations based on the symptoms entered by the user.

User-Friendly: The GUI interface of the system is user-friendly, making it easy for users to enter their symptoms and get disease recommendations.

Personalization: The system allows storing user data and results in a database for future preferences, which enables personalised disease prediction based on the user's history.

Cost-Effective: The project is cost-effective compared to traditional disease diagnosis methods that require expensive tests and consultations with healthcare professionals.

Accessibility: The system can be accessed from anywhere with an internet connection, making it accessible to people in remote areas with limited access to healthcare facilities.

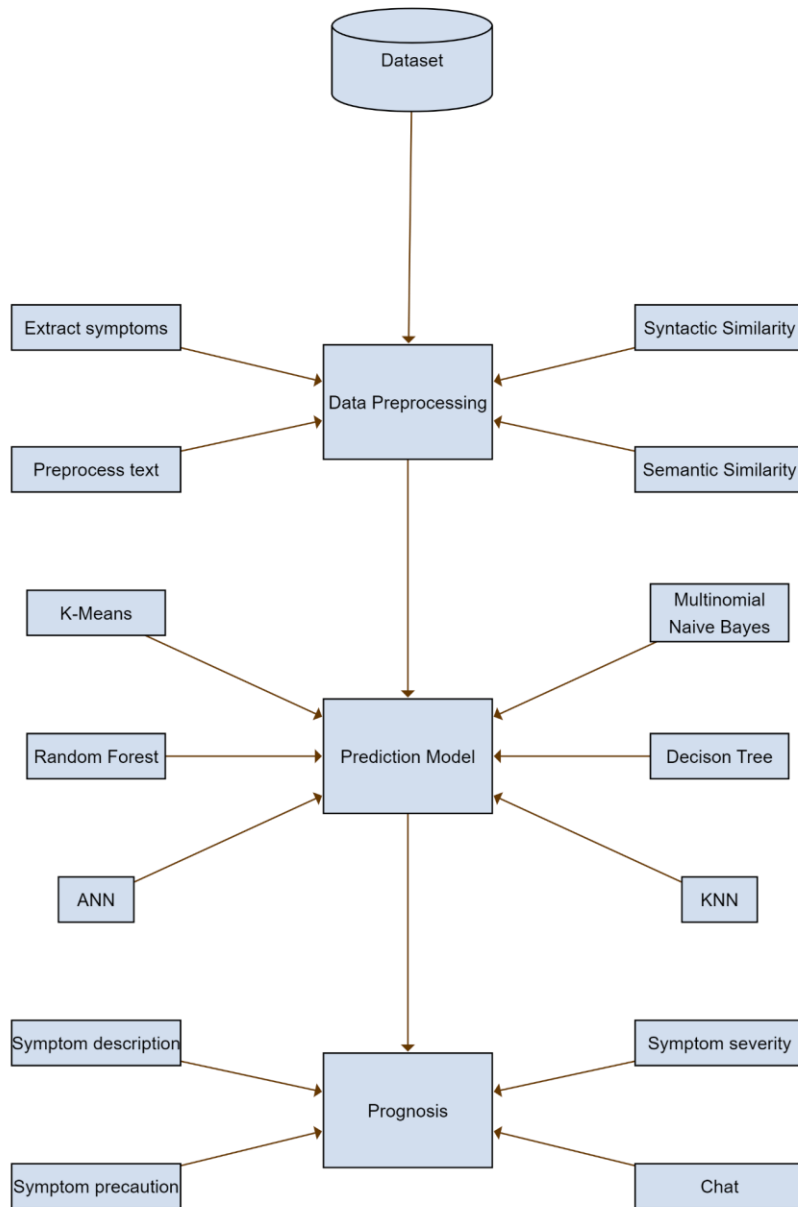
1.3 FEATURES

1. Disease prediction based on symptoms entered by the user using machine learning algorithms.
2. User-friendly GUI interface for easy interaction with the system.
3. Multiple machine learning algorithms (Decision Tree, Random Forest, Naive Bayes, and K-Nearest Neighbour) for disease prediction, providing accuracy and reliability.
4. Storage of user data and results in a database for future preferences and personalised disease prediction.
5. Cost-effective and accessible healthcare solution, enabling early detection of diseases, saving time, and improving outcomes.

CHAPTER 2

BLOCK DIAGRAM AND ITS COMPONENTS

2.1 BLOCK DIAGRAM



2.2. LIBRARIES USED

This project utilises standard libraries for database analysis and model creation. The following libraries are used: nltk, numpy, pandas, and sklearn.

NLTK (Natural Language Toolkit) is a Python library that provides a suite of tools and resources for natural language processing (NLP). It offers functionalities such as tokenization, stemming, tagging, parsing, and more, making it a valuable resource for researchers and developers working with text data.

Numpy is a core library for scientific computing in Python that provides powerful tools for handling multi-dimensional arrays. It is useful for creating n-dimensional arrays and processing their contents using various methods like sum, mean, and max.

Pandas is a popular Python library used for data analysis. It provides optimized performance with source code written in C or Python. It offers two ways to analyze data: series and dataframes. Dataframes are used extensively in this project for manipulating and preprocessing datasets.

Sklearn is an open-source Python library that implements a wide range of machine-learning algorithms, pre-processing techniques, cross-validation, and visualisation tools. It offers various classification, regression, and clustering algorithms like decision trees and KNN. In this project, we used sklearn's built-in classification algorithms, cross-validation, and visualisation features like accuracy score and confusion matrix.

2.3 ML Models Used:

Decision Tree: Decision Tree is a simple and widely used classification algorithm. It works by splitting the dataset into smaller subsets, eventually creating a tree-like model of decisions and their possible consequences. Each node in the decision tree represents a feature and each branch represents a decision or rule. The model uses a set of if-else decision rules to predict the target variable. Decision tree models are easy to understand and interpret, and can handle both numerical and categorical data.

Random Forest: Random Forest is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. The random forest algorithm creates multiple decision trees, and then selects the best performing tree by means of voting. The model is highly accurate and works well for large datasets with multiple features.

Gaussian Naïve Bayes: Gaussian Naïve Bayes is a probabilistic algorithm that works on the Bayes theorem of probability. It assumes that the features are independent and have a Gaussian distribution. The model calculates the probability of each class for a given set of features, and then selects the class with the highest probability. Naïve Bayes is a simple, fast and highly scalable algorithm and works well for high dimensional datasets.

K-Nearest Neighbors (KNN): KNN is a non-parametric algorithm used for classification and regression. It works by finding the K-nearest neighbours of a data point from the training set, and then assigns the class to the data point based on the most common class among its neighbours. KNN is simple and easy to implement and works well for small datasets with fewer features. It can handle both numerical and categorical data, but can be computationally expensive for large datasets.

K-means: K-means is a clustering algorithm used to group similar data points in a dataset. It works by iteratively assigning each data point to a cluster based on the closest centroid and then updating the centroid based on the mean of the data points in that cluster. The algorithm converges when the assignment of data points to clusters no longer changes. K-means is widely used in data mining, image segmentation, and market segmentation.

2.4 SOFTWARE SPECIFICATIONS

- Python 3.9
- Google Colab

CHAPTER 3

SYSTEM IMPLEMENTATION AND ANALYSIS

3.1 DATA PREPROCESSING

3.1.1 GET SYMPTOMS

We perform data cleaning and preprocessing of symptom names, check if they have synonyms in the WordNet lexical database, and create two separate lists of symptom names based on whether they have synonyms or not. This methodology can be used to identify symptoms that may need to be further analysed or have their synonyms added to the database.

3.1.2 PREPROCESS TEXT

We perform preprocessing of symptom data using spacy and stop words, generate a new list of preprocessed symptom names, and create a dictionary to map preprocessed symptom names to their corresponding original column names. This methodology can be used for further analysis or modelling of the symptom data.

3.1.3 SYNTACTIC SIMILARITY

The code block defines several functions for checking similarity between symptoms and finding patterns in a list of symptoms.

The *jaccard_set* function computes Jaccard similarity between two input strings based on the intersection and union of their tokens.

The *syntactic_similarity* function takes a symptom string and a list of all preprocessed symptoms as inputs, and returns the most similar symptoms from the list based on Jaccard similarity. It also checks if the input symptom already exists in the list and returns it if true.

The *powerset* function returns all possible subsets of a given set. The sort function sorts a given list based on the length of its elements and returns it. The permutations function returns all permutations of the input list as a list of strings.

The *DoesExist* function checks if a given symptom already exists in the preprocessed symptom list. It does so by generating all possible permutations of all possible subsets of the input symptom, and checking if any of them exist in the preprocessed symptom list.

The *check_pattern* function takes a string pattern and a list of preprocessed symptoms as inputs, and returns a list of all preprocessed symptoms that match the pattern using regular expressions.

3.1.4 SEMANTIC SIMILARITY

This code block includes functions to perform semantic similarity between symptoms, suggest synonyms for a given symptom, and create a one-hot vector encoding of a patient's symptoms.

The *WSD* function uses the Lesk algorithm from the nltk library to perform word sense disambiguation on a given word using the provided context. The *semanticD* function calculates the semantic similarity between two documents (symptoms) using the *WSD* function to disambiguate words and the Wu-Palmer similarity measure.

The *semantic_similarity* function takes a symptom and a corpus of symptoms and returns the most similar symptom from the corpus along with the similarity score. The *suggest_syn* function takes a symptom and suggests synonyms for it using WordNet and the *semantic_similarity* function.

The *OHV* function takes a list of symptoms and a list of all symptoms and returns a one-hot vector encoding of the symptoms. The *contains* function checks if a list of symptoms is present in another list of symptoms. The *possible_diseases* function takes a list of symptoms and returns a list of possible diseases that match all the symptoms in the list. Finally, the *symVONdisease* function takes a disease name and returns a list of symptoms associated with that disease.

3.2 PREDICTION MODELS

3.2.1 K-NEAREST NEIGHBOURS

K-nearest neighbours (K-NN) is a simple yet powerful machine learning algorithm that can be used for both regression and classification tasks. Given a new input data point, the algorithm finds the K training data points that are closest to it in feature space, and assigns the output label of the majority class in these K neighbours to the new point. The value of K is a hyperparameter that can be tuned to optimise performance on a given dataset. One of the key advantages of K-NN is its simplicity and interpretability, as it does not require explicit model training and can easily handle nonlinear decision boundaries. However, K-NN can be sensitive to noisy or irrelevant features and may suffer from the curse of dimensionality, where the performance degrades rapidly as the number of features increases. Additionally, K-NN can be computationally expensive, particularly for large datasets or high-dimensional feature spaces.

```
Best hyperparameters: {'n_neighbors': 1, 'weights': 'uniform'}  
Best accuracy score: 1.0  
Classification report:
```


Classification report:

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.00	1.00	1.00	1
AIDS	1.00	1.00	1.00	1
Acne	1.00	1.00	1.00	1
Alcoholic hepatitis	1.00	1.00	1.00	1
Allergy	1.00	1.00	1.00	1
Arthritis	1.00	1.00	1.00	1
Bronchial Asthma	1.00	1.00	1.00	1
Cervical spondylosis	1.00	1.00	1.00	1
Chicken pox	1.00	1.00	1.00	1
Chronic cholestasis	1.00	1.00	1.00	1
Common Cold	1.00	1.00	1.00	1
Dengue	1.00	1.00	1.00	1
Diabetes	1.00	1.00	1.00	1
Dimorphic hemmorhoids(piles)	1.00	1.00	1.00	1
Drug Reaction	1.00	1.00	1.00	1
Fungal infection	1.00	1.00	1.00	1
GERD	1.00	1.00	1.00	1
Gastroenteritis	1.00	1.00	1.00	1
Heart attack	1.00	1.00	1.00	1
Hepatitis B	1.00	1.00	1.00	1
Hepatitis C	1.00	1.00	1.00	1
Hepatitis D	1.00	1.00	1.00	1
Hepatitis E	1.00	1.00	1.00	1
Hypertension	1.00	1.00	1.00	1
Hyperthyroidism	1.00	1.00	1.00	1
Hypoglycemia	1.00	1.00	1.00	1
Hypothyroidism	1.00	1.00	1.00	1
Impetigo	1.00	1.00	1.00	1
Jaundice	1.00	1.00	1.00	1
Malaria	1.00	1.00	1.00	1
Migraine	1.00	1.00	1.00	1
Osteoarthritis	1.00	1.00	1.00	1
Paralysis (brain hemorrhage)	1.00	1.00	1.00	1
Peptic ulcer disease	1.00	1.00	1.00	1
Pneumonia	1.00	1.00	1.00	1
Psoriasis	1.00	1.00	1.00	1
Tuberculosis	1.00	1.00	1.00	1
Typhoid	1.00	1.00	1.00	1
Urinary tract infection	1.00	1.00	1.00	1
Varicose veins	1.00	1.00	1.00	1
hepatitis A	1.00	1.00	1.00	1
accuracy			1.00	41
macro avg	1.00	1.00	1.00	41
weighted avg	1.00	1.00	1.00	41

3.2.2 DECISION TREE

A decision tree is a type of algorithm used in machine learning for decision-making tasks. It is a tree-like structure that represents a set of decisions and their possible consequences. Each node in the tree represents a decision based on a specific feature or attribute, and each branch represents the possible outcome of that decision. The goal of a decision tree is to create a model that predicts the value of a target variable based on several input variables. The tree is constructed by recursively splitting the dataset into smaller subsets based on the most important features, with the aim of minimising the entropy or impurity of the resulting subsets. Decision trees are widely used in various fields, including finance, medicine, and marketing, because they are easy to interpret and can handle both categorical and numerical data.

```
Best parameters: {'max_depth': 7, 'min_samples_leaf': 2, 'min_samples_split': 10}
Best score: 0.21463414634146344
Test set accuracy: 0.21951219512195122
```

3.2.3 K-MEANS

K-means is a clustering algorithm that aims to partition a given dataset into K clusters, where K is a pre-defined number. The algorithm works by first randomly selecting K points as initial cluster centroids, then assigning each data point to the nearest centroid based on Euclidean distance. After the assignment of all data points, the cluster centroids are recomputed by taking the mean of all the points assigned to each cluster. The previous two steps are repeated iteratively until the cluster assignments and centroids no longer change or the maximum number of iterations is reached. The resulting K clusters can be used to explore patterns in the data or as inputs to other algorithms. K-means is widely used in various fields such as image segmentation, document clustering, and customer segmentation. However, it is sensitive to the initial random selection of centroids and can get stuck in local optima. Therefore, multiple runs with different initializations and selection of the optimal K value are recommended for better results.

```
Best parameters: {'max_iter': 50, 'n_clusters': 5}
Best score: -4588.454588970504
Silhouette score: 0.09686414253872007
```

3.2.3 MULTINOMIAL NAIVE BAYES

The Multinomial Naive Bayes algorithm is a probabilistic algorithm used for classification tasks, particularly in natural language processing. It is based on Bayes' theorem, which states that the probability of a hypothesis (in this case, a class label) given the evidence (in this case, a set of features or words) is proportional to the probability of the evidence given the hypothesis multiplied by the prior probability of the hypothesis. The Multinomial Naive Bayes algorithm assumes that the features are categorical and counts the occurrences of each feature in each class to compute the probabilities. It also assumes that the features are conditionally independent given the class label. Despite its simplicity, the Multinomial Naive Bayes algorithm has been shown to perform well in many text classification tasks, especially when the number of features is large compared to the number of instances.

Accuracy: 1.0

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.00	1.00	1.00	1
AIDS	1.00	1.00	1.00	1
Acne	1.00	1.00	1.00	1
Alcoholic hepatitis	1.00	1.00	1.00	1
Allergy	1.00	1.00	1.00	1
Arthritis	1.00	1.00	1.00	1
Bronchial Asthma	1.00	1.00	1.00	1
Cervical spondylosis	1.00	1.00	1.00	1
Chicken pox	1.00	1.00	1.00	1
Chronic cholestasis	1.00	1.00	1.00	1
Common Cold	1.00	1.00	1.00	1
Dengue	1.00	1.00	1.00	1
Diabetes	1.00	1.00	1.00	1
Dimorphic hemmorhoids(piles)	1.00	1.00	1.00	1
Drug Reaction	1.00	1.00	1.00	1
Fungal infection	1.00	1.00	1.00	1
GERD	1.00	1.00	1.00	1
Gastroenteritis	1.00	1.00	1.00	1
Heart attack	1.00	1.00	1.00	1
Hepatitis B	1.00	1.00	1.00	1
Hepatitis C	1.00	1.00	1.00	1
Hepatitis D	1.00	1.00	1.00	1
Hepatitis E	1.00	1.00	1.00	1
Hypertension	1.00	1.00	1.00	1
Hyperthyroidism	1.00	1.00	1.00	1
Hypoglycemia	1.00	1.00	1.00	1
Hypothyroidism	1.00	1.00	1.00	1
Impetigo	1.00	1.00	1.00	1
Jaundice	1.00	1.00	1.00	1
Malaria	1.00	1.00	1.00	1
Migraine	1.00	1.00	1.00	1
Osteoarthritis	1.00	1.00	1.00	1
Paralysis (brain hemorrhage)	1.00	1.00	1.00	1
Peptic ulcer disease	1.00	1.00	1.00	1
Pneumonia	1.00	1.00	1.00	1
Psoriasis	1.00	1.00	1.00	1
Tuberculosis	1.00	1.00	1.00	1
Typhoid	1.00	1.00	1.00	1
Urinary tract infection	1.00	1.00	1.00	1
Varicose veins	1.00	1.00	1.00	1
hepatitis A	1.00	1.00	1.00	1
accuracy			1.00	41
macro avg	1.00	1.00	1.00	41
weighted avg	1.00	1.00	1.00	41

3.2.3 RANDOM FOREST

Random Forest is an ensemble learning algorithm that is used for classification and regression tasks. It is based on the idea of decision trees, where multiple decision trees are constructed during training and the output is the mode of the classes (classification) or mean prediction (regression) of individual trees. The trees are constructed by randomly selecting a subset of features at each node and selecting the best split among those features. Random Forest helps to overcome the problem of overfitting, as the algorithm creates multiple trees and combines them to make a final decision, resulting in a more robust and accurate model. Random Forest is widely used in a variety of applications such as bioinformatics, finance, and image classification.

Fitting 5 folds for each of 90 candidates, totalling 450 fits
 Accuracy: 1.0

	precision	recall	f1-score	support
(vertigo) Paroymsal Positional Vertigo	1.00	1.00	1.00	1
AIDS	1.00	1.00	1.00	1
Acne	1.00	1.00	1.00	1
Alcoholic hepatitis	1.00	1.00	1.00	1
Allergy	1.00	1.00	1.00	1
Arthritis	1.00	1.00	1.00	1
Bronchial Asthma	1.00	1.00	1.00	1
Cervical spondylosis	1.00	1.00	1.00	1
Chicken pox	1.00	1.00	1.00	1
Chronic cholestasis	1.00	1.00	1.00	1
Common Cold	1.00	1.00	1.00	1
Dengue	1.00	1.00	1.00	1
Diabetes	1.00	1.00	1.00	1
Dimorphic hemmorhoids(piles)	1.00	1.00	1.00	1
Drug Reaction	1.00	1.00	1.00	1
Fungal infection	1.00	1.00	1.00	1
GERD	1.00	1.00	1.00	1
Gastroenteritis	1.00	1.00	1.00	1
Heart attack	1.00	1.00	1.00	1
Hepatitis B	1.00	1.00	1.00	1
Hepatitis C	1.00	1.00	1.00	1
Hepatitis D	1.00	1.00	1.00	1
Hepatitis E	1.00	1.00	1.00	1
Hypertension	1.00	1.00	1.00	1
Hyperthyroidism	1.00	1.00	1.00	1
Hypoglycemia	1.00	1.00	1.00	1
Hypothyroidism	1.00	1.00	1.00	1
Impetigo	1.00	1.00	1.00	1
Jaundice	1.00	1.00	1.00	1
Malaria	1.00	1.00	1.00	1
Migraine	1.00	1.00	1.00	1
Osteoarthritis	1.00	1.00	1.00	1
Paralysis (brain hemorrhage)	1.00	1.00	1.00	1
Peptic ulcer disease	1.00	1.00	1.00	1
Pneumonia	1.00	1.00	1.00	1
Psoriasis	1.00	1.00	1.00	1
Tuberculosis	1.00	1.00	1.00	1
Typhoid	1.00	1.00	1.00	1
Urinary tract infection	1.00	1.00	1.00	1
Varicose veins	1.00	1.00	1.00	1
hepatitis A	1.00	1.00	1.00	1
accuracy			1.00	41
macro avg	1.00	1.00	1.00	41
weighted avg	1.00	1.00	1.00	41

3.2.3 ANN

We implement a neural network to predict the top five probable diseases based on a given set of symptoms. The neural network has two layers, both of which are dense layers. The first dense layer has 132 neurons and uses the ReLU activation function. The input shape of this layer is (132,), which corresponds to the number of symptoms being considered. The second dense layer has 41 neurons, which corresponds to the number of unique diseases in the dataset. This layer uses the sigmoid activation function.

The model is compiled using the 'adam' optimizer and the 'sparse_categorical_crossentropy' loss function. The model is trained using the 'fit' method on the training data. The evaluation of the model is performed using the 'evaluate' method on the test data.

The 'predict' method is then used to predict the top five probable diseases for each set of symptoms in the test data. The predicted diseases are extracted by finding the top five indices with the highest predicted probabilities using the 'argsort' method. These indices are then used to extract the corresponding disease names from the 'diseases' array.

Finally, a summary table is created that shows the symptoms and the top five predicted diseases for each set of symptoms in the test data.

```
( 1 ) Symptoms: itching, skin_rash, nodal_skin_eruptions, dischromic_patches,
        Predictions: Fungal infection, Drug Reaction, Acne, Urinary tract infection, Chicken pox,
( 2 ) Symptoms: continuous_sneezing, shivering, chills, watering_from_eyes,
        Predictions: Allergy, Urinary tract infection, Fungal infection, Osteoarthritis, Malaria,
( 3 ) Symptoms: stomach_pain, acidity, ulcers_on_tongue, vomiting, cough, chest_pain,
        Predictions: GERD, Heart attack, Chronic cholestasis, Drug Reaction, Migraine,
( 4 ) Symptoms: itching, vomiting, yellowish_skin, nausea, loss_of_appetite, abdominal_pain, yellowing_of_eyes,
        Predictions: Chronic cholestasis, Hepatitis C, hepatitis A, Hepatitis D, Jaundice,
( 5 ) Symptoms: itching, skin_rash, stomach_pain, burning_micturition, spotting_urination,
        Predictions: Drug Reaction, Fungal infection, Chicken pox, Urinary tract infection, Cervical spondylosis,
( 6 ) Symptoms: vomiting, indigestion, loss_of_appetite, abdominal_pain, passage_of_gases, internal_itching,
        Predictions: Peptic ulcer disease, Chronic cholestasis, hepatitis A, Hepatitis C, Gastroenteritis,
```

3.3 PROGNOSIS

3.3.1 SEVERITY DESCRIPTION

This code block defines functions to get severity, description, and precaution details for symptoms, reads data from CSV files, and stores them in Python dictionaries. It also defines a function to calculate the severity of a condition based on the symptoms and number of days a person has been experiencing them.

The *calc_condition(exp,days)* function takes a list of symptoms and the number of days a person has been experiencing them as input parameters. It calculates the severity of the condition based on the sum of severity levels of the symptoms and the number of days. If the severity value is greater than 13, it returns 1, which indicates that the person should consult a doctor. Otherwise, it returns 0, which indicates that the person should take precautions.

```
1 severityDictionary
```

```
{'itching': 1,  
 'skin_rash': 3,  
 'nodal_skin_eruptions': 4,  
 'continuous_sneezing': 4,  
 'shivering': 5,  
 'chills': 3,  
 'joint_pain': 3,  
 'stomach_pain': 5,  
 'acidity': 3,  
 'ulcers_on_tongue': 4,  
 'muscle_wasting': 3,  
 'vomiting': 5,  
 'burning_micturition': 6,  
 'spotting_urination': 6,  
 'fatigue': 4,  
 'weight_gain': 3,  
 'anxiety': 4,  
 'cold_hands_and_feets': 5,  
 'mood_swings': 3,  
 'weight_loss': 3,  
 'restlessness': 5,  
 'lethargy': 2,
```


3.3.2 CHAT

We use natural language processing (NLP) and machine learning (ML) techniques to predict possible diseases based on the symptoms provided by the user.

The program first takes the user's name as input and then asks for the two primary symptoms. It then preprocesses the symptoms, checks for similarity using both syntactic and semantic methods, and suggests possible symptoms based on all data and input symptom synonyms. If no similar syntactic, semantic and suggested symptoms are found, the program returns None and asks for clarification.

If at least one symptom is found, the program duplicates it and proceeds with the prediction of possible diseases. It then asks the user whether they are experiencing any additional symptoms and suggests possible diseases based on the input symptoms.

Finally, the program outputs a possible disease and its description, along with some precautions and a recommendation for medical consultation if needed. The program then asks if the user needs another medical consultation and ends if the answer is no. The ML model used in this program is a KNN classifier.

3.4 RESULTS AND INFERENCES

The KNN algorithm achieved perfect accuracy on the evaluated dataset, which is a good sign.

The Decision Tree algorithm has been optimised using a grid search to find the best hyperparameters. The best score achieved by the algorithm is 0.2146, which is relatively low. Moreover, the test set accuracy is only 0.2195, which indicates that the model may not generalise well to new data.

The KMeans algorithm has been optimised using a grid search to find the best hyperparameters. The best silhouette score achieved by the algorithm is 0.0968, which indicates that the clustering performance is relatively low. However, it is

important to note that the performance of clustering algorithms heavily depends on the dataset's characteristics and the chosen number of clusters.

The Naive Bayes algorithm and Random forest algorithm have achieved perfect accuracy on the evaluated dataset, which is a good sign.

ANN, used for predicting the top 5 probable diseases has also achieved a perfect accuracy on the evaluated dataset which is a good sign.

CODE ANALYSIS

- **Number of Lines: 786**
- **Number of Functions/Routines: 34**

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 CONCLUSION

In conclusion, the Disease Prediction System using Symptoms and Machine Learning algorithms provides a user-friendly, accurate, and reliable solution for early detection of diseases based on symptoms entered by the user. The project's multiple machine learning algorithms enable personalised disease prediction based on the user's history and preferences. The system is cost-effective and accessible, making it a valuable tool for healthcare professionals and individuals seeking medical information. The project's success demonstrates the potential of machine learning and artificial intelligence in the healthcare industry and provides a framework for future developments in this field.

4.2 FUTURE WORK

- Incorporating more advanced machine learning algorithms to improve accuracy and performance.
- Adding more data to the training set to increase the number of diseases and symptoms covered by the system.
- Implementing natural language processing techniques to allow users to enter symptoms in a more conversational manner, improving the user experience.
- Integrating the system with electronic health records to access more comprehensive patient information and improve disease prediction accuracy.
- Implementing real-time disease tracking to provide users with up-to-date information on disease outbreaks and epidemics in their area.
- Developing a mobile application to make the system more accessible and convenient for users on-the-go.
- Incorporating a chatbot feature to provide users with personalised medical advice and recommendations based on their symptoms and medical history.

- Adding a feature for doctors to input patient data and receive disease predictions and treatment recommendations based on their expertise and the system's algorithms.
- Incorporating genetic data to enable personalised disease prediction and prevention based on individual genetic profiles.

APPENDIX

CHAPTER 5

REFERENCES

1. <https://www.ijraset.com/best-journal/disease-prediction-using-machine-learning-algorithms-knn-and-cnn>
2. <https://www.hindawi.com/journals/cin/2022/3287068/>
3. <https://www.ijsr.net/archive/v5i1/NOV153131.pdf>
4. https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/24065/final/fin_irjmets1653367944.pdf
5. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3673232

BIODATA



Name : Rudhreesh Kumar R B R
Mobile Number : 9445475349
E-mail : rudhreeshkumaar.rbr2020@vitstudent.ac.in
Permanent Address : No.12, Palani Street, Rajaji Nagar,
Villivakkam, Chennai-600049



Name : B. Manasa Reddy
Mobile Number : 9014763464
E-mail : beerellymanasareddy2001@gmail.com
Permanent Address : plot no 145, road no-4, hasthinapur, hyderabad



Name : Harshitha K
Mobile Number : 9704104090
E-mail : harshitha10shetty@gmail.com
Permanent Address : 10-193, mundy bazaar, guntakal-515801.



Name : Namitha M V
Mobile Number : 8074254575
E-mail : makamnamitha96@gmail.com
Permanent Address : NarasappaCompound,BalajiCircle,Hindupur-515201

APPENDIX

DATASET LINK:

[EDA Dataset](#)

OVERALL DEMO – VIDEO LINK:

[https://drive.google.com/file/d/1FqSCntHLeJduFU0RzYx-MmlaxqqNEg3p/view?usp=share link](https://drive.google.com/file/d/1FqSCntHLeJduFU0RzYx-MmlaxqqNEg3p/view?usp=share_link)

SOURCE CODE LINK: <https://colab.research.google.com/drive/1EmXCY-7uVC7ri0klCvpXD9j-GdQEIgp8?usp=sharing>