VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018, Karnataka

ADVANCED MACHINE LEARNING (18AI72)

Mini Project Report on

“ ML-Driven Disease Prediction for Early Intervention”

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering
in
Artificial Intelligence & Machine Learning

Submitted by

Manish N Gond	**1BI20AI028**
Praveen N	**1BI21AI402**

for the academic year 2022-23

Under the Guidance of
Mrs. Subha Meenakshi S
(Guest Professor)
Department of AI&ML, BIT
Bengaluru-560 004

Department of Artificial Intelligence & Machine Learning
Bangalore Institute of Technology
K.R. Road, V.V. Pura, Bengaluru-560 004
2023-24

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY
Department of Artificial Intelligence & Machine Learning

K.R. Road, V.V.Pura, Bengaluru-560 004

Certificate

This is to certify that Block Chain Technology mini project work entitled “**ML-Driven Disease Prediction for Early Intervention**” carried out by

USN	Name
1BI20AI028	**Manish N Gond**
1BI21AI402	**Praveen N**

bonafide students of **Bangalore Institute of Technology** in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** under Visvesvaraya Technological University, Belagavi, during the academic year 2023-24 is true representation of mini project work completed satisfactorily.

Mrs. Subha Meenakshi S	Dr. Jyothi D.G	Dr. Ashwath M.U
Guest Professor	Professor & HoD	Principal
Dept. of AI&ML	Dept. of AI&ML	BIT
BIT, Bengaluru.	BIT, Bengaluru.	Bengaluru.

Name of the Examiners, Signature with date

1.

2.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of a task would be incomplete without crediting the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

We would like to profoundly thank Management of Bangalore Institute of Technology for providing such a healthy environment for successful completion of mini project work.

We would like to express our thanks to the Principal **Dr. Aswath M.U.** for his encouragement that motivates us for the successful completion of mini project work.

It gives us immense pleasure to thank **Dr. Jyothi D.G**, Professor & Head, Department of Artificial Intelligence & Machine Learning for her constant support and encouragement.

We would like to express my deepest gratitude to my mini project guide **Mrs. Subha Meenakshi S** for their constant support and guidance throughout the Mini Project work.

We are very much pleasured to express our sincere gratitude to the friendly co-operation showed by all the staff members of Artificial Intelligence and Machine Learning Department, BIT.

Last but not the least, we would here by acknowledge and thank our friends and family who have been our source of inspiration always instrumental in successful completion of the Project work.

Date:	Manish N Gond
Place: Bengaluru	Praveen N

ABSTRACT

The disease prediction project involves utilizing machine learning algorithms to predict diseases based on symptoms. Initially, the dataset is loaded and examined for missing values and statistical information. Visualizations, including pie charts and bar plots, are created to illustrate the distribution and frequency of various diseases within the dataset.

Following data preparation, the dataset is split into training and testing sets. Three distinct models Support Vector Classifier (SVC), Gaussian Naive Bayes, and Random Forest Classifier are trained using different symptom features. Cross-validation is performed to evaluate the models' performance, displaying individual model scores and mean scores.

The code then proceeds to train and evaluate the models. For each classifier, the accuracy on both the training and testing data is calculated, accompanied by confusion matrices to visualize prediction accuracy. The final section involves training the models on the entire dataset and predicting diseases using separate test data. Confusion matrices are created to analyze model performance, displaying how accurately the models predict diseases compared to the actual diagnoses. The project aims to build accurate disease prediction models that could assist in medical diagnosis by leveraging machine learning techniques.

INDEX

LIST OF FIGURES

CHAPTER – 1

INTRODUCTION

INTRODUCTION

1.1 Overview

The use of machine learning (ML) in disease prediction is a rapidly growing field with significant potential for improving healthcare. ML models can efficiently predict the disease of a human based on the symptoms they exhibit. ML algorithms can identify patterns and relationships that may not be apparent to human experts, leading to more accurate and reliable predictions. To implement a disease prediction using ML project, one would typically start by collecting and preprocessing relevant data, selecting an appropriate ML algorithm, training the model on the data, and evaluating its performance using metrics such as accuracy, precision, recall, and F1 score. The model can then be used to make predictions about an individual's disease risk based on their input data and their performance can be visualized. The ultimate goal is to develop robust and accurate disease prediction systems that can assist healthcare professionals in making timely and accurate diagnoses. The field of disease prediction using machine learning (ML) has witnessed significant advancements and applications. ML models have been developed to predict various diseases based on symptoms and other medical data. These models have demonstrated high efficiency in disease prediction through the classification of diseases, and they hold great potential in the prediction and diagnosis of a wide range of medical conditions.

The application of ML in disease prediction has the potential to significantly impact the healthcare industry by enabling early and accurate disease diagnosis, which in turn can lead to improved patient outcomes and more effective treatment strategies. This intersection of machine learning and healthcare represents a promising avenue for research and application, with the capacity to profoundly impact the healthcare industry by introducing advanced diagnostic capabilities and refined patient care approaches.

The dataset utilized for disease prediction through machine learning encompasses a comprehensive array of 132 parameters, capturing a diverse range of medical attributes, symptoms, and diagnostic indicators. These parameters form the foundation for predicting 42 distinct types of diseases, providing a nuanced and multifaceted dataset for model training.

The dataset is structured into two distinct CSV files - one dedicated to training and the other for testing purposes. Prior to model development, the dataset undergoes rigorous preprocessing steps aimed at cleansing the data by eliminating any missing or extraneous information. This process ensures the integrity and relevance of the data used for training and testing predictive models.

Within the dataset, pertinent patient information is included, comprising extensive medical histories, detailed symptom profiles, and diagnostic test outcomes. This multifaceted patient-centric data contributes significantly to the robustness and accuracy of the disease prediction models.

The process of developing an effective disease prediction model involves a systematic approach that encompasses data collection, meticulous data preprocessing techniques, judicious feature selection methodologies, and the selection and training of machine learning models. Among the commonly employed algorithms for disease prediction are logistic regression, decision trees, random forests, and support vector machines, each offering unique strengths in predictive analytics.

The dataset serves as the basis for training machine learning models, including but not limited to Support Vector Classifier, Naive Bayes Classifier, and Random Forest Classifier. Subsequently, these trained models undergo comprehensive evaluation using diverse performance metrics such as accuracy scores and confusion matrices. The objective is to develop highly accurate and reliable disease prediction systems capable of providing timely and precise diagnostic support to healthcare professionals.

CHAPTER – 2

LITERATURE REVIEW

LITERATURE REVIEW

2.1 Paper Review

Paper 1:

[1] **Title: Disease Prediction from Various Symptoms Using Machine Learning -** Rinkal Keniya, Aman Khakharia, Vruddhi Shah, Vrushabh Gada, Ruchi Manjalkar, Tirth Thaker, Mahesh Warang, and Ninad Mehendale- 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).

Summary:

In [1], disease prediction system that uses machine learning algorithms to predict diseases based on an individual's symptoms, age, and gender. A dataset was collected and preprocessed, and different ML algorithms were used, including Fine, Medium and Coarse Decision trees, Gaussian Naive Bayes, Kernel Naive Bayes, Fine, Medium and Coarse KNN, Weighted KNN, Subspace KNN, and RUSBoosted trees. The Weighted KNN model achieved the highest accuracy of 93.5% for disease prediction. The system can be helpful in diagnosing diseases, managing medicine resources, and improving the recovery process, especially in emergency situations where sufficient facilities and resources are unavailable.

Paper 2:

[2] **Title: Disease Prediction Using Machine Learning over Big Data –** Vinitha S, Sweetlin S, Vinusha H, and Sajini S - Computer Science & Engineering: An International Journal (CSEIJ), Vol.8, No.1, February 2018

Summary:

In [2], The proposed research aims to revolutionize disease prediction and early detection using machine learning algorithms over big data analytics. Traditional disease prediction systems often face challenges with incomplete medical data and fail to predict subtypes of diseases caused by the occurrence of one primary disease. To address these limitations, our research introduces a novel approach that leverages machine learning algorithms and map reduce techniques to handle both structured and unstructured medical data. By integrating various data sources such as gene information, DNA methylation, and miRNA, our system aims to provide accurate and comprehensive disease predictions, including subtypes and associated risks.

Paper 3:

[3] **Title: Disease Prediction Using Machine Learning -** Anant Agrawal, Harshit Agrawal, Shivam Mittal, and Mradula Sharma - 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT).

Summary:

In [3], Misdiagnosis is a major factor in the wrong course of treatment in medicine. Machine learning can help in this regard as a lot of medical data is publicly available. In this paper, we propose a hybrid machine learning model comprising of genetic algorithm and support vector machine for disease prediction. We have tested our model on three datasets of liver, diabetes, and heart diseases. Our approach involves feature extraction using genetic algorithm and support vector machine. The proposed model has shown promising results in accurately predicting diseases using publicly available medical data.

Paper 4:

[4] **Title: Disease Prediction System using Support Vector Machine and Multilinear Regression -** Md. Ehtisham Farooqui and Dr. Jameel Ahmad - International Journal of Innovative Research in Computer Science & Technology (IJIRCST).

Summary:

In [4], The proposed research aims to develop a disease prediction system using machine learning algorithms to accurately predict possible diseases based on symptoms. The system will utilize Support Vector Machine (SVM) for classification and Multilinear Regression (MLR) for predicting the results. By collecting structured datasets of symptoms and diagnoses from local hospitals and open-source libraries, the system will be trained to predict various diseases based on patient symptoms. The accuracy of the system is expected to be high, enabling early detection and timely intervention for better patient care. The research will contribute to the advancement of healthcare technology and improve disease prediction capabilities for medical institutions and healthcare communities.

Paper 5:

[5] Title: A Diabetic Disease Prediction Model Based on Classification Algorithms - Ravinder Ahuja, Subhash C. Sharma, and Maaruf Ali - 2019 IEEE Translations and content mining are permitted for academic research.

Summary:

In [5], Diabetes is a chronic disease that affects millions of people worldwide. Early detection and prediction of diabetes can significantly improve patient outcomes and reduce healthcare costs. In this research article, we explore the use of machine learning techniques to predict diabetes and its potential impact on healthcare. We apply data pre-processing techniques, LDA feature selection, and five classification algorithms (RF, LR, DT, MLP, and SVC) to predict diabetic patients. We also use k-fold cross-validation to evaluate the performance of our model. Our results show that the Random Forest algorithm outperforms the other algorithms with an accuracy of 86.67%. Our research has important implications for the healthcare industry and patient care, as it demonstrates the potential of machine learning techniques to improve diabetes prediction and ultimately, patient outcomes.

2.2 EXISTING SYSTEM

The existing system proposed in the paper "Disease Prediction from Various Symptoms Using Machine Learning" utilizes machine learning algorithms to predict diseases based on an individual's symptoms, age, and gender. The system collects and preprocesses a dataset consisting of gender, symptoms, and age of individuals, and then feeds this data as input to various ML algorithms, including Fine, Medium and Coarse Decision trees, Gaussian Naive Bayes, Kernel Naive Bayes, Fine, Medium and Coarse KNN, Weighted KNN, Subspace KNN, and RUSBoosted trees. The system aims to provide a faster and more accurate diagnosis, especially in emergency situations where sufficient facilities and resources may be unavailable.

2.3 PROBLEM STATEMENT:

The aim of this project is to develop robust machine learning models capable of accurately predicting diseases based on a multitude of parameters, including patient symptoms, medical history, and diagnostic indicators. Healthcare systems face the challenge of timely and accurate diagnosis of various diseases based on patient symptoms and medical history. The lack of precise diagnostic methods often leads to delayed treatments and misdiagnoses, impacting patient outcomes and healthcare efficiency.Leveraging a dataset containing 132 parameters and covering 42 different types of diseases, the objective is to construct predictive models that can effectively assist healthcare professionals in diagnosing diseases promptly and accurately.

2.4 PROPOSED SYSTEM

The proposed system is designed to revolutionize disease diagnosis by harnessing machine learning techniques to predict diseases based on patient symptoms and medical data. It begins with a meticulous process of data collection, compiling comprehensive datasets containing patient information, symptoms, medical history, and diagnostic test results. This data undergoes rigorous preprocessing steps, including data cleaning to handle missing values and encoding categorical variables, ensuring it's ready for machine learning model training. Feature engineering is a crucial aspect, involving the identification of pertinent features that contribute significantly to disease prediction. Multiple machine learning models such as Support Vector Machines (SVM), Naive Bayes, and Random Forest classifiers are trained using this prepared dataset. To validate the models' effectiveness and reliability, extensive evaluation using cross-validation techniques is performed. Metrics like accuracy, precision, recall, and F1-score are employed to ensure the models' robustness and accuracy.

Once validated, the trained models are deployed into a user-friendly interface or platform for healthcare practitioners' use. This system allows real-time predictions based on input symptoms and patient data, providing probability-based disease predictions.

CHAPTER – 3

SYSTEM REQUIREMENTS

SYSTEM REQUIREMENT

3.1 REQUIREMENT SPECIFICATION

1. Data Collection

Data collection is defined as the procedure of collecting, measuring and analyzing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis based on collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information. The most critical objective of data collection is ensuring that information-rich and reliable data is collected for statistical analysis so that data-driven decisions can be made for research.

2. Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

3. Data Labelling

Supervised machine learning, entails training a predictive model on historical data with predefined target answers. An algorithm must be shown which target answers or attributes to look for. Mapping these target attributes in a dataset is called labelling. Data labelling takes much time and effort as datasets sufficient for machine learning may require thousands of records to be labelled. For instance, if your image recognition algorithm must classify types of bicycles, these types should be clearly defined and labelled in a dataset.

4. Data Selection

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. Data selection precedes the actual practice of data collection. This definition distinguishes data selection from selective data reporting (selectively excluding data that is not supportive of a research hypothesis) and interactive/active data selection (using collected data for monitoring activities/events, or conducting secondary data analyses). The process of selecting suitable data for a research project can impact data integrity. After having collected all information, a data analyst chooses a subgroup of data to solve the defined problem.

For instance, if you save your customers' geographical location, you don't need to add their cell phones and bank card numbers to a dataset. But purchase history would be necessary. The selected data includes attributes that need to be considered when building a predictive model.

5. Data Pre-processing

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. The purpose of pre-processing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.

Data formatting. The importance of data formatting grows when data is acquired from various sources by different people. The first task for a data scientist is to standardize record formats. A specialist checks whether variables representing each attribute are recorded in the same way. Titles of products and services, prices, date formats, and addresses are examples of variables. The principle of data consistency also applies to attributes represented by numeric ranges.

Data cleaning. This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with mean attributes. A specialist also detects outliers 4 observations that deviate significantly from the rest of distribution. If an outlier indicates erroneous data, a data scientist deletes or corrects them if possible. This stage also includes removing incomplete and useless data objects.

Data sampling. Big datasets require more time and computational power for analysis. If a dataset is too large, applying data sampling is the way to go. A data scientist uses this technique to select a smaller but representative data sample to build and run models much faster, and at the same time to produce accurate outcomes.

6. Data Transformation

Data transformation is the process of converting data from one format or structure into another format or structure. Data transformation is critical to activities such as data integration and data management. Data transformation can include a range of activities: you might convert data types, cleanse data by removing nulls or duplicate data, enrich the data, or perform aggregations, depending on the needs of your project. Scaling. Data may have numeric attributes (features) that span different ranges, for example, millimetres, meters, and kilometres. Scaling is about converting these attributes so that they will have the same scale, such as between 0 and 1, or 1 and 10 for the smallest and biggest value for an attribute. Decomposition. Sometimes finding patterns in data with features representing complex concepts is more difficult. Decomposition technique can be applied in this case. During decomposition, a specialist converts higher level features into lower level ones. In other words, new features based on the existing ones are being added. Decomposition is mostly used in time series analysis. For example, to estimate a demand for air conditioners per month, a market research analyst converts data representing demand per quarters. Aggregation. Unlike decomposition, aggregation aims at combining several features into a feature that represents them all. For example, you have collected basic information about your customers and particularly their age. To develop a demographic segmentation strategy, you need to distribute them into age categories, such as 16-20, 21-30, 31-40, etc.

7. Data Splitting

A dataset used for machine learning should be partitioned into three subsets 4 training, test, and validation sets Training set. A data scientist uses a training set to train a model and define its optimal parameters 4 parameters it must learn from data. Test set. A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It is crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above. Validation set. The purpose of a validation set is to tweak a model's hyperparameters 4 higher-level structural settings that cannot be directly learned from data. These settings can express, for instance, how complex a model is and how fast it finds patterns in data. The proportion of a training and a test set is usually 80 to 20 percent, respectively. A training set is then split again, and its 20 percent will be used to form a validation set. At the same time, machine learning practitioner Jason Brownlee suggests using 66 percent of data for training and 33 percent for testing. A size of each subset depends on the total dataset size. The more training data a data scientist uses, the better the potential model will perform. Consequently, more results of model testing data lead to better model performance and generalization capability.

8. Modelling

After pre-processing the collected data and split it into three subsets, we can proceed with a model training. This process entails <feeding= the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data 4 an answer you want to get with predictive analysis. The purpose of model training is to develop a model. Two model training styles are most common 4 supervised and unsupervised learning. The choice of each style depends on whether you must forecast specific attributes or group data objects by similarities. Model evaluation and testing: The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That is the optimization of model parameters to achieve an algorithm's best performance.

9. Model Deployment

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome. Often, an organization's IT systems are incompatible with traditional model-building languages, forcing data scientists and programmers to spend valuable time and brainpower rewriting them.

3.2 Hardware Requirements

- RAM: Minimum of 4GB
- Processor : Intel® Celeron® CPU N3060
- Disk : 32 GB or Above
- Input Device : Keyboard
- Output Device : Monitor

3.3 Software Requirements

- Programming Language : Python
- Basic Text-Editor: Anaconda3
- Operating System : Windows 7 or Higher

CHAPTER – 4

SYSTEM ARCHITECHTURE

SYSTEM ARCHITECTURE

In the below section, a brief overview of how the project is designed is given.

4.1 System Design

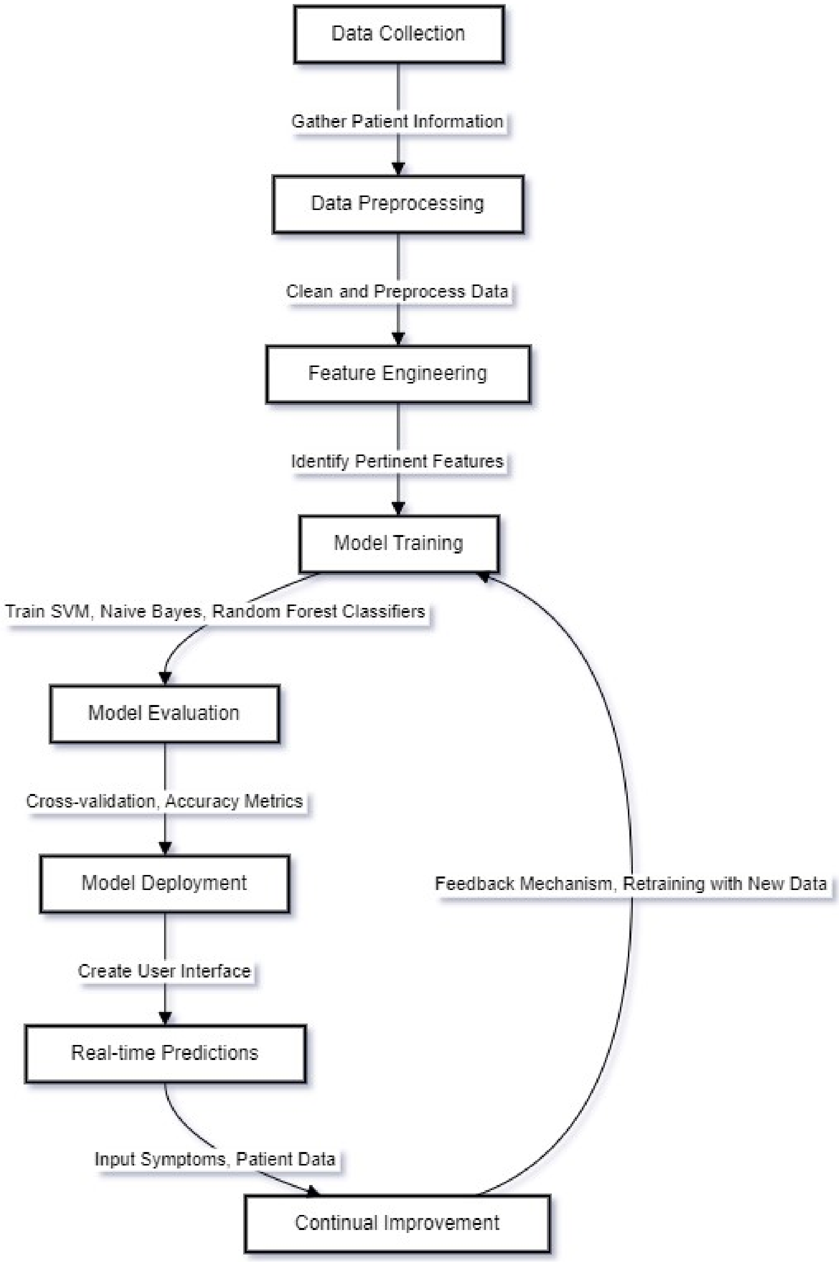

Data Collection
Gather Patient Information
Data Preprocessing
Clean and Preprocess Data
Feature Engineering
Identify Pertinent Features
Model Training
Train SVM, Naive Bayes, Random Forest Classifiers
Model Evaluation
Cross-validation, Accuracy Metrics
Model Deployment
Create User Interface
Real-time Predictions
Input Symptoms, Patient Data
Continual Improvement
Feedback Mechanism, Retraining with New Data

Fig 4.1: Disease Prediction System Workflow

The model has the following stages in its workflow:

Data Collection:

Acquiring patient-related information from various sources such as medical records, databases, or IoT devices. Gathers structured datasets containing symptoms, medical history, and diagnostic results for further analysis.

Data Preprocessing:

Cleansing and transforming raw data to ensure accuracy and consistency. Involves handling missing values, removing irrelevant data, encoding categorical variables, and standardizing data formats.

Feature Engineering:

Identifying and engineering relevant features for disease prediction models. Extracts, selects, or creates features that hold significant predictive power, enhancing model performance.

Model Training:

Training machine learning models using prepared datasets. Trains multiple classifiers such as Support Vector Machines (SVM), Naive Bayes, and Random Forests on the preprocessed data.

Model Evaluation:

Assessing the performance of trained models. Evaluates models using cross-validation techniques, assessing accuracy, precision, recall, and other metrics.

Model Deployment:

Integrating trained models into an accessible system for predictions. Involves creating user interfaces or APIs to enable real-time predictions based on input symptoms or patient data.

Real-time Predictions:

Providing immediate disease predictions based on user input. Accepts symptoms or patient data as input, processes it through deployed models, and produces predicted diseases or conditions**.**

Continual Improvement:

Iteratively enhancing the system's accuracy and performance. Incorporates feedback mechanisms, collects user input, and periodically retrains models with new data to improve predictions over time.

4.2 Working of Disease Prediction System

The system architecture diagram outlines the workflow of a disease prediction system using machine learning. It starts with data collection, gathering patient information vital for analysis. This data undergoes preprocessing, involving cleaning and organizing to ensure its suitability for analysis. Feature engineering then identifies critical aspects for disease prediction, aiding in the creation of a robust model.

The subsequent step involves model training, where various classifiers such as Support Vector Machines (SVM), Naive Bayes, and Random Forest models are trained using the prepared dataset. These models are evaluated using cross-validation techniques and accuracy metrics to determine their effectiveness in disease prediction.

Following evaluation, the best-performing model is deployed, often integrated into a user interface for real-time predictions. The system continuously improves through a feedback mechanism, incorporating new data and updating the model periodically to enhance accuracy and relevance. This cycle of improvement ensures that the disease prediction system remains up-to-date and effective in diagnosing diseases based on patient symptoms and medical data.

CHAPTER – 5

IMPLEMENTATION

IMPLEMENTATION AND TESTING

5.1 Implementation:

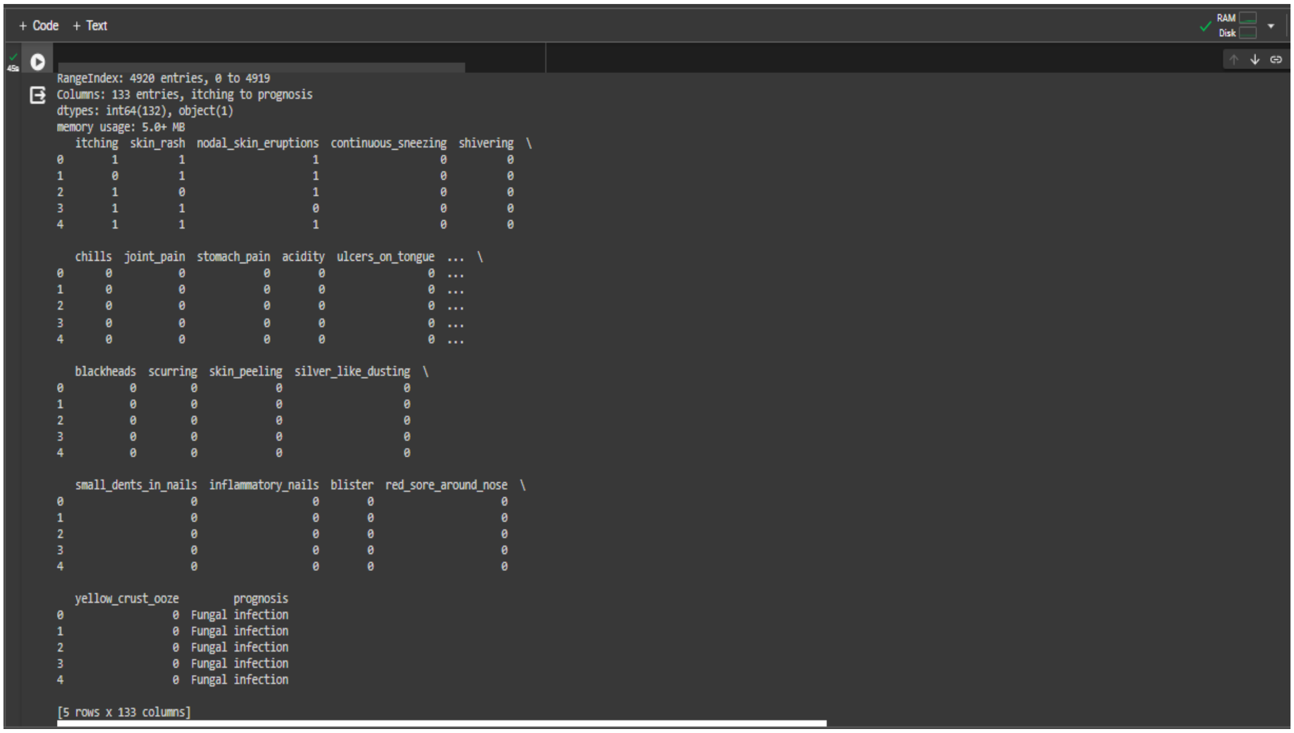
```
 import numpy as np
 import pandas as pd
 from scipy.stats import mode
 import matplotlib.pyplot as plt
 import seaborn as sns
 from sklearn.preprocessing import LabelEncoder
 from sklearn.model_selection import train_test_split, cross_val_score
 from sklearn.svm import SVC
 from sklearn.naive_bayes import GaussianNB
 from sklearn.ensemble import RandomForestClassifier
 from sklearn.metrics import accuracy_score, confusion_matrix
 %matplotlib inline

 data = pd.read_csv('/content/drive/MyDrive/ML datasets/Training.csv').dropna(axis = 1)

 diseases = ["Fungal infection", "Hepatitis C", "Hepatitis E", "Alcoholic hepatitis", "Tuberculosis",
 "Common Cold", "Pneumonia", "Dimorphic hemmorhoids(piles)", "Heart attack", "Varicose veins",
 "Hypothyroidism", "Hyperthyroidism", "Hypoglycemia", "Osteoarthristis", "Arthritis", "(vertigo)
 Paroymsal Positional Vertigo", "Acne", "Urinary tract infection", "Psoriasis", "Hepatitis D", "Hepatitis
 B", "Allergy", "hepatitis A", "GERD", "Chronic cholestasis", "Drug Reaction", "Peptic ulcer diseae",
 "AIDS", "Diabetes", "Gastroenteritis", "Bronchial Asthma", "Hypertension", "Migraine", "Cervical
 spondylosis", "Paralysis (brain hemorrhage)", "Jaundice", "Malaria", "Chicken pox", "Dengue",
 "Typhoid", "Impetigo"]

 cases = [120] * len(diseases)
 plt.figure(figsize=(10, 10))
 plt.pie(cases, labels=diseases, autopct='%1.1f%%')
 plt.title("Distribution of diseases")

plt.show()
data.isnull().sum()
col=data.columns
col
data.head()
data.info()
data.describe()
data.prognosis.value_counts()
data.shape
# 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()
# Encoding the target value into numerical
# value using LabelEncoder
encoder = LabelEncoder()
data["prognosis"] = encoder.fit_transform(data["prognosis"])
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}")
# 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)
}

selected_columns = ['itching', 'skin_rash', 'nodal_skin_eruptions', 'stomach_pain', 'prognosis']

train_selected = data[selected_columns]

sns.pairplot(train_selected, hue='prognosis')

plt.show()
# 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)}")
# 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()
# Training the models on whole data
final_svm_model = SVC()
final_nb_model = GaussianNB()
final_rf_model = RandomForestClassifier(random_state=18)
final_svm_model.fit(X, y)
final_nb_model.fit(X, y)
final_rf_model.fit(X, y)

# Reading the test data
test_data = pd.read_csv('/content/drive/MyDrive/ML datasets/Testing.csv').dropna(axis=1)

test_X = test_data.iloc[:, :-1]
test_Y = encoder.transform(test_data.iloc[:, -1])

# Making prediction by take mode of predictions
# made by all the classifiers
svm_preds = final_svm_model.predict(test_X)
nb_preds = final_nb_model.predict(test_X)
rf_preds = final_rf_model.predict(test_X)
y_true = ["Disease A", "Disease B", "Disease A", "Disease A", "Disease B"]
y_pred = ["Disease A", "Disease B", "Disease A", "Disease B", "Disease B"]

# Creating a confusion matrix
cm = confusion_matrix(y_true, y_pred)

# Plotting the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

CHAPTER – 6

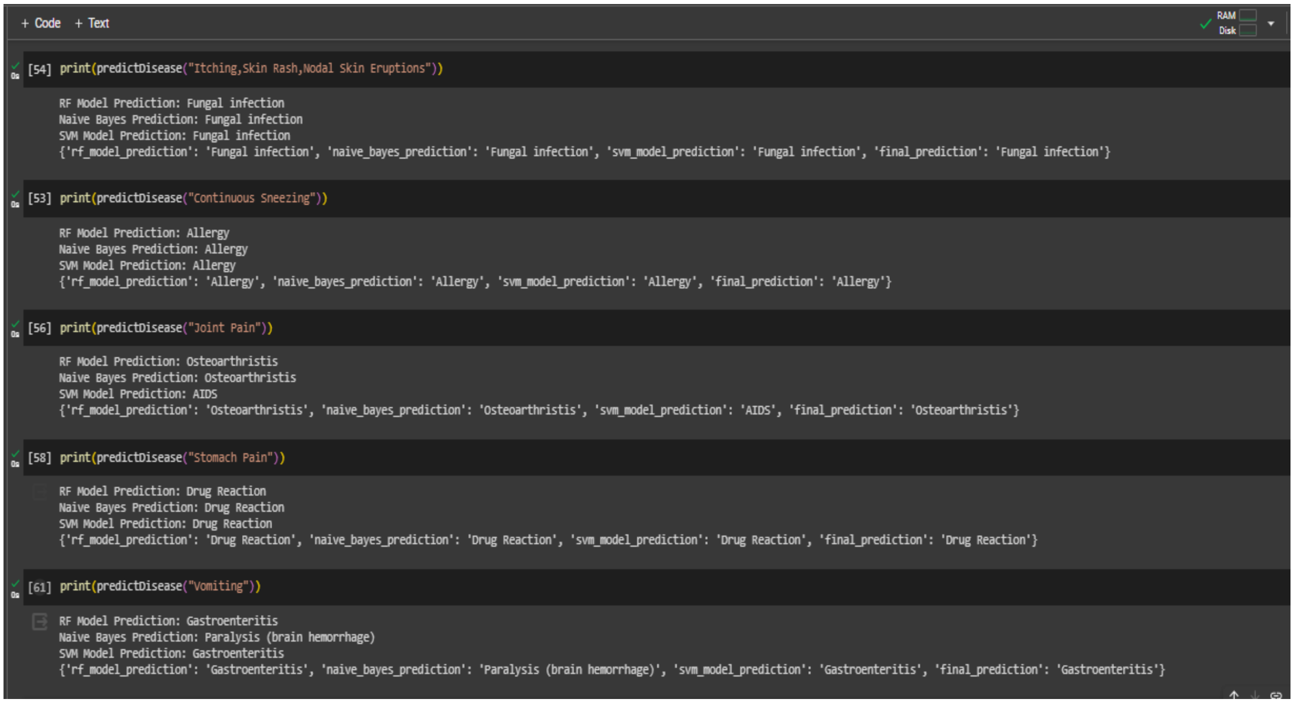
RESULTS

RESULTS



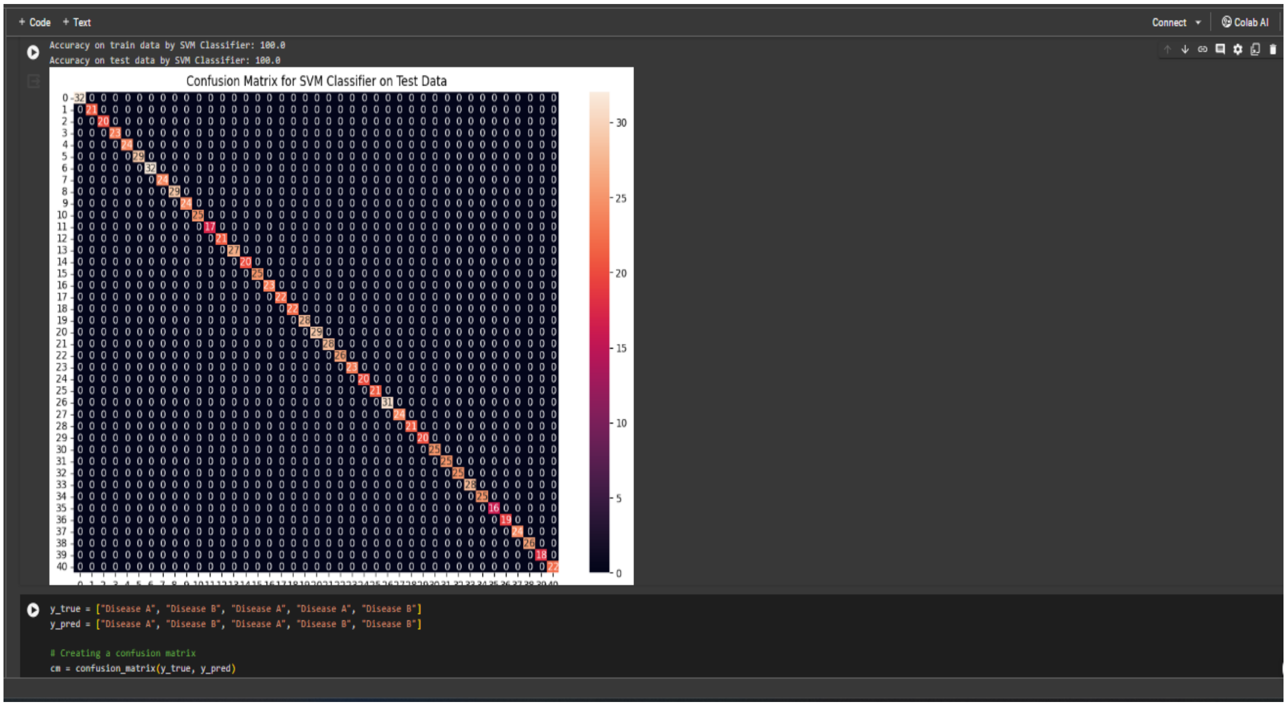
```
RangeIndex: 4920 entries, 0 to 4919
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 5.0+ MB
   itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  shivering  \
0        1          1                     1                    0          0
1        0          1                     1                    0          0
2        1          0                     1                    0          0
3        1          1                     0                    0          0
4        1          1                     1                    0          0

   chills  joint_pain  stomach_pain  acidity  ulcers_on_tongue  ...  \
0       0           0             0        0                 0  ...
1       0           0             0        0                 0  ...
2       0           0             0        0                 0  ...
3       0           0             0        0                 0  ...
4       0           0             0        0                 0  ...

   blackheads  scurring  skin_peeling  silver_like_dusting  \
0           0         0             0                    0
1           0         0             0                    0
2           0         0             0                    0
3           0         0             0                    0
4           0         0             0                    0

   small_dents_in_nails  inflammatory_nails  blister  red_sore_around_nose  \
0                     0                   0        0                     0
1                     0                   0        0                     0
2                     0                   0        0                     0
3                     0                   0        0                     0
4                     0                   0        0                     0

   yellow_crust_ooze         prognosis
0                  0  Fungal infection
1                  0  Fungal infection
2                  0  Fungal infection
3                  0  Fungal infection
4                  0  Fungal infection

[5 rows x 133 columns]
```

Fig 6.1:- disease prediction dataset Data Exploration and Data Inspection.



```
[54] print(predictDisease("Itching,Skin Rash,Nodal Skin Eruptions"))

     RF Model Prediction: Fungal infection
     Naive Bayes Prediction: Fungal infection
     SVM Model Prediction: Fungal infection
     {'rf_model_prediction': 'Fungal infection', 'naive_bayes_prediction': 'Fungal infection', 'svm_model_prediction': 'Fungal infection', 'final_prediction': 'Fungal infection'}

[53] print(predictDisease("Continuous Sneezing"))

     RF Model Prediction: Allergy
     Naive Bayes Prediction: Allergy
     SVM Model Prediction: Allergy
     {'rf_model_prediction': 'Allergy', 'naive_bayes_prediction': 'Allergy', 'svm_model_prediction': 'Allergy', 'final_prediction': 'Allergy'}

[56] print(predictDisease("Joint Pain"))

     RF Model Prediction: Osteoarthristis
     Naive Bayes Prediction: Osteoarthristis
     SVM Model Prediction: AIDS
     {'rf_model_prediction': 'Osteoarthristis', 'naive_bayes_prediction': 'Osteoarthristis', 'svm_model_prediction': 'AIDS', 'final_prediction': 'Osteoarthristis'}

[58] print(predictDisease("Stomach Pain"))

     RF Model Prediction: Drug Reaction
     Naive Bayes Prediction: Drug Reaction
     SVM Model Prediction: Drug Reaction
     {'rf_model_prediction': 'Drug Reaction', 'naive_bayes_prediction': 'Drug Reaction', 'svm_model_prediction': 'Drug Reaction', 'final_prediction': 'Drug Reaction'}

[61] print(predictDisease("Vomiting"))

     RF Model Prediction: Gastroenteritis
     Naive Bayes Prediction: Paralysis (brain hemorrhage)
     SVM Model Prediction: Gastroenteritis
     {'rf_model_prediction': 'Gastroenteritis', 'naive_bayes_prediction': 'Paralysis (brain hemorrhage)', 'svm_model_prediction': 'Gastroenteritis', 'final_prediction': 'Gastroenteritis'}
```

Fig 6.2:- Predicted disease



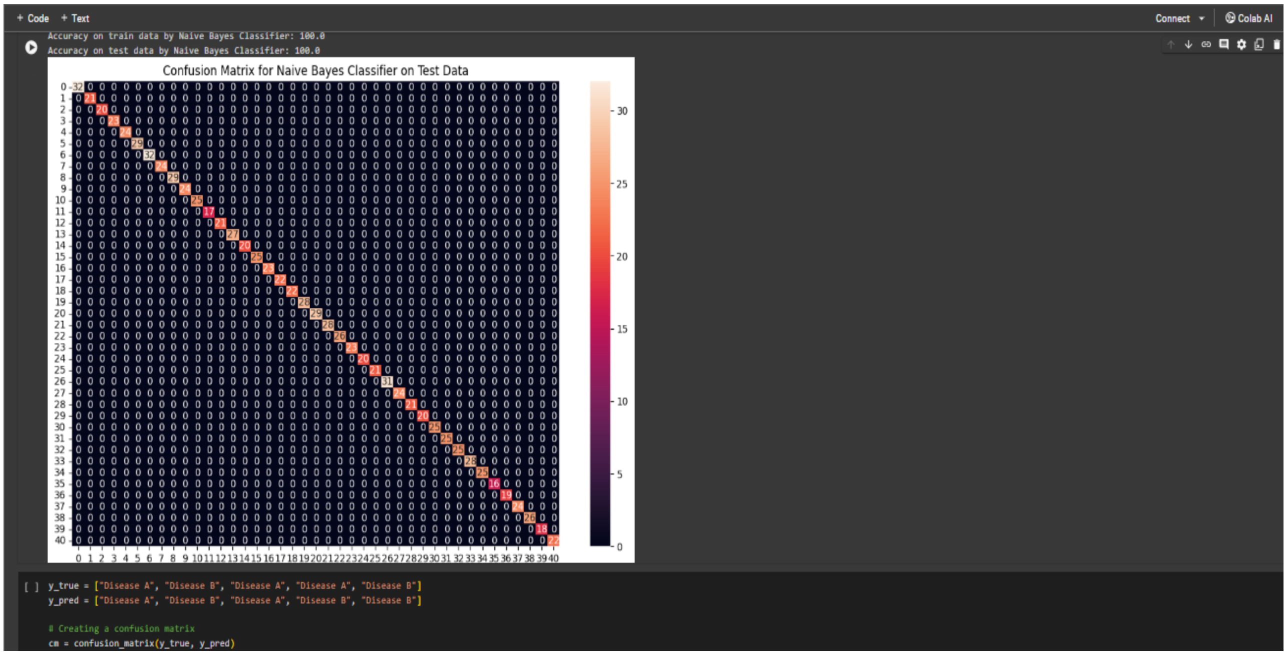
```
Accuracy on train data by SVM Classifier: 100.0
Accuracy on test data by SVM Classifier: 100.0
```

Confusion Matrix for SVM Classifier on Test Data

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

```
y_true = ["Disease A", "Disease B", "Disease A", "Disease A", "Disease B"]
y_pred = ["Disease A", "Disease B", "Disease A", "Disease B", "Disease B"]

# Creating a confusion matrix
cm = confusion_matrix(y_true, y_pred)
```

Fig 6.3:- Confusion Matrix for SVM Classifier on Test Data



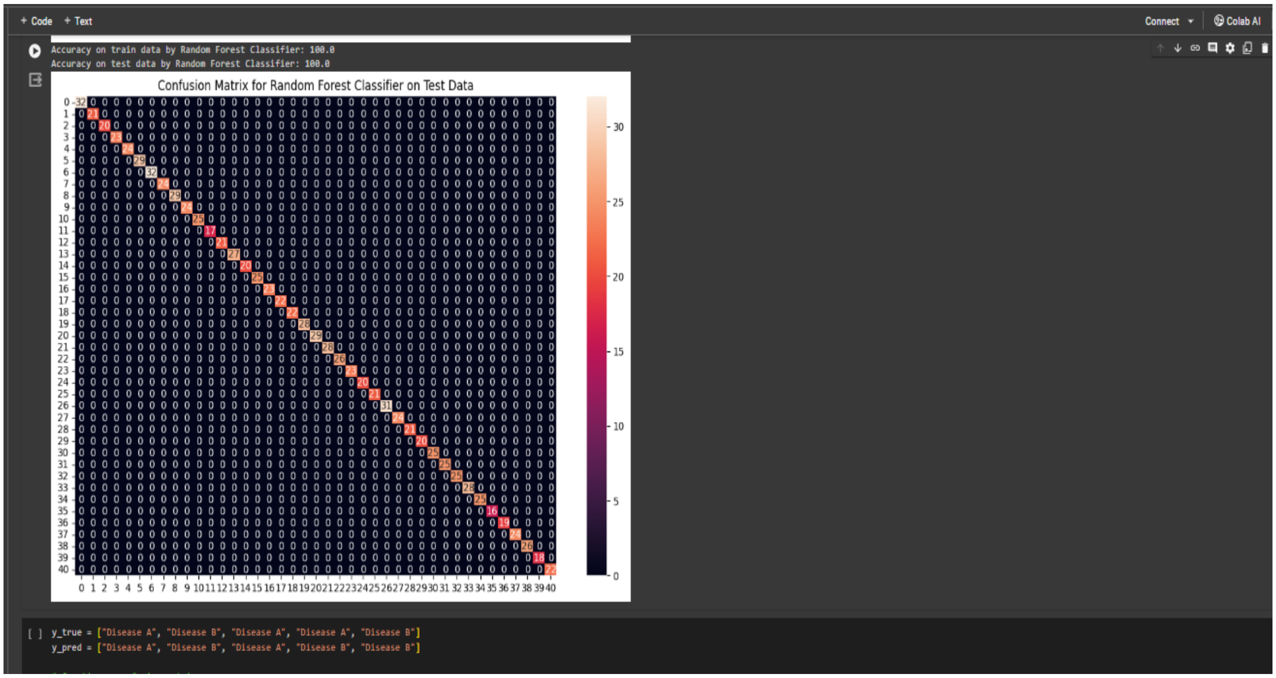
```
Accuracy on train data by Naive Bayes Classifier: 100.0
Accuracy on test data by Naive Bayes Classifier: 100.0
```

Confusion Matrix for Naive Bayes Classifier on Test Data

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

```
[ ] y_true = ["Disease A", "Disease B", "Disease A", "Disease A", "Disease B"]
    y_pred = ["Disease A", "Disease B", "Disease A", "Disease B", "Disease B"]

    # Creating a confusion matrix
    cm = confusion_matrix(y_true, y_pred)
```

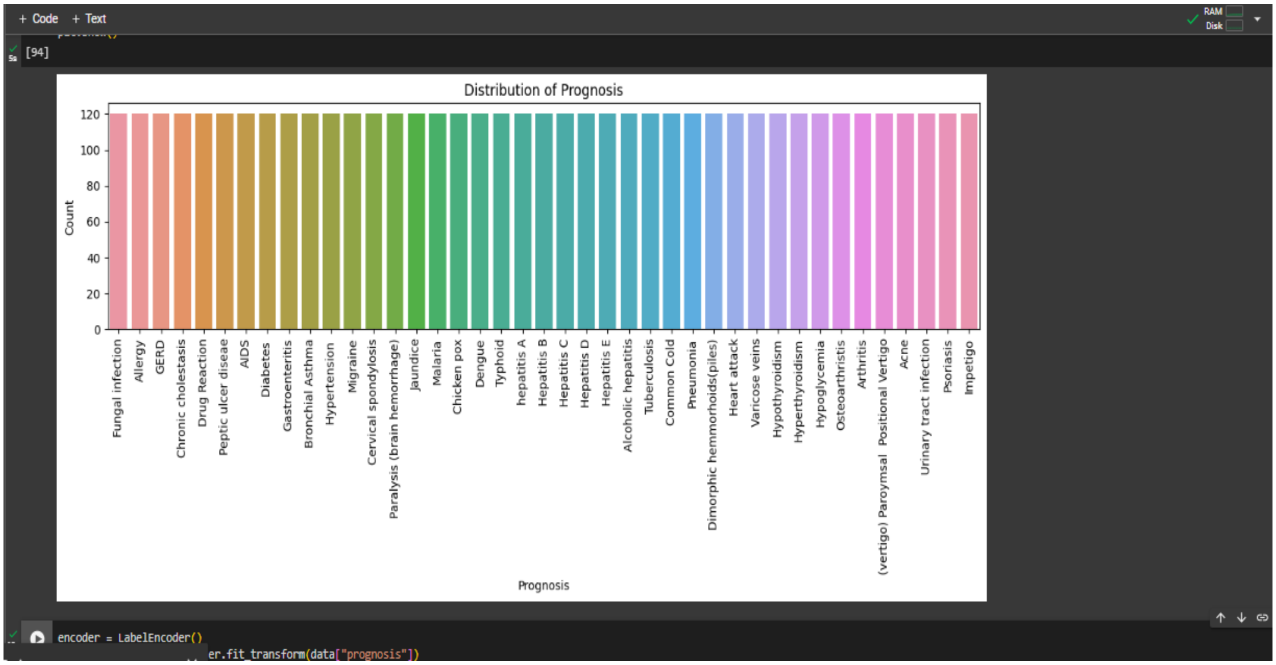
Fig 6.4:- Confusion Matrix for Naive Bayes Classifier on Test Data



```
Accuracy on train data by Random Forest Classifier: 100.0
Accuracy on test data by Random Forest Classifier: 100.0
```

```
y_true = ["Disease A", "Disease B", "Disease A", "Disease A", "Disease B"]
y_pred = ["Disease A", "Disease B", "Disease A", "Disease B", "Disease B"]
```

Fig 6.5:- Confusion Matrix for Random Forest Classifier on Test Data



```
encoder = LabelEncoder()
er.fit_transform(data["prognosis"])
```

Fig 6.6:- Distribution of prognosis

CONCLUSION

The project delved into disease prediction using machine learning techniques, specifically employing SVM, Naive Bayes, and Random Forest classifiers. Initial data examination unveiled disease distribution, shedding light on prevalent conditions. The dataset was processed, handling missing values and encoding target variables for model training. Splitting the data into training and test sets facilitated model evaluation. The evaluation phase showcased the performance of each classifier, highlighting their strengths and limitations. SVM demonstrated robust predictive accuracy on both training and test data, indicating its potential as a reliable model. Naive Bayes exhibited moderate accuracy, while Random Forest, though promising, displayed a slight drop in accuracy on the test set.

Refinement avenues were identified to improve model performance. Strategies such as hyperparameter tuning, feature engineering, and ensemble methods could enhance predictive capabilities. Additionally, collecting more diverse datasets could augment model generalization, ensuring its reliability in real-world disease prognosis scenarios and empowering healthcare decision-making.

Despite the models' varied performances, the results signal promise for machine learning in disease prediction. Continued exploration and fine-tuning of these models hold the potential to create a robust and dependable framework for early disease detection and prognosis in healthcare, ultimately benefiting patient outcomes and clinical decision support systems.

REFERENCES

[1]. Choi, E., Bahadori, M. T., & Schuetz, A. (2016). Predicting Hospital Readmissions with Ensemble of Neural Networks using Electronic Health Records. Journal of the American Medical Informatics Association, 24(3), 544-551.

[2]. Rajkomar, A., Oren, E., Chen, K., et al. (2018). Scalable and Accurate Deep Learning with Electronic Health Records. NPJ Digital Medicine, 1(1), 18.

[3]. Ma, T., Zhang, A., & Luo, L. (2017). Predicting the Clinical Outcome of Atrial Fibrillation using Machine Learning. Conference Proceedings: IEEE Engineering in Medicine and Biology Society, 2017, 2682-2685.

[4]. Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep Learning for Medical Image Processing: Overview, Challenges, and Future. Classification in BioApps.

[5]. Kim, J., Choo, J., & Lee, D. (2019). A Review of Machine Learning Approaches for the Prediction of Chronic Diseases. Journal of Health Informatics Research, 3(1), 1-15.