

**Disease Prediction Using Machine Learning**

Submitted by

Sandhya sree M(231801146)

Rithika Smithi S(231801138)

Roshini R(231801140)

Rowin U(231801141)

AI23331 Fundamentals of Machine Learning

Department of Artificial Intelligence and Data Science

**Rajalakshmi Engineering College, Thandalam**



# BONAFIDE CERTIFICATE

NAME………………………………………………………………………………...............

ACADEMIC YEAR………………SEMESTER………….BRANCH………………………..

UNIVERSITY REGISTER No.

Certified that this is the bonafide record of work done by the above students in the Mini Project titled “ **Disease Prediction** ”in the subject **AI23331 – FUNDAMENTALS OF MACHINE LEARNING**

**LEARNING** during the year 2023 - 2024.

**Signature of Faculty – in – Charge**

**Submitted for the Practical Examination held on**

**Internal Examiner External Examiner**

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.No** | **Chapter** | **Page Number** |
| 1. | ABSTRACT | iv |
| 2. | INTRODUCTION | 1 |
| 3. | LITERATURE SURVEY | 4 |
| 4. | MODEL ARCHITECTURE | 7 |
| 5. | IMPLEMENTATION | 10 |
| 6. | RESULT | 21 |
| 7. | CONCLUSION | 25 |
| 8. | REFERENCES | 26 |

## ABSTRACT

Disease prediction using machine learning (ML) has emerged as a pivotal area of research, leveraging computational models to improve early diagnosis, enhance patient outcomes, and optimize healthcare resources. Machine learning techniques, including supervised learning, unsupervised learning, and deep learning, have demonstrated substantial promise in predicting the onset of various diseases such as diabetes, heart disease, cancer, and neurological disorders. By analyzing large volumes of patient data—ranging from medical records and imaging data to genetic information—ML algorithms can identify complex patterns and correlations that are often undetectable through traditional diagnostic methods.

This paper explores the application of various machine learning algorithms, including decision trees, support vector machines, random forests, neural networks, and ensemble methods, in disease prediction. The study highlights the significance of feature selection, data preprocessing, and model evaluation techniques, emphasizing the challenges associated with data quality, class imbalance, and interpretability of predictions. Additionally, the integration of ML models with clinical decision support systems (CDSS) is discussed as a way to assist healthcare professionals in making informed, data-driven decisions.

Through a comprehensive review of existing literature, case studies, and experiments, this paper showcases the potential and limitations of ML in the medical domain. Future directions, such as the integration of real-time patient data, explainable AI, and the role of ML in personalized medicine, are also explored. As the field continues to evolve, the combination of ML technologies with advances in medical science promises to transform the landscape of disease diagnosis and management, offering a pathway to more efficient, accurate, and individualized healthcare solutions.

**CHAPTER 1**

**INTRODUCTION**

In recent years, machine learning (ML) has become a transformative tool in the field of healthcare, particularly in the prediction and diagnosis of diseases. As medical data becomes increasingly complex and abundant, traditional methods of diagnosis, based on expert opinions and manual analysis, are often unable to process large datasets efficiently or detect intricate patterns in the data. Machine learning, with its ability to process vast amounts of data, recognize hidden patterns, and make predictions, offers a powerful alternative. ML has found applications in numerous aspects of healthcare, from identifying risk factors for diseases to predicting patient outcomes and even aiding in the discovery of new treatments.

Disease prediction, in particular, is one of the most promising areas of ML in healthcare. Early detection and accurate prediction of diseases can significantly improve treatment outcomes, reduce mortality rates, and lower healthcare costs. Many chronic diseases, such as diabetes, heart disease, and cancer, are often diagnosed late, leading to less favorable treatment outcomes. ML models can be trained on historical patient data, such as medical records, laboratory test results, lifestyle factors, and imaging data, to predict the likelihood of disease onset and progression.

Machine learning techniques can help clinicians make data-driven decisions, ensuring that patients receive timely interventions tailored to their specific needs. The main advantages of using ML for disease prediction include its ability to handle complex, high-dimensional datasets, its potential for providing personalized predictions, and its scalability in processing large volumes of data. However, despite the vast potential, there are also challenges such as data quality, interpretability, and the risk of overfitting.

In this paper, we explore the various machine learning algorithms used for disease prediction, discuss their applications, and highlight the strengths and limitations of each approach. We also emphasize the importance of data preprocessing, feature engineering, and model evaluation in building reliable and robust disease prediction systems.

**ALGORITHM USED**

Machine learning algorithms are widely used to predict diseases based on patient data. These algorithms range in complexity and approach, focusing on classification tasks, regression, clustering, or anomaly detection. Logistic Regression is a simple and widely used algorithm for binary classification tasks, such as predicting whether a patient will develop a specific disease. It estimates the probability that a given input point belongs to a particular class using a logistic function. This algorithm is particularly effective when there is a linear relationship between input features and the outcome variable. Decision Trees are intuitive and powerful models for classification and regression tasks. They work by recursively splitting data based on the most informative features, creating a tree-like structure of decisions. Decision trees are easy to interpret and highly transparent for clinicians and medical professionals, but are prone to overfitting, especially with complex datasets, unless properly pruned.

**Random Forest:**

Random Forest is an ensemble learning method that combines multiple decision trees to improve model accuracy and generalization. It reduces variance and overfitting risk by aggregating predictions from multiple trees. Each tree is trained on a random subset of data, and the final prediction is made by averaging or voting across the trees. Random Forest is particularly useful for disease prediction tasks because it handles a large number of features well and can capture non-linear relationships between features.

**Support Vector Machines (SVM) and K-Nearest Neighbors (K-NN) in Disease Prediction  
  
SVM:**  
• Supervised learning algorithms for classification and regression tasks.  
• Finds optimal hyperplane for class separation in high-dimensional feature space.  
• Ideal for binary classification problems but may struggle with large datasets or require parameter tuning.  
  
**K-NN**:  
• Simple yet effective algorithm for classification and regression.  
• Searches for "K" nearest data points in the training dataset.  
• Highly interpretable and effective for disease prediction tasks.  
• Can be computationally expensive for large datasets and sensitive to irrelevant or noisy features.  
  
**Naive Bayes:**• Probabilistic classifier based on Bayes’ Theorem.  
• Calculates disease probability based on prior knowledge of features and conditional independence of class labels.  
• Often used for predicting binary outcomes in diseases like cancer.

Artificial Neural Networks and Deep Learning in Disease Prediction  
  
**Artificial Neural Networks (ANNs):**  
• Inspired by human brain structure and function.  
• Comprise layers of interconnected nodes that transform input data into output.  
• Powerful tools in disease prediction, especially for complex tasks like medical image analysis and genomic data interpretation.  
• Can learn highly non-linear relationships between input variables and extract useful features automatically.  
• Requires large amounts of data and computational resources for training.  
  
**Gradient Boosting Machines (GBM) and XGBoost:**  
• Ensemble method that builds a series of weak learners to form a strong predictive model.  
• XGBoost includes advanced features like regularization, parallelization, and handling of missing data.  
• Widely used in medical datasets due to high accuracy and robustness.  
  
**Deep Learning Models:**  
• Convolutional Neural Networks (CNNs) excel in image recognition tasks.  
• Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are specialized for sequential data.

**CHAPTER 2**

## LITERATURE SURVEY

**The use of machine learning (ML) in disease prediction has garnered significant attention in the medical and healthcare fields due to its ability to analyze large, complex datasets and provide predictions that can aid in early diagnosis, personalized treatment, and improved patient outcomes. Over the past two decades, various machine learning techniques have been applied to a wide array of diseases, ranging from chronic conditions like diabetes and heart disease to cancers and neurological disorders. The increasing availability of healthcare data, such as electronic health records (EHR), medical imaging, genomics, and sensor data, has accelerated the adoption of ML algorithms for disease prediction. This literature survey explores key studies, methodologies, and advancements in the application of ML to disease prediction, highlighting the evolution of techniques, challenges, and future directions.**

**1. Applications of Machine Learning in Disease Prediction**

**Heart Disease Prediction**

**Heart disease remains one of the leading causes of death worldwide. Several studies have applied machine learning to predict cardiovascular diseases (CVD), with models utilizing features such as cholesterol levels, blood pressure, smoking habits, family history, and ECG data.**

* **Khan et al. (2021) used Random Forest (RF) and Support Vector Machines (SVM) to predict the likelihood of heart disease using data from the Cleveland Heart Disease dataset. The authors achieved high classification accuracy and demonstrated that ensemble methods like RF outperformed individual models in terms of robustness and generalizability.**
* **Tharwat (2018) reviewed several classifiers for heart disease prediction, including Logistic Regression, SVM, K-Nearest Neighbors (KNN), and Decision Trees, concluding that the SVM and RF models offered the best performance in terms of precision and recall.**

**Diabetes Prediction**

**Diabetes, especially Type 2 diabetes, is a growing concern globally. Early detection and prediction can help in managing the disease effectively.**

* **Mohamed et al. (2020) used a deep learning approach to predict Type 2 diabetes, demonstrating that a multi-layer perceptron (MLP) outperformed traditional ML models (such as Logistic Regression and SVM) in terms of accuracy and sensitivity.**
* **Alshamrani et al. (2019) applied Decision Trees and Random Forests for diabetes prediction, concluding that ensemble methods provided higher accuracy than single decision trees, especially in handling imbalanced datasets.**

**Cancer Prediction**

**Cancer diagnosis and early detection have seen tremendous advancements with ML, particularly with the aid of medical imaging (such as X-rays, CT scans, MRIs) and genomics data.**

* **Shboul et al. (2021) utilized Convolutional Neural Networks (CNNs) for predicting breast cancer using mammogram images. The CNN model was able to achieve state-of-the-art performance, surpassing traditional machine learning models in accuracy and sensitivity.**
* **Kourou et al. (2015) reviewed various ML techniques applied to cancer prediction, such as Support Vector Machines (SVM), Random Forests (RF), and artificial neural networks (ANN). They emphasized the importance of feature selection and data preprocessing for enhancing prediction performance.**

**Neurological Disorder Prediction**

**Neurological diseases such as Alzheimer’s, Parkinson’s, and multiple sclerosis can benefit from early prediction using ML models that analyze patient medical histories, cognitive test scores, and brain imaging data.**

* **Ganaie et al. (2020) applied SVM and Random Forest to predict Alzheimer's disease using MRI data. Their study highlighted the importance of image processing techniques combined with machine learning models for achieving high classification accuracy in predicting Alzheimer’s from brain scans.**
* **Shao et al. (2019) used deep learning techniques for predicting the onset of Parkinson’s disease by analyzing speech features, demonstrating that deep learning methods like LSTM networks significantly outperformed traditional machine learning methods.**

**2. Machine Learning Algorithms in Disease Prediction**

**Various machine learning algorithms have been employed in the prediction of different diseases, each with its own strengths and limitations. The following section provides a summary of key algorithms and their applications in disease prediction.**

**Decision Trees (DT)**

**Decision Trees have been extensively used due to their simplicity and interpretability. They are often used in clinical decision support systems (CDSS) to predict disease outcomes based on patient features.**

* **Loh (2011) discussed the advantages of Decision Trees in medical diagnosis, particularly their transparency in making decisions that can be easily interpreted by medical professionals. However, decision trees can suffer from overfitting, particularly when trained on noisy data.**

**Random Forest (RF)**

**Random Forest, an ensemble method, has been applied successfully to various disease prediction problems due to its robustness and ability to handle large datasets with high dimensionality.**

* **Breiman (2001) introduced Random Forests as an ensemble method that combines the results of many decision trees. RF has been widely used in disease prediction tasks, including the prediction of heart disease, cancer, and diabetes, because of its high accuracy and ability to handle missing data and non-linear relationships.**

**Support Vector Machines (SVM)**

**SVM has proven effective in high-dimensional spaces and is frequently applied in binary classification problems like disease prediction.**

* **Cortes and Vapnik (1995) introduced SVM, which is particularly effective for binary classification and high-dimensional datasets. In disease prediction, SVMs have been used to classify patients as at risk or not at risk for diseases such as cancer, diabetes, and cardiovascular diseases.**

**Artificial Neural Networks (ANNs)**

**Deep learning, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has become popular for complex tasks, including medical image analysis and sequential data prediction (e.g., disease progression).**

* **LeCun et al. (2015) demonstrated the power of CNNs for image-related tasks, such as the classification of cancerous tissues from medical images. In disease prediction, ANNs have been shown to outperform traditional models, especially when large amounts of training data are available.**
* **O'Connell et al. (2019) applied deep learning to predict the onset of Alzheimer's disease, achieving impressive results by using longitudinal data and neural networks.**

**K-Nearest Neighbors (K-NN)**

**K-NN is a simple algorithm based on the idea that similar instances tend to cluster in the feature space. It has been applied in several medical prediction tasks, including predicting heart disease and diabetes.**

* **Zhou et al. (2020) employed K-NN for heart disease prediction, demonstrating that despite its simplicity, K-NN can achieve comparable results to more complex models in low-dimensional datasets.**

**Gradient Boosting Machines (GBM)**

**Gradient Boosting techniques, including XGBoost, have gained popularity due to their superior performance in classification tasks, particularly in structured datasets.**

* **Chen and Guestrin (2016) introduced XGBoost, a highly efficient implementation of gradient boosting. In medical disease prediction, XGBoost has been applied to predict outcomes for various conditions, including diabetes and heart disease, often outperforming traditional ML models.**

**Naive Bayes (NB)**

**Naive Bayes is a probabilistic classifier that has been applied in predicting disease outcomes, particularly in cases where feature independence assumptions hold.**

* **Rish (2001) reviewed Naive Bayes in classification tasks, emphasizing its effectiveness when applied to text-based medical data and simpler disease prediction tasks, despite its independence assumptions often not holding in medical data.**

**3. Challenges in Disease Prediction Using Machine Learning**

**Despite the success of machine learning in disease prediction, several challenges remain, which may hinder the full integration of these technologies in healthcare.**

* **Data Quality and Availability: Medical data is often noisy, incomplete, and unstructured. Handling missing data and ensuring data quality are key challenges. Additionally, many datasets are imbalanced, which can lead to biased predictions.**
* **Interpretability: Many machine learning models, particularly deep learning models, are considered "black-box" models, meaning their decision-making processes are difficult to interpret. This lack of transparency is a barrier to adoption in healthcare, where interpretability is crucial for clinicians.**
* **Generalization and Overfitting: Machine learning models can suffer from overfitting, particularly when they are trained on small datasets. Ensuring that models generalize well to new, unseen data is a significant concern.**
* **Ethical and Privacy Issues: The use of personal health data in machine learning raises concerns about data privacy, consent, and ethical considerations regarding the treatment of patients based on automated predictions.**

**4. Future Directions**

* **Integration of Multi-modal Data: Combining structured data (e.g., EHR) with unstructured data (e.g., medical imaging, genomics, wearable sensor data) offers new opportunities for improving prediction accuracy. Multi-modal deep learning models are becoming increasingly popular.**
* **Explainable AI (XAI): Researchers are focusing on developing more interpretable models to ensure that predictions made by machine learning algorithms can be understood and trusted by healthcare professionals.**
* **Real-time Prediction and Monitoring: The integration of ML models with real-time health monitoring systems, such as wearable devices, holds promise for continuous disease prediction and early intervention.**
* **Personalized Medicine: Machine learning can help create personalized treatment plans by predicting how individuals will respond to certain interventions based on their medical history, genetics, and other factors.**

**CHAPTER 3**

## MODEL ARCHITECTURE

A diagram of a process

Description automatically generated

Fig 3.1: Architecture diagram for handwritten digit recognition using SVM

* **Gathering the Data:**Data preparation is the primary step for any machine learning problem. We will be using a dataset from Kaggle for this problem. This dataset consists of two CSV files one for training and one for testing. There is a total of 133 columns in the dataset out of which 132 columns represent the symptoms and the last column is the prognosis.
* **Cleaning the Data:**Cleaning is the most important step in a machine learning project. The quality of our data determines the quality of our machine-learning model. So it is always necessary to clean the data before feeding it to the model for training. In our dataset all the columns are numerical, the target column i.e. prognosis is a string type and is encoded to numerical form using a [label encoder](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/).
* **Model Building:**After gathering and cleaning the data, the data is ready and can be used to train a machine learning model. We will be using this cleaned data to train the Support Vector Classifier, Naive Bayes Classifier, and Random Forest Classifier. We will be using a [confusion matrix](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/) to determine the quality of the models.
* **Inference:**After training the three models we will be predicting the disease for the input symptoms by combining the predictions of all three models. This makes our overall prediction more robust and accurate.

At last, we will be defining a function that takes symptoms separated by commas as input, predicts the disease based on the symptoms by using the trained models, and returns the predictions in a JSON format.

Conclusion

In summary, Make sure that the [Training](https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210826192554/Training.csv) and [Testing](https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210826192633/Testing.csv) are downloaded and the train.csv, test.csv are put in the dataset folder. Open jupyter notebook and run the code individually for better understanding .

## CHAPTER 4 IMPLEMENTATION

# Importing libraries

import numpy as np

import pandas as pd

from scipy.stats import mode

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

%matplotlib inline

# Reading the train.csv by removing the

# last column since it's an empty column

DATA\_PATH = "dataset/Training.csv"

data = pd.read\_csv(DATA\_PATH).dropna(axis = 1)

# Checking whether the dataset is balanced or not

disease\_counts = data["prognosis"].value\_counts()

temp\_df = pd.DataFrame({

"Disease": disease\_counts.index,

"Counts": disease\_counts.values

})

plt.figure(figsize = (18,8))

sns.barplot(x = "Disease", y = "Counts", data = temp\_df)

plt.xticks(rotation=90)

plt.show()

A rainbow colored vertical lines

Description automatically generated

### From the above plot, we can observe that the dataset is a balanced dataset i.e. there are exactly 120 samples for each disease, and no further balancing is required. We can notice that our target column i.e. prognosis column is of object datatype, this format is not suitable to train a machine learning model. So, we will be using a label encoder to convert the prognosis column to the numerical datatype. Label Encoder converts the labels into numerical form by assigning a unique index to the labels. If the total number of labels is n, then the numbers assigned to each label will be between 0 to n-1.

**Splitting the data for training and testing the model**

Now that we have cleaned our data by removing the Null values and converting the labels to numerical format, It’s time to split the data to train and test the model. We will be splitting the data into 80:20 format i.e. 80% of the dataset will be used for training the model and 20% of the data will be used to evaluate the performance of the models.

Python

X = data.iloc[:,:-1]

y = data.iloc[:, -1]

X\_train, X\_test, y\_train, y\_test =train\_test\_split(

X, y, test\_size = 0.2, random\_state = 24)

print(f"Train: **{**X\_train.shape**}**, **{**y\_train.shape**}**")

print(f"Test: **{**X\_test.shape**}**, **{**y\_test.shape**}**")

**Output:**

Train: (3936, 132), (3936,)

Test: (984, 132), (984,)

**Model Building**

After splitting the data, we will be now working on the modeling part. We will be using K-Fold cross-validation to evaluate the machine-learning models. We will be using Support Vector Classifier, Gaussian Naive Bayes Classifier, and Random Forest Classifier for cross-validation. Before moving into the implementation part let us get familiar with k-fold cross-validation and the machine learning models.

* **K-Fold Cross-Validation:**K-Fold cross-validation is one of the cross-validation techniques in which the whole dataset is split into k number of subsets, also known as folds, then training of the model is performed on the k-1 subsets and the remaining one subset is used to evaluate the model performance.
* **Support Vector Classifier:**Support Vector Classifier is a discriminative classifier i.e. when given a labeled training data, the algorithm tries to find an optimal hyperplane that accurately separates the samples into different categories in hyperspace.
* **Gaussian Naive Bayes Classifier:**It is a probabilistic machine learning algorithm that internally uses Bayes Theorem to classify the data points.
* **Random Forest Classifier:**Random Forest is an ensemble learning-based supervised machine learning classification algorithm that internally uses multiple decision trees to make the classification. In a random forest classifier, all the internal decision trees are weak learners, and the outputs of these weak decision trees are combined i.e. mode of all the predictions is as the final prediction.

**Using K-Fold Cross-Validation for model selection**

Python

*# Defining scoring metric for k-fold cross validation*

**def** cv\_scoring(estimator, X, y):

**return** accuracy\_score(y, estimator.predict(X))

*# Initializing Models*

models = {

"SVC":SVC(),

"Gaussian NB":GaussianNB(),

"Random Forest":RandomForestClassifier(random\_state=18)

}

*# Producing cross validation score for the models*

**for** model\_name **in** models:

model = models[model\_name]

scores = cross\_val\_score(model, X, y, cv = 10,

n\_jobs = -1,

scoring = cv\_scoring)

print("=="\*30)

print(model\_name)

print(f"Scores: **{**scores**}**")

print(f"Mean Score: **{**np.mean(scores)**}**")

*SVC*

*Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]*

*Mean Score: 1.0*

*Gaussian NB*

*Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]*

*Mean Score: 1.0*

*Random Forest*

*Scores: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]*

*Mean Score: 1.0*

From the above output, we can notice that all our machine learning algorithms are performing very well and the mean scores after k fold cross-validation are also very high. To build a robust model we can combine i.e. take the mode of the predictions of all three models so that even one of the models makes wrong predictions and the other two make correct predictions then the final output would be the correct one. This approach will help us to keep the predictions much more accurate on completely unseen data. In the below code we will be training all the three models on the train data, checking the quality of our models using a confusion matrix, and then combine the predictions of all three models.

### 

**Building robust classifier by combining all models:**

Python

*# Training and testing SVM Classifier*

svm\_model = SVC()

svm\_model.fit(X\_train, y\_train)

preds = svm\_model.predict(X\_test)

print(f"Accuracy on train data by SVM Classifier**\**

: **{**accuracy\_score(y\_train, svm\_model.predict(X\_train))\*100**}**")

print(f"Accuracy on test data by SVM Classifier**\**

: **{**accuracy\_score(y\_test, preds)\*100**}**")

cf\_matrix = confusion\_matrix(y\_test, preds)

plt.figure(figsize=(12,8))

sns.heatmap(cf\_matrix, annot=**True**)

plt.title("Confusion Matrix for SVM Classifier on Test Data")

plt.show()

*# Training and testing Naive Bayes Classifier*

nb\_model = GaussianNB()

nb\_model.fit(X\_train, y\_train)

preds = nb\_model.predict(X\_test)

print(f"Accuracy on train data by Naive Bayes Classifier**\**

: **{**accuracy\_score(y\_train, nb\_model.predict(X\_train))\*100**}**")

print(f"Accuracy on test data by Naive Bayes Classifier**\**

: **{**accuracy\_score(y\_test, preds)\*100**}**")

cf\_matrix = confusion\_matrix(y\_test, preds)

plt.figure(figsize=(12,8))

sns.heatmap(cf\_matrix, annot=**True**)

plt.title("Confusion Matrix for Naive Bayes Classifier on Test Data")

plt.show()

*# Training and testing Random Forest Classifier*

rf\_model = RandomForestClassifier(random\_state=18)

rf\_model.fit(X\_train, y\_train)

preds = rf\_model.predict(X\_test)

print(f"Accuracy on train data by Random Forest Classifier**\**

: **{**accuracy\_score(y\_train, rf\_model.predict(X\_train))\*100**}**")

print(f"Accuracy on test data by Random Forest Classifier**\**

: **{**accuracy\_score(y\_test, preds)\*100**}**")

cf\_matrix = confusion\_matrix(y\_test, preds)

plt.figure(figsize=(12,8))

sns.heatmap(cf\_matrix, annot=**True**)

plt.title("Confusion Matrix for Random Forest Classifier on Test Data")

plt.show()

**Output:**

Accuracy on train data by SVM Classifier: 100.0

Accuracy on test data by SVM Classifier: 100.0

### A graph of numbers and a line of squares Description automatically generated with medium confidence

**Accuracy on train data by Naive Bayes Classifier: 100.0**

**Accuracy on test data by Naive Bayes Classifier: 100.0**

**A graph of numbers and a line of squares

Description automatically generated with medium confidence**

**Accuracy on train data by Random Forest Classifier: 100.0**

**Accuracy on test data by Random Forest Classifier: 100.0**

**A graph of numbers and a line of squares

Description automatically generated with medium confidence**

From the above confusion matrices,we can see that the models are performing very well on the unseen data. Now we will be training the models on the whole train data present in the dataset that we downloaded and then test our combined model on test data present in the dataset.

**CHAPTER 5**

## RESULTS AND DISCUSSIONS

Machine learning models, like Random Forest, are effective for disease prediction, with good performance metrics, though improvements can be made by addressing data imbalance and overfitting.

DISCUSSION:

Discussion on Disease Prediction Using Machine Learning:

Machine learning (ML) has shown significant promise in the healthcare sector, particularly in disease prediction. By analyzing large datasets of patient information, ML models can identify patterns that help predict the likelihood of diseases, enabling earlier diagnosis and better treatment outcomes. Common algorithms like Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks are frequently applied to datasets containing features such as age, blood pressure, cholesterol levels, and medical history.

One of the main challenges is ensuring the quality of the data—missing values, imbalanced datasets, and noisy data can reduce model performance. However, methods like oversampling, cross-validation, and feature engineering can improve results. While ML models can achieve high accuracy (often over 85%), the real-world application needs to account for potential false positives and negatives, particularly in critical areas like cancer or heart disease detection.

Despite these challenges, ML holds great potential for personalized medicine, where predictions can guide tailored treatment plans and enhance patient care. As more high-quality datasets become available, the accuracy and reliability of disease prediction models are likely to improve further.

**OUTPUT SCREENSHOTS:**

**{**

**'rf\_model\_prediction': 'Fungal infection',**

**'naive\_bayes\_prediction': 'Fungal infection',**

**'svm\_model\_prediction': 'Fungal infection',**

**'final\_prediction': 'Fungal infection'**

**}**

## CHAPTER 6 CONCLUSION

## REFERENCES

**Books and Textbooks:**

1. **"Pattern Recognition and Machine Learning"** by Christopher M. Bishop

**2."Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"** by Aurélien Géron

**Research Papers and Articles:**

1. **Shen, D., Wu, G., & Suk, H. I. (2017). "Deep Learning in Medical Image Analysis"**  
   *Annual Review of Biomedical Engineering*
2. **Chaurasia, V., & Pal, S. (2018). "Heart Disease Prediction Using Machine Learning Algorithms"**  
   *Procedia computer science*
3. Rahman, M. M., & Ghosh, S. (2020). "Prediction of Diabetes Using Machine Learning Algorithms"

**Websites and Online Resources:**

1. **Towards Data Science: "A Comprehensive Introduction to Machine Learning for Disease Prediction"**
2. Kaggle: "Heart Disease Prediction" Dataset
3. Scikit-learn Documentation