JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

LIFESTYLE BASED DISEASE PREDICTION

[1]Gude Manikanta, [2] G.Vijaya lakshmi,
[1]MCA 2nd Year, [2]Assistant Professor,
[1]Master of Computer Applications,
[1]Sanketika Vidya Parishad Engineering college, Visakhapatnam, India

ABSTRACT

The current burden of chronic diseases reflects the cumulative effects of unhealthy lifestyles and the resulting risk factors over the life span of people. Lifestyle diseases is common among the population today not only in India but also in almost every country. Lifestyle diseases are caused because of the habits that we have in day-to-day basis. It includes heart disease, hypertension etc. which all may heard of. We also came across many such people who died because they are not aware of their disease and left with no appropriate time for the treatment. That is why, we decided to develop the model which will analyze the data entered by the user and will give the predictions of the diseases which he or she may have chances to suffer from. This study aims to understand Random forest algorithm and use it to predict lifestyle diseases that an individual might be susceptible to. This project aware the person about his health so that he will have the treatment well in time if required and will save the lives of many people. This project covers three main aspects which are prediction, prevention and management of lifestyle diseases.

KEYWORDS: — Lifestyle, heart disease, hypertension, Random forest.

INTRODUCTION

Lifestyle-based disease prediction using machine learning is a growing field that uses artificial intelligence and machine learning algorithms to analyze various lifestyle factors and predict the likelihood of developing certain health conditions. These conditions can range from chronic diseases like heart disease, diabetes, and cancer to lifestyle-related illnesses like obesity and high blood pressure.

Machine learning algorithms analyze data from various sources, such as electronic health records, wearable devices, and self-reported data, to identify patterns and correlations between lifestyle factors and disease. The algorithms use this information to make predictions about an individual's future health based on their current lifestyle habits.

Some of the lifestyle factors that are commonly analyzed in lifestyle-based disease prediction include diet, physical activity, stress levels, sleep patterns, smoking habits, and alcohol consumption. The goal of this type of disease prediction is to help individuals make proactive changes to their lifestyles that can reduce their risk of developing certain diseases.

The use of machine learning in lifestyle-based disease prediction has the potential to revolutionize the way that healthcare professionals approach disease prevention and treatment. By providing more personalized and accurate predictions, it can help individuals make more informed decisions about their health and take proactive steps to reduce their risk of developing certain diseases.

Monitoring a person's lifestyle over a long period of time can be helpful for the following

EARLY IDENTIFICATION OF THE LIFESTYLE

A person is generally at risk of a lifestyle disease when he/she lives his/her life in a certain style. Moreover, initial signs and symptoms of such diseases appear in people with irregular life patterns. By long term activity monitoring, it would be possible to identify such styles and determine the risk of the disease before the actual disease appears.

PREVENTION OF LIFESTYLE DISEASES

It would be possible to prevent a disease at the first place with the identification of the lifestyle. A healthy lifestyle and physical activity help prevent obesity, heart disease, hypertension, diabetes, colon cancer, and premature mortality.

BACKGROUND AND MOTIVATION

The background and motivation for a study or project on lifestyle-based disease prediction using machine learning would provide context for why this area of research is important and why it is necessary to use machine learning algorithms to address the problem. A background and motivation section might look something like this:

BACKGROUND

Chronic diseases such as heart disease, diabetes, and cancer are leading causes of morbidity and mortality worldwide. While many of these diseases can be prevented through lifestyle modifications, individuals often lack the knowledge or motivation to make necessary changes. In order to effectively prevent and manage these diseases, it is important to accurately predict an individual's risk of developing these conditions based on their lifestyle habits.

MOTIVATION

The motivation behind using machine learning for lifestyle-based disease prediction is to provide individuals with more accurate and personalized predictions about their risk of developing certain diseases. The use of machine learning algorithms can help to overcome

some of the limitations of traditional approaches to disease prediction, such as relying on demographic data or using a one-size-fits-all approach to disease prevention.

By analyzing a wide range of lifestyle factors, machine learning algorithms can provide a more nuanced and comprehensive view of an individual's health and risk of disease. This information can then be used to create personalized recommendations for disease prevention that are tailored to the individual's specific needs and lifestyle habits.

In addition to providing more accurate and personalized predictions, the use of machine learning in lifestyle-based disease prediction can also help to improve disease prevention and management by providing individuals with more actionable and meaningful information about their health. By identifying specific lifestyle factors that contribute to disease risk, machine learning algorithms can help individuals to understand the impact of their habits on their health and make informed decisions about how to improve their health outcomes.

PROPOSED METHOD

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

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.

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.

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.

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, these helps the user to overcome or reduce the intensity of the disease.

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.

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.

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.

Overall, a proposed system for lifestyle-based disease prediction using machine learning would be designed to provide accurate predictions about the likelihood of a user developing a particular disease based on their lifestyle factors, and to provide feedback and reporting mechanisms to help the user and healthcare providers monitor and manage their health.

SYSTEM REQUIREMENTS

Functional Requirements

Here are some functional requirements for a web application for lifestyle-based disease prediction:

Data visualization

The application should provide an interactive data visualization that enables users to see their data history and trends. Users should be able to compare their data against standard medical guidelines.

Integration with wearables

The application should be able to integrate with various wearable devices such as fitness trackers, smartwatches, and heart rate monitors to collect data

Security and privacy: The application should ensure the privacy and security of users' personal and health information. The application should follow best practices to protect user data such as encryption, two-factor authentication, and regular backups.

User support

The application should provide users with technical support through multiple channels such as email, chat, and phone support.

These are just some examples of the functional requirements that a web application for lifestyle-based disease prediction may have.

Collecting input parameters

We listed out the resources like stress, BMI, Age, Exercise which are required to predict the chance of a disease the user may suffer from.

Creating models

We Created models that creates the efficient technologies to save and conserve the natural resources, increase the production and productivity while concurrently conserving the environment.

TRAINING

We train the data using machine learning algorithms like random forest to predict the Disease the user should take care of for the given input parameters.

Testing

Tested The model weather the predicted disease is correct or not by checking the data collected and analyzed by

Disease prediction

The application should be able to predict the likelihood of a user developing a particular lifestyle-based disease based on their lifestyle factors. The application should also be able to provide suggestions to the user to prevent the disease.

Disease prediction
The application should be able to predict the likelihood of a user developing a particular lifestyle-based disease based on their lifestyle factors. The application should also be able to provide suggestions to the user to prevent the disease.
Code Management
Code management in the process of handling changes to computer code. code management deals in the precise management of code modules or collections of lines of code in order to support changes or particular goals such as maintenance or debugging

Non-functional Requirements
Scalability
Our system must be able to accommodate larger amounts of datasets which are used for analyzing and prediction for a long time i.e that the system must be able to accommodate larger volumes (whether of users, throughput, data) over time, and also includes NFRs such as elasticity, which is the ability to scale up and down quickly, as needed.
Maintainability
Our system should be able to maintain the large amounts of datasets and it should be ease with which faults in a software system can be found and fixed. Maintainability requirements address the user concern for how easy it is to upkeep and repair the system.
Performance
In this project our system must perform with greater accuracy. It defines how well the software system accomplishes certain functions under specific conditions. Includes the software's speed of response, throughput, execution time and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks.
Adaptability
The adaptability of our system is to have the ability of a to adapt to change in an environment without any change in its behavior.
Flexibility
Flexibility requirements of our system should address the user concern for how easy it is to modify the system to work in different

SRS Table:

S.NO	Requirements	Requirement Number	Essential/Desirable	Description
1.	Collecting input parameters	RS1	Essential	Collecting the input values and datasets
2.	Creating models	RS2	Essential	Descriptive diagram of relationships between various types of information in the dataset is to be created.
3.	Training	RS3	Essential	Need to train the data for prediction.
4.	Testing	RS4	Essential	Testing the model whether predicted result is correct or not.

5.	Prediction	RS5	Essential	If there is a chance of any disease, it is to be predicted.
6.	Display result	RS6	Essential	The final result is displayed on the web app.

Hardware Requirements
Good working laptop or System, keyboard, Mouse
Software Requirements
Operating system
Windows XP / 7/8/10 /11 –
Windows are highly preferred operating systems for web developers. Easy and flexible to use. Windows is considered one of the best OS for programmers. Windows 10 is good for programming as it supports many programs and languages. Windows 10 is an improved version over the other versions of Windows and comes with various customization and compatibility options
Coding Languages
Python
The practical implementation of Python in machine learning projects and tasks has made the work easier for developers, data scientists, and machine learning engineers. Python can be easily used to analyze and compose available data, which also makes it one of the most popular languages in data science.
Flask
Flask is also a Python-based microframework that is used for web application development.
Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided.
HTML
HTML stands for Hyper Text Markup Language.

HTML is the language for describing the structure of Web pages. HTML gives authors the means to: Publish online documents with headings, text, tables, lists, photos, etc. Retrieve online information via hypertext links, at the click of a button.

Methodology
IMPLEMENTATION DETAILS
Solution to the problem:
Our system predicts the chances of Chronic diseases such as heart disease, diabetes, and cancer are major health challenges facing societies worldwide. Lifestyle factors, such as diet, physical activity, stress levels, and sleep patterns, have a significant impact on disease risk. In this study, we aimed to use machine learning algorithms to analyze individual lifestyle factors and predict the likelihood of developing certain diseases.
Steps involve in data set processing
1.Import the libraries.
2.Import the data set.
3.Get the basic information about the data set.
4.Get the correlation parameter for the variables.
5.If required plot and analyse the parameters.
6.And also need to purify the data set.
7.Check for nulls in the data set.
8.If any drop those rows.
9.Check for any special characters in the columns.
10.Just convert them in to nulls and drop those rows.
11.Now need to select the features for the project
12.Taking all the inputs and predicts the chances of disease as output.
13.Now need to split the data set into sample and verification data set.
14.And train the machine using the Randomforest.
15.And finally using this trained model is routed to the flask application which can create an application to work.
16.To take the inputs from the user and display the result.

Algorithm
Random Forest is a machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the model.
The Random Forest algorithm works by creating multiple decision trees, each using a random subset of the training data and a random subset of the features for each split. Each decision tree is trained independently, and the final prediction is obtained by aggregating the predictions of all the trees in the forest.
The algorithm works as follows:
1.Select a random subset of the training data and a random subset of the features.

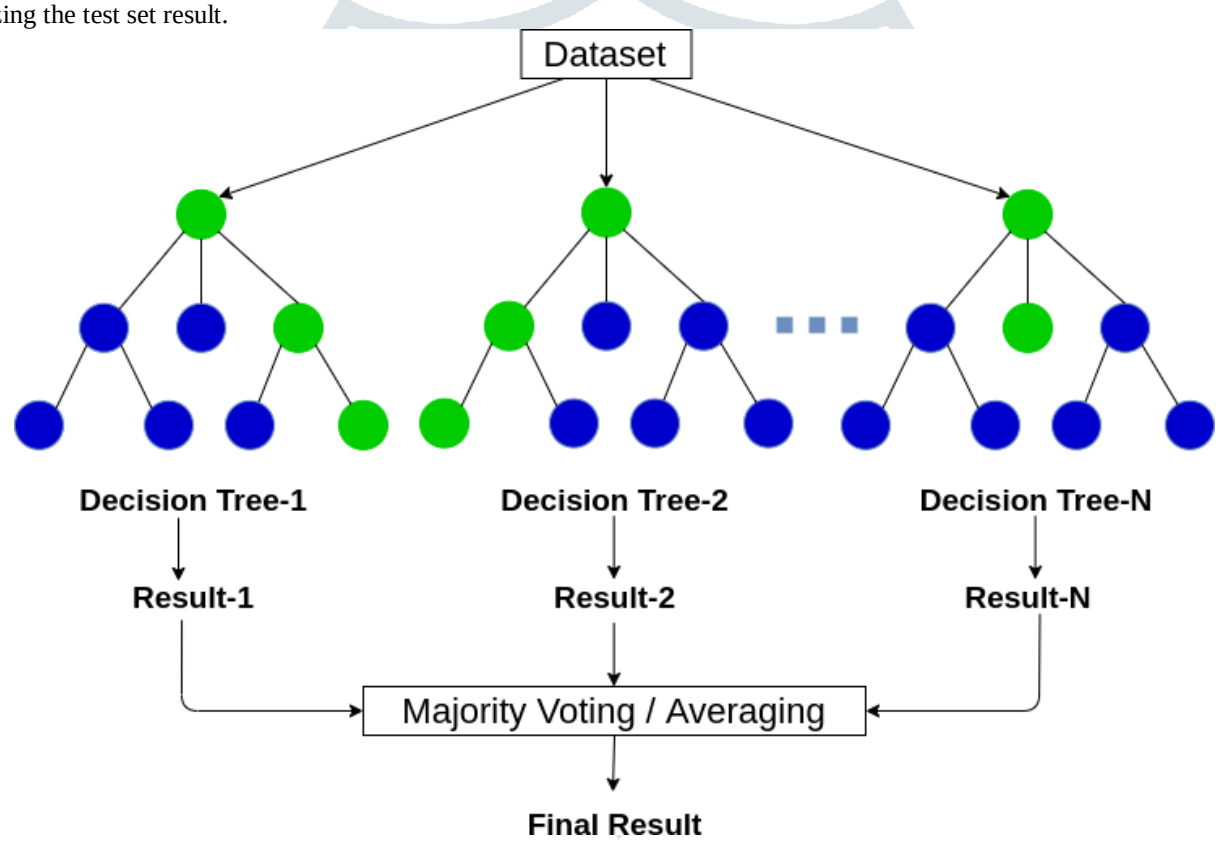
2.Build a decision tree based on the selected data and features.
3.Repeat steps 1 and 2 to create multiple decision trees.
4.To make a prediction for a new data point, pass the data point through all the decision trees and take the majority vote of the predictions.
Random Forest has several advantages over traditional decision trees, including:
1.It reduces overfitting by aggregating the predictions of multiple trees.
2.It can handle both categorical and continuous data.
3.It can handle missing data.
4.It is less sensitive to the choice of hyperparameters compared to other algorithms.
5.It can provide feature importance measures, which can be used for feature selection and understanding the importance of different features in the model.
6.Random Forest has been widely used in many applications such as bioinformatics, finance, and remote sensing, and it has been shown to perform well in both classification and regression tasks.

Python Implementation of Random Forest Algorithm:

Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user_data.csv", which we have used in previous classification models. By using the same dataset, we can compare the Random Forest classifier with other classification models such as Decision tree Classifier , KNN, Logistic Regression, etc.

Implementation Steps are given below:

1.Data Pre-processing step
2.Fitting the Randomforest algorithm to the Training set
3.Predicting the test result
4.Test accuracy of the result (Creation of Confusion matrix)
5.Visualizing the test set result.



Dataset

Datasets containing values 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 are loaded into the decision tree algorithm. These datasets contain values of different scenarios in the fields in order to train the model accurately.

After applying random forest algorithm on the sensed datasets, an output containing the decision i.e which disease may occur. This output containing decision is accessed by the users through a Web application.

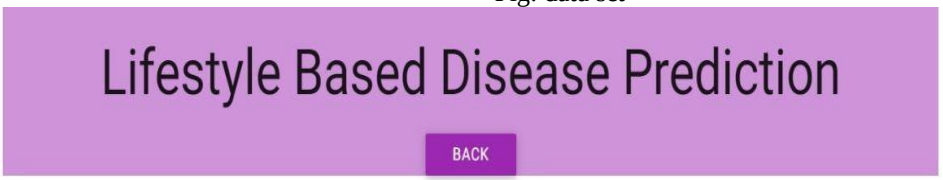
We can collect the previous data and analysis for our analysis and to obtain an efficient result.

In [3]:
```
data.head()
```

Out[3]:

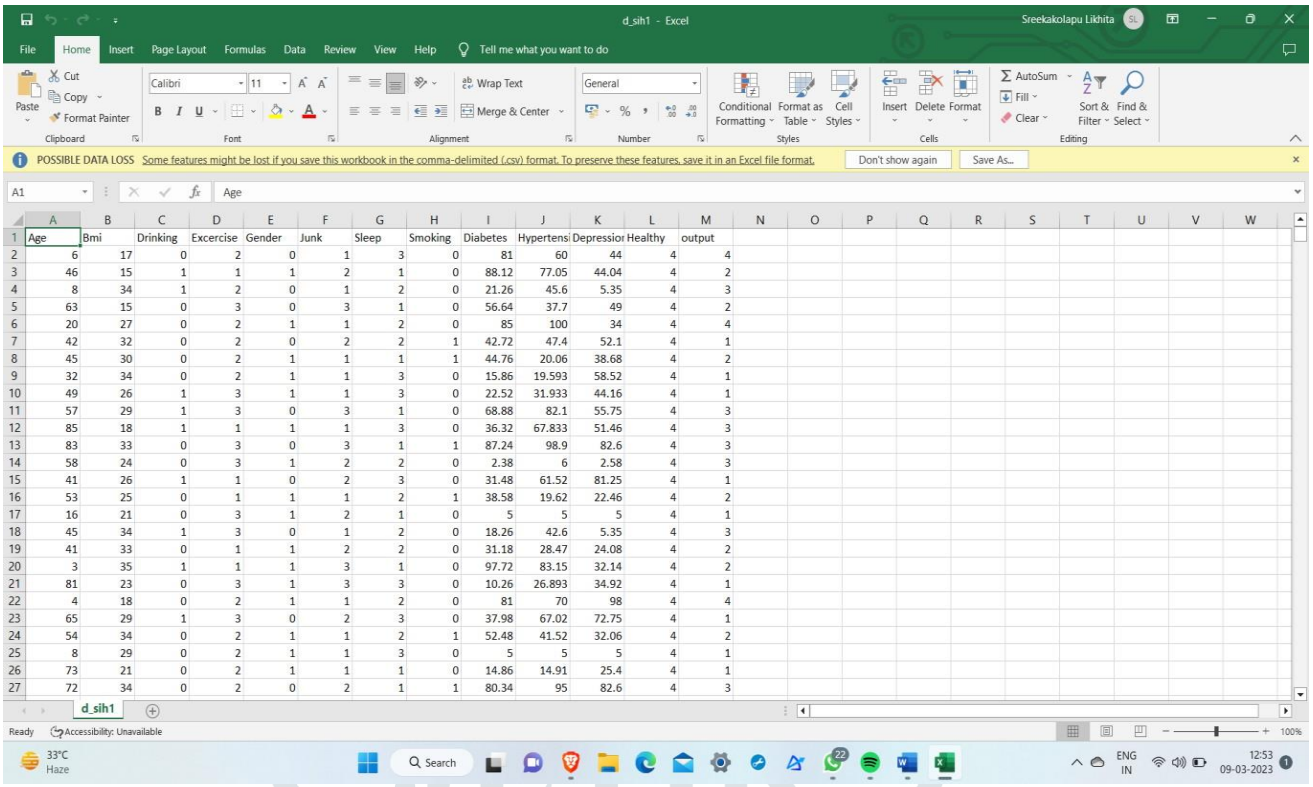
	Age	Bmi	Drinking	Excercise	Gender	Junk	Sleep	Smoking	Diabetes	Hypertension	Depression	output
0	46	15	1	1	1	2	1	0	88.12	77.05	44.04	2
1	8	34	1	2	0	1	2	0	21.26	45.60	5.35	3
2	63	15	0	3	0	3	1	0	56.64	37.70	49.00	2
3	42	32	0	2	0	2	2	1	42.72	47.40	52.10	1
4	45	30	0	2	1	1	1	1	44.76	20.06	38.68	2

Fig: data set



	Age	Bmi	Drinking	Excercise	Gender	Junk	Sleep	Smoking	Diabetes	Hypertension	Depression	output
0	46	15	1	1	1	2	1	0	88.12	77.050	44.04	2
1	8	34	1	2	0	1	2	0	21.26	45.600	5.35	3
2	63	15	0	3	0	3	1	0	56.64	37.700	49.00	2
3	42	32	0	2	0	2	2	1	42.72	47.400	52.10	1
4	45	30	0	2	1	1	1	1	44.76	20.060	38.68	2
5	32	34	0	2	1	1	3	0	15.86	19.593	58.52	1
6	49	26	1	3	1	1	3	0	22.52	31.933	44.16	1
7	57	29	1	3	0	3	1	0	68.88	82.100	55.75	3
8	85	18	1	1	1	1	3	0	36.32	67.833	51.46	3
9	83	33	0	3	0	3	1	1	87.24	98.900	82.60	3

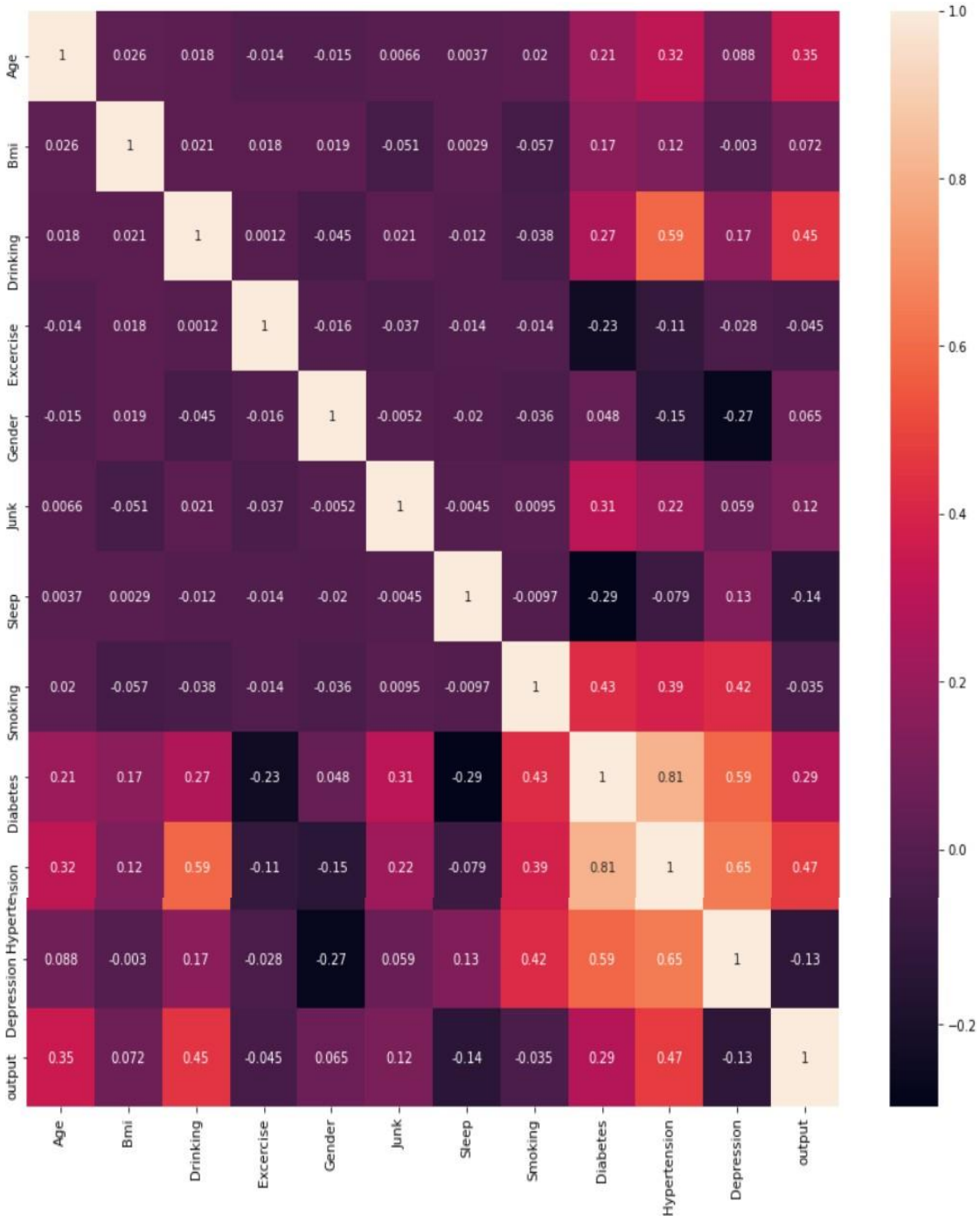
Fig.9 Dataset



Age	Bmi	Drinking	Excercise	Gender	Junk	Sleep	Smoking	Diabetes	Hypertens	Depressior	Healthy	output
6	17	0	2	0	1	3	0	81	60	44	4	4
46	15	1	1	1	2	1	0	88.12	77.05	44.04	4	2
8	34	1	2	0	1	2	0	21.26	45.6	5.35	4	3
63	15	0	3	0	3	1	0	56.64	37.7	49	4	2
20	27	0	2	1	1	2	0	85	100	34	4	4
42	32	0	2	0	2	2	1	42.72	47.4	52.1	4	1
45	30	0	2	1	1	1	1	44.76	20.06	38.68	4	2
32	34	0	2	1	1	3	0	15.86	19.593	58.52	4	1
49	26	1	3	1	1	3	0	22.52	31.933	44.16	4	1
57	29	1	3	0	3	1	0	68.88	82.1	55.75	4	3
85	18	1	1	1	1	3	0	36.32	67.833	51.46	4	3
83	33	0	3	0	3	1	1	87.24	98.9	82.6	4	3
58	24	0	3	1	2	2	0	2.38	6	2.58	4	3
41	26	1	1	0	2	3	0	31.48	61.52	81.25	4	1
53	25	0	1	1	1	2	1	38.58	19.62	22.46	4	2
16	21	0	3	1	2	1	0	5	5	5	4	1
45	34	1	3	0	1	2	0	18.26	42.6	5.35	4	3
41	33	0	1	1	2	2	0	31.18	28.47	24.08	4	2
3	35	1	1	1	3	1	0	97.72	83.15	32.14	4	2
81	23	0	3	1	3	3	0	10.26	26.893	34.92	4	1
4	18	0	2	1	1	2	0	81	70	98	4	4
65	29	1	3	0	2	3	0	37.98	67.02	72.75	4	1
54	34	0	2	1	1	2	1	52.48	41.52	32.06	4	2
8	29	0	2	1	1	3	0	5	5	5	4	1
73	21	0	2	1	1	1	0	14.86	14.91	25.4	4	1
72	34	0	2	0	2	1	1	80.34	95	82.6	4	3

Fig.10 Dataset

Heatmap:



Black box testing:

Black box testing is a type of testing where the tester does not have access to the internal workings of the system being tested. In other words, the tester is not concerned with the implementation details or code of the system, but rather the external behavior and functionality of the system. Black box testing is focused on testing the system's inputs and outputs, and verifying that the system meets the specified requirements.

In the context of a lifestyle-based disease prediction system using machine learning, black box testing could involve:

Testing the input validation: The tester would provide various types of inputs to the system, such as valid and invalid user lifestyle data, and ensure that the system is able to handle each type of input appropriately.

Testing the prediction accuracy: The tester would provide various types of input data where the predicted disease is known, and compare the system's prediction to the known outcome to ensure that the system is accurately predicting diseases based on the user's lifestyle factors.

Testing the user interface: The tester would evaluate the user interface and test the system's ability to guide users through the input data process and present prediction results in a clear and understandable format.
Testing the system response time: The tester would provide the system with input data and measure the time it takes for the system to provide a prediction to ensure that the system responds within an acceptable time frame.
Testing the handling of unexpected input: The tester would provide invalid or unexpected input data to the system and ensure that the system provides an appropriate error message.
By conducting black box testing, it can be ensured that the system is functioning properly and accurately predicting diseases based on the user's lifestyle factors without the need for the tester to have knowledge of the system's internal workings or implementation details.

White Box Testing:

White box testing is a type of testing where the tester has access to the internal workings of the system being tested. In other words, the tester is concerned with the implementation details or code of the system, and uses this knowledge to design test cases to verify that the system functions correctly. White box testing is focused on testing the system's internal logic and verifying that the system is performing operations as intended.
In the context of a lifestyle-based disease prediction system using machine learning, white box testing could involve:
Testing the machine learning algorithms: The tester would analyze the machine learning algorithms used in the system and design test cases to ensure that the algorithms are functioning correctly and providing accurate predictions based on the user's lifestyle factors.
Testing the data storage and retrieval: The tester would examine the database used in the system and design test cases to ensure that the system is correctly storing and retrieving user lifestyle data.
Testing the system's error handling: The tester would design test cases to verify that the system is properly handling errors such as invalid input data or database errors.
Testing the system's security: The tester would examine the system's security measures such as encryption and access control and design test cases to ensure that user data is being protected and secured properly.
Testing the system's performance: The tester would analyze the system's code and design test cases to measure the system's performance, such as the time it takes to provide a prediction or the system's ability to handle a large number of requests at the same time.
By conducting white box testing, it can be ensured that the system's internal logic and implementation details are functioning properly, and that the system is performing operations as intended.

TEST CASES:

Test Case 1:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
1.	RS1	Collecting input parameters	1.input the valid parameter values in their specified order. 2.mount the valid dataset	dataset	Data should be mounted and collected successfully	Dataset is displayed in the output after mounting it.	PASS

Test Case 2:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
2.	RS2	Creating models	Descriptive diagram of relationship between various types of information in the dataset is to be created	ML Algorithm (Random forest)	Graph should be created	Graph is created	PASS

Test Case 3:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
3.	RS3	Training	By analyzing the dataset, data model is to be created i.e Accuracy score is provided	Dataset	Accuracy score should be provided	Accuracy score is provided	PASS

Test Case 4:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
4.	RS4	Testing	Testing the model weather predicted result is correct or not.	ML Algorithm (Random forest)	Should predict a disease	Testing whether predicted disease is correct or not	PASS

Test Case 5:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
5.	RS5	Prediction	By considering the user inputs the disease is predicted	Dataset	Should predict the disease	Risk of disease is predicted	PASS

Test Case 6:

S.NO	Requirement id	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
6.	RS6	Display the result in the web application	checking the model weather predicted result is correct or not.	Disease data	Should predict the risk of the disease	Disease is predicted and precautions are provided according to the disease	PASS

CONCLUSION

In conclusion, this study demonstrates the potential of machine learning for improving the accuracy and effectiveness of disease prediction, and for empowering individuals to take a proactive approach to their health and reduce their risk of developing certain diseases."

Further additions to the model would include when an individual enters his/her details (i.e., input to the predictive model), the model would determine his/her identity based on several inputs, show an individual's current status of his/her health contrary to a desired ideal health using graphs, let know lifestyle changes, provide balanced diet and doctor consultations, recommend exercises, etc. The model would take into account climatic conditions and pollution levels and rank cities and suburbs with an ideal environment as to the precautionary measures that an individual could take making the model more content specific, accessible, and flexible in terms of customization. The fact that Deep Learning (DL) is overtaking ML algorithms in terms of accuracy would suggest the possibility of SVM being replaced by DL in the near future

REFRENCES

[1] Sharma, M. and Majumdar, P.K., 2009. Occupational lifestyle diseases: An emerging issue. Indian Journal of Occupational and Environmental medicine, 13(3), pp. 109–112.

[2] Anand, A. and Shakti, D., 2015. Prediction of diabetes based on personal lifestyle indicators. In Next generation computing technologies (NGCT), 2015 1st international conference on (pp. 673– 676). IEEE.

[3] Sayali Ambekar and Dr.Rashmi Phalnikar, 2018. Disease prediction by using machine learning, International Journal of Computer Engineering and Applications, vol. 12, pp. 1–6.

[4] Mishra, A.K., Keserwani, P.K., Samaddar, S.G., Lamichaney, H.B. and Mishra, A.K., 2018. A decision support system in healthcare prediction. In Advanced Computational and Communication Paradigms (pp. 156–167). Springer, Singapore.

[5]Explaining the basics of machine learning, algorithms and applications, Available [Online] https://www.hackerearth.com/blog/machine-learning/explainingbasics-machine-learning- algorithms-applications/ [Accessed on June 27, 2018].

[6]https://upload.wikimedia.org/wikipedia/commons/thumb/b/b5/Svm_ separating_hyperplanes_%28SVG%29.svg/220pxSvm_separating_hyperplanes_%28SVG%29.svg.png.

BIBLOGRAPHY

Gude manikanta is Studying His 2nd Year, Master of Computer Applications In Sanketika Vidya Parishad Engineering College, Affiliated To Andhra University, Accredited By NAAC Visakhapatnam.With His Interest In Python Language, And Machine Leaning Method As A Part Of Academic Project, he Used A Driving Decision Strategy DDS For an Autonomous Vehicle Based on Machine Learning As A Result Of Desired To Comprehend The Flaws In Conventional Reporting And To Preserve Timely And High Quality Report Output In Driving Decision Strategy. A Completely Developed Project Along With Code Has Been Submitted For Andhra University As An Academic Project. In Completion Of His MCA

G.Vijaya laxshmi working as Assistant professor in Master of computer application (MCA) Sanketika Vidya Parishad Engineering college,Visakhapatnam,Andhrapradesh.with 6 years of experience in master of computer applications(MCA),Accredited by NAAC.with her area of interests in C,Computer Organization, Software Engineering,IOT.