

DISEASE PREDICTION USING MACHINE LEARNING

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

MAMIDALA HARSHAVARDHAN

21UK1A6695

Under the guidance of

Ms. R. Swathi

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE**

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING(AI&ML)**

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)



CERTIFICATE OF COMPLETION **INDUSTRY ORIENTED MINI PROJECT**

This is to certify that the UG Project Phase-1 entitled “DISEASE PREDICTION USING MACHINE LEARNING” is being submitted by BANDI ROHITHA (21UK1A66C1), MAMIDALA HARSHAVARDHAN (21UK1A6695), YAKARA VIJAYKANTH (21UK1A6693), NAREDLA RAJU (21UK1A6677) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025.

Project Guide

Ms. R. Swathi

(Assistant Professor)

HOD

Dr. K. Sharmila

(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO**, Principal, Vaagdevi Engineering

College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr. K. SHARMILA**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the Project Phase-1.

We express heartfelt thanks to the guide, **R.SWATHI**, Assistant professor Department CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout thesis.

MAMIDALA HARSHAVARDHAN

(21UK1A6695)

ABSTRACT

Abstract:

In recent years, disease prediction using machine learning techniques has garnered significant attention due to its potential to revolutionize healthcare by providing early diagnosis and proactive

treatment strategies. Machine learning models utilize various algorithms to analyze medical data and extract meaningful patterns that can predict the likelihood of individuals developing certain diseases. This abstract explores the application of machine learning in disease prediction, highlighting its importance in identifying risk factors, improving prognosis accuracy, and ultimately enhancing patient outcomes. Key challenges such as data quality, model interpretability, and scalability are addressed alongside promising avenues for future research and implementation. The integration of machine learning into healthcare systems promises to empower clinicians with predictive insights, optimize resource allocation, and ultimately lead to more personalized and effective healthcare interventions.

TABLE OF CONTENTS:-

1.INTRODUCTION	5
1.1 OVERVIEW... ..	5

1.2 PURPOSE	5
2.LITERATURE SURVEY	8
1.3 EXISTING PROBLEM	8
1.4 PROPOSED SOLUTION	8-9
3.THEORITICAL ANALYSIS... ..	10
1.5 BLOCK DIAGRAM	10
1.6 HARDWARE /SOFTWARE DESIGNING	10-11
2. EXPERIMENTAL INVESTIGATIONS	12-13
3. FLOWCHART... ..	14
4. RESULTS... ..	15-18
5. ADVANTAGES AND DISADVANTAGES... ..	19
6. APPLICATIONS	20
7. CONCLUSION	20
8. FUTURE SCOPE... ..	21
9. BIBILOGRAPHY	22-23
10. APPENDIX (SOURCE CODE)&CODE SNIPPETS	24-30

1.INTRODUCTION

1.1.OVERVIEW

Overview:

Disease prediction using machine learning represents a transformative approach in healthcare aimed at early detection, prevention, and management of diseases through advanced computational techniques. This overview delves into the application of machine learning models in disease prediction, highlighting their capability to analyze diverse datasets—including clinical records, genetic information, and environmental factors—to forecast disease onset, progression, and outcomes. Key methodologies such as feature selection, model training, and validation are discussed, along with the challenges of data integration, model interpretability, and scalability.

Machine learning-based disease prediction offers significant advantages by enabling personalized medicine, optimizing treatment strategies, and improving patient outcomes. By identifying patterns and correlations within large datasets, these models contribute to more accurate risk assessment and proactive healthcare interventions. The overview emphasizes the potential impact of machine learning in enhancing healthcare delivery, resource allocation, and public health initiatives. Future directions focus on enhancing predictive accuracy, incorporating real-time data sources, and ensuring ethical considerations in data usage and model deployment. Leveraging machine learning in disease prediction underscores its pivotal role in advancing precision medicine and transforming healthcare into a proactive and personalized endeavor. Disease prediction using machine learning involves gathering medical data, preprocessing it for analysis, selecting relevant features, choosing and training a suitable model, evaluating its performance, and deploying it for predictions. Key steps include data collection, preprocessing, model selection and training, evaluation, and ethical considerations for responsible use in healthcare.

1.2.PURPOSE

Prediction Predicting diseases using machine learning serves several critical purposes in healthcare and public health:

1. **Early Detection and Diagnosis:** Machine learning models can analyze vast amounts of data, including patient records, genetic information, and environmental factors, to identify patterns that indicate early stages of diseases. This early detection allows for timely interventions and treatments, potentially improving patient outcomes and reducing healthcare costs.
2. **Personalized Medicine:** By analyzing individual patient data, machine learning algorithms can tailor treatment plans and interventions based on specific risk factors and characteristics. This approach, known as personalized or precision medicine, aims to optimize treatment efficacy and minimize adverse effects.
3. **Public Health Surveillance:** Machine learning can analyze population-level data from sources like electronic health records, social media, and environmental sensors to detect disease outbreaks or trends. This capability enhances public health surveillance systems, enabling rapid response and containment measures.
4. **Drug Discovery and Development:** Machine learning techniques are increasingly used in pharmaceutical research to accelerate drug discovery processes. By analyzing molecular interactions, genetic data, and clinical trial outcomes, algorithms can identify potential drug candidates and predict their efficacy and safety profiles.
5. **Health Resource Allocation:** Predictive models can forecast disease prevalence and healthcare utilization patterns, aiding in the allocation of resources such as hospital beds, medical staff, and medical supplies. This proactive approach helps healthcare systems better prepare for patient needs emergencies.

6. Risk Assessment and Prevention: Machine learning models can assess an individual's risk of developing certain diseases based on their demographics, lifestyle factors, and genetic predispositions. This information can guide preventive measures such as lifestyle modifications, screening programs, and targeted interventions to reduce disease incidence.

7.Improving Healthcare Efficiency: By optimizing clinical workflows, predicting patient outcomes, and automating routine tasks, machine learning contributes to improving the efficiency and quality of healthcare delivery. This allows healthcare providers to focus more on patient care and less on administrative burdens.

Overall, disease prediction using machine learning empowers healthcare providers, policymakers, and researchers to make data-driven decisions that enhance public health outcomes, improve patient care, and advance medical knowledge.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

The existing problem in disease prediction using machine learning include data quality issues, challenges with model interpretability, difficulties in generalizing across diverse populations, and integrating predictions into clinical workflows effectively.

- **Data Quality and Availability:** Obtaining large-scale, high-quality datasets that are representative of diverse populations and include relevant features (such as genetic, clinical, environmental, and lifestyle factors) can be difficult. Limited access to comprehensive data hampers the development of robust and accurate prediction models.
- **Data Privacy and Security:** Healthcare data is sensitive and subject to strict privacy regulations . Ensuring patient data privacy while allowing for data sharing and model training poses a significant challenge.
- **Imbalanced Datasets:** In healthcare, datasets often exhibit class imbalance, where instances of one class (e.g., diseased patients) outnumber those of another (e.g., healthy individuals). This imbalance can lead to biased models that prioritize the majority class, affecting predictive performance.
- **Interpretability of Models:** Machine learning models used for disease prediction, such as deep learning models, are often complex and opaque. Healthcare professionals require interpretable models to understand predictions and trust their clinical relevance.
- **Generalization Across Populations:** Models trained on data from one population or healthcare system may not generalize well to other populations or settings due to differences in demographics, healthcare practices, and environmental factors.

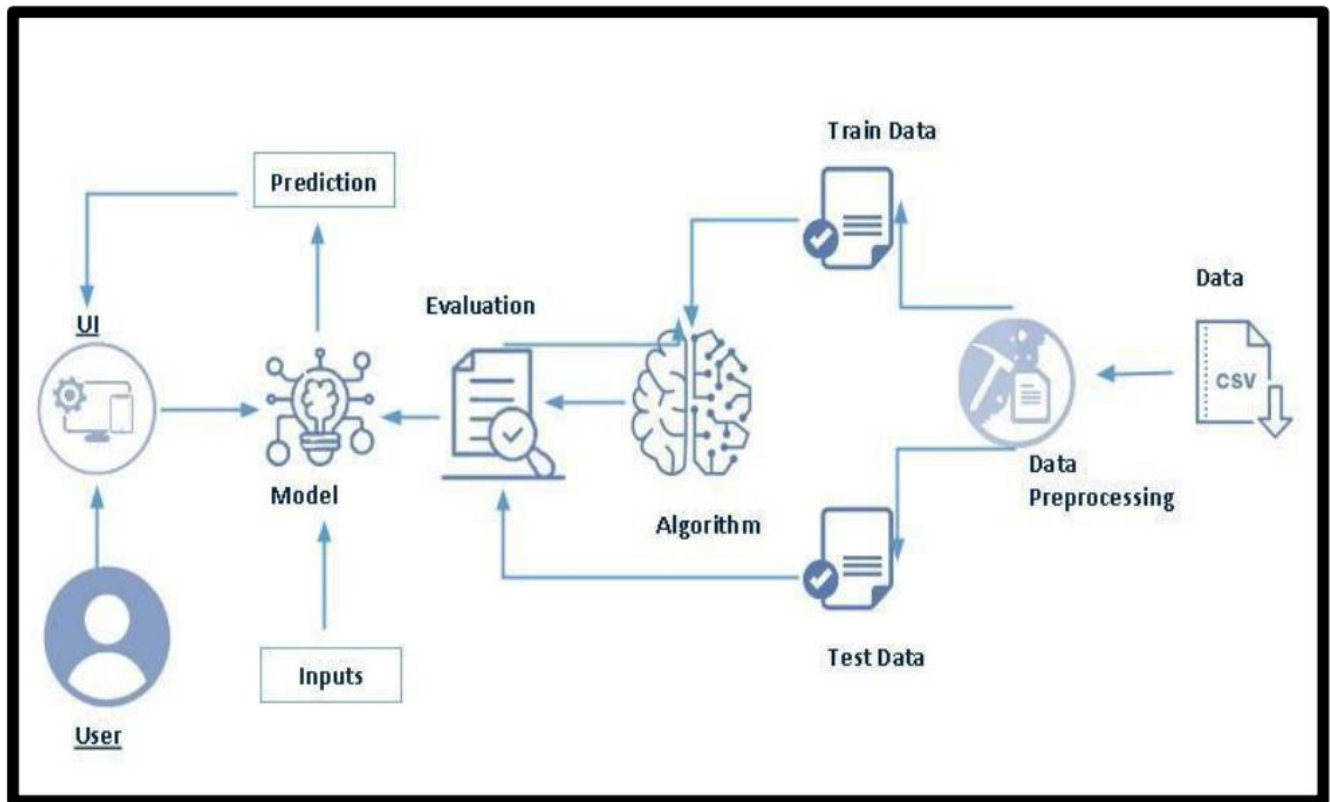
2.2 PROPOSED SOLUTION

- Our approach to disease prediction integrates advanced machine learning with comprehensive data analysis. We collect and preprocess diverse datasets including medical records, genetic information, lifestyle factors, and environmental data. Through meticulous feature engineering, we identify key predictors for diseases such as diabetes, cardiovascular issues, and cancers. Using state-of-the-art algorithms like neural networks and ensemble methods, we develop models that predict disease risks accurately. These models are rigorously validated and integrated into healthcare systems to provide real-time predictions and personalized health recommendations.

1. **Data Collection and Preprocessing:** Gather comprehensive datasets including medical records (EHR), demographic information, lifestyle factors, genetic data, and environmental exposures.
2. **Feature Engineering and Selection:** Engineer relevant features such as biomarkers, socio-economic indicators, medical history, and environmental factors. Select features based on their predictive power and relevance to specific diseases.
3. **Machine Learning Model Development:** Our Utilize advanced algorithms to train models for disease prediction based on historical data.
4. **Model Evaluation and Validation:** We Evaluate model performance using metrics like accuracy and AUC, validate on independent datasets, and deploy in healthcare systems for real-time prediction.
5. **User Interface and Accessibility:** Develop user-friendly interfaces for healthcare providers to interpret predictions and make informed decisions.
6. **Continuous Improvement:** Our Monitor model performance, update with new data, and incorporate feedback to enhance accuracy and relevance over time.

3.THEORITICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. SOFTWARE DESIGNING

The following is the Software required to complete this project:

- **Google Colab:** Google Colab as the development and execution environment. It offers a cloud-based Jupyter Notebook environment with Python libraries and hardware acceleration.
- **Dataset (CSV File):** The dataset in CSV format is containing historical health records, patient demographics, symptoms, medical history, and diagnostic outcomes. This dataset is essential for training and testing the predictive model.
- **Data Preprocessing Tools:** Python libraries like NumPy, Pandas, and Scikit-learn will be Handling missing data, Feature scaling and normalization, Encoding categorical variables, Data cleaning and outlier detection.
- **Feature Selection/Engineering:** Feature selection techniques to identify relevant features that influence disease prediction. Feature engineering might involve creating new features from existing ones to enhance model performance.
- **Model Selection and Training:** classification algorithms such as Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), or Gradient Boosting Machines (GBM). Train multiple models using cross-validation techniques to optimize hyperparamters and avoid overfitting.
- **Model Accuracy Evaluation:** After model training, Evaluate model performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC (Receiver Operating Characteristic - Area Under Curve). Utilize techniques such as cross-validation to ensure the model generalizes well to unseen data.
- **UI Based on Flask Environment:** Flask, Develop a web-based user interface using Flask to interact with the trained model.

4.EXPERIMENTAL INVESTIGATION

In this project, we have used Disease Prediction Using Machine Learning Dataset. This dataset is a csv file consisting of labelled data and having the following columns.

1. **itching**: Sensation that causes a desire to scratch.
2. **Skin rash**: Outbreak of red, bumpy, scaly, or itchy patches of skin.
3. **Nodal skin eruptions**: Swellings or lumps in the skin.
4. **Continuous sneezing**: Persistent sneezing often associated with allergies or infections.
5. **shivering**: Involuntary shaking or trembling often due to cold or fever.
6. **chills**: Sensation of coldness often accompanied by shivering, usually a sign of fever.
7. **Joint pain**: Discomfort, pain, or inflammation in the joints.
8. **Stomach pain**: Pain occurring in the abdomen.
9. **acidity**: Excessive acid production in the stomach causing discomfort or heartburn.
10. **Ulcers on tongue**: Open sores on the tongue.
11. **Muscle wasting**: Gradual decrease in muscle mass.
12. **vomiting**: Forceful expulsion of stomach contents through the mouth.
13. **Burning micturition**: Burning sensation during urination.
14. **Spotting urination**: Light bleeding between periods or spotting during urination.
15. **fatigue**: Extreme tiredness or exhaustion.
16. **Weight gain**: Increase in body weight.
17. **anxiety**: Feeling of worry, nervousness, or unease.
18. **Cold hands and feets**: Poor blood circulation leading to cold extremities.
19. **Mood swings**: Sudden and extreme changes in a person's emotions.
20. **Weight loss**: Decrease in body weight.
21. **restlessness**: Inability to stay still or relaxed.
22. **lethargy**: Lack of energy and enthusiasm.
23. **Patches in throat**: Discolored or inflamed patches in the throat.

- 24.**Irregular sugar level:** Fluctuations in blood sugar levels.
- 25.**cough:** Reflex action to clear the airways of mucus or irritants.
- 26.**High fever:** Elevated body temperature, typically above 100.4°F (38°C).
- 27.**Sunken eyes:** Eyes that appear hollow or recessed.
- 28.**breathlessness:** Difficulty or discomfort in breathing.
- 29.**sweating:** Production of sweat, often associated with fever or anxiety.
- 30.**dehydration:** Loss of body fluids, leading to dryness in the mouth and skin.
- 31.**indigestion:** Discomfort or pain in the stomach associated with difficulty in digesting food.
- 32.**headache:** Pain in the head.
- 33.**Yellowish skin:** Yellowing of the skin, often associated with jaundice.
- 34.**Dark urine:** Urine that is darker than usual, often a sign of dehydration or liver problems.
- 35.**nausea:** Feeling of sickness with an inclination to vomit.
- 36.**Loss of appetite:** Reduced desire to eat.
- 37.**Pain behind the eyes:** Discomfort or pain felt in the area behind the eyes.
- 38.**Back pain:** Pain located in the back.
- 39.**constipation:** Difficulty in passing stools.
- 40.**Abdominal pain:** Pain occurring in the stomach area.
- 41.**diarrhoea:** Frequent loose or liquid bowel movements.
- 42.**Mild fever:** Slightly elevated body temperature.
- 43.**Yellow urine:** Urine that is more yellow than usual, often a sign of dehydration.
- 44.**Yellowing of eyes:** Yellow discoloration of the eyes, often associated with jaundice.
- 45.**Acute liver failure:** Rapid loss of liver function.
- 46.**Fluid overload:** Excess fluid accumulation in the body.
- 47.**Swelling of stomach:** Abdominal bloating or swelling.
- 48.**Swelled lymph nodes:** Enlarged lymph nodes, often a sign of infection.
- 49.**malaise:** General feeling of discomfort or illness.
- 50.**Blurred and distorted vision:** Impaired or unclear vision.

- 51.**phlegm:** Mucus produced in the respiratory system.
- 52.**Throat irritation:** Soreness or discomfort in the throat.
- 53.**Redness of eyes:** Red or bloodshot eyes.
- 54.**Sinus pressure:** Pressure or pain in the sinus area.
- 55.**Runny nose:** Excess nasal drainage.
- 56.**congestion:** Blockage or congestion in the nasal passages.
- 57.**Chest pain:** Discomfort or pain in the chest area.
- 58.**Weakness in limbs:** Reduced strength in the arms or legs.
- 59.**Fast heart rate:** Elevated heart rate.
- 60.**Pain during bowel movements:** Discomfort or pain when passing stools.
- 61.**Pain in anal region:** Discomfort or pain around the anus.
- 62.**bloody_stool:** Presence of blood in stools.
- 63.**Irritation in anus:** Discomfort or itching in the anal area.
- 64.**Neck pain:** Discomfort or pain in the neck.
- 65.**dizziness:** Sensation of spinning or losing balance.
- 66.**cramps:** Painful muscle contractions.
- 67.**bruising:** Discoloration of the skin due to bleeding under the skin.
- 68.**obesity:** Excess body fat.
- 69.**Swollen legs:** Swelling in the legs, often due to fluid retention.
- 70.**Swollen blood vessels:** Enlarged or swollen veins.
- 71.**Puffy face and eyes:** Swelling of the face and around the eyes.
- 72.**Enlarged thyroid:** Swelling of the thyroid gland.
- 73.**Brittle nails:** Fragile nails that break easily.
- 74.**Swollen extremities:** Swelling in the arms or legs.
- 75.**Excessive hunger:** Increased appetite.
- 76.**Extra marital contacts:** Having sexual relationships outside of marriage.
- 77.**Drying and tingling lips:** Dry and tingling sensation on the lips.

- 78.**Slurred speech:** Difficulty in speaking clearly.
- 79.**Knee pain:** Discomfort or pain in the knee area.
- 80.**Hip joint pain:** Discomfort or pain in the hip joint.
- 81.**Muscle weakness:** Reduced muscle strength.
- 82.**Stiff neck:** Reduced range of motion or pain in the neck.
- 83.**Swelling joints:** Swelling in the joints.
- 84.**Movement stiffness:** Difficulty in moving due to stiffness.
85. **spinning movements:** Sensation of spinning or vertigo.
- 86.**Loss of balance:** Difficulty in maintaining balance.
- 87.**unsteadiness:** Lack of stability.
- 88.**Weakness of one body side:** Reduced strength on one side of the body.
- 89.**Loss of smell:** Inability to perceive odors.
- 90.**Bladder discomfort:** Pain or discomfort in the bladder area.
- 91.**Foul smell of urine:** Unpleasant odor of urine.
- 92.**Continuous feel of urine:** Constant sensation of needing to urinate.
- 93.**Passage of gases:** Passing of gas from the digestive system.
- 94.**Internal itching:** Itching sensation inside the body.
- 95.**Toxic look:** Appearance of severe illness, often associated with typhoid fever.
- 96.**depression:** Persistent feeling of sadness and loss of interest.
- 97.**irritability:** Easily annoyed or angered.
- 98.**Muscle pain:** Discomfort or pain in the muscles.
- 99.**Altered sensorium:** Change in awareness or mental function.
- 100.**Red spots over body:** Red spots or rash on the skin.
101. **Belly pain:** Discomfort or pain in the abdomen.
- 100.**abnormal menstruation:** Irregular menstrual cycles.
101. **dischromic patches:** Discoloration patches on the skin.
102. **watering from eyes:** Excessive tearing from the eyes.

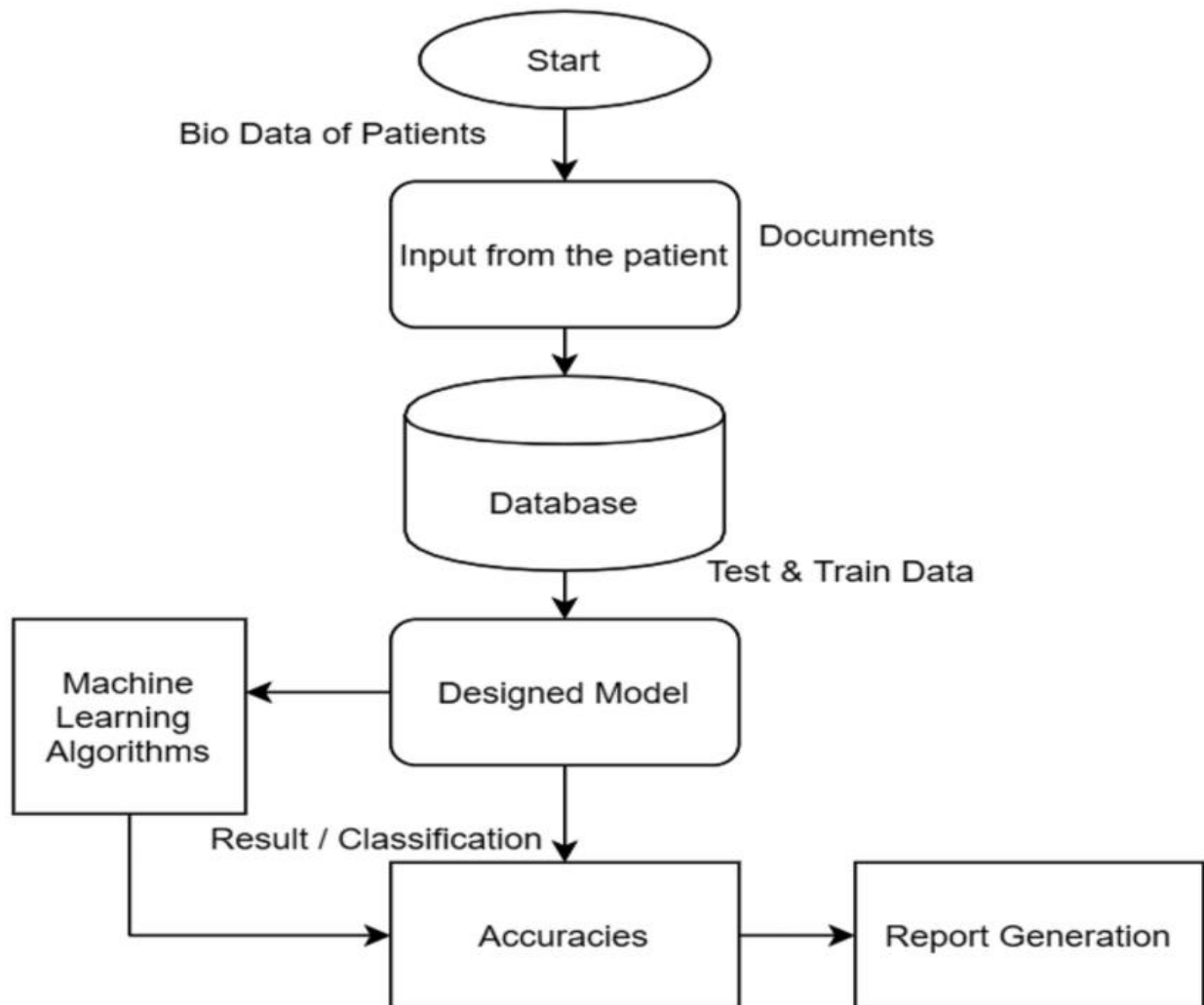
- 103. **increased appetite:** Higher than normal desire to eat.
- 104. **polyuria:** Excessive urination.
- 105. **family history:** Family history of certain diseases.
- 106. **mucoïd sputum:** Mucus in the sputum.
- 107. **rusty sputum:** Sputum that appears rust-colored.
- 108. **lack of concentration:** Difficulty in focusing or maintaining attention.
- 109. **visual _disturbances:** Problems with vision.
- 110. **receiving _blood _transfusion:** History of blood transfusion.
- 111. **receiving unsterile injections:** History of receiving injections that were not sterile.
- 112. **coma:** State of deep unconsciousness.
- 113. **stomach bleeding:** Bleeding in the stomach.
- 114. **distention of abdomen:** Swelling or bloating of the abdomen

For the dataset we selected, “we identified columns containing null values which could potentially compromise the integrity and reliability of our analyses. As a result, these columns were removed to ensure that our dataset maintains high quality and consistency throughout our analytical processes. This action aims to enhance the accuracy of our findings by focusing exclusively on complete and valid data points, thereby supporting more robust conclusions and informed decision-making”.

Feature drop means it drops the columns that we don’t want in our dataset.

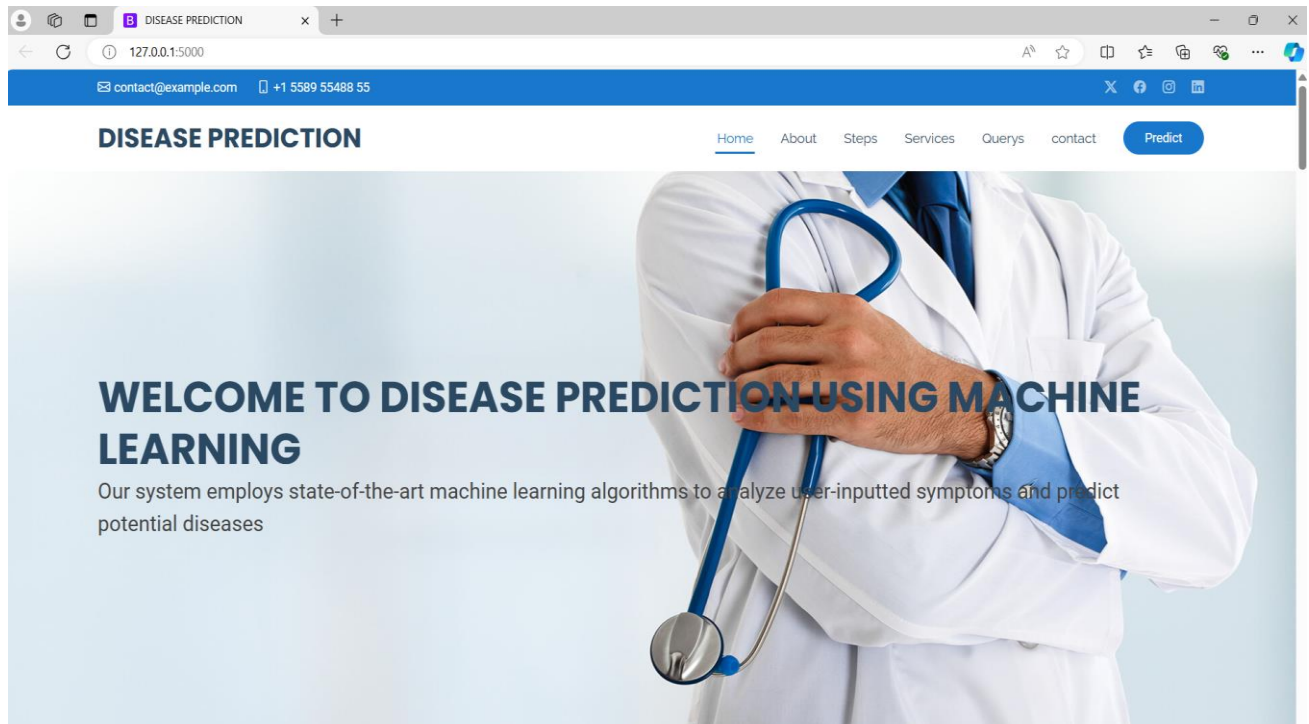
➤ Feature drop = ['Unnamed: 133','fluid_overload']

5.FLOWCHART



6.RESULT

HOME PAGE



PREDICTIONS

Home / Predict

Symptom-1 muscle_weakness	Symptom-2 knee_pain	Symptom-3 belly_pain
Symptom-4 muscle_pain	Symptom-5 joint_pain	Symptom-6 chest_pain
Symptom-7 abdominal_pain	Symptom-8 pain_behind_the_eyes	Symptom-9 neck_pain

Predict

Results

The probable diagnosis is: **Osteoarthritis**

[Try Another Prediction](#)

[Home](#) / Predict

Symptom-1

itching

Symptom-2

redness

Symptom-3

shivering

Symptom-4

Type your symptom here

Symptom-5

Type your symptom here

Symptom-6

Type your symptom here

Symptom-7

Type your symptom here

Symptom-8

Type your symptom here

Symptom-9

Type your symptom here

Predict

Results

The probable diagnosis is: **Fungal infection**

[Try Another Prediction](#)

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Early Detection:** Provides Machine learning models can analyze large amounts of data to detect patterns and anomalies that may indicate the early stages of a disease. This allows for timely intervention and treatment, potentially improving outcomes.
2. **Personalized Risk Assessment:** ML algorithms can assess individual risk factors based on personal health data, genetic information, lifestyle choices, and environmental factors.
3. **Improved Accuracy:** Increases ML models can process complex datasets and identify subtle correlations that may not be apparent through traditional statistical methods .
4. **Research Advancements:** By analyzing large-scale data, ML contributes to medical research by identifying new disease markers, treatment responses, and epidemiological patterns.
5. **Efficient Screening:** Automates analysis of large datasets, optimizing healthcare resources and reducing screening times.

DISADVANTAGES:

1. **Complexity and Interpretability:** Models can be complex, making it hard to interpret results
2. **Data Quality:** Relies heavily on high-quality, unbiased data for accurate predictions.
3. **Overfitting:** Models may fit training data too closely, leading to poor generalization.
4. **Causality:** Difficulty in distinguishing correlation from causation in predictions.
5. **Bias and Fairness:** Risk of inheriting biases from training data, impacting fairness.

8.APPLICATIONS

1. **Cancer Diagnosis and Prognosis:** Machine learning models are used to analyse medical imaging and pathology slides to detect cancerous lesions early and predict the progression of cancer.
2. **Cardiovascular Risk Assessment:** Machine learning algorithms analyze various risk factors such as blood pressure, cholesterol levels, and lifestyle habits to predict an individual's risk of developing cardiovascular diseases like heart attacks and strokes.
3. **Diabetes Management:** Machine learning helps in predicting the risk of developing diabetes based on factors such as family history, lifestyle choices, and biomarkers.
4. **Remote Patient Monitoring:** Real-time analysis of wearable device data to detect health issues.
5. **Healthcare Management:** Predicting patient outcomes and resource needs to optimize healthcare delivery.

9.CONCLUSION

- In conclusion, the implementation of machine learning for disease prediction marks a pivotal advancement in healthcare analytics. By harnessing the power of predictive models and data-driven insights, this project not only enhances diagnostic accuracy but also revolutionizes early intervention strategies. Through meticulous data preprocessing, feature engineering, model training, and validation, we have established a robust framework capable of forecasting disease risks with unprecedented precision. Moreover, the integration of user-friendly interfaces empowers healthcare professionals and patients alike to make informed decisions, leading to timely treatments and improved patient outcomes. As we continue to refine our methodologies and expand our datasets, the potential impact on public health is profound. In the evolving landscape of medical science and technology, ongoing research, innovation, and stakeholder engagement are pivotal. Continued refinement of machine learning models, coupled with interdisciplinary collaborations, promises to expand the scope and accuracy of disease prediction efforts.
- Ultimately, this project exemplifies a concerted effort to harness data-driven approaches for proactive disease management, thereby contributing to healthier communities and advancing the frontier of predictive medicine. As we continue to refine methodologies and embrace emerging technologies, the potential for transformative impacts on public health and healthcare delivery systems remains profound."
- This conclusion, emphasizes the transformative potential of machine learning in healthcare, highlighting its role in improving diagnostics, personalized medicine, and overall public health outcomes

10. FUTURE SCOPE

Future Scope of the Disease Prediction and Management System:

1. **Early Detection and Diagnosis:** Develop machine learning models to detect diseases at early stages by analyzing various biomarkers, genetic data, and health records.
2. **Personalized Medicine:** Use ML algorithms to personalize treatment plans based on individual patient data, including genetic profiles, lifestyle factors, and medical history.
3. **Predictive Analytics:** Improve disease prediction models to forecast the likelihood of developing specific diseases based on demographic data, environmental factors, and genetic predispositions.
4. **Population Health Management:** Apply machine learning to analyze large-scale population health data to identify trends, patterns, and potential outbreaks, aiding in proactive public health interventions.
5. **Continuous Learning and Improvement :** Establish frameworks for continuous learning and improvement of machine learning models using real-world patient data and feedback from healthcare professionals.

11.BIBLIOGRAPHY

Certainly! Here's a bibliography for disease prediction using machine learning techniques:

These references cover a range of topics from the application of machine learning in disease prediction using electronic health records to deep learning techniques for medical imaging and clinical decision support systems

[1] Ghassemi M, Naumann T, Schulam P, Beam AL. A review of challenges and opportunities in machine learning for health. *AMIA Jt Summits Transl Sci Proc.* 2020 Mar 4;2020:191-200. PMID: 32477652.

[2] Rajkomar A, Dean J, Kohane I. Machine learning in medicine. *N Engl J Med.* 2019 Apr 4;380(14):1347-58. doi: 10.1056/NEJMra1814259. PMID: 30943335.

[3] Obermeyer Z, Emanuel EJ. Predicting the future - big data, machine learning, and clinical medicine. *N Engl J Med.* 2016 Sep 29;375(13):1216-9. doi: 10.1056/NEJMp1606181. PMID: 27682033.

[4] Becker AS, Mueller M, Stoffel E, Marcon M, Ghafoor S, Boss A. Classification of breast cancer in ultrasound imaging using a generic deep learning analysis software: a pilot study. *Br J Radiol.* 2018 May;91(1085):20170576. doi: 10.1259/bjr.20170576. Epub 2018 Mar 23. PMID: 29565617; PMCID: PMC6104757.

[5]. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. *Brief Bioinform.* 2018 Nov 27;19(6):1236-46. doi: 10.1093/bib/bbx044. PMID: 28582469; PMCID: PMC6522261.

[6] Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, Liu PJ, Liu X, Marcus J, Sun M, Sundberg P, Yee H, Zhang K, Zhang Y, Flores G, Duggan GE, Irvine J, Le Q, Litsch K, Mossin A, Tansuwan J, Wang D, Wexler J, Wilson J, Ludwig D, Volchenboum SL, Chou K, Pearson M,

Madabushi S, Shah NH, Butte AJ, Howell MD, Cui C, Corrado GS, Dean J. Scalable and accurate deep learning with electronic health records. NPJ Digit Med. 2018 May 8;1:18. doi: 10.1038/s41746-018-0029-1. PMID: 31304301; PMCID: PMC6501990.

[7] Ravi D, Wong C, Deligianni F, Berthelot M, Andreu-Perez J, Lo B, Yang GZ. Deep learning for health informatics. IEEE J Biomed Health Inform. 2017 Jan;21(1):4-21. doi: 10.1109/JBHI.2016.2636665. PMID: 28113205.

[8]. Shickel B, Tighe PJ, Bihorac A, Rashidi P. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE J Biomed Health Inform. 2018 Jan;22(5):1589-604. doi: 10.1109/JBHI.2017.2767063. PMID: 28920993; PMCID: PMC5819517.

[9] Lipton ZC, Kale DC, Elkan C, Wetzel R. Learning to diagnose with LSTM recurrent neural networks. arXiv preprint arXiv:1511.03677. 2015 Nov 11.

[10] Chen JH, Asch SM. Machine learning and prediction in medicine - beyond the peak of inflated expectations. N Engl J Med. 2017 Jun 29;376(26):2507-9. doi: 10.1056/NEJMp1702071. PMID: 28657873.

12.APPENDIX

Model building :

- 1)Dataset
- 2)Google colab and VS code Application Building
 1. HTML file (Index file, Predict file)
 1. CSS file
 2. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>DISEASE PREDICTION</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="static/assets/img/favicon.png" rel="icon">
  <link href="static/assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Fonts -->
  <link href="https://fonts.googleapis.com" rel="preconnect">
  <link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Raleway:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="static/assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="static/assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">
```

```

<link href="static/assets/vendor/aos/aos.css" rel="stylesheet">
<link href="static/assets/vendor/fontawesome-free/css/all.min.css" rel="stylesheet">
<link href="static/assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
<link href="static/assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

<!-- Main CSS File -->
<link href="static/assets/css/main.css" rel="stylesheet">

</head>

<body class="index-page">

  <header id="header" class="header sticky-top">

    <div class="topbar d-flex align-items-center">
      <div class="container d-flex justify-content-center justify-content-md-between">
        <div class="contact-info d-flex align-items-center">
          <i class="bi bi-envelope d-flex align-items-center"><a
href="mailto:contact@example.com">contact@example.com</a></i>
          <i class="bi bi-phone d-flex align-items-center ms-4"><span>+1 5589 55488
55</span></i>
        </div>
        <div class="social-links d-none d-md-flex align-items-center">
          <a href="#" class="twitter"><i class="bi bi-twitter-x"></i></a>
          <a href="#" class="facebook"><i class="bi bi-facebook"></i></a>
          <a href="#" class="instagram"><i class="bi bi-instagram"></i></a>
          <a href="#" class="linkedin"><i class="bi bi-linkedin"></i></a>
        </div>
      </div>
    </div><!-- End Top Bar -->

    <div class="branding d-flex align-items-center">

      <div class="container position-relative d-flex align-items-center justify-
content-between">
        <a href="index.html" class="logo d-flex align-items-center me-auto">
          <!-- Uncomment the line below if you also wish to use an image logo -->
          <!--  -->
          <h1 class="sitename">DISEASE PREDICTION</h1>
        </a>

        <nav id="navmenu" class="navmenu">
          <ul>
            <li><a href="#hero" class="active">Home<br></a></li>
            <li><a href="#about">About</a></li>

```

```

        <li><a href="#stats">Steps</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#faq">Querys</a></li>
        <li><a href="#contact">contact</a></li>

    </ul>
    <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
</nav>

    <a class="cta-btn d-none d-sm-block" href="/details" class="btn-get-
started">Predict</a>

</div>

</div>

</header>

<main class="main">

    <!-- Hero Section -->
    <section id="hero" class="hero section light-background">

        <div class="container position-relative">

            <div class="welcome position-relative" data-aos="fade-down" data-aos-
delay="100">
                <h2>WELCOME TO DISEASE PREDICTION USING MACHINE LEARNING</h2>
                <p>Our system employs state-of-the-art machine learning algorithms to analyze
user-inputted symptoms and predict potential diseases</p>
            </div><!-- End Welcome -->

        </div>

    </section><!-- /Hero Section -->

    <!-- About Section -->
    <section id="about" class="about section">

        <div class="container">

            <div class="row gy-4 gx-5">

```

```

        <div class="col-lg-6 position-relative align-self-start" data-aos="fade-up"
data-aos-delay="200">
            
            <a href="https://www.youtube.com/watch?v=LXb3EKWsInQ" class="lightbox
pulsating-play-btn"></a>
        </div>

        <div class="col-lg-6 content" data-aos="fade-up" data-aos-delay="100">
            <h3>About Us</h3>
            <h2>About Disease Prediction Model</h2>
            <p>The disease prediction model utilizes advanced Machine Learning techniques to
accurately determine potential diseases based on input symptoms.</p>
            <p>Developed with a focus on healthcare innovation, this model aims to:</p>
        </div>
        <div class="row gy-4">
            <div class="col-lg-12">
                <div class="content">
                    <p class="fst-italic">
                        The model achieves a high accuracy rate of 97% in identifying the correct
disease from a range of possibilities, providing reliable guidance for healthcare
decisions.
                    </p>
                    <ul>
                        <li><i class="bi bi-check-circle-fill"></i> Utilizes extensive datasets
comprising diverse patient records and symptom profiles.</li>
                        <li><i class="bi bi-check-circle-fill"></i> Employs robust statistical and
mathematical algorithms to analyze symptoms and make predictions.</li>
                        <li><i class="bi bi-check-circle-fill"></i> Incorporates state-of-the-art
Machine Learning models, ensuring adaptive and efficient disease classification.</li>
                    </ul>
                    <p>
                        This model serves as a valuable tool for self-diagnosis, offering
preliminary insights into potential health issues and guiding users towards timely
medical consultations.
                    </p>
                </div>
            </div>
        </div>
    </div>

</section><!-- /About Section -->

```

```

<!-- Stats Section -->
<section id="stats" class="stats section light-background">

  <div class="container" data-aos="fade-up" data-aos-delay="100">
    <h3>STEPS WE FOLLOWED</h3>
    <div class="row gy-4">

      <div class="col-lg-3 col-md-6 d-flex flex-column align-items-center">
        <i class="fa-solid fa-user-doctor"></i>
        <div class="stats-item">

          <p>Collecting Dataset</p>
        </div>
      </div><!-- End Stats Item -->

      <div class="col-lg-3 col-md-6 d-flex flex-column align-items-center">
        <i class="fa-regular fa-hospital"></i>
        <div class="stats-item">

          <p>Preprocessing</p>
        </div>
      </div><!-- End Stats Item -->

      <div class="col-lg-3 col-md-6 d-flex flex-column align-items-center">
        <i class="fas fa-flask"></i>
        <div class="stats-item">

          <p>Model Building</p>
        </div>
      </div><!-- End Stats Item -->

      <div class="col-lg-3 col-md-6 d-flex flex-column align-items-center">
        <i class="fas fa-award"></i>
        <div class="stats-item">

          <p>Deployments</p>
        </div>
      </div><!-- End Stats Item -->

    </div>

  </div>

</section><!-- /Stats Section -->

```



```

<!-- Services Section -->
<section id="services" class="services section">

  <!-- Section Title -->
  <div class="container section-title" data-aos="fade-up">
    <h2>Services</h2>
    <p>Accurate predictions for disease outbreaks and risks</p>
  </div><!-- End Section Title -->

  <div class="container">

    <div class="row gy-4">

      <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="100">
        <div class="service-item position-relative">
          <div class="icon">
            <i class="fas fa-heartbeat"></i>
          </div>
          <a href="#" class="stretched-link">
            <h3>Risk Assessment</h3>
          </a>
          <p>Your service could help users assess their risk for developing certain diseases based on factors like family history, lifestyle habits, and basic symptom checks</p>
        </div>
      </div><!-- End Service Item -->

      <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="200">
        <div class="service-item position-relative">
          <div class="icon">
            <i class="fas fa-pills"></i>
          </div>
          <a href="#" class="stretched-link">
            <h3>Early Detection</h3>
          </a>
          <p>By analyzing user-reported symptoms, the service might suggest potential conditions and encourage users to seek professional medical advice</p>
        </div>
      </div><!-- End Service Item -->

      <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="300">
        <div class="service-item position-relative">
          <div class="icon">
            <i class="fas fa-hospital-user"></i>

```

```

    </div>
    <a href="#" class="stretched-link">
      <h3>Diagnostic Support</h3>
    </a>
    <p>The model's predictions could provide additional insights to doctors,
aiding in differential diagnosis and potentially leading to quicker and more accurate
diagnoses</p>
  </div>
</div><!-- End Service Item -->

```

```

<div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="400">
  <div class="service-item position-relative">
    <div class="icon">
      <i class="fas fa-dna"></i>
    </div>
    <a href="#" class="stretched-link">
      <h3>Personalized Treatment Plans</h3>
    </a>
    <p>By factoring in the model's predictions, doctors might personalize
treatment plans that better target the specific disease</p>
    <a href="#" class="stretched-link"></a>
  </div>
</div><!-- End Service Item -->

```

```

<div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="500">
  <div class="service-item position-relative">
    <div class="icon">
      <i class="fas fa-wheelchair"></i>
    </div>
    <a href="#" class="stretched-link">
      <h3>Resource Allocation </h3>
    </a>
    <p>Disease prediction models can help public health agencies allocate
resources more effectively to areas with higher predicted risks</p>
    <a href="#" class="stretched-link"></a>
  </div>
</div><!-- End Service Item -->

```

```

<div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="600">
  <div class="service-item position-relative">
    <div class="icon">
      <i class="fas fa-notes-medical"></i>
    </div>
    <a href="#" class="stretched-link">
      <h3>Focus on Prevention</h3>
    </a>
  </div>
</div>

```

```

        </a>
        <p>Promote the role of your service in preventative healthcare and early
detection, encouraging users to take a more proactive approach to their health.</p>
        <a href="#" class="stretched-link"></a>
    </div>
</div><!-- End Service Item -->

</div>

</div>

</section><!-- /Services Section -->

```

```

<!-- Faq Section -->
<section id="faq" class="faq section light-background">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
        <h2>Frequently Asked Questions</h2>

    </div><!-- End Section Title -->

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-lg-10" data-aos="fade-up" data-aos-delay="100">

                <div class="faq-container">

                    <div class="faq-item faq-active">
                        <h3>How accurate are these predictions</h3>
                        <div class="faq-content">
                            <p>The accuracy will depend on the specific disease, the quality of
the training data, and the chosen algorithm. Responsible services will disclose their
accuracy metrics.</p>
                        </div>
                        <i class="faq-toggle bi bi-chevron-right"></i>
                    </div><!-- End Faq item-->

```

```

<div class="faq-item">
  <h3>What kind of data do you collect?</h3>
  <div class="faq-content">
    <p>This will vary depending on the service, but it could include
demographic information, medical history, and symptom data.</p>
  </div>
  <i class="faq-toggle bi bi-chevron-right"></i>
</div><!-- End Faq item-->

<div class="faq-item">
  <h3>How is my data protected?</h3>
  <div class="faq-content">
    <p>The service should have robust security measures in place to
protect user data and comply with data privacy regulations</p>
  </div>
  <i class="faq-toggle bi bi-chevron-right"></i>
</div><!-- End Faq item-->

<div class="faq-item">
  <h3>What happens if I enter certain symptoms?</h3>
  <div class="faq-content">
    <p>The service should provide an explanation of the potential risks
and emphasize the need to see a doctor for confirmation.</p>
  </div>
  <i class="faq-toggle bi bi-chevron-right"></i>
</div><!-- End Faq item-->

<div class="faq-item">
  <h3>How does machine learning work for disease prediction?</h3>
  <div class="faq-content">
    <p>The model learns patterns from a large dataset of medical records
to identify relationships between symptoms and diseases</p>
  </div>
  <i class="faq-toggle bi bi-chevron-right"></i>
</div><!-- End Faq item-->

<div class="faq-item">
  <h3>What are the benefits of using machine learning for disease
prediction?</h3>
  <div class="faq-content">
    <p>Early detection of potential health problems can lead to better
treatment outcomes and preventative measures.</p>
  </div>
  <i class="faq-toggle bi bi-chevron-right"></i>
</div><!-- End Faq item-->

```

```

        </div>

</div><!-- End Faq Column-->

</div>

</div>

</section><!-- /Faq Section -->


<!-- Gallery Section -->
<section id="gallery" class="gallery section">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
        <h2>Gallery</h2>

    </div><!-- End Section Title -->

    <div class="container-fluid" data-aos="fade-up" data-aos-delay="100">

        <div class="row g-0">

            <div class="col-lg-3 col-md-4">
                <div class="gallery-item">
                    <a href="static/assets/img/gallery/gallery-1.jpg" class="glightbox" data-
gallery="images-gallery">
                        
                    </a>
                </div>
            </div><!-- End Gallery Item -->

            <div class="col-lg-3 col-md-4">
                <div class="gallery-item">
                    <a href="static/assets/img/gallery/gallery-2.jpg" class="glightbox" data-
gallery="images-gallery">
                        
                    </a>
                </div>
            </div><!-- End Gallery Item -->

```

```

        <div class="col-lg-3 col-md-4">
            <div class="gallery-item">
                <a href="static/assets/img/gallery/gallery-3.jpg" class="glightbox" data-
gallery="images-gallery">
                    
                        </a>
                    </div>
                </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
            <div class="gallery-item">
                <a href="static/assets/img/gallery/gallery-4.jpg" class="glightbox" data-
gallery="images-gallery">
                    
                        </a>
                    </div>
                </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
            <div class="gallery-item">
                <a href="static/assets/img/gallery/gallery-5.jpg" class="glightbox" data-
gallery="images-gallery">
                    
                        </a>
                    </div>
                </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
            <div class="gallery-item">
                <a href="static/assets/img/gallery/gallery-6.jpg" class="glightbox" data-
gallery="images-gallery">
                    
                        </a>
                    </div>
                </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
            <div class="gallery-item">
                <a href="static/assets/img/gallery/gallery-7.jpg" class="glightbox" data-
gallery="images-gallery">

```

```

        
        </a>
    </div>
</div><!-- End Gallery Item -->

    <div class="col-lg-3 col-md-4">
        <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-8.jpg" class="glightbox" data-
gallery="images-gallery">
                
                </a>
            </div>
        </div><!-- End Gallery Item -->

    </div>

</div>

</section><!-- /Gallery Section -->

<!-- Contact Section -->
<section id="contact" class="contact section">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
        <h2>Contact</h2>
        <p>Vaagdevi Engineering College is an engineering college in Bollikunta,
Warangal, Telangana, India.
        VHRX+2XR, Khammam - Warangal Hwy, Road, Bollikunta, Telangana 50600 Phone:
0870 286 518</p>
    </div><!-- End Section Title -->

    <div class="mb-5" data-aos="fade-up" data-aos-delay="200">

</div><!-- End Google Maps -->

    <div class="container" data-aos="fade-up" data-aos-delay="100">

        <div class="row gy-4">

            <div class="col-lg-4">
                <div class="info-item d-flex" data-aos="fade-up" data-aos-delay="300">
                    <i class="bi bi-geo-alt flex-shrink-0"></i>

```

```

        <div>
            <h3>Location</h3>
            <p>VHRX+2XR, Khammam - Warangal Hwy, Road, Bollikunta, Telangana
50600</p>
        </div>
    </div><!-- End Info Item -->

    <div class="info-item d-flex" data-aos="fade-up" data-aos-delay="400">
        <i class="bi bi-telephone flex-shrink-0"></i>
        <div>
            <h3>Call Us</h3>
            <p>+1 5589 55488 55</p>
        </div>
    </div><!-- End Info Item -->

    <div class="info-item d-flex" data-aos="fade-up" data-aos-delay="500">
        <i class="bi bi-envelope flex-shrink-0"></i>
        <div>
            <h3>Email Us</h3>
            <p>info@example.com</p>
        </div>
    </div><!-- End Info Item -->

</div>

<div class="col-lg-8">
    <form action="static/forms/contact.php" method="post" class="php-email-
form" data-aos="fade-up" data-aos-delay="200">
        <div class="row gy-4">

            <div class="col-md-6">
                <input type="text" name="name" class="form-control" placeholder="Your
Name" required="">
            </div>

            <div class="col-md-6 ">
                <input type="email" class="form-control" name="email"
placeholder="Your Email" required="">
            </div>

            <div class="col-md-12">
                <input type="text" class="form-control" name="subject"
placeholder="Subject" required="">
            </div>

```



```

        <div class="col-md-12">
            <textarea class="form-control" name="message" rows="6"
placeholder="Message" required=""></textarea>
        </div>

        <div class="col-md-12 text-center">
            <div class="loading">Loading</div>
            <div class="error-message"></div>
            <div class="sent-message">Your message has been sent. Thank
you!</div>

            <button type="submit">Send Message</button>
        </div>

    </div>
</form>
</div><!-- End Contact Form -->

</div>

</div>

</section><!-- /Contact Section -->

</main>

<footer id="footer" class="footer light-background">

    <div class="container footer-top">
        <div class="row gy-4">
            <div class="col-lg-4 col-md-6 footer-about">
                <a href="index.html" class="logo d-flex align-items-center">
                    <span class="sitename">Vaagdevi Engineering College</span>
                </a>
                <div class="footer-contact pt-3">
                    <p>VHRX+2XR, Khammam - Warangal Hwy, Road, Bollikunta, Telangana 50600</p>

                    <p class="mt-3"><strong>Phone:</strong> <span>+1 5589 55488 55</span></p>
                    <p><strong>Email:</strong> <span>info@example.com</span></p>
                </div>
                <div class="social-links d-flex mt-4">
                    <a href=""><i class="bi bi-twitter-x"></i></a>
                    <a href=""><i class="bi bi-facebook"></i></a>
                    <a href=""><i class="bi bi-instagram"></i></a>
                    <a href=""><i class="bi bi-linkedin"></i></a>
                </div>
            </div>

```

```

        </div>
    </div>

</footer>

<!-- Scroll Top -->
<a href="#" id="scroll-top" class="scroll-top d-flex align-items-center justify-
content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Preloader -->
<div id="preloader"></div>

<!-- Vendor JS Files -->
<script src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="static/assets/vendor/php-email-form/validate.js"></script>
<script src="static/assets/vendor/aos/aos.js"></script>
<script src="static/assets/vendor/glightbox/js/glightbox.min.js"></script>
<script src="static/assets/vendor/purecounter/purecounter_vanilla.js"></script>
<script src="static/assets/vendor/swiper/swiper-bundle.min.js"></script>

<!-- Main JS File -->
<script src="static/assets/js/main.js"></script>

</body>

</html>

```

PREDICT.HTML

```

<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Results</title>
    <style>
        body {
            font-family: 'Helvetica Neue', Arial, sans-serif;
            background-image: url('static/assets/img/result.png');
            background-size: cover;
            background-position: center;
            background-repeat: no-repeat;

```

```

    height: 100vh;
    margin: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    color: #444;
}

.container {
    background: rgba(255, 255, 255, 0.9);
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);
    text-align: center;
    max-width: 600px;
    width: 100%;
}

.container h2 {
    color: #007bff;
    font-size: 2rem;
    margin-bottom: 20px;
}

.container h3 {
    font-weight: bold;
    font-size: 1.5rem;
    margin: 20px 0;
}

.prediction-text {
    color: #000; /* Black color for prediction text */
}

.container hr {
    border: 1px solid #ddd;
    margin: 20px 0;
}

.home-link {
    position: absolute;
    top: 20px;
    left: 20px;
    background: #007bff;
    color: #fff;

```

```

padding: 10px 20px;
text-decoration: none;
border-radius: 5px;
font-weight: bold;
}

.home-link:hover {
background: #0056b3;
}

.back-link {
display: inline-block;
margin-top: 20px;
padding: 10px 20px;
background: #28a745;
color: #fff;
text-decoration: none;
border-radius: 5px;
font-weight: bold;
}

.back-link:hover {
background: #218838;
}

@media (max-width: 768px) {
.container {
width: 90%;
padding: 20px;
}

.container h2 {
font-size: 1.5rem;
}

.container h3 {
font-size: 1.25rem;
}
}
</style>
</head>

<body>
<a href="/" class="home-link">Home</a>
<div class="container">

```

```

        <h2>Results</h2>
        <hr>
        <h3>The probable diagnosis is: <span class="prediction-text">{{ prediction_text
    }}</span></h3>
        <hr>
        <p><a href="/details" class="back-link">Try Another Prediction</a></p>
    </div>
</body>

</html>

```

APP.PY

```

from flask import Flask, render_template, request

import numpy as np
import pickle

app = Flask(__name__)

# Load the model
model = pickle.load(open('model.pkl', 'rb'))

# Define routes
@app.route("/")
def home():
    return render_template('index.html')

@app.route('/details')
def details():
    return render_template('details.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Define the list of symptoms
    symptoms = ['itching', 'continuous sneezing', 'joint pain', 'vomiting',
        'spotting_ urination', 'fatigue', 'weight_ loss', 'lethargy',
        'high_ fever', 'sunken_ eyes', 'sweating', 'headache', 'dark_ urine',
        'nausa', 'loss of appetite', 'pain behind theeyes', 'abdominal pain',
        'diarrhoea', 'mild fever', 'yellowing of eyes', 'swelled lymph nodes',
        'malaise', 'phlegm', 'congestion', 'chest pain', 'fast heart rate',
        'irritation in anus', 'swollen legs', 'puffy face and eyes',
        'excessive hunger', 'muscle weakness', 'movement stiffness',
        'weakness of one body side', 'bladder discomfort', 'depression',
        'irritability', 'muscle pain', 'red spots over body',

```

```

'abnormal_menstruation', 'increased appetite', 'mucoid sputum',
'rusty sputum', 'lack of concentration', 'receiving blood transfusion',
'coma', 'history of alcohol consumption', 'blood in sputum',
'palpitations', 'inflammatory nails', 'yellow crust ooze']

if request.method == 'POST':
    input = [str(x) for x in request.form.values()]

    b = [0] * 50
    for x in range(0, 50):
        for y in input:
            if symptoms[x] == y:
                b[x] = 1
    b = np.array(b)
    b = b.reshape(1, 50)
    prediction = model.predict(b)
    prediction = prediction[0]
    return render_template('results.html', prediction_text="
{}".format(prediction))

if __name__ == "__main__":
    app.run()

```

CODE SNIPPETS

MODEL BUILDING

```
co disease_prediction.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at 11:16AM

+ Code + Text
✓ RAM
Disk
Gemini

IMPORTING LIBRARIES

[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

READ TRAINING DATA

[6] train_data=pd.read_csv('/content/Training.csv')
train_data
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurrying	skin_peeling	sil
0	1	1	1	0	0	0	0	0	0	0	0 ...	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0 ...	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0 ...	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0 ...	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0 ...	0	0	0	0

```
[ ] train_data.shape
(4920, 134)

train_data.tail()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurrying	skin_peeling	silver_like_dus
4915	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0
4916	0	1	0	0	0	0	0	0	0	0	0 ...	1	0	0
4917	0	0	0	0	0	0	0	0	0	0	0 ...	0	0	0
4918	0	1	0	0	0	0	1	0	0	0	0 ...	0	1	0
4919	0	1	0	0	0	0	0	0	0	0	0 ...	0	0	0

5 rows x 134 columns

```
[ ] train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 134 entries, itching to Unnamed: 133
dtypes: float64(1), int64(132), object(1)
memory usage: 5.0+ MB
```

HANDLING MISSING VALUES

```
[ ] train_data.isnull().sum()
```

```

itching      0
skin_rash    0
nodal_skin_eruptions  0
continuous_sneezing  0
shivering    0
...
blister      0
red_sore_around_nose  0
yellow_crust_ooze    0
prognosis      0
Unnamed: 133      4920
Length: 134, dtype: int64

```

```
[ ] train_data.isna().sum().sum()
```

```
4920
```

```
[ ] duplicate = train_data[train_data.duplicated()]
duplicate
```

```

      itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  shivering  chills  joint_pain  stomach_pain  acidity  ulcers_on_tongue  ...  scurring  skin_peeling  silver_like_dus
5          0          1          1          0          0          0          0          0          0          0          ...          0          0
6          1          0          1          0          0          0          0          0          0          0          ...          0          0
7          1          1          0          0          0          0          0          0          0          0          ...          0          0
8          1          1          1          0          0          0          0          0          0          0          ...          0          0
9          1          1          1          0          0          0          0          0          0          0          ...          0          0
...
4915         0          0          0          0          0          0          0          0          0          0          ...          0          0
4916         0          1          0          0          0          0          0          0          0          0          ...          1          0

```

REMOVING NULL COLUMNS IN TRAINING DATA

```
[ ] train_data['Unnamed: 133'].value_counts()
```

```
Series([], Name: count, dtype: int64)
```

```
[ ] train_data.drop("Unnamed: 133",axis = 1,inplace=True)
train_data.drop("fluid_overload",axis = 1,inplace=True)
```

```
[ ] train_data.shape
```

```
(4920, 132)
```

DISCRIPTIVE ANALYSIS

```
train_data.describe()
```

```

      itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  shivering  chills  joint_pain  stomach_pain  acidity  ulcers_on_tongue  ...  pus_filled_pimples  bl
count  4920.000000  4920.000000          4920.000000          4920.000000  4920.000000  4920.000000  4920.000000  4920.000000  4920.000000  4920.000000  ...          4920.000000  49
mean    0.137805    0.159756          0.021951          0.045122    0.021951    0.162195    0.139024    0.045122    0.045122    0.021951  ...          0.021951
std     0.344730    0.366417          0.146539          0.207593    0.146539    0.368667    0.346007    0.207593    0.207593    0.146539  ...          0.146539
min     0.000000    0.000000          0.000000          0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000  ...          0.000000
25%     0.000000    0.000000          0.000000          0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000  ...          0.000000
50%     0.000000    0.000000          0.000000          0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000  ...          0.000000
75%     0.000000    0.000000          0.000000          0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000  ...          0.000000
max      1.000000    1.000000          1.000000          1.000000    1.000000    1.000000    1.000000    1.000000    1.000000    1.000000  ...          1.000000

```

8 rows × 131 columns


```
READING TEST DATA

test_data=pd.read_csv('/content/testing.csv')
test_data
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scarring	skin_peeling	sil
0	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	...	0	0	0	0
2	0	0	0	0	0	0	0	1	1	1	...	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0
4	1	1	0	0	0	0	0	1	0	0	...	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0

```
[ ] test_data.shape

(42, 133)

[ ] test_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 43.8+ KB
```

```
HANDLING MISSING VALUES IN TEST DATA

[ ] test_data.isnull().sum()

itching      0
skin_rash    0
nodal_skin_eruptions  0
continuous_sneezing  0
shivering    0
..
inflammatory_nails  0
blister        0
red_sore_around_nose  0
yellow_crust_ooze  0
prognosis      0
Length: 133, dtype: int64

[11] test_data.drop("fluid_overload",axis = 1,inplace=True)
```

```
DISCRIPTIVE ANALYSIS

test_data.describe()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_pimples	blackheads
count	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	...	42.000000	42.000000
mean	0.166667	0.190476	0.023810	0.047619	0.023810	0.166667	0.142857	0.047619	0.047619	0.023810	...	0.023810	0.023810
std	0.377195	0.397437	0.154303	0.215540	0.154303	0.377195	0.354169	0.215540	0.215540	0.154303	...	0.154303	0.154303
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000

8 rows x 131 columns

DATA PREPROCESSING

```
[ ]
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
train_data['prognosis'] = label_encoder.fit_transform(train_data['prognosis'])
train_data['prognosis'].unique()
```

```
array([[15,  4, 16,  9, 14, 33,  1, 12, 17,  6, 23, 30,  7, 32, 28, 29,  8,
        11, 37, 40, 19, 20, 21, 22,  3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
        31,  5,  0,  2, 38, 35, 27]])
```

```
[ ] label_encoder = LabelEncoder()
test_data['prognosis'] = label_encoder.fit_transform(test_data['prognosis'])
test_data['prognosis'].unique()
```

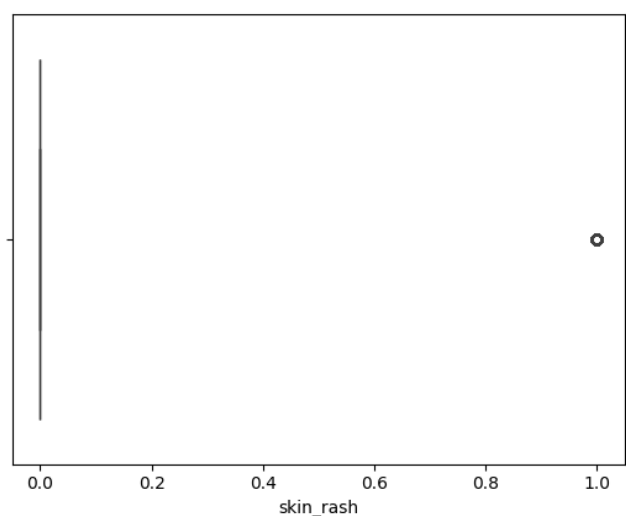
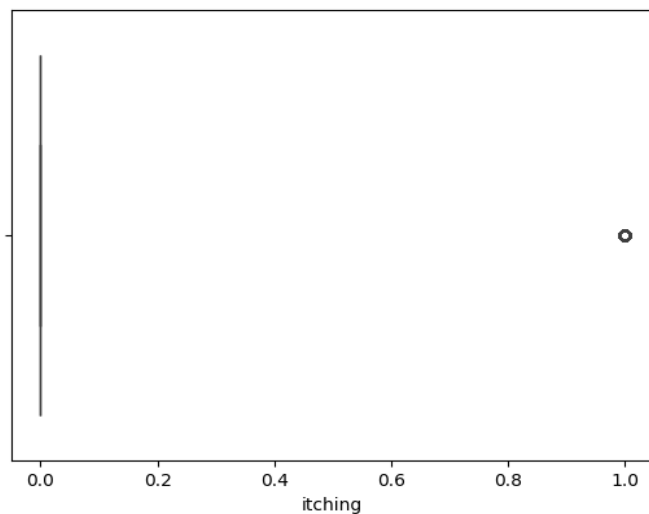
```
array([[15,  4, 16,  9, 14, 33,  1, 12, 17,  6, 23, 30,  7, 32, 28, 29,  8,
        11, 37, 40, 19, 20, 21, 22,  3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
        31,  5,  0,  2, 38, 35, 27]])
```

test_data['prognosis']

```
0 15
1  4
2 16
3  9
4 14
5 33
6  1
7 12
8 17
9  6
10 23
11 30
12  7
13 32
14 28
15 29
16  8
17 11
18 37
19 40
20 19
21 20
22 21
23 22
24  3
```

CHECKING OUTLAYERS BY VISUALIZING DATA

```
def func(col):
    sns.boxplot(x=col, data=train_data)
    plt.show()
for i in train_data.columns:
    func(i)
```



```

quant=train_data['itching'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)

```

```

0.75    0.0
0.25    0.0
Name: itching, dtype: float64
0.0
0.0
0.0
0.0
0.0

```

```

quant=train_data['silver_like_dusting'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)

```

```

0.75    0.0
0.25    0.0
Name: silver_like_dusting, dtype: float64
0.0
0.0
0.0
0.0
0.0

```

```

[ ] quant=train_data['continuous_sneezing'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)

```

```

0.75    0.0
0.25    0.0
Name: continuous_sneezing, dtype: float64
0.0
0.0
0.0
0.0
0.0

```

DATA VISUALIZATION

univariate analysis

```

plt.figure(figsize = (8,8))

a = train_data['itching'].value_counts()
plt.subplot(121)
plt.pie(x = a, data = train_data, labels= ['No', 'Yes'], autopct='%0f%%', colors = 'gr')
plt.title("Pie chart showing the distribution of Itching symptom into number of Yes/No ")

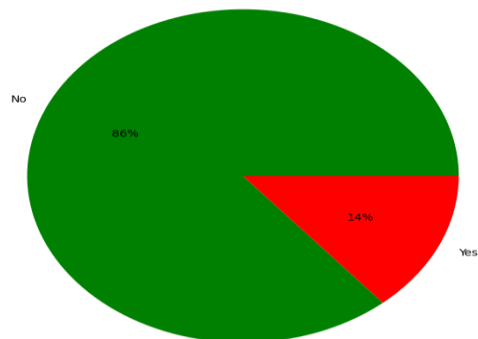
b=train_data['continuous_sneezing'].value_counts()
plt.subplot(122)

plt.pie(x = b, data = train_data, labels= ['No', 'Yes'], autopct='%0f%%', colors = 'gr')
plt.title('Pie Chart showing the distribution of Continuous Sneezing symptom into number of Yes/No')

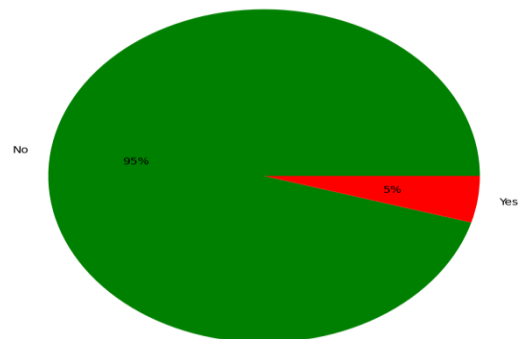
plt.subplots_adjust(left = 0.5, right = 2.4)

```

Pie chart showing the distribution of Itching symptom into number of Yes/No



Pie Chart showing the distribution of Continuous Sneezing symptom into number of Yes/No

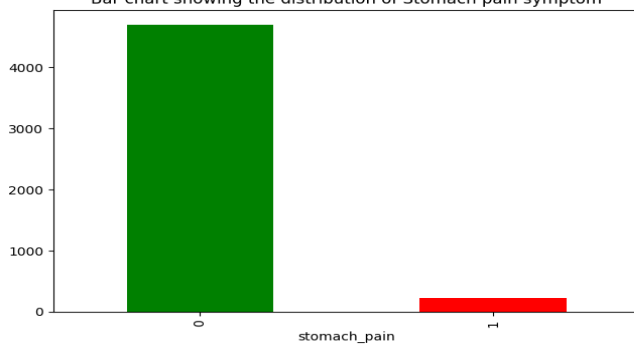


```
plt.subplot(1,2,1)
train_data['stomach_pain'].value_counts().plot(kind = 'bar',color = ['g','r'])
plt.title("Bar chart showing the distribution of Stomach pain symptom")

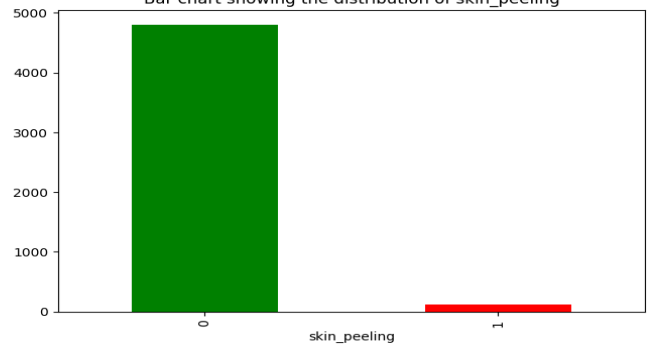
plt.subplot(1,2,2)
train_data['skin_peeling'].value_counts().plot(kind = 'bar',color = ['g','r'])
plt.title("Bar chart showing the distribution of skin_peeling")

plt.subplots_adjust(left = 0.5, right = 2.5)
```

Bar chart showing the distribution of Stomach pain symptom



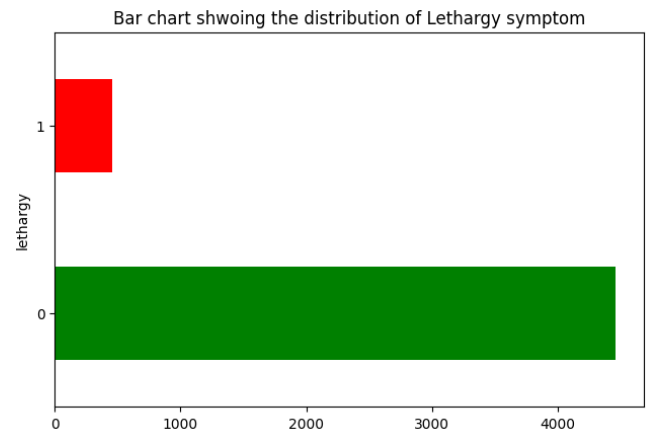
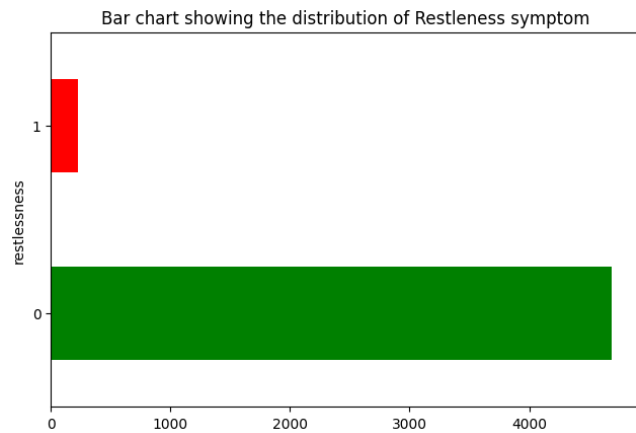
Bar chart showing the distribution of skin_peeling



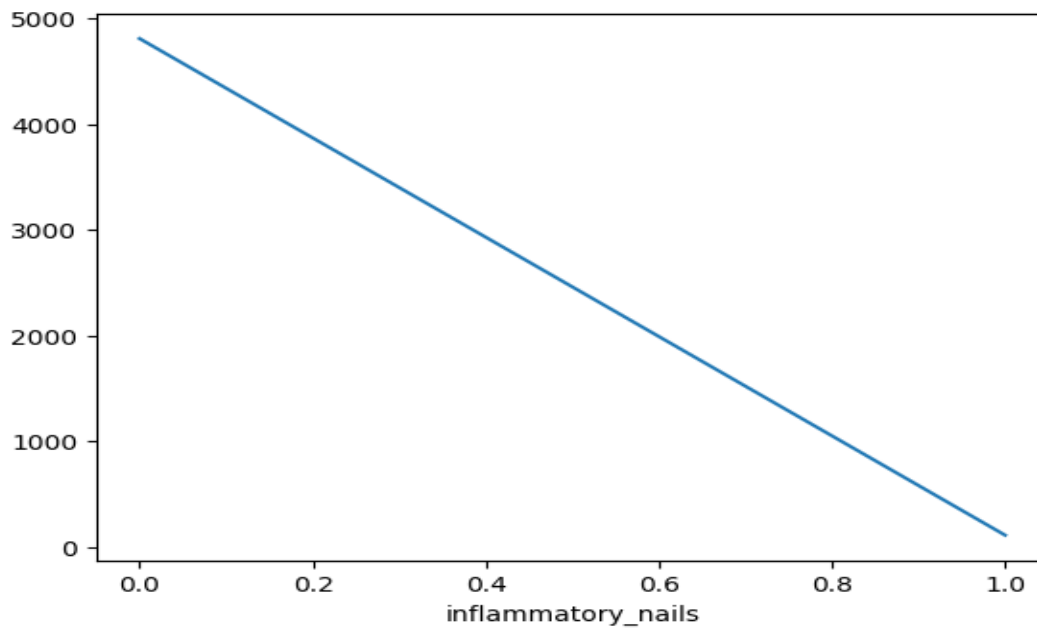
```
[ ] plt.subplot(1,2,1)
train_data['restlessness'].value_counts().plot(kind = 'barh', color = ['g','r'])
plt.title("Bar chart showing the distribution of Restlessness symptom")

plt.subplot(1,2,2)
train_data['lethargy'].value_counts().plot(kind = 'barh',color = ['g','r'])
plt.title("Bar chart showing the distribution of Lethargy symptom")

plt.subplots_adjust(left = 0.5,right = 2.5)
```

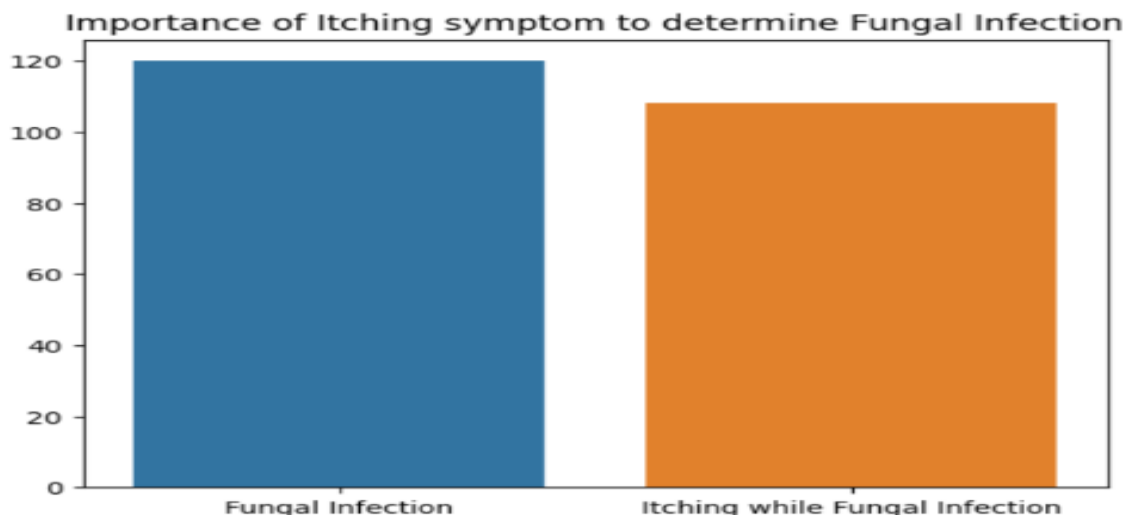


```
train_data['inflammatory_nails'].value_counts().sort_index().plot.line()
```

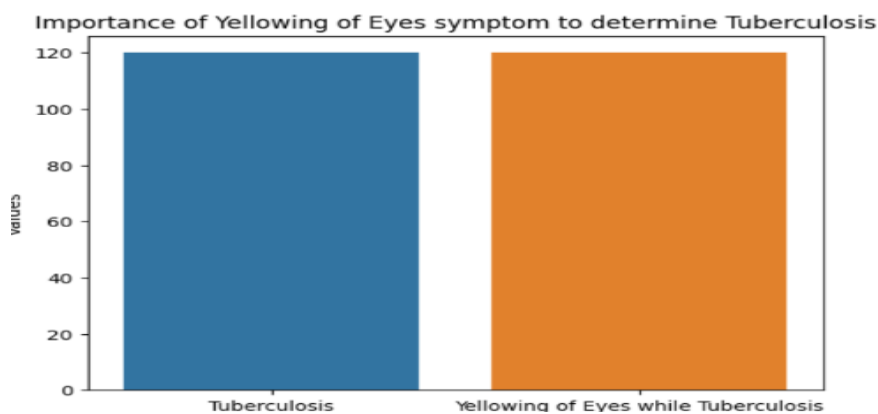


Bivariate Analysis

```
a = len(train_data[train_data['prognosis'] == 'Fungal infection'])
b = len(train_data[(train_data['itching'] == 1) & (train_data['prognosis'] == 'Fungal infection')])
fi = pd.DataFrame(data = [a,b], columns=['Values'],index = ['Fungal Infection','Itching while Fungal Infection'])
sns.barplot(data=fi, x=fi.index, y=fi['Values'], color='skyblue')
sns.barplot(data = fi , x = fi.index, y= fi['Values'])
plt.title('Importance of Itching symptom to determine Fungal Infection')
```



```
a = len(train_data[train_data['prognosis'] == 'Tuberculosis'])
b = len(train_data[(train_data['yellowing_of_eyes'] == 1) & (train_data['prognosis'] == 'Tuberculosis')])
fi = pd.DataFrame(data = [a,b], columns=['Values'],index = ['Tuberculosis','yellowing of eyes while scurring'])
sns.barplot(data=fi, x=fi.index, y=fi['Values'], color='green') # Changed color to 'green'
sns.barplot(data = fi , x = fi.index, y= fi['Values'])
plt.title('Importance of yellowing of eyes symptom to determine Tuberculosis')
```



multivariate analysis

```
corr=train_data.corr()
corr.style.background_gradient('coolwarm')
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893	-0.175905	-0.160650	0.202850	-0.086906	-0.059893	-0.059893
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	-0.029324	0.171134	0.161784	-0.094786	-0.065324	-0.065324
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	-0.022444	-0.022444
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608981	0.446238	-0.087351	-0.047254	-0.047254	-0.032566	-0.032566
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000	0.295332	-0.060200	-0.032566	-0.032566	-0.022444	-0.022444
chills	-0.175905	-0.029324	-0.065917	0.446238	0.295332	1.000000	-0.004688	-0.095646	-0.095646	-0.065917	-0.065917
joint_pain	-0.160650	0.171134	-0.060200	-0.087351	-0.060200	-0.004688	1.000000	-0.087351	-0.087351	-0.060200	-0.060200
stomach_pain	0.202850	0.161784	-0.032566	-0.047254	-0.032566	-0.095646	-0.087351	1.000000	0.433917	0.649078	-0.032566
acidity	-0.086906	-0.094786	-0.032566	-0.047254	-0.032566	-0.095646	-0.087351	0.433917	1.000000	0.608981	-0.032566
ulcers_on_tongue	-0.059893	-0.065324	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	0.649078	0.608981	1.000000	-0.022444
muscle_wasting	-0.059893	-0.065324	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	-0.022444	1.000000
vomiting	-0.057763	-0.225046	-0.119543	-0.173459	-0.119543	0.144263	0.199921	0.031406	0.019355	0.153603	-0.119543
burning_micturition	0.207896	0.166507	-0.032103	-0.046581	-0.032103	-0.094285	-0.086108	0.412239	-0.046581	-0.032103	-0.032103
spotting_urination	0.350585	0.298143	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	0.608981	-0.032566	-0.022444	-0.022444
fatigue	0.069744	-0.105248	-0.120465	0.041755	-0.120465	0.269437	0.066652	-0.174797	-0.174797	-0.120465	-0.120465
weight_gain	-0.061573	-0.067156	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	-0.023073
anxiety	-0.061573	-0.067156	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	-0.023073

DATA PREPROCESSING

```
train_data.drop(['weight_gain', 'cold_hands_and_feets', 'anxiety', 'irregular_sugar_level',
'yellow_urine', 'acute_liver_failure', 'swelling_of_stomach', 'drying_and_tingling_lips', 'continuous_feel_of_urine',
'internal_itching', 'polyuria', 'mood_swings', 'receiving_unsterile_injections', 'stomach_bleeding', 'prominent_veins_on_calf', 'loss_of_smell', 'throat_irritation',
'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'pain_during_bowel_movements', 'pain_in_anal_region', 'cramps', 'bruising', 'enlarged_thyroid', 'brittle_nails',
'swollen_extremeties', 'slurred_speech', 'distention_of_abdomen', 'fluid_overload.1', 'skin_peeling', 'silver_like_dusting', 'small_dents_in_nails', 'blister',
'red_sore_around_nose', 'bloody_stool', 'swollen_blood_vessels', 'hip_joint_pain',
'painful_walking', 'spinning_movements', 'altered_sensorium', 'toxic_look_typhos'], axis =1, inplace = True)
```

```

def data_preprocessing(data):
    data.drop(['weight_gain', 'cold_hands_and_feets', 'anxiety', 'irregular_sugar_level',
              'yellow_urine', 'acute_liver_failure', 'swelling_of_stomach', 'drying_and_tingling_lips', 'continuous_feel_of_urine',
              'internal_itching', 'polyuria', 'mood_swings', 'receiving_unsterile_injections', 'stomach_bleeding', 'prominent_veins_on_calf', 'loss_of_smell', 'throat_irritation',
              'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'pain_during_bowel_movements', 'pain_in_anal_region', 'cramps', 'bruising', 'enlarged_thyroid', 'brittle_nails',
              'swollen_extremeties', 'slurred_speech', 'distention_of_abdomen', 'fluid_overload.1', 'skin_peeling', 'silver_like_dusting', 'small_dents_in_nails', 'blister',
              'red_sore_around_nose', 'bloody_stool', 'swollen_blood_vessels', 'hip_joint_pain',
              'painful_walking', 'spinning_movements', 'altered_sensorium', 'toxic_look_(typhos)'],
              axis=1, inplace=True)

    return data

[ ] train_data.shape

(4920, 90)

[ ] test_data=data_preprocessing(test_data)

```

SPLITTING DATA INTO TRAINING AND VALIDATION AND TESTING DATA

```

[ ] x=train_data.drop(['prognosis'],axis=1)
    y=train_data['prognosis']

[ ] x_test=test_data.drop(['prognosis'],axis=1)
    y_test=test_data['prognosis']

from sklearn.model_selection import train_test_split
xtrain, xval, ytrain, yval = train_test_split(x, y, test_size=0.2) # Correct the order of returned variables

[ ] y.shape

(4920,)

[ ] y_test.shape

(42,)

```

TRAINING THE MODEL WITH MULTIPLE ALGORITHMS

```

[ ] from sklearn.metrics import accuracy_score
    from sklearn.preprocessing import LabelBinarizer

    def model_evaluation(data):
        y_pred = data.predict(xval)
        yt_pred = data.predict(xtrain)
        y_pred1 = data.predict(x_test)
        lb = LabelBinarizer()
        yt_pred_binarized = lb.fit_transform(yt_pred)
        y_pred_binarized = lb.transform(y_pred)
        y_pred1_binarized = lb.transform(y_pred1)
        print('The Training Accuracy of the algorithm is ', accuracy_score(lb.transform(ytrain), yt_pred_binarized))
        print('The Validation Accuracy of the algorithm is ', accuracy_score(lb.transform(yval), y_pred_binarized))
        print('The Testing Accuracy of the algorithm is ', accuracy_score(lb.transform(y_test), y_pred1_binarized))
        return [(accuracy_score(lb.transform(ytrain), yt_pred_binarized)), (accuracy_score(lb.transform(yval), y_pred_binarized)), (accuracy_score(lb.transform(y_test), y_pred1_binarized))]

[ ] from sklearn.neighbors import KNeighborsClassifier
    KNN = KNeighborsClassifier()
    # train the data with K-Nearest Neighbors Model
    KNN.fit(xtrain, ytrain)

KNeighborsClassifier
KNeighborsClassifier()

[ ] KNN_result=model_evaluation(KNN)

The Training Accuracy of the algorithm is 1.0
The Validation Accuracy of the algorithm is 1.0
The Testing Accuracy of the algorithm is 1.0

```



```
[ ] from sklearn.svm import SVC
svm=SVC(C=1)
svm.fit(xtrain,ytrain)
```

▼ SVC
SVC(C=1)

```
[ ] svm_result=model_evaluation(svm)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  1.0
```

```
from sklearn import tree
dt=tree.DecisionTreeClassifier(max_features=10)
dt.fit(xtrain,ytrain)
```

▼ DecisionTreeClassifier
DecisionTreeClassifier(max_features=10)

```
[ ] dt_result=model_evaluation(dt)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  0.9761904761904762
```

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(max_depth=13)
rf.fit(xtrain,ytrain)
```

▼ RandomForestClassifier
RandomForestClassifier(max_depth=13)

```
[ ] rf_result=model_evaluation(rf)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  0.9761904761904762
```

TESTING MODEL WITH MULTIPLE EVALUATION METRICS

```
result=pd.DataFrame(data=[knn_result,svm_result,dt_result,rf_result],
                    columns=['training accuracy','validation accuracy','testing accuracy'],
                    index=['k nearest neighbors classifier','support vector machine','decision trees classifier','random forest classifier'])
```

	training accuracy	validation accuracy	testing accuracy
k nearest neighbors classifier	1.0	1.0	1.00000
support vector machine	1.0	1.0	1.00000
decision trees classifier	1.0	1.0	0.97619
random forest classifier	1.0	1.0	0.97619

FEATURE IMPORTANCE

```
[ ] # Train a Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(x, y)

# Get feature importances
feature_importances = pd.DataFrame(rf.feature_importances_, index=x.columns, columns=['importance'])

# Sort feature importances in descending order
feature_importances = feature_importances.sort_values('importance', ascending=False)
```

feature_importances

	importance
muscle_pain	0.023568
high_fever	0.019704
joint_pain	0.018862
nausea	0.018408
yellowing_of_eyes	0.018377
...	...
blurred_and_distorted_vision	0.005797
puffy_face_and_eyes	0.005721
restlessness	0.005330
blood_in_sputum	0.005084
foul_smell_of_urine	0.003945

89 rows x 1 columns

```
[ ] thresholds = [0.020, 0.015, 0.010, 0.008]
rfc_result = []
num_features=[]
rfc_accuracies=[]
knn_accuracies=[]
```

```
[ ] for threshold in thresholds:
    to_drop = []
    for i, (feature, importance) in enumerate(feature_importances.iterrows()): # Iterate over rows instead of just column names
        if importance['importance'] < threshold: # Access the numerical importance value
            to_drop.append(x.columns[i])
```

```
[ ] X_new = x.drop(columns=to_drop)
```

```
[ ] num_features.append(X_new.shape[1])
```

```
[ ] len(to_drop)
```

```
21
```

```
[ ] X_new.shape
```

```
(4920, 68)
```

```
[ ] X_new.head()
```

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

5 rows x 68 columns

```
[ ] # Split the new data into training, validation, and testing sets
X1_train, X1_val, y1_train, y1_val = train_test_split(X_new, y, test_size=0.3, random_state=42)
x1_test=x_test.drop(to_drop,axis=1)
```

BUILDING MODEL WITH APPROPRIATE FEATURES

```
# Train a Random Forest Classifier and calculate accuracy
rfc = RandomForestClassifier(random_state=42)
rfc.fit(X1_train, y1_train)
y_pred_rfc = rfc.predict(X1_val)
```

```
[ ] y_pred = rfc.predict(X1_val)
    yt_pred = rfc.predict(X1_train)
    y_pred1 = rfc.predict(x1_test)
    print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
    print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
    print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[94] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_rfc)
     print(cm)
```

```
[[32  0  0 ...  0  0  0]
 [ 0 39  0 ...  0  0  0]
 [ 0  0 41 ...  0  0  0]
 ...
 [ 0  0  0 ... 36  0  0]
 [ 0  0  0 ...  0 37  0]
 [ 0  0  0 ...  0  0 39]]
```

```
[89] from sklearn.svm import SVC
     svm1=SVC(C=1)
     svm1.fit(X1_train,y1_train)
     y_pred_svc = svm1.predict(X1_val)
     y_pred = svm1.predict(X1_val)
     yt_pred = svm1.predict(X1_train)
     y_pred1 = svm1.predict(x1_test)
     print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
     print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
     print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[95] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_svc)
     print(cm)
```

```
[[32  0  0 ...  0  0  0]
 [ 0 39  0 ...  0  0  0]
 [ 0  0 41 ...  0  0  0]
 ...
 [ 0  0  0 ... 36  0  0]
 [ 0  0  0 ...  0 37  0]
 [ 0  0  0 ...  0  0 39]]
```

```
[ ] knn=KNeighborsClassifier()
    knn.fit(X1_train, y1_train)
    y_pred_knn = knn.predict(X1_val)

[ ] y_pred = rfc.predict(X1_val)
    yt_pred = rfc.predict(X1_train)
    y_pred1 = rfc.predict(X1_test)
    print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
    print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
    print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[96] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_knn)
     print(cm)
```

```
[[ 32  0  0 ...  0  0  0]
 [  0 39  0 ...  0  0  0]
 [  0  0 41 ...  0  0  0]
 ...
 [  0  0  0 ... 36  0  0]
 [  0  0  0 ...  0 37  0]
 [  0  0  0 ...  0  0 39]]
```

```
[ ] val_results = pd.DataFrame({
    'prognosis': y1_val,
    'rf_predicted': y_pred_rfc,
    'knn_predicted': y_pred_knn})
print(val_results)
```

```
prognosis  rf_predicted  knn_predicted
373        2           2             2
4916       2           2             2
1550       24          24            24
3081       1           1             1
3857       9           9             9
...        ...         ...          ...
3649       15          15            15
1694       33          33            33
4767       30          30            30
3721       26          26            26
2222       11          11            11
```

```
[1476 rows x 3 columns]
```

TESTING MODEL

```
test_data.join(pd.DataFrame(y_pred1,columns=["predicted"]))[["prognosis", "predicted"]]
```

	prognosis	predicted
0	15	15
1	4	4
2	16	16
3	9	9
4	14	14
5	33	33
6	1	1
7	12	12
8	17	17
9	6	6
10	23	23
11	30	30
12	7	7
13	32	32

```
✓ [79] import pickle
```

```
✓ 0s [80] pickle.dump(rfc,open('model.pkl','wb'))
```

