# DISEASE PREDICTION MODEL

A PROJECT REPORT
for
Mini Project-I (K24MCA18P)
Session (2024-25)

Submitted by

VIDUSHI AGRAWAL
(202410116100243)
VAISHNAVI YADAV
(202410116100234)
VAIBHAV SINGH KALURA
(202410116100233)

Submitted in partial fulfilment of the
Requirements for the Degree of

## MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(DECEMBER- 2024)

# CERTIFICATE

Certified that **Vidushi Agrawal (2426MCA316), Vaishnavi Yadav (2426MCA196)**, **Vaibhav Singh Kalura (2426MCA2325)** has/ have carried out the project work having "**Disease Prediction Model**" (**Mini Project-I, K24MCA18P**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

| | |
|---|---|
| **Ms. Divya Singhal** | **Dr. Arun Kr. Tripathi** |
| **Assistant Professor** | **Dean** |
| **Department of Computer Applications** | **Department of Computer Applications** |
| **KIET Group of Institutions, Ghaziabad** | **KIET Group of Institutions, Ghaziabad** |

## Disease Prediction Model

**Vidushi Agrawal**
**Vaishnavi Yadav**
**Vaibhav Singh Kalura**

## ABSTRACT

It is a system which provides the user the information and tricks to take care of the health system of the user and it provides how to search out the disease using this prediction. Now a day’s health industry plays major role in curing the diseases of the patients so this is often also some quite help for the health industry to inform the user and also it\'s useful for the user just in case he/she doesn’t want to travel to the hospital or the other clinics, so just by entering the symptoms and every one other useful information the user can get to grasp the disease he/she is affected by and also the health industry may also get enjoy this method by just asking the symptoms from the stoner and entering within the system and in only many seconds they'll tell the precise and over to some extent the accurate conditions. This Disease Prediction Using Machine Learning is totally through with the assistance of Machine Learning and Python programming language and also using the dataset that\'s available previously by the hospitals using that we are going to predict the diseases.

**Keywords:** Disease Diagnosis, Medical Dataset, Health Industry, Symptom Analysis

# ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Vidushi Agrawal**

**Vaishnavi Yadav**

**Vaibhav Singh Kalura**

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 OVERVIEW

Our project, the "Disease Prediction System Using Machine Learning," addresses a critical issue in healthcare – the timely and accurate diagnosis of diseases. The traditional healthcare approach often focuses on one disease at a time, leading to fragmented analyses and delayed interventions. This system revolutionizes disease prediction by utilizing machine learning algorithms, specifically the Random-Forest-Classifier, to predict a range of diseases based on a set of symptoms.

The significance of this project lies in its potential to impact people's lives positively. By enabling a swift and accurate prediction of diseases, individuals can receive early warnings and timely medical attention. This not only contributes to more effective treatments but also aids in preventing the escalation of health conditions, ultimately reducing mortality rates.

The user-friendly interface, built with Tkinter, facilitates easy interaction, making it accessible to a wide range of users. Individuals can input their symptoms, and the system provides instant predictions, empowering them with valuable health insights. This project serves as a valuable tool for both individuals monitoring their health and healthcare professionals seeking quick and accurate preliminary assessments.

In essence, the "Disease Prediction System Using Machine Learning" leverages advanced technology to address a pressing healthcare challenge, offering a practical solution that can positively impact public health outcomes.

## 1.2 PROJECT

The "Disease Prediction using Machine Learning" project emerges as a pioneering endeavour, seamlessly intertwining advancements in machine learning with the complexities of healthcare diagnostics. In a realm where the demand for rapid and accurate disease identification is ever-increasing, this project takes a quantum leap towards a future where technology becomes an indispensable ally in the pursuit of better health outcomes. At its core, the project seeks to harness the predictive prowess of machine learning, specifically employing the Random Forest Classifier, to revolutionize the landscape of disease diagnosis. This fragmentation can lead to delays in treatment and adverse health outcomes. By employing machine learning (ML) algorithms, specifically the Random Forest Classifier, this project revolutionizes healthcare diagnostics.

The application allows users to input symptoms and receive a prediction of potential diseases, fostering early detection and intervention. By leveraging advanced data processing techniques and a user-friendly interface developed with Tkinter, the system is designed to empower users, including healthcare professionals and the general public, to make informed decisions.

**Significance and Impact**

Early disease detection can be life-saving, as it often enables more effective treatment plans, reduces healthcare costs, and prevents complications. The system democratizes healthcare by making advanced diagnostic tools accessible to a wider audience, including those in remote or underserved areas.

By adopting a proactive rather than reactive approach, this project aligns with the global healthcare paradigm shift toward preventive care. As machine learning continues to evolve, this system lays the groundwork for integrating more sophisticated models, potentially accommodating rare diseases and complex medical conditions.

## 1.3 AIM

The aim of this study is to test the proposed hypothesis that supervised ML algorithms can improve health care by the accurate and early detection of diseases. In this study, we investigate studies that utilize more than one supervised ML model for each disease recognition problem. This approach renders more comprehensiveness and precision because the evaluation of the performance of a singlealgorithm over various study settings induces bias which generates imprecise results. The analysis ofML models will be conducted on few diseases located at heart, kidney, breast, and brain.

The overarching aim of this project is to explore the hypothesis that supervised ML algorithms can significantly enhance healthcare outcomes by providing early and accurate disease detection. Specifically, it evaluates the performance of supervised ML models in diagnosing diseases related to the heart, kidney, breast, and brain.

This aim is ambitious yet achievable, reflecting a commitment to advancing healthcare through technology. It also underscores the necessity of using multiple supervised learning models to counteract biases inherent in single-algorithm systems, ensuring comprehensive and reliable outcomes.

**Broader Objectives of the Aim**

1. To evaluate the role of ML in reducing diagnostic errors.
2. To facilitate a comparative study of algorithms like Random Forest, KNN, SVM, and CNN for diverse diseases.
3. To ensure scalability and adaptability, allowing future incorporation of additional diseases and symptoms.

## 1.4 BACKGROUND

Traditional healthcare diagnostics often grapple with challenges related to time consumption, subjectivity, and the potential for human biases. In this era of technological evolution, there arises an opportunity to transcend these limitations through the fusion of data-driven approaches and medical science. This project's genesis lies in the recognition of the transformative potential that machine learning holds in expediting and enhancing the accuracy of disease prediction. By automating diagnostic processes, we aspire to create a healthcare paradigm that not only meets but exceeds contemporary standards.

## 1.5 OBJECTIVES

This project is underpinned by a set of overarching objectives that drive its development and implementation:

- **Automated Diagnosis:** Introduce an automated diagnostic system that outpaces conventional manual methods, offering swifter and more objective disease predictions.
- **Enhanced Accuracy:** Utilize machine learning algorithms, notably the Random Forest Classifier, to elevate the precision of disease predictions, mitigating the errors associated with subjective human assessments.
- **User-Friendly Interface:** Develop a user-friendly interface, powered by Tkinter, ensuring accessibility for both healthcare professionals and individuals seeking preliminary health assessments.
- **Algorithm Analysis:** Evaluate various ML algorithms to determine the most suitable models for specific disease categories.
- **Data-Driven Insights:** Utilize patient data to refine the system continuously, improving its diagnostic capabilities.

## 1.6 SCOPE

The scope of this project transcends the mere prediction of diseases; it extends to revolutionizing the diagnostic landscape. Envisaged as a versatile tool, the system accommodates a diverse range of symptoms and diseases, making it adaptable to various medical scenarios. By providing a user-friendly interface, the project endeavors to democratize healthcare insights, ensuring accessibility without compromising the depth and accuracy of disease predictions.

As we embark on a detailed exploration of this project's background, methodologies, software requirements, and potential impact, we aim to illuminate the trajectory of "Disease Prediction using Machine Learning" within the broader context of healthcare innovation. Through this exploration, we envisage a future where technology and healthcare seamlessly converge, offering unprecedented advancements in disease prediction and diagnostics.

Detection of the disease, numerous methodologies will be evaluated such as KNN, NB, DT, CNN, SVM, and LR. At the end of this literature, the best performing ML models in respect of each diseasewill be concluded.

### Short-Term Scope

In its initial implementation, the system focuses on predicting diseases associated with heart, kidney, brain, and breast conditions. The choice of these categories reflects their prevalence and critical impact on public health.

### Long-Term Scope

The long-term vision includes:

- **Integration with Wearable Technology:** Incorporating real-time health data from devices like fitness trackers and smartwatches.
- **Global Accessibility:** Adapting the system for use in multiple languages and diverse cultural contexts.
- **Expanding Disease Categories:** Accommodating rare diseases, mental health conditions, and personalized medicine.
- **Telemedicine Integration:** Enhancing virtual healthcare consultations by providing preliminary diagnoses.

### Key Features Within Scope

1. **Versatility:** The system is designed to accommodate a wide range of symptoms, making it adaptable to numerous diagnostic scenarios.
2. **Scalability:** By leveraging cloud computing, the system can handle increasing data volumes and computational demands.
3. **Privacy and Security:** Ensuring compliance with regulations like GDPR and HIPAA to safeguard user data.

## Ethical and Societal Impact

The project recognizes the importance of ethical considerations, particularly when dealing with sensitive health data. By adhering to strict privacy protocols, it aims to build trust among users. Additionally, the democratization of advanced diagnostics has the potential to reduce healthcare disparities, empowering individuals in low-resource settings.

# CHAPTER 2

# FEASIBILITY STUDY

The **feasibility study** for the Disease Prediction Model is a critical assessment to determine whether the project can be successfully developed, implemented, and sustained. It systematically evaluates the technical, operational, and economic dimensions of the project, ensuring that the proposed system meets its goals within the constraints of resources, time, and practicality. This section elaborates on these dimensions in detail, expanding their scope and providing insights into the project's potential viability.

## 2.1 Technical Feasibility

Technical feasibility focuses on determining whether the technological infrastructure, tools, and expertise required to build and operate the Disease Prediction System are available and adequate.

- **Core Technologies**

The "Disease Prediction System" leverages state-of-the-art machine learning algorithms and tools.

- ➢ **Programming Language and Libraries:** Python, being versatile and developer-friendly, forms the backbone of this project. Libraries like Scikit-learn, Pandas, NumPy, and TensorFlow provide robust support for data manipulation, machine learning, and model evaluation.
- ➢ **Scikit-learn**: Ideal for implementing algorithms like Random Forest, Naïve Bayes, and Support Vector Machines.
- ➢ **TensorFlow and Keras**: Useful for deep learning models, such as Convolutional Neural Networks (CNN).
- ➢ **Tkinter:** Used for building a user-friendly graphical user interface (GUI).
- ➢ **Hardware Requirements**:

  The project requires minimal computational resources during deployment. For development and training, a standard system with the following specifications suffices:

- Processor: Multi-core (i5 or higher).
- RAM: At least 8 GB (16 GB recommended).
- Storage: SSD with a capacity of at least 256 GB for fast data processing.
- Cloud environments such as Google Collab or AWS Sage Maker can be leveraged to train machine learning models efficiently.

- ➢ **Data Availability**: Public datasets containing disease-related information, such as symptoms and diagnoses, are critical. Repositories like the UCI Machine Learning Repository or Kaggle provide datasets to build and train models. Additionally, data from hospitals and clinics can be integrated for increased accuracy.

- **Model Selection**

- ➢ **Algorithm Choice:**
    - Random Forest Classifier: Reliable for multi-class classification problems.
    - Naïve Baayes: Works well with categorical data like symptoms.
    - CNN: Exceptional for image-based diagnostics, such as X-rays or MRIs.
    - K-Nearest Neighbors (KNN): Effective for quick, low-complexity predictions.
- ➢ **Model Training and Testing:** Models will undergo rigorous training using labeled datasets. Techniques like k-fold cross-validation ensure robustness and generalizability.

- **Software Environment**

The system can be developed and deployed using:

- ➢ **Operating Systems:** Windows 11, Linux, or macOS.
- ➢ **Development Environments:** IDEs such as Jupyter Notebook, Spyder, and PyCharm.
- ➢ **Version Control:** GitHub or GitLab to maintain code integrity and manage collaborative development

- **Scalability**

The proposed system is highly scalable. It can handle larger datasets or more complex algorithms by:

- Upgrading hardware (e.g., GPU support for neural networks).
- Utilizing cloud computing for real-time data processing and prediction.

## 2.2 Operational Feasibility

Operational feasibility assesses whether the system can function effectively in its intended environment, considering usability, stakeholder needs, and alignment with current practices.

- **Stakeholder Analysis**
- **Primary Users:**
  - Individuals seeking preliminary diagnoses based on symptoms.
  - Healthcare professionals requiring quick diagnostic support.
- **Secondary Users:**
  - Researchers and data scientists analyzing trends in disease prevalence.
  - Hospital administrators optimizing patient workflows.
- **User Interface and Experience**

The system's GUI is designed for simplicity and efficiency. Built using Tkinter, it includes:

- **Input Fields**: Allowing users to enter symptoms in a structured manner.
- **Output Display:** Providing disease predictions along with confidence scores.
- **Accessibility Features:** Ensuring compatibility with assistive technologies for differently-abled users.

- **System Integration**
- **Healthcare Integration:** The system can be integrated into existing electronic health record (EHR) systems, enabling seamless data transfer.

- **Telemedicine Platforms**: This system can enhance online consultations by providing preliminary diagnoses.
- **API Availability:** A Flask API enables third-party integration, extending the system’s reach to mobile apps and web services.

- **Ethical and Privacy Considerations**

Given the sensitive nature of health data, the system adheres to strict privacy standards:

- Data Encryption: Ensuring secure transmission and storage of patient data.
- Compliance with Regulations: Adhering to HIPAA (Health Insurance Portability and Accountability Act) and GDPR (General Data Protection Regulation).
- Anonymization: Protecting patient identities in shared datasets.

- **Training and Support**

Operational success depends on user education. Resources include:

- Online tutorials and documentation.
- Customer support helplines for troubleshooting.

With these measures, the project ensures operational feasibility, addressing both user needs and ethical considerations.

### 2.3 Economic Feasibility

Economic feasibility evaluates whether the benefits of the project outweigh its costs, making it financially viable.

- **Development Costs**

The costs involved in the system’s development include:

- Human Resources: Salaries for developers, data scientists, and domain experts.
- Hardware and Software: Acquisition of systems, cloud services, and licensed tools if necessary.

Estimated development cost breakdown:

- Initial Investment: $10,000 - $15,000 (including salaries, hardware, and software).
- Cloud Services: $200/month for training and hosting (scalable based on usage).

- **Maintenance Costs**
- Regular Updates: Retraining models with new data.
- Bug Fixes and Feature Enhancements: Ensuring smooth functionality.
- Technical Support: Offering ongoing user assistance.

- **Revenue Generation**

The system can generate revenue through:

- Licensing: Hospitals and clinics can subscribe to the service.
- Freemium Model: Basic features for free, with advanced analytics offered at a premium.
- API Monetization: Charging third-party developers for access to the prediction API.

- **Cost-Benefit Analysis**

Benefits include:

- Reduction in healthcare costs by minimizing unnecessary hospital visits.
- Improved patient outcomes through early diagnosis.
- Revenue opportunities from commercial partnerships and licensing.

In summary, the system's economic benefits far outweigh its costs, ensuring financial feasibility.

### 2.4 Additional Considerations

- **Risk Analysis**
- Technical Risks:
    - Insufficient data quality leading to inaccurate predictions.
    - Overfitting or bias in machine learning models.
- Operational Risks:
    - Resistance from healthcare professionals accustomed to traditional diagnostics.

- Mitigation Strategies:
    - Regular model evaluation and updates.
    - User feedback mechanisms to improve system performance.

- **Long-term Vision**

The system's future enhancements include:

- Integration with wearable devices for real-time health monitoring.
- Expansion to include rare diseases and mental health conditions.
- Multilingual support for global accessibility.

# CHAPTER 3

# PROJECT OBJECTIVE

The objective of this project is to build a disease prediction system using machine learning (ML), where the user inputs symptoms, and the system predicts the most likely disease based on these symptoms. The project utilizes a Random Forest Classifier to train a model on a dataset of symptoms and corresponding diseases. The application is built using Python's Tkinter library to create a graphical user interface (GUI) that allows users to enter up to five symptoms and get the predicted disease in response.

**Key Objectives:**

1. Disease Prediction: Based on a set of symptoms entered by the user, the system predicts the most likely disease from a predefined list using machine learning.
2. Machine Learning Model: The project uses a Random Forest Classifier, which is a popular machine learning model for classification problems, to predict the disease. The model is trained on historical symptom-disease data.
3. Graphical User Interface (GUI): A simple GUI is created using the Tkinter library, where users can enter symptoms from a dropdown menu, input their name, and click a button to get the predicted disease.
4. Data Handling: The project reads and processes the training and testing data (CSV files) that contains symptoms and corresponding disease labels, replacing categorical disease names with numeric values for machine learning compatibility.
5. Accuracy Evaluation: After training the machine learning model, the accuracy of the predictions is evaluated using the test dataset to ensure the system is functioning correctly.
6. User Interaction: The system provides an interactive interface where the user can select symptoms from a list and get the disease prediction in a text field.

**High-level Steps:**

- Load and process the symptom-disease data.
- Train the Random Forest Classifier on the processed data.

- Use the classifier to predict the disease based on user input (symptoms).
- Display the predicted disease in the GUI.

Overall Aim:

To help users identify potential diseases based on their symptoms by leveraging machine learning techniques in a user-friendly interface.

Core Objective: Disease Prediction

Automated and Accurate Diagnosis

The primary goal is to provide an automated disease prediction system that utilizes input symptoms to predict the most likely disease from a predefined set. This objective addresses several critical healthcare challenges:

1. Timeliness: Early detection of diseases is crucial in preventing complications and ensuring better outcomes. For instance, detecting cardiac symptoms early could mitigate the risk of a heart attack.
2. Accuracy: By minimizing human bias and error, the system ensures that predictions are objective, consistent, and data-driven.
3. Scalability: Unlike manual diagnostic processes, this system can handle large volumes of cases simultaneously, making it suitable for widespread deployment.

Machine Learning Integration

The system employs the Random Forest Classifier, a robust and reliable algorithm for multi-class classification problems. Random Forest leverages an ensemble of decision trees to improve prediction accuracy and prevent overfitting. This ensures that the model is not only accurate but also generalizable across diverse datasets.

Supporting Objectives

1. Development of a User-Friendly Interface

   A key objective is to ensure that the system is accessible to all users, regardless of their technical expertise. The Tkinter-based graphical user interface (GUI) provides an intuitive platform where users can input symptoms and receive predictions effortlessly.

   - Simplicity: Dropdown menus allow users to select symptoms from a predefined list, eliminating the need for extensive typing or specialized knowledge.

- Clarity: The interface clearly labels input fields and prediction outputs, ensuring that users understand the system's functionality.
- Accessibility: The GUI design accommodates users with varying levels of technical proficiency, making it suitable for both patients and healthcare professionals.

2. Evaluation of ML Models

   Another critical objective is to assess the performance of different ML algorithms, ensuring that the best-performing models are chosen for specific disease categories. This comparative analysis includes:

   - Algorithm Diversity: The system tests algorithms like K-Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machines (SVM), and Convolutional Neural Networks (CNN).
   - Dataset Compatibility: Each algorithm is evaluated for its ability to handle specific types of data, such as categorical symptoms or image-based diagnostics.
   - Performance Metrics: Key metrics like accuracy, precision, recall, and F1 score guide the selection process.

Technical Objectives

1. Data Handling and Preprocessing

   Effective disease prediction relies on high-quality data. This objective focuses on:

   - Data Collection: Gathering comprehensive datasets containing symptoms and their corresponding diseases from trusted repositories like UCI and Kaggle.
   - Data Cleaning: Removing inconsistencies, missing values, and outliers to ensure the integrity of the training and testing datasets.
   - Feature Engineering: Transforming raw data into meaningful inputs for the ML models, such as encoding categorical variables or normalizing numerical features.

2. Model Training and Testing

   The system aims to train and test the chosen ML models rigorously to ensure optimal performance. This involves:

   - Cross-Validation: Techniques like k-fold cross-validation are employed to evaluate model stability and reduce overfitting.

    - Hyperparameter Optimization: Fine-tuning parameters like the number of trees in the Random Forest to maximize accuracy.
    - Validation: Using separate validation datasets to assess the model's performance on unseen data.

3. Integration with Real-Time Systems

   To enhance its practicality, the system is designed for integration with real-time platforms, such as telemedicine applications or electronic health records (EHRs). This allows:

    - Dynamic Data Input: Incorporating data from wearable devices or online consultations.
    - APIs: Providing an API for seamless integration with third-party applications.

Societal Objectives

1. Democratizing Healthcare

   The system aims to bridge the gap between advanced diagnostic tools and underserved populations.

    - Affordability: By reducing dependency on expensive diagnostic tests, the system makes healthcare more affordable.
    - Accessibility: The simple GUI ensures that individuals in remote or resource-poor settings can benefit from advanced diagnostics.

2. Preventive Healthcare

   Encouraging early detection and intervention is central to the system's design. This objective aligns with global healthcare trends focusing on prevention rather than cure.

3. Educational Tool

   The system doubles as an educational resource, providing insights into disease symptoms and their implications. This empowers users to make informed health decisions.

Operational Objectives

1. Ease of Deployment

   The system is designed for quick and hassle-free deployment, requiring minimal hardware and software resources.

    - Platform Compatibility: Works seamlessly on Windows, Linux, and macOS.
    - Cloud Integration: Cloud-based training environments like Google Colab or AWS SageMaker support scalability.
2. Stakeholder Engagement

   The system caters to a broad spectrum of users:
    - Primary Users: Patients seeking preliminary health assessments and healthcare providers requiring diagnostic support.
    - Secondary Users: Researchers analyzing health trends and hospital administrators optimizing patient workflows.

Ethical Objectives

1. Privacy and Security

   Given the sensitive nature of health data, the system prioritizes strict privacy and security measures.
    - Data Encryption: Ensures secure data transmission and storage.
    - Regulatory Compliance: Adheres to standards like HIPAA and GDPR.
    - Anonymization: Protects user identities in shared datasets.
2. Bias Mitigation

   The system is designed to minimize biases in disease prediction by:
    - Diverse Datasets: Training models on data representing various demographics.
    - Algorithmic Fairness: Regularly evaluating models for potential biases.

Future Objectives

1. Expansion of Disease Categories

   The system aims to include rare and complex diseases, enhancing its diagnostic capabilities.
2. Integration with AI Technologies

   Incorporating advanced AI techniques, such as natural language processing (NLP) for unstructured data like medical reports, is a key future goal.
3. Global Reach

   The long-term vision includes making the system multilingual and culturally adaptable for global accessibility.

By focusing on accuracy, accessibility, and user empowerment, the project aims to redefine healthcare diagnostics, making it more efficient, inclusive, and impactful. This comprehensive approach ensures that the system not only meets immediate needs but also adapts to future healthcare challenges.

# CHAPTER 4

# HARDWARE AND SOFTWARE REQUIREMENTS

**Hardware Requirements:**

1. **Processor (CPU):**
    - A minimum of Intel i3 processor or equivalent.
    - Recommended: Intel i5 or higher for faster data processing and model training.
2. **RAM:**
    - Minimum: 4 GB of RAM.
    - Recommended: 8 GB or higher, especially if you work with large datasets or more complex models.
3. **Hard Disk Space:**
    - Minimum: 2 GB of free disk space to store project files and libraries.
    - Recommended: 5 GB or more if the project involves large datasets or additional libraries.
4. **Graphics Card (GPU):**
    - Not mandatory for this project, as it does not involve heavy graphical computations or deep learning. The standard onboard graphics or CPU will suffice.
5. **Display:**
    - Minimum resolution: 1024x768 for displaying the graphical user interface.
6. **Internet Connection:**
    - A stable internet connection is required for downloading libraries and datasets, as well as for any online resources, tools, or API usage.

**Software Requirements:**

1. **Operating System:**
    - Windows 7/8/10/11 or Linux (Ubuntu) or MacOS.
    - The software should be compatible with Python and required libraries.
2. **Programming Language:**
    - Python 3.x – Python is the primary language for developing this project.

3. **Integrated Development Environment (IDE):**
    - Visual Studio Code, PyCharm, Jupyter Notebook, or any Python IDE of your choice.
    - Text Editor: gedit or VSCode (for code editing and debugging).
4. **Python Libraries:**
    - Tkinter (for GUI development) – This comes pre-installed with Python.
    - NumPy – For numerical operations and array manipulations.
    - Pandas – For data manipulation and analysis (especially for handling CSV data).
    - Scikit-learn – For machine learning algorithms, especially the Random Forest Classifier.
    - Matplotlib (optional) – For data visualization (if needed for graph plotting during data analysis).
    - Pillow (optional) – If any images are involved in the project.

To install these libraries, you can use the following commands:

bash

Copy code

```
pip install numpy pandas scikit-learn matplotlib pillow
```

5. **Additional Tools (Optional):**
    - Git – For version control and collaborative development.
    - Jupyter Notebook – For an interactive development environment, especially useful when working with machine learning and data analysis.
    - Anaconda – An optional Python distribution that comes with many data science libraries pre-installed, which can be helpful for managing environments and dependencies.

# CHAPTER 5

# PROJECT FLOW

## 4.1 DATA-FLOW DIAGRAM

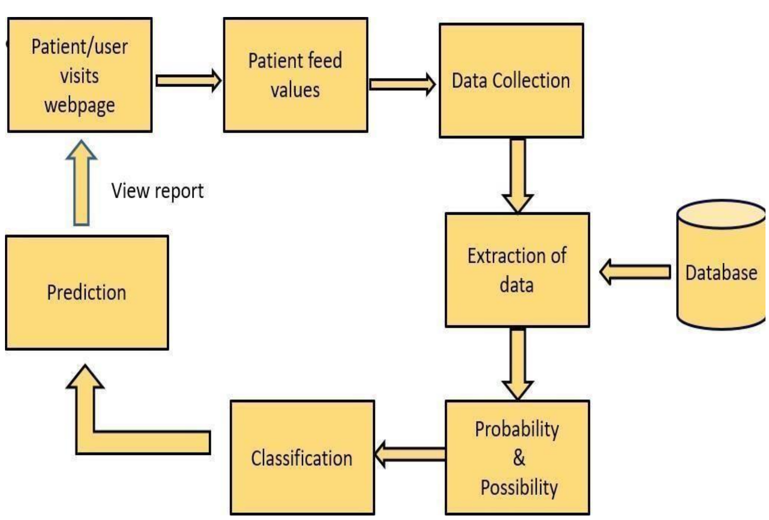


Fig 5.1 Data Flow Diagram

**Flow Explanation:**

1. **Patient/User Visits Webpage:**
   - A patient or user accesses the system through a webpage interface to interact with the platform.
2. **Patient Feed Values:**
   - The patient enters their values or data (e.g., symptoms, health information) into the system.
3. **Data Collection:**
   - The system collects these inputs and prepares them for processing.

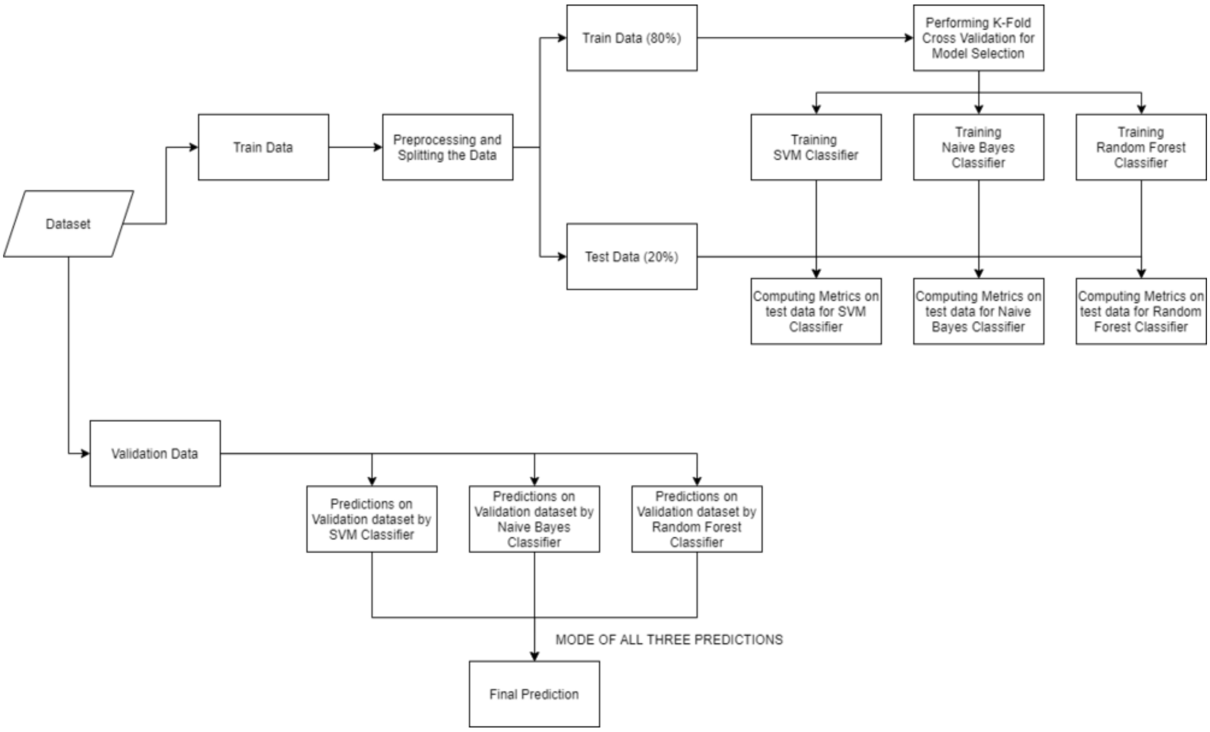
4. **Extraction of Data:**
   - Relevant information is extracted from the patient inputs and transferred to a Database for storage and further use.

5. **Probability & Possibility:**
    - The system processes the extracted data to analyze possibilities and probabilities for specific outcomes. This could involve statistical calculations or predictive models.
6. **Classification:**
    - Based on the analysis, the data is classified into specific categories (e.g., disease risk levels, symptoms grouping, etc.).
7. **Prediction:**
    - The classified data is used to generate predictions, such as health outcomes or possible diagnoses.
8. **View Report:**
    - The results (e.g., predictions or insights) are displayed back to the patient or user, who can view the report.

**Key Components:**

- Database: Stores the extracted patient data for analysis and future reference.
- Flow Arrows: Represent the direction of data movement between components.
- Processes: Steps like Data Collection, Extraction of Data, Classification, and Prediction form the core processing stages.

## 5.2 SYSTEM FLOW DIAGRAM



Dataset
Train Data
Preprocessing and Splitting the Data
Train Data (80%)
Test Data (20%)
Performing K-Fold Cross Validation for Model Selection
Training SVM Classifier
Training Naive Bayes Classifier
Training Random Forest Classifier
Computing Metrics on test data for SVM Classifier
Computing Metrics on test data for Naive Bayes Classifier
Computing Metrics on test data for Random Forest Classifier
Validation Data
Predictions on Validation dataset by SVM Classifier
Predictions on Validation dataset by Naive Bayes Classifier
Predictions on Validation dataset by Random Forest Classifier
MODE OF ALL THREE PREDICTIONS
Final Prediction

Fig 5.2 System Flow Diagram

**Flow Breakdown:**

1. **Dataset**:
   - The system starts with a dataset that will be split into training, testing, and validation sets.
2. **Preprocessing and Splitting the Data**:
   - The dataset is divided into Train Data (80%) and Test Data (20%).
   - A separate Validation Data is also used later for final prediction evaluation.
3. **Model Training Process:**
   - Using the Train Data:

        - The data is used to train three different classifiers:
            - **Support Vector Machine (SVM) Classifier**
            - **Naive Bayes Classifier**
            - **Random Forest Classifier**
    - **K-Fold Cross-Validation** is performed during training to ensure better model selection and reduce overfitting.

4. **Model Testing Process**:
    - Each trained classifier (SVM, Naive Bayes, and Random Forest) is tested on the Test Data.
    - Metrics are computed for all three classifiers:
        - Computing Metrics on Test Data for SVM Classifier
        - Computing Metrics on Test Data for Naive Bayes Classifier
        - Computing Metrics on Test Data for Random Forest Classifier
5. **Validation Stage**:
    - The Validation Data is fed into the trained classifiers to make predictions:
        - Predictions on Validation Dataset by SVM Classifier
        - Predictions on Validation Dataset by Naive Bayes Classifier
        - Predictions on Validation Dataset by Random Forest Classifier
6. **Mode of All Three Predictions**:
    - The predictions from all three classifiers are combined, and the **mode** (the most frequent prediction) is selected as the final output. This ensemble method ensures more reliable results by leveraging the strengths of multiple classifiers.
7. **Final Prediction**:
    - The system produces the final prediction as the output, which is derived from the combined predictions of all three classifiers.

# CHAPTER 6

# PROJECT OUTCOME

## 6.1 User Interface

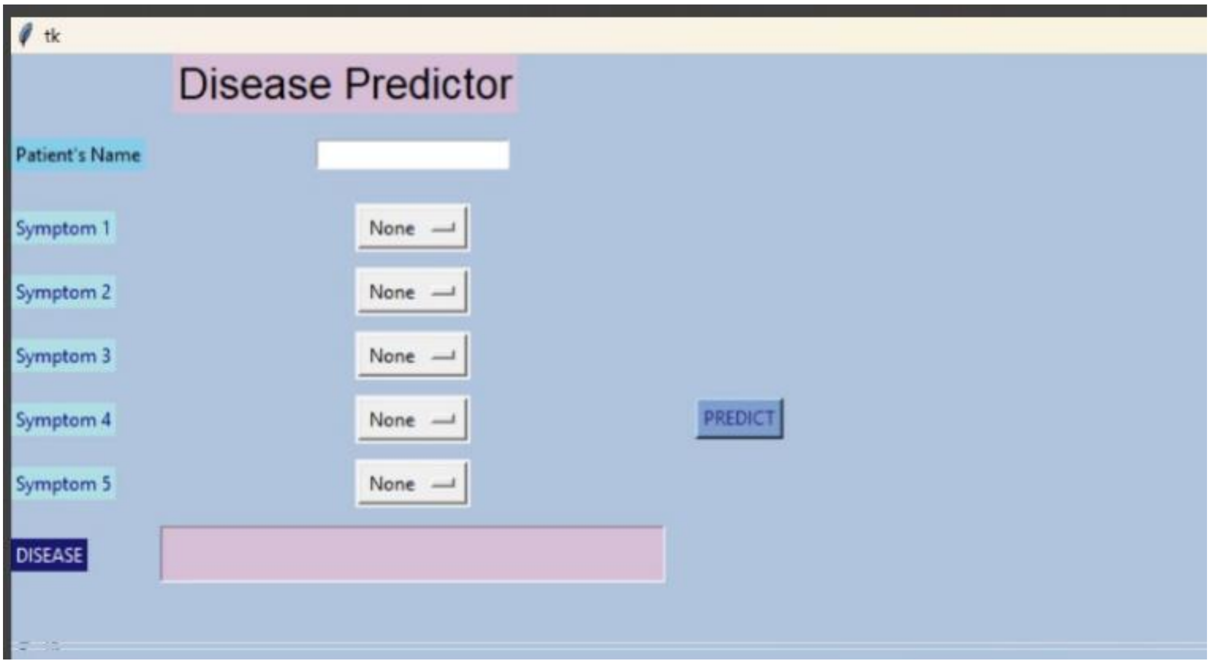


Fig 6.1 Input Fields Using Tkinter

1. **Patient’s Name Field:**
   - This text box is blank and awaits user input to personalize the prediction process for a specific patient.
2. **Symptom Dropdown Menus:**
   - Five dropdown menus, labeled "Symptom 1" through "Symptom 5," are set to the default value “None.” This indicates that no symptoms have been selected yet.
   - Users can open these dropdowns to choose relevant symptoms from a predefined list to provide input for the prediction process.
3. **Predict Button:**
   - A “PREDICT” button is positioned centrally and is inactive until at least one symptom is selected. Clicking it will trigger the disease prediction process.

4. **Output Section:**
   - The output field, labeled “DISEASE,” is empty, awaiting the result of the prediction process.

**Purpose:**

This clean, user-friendly layout allows users to start from a blank slate, progressively inputting data to receive a disease prediction. The default "None" values ensure users do not accidentally input incorrect symptoms, providing a clear and structured interface for disease diagnosis.

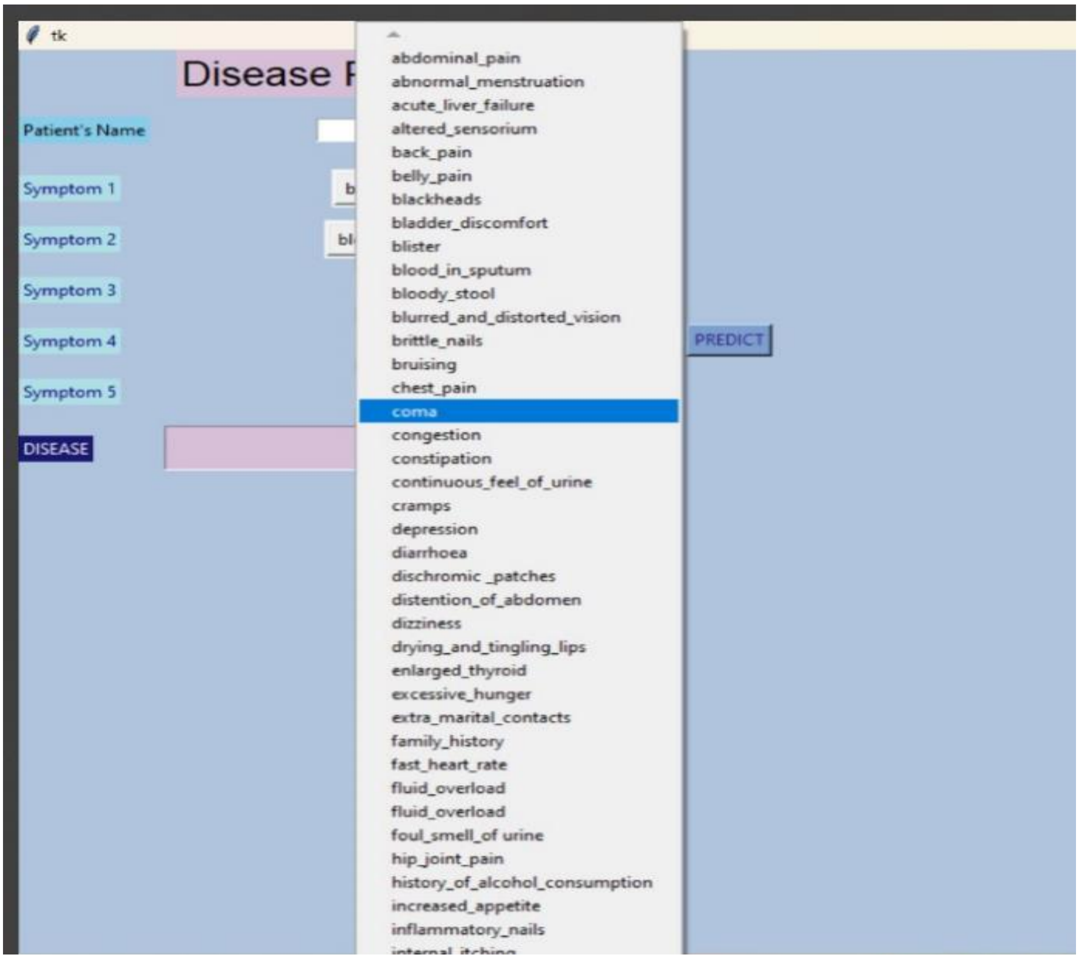
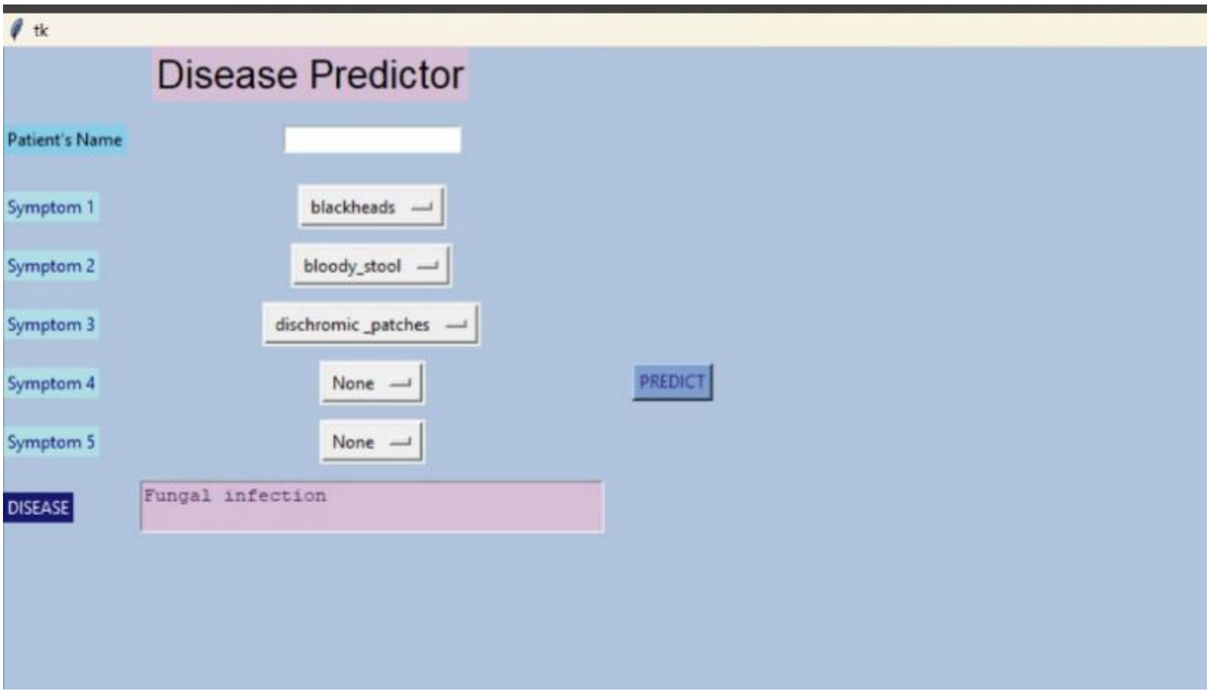


Fig 6.2 List of Symptoms

- Comprehensive Symptom List:
- The dropdown includes a wide range of symptoms, such as "abdominal_pain," "chest_pain," "dizziness," "depression," "excessive_hunger," "coma," and more. This suggests the system is designed to handle a broad spectrum of medical conditions.

- Ease of Selection:
- Users can quickly scroll through the list and select symptoms that apply to the patient. The organization appears alphabetical, simplifying navigation.
- Dynamic Functionality:
- Once a symptom is selected, it is likely sent to the backend for processing. Multiple dropdowns ensure that multiple symptoms can be chosen, improving diagnostic accuracy.
- Application Purpose:
- This setup caters to efficient data input for disease prediction, ensuring users have access to a detailed symptom repository that the prediction model can analyze.

## 5.2 OUTPUT SCREENSHOTS



Patient's Name

Symptom 1 blackheads

Symptom 2 bloody_stool

Symptom 3 dischromic _patches

Symptom 4 None

Symptom 5 None

DISEASE Fungal infection

Fig 6.3  The Output of the given symptoms entered by user

- **Symptom Selection Dropdowns:**
- The application features five dropdown menus labeled "Symptom 1" to "Symptom 5."
- These dropdowns allow users to select symptoms from predefined options. Some dropdowns already show selected values, such as:
- **Symptom 1:** Blackheads

- **Symptom 2:** Bloody stool
- **Symptom 3:** Dischromic patches
- **Symptom 4 and Symptom 5:** Set to "None," implying no additional symptoms were reported.
- The dropdown menu likely includes a comprehensive list of possible symptoms a patient might experience. Selecting "None" in a dropdown ensures flexibility when fewer than five symptoms need to be provided.
- **Prediction Button:**
- A button labeled "PREDICT" is centrally positioned in the interface. This button triggers the underlying disease prediction algorithm once clicked.
- **Output Section:**
- A section labeled "DISEASE" at the bottom left displays the predicted disease. In the example provided, the output is "Fungal infection," shown in a distinct purple box for visibility.
- **Color Scheme:**
- The background is light blue, with various elements such as buttons, fields, and output boxes colored differently to enhance usability and aesthetics. Each section's layout is clean and structured to guide the user intuitively.
-

---

- **Functionality and Workflow**
- **Input Collection:**
- The user begins by entering the patient's name in the input field. While optional in terms of disease prediction, this makes the tool patient-specific.
- The user then selects symptoms from the dropdown menus. The dropdown options are likely mapped to a precompiled list of symptoms that the disease prediction model recognizes.
- **Prediction Process:**

- Clicking the "PREDICT" button executes the disease prediction process. The button interacts with a backend system where:
- Symptoms are processed as inputs.
- A machine learning (ML) model or a rule-based algorithm predicts the disease based on the symptoms provided.
- The disease prediction algorithm could rely on a dataset containing correlations between symptoms and diseases.
- **Output Display:**
- The result, in this case, "Fungal infection," is displayed in the output section labeled “DISEASE.” This response is generated based on the algorithm’s analysis of the input symptoms.

# REFERENCES

- S. Cheema, S. Srivastava, P. K. Srivastava and B. K. Murthy, "A standard compliant Blood Bank Management System with enforcing mechanism," 2015 International Conference on
- Computing, Communication and Security (ICCCS)
- M. Sarode, A. Ghanekar, S. Krishnadas, Y. Patil and M. Parmar, "Intelligent Blood Management System," 2019 IEEE Bombay Section Signature Conference (IBSSC), Mumbai, India, 2019
- M. Y. Esmail and Y. S. H. Osman, "Computerized Central Blood Bank Management System (CCBBMS)," 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 2018
- A. S. Cheema, S. Srivastava, P. K. Srivastava and B. K. Murthy, "A standard compliant Blood Bank Management System with enforcing mechanism," 2015 International Conference on Computing, Communication and Security (ICCCS), Pointe aux
- Piments, Mauritius, 2015