



IJIRCCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 1, January 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.379



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

Disease Prediction Using Symptoms

NISHI MISHRA ⁽¹⁾, KONETI SUCHITHA ⁽²⁾, SOWMIYA N ⁽³⁾

School of Computer Science and Engineering, Presidency University, Bangalore, Karnataka, India

ABSTRACT: Disease prediction of a human means predicting the probability of a patient's disease after examining the combinations of the patient's symptoms. Monitoring a patient's condition and health information at the initial examination can help doctors to treat a patient's condition effectively. This analysis in the medical industry would lead to streamlined and expedited treatment of patients.

The previous researchers have primarily emphasized machine learning models mainly Support Vector Machine (SVM), K-nearest neighbors (KNN), and RUSBoost for the detection of diseases with the symptoms as parameters. However, the data used by the prior researchers for training the model is not transformed and the model is completely dependent on the symptoms, while their accuracy is poor. Nevertheless, there is a need to design a modified model for better accuracy and early prediction of human disease. The proposed model has improved the efficacy and accuracy model, by resolving the issue of the earlier researcher's models.

The proposed model is using the medical dataset from Kaggle and transforms the data by assigning the weights based on their rarity. This dataset is then trained using a combination of machine learning algorithms: Random Forest, Long Short-Term Memory (LSTM), and SVM. Parallel to this, the history of the patient can be analyzed using the LSTM Algorithm. SVM is then used to conclude the possible disease.

KEYWORDS: Symptoms, Disease, Symptoms_Severity, Symptoms_Description, Symptoms_Precautions, Accuracy, F1 score, Random forest, Decision Tree classifier, Naive Bayes,

I. INTRODUCTION

Human disease prediction is a crucial part of human life. Early disease prediction of a human is an important step in the treatment of disease. Since the very beginning, a doctor has handled it almost exclusively. Thus, the healthcare industry thrives on innovation to make logistics efficient. Innovation is the heart of the medical industry. It is what drives new treatments, cures and therapies. Innovation is also what keeps the medical industry current and relevant. The scope of development in the medical industry is vast. There are many areas where innovation is needed to make progress. Some of these include developing new treatments for diseases, finding ways to improve patient care, and making medical procedures more efficient.

One of the emerging technologies that has benefited individuals in many different industries is machine learning. It includes numerous algorithms for solving regression and classification issues. When working with predictions, machine learning is the first technology that springs to mind. Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and KNN are machine learning algorithms that classify data into different groups. The model can be trained using a preprocessed dataset with the aid of machine learning. Artificial intellect (AI) is a field that aims to imbue robots with human intellect in order to perform activities that humans perform. One subset of AI is machine learning.

To predict the disease Naive Bayes, Decision trees, and Random forest algorithms were used. After training the three models using these three algorithms, the most occurred disease will be considered as the final result. Feature selection is done based on the symptoms given by the user. After pre-processing, the dataset was split into training data and testing data. Soon after training the models, models will be checked with testing data to find the accuracies of all three models, and confusion matrices were drawn to analyze the results based on training and testing data.

The chatbot in our project is used for information acquisition. It acquires the patient's information along with the symptoms and the disease is predicted on the basis of the symptoms. The disease prediction chatbot is designed using the concepts of NLP and machine learning algorithms.

II. LITERATURE REVIEW

In this overview work, different machine learning techniques for identifying conditions including diabetes and heart disease are compared and contrasted. It centers on a set of machine learning techniques and algorithms that can be applied to the detection and interpretation of medical conditions. Based on this test data, a training set of examples with acceptable targets was generated, and algorithms responded adequately to all possible inputs. SVM provides good outcomes across several application fields. Positive attributes were found with the use of the FS approach. In response to these characteristics, SVM attains an accuracy rate of 84.5 percent. For every data set, there are various research and learning methodologies, as well as a few different data kinds. Every data set has a different learning and research methodology, as do some types of data.

Data mining for symptom-based health prediction was started by S. Vijava Shetty and colleagues. This project aims to build a machine learning model that can predict common diseases based on real symptoms by using significant symptoms and diseases. For instance, a straightforward equation cannot be used to correctly identify body organs. Thus, instance-based training is primarily required for pattern recognition. In the realm of biomedicine, machine learning and pattern recognition have the potential to increase the precision of disease identification and treatment. They also respect the principle of impartiality in decision-making. The proposed approach integrates text processing with different Machine Learning techniques to achieve accurate prediction.

Using the machine learning approach proposed by Dhiraj Dahiwade, create a disease prediction model. The matching of accuracy was the key idea. Data processing uncovers patterns and information hidden in an astronomically large amount of medical data with the use of illness data. This published paper employs K-Nearest Neighbor (KNN) and Convolutional Neural Network (CNN) methods. In this investigation, the sufferer's habits are essential to making a precise forecast. Following the prediction of general disease, this approach is prepared to provide the overall disease likelihood, which can be either higher or lower. Both structured and unstructured datasets are supported. The accuracy is just 84.5%, which is the constraint.

P. Hamsagayathri et al. presented Symptoms Based Disease Prediction Using Machine Learning Techniques. In this overview work, different machine learning techniques for identifying conditions including diabetes and heart disease are compared and contrasted.

It centers on a set of machine learning techniques and algorithms that can be applied to the detection and interpretation of medical conditions. Based on this test data, a training set of examples with acceptable targets was generated, and algorithms responded adequately to all possible inputs. SVM provides good outcomes across several application fields. Positive attributes were found with the use of the FS approach. In response to these characteristics, SVM attains an accuracy rate of 84.5 percent. For every data set, there are various research and learning methodologies, as well as a few different data kinds. Every data set has a different learning and research methodology, as do some types of data.

Despite being widely employed for distinction, the WAKE tool's efficiency pales in comparison to Naive Bayes. The survey illustrates the benefits and drawbacks of these kinds of algorithms. A collection of AI community-developed tools is also included in this survey paper.

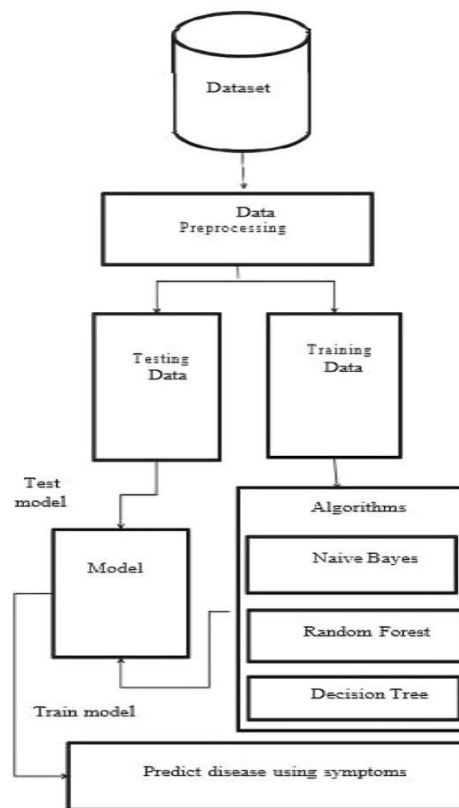
Min Chen et al. proposed Disease Prediction by Machine Learning Over Big Data from Healthcare Communities. Improved prediction models and actual hospital data from 2013 to 2015 in central China serve as the foundation for the testing. The hospital provides a variety of data, including gene, EHR, and medical picture data. They tackle the problem of incomplete data by reconstructing missing data using a latent component model. The goal of this work was to simplify machine learning methods for predicting chronic illness outbreaks in communities that are prone to sickness. The NB, KNN, and DT algorithms are used to forecast the risk of cerebral infarction disease. A training data set is provided for KNN classification, and the nearest k instances are located within the training data set.

III. METHODOLOGY

The dataset that was obtained from Kaggle in the first module underwent data preprocessing. Data pre-processing increases the model's accuracy. Every symptom in the symptom severity dataset will have a priority. This priority will

be applied to every symptom in the collection of illness symptoms. This helps to accurately forecast sickness. In the second module, a model was created using the Decision Tree Classification technique. At every node in this decision tree, decisions are made using the Gini index. The Gini index was used to make a choice for the training dataset that had been pre-processed. A confusion matrix was made to analyze the outcomes of this decision tree classifier model, and testing data was used to assess the correctness of the decision tree model once it had been trained.

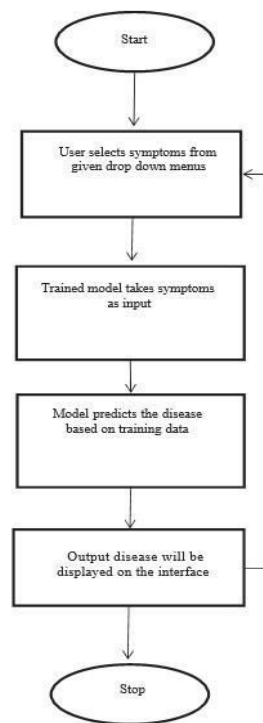
In the module, a Random Forest classification technique was used to build a model. With the exception of building numerous decision trees using a random forest classifier, this Random Forest approach works similarly to a decision tree. All of the outputs from each tree will be taken into account to determine the final output. The model was trained using the training data and tested using the testing data. To investigate the results obtained with this trained random forest model and the precision of this trained model, a confusion matrix was constructed. In the fourth module, a model was created using Naive Bayes classifier techniques.



A proposed system for disease prediction using symptoms based on machine learning could include the following components:

- a. User Interface: A user interface that allows the user to input their lifestyle factors, such as age, gender, BMI, physical activity level, diet (i.e how many times of junk food intake in a week), Sleep hours, Smoking and Drinking habits.
- b. Data Preprocessing: A component that preprocesses the user input data to remove any inconsistencies or errors, and prepares it for use in the machine learning algorithm.
- c. Machine Learning Model: A machine learning algorithm that uses the preprocessed user input data to predict the likelihood of the user developing a particular lifestyle-based disease, such as diabetes, hypertension, Depression or Healthy. The algorithm may use different techniques, such as decision trees, support vector machines, or Random forest for the accurate prediction of the disease for the given input by the user.

- d. Disease Prediction: A component that takes the output of the machine learning model and generates a prediction about the likelihood of the user developing a particular disease or the person is healthy. Some predictions are also displayed as output on the interface according to the predicted disease, this helps the user to overcome or reduce the intensity of the disease.
- e. Data Retrieval: A component that uses the user's input data, predictions, and any other relevant information in a database, and retrieves it as needed for use in the machine learning algorithm or for display to the user.
- f. Prediction Analysis: The User is also provided a way to Consult a Doctor, book an appointment. Our model is also provided with a feature to Analyze about the disease i.e predicted disease. The user can see the dataset we have used for the machine learning model to predict the disease.
- g. Reporting: A reporting component that generates reports on the user's lifestyle data and disease predictions, which can be used by the user or healthcare providers to track the user's health over time and identify any trends or potential issues.



Module 1: Data Preparation

The dataset that was previously covered in this module is preprocessed. There are two datasets used in this project. The diseases' symptoms are described in the first, and the severity of the disease is described in the second. First things first, we need to make sure that the data in this data preparation module is supplied in a standard format. In the event that this is not the case, a common format for the data should be created. This should be done first before continuing because creating models with null values and missing data has a significant impact on the model's performance. After the null values have been removed, each symptom must be assigned a severity level. If a symptom does not have a severity value in the symptom-severity table, it should be given a value of zero.

Step 1: Gather the data such that all the records that are needed to train and test the model in a single table.

Step 2: Find all the NULL values, missing values and make the values zero.

`dafr.isna().sum(); dafr.isnull().sum()`

Step- 3: Assign every symptom with their respective weight as per symptom- severity.csv.

Step- 4: If no weight is present in the above file, assign it to zero.



Module 2: Building the Model using Decision Tree Classifier

The Dataset must be divided into training and testing data after preprocessing. A decision tree classifier is used to build a model, and the Gini index is utilized to make decisions at each node of the tree. These are the choices that lead to the classification of symptoms into different groups. The Gini index is calculated using probabilities of each class.

Step 1: Split the data into two parts i.e., for training 80% of the data and for testing 20%.

```
x_t, x_te, y_t, y_te = train_test_split(dataset_used)
```

Step 2: By using random forest classifier function, develop a random forest classifier model.

```
rfconmodel = RandomForestClassifier(random_state=11)
```

Step 3: The model determines the optimal solution from a series of decision trees created from a few randomly chosen subsets of the training set.

```
rfconmodel.fit(x_t, y_t)
```

Step 4: Predict values using the random forest classifier model that we have generated so far.

```
predictrf = rfconmodel.predict(x_te)
```

Module 3: Building the Model using Random Forest

A random forest approach was used to generate a model in the third module. Some decision trees will be built, and the most often occurring output from all of the decision trees will be used as the random forest classifier's final output. After the model has been trained, predictions will be made using testing data.

Step 1: Using the decision tree classifier function, generate a decision tree classifier model.

```
detr = DecisionTreeClassifier(criterion = "gini")
```

Step 2: Use Gini Index as the Attribute Selection Measure to split the records and fit the training data.

```
Detr_fit=detr.fit(x_t, y_t)
```

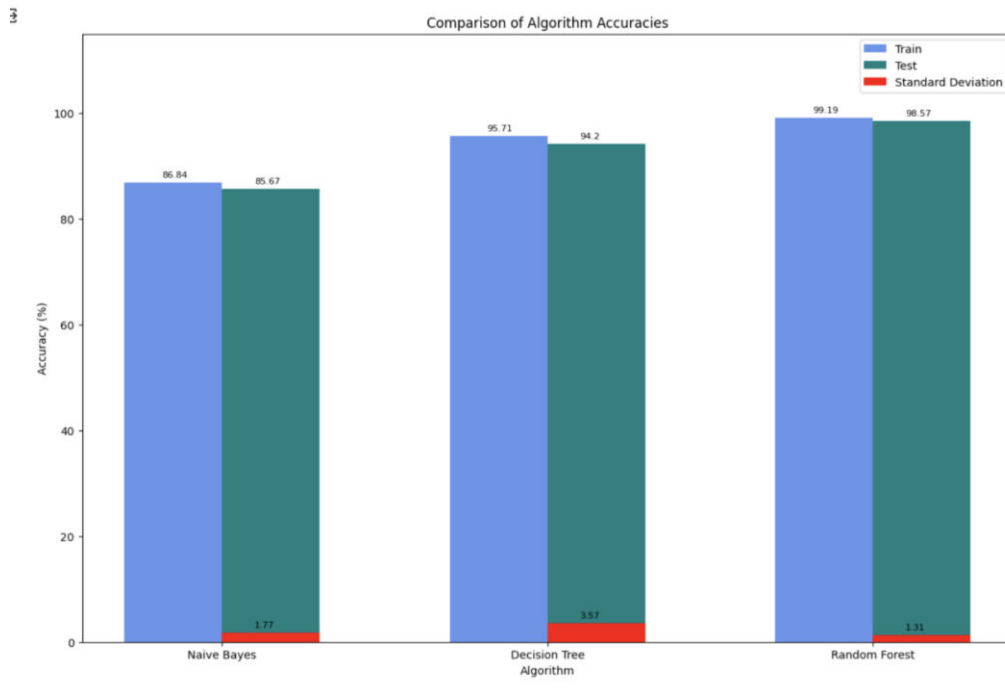
Step 3: Predict values using the above created model.

```
Predict_detr = Detr_fit.predict(x_te)
```

Step 4: Plot the confusion matrix and find the accuracy.

```
print(f: {accuracy_score(y_te, Predict_detr)*100}")
```

SL. NO	ALGORITHMS USED	ACCURACY
1	ONLY NAIVE BAYES	85.67%
2	ONLY DECISION TREE	94.2%
3	ONLY RANDOM FOREST	98.65%



IV. SYSTEM DESIGN & IMPLEMENTATION

Each data set is pre-processed in the first stage. The pre-processed datasets are fed into the various machine learning algorithms in the second stage. The models' output is subsequently examined using a variety of measures in the third phase. In a subsequent stage, the model with the best accuracy is used to identify diabetes in any person and is integrated with an online application. This web application is created in Python using the Flask programming language.

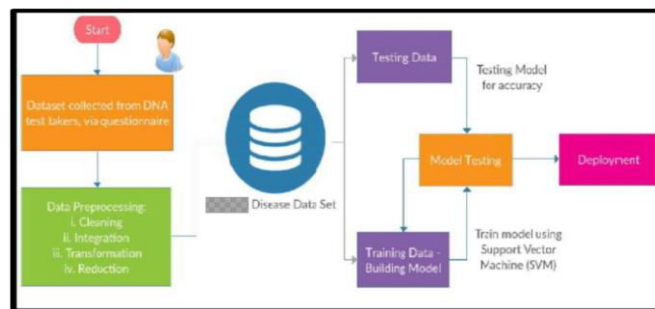


Figure 5: System design architecture

The following is a summary of the research's contributions:

Our initial contribution is the training of multiple machine learning algorithms for diabetes detection utilizing four distinct clinical datasets. Various preprocessing procedures are applied to all of the datasets.

Second, the effectiveness of each machine learning method is evaluated using four datasets and a number of characteristics, including accuracy, ROC curve, precision, recall, and f1-score.

Furthermore, by employing several feature selection techniques like correlation and chi-square, among others, we have discovered a number of significant features or traits. The feature selection techniques identify the traits that are most strongly associated with diabetes. On the smaller set of qualities, the ML algorithms' performances were also examined.



SYSTEM DESIGN

Data Collection

Data is collected from individuals who have taken DNA tests, and additional information is gathered through questionnaires. This data is likely to include genetic information as well as personal health information that individuals report.

Data Preprocessing

The collected data undergoes several preprocessing steps to prepare it for analysis:

Cleaning: Removing inaccuracies and correcting the data.

Integration: Combining data from different sources to create a cohesive dataset.

Transformation: Modifying the data into a suitable format or structure for analysis.

Reduction: Decreasing the data size by eliminating redundancy and focusing on relevant features.

Disease Dataset

The processed data forms a comprehensive dataset of disease-related information. This dataset is used to train and test the machine learning models.

Training Data - Building Model

A portion of the dataset, known as the training data, is used to build the machine learning model. In this case, the model is trained using a Support Vector Machine (SVM), which is a supervised learning algorithm known for its effectiveness in classification tasks.

Model Testing

The trained model is then tested for its predictive accuracy using a separate portion of the dataset, typically known as the testing data. This helps in evaluating how well the model has learned from the training data and can generalize to new, unseen data.

Testing Model for Accuracy

This stage involves assessing the performance of the machine learning model using various metrics such as accuracy, precision, recall, and F1-score. The goal is to determine how accurately the model can predict diseases based on the test data.

Deployment

Once the model has been trained and tested satisfactorily, it is deployed into a production environment where it can be used by healthcare professionals or patients. Deployment makes the model accessible to users for practical use, such as predicting diseases based on new patient data.

DATASETS USED FOR IMPLEMENTATION

The dataset that was acquired from Kaggle in the first module. The accuracy of the model is improved by data pre-processing. In the symptom severity dataset, each symptom will be given a priority. All the symptoms in the disease symptoms dataset will be assigned with these priorities[11]. This aids in the precise prediction of illness. A model was developed utilizing a Decision Tree Classification algorithm in the second module. The Gini index is used to make decisions at each and every node in this decision tree. A decision was made for the pre-processed training dataset using the Gini index. After the training of this decision tree model, it was evaluated using testing data to determine its accuracy, and a confusion matrix was created to examine the results of this decision tree classifier model.

Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Symptom_9	Symptom_10	Symptom_11	Symptom_12	Symptom_13
373	Acne	skin rash	blackheads	scurring	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4916	Acne	skin rash	pus filled pimples	blackheads	scurring	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1550	Hyperthyroidism	fatigue	mood swings	weight loss	restlessness	sweating	diarrhoea	fast heart rate	excessive hunger	muscle weakness	irritability	abnormal menstruation	NaN
3081	AIDS	muscle wasting	patches in throat	high fever	extra marital contacts	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3857	Chronic cholestasis	itching	vomiting	yellowish skin	nausea	loss of appetite	abdominal pain	yellowing of eyes	NaN	NaN	NaN	NaN	NaN

Table 2: Disease dataset

	Symptom	weight
51	throat_irritation	4
69	swollen_blood_vessels	5
31	headache	3
42	yellow_urine	4
117	fluid_overload	4

Table 3: Symptoms severity

	Disease	Description
24	Paralysis (brain hemorrhage)	Intracerebral hemorrhage (ICH) is when blood s...
13	Impetigo	Impetigo (im-puh-TIE-go) is a common and highl...
8	Osteoarthritis	Osteoarthritis is the most common form of arth...
25	Typhoid	An acute illness characterized by fever caused...
4	Psoriasis	Psoriasis is a common skin disorder that forms...

Table 4: Description dataset

	Disease	Precaution_1	Precaution_2	Precaution_3	Precaution_4
24	Paralysis (brain hemorrhage)	massage	eat healthy	exercise	consult doctor
13	Impetigo	soak affected area in warm water	use antibiotics	remove scabs with wet compressed cloth	consult doctor
8	Osteoarthritis	acetaminophen	consult nearest hospital	follow up	salt baths
25	Typhoid	eat high calorie vegetables	antibiotic therapy	consult doctor	medication
4	Psoriasis	wash hands with warm soapy water	stop bleeding using pressure	consult doctor	salt baths

Table 5: Precautions dataset

Loading the Dataset:

```
df = pd.read_csv('../content/dataset.csv')
df = shuffle(df, random_state=42)
df.head()
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7
373	Acne	skin_rash	blackheads	scurring	NaN	NaN	NaN	NaN
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring	NaN	NaN	NaN
1550	Hyperthyroidism	fatigue	mood_swings	weight_loss	restlessness	sweating	diarrhoea	fast_heart_rate
3081	AIDS	muscle_wasting	patches_in_throat	high_fever	extra_marital_contacts	NaN	NaN	NaN
3857	Chronic cholestasis	itching	vomiting	yellowish_skin	nausea	loss_of_appetite	abdominal_pain	yellowing_of_eyes

```
[ ] for col in df.columns:
    df[col] = df[col].str.replace('_', ' ')
df.head()
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Symptom_9	Symptom_10
373	Acne	skin rash	blackheads	scurring	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4916	Acne	skin rash	pus filled pimples	blackheads	scurring	NaN	NaN	NaN	NaN	NaN	NaN
1550	Hyperthyroidism	fatigue	mood swings	weight loss	restlessness	sweating	diarrhoea	fast heart rate	excessive hunger	muscle weakness	irritability
3081	AIDS	muscle wasting	patches in throat	high fever	extra marital contacts	NaN	NaN	NaN	NaN	NaN	NaN

Figure 6 : Datasets

Checking the null value:

```
null_checker = df.apply(lambda x: sum(x.isnull()), to_frame(name='count'))
print(null_checker)
```

	count
Disease	0
Symptom_1	0
Symptom_2	0
Symptom_3	0
Symptom_4	348
Symptom_5	1206
Symptom_6	1986
Symptom_7	2652
Symptom_8	2976
Symptom_9	3228
Symptom_10	3408
Symptom_11	3726
Symptom_12	4176
Symptom_13	4416
Symptom_14	4614
Symptom_15	4680
Symptom_16	4728
Symptom_17	4848

Figure 7: Null Value

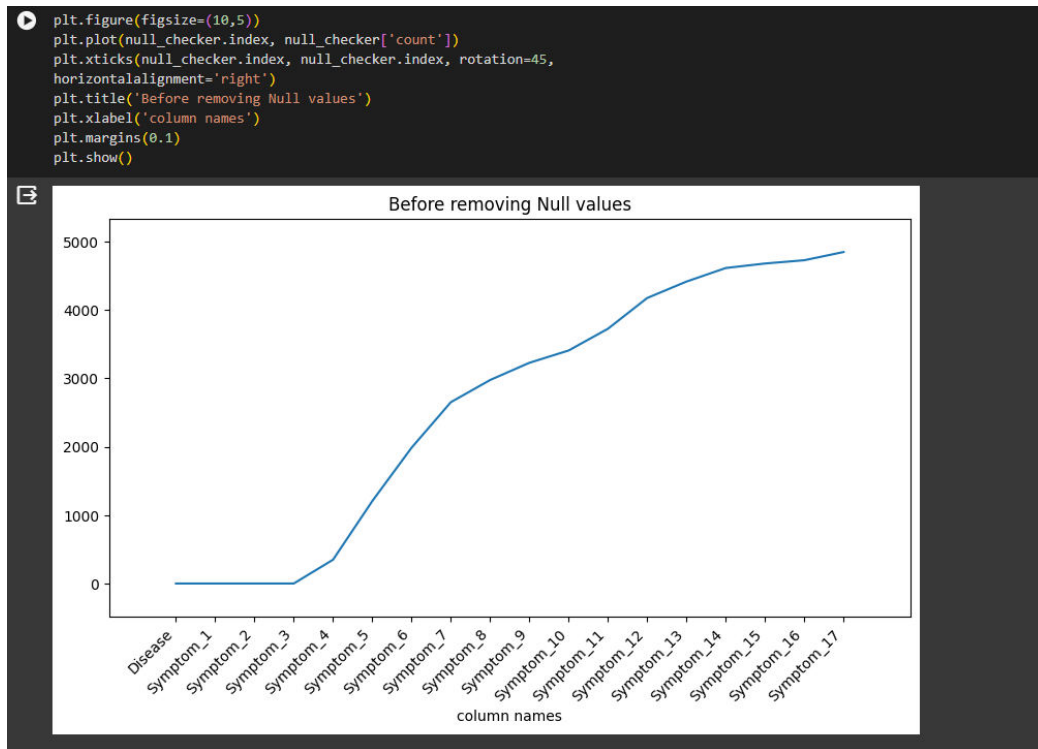


Figure 8: Null Checker Before Removing Null Values

Loading Symptoms Severity Dataset:

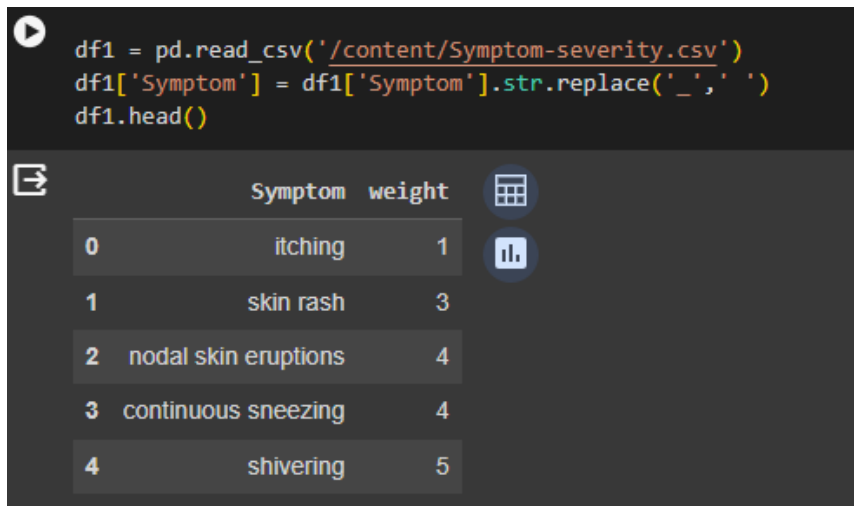


Figure 9: Symptoms Severity Dataset

```
df1['Symptom'].unique()

array(['itching', 'skin rash', 'nodal skin eruptions',
'continuous sneezing', 'shivering', 'chills', 'joint pain',
'stomach pain', 'acidity', 'ulcers on tongue', 'muscle wasting',
'vomiting', 'burning micturition', 'spotting urination', 'fatigue',
'weight gain', 'anxiety', 'cold hands and feets', 'mood swings',
'weight loss', 'restlessness', 'lethargy', 'patches in throat',
'irregular sugar level', 'cough', 'high fever', 'sunken eyes',
'breathlessness', 'sweating', 'dehydration', 'indigestion',
'headache', 'yellowish skin', 'dark urine', 'nausea',
'loss of appetite', 'pain behind the eyes', 'back pain',
'constipation', 'abdominal pain', 'diarrhoea', 'mild fever',
'yellow urine', 'yellowing of eyes', 'acute liver failure',
'fluid overload', 'swelling of stomach', 'swelled lymph nodes',
'malaise', 'blurred and distorted vision', 'phlegm',
'throat irritation', 'redness of eyes', 'sinus pressure',
'runny nose', 'congestion', 'chest pain', 'weakness in limbs',
'fast heart rate', 'pain during bowel movements',
'pain in anal region', 'bloody stool', 'irritation in anus',
'neck pain', 'dizziness', 'cramps', 'bruising', 'obesity',
'swollen legs', 'swollen blood vessels', 'puffy face and eyes',
'enlarged thyroid', 'brittle nails', 'swollen extremities',
'excessive hunger', 'extra marital contacts',
'drying and tingling lips', 'slurred speech', 'knee pain',
'hip joint pain', 'muscle weakness', 'stiff neck',
'swelling joints', 'movement stiffness', 'spinning movements',
'loss of balance', 'unsteadiness', 'weakness of one body side',
'loss of smell', 'bladder discomfort', 'foul smell of urine',
'continuous feel of urine', 'passage of gases', 'internal itching',
'toxic look (typhos)', 'depression', 'irritability', 'muscle pain',
'altered sensorium', 'red spots over body', 'belly pain',
'abnormal menstruation', 'dischromic patches',
'watering from eyes', 'increased appetite', 'polyuria',
'family history', 'mucoid sputum', 'rusty sputum',
'lack of concentration', 'visual disturbances',
'receiving blood transfusion', 'receiving unsterile injections',
'coma', 'stomach bleeding', 'distention of abdomen',
'history of alcohol consumption', 'blood in sputum',
'prominent veins on calf', 'palpitations', 'painful walking',
'pus filled pimples', 'blackheads', 'scurring', 'skin peeling',
'silver like dusting', 'small dents in nails',
'inflammatory nails', 'blister', 'red sore around nose',
'yellow crust ooze', 'prognosis'], dtype=object)
```

Figure 10: List of Symptoms Present

Replacing Null Values:

```
vals = df.values
symptoms = df1['Symptom'].unique()

for i in range(len(symptoms)):
    vals[vals == symptoms[i]] = df1[df1['Symptom'] == symptoms[i]]['weight'].values[0]

d = pd.DataFrame(vals, columns=cols)
d.head()

Disease Symptom_1 Symptom_2 Symptom_3 Symptom_4 Symptom_5 Symptom_6 Symptom_7 Symptom_8 Symptom_9 Symptom_10 Symptom_11 Symptom_12 Symptom_13 Symptom_14 Symptom_15 Symptom_16 Symptom_17
0 Acne 3 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
1 Acne 3 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
2 Hypertthyroidism 4 3 3 5 3 6 5 4 2 2 6 0 0 0 0 0 0
3 AIDS 3 6 7 5 0 0 0 0 0 0 0 0 0 0 0 0 0
4 Chronic cholestasis 1 5 3 5 4 4 4 0 0 0 0 0 0 0 0 0 0

d = d.replace('dischromic patches', 0)
d = d.replace('spotting urination', 0)
df = d.replace('foul smell of urine', 0)
df.head(10)

Disease Symptom_1 Symptom_2 Symptom_3 Symptom_4 Symptom_5 Symptom_6 Symptom_7 Symptom_8 Symptom_9 Symptom_10 Symptom_11 Symptom_12 Symptom_13 Symptom_14 Symptom_15 Symptom_16 Symptom_17
0 Acne 3 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
1 Acne 3 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
2 Hypertthyroidism 4 3 3 5 3 6 5 4 2 2 6 0 0 0 0 0 0
3 AIDS 3 6 7 5 0 0 0 0 0 0 0 0 0 0 0 0 0
4 Chronic cholestasis 1 5 3 5 4 4 4 0 0 0 0 0 0 0 0 0 0
5 Hypertension 3 7 4 4 3 0 0 0 0 0 0 0 0 0 0 0 0
6 Hypoglycemia 5 4 4 3 3 5 5 4 4 4 2 4 0 0 0 0 0
7 Asthma 2 1 5 2 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figure 11: Replacing Null Values With Zeros

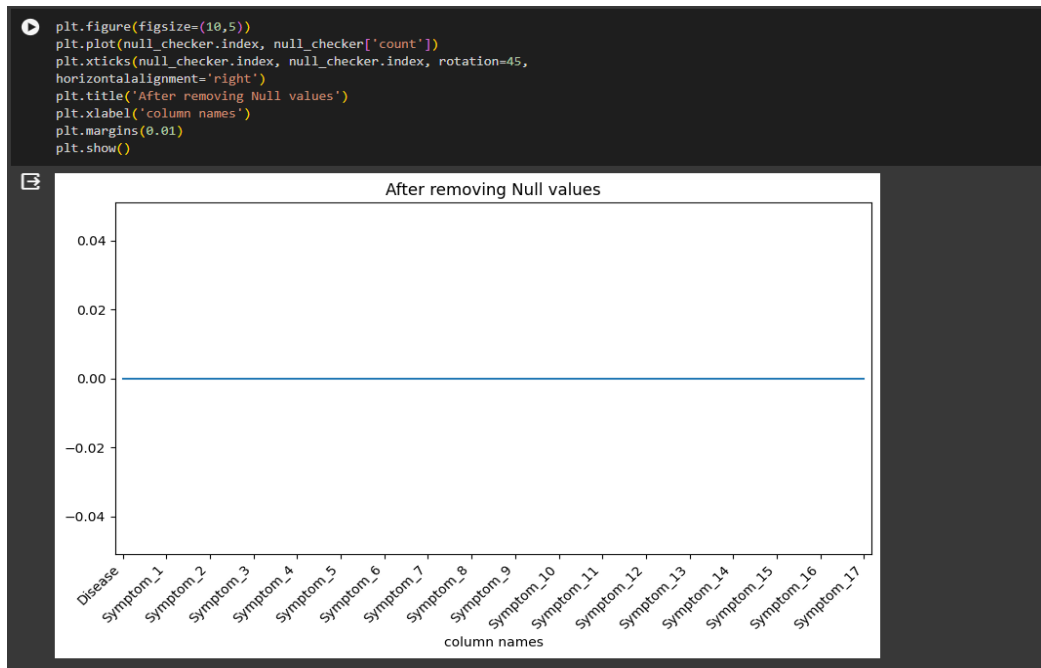


Figure 12: Null Checker After Removing Null Values

Train Test Dataset:

```
[ ] print("Number of symptoms used to identify the disease ",len(df1['Symptom'].unique()))
print("Number of diseases that can be identified ",len(df['Disease'].unique()))

Number of symptoms used to identify the disease 132
Number of diseases that can be identified 41

[ ] data = df.iloc[:,1:].values
labels = df['Disease'].values

[ ] x_train, x_test, y_train, y_test = train_test_split(data, labels, train_size = 0.8,random_state=42)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)

(3936, 17) (984, 17) (3936,) (984,)
```

Figure 13: Train_Test Dataset

Accuracy Score, F1 Score & Precision Score using SVM(Unhyperd):

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for both classification and regression. Though we are saying regression issues as properly it's exceptionally desirable for classification. The goal of the SVM algorithm is to discover a hyperplane in an N-dimensional space that exceptionally classifies the information points.

```
[ ] SVM_unhyperd= SVC()
SVM_unhyperd.fit(x_train, y_train)

[ ] preds = SVM_unhyperd.predict(x_test)
conf_mat = confusion_matrix(y_test, preds)
df_cm = pd.DataFrame(conf_mat, index=df['Disease'].unique(), columns=df['Disease'].unique())
print('F1-score% =', f1_score(y_test, preds, average='macro')*100, '|', 'Accuracy% =', accuracy_score(y_test, preds)*100, '|',
      'Precision% =', precision_score(y_test, preds,average='macro')*100)

F1-score% = 93.10485856410196 | Accuracy% = 93.4959349593496 | Precision% = 94.2446688327875
```

Figure 14: SVM(Unhyperd)

Naive Bayes:

Naive Bayes classifiers are a set of classification algorithms based on Bayes' Theorem. It isn't a single algorithm but a family of algorithms where they all share a common principle, i.e., each pair of capabilities being categorized is unbiased of each other.

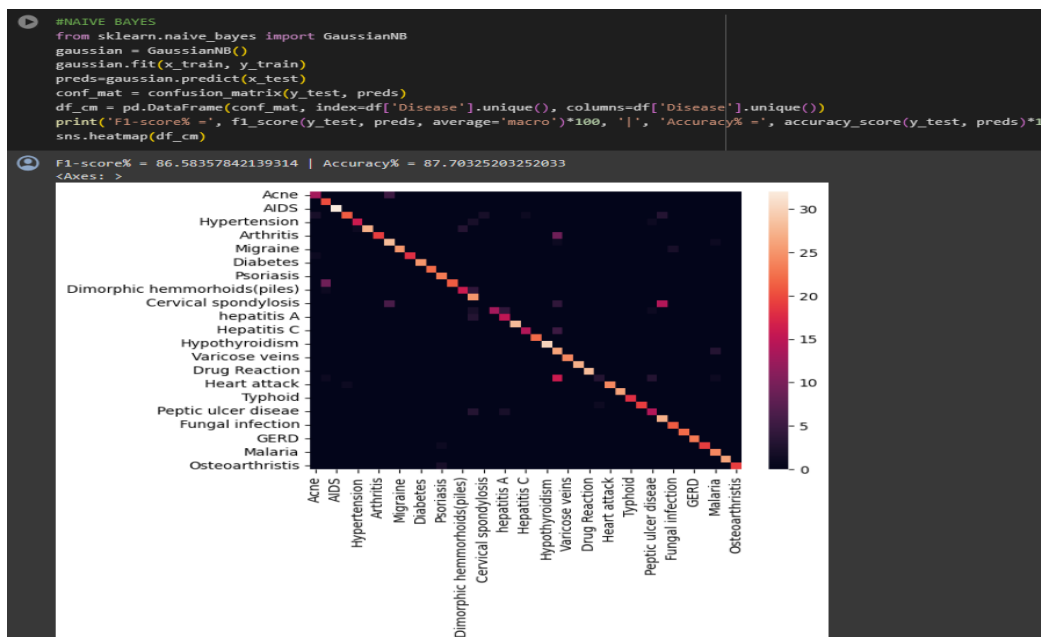


Figure 15: Naive Bayes

Decision Tree:

Decision tree is the most effective and famous tool for prediction and classification. A Decision tree is a flowchart like tree structure, where every internal node denotes a check on an attribute, every branch represents an outcome of the test, and every leaf node (terminal node) holds a class label.

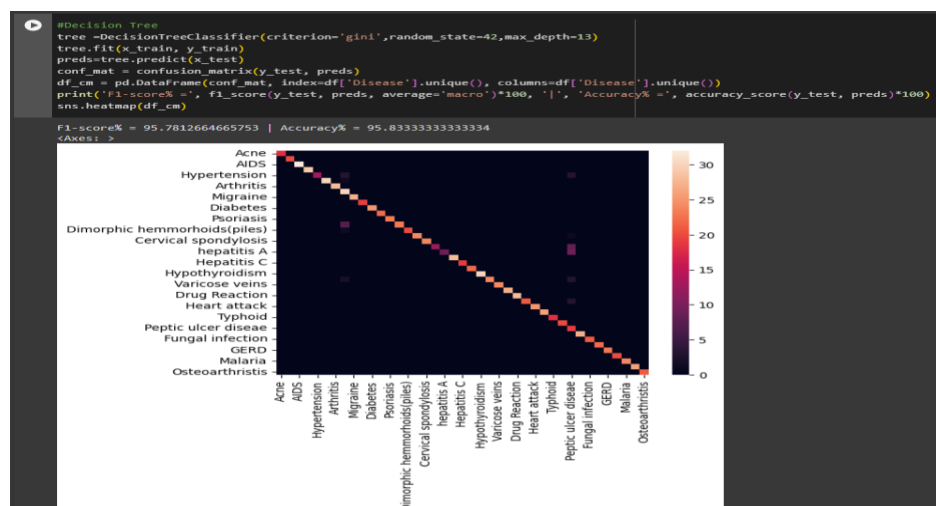


Figure 16: Decision Tree

Random Forest:

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset

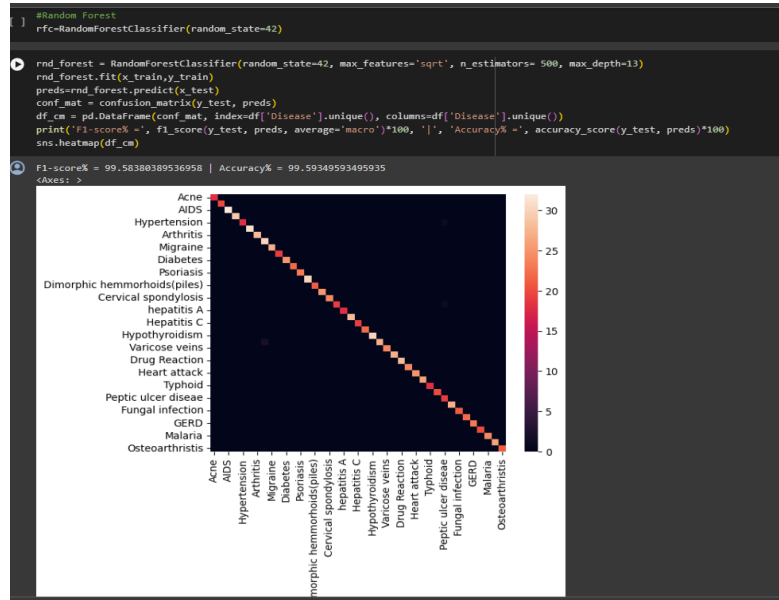


Figure 17: Random Forest

Function to manually test the models:

```

[ ] #Function to manually test the models
def predd(S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17,x):
    symptoms = [S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S12,S13,S14,S15,S16,S17]
    print(symptoms)
    a = np.array(df1["Symptom"])
    b = np.array(df1["weight"])
    for j in range(len(psymptoms)):
        for k in range(len(a)):
            if psymptoms[j]==a[k]:
                psymptoms[j]=b[k]
    psy = [psymptoms]
    pred2 = x.predict(psy)
    print("The prediction is",pred2[0])

[ ] symplist=df1["Symptom"].to_list()
predd(symplist[7],symplist[5],symplist[2],symplist[80],0,0,0,0,0,0,0,0,0,0,0,0,0,0,rnd_forest)

['stomach pain', 'chills', 'nodal skin eruptions', 'muscle weakness', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
The prediction is Paralysis (brain hemorrhage)

[ ] symplist=df1["Symptom"].to_list()
predd(symplist[8],symplist[1],symplist[2],symplist[80],0,0,0,0,0,0,0,0,0,0,0,0,0,0,SVM_hyperd)

['acidity', 'skin rash', 'nodal skin eruptions', 'muscle weakness', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
The prediction is Paralysis (brain hemorrhage)

[ ] symplist=df1["Symptom"].to_list()
predd(symplist[8],symplist[5],symplist[2],symplist[80],0,0,0,0,0,0,0,0,0,0,0,0,0,0,SVM_unhyperd)

['acidity', 'chills', 'nodal skin eruptions', 'muscle weakness', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
The prediction is Allergy

```

Figure18: manually testing of the models

Comparison of Algorithm Accuracies:

```

n_groups = 5
algorithms = ['Naive Bayes', 'Unhyperd SVM', 'Hyperd SVM', 'Decision Tree', 'Random Forest']
train_accuracy = (gaussian_train.mean()*100.0,
                  SVM_unhyperd_train.mean()*100.0,
                  SVM_hyperd_train.mean()*100.0,
                  DS_train.mean()*100.0,
                  rnd_forest_train.mean()*100.0,
                  )

test_accuracy = (gaussian_test.mean()*100.0,
                 SVM_unhyperd_test.mean()*100.0,
                 SVM_hyperd_test.mean()*100.0,
                 DS_test.mean()*100.0,
                 rnd_forest_test.mean()*100.0,
                 )

Standard_Deviation=(gaussian_test.std()*100.0,
                    SVM_unhyperd_test.std()*100.0,
                    SVM_hyperd_test.std()*100.0,
                    DS_test.std()*100.0,
                    rnd_forest_test.std()*100.0,
                    )

# create plot
fig, ax = plt.subplots(figsize=(15, 10))
index = np.arange(n_groups)
bar_width = 0.3
opacity = 1
rects1 = plt.bar(index, train_accuracy, bar_width, alpha = opacity, color='Cornflowerblue', label='Train')
rects2 = plt.bar(index + bar_width, test_accuracy, bar_width, alpha = opacity, color='Teal', label='Test')
rects3 = plt.bar(index + bar_width, Standard_Deviation, bar_width, alpha = opacity, color='red', label='Standard Deviation')
plt.xlabel('Algorithm') # x axis label
plt.ylabel('Accuracy (%)') # y axis label
plt.ylim(0, 115)
plt.title('Comparison of Algorithm Accuracies') # plot title
plt.xticks(index + bar_width * 0.5, algorithms) # x axis data labels
plt.legend(loc = 'upper right') # show legend
for index, data in enumerate(train_accuracy):
    plt.text(x = index - 0.035, y = data + 1, s = round(data, 2), fontdict = dict(fontsize = 8))
for index, data in enumerate(test_accuracy):
    plt.text(x = index + 0.25, y = data + 1, s = round(data, 2), fontdict = dict(fontsize = 8))
for index, data in enumerate(Standard_Deviation):
    plt.text(x = index + 0.25, y = data + 1, s = round(data, 2), fontdict = dict(fontsize = 8))
    
```

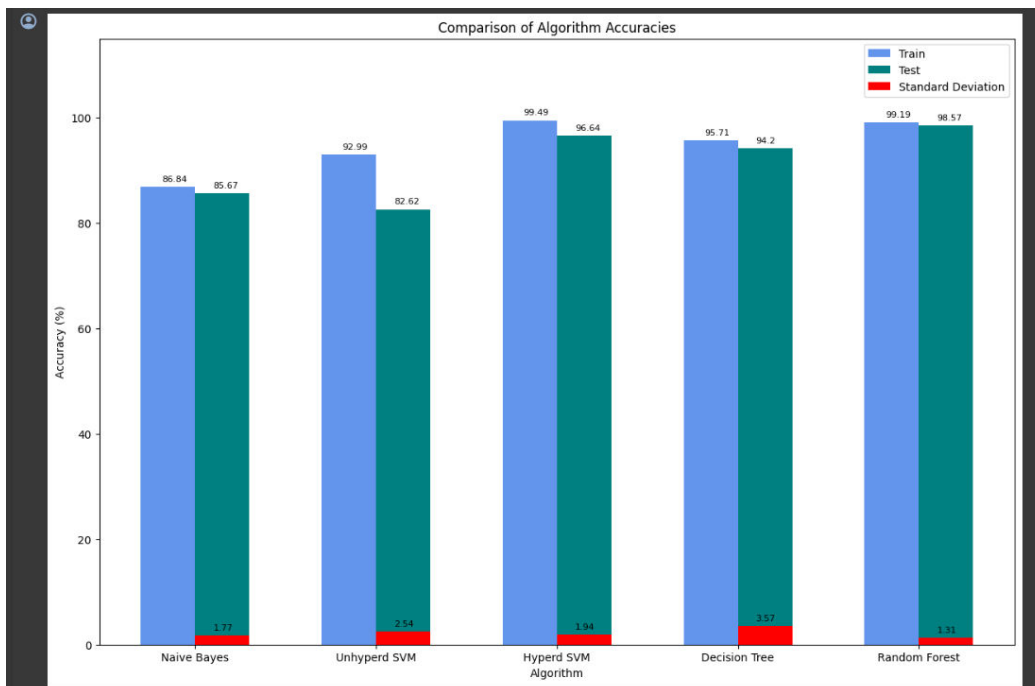


Figure 18: Comparison of Algorithm Accuracies

V.OUTCOMES

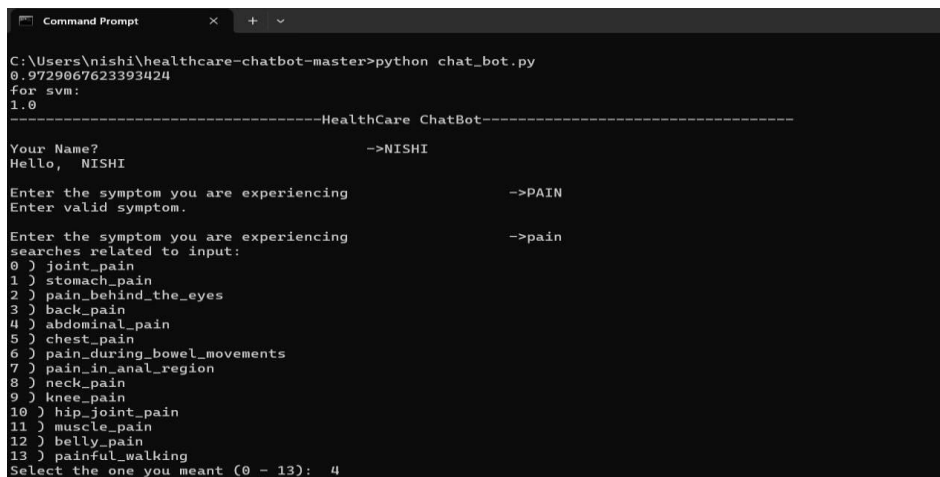
The outcome of the "Disease Prediction using Symptoms" project in Python employing machine learning models is a comprehensive application designed to assist users in assessing potential health conditions based on input symptoms. The project yields a user-friendly interface where individuals can easily input their symptoms, and the system, powered by a trained machine learning model, delivers real-time predictions of potential diseases with accompanying confidence scores. The application encompasses robust data preprocessing mechanisms to handle missing values and outliers, ensuring the quality of the input data. Prediction results are presented in a clear and interpretable manner, including relevant performance metrics to establish transparency and trust in the predictions. The project also prioritizes user privacy and data security, implementing measures to safeguard sensitive information. Thorough documentation, including user manuals and technical guides, accompanies the project, facilitating ease of use and future development. Additionally, the application undergoes rigorous testing, including user acceptance testing, to identify and address any potential issues. Upon successful testing, the application is deployed to a chosen hosting platform, making it accessible to users. Post-deployment maintenance, scalability considerations, and a user feedback mechanism contribute to the overall success of the project, with the ultimate goal of providing a valuable and reliable tool for health assessment based on symptoms.

As a future enhancement, we also look forward to executing multilingual summarization and multi-document summarization. The files which we give as input may also contain native languages, hence health records can be collected from various parts of the world and can be easily summarized using multilingual summarization. As of now, the paper proceeds with the global language (English). This paper clearly defines disease prediction using highly personalized training data sets and also some of the related tasks like fixing appointments and tracing the nearest health center

VI. RESULTS AND DISCUSSIONS

Prediction results are presented in a clear and interpretable manner, including relevant performance metrics to establish transparency and trust in the predictions. The project also prioritizes user privacy and data security, implementing measures to safeguard sensitive information. Thorough documentation, including user manuals and technical guides, accompanies the project, facilitating ease of use and future development. Additionally, the application undergoes rigorous testing, including user acceptance testing, to identify and address any potential issues. Upon successful testing, the application is deployed to a chosen hosting platform, making it accessible to users. Post-deployment maintenance, scalability considerations, and a user feedback mechanism contribute to the overall success of the project, with the ultimate goal of providing a valuable and reliable tool for health assessment based on symptoms.

In our project, information is gathered using the chatbot. It gathers the patient's medical history in addition to the symptoms, and uses the symptoms to forecast the illness. NLP and machine learning techniques are used in the creation of the disease prediction chatbot



```
C:\Users\nishi\healthcare-chatbot-master>python chat_bot.py
0.9729067623393424
for svm:
1.0
-----HealthCare ChatBot-----
Your Name?                ->NISHI
Hello, NISHI
Enter the symptom you are experiencing ->PAIN
Enter valid symptom.
Enter the symptom you are experiencing ->pain
searches related to input:
0 ) joint_pain
1 ) stomach_pain
2 ) pain_behind_the_eyes
3 ) back_pain
4 ) abdominal_pain
5 ) chest_pain
6 ) pain_during_bowel_movements
7 ) pain_in_anal_region
8 ) neck_pain
9 ) knee_pain
10 ) hip_joint_pain
11 ) muscle_pain
12 ) belly_pain
13 ) painful_walking
Select the one you meant (0 - 13): 4
```

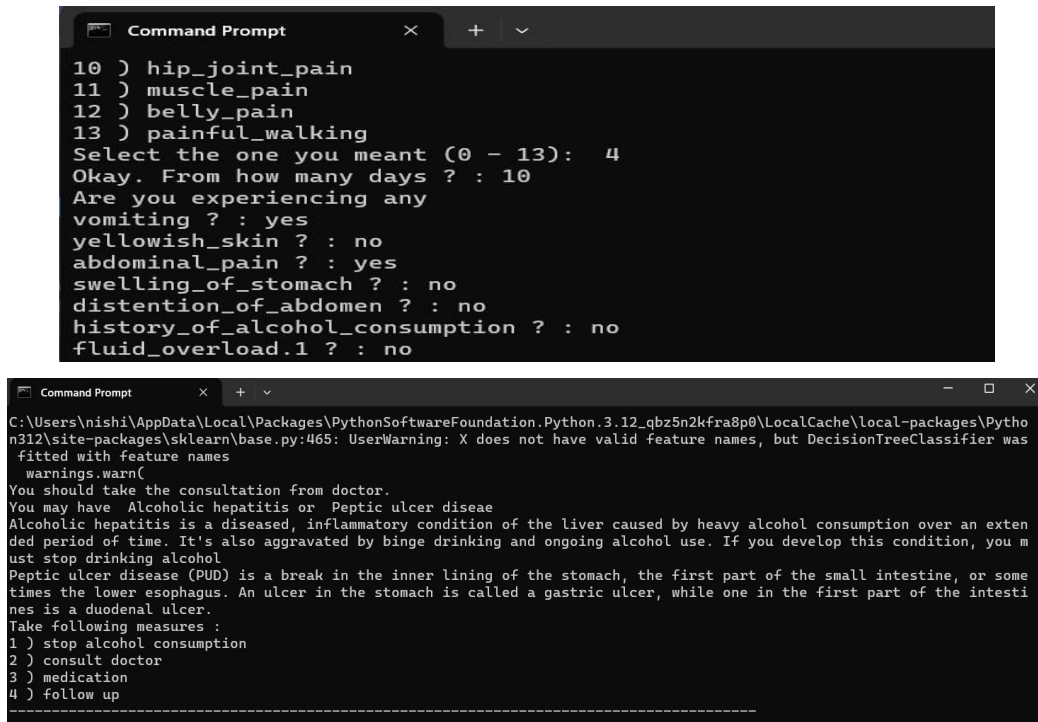


Figure 22: Chatbot

Result Analysis:

Accuracy and confusion matrices were used to analyze the results of the proposed system. Fig 4 shows the accuracy and confusion matrix for Naive bayes classifier. The accuracy of the model using Naïve Bayes is 85.58%.

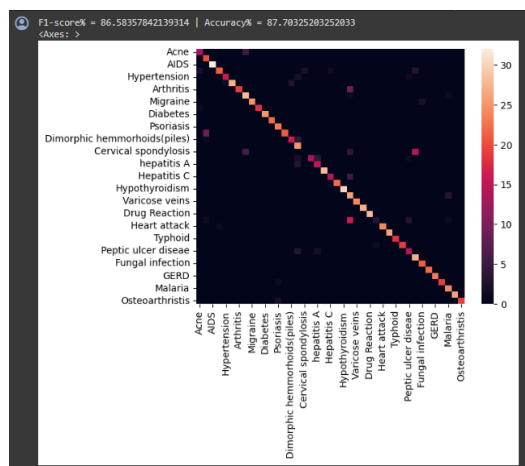


Figure 23: Performance analysis for Naive Bayes

F-1 Score is also calculated to analyze the results and performance of three models constructed. Fig 5 depicts confusion matrix, accuracy and F-1 score for Random forest algorithm. The accuracy of the model using the Random forest algorithm is 99.58%.

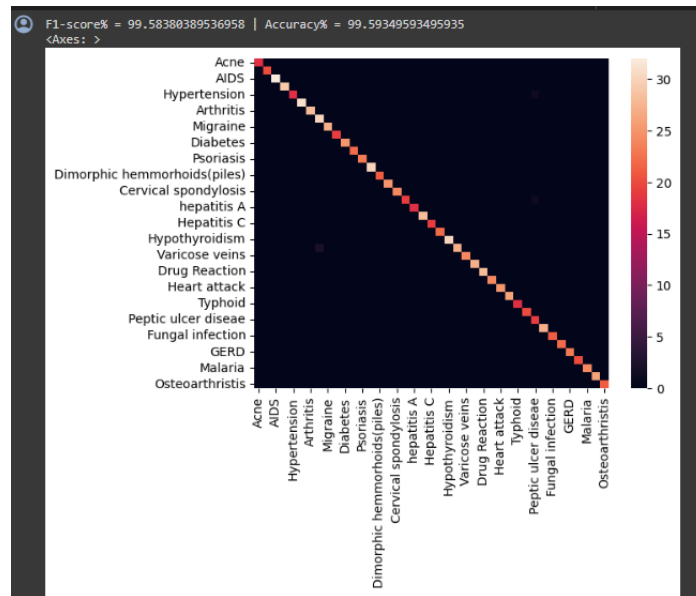


Figure 24: Performance analysis for Random Forest

Displays the decision tree classifier algorithm's F-1 score, accuracy, and confusion matrix.

Using the decision tree approach, the model's accuracy is 95.78%.

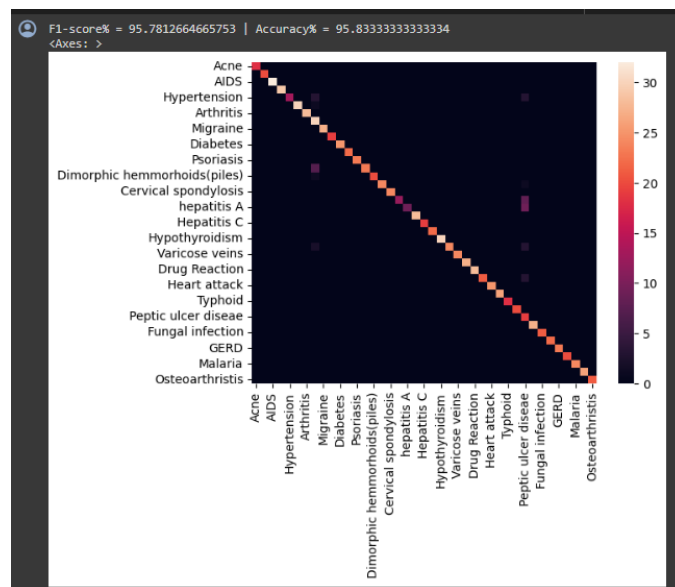


Figure 25: Performance analysis for Decision Tree

VII. CONCLUSION

The manuscript presented the technique of predicting the disease based on the symptoms, almost all the ML models gave good accuracy values. As some models were dependent on the parameters, they couldn't predict the disease and the accuracy percentage was quite low. Once the disease is predicted, we could easily manage the medical resources required for the treatment. This model would help in lowering the cost required in dealing with the disease and would also improve the recovery process.

The problems faced by the medical industry with the unaffordability of the patients to seek dictators and the unavailability of the medical staff can be diminished. This can happen by automating the channelization of the patients to a specialist instead of a generalist. This can happen via the use of a disease prediction system. This system will input the patient's symptoms and produce possible disease as an output with 97% accuracy as compared to earlier models. The proposed model can assist the healthcare industry by:

1. Reduction in healthcare costs: By improving patient outcomes and reducing the need for unnecessary tests and treatments, disease prediction applications can help reduce healthcare costs and improve the overall efficiency of the healthcare system.
2. Improved patient outcomes: By providing healthcare providers with valuable insights into a patient's disease risk, disease prediction applications can help improve patient outcomes by allowing for earlier and more effective interventions.
3. Early diagnosis: By analyzing patient data and identifying risk factors for specific diseases, disease prediction applications can help healthcare providers make an early diagnosis, which is critical for improving patient outcomes. At last, we conclude that our model can provide increased accuracy and a reliable model for the prediction of the disease through symptoms.

REFERENCES

1. IEEE: Diseases Prediction based on Symptoms using Database and GUI
<https://ieeexplore.ieee.org/document/9753707>
2. DISEASE DIAGNOSIS AND RECOMMENDED REMEDY
<https://ieee-dataport.org/documents/disease-diagnosis-and-recommended-remedy>
3. Symptoms Based Disease Prediction Using Machine Learning Techniques
<https://ieeexplore.ieee.org/document/9388603>
4. Disease Prediction using Symptoms based on Machine Learning Algorithms
<https://ieeexplore.ieee.org/document/9914945>
5. Prediction of Diseases Using Different Machine Learning Approaches
<https://ieeexplore.ieee.org/document/9853132>
6. Anjana, R. M., Pradeepa, R., & Deepa, M. (2011). Cardiovascular diseases in India. *Indian Heart J*, 63(4), 243-249.
7. Gupta, V., Chandragiri, V., & Reddy, K. J. (2017). Non-communicable diseases in India: an epidemiological overview. *Int J Med Res Health Sci*, 6(2), 35-43.
8. Jain, K., Aggarwal, N., & Mittal, R. (2017). Brain tumor detection and classification using machine learning approaches: A review. *World Journal of Neuroscience*, 8(4), 269-281.
9. Kanjani, R., Agarwal, V., & Bhattacharya, P. (2015). Machine learning in healthcare: challenges and opportunities. *Journal of Medical Systems*, 39(10), 88.
10. Kaur, A., Singh, B., & Mittal, S. (2021). Automated brain tumor detection and classification using transfer learning and deep learning techniques. *Journal of King Saud University - Computer and Information Sciences*, 33(8), 3543-3555.
11. Mehta, R., Goyal, R., Khosla, V., & Mittal, A. (2019). Brain tumor detection and classification using deep learning: a review. *Int J Comput Appl*, 187(8), 40-45. Mishra, A., Kumar, V., & Rathee, G. (2018). Machine learning and data mining for predicting disease outbreaks: a survey. *arXiv preprint arXiv:1803.05407*.
12. Muthu Rama Krishnan, V., & Gnanasekaran, R. (2020). Brain tumor detection using deep learning algorithms: a review. *International Journal of Engineering and Computer Science*, 7(8), 21681-21685.
13. Patel, V., Gandhi, T., Shah, N., & Panchal, S. (2018). Heart disease prediction using machine learning. In 2018 5th International Conference on Signal Processing and Integrated Networks (ICSIM) (pp. 753-758). IEEE.
14. Pereira, F., Pinto, N., Aguiar, P., & Fernandes, T. (2016). Brain tumor segmentation using convolutional neural networks: a preliminary study. In *Iberian Conference on Pattern Recognition and Image Analysis* (pp. 399-406). Springer, Cham.



INNO  **SPACE**
SJIF Scientific Journal Impact Factor
Impact Factor: 8.379

doi[®]
CROSS **ref**

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 **9940 572 462**  **6381 907 438**  **ijircce@gmail.com**



www.ijircce.com

Scan to save the contact details