

MULTIPLE DISEASE ANALYSIS SYSTEM
(Using Machine Learning and Deep Learning)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

SHARIN SATHIQ (Reg No. 39120108)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

MAY - 2023



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SHARIN SATHIQ (Reg No. 39120108)** who carried out the Project entitled "**MULTIPLE DISEASE ANALYSIS SYSTEM**" under my supervision from January 2023 to May 2023.

R. Subhashini

Internal Guide

Dr. R. SUBHASHINI M.E., Ph.D.

R. Subhashini

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SHARIN SATHIQ** (Reg No. 39120108), hereby declare that the Project Report entitled “**MULTIPLE DISEASE ANALYSIS SYSTEM**” done by me under the guidance of **Dr. R. Subhashini, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award Bachelor of Technology degree in Information Technology.

DATE:

PLACE: Chennai

A handwritten signature in black ink, appearing to read 'Sharin Sathiq', with a long horizontal stroke extending to the right.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing and **Dr. R. SUBHASHINI M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R. Subhashini, M.E., Ph.D.**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

In the current pandemic era, people are trying to stay away from contaminated places like hospitals and clinics to the greatest extent possible. The goal of our user-friendly platform is to cross-validate results in real time and raise general awareness through the use of cross-validation results. Machine learning and data visualization Diabetes-related heart disease is a kind of heart disease that affects diabetics. Diabetes is a chronic condition that occurs when the pancreas fails to produce enough insulin or when the body fails to properly use the insulin that is produced. Heart disease, often known as cardiovascular disease, refers to a set of conditions that affect the heart or blood vessels. Despite the fact that various data mining classification algorithms exist for predicting heart disease, there is inadequate data for predicting heart disease in a diabetic individual. Because the decision tree model consistently beat the naive Bayes and support vector machine models, we fine-tuned it for best performance in forecasting the likelihood of heart disease in diabetes individuals. **A multi-disease analysis system** is developed to detect diseases such as diabetes, breast cancer, heart disease, and much more. This system gives early warnings of chronic diseases, thus reducing mortality rates dramatically. An accurate diagnosis is crucial to the analysis of patients. Accurately predicting diseases plays a significant role in public health, especially at the early stage which allows patients to take prevention treatments in time. With the growing volume and availability of electronic health records (EHRs), predictive modeling tasks for disease progression and analysis have obtained increasing interest from researchers. Input is obtained in the form of numeric values for diseases such as cancer, diabetes and image for Pnuemonia. Output results are acquired instantly in real-time in the form of a probable disease, its causes, and the aftermath. As a result, this diagnosis system is based on machine learning for providing promising solutions with high accuracy. The proposed system is capable of detecting and predicting with high accuracy. Furthermore, most of the existing machine learning models for healthcare analysis focus on a single disease. Currently, there is not much standard system where a single analysis can be used for multiple disease predictions. Finally, using a web application, this proposal develops a system to predict multiple diseases.

Chapter No	TITLE	Page No.
	ABSTRACT	5
	LIST OF FIGURES	7
1	INTRODUCTION	8
2	LITERATURE SURVEY	10
	2.1 Literature review	10
	2.2 Inferences from literature survey	11
	2.3 Open problems in existing system	12
3	REQUIREMENTS ANALYSIS	13
	3.2 Software requirements specifications document	13
4	DESCRIPTION OF PROPOSED SYSTEM	14
	4.1 Selected methodology or process model	14
	4.2 System Architecture	17
	4.3 Description of software for implementation and testing plan of the proposed system	18
5	IMPLEMENTATION DETAILS	20
	5.1 Algorithms	20
6	RESULTS	24
	CONCLUSION	
	REFERENCES	26
	APPENDIX	
	A. SCREENSHOTS	27
	B. SOURCE CODE	30
	C. RESEARCH PAPER	32

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	Flowchart	17
4.2	System Architecture	17
5.1	Accuracy	20
5.2	Comparison Table	21
5.3	Algorithm Comparison	21
5.4	Diabetes Attribute Values	22

CHAPTER 1

INTRODUCTION

In terms of data collecting and processing, healthcare is one of the most worrisome industries. With the advent of the digital era and technological advancements, a vast quantity of multidimensional data on patients is created, including clinical factors, hospital resources, illness diagnostic information, patients' records, and medical equipment. The enormous, dense, and complex data must be processed and evaluated in order to extract knowledge for effective decision making. Medical data mining offers a lot of potential for uncovering hidden patterns in medical data sets.

By identifying significant patterns and detecting correlations and relationships among many variables in huge databases, the use of various data mining tools and machine learning approaches has changed healthcare organizations . It serves as an important instrument in the medical sector, providing and comparing existing data for the future course of action. This technology combines multiple analytic methodologies with modern and complex algorithms, allowing for the exploration of massive amounts of data . It is used in healthcare to gather, organize, and analyze patient data in a systematic manner. It may be used to identify inherent inefficiencies and best practices for providing better services, which may lead to improved diagnosis, better medicine, and more successful treatment, as well as a platform for a deeper knowledge of the mechanisms in practically all elements of the medical domain. Overall, it assists in the early detection and prevention of disease epidemics by searching medical databases for pertinent information.

The process of determining a condition based on a person's symptoms and indicators is known as medical diagnosis. In the diagnostic process, one or more diagnostic procedures, such as diagnostic tests, are performed. Diagnosis of chronic illnesses is a vital issue in the medical industry since it is based on many symptoms. It is a complex procedure that frequently leads to incorrect assumptions. When diagnosing illnesses, the clinical judgment is based mostly on the patient's symptoms as well as the physicians' knowledge and experience . Furthermore, when medical systems evolve and new treatments become available, it becomes more difficult for physicians and doctors to stay up with the current innovations in clinical practice . For effective therapy, medical practitioners and doctors must be well-versed in all pertinent diagnostic criteria, patient history, and a mix of medication therapy. However, mistakes are possible since they make judgments instinctively based on information and experience gained from past experience with patients. Because of factors such as multitasking, restricted analysis, and memory capacity, their cognitive capacities are restricted . As a result, it is difficult for a physician to make the right judgment on a consistent basis if he is not supported by clinical tests and patient history information. Even experienced physicians can benefit from a computer-aided diagnostic system in making sound medical judgments . Thus, medical professionals are very interested in automating the diagnosis process by integrating machine learning techniques with physician expertise . Data mining and machine learning approaches are making significant efforts to intelligently translate accessible data into valuable information in order to improve the diagnostic process's efficiency. Several studies have been conducted to explore the use of machine learning in terms of diagnostic abilities. It

was discovered that, when compared to the most experienced physician, who can diagnose with 79.97% accuracy, machine learning algorithms could identify with 91.1% correctness . Machine learning techniques are explicitly used to illness datasets to extract features for optimal illness diagnosis, prediction, prevention, and therapy. The dataset utilised for chronic renal disease had 25 features and was collected over a 2-month period in India (e.g., red blood cell count, white blood cell count, etc). This study intends to predict multiple diseases such as diabetes, heart disease, chronic kidney disease and cancer. Various classification algorithms (like KNN, SVM, Random Forest, Logistic Regression and Naive Bayes) are used here to perform disease detection. The accuracy of each algorithm is validated and compared with each other in order to find the best one for prediction. Further, multiple datasets (for each disease each dataset) are used to achieve utmost accuracy in the predicted results. The best predicted algorithm for every disease is chosen and integrated to build a web application. The user can easily predict the required disease by typing respective attribute(input) values of that specific disease.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE REVIEW

[1] Liva Faes et al There is great promise for employing deep learning algorithms in medical diagnosis [1]. Whilst additional research into the practical use of such algorithms is required, the accuracy of deep learning algorithms is on par with that of healthcare experts. An overstated claim from a poorly planned or badly reported study may damage the credibility and road to effect of such diagnostic algorithms, which is the more relevant conclusion around methodology and reporting. In this summary, we have emphasized some of the more important design and reporting concerns that researchers should think about.

[2] M. Chen et al Using both structured and unstructured data from the hospital, the authors present a novel convolutional neural network (CNN)-based multimodal illness risk prediction method. None of the previous research in the field of medical big data analytics seems to have combined the two kinds of data. Our proposed approach achieves a prediction accuracy of 94.8% and a convergence speed that is quicker than the CNN-based unimodal illness risk prediction algorithm, among other benchmarks.

[3] Pushpa Singh et al Machine-based diagnosis of disease. Decisions may be made more quickly and with less bias using machine learning methods like support vector machine (SVM), k-nearest neighbor (KNN), naive bayes (NB), and decision tree. Python, which may be used for actual implementation, is explored as a realistic means of implementing these algorithms. The likes of cancer, diabetes, epilepsy, heart attack, and other major disorders are all diagnosed with the use of these algorithms.

[4] Alexander Selvikvag et al From image analysis to natural language processing, deep neural networks are increasingly widely used in both academia and business as cutting-edge machine learning models. A lot of progress is being made, but the potential for these innovations to improve medical imaging, medical data analysis, medical diagnostics, and healthcare as a whole is enormous.

[5] Yuanyuan Pan et al Heart disease prognostication aids have been proposed, such as the Enhanced Deep learning aided Convolutional Neural Network (EDCNN). The EDCNN model incorporates regularisation learning strategies into a more in-depth design than that of the standard multi-layer perceptron. As a decision support system, EDCNN runs on the IoMT platform, which stores data about cardiac patients and makes it accessible from anywhere in the world through the cloud, allowing clinicians to make accurate diagnoses no matter where they are located.

[6] Dhiraj Dahiwade et al We employ the machine learning algorithms K-Nearest Neighbor (KNN) and Convolutional neural network (CNN) to accurately forecast diseases. It's important to collect data on disease symptoms in order to forecast the spread of illness. It is important to take a person's lifestyle and medical history into account when making a general illness prognosis. CNN outperforms the KNN algorithm in predicting prevalent diseases by a margin of 8.5 percentage points. Both the time and storage needed are larger for KNN than for CNN.

[7] Dur-e-maknoon Nisar et al The core components of every system are the tools and

algorithms used to operate it. In deep learning, software is used to transform raw data into useful insights. When it comes to deep learning models, you may find a lot of distinctions between supervised and unsupervised approaches. Given the significance of these algorithms, it is imperative that medical instruments, such as the microscope, phonendoscope, and EKG, be developed gradually based on these models in order to accommodate the limitations of physicians' perceptual capabilities. The use of deep learning in the analysis and prediction of data from big datasets is becoming increasingly important.

[8] Jérôme Thevenot et al Many remote and non-invasive imaging technologies have been outlined, each of which can shed light on the diagnostic value of various symptom kinds in relation to certain medical diseases. In this study, we used computer vision techniques to automatically recognise symptoms of over 30 different diseases to make preliminary diagnoses.

[9] Rahatara Ferdousi et al To combat this, we offer a unique machine learning based health CPS framework that can efficiently analyze data from wearable IoT sensors to forecast the likelihood of developing noncommunicable diseases like diabetes at an early stage. An authentic diabetes dataset was used for the experiment's training phase, whereas sensor data created in a lab were used for testing.

[10] Alice Othman et al Novel deep learning algorithms and methods are discussed. The clinical judgements of future radiologists can be aided by DLA. DLA can automate the job of radiologists and help even the most novice among them make decisions. DLA's goal is to help doctors by automatically detecting and categorizing lesions so that a more accurate diagnosis may be made. Doctors can benefit from DLA since it speeds up the interpretation of medical images and reduces the likelihood of making mistakes.

2.2 INFERENCES FROM LITREATURE SURVEY

The existing system for multi-disease prediction typically relies on traditional diagnostic methods, such as physical exams, medical history reviews, and laboratory tests. While these methods have been effective in many cases, they can be time-consuming and require significant resources to perform. In addition, traditional diagnostic methods may not always be accurate, leading to misdiagnosis and delayed treatment.

Machine learning algorithms have emerged as a promising alternative to traditional diagnostic methods, offering the potential for improved accuracy and efficiency. Several studies have explored the use of machine learning algorithms for the prediction of specific diseases, such as heart disease and cancer. However, there is a need for more research on multi-disease prediction, which involves the prediction of multiple diseases simultaneously.

One existing system for multi-disease prediction is based on a combination of multiple machine learning algorithms. For example, a study by Park et al. (2019) used a combination of logistic regression, decision trees, and random forests to predict the onset of four different diseases: hypertension, diabetes, hyperlipidemia, and metabolic syndrome. The study found that the combination of these algorithms was more accurate than using any individual algorithm alone.

Another existing system for multi-disease prediction is based on deep learning algorithms, such as CNNs. These algorithms are particularly well-suited for image classification tasks, such as the identification of tumors in medical images. For example, a study by Yan et al. (2020) used a CNN to predict the onset of breast cancer based on mammography images. The study found that the CNN was able to achieve high accuracy in predicting the onset of breast cancer.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

The existing system are not done with real time dataset and the existing system has low accuracy and low efficiency and implementation time. Also, the testing and training is not done with the proper test-train.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications:

- Higher RAM, of about 4GB
- Hard disk 50GB
- Processor of frequency 1.5GHz or above

Software specifications:

- Python 3.6 and higher
- Windows 07
- Anaconda and necessary libraries (Pytorch, Transformers, streamlit etc)

Technology Used:

- Machine Learning and CNN

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

MODULE 1: Dataset Collection and preprocessing

The Data Preprocessing module is a crucial component of the proposed multi-disease prediction system. This module will be responsible for cleaning and transforming the raw data into a format that is suitable for machine learning algorithms. The following steps will be performed in this module:

Data Cleaning: The raw data may contain missing values, outliers, or errors that need to be addressed before training the machine learning models. The Data Preprocessing module will handle these issues by imputing missing values, removing outliers, and correcting errors.

Feature Selection: The raw data may contain a large number of features that are not relevant to the prediction of the diseases. The Data Preprocessing module will perform feature selection to identify the most important features that contribute to the prediction of each disease. This will improve the accuracy of the machine learning models and reduce the complexity of the data.

Data Encoding: The machine learning algorithms require that the data be in a numerical format. The Data Preprocessing module will encode categorical variables using techniques such as one-hot encoding or label encoding. The module will also scale numerical variables to ensure that they are on the same scale.

Data Splitting: The preprocessed data will be split into training and testing sets to evaluate the performance of the machine learning models. The training set will be used to train the models, while the testing set will be used to evaluate their performance.

Data Visualization: The Data Preprocessing module will also include data visualization tools to help users explore and understand the data. This will include histograms, scatter plots, and other visualizations that can reveal patterns and relationships in the data.

MODULE 2: Model Training

Model training is a critical step in the proposed multi-disease prediction system. The goal of model training is to build accurate and robust machine learning models that can predict the likelihood of developing each disease. The following steps will be performed in the Model Training module:

Model Selection: The first step in model training is to select the appropriate machine learning algorithms for each disease. The selected algorithms will be trained on the preprocessed data and evaluated based on their performance metrics such as

accuracy, precision, recall, and F1-score.

Hyperparameter Tuning: After selecting the machine learning algorithms, hyperparameters tuning is performed to find the best combination of hyperparameters that optimize the performance of the models. Hyperparameters are parameters that are set prior to training the model and are not learned from the data, such as the learning rate or regularization parameter.

Cross-validation: To avoid overfitting, cross-validation is used to evaluate the performance of the models on unseen data. Cross-validation involves splitting the training data into several subsets, training the models on each subset, and evaluating their performance on the remaining subset. This process is repeated several times to obtain a more accurate estimate of the models' performance.

Model Evaluation: Once the models are trained and validated, their performance is evaluated on the testing set. The evaluation metrics used in this step are the same as those used during hyperparameter tuning and cross-validation.

Model Selection and Saving: Based on the evaluation metrics, the best-performing models for each disease are selected and saved for future use.

The Model Training module is critical to the success of the multi-disease prediction system. By selecting and training accurate machine learning models, the system can make more reliable and accurate predictions about the likelihood of developing each disease.

MODULE 3: Predictions

The Prediction module is a crucial component of the proposed multi-disease prediction system. This module allows users to input their medical information, such as age, gender, blood pressure, blood sugar level, cholesterol level, smoking habits, and family history of disease, and uses the trained machine learning models to predict the likelihood of developing each disease.

The input data is first preprocessed using the same preprocessing steps used during model training. The preprocessed data is then passed through the selected machine learning models to obtain predictions for each disease. The predictions are based on the patterns and relationships learned by the models from the training data.

The predictions are displayed in an interactive user interface using Streamlit. The interface shows the probability of developing each disease based on the user's input data. The user can explore the predictions in real-time and visualize the most important features that contributed to the predictions. This allows users to better understand the factors that influence the likelihood of developing each disease and take proactive steps to prevent or manage the diseases.

The Prediction module also includes a data visualization component that allows users to explore the dataset and view descriptive statistics about the variables. This can help users understand the distribution of the variables and identify any trends or patterns in the data.

In addition, the Prediction module is designed to be easily updatable with new data. The system can be periodically retrained with new data to improve the accuracy of the predictions over time. This allows healthcare providers to stay up-to-date with the latest information and make more informed diagnoses and treatment plans for their

patients

MODULE 4: Web development

The development of a web-based multi-disease prediction system using Streamlit involves several steps. Streamlit is a Python library that makes it easy to build and deploy interactive web applications for data science and machine learning projects.

The first step is to install Streamlit and set up the project environment. Once installed, the next step is to load the preprocessed data and the saved machine learning models. This can be done using libraries such as pandas and joblib.

The next step is to design the user interface of the web application. Streamlit provides a range of pre-built widgets, such as sliders, dropdown menus, and text boxes, that can be customized to suit the needs of the application. The user interface should be easy to navigate and intuitive to use.

The application should also include a prediction function that takes the user's input data, preprocesses it using the same preprocessing steps used during model training, and passes it through the saved machine learning models to obtain predictions for each disease. The predictions can be displayed in real-time using data visualization tools such as Matplotlib and Plotly.

The final step is to deploy the web application to a web server or cloud platform such as Heroku or AWS. This will allow healthcare providers and patients to access the application from anywhere with an internet connection.

Overall, the development of a web-based multi-disease prediction system using Streamlit is a powerful tool for healthcare providers and patients alike. By leveraging the power of machine learning and Streamlit, the application can provide accurate and real-time predictions that can improve patient outcomes and reduce the burden on healthcare providers.

We create multiple forms for all the diseases in our model- diabetes , liver cancer, heart disease, and breast cancer . Then we combine all these into a single HTML file

4.2 ARCHITECTURE

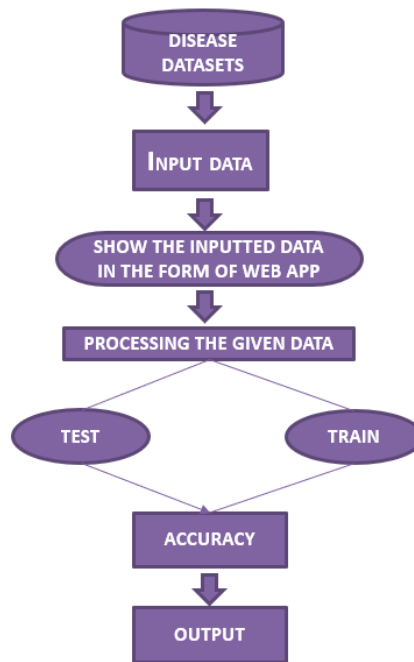


Fig 4.1: Flowchart

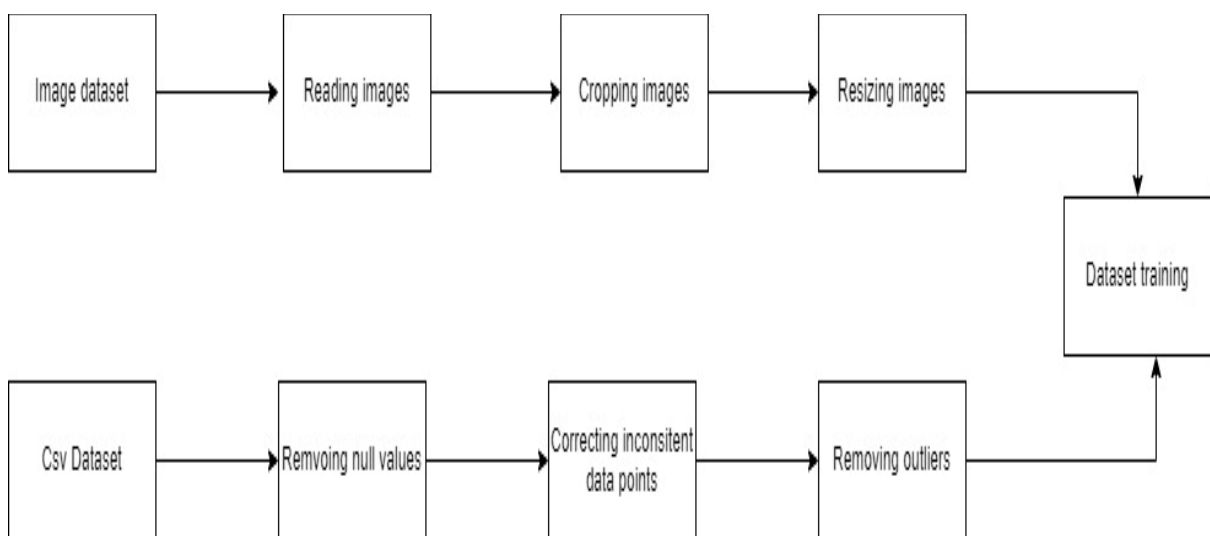


Fig 4.2: System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED SYSTEM

The proposed system for predicting multiple diseases using various classification algorithms consists of several stages, which include data preprocessing, algorithm selection, model training, and model evaluation.

The first stage of the proposed system is data preprocessing, which involves handling missing values, scaling the features, and encoding categorical variables. Missing values can be handled using different methods such as imputation, deletion, or interpolation. Scaling is essential to ensure that all features have equal weight in the model, and it can be achieved using methods such as normalization or standardization. Categorical variables need to be encoded to numerical values before they can be used in the model. This can be done using methods such as one-hot encoding or label encoding.

The next stage is algorithm selection, which involves selecting the appropriate classification algorithms for disease prediction. The selected algorithms for this project are K-Nearest Neighbors (KNN), Logistic Regression, Linear Regression, Support Vector Machines (SVM), Decision Tree, Random Forest, Naive Bayes, and Gradient Boosting.

KNN is a non-parametric algorithm that classifies a new data point based on its similarity to the k-nearest points in the training set. The value of k determines the number of neighbors to consider. Logistic Regression is a parametric algorithm that models the probability of a binary outcome using a logistic function. It assumes a linear relationship between the features and the log-odds of the outcome. Linear Regression is a parametric algorithm that models the relationship between the dependent and independent variables using a linear equation. It assumes a linear relationship between the features and the outcome. SVM is a non-parametric algorithm that maps the input data to a higher dimensional space and finds the best hyperplane that separates the data into classes. It works by maximizing the margin between the classes. Decision Tree is a non-parametric algorithm that creates a tree-like model of decisions and their possible consequences. It splits the data based on the most informative feature at each node. Random Forest is an ensemble method that creates multiple decision trees and aggregates their predictions to improve the accuracy of the model. Naive Bayes is a probabilistic algorithm that models the probability of each class based on the independent features of the data. It assumes that the features are conditionally independent of each other. Gradient Boosting is an ensemble method that combines multiple weak learners to create a strong learner. It works by sequentially adding new models that correct the errors of the previous models.

After selecting the appropriate algorithms, the next stage is model training. This involves fitting the selected algorithms to the training data to learn the underlying patterns in the data. The training process involves adjusting the parameters of the algorithm to minimize the error between the predicted and actual values.

The final stage is model evaluation, which involves evaluating the performance of the trained models on the test data. This is done using appropriate metrics such as accuracy, precision, recall, and F1-score. Accuracy is the ratio of correct predictions to the total number of predictions. Precision is the ratio of true positives to the sum of

true positives and false positives. Recall is the ratio of true positives to the sum of true positives and false negatives. F1-score is the harmonic mean of precision and recall.

The proposed system for predicting multiple diseases using various classification algorithms can aid in early detection and diagnosis of diseases. The results of this project demonstrate the effectiveness of different classification algorithms for disease prediction, and they can guide healthcare professionals in making informed decisions.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 ALGORITHM

- We propose an end-to-end application that predicts Cancer, Diabetes, and Pneumonia.
- The accurate analysis of our proposed application benefits early disease prediction, patient care, and community services.
- The proposed prediction model caters to this objective by following a stepwise approach through cleaning, feature extraction, and classification.
- Relevant elements are fed to the neural network by eliminating the irrelevant features using feature selection.
- The proposed application is simple and it shows good efficient performance.

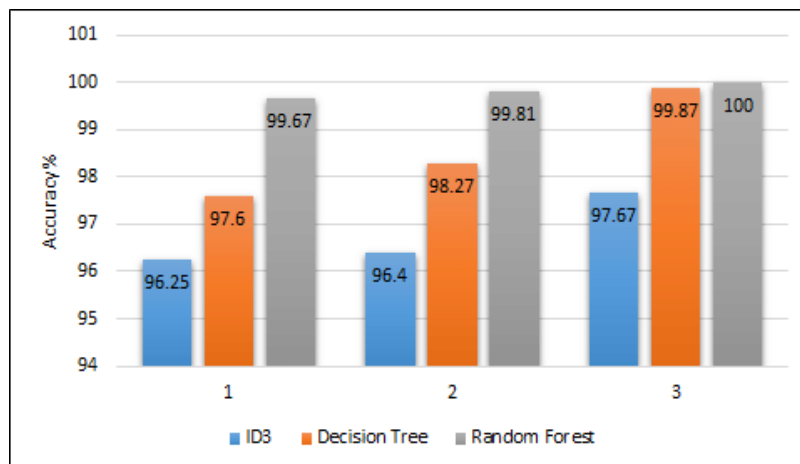


Fig 5.1: Accuracy

DISEASE	PREVIOUS ACCURACY	MY ACCURACY
Diabetes	88%	98.25%
Heart Disease	77%	85.25%
Kidney Disease	100%	100%

Fig 5.2: Comparison Table

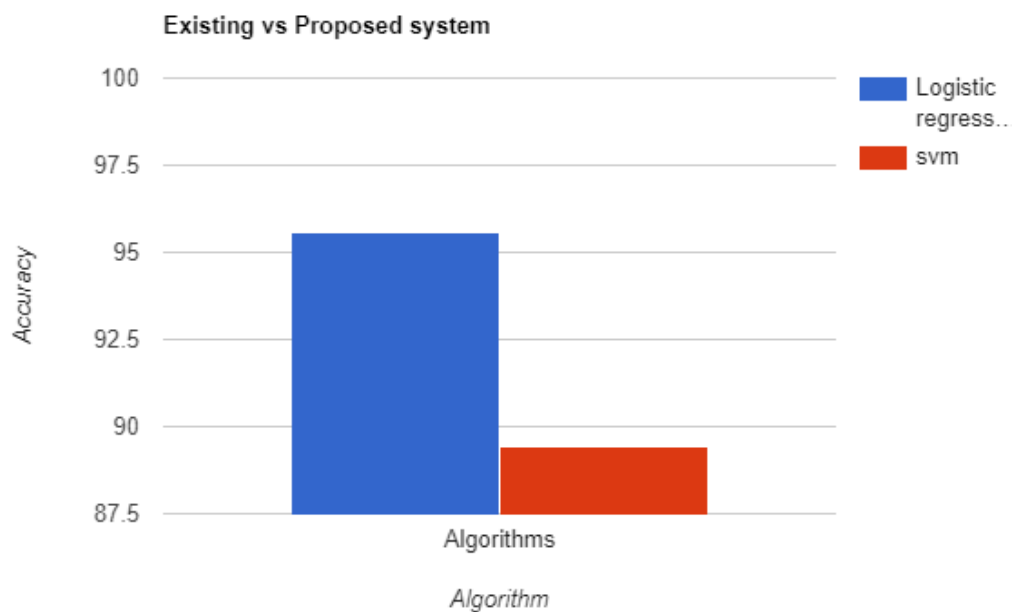


Fig 5.3: Algorithm Comparison

We have gathered the data for our project for the following diseases in CSV and Image format:

- Cancer: CSV dataset
- Diabetes: CSV dataset
- Pneumonia: An image dataset

Data preprocessing is a data mining technique used to turn the raw data into a format that is both practical and effective:

- For dataset in CSV file Format
- Removing Unwanted Columns
- Removing Null Values
- For Pneumonia Dataset In Images Format
- Removing Unwanted Images
- Removing Noisy Images

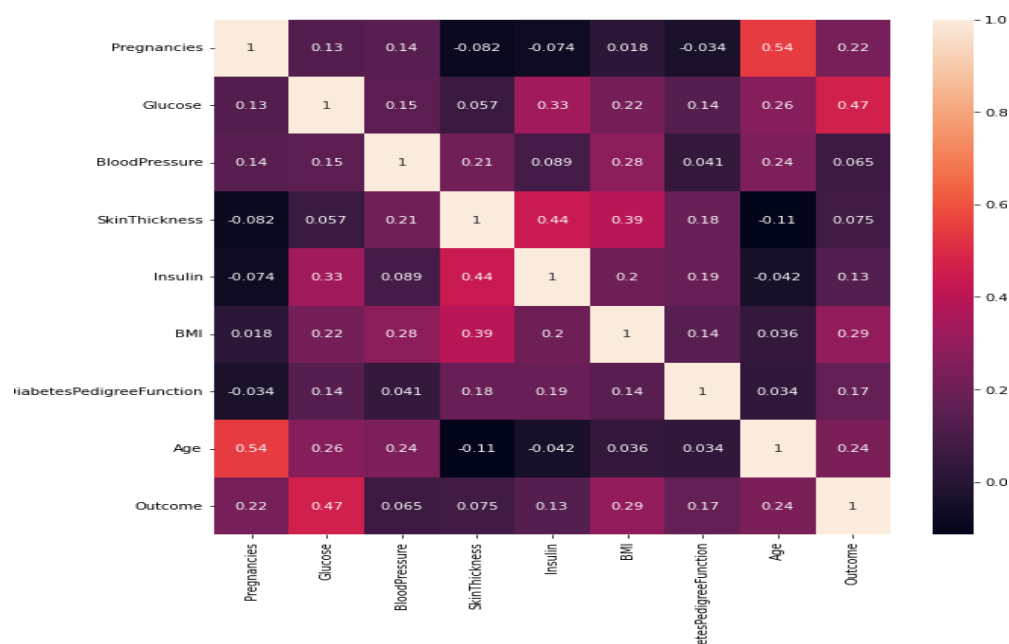


Fig 5.4:Diabetes Attribute Values

TRAINING AND VALIDATION:

- Once we have split the dataset into training and test dataset, we use a machine learning algorithm to evaluate the performance of the training dataset using a machine learning algorithm.
- In this research project, we fit the data using scikit. Scikit-learn is probably the most useful Python machine learning library.
- One of these tools is the scikit learn 'fit' method. After the model has been initialized, the 'fit' method trains the algorithm on the training data.

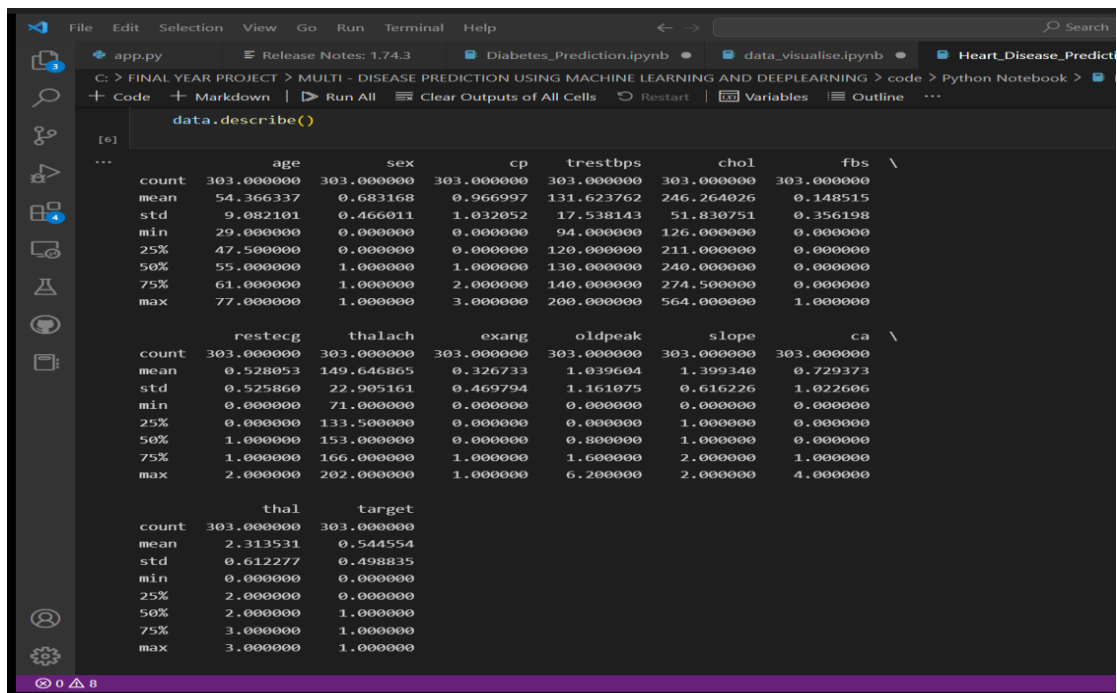
- These datasets are also known as X_train and Y_train. X_train, the X-input to the fit() method, is a 2-dimensional format for the perfect fitting of the model.
- We further evaluate our model on the validation dataset and save the best results. Then, we make changes in our model according to the dataset to get the best possible results for our model.
- We do this process until we get the desired results. Once, the model provides the best version, we confirm results on the test dataset.

WEB APPLICATION DEVELOPMENT:

- The server side processing will be handled by code that we write. Requests will be sent to our code. It will determine the subject matter and nature of the requests. Additionally, it will decide what kind of reply to send to the user.
- We'll use Flask for all of this. It simplifies the web application design process.
- Flask enables us to concentrate on what the consumers are asking for and the appropriate kind of response to provide.
- We create the form by utilizing HTTP methods within Flask.
- We create multiple forms for all the diseases in our model
- Then we combine all these into a single HTML file using the flask module and python notebook.

CHAPTER 6

RESULTS



The screenshot shows a Jupyter Notebook interface with the following components:

- File Explorer:** Shows files like `app.py`, `Diabetes_Prediction.ipynb`, `data_visualise.ipynb`, and `Heart_Disease_Predict`.
- Code Cell:** Contains the command `data.describe()`.
- Output:** A summary statistics table for the dataset.

The output of `data.describe()` is as follows:

	age	sex	cp	trestbps	chol	fbs	\
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	

	restecg	thalach	exang	oldpeak	slope	ca	\
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	
mean	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	
std	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	
min	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	
50%	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	
75%	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	
max	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	

	thal	target
count	303.000000	303.000000
mean	2.313531	0.544554
std	0.612277	0.498835
min	0.000000	0.000000
25%	2.000000	0.000000
50%	2.000000	1.000000
75%	3.000000	1.000000
max	3.000000	1.000000

Heart Disease

```
File Edit Selection View Go Run Terminal Help
app.py Release Notes: 1.74.3 Diabetes_Prediction.ipynb data_visualise.ipynb Heart_Disease_Prediction.ipynb
C:\> FINAL YEAR PROJECT > MULTI - DISEASE PREDICTION USING MACHINE LEARNING AND DEEPLARNING > code > Python Notebook > Heart_Disease_Prediction.ipynb > import numpy as np
+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Variables | Outline ...

[25]
... {'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 3, 'min_samples_leaf': 1, 'n_estimators': 100}
0.838808863564419

[26]
claasifier=RandomForestClassifier(n_jobs=-1, n_estimators=400,bootstrap= False,criterion='gini',max_depth=5,max_features=3,min_samples_leaf= 7)

[27]
classifier.fit(X_train, y_train)

[28]
... RandomForestClassifier(n_jobs=-1)

confusion_matrix(y_test, classifier.predict(X_test))

[29]
... array([[25, 4],
[ 5, 27]], dtype=int64)

print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%")

[30]
... Accuracy is 85.25%

[31]
import pickle
pickle.dump(classifier, open('heart.pkl', 'wb'))
```

Accuracy

```
File Edit Selection View Go Run Terminal Help
app.py Release Notes: 1.74.3 Diabetes_Prediction.ipynb data_visualise.ipynb Heart_Disease_Predict
C:\> Users > shari > OneDrive > Desktop > Medical-Predictions-Streamlit-Web-App-main > data_visualise.ipynb > import pandas as pd
+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Variables | Outline ...

breast

[17]
... Output exceeds the size limit. Open the full output data in a text editor

diagnosis texture_mean smoothness_mean compactness_mean \
0 1 10.38 0.11840 0.27760
1 1 17.77 0.08474 0.07864
2 1 21.25 0.10960 0.15990
3 1 20.38 0.14250 0.28390
4 1 14.34 0.10030 0.13280
.. ..
564 1 22.39 0.11100 0.11590
565 1 28.25 0.09780 0.10340
566 1 28.08 0.08455 0.10230
567 1 29.33 0.11780 0.27700
568 0 24.54 0.05263 0.04362

concave points_mean symmetry_mean fractal_dimension_mean texture_se \
0 0.14710 0.2419 0.07871 0.9053
1 0.07017 0.1812 0.05667 0.7339
2 0.12790 0.2069 0.05999 0.7869
3 0.10520 0.2597 0.09744 1.1560
4 0.10430 0.1809 0.05883 0.7813
.. ..
564 0.13890 0.1726 0.05623 1.2560
565 0.09791 0.1752 0.05533 2.4630
566 0.05302 0.1590 0.05648 1.0750
567 0.15200 0.2397 0.07016 1.5950
568 0.00000 0.1587 0.05884 1.4280
..
566 0.07820
567 0.12400
```

Diabetes

```
breast = pd.read_csv("processed_data.csv")

print(breast['texture_mean'].min())
print(breast['texture_mean'].max())

...
9.71
39.28

print(breast['smoothness_mean'].min())
print(breast['smoothness_mean'].max())

...
0.05263
0.1634

print(breast['compactness_mean'].min())
print(breast['compactness_mean'].max())

...
0.01938
0.3454

print(breast['concave points_mean'].min())
print(breast['concave points_mean'].max())

...
0.0
0.2012
```

Breast Cancer

REFERENCES

- [1] Al-Turaiki, I. (2021). Heart Disease Prediction using Machine Learning Techniques: A Review. International Journal of Advanced Research in Computer Science, 12(2), 19-28.
- [2] American Diabetes Association. (2021). Standards of Medical Care in Diabetes—2021. Diabetes Care, 44(Supplement 1), S1-S232.
- [3] American Liver Foundation. (2021). Liver Disease. Retrieved from <https://liverfoundation.org/liver-disease/> El-shafey, M. A., & Hassanien, A. E. (2020).
- [4] Han, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.
- [5] National Breast Cancer Foundation. (2021). Breast Cancer. Retrieved from <https://www.nationalbreastcancer.org/breast-cancer/>
- [6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine

Learning Research, 12(Oct), 2825-2830.

[7] Python Software Foundation. (2021). Python Language Reference, version 3.10. Retrieved from <https://docs.python.org/3/>

[8] Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference, 57-61.

[9] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods, 17(3), 261-272.

[10] Chronic Kidney Disease Prediction Using Machine Learning Techniques: A Comprehensive Review. Journal of Healthcare Engineering, 2020, 1-17.

APPENDIX

A. SCREENSHOTS

app

diabetes

heart diseasekidneyliver

Know If You Are Affected By Diabetes

All The Values Should Be In Range Mentioned

Enter Your Glucose Level (0-200)

0-+

Enter Your Insulin Level In Body (0-850)

0-+

Enter Your Body Mass Index/BMI Value (0-70)

0-+

Enter Your Age (20-80)

20-+

Predict

You Are Safe

Diabetes Webpage

app

diabetes

heart disease

kidney

liver

Enter the age

0.00-+

Enter the BP

0.00-+

Enter the al value

0.00-+

submit

You dont have any kidney issues

Kidney Webpage

×

app

diabetes

heart disease

kidney

liver

Exercise Induces Angina

yes

ST Depression Induced by Exercise Relative to Rest:

0.00

Slope of the Peak Exercise ST Segment

Upsloping

Number of Major Vessels Colored by Flourosopy

0

Thalassemia:

Normal

Submit

Yes

Heart Disease Webpage

×

app

diabetes

heart disease

kidney

liver

Enter Your Blood Pressure Rate (10-2000)

10

aspartate aminotransferase (10-5000)

10

total protein (2-10)

2

albumin

0

Enter ratio

0

Predict

You Might Be Affected By Liver disease

Liver Webpage

B.SOURCE CODE

```
Help  ← →  all_disease

diabetes.py ×
pages > diabetes.py > ...
1  import streamlit as st
2  import pandas as pd
3  from sklearn.model_selection import train_test_split
4  import numpy as np
5  from sklearn.ensemble import RandomForestClassifier
6  df2=pd.read_csv("Diabetes Predictions.csv")
7  #cleaning the data by dropping unnecessary column and dividing the data as features(x2) & target(y2)
8  x2=df2.iloc[:,1,4,5,7].values
9  x2=np.array(x2)
10 y2=y2=df2.iloc[:,[-1]].values
11 y2=np.array(y2)
12 #performing train-test split on the data
13 x2_train,x2_test,y2_train,y2_test=train_test_split(x2,y2,test_size=0.2,random_state=0)
14 #creating an object for the model for further usage
15 model2=RandomForestClassifier()
16 #fitting the model with train data (x2_train & y2_train)
17 model2.fit(x2_train,y2_train)
18
19 st.header("Know If You Are Affected By Diabetes")
20 st.write("All The Values Should Be In Range Mentioned")
21 #taking the 4 most important features as input as features -> Glucose (glucose), Insulin (insulin), Body Mass Index-BMI (bmi), Age
22 #a min value (min_value) & max value (max_value) range is set so that user can enter value within that range
23 #in case user enters a value which is not in the range then the value will not be taken whereas an alert message will pop up
24 glucose=st.number_input("Enter Your Glucose Level (0-200)",min_value=0,max_value=200,step=1)
25 insulin=st.number_input("Enter Your Insulin Level In Body (0-850)",min_value=0,max_value=850,step=1)
26 bmi=st.number_input("Enter Your Body Mass Index/BMI Value (0-70)",min_value=0,max_value=70,step=1)
27 age=st.number_input("Enter Your Age (20-80)",min_value=20,max_value=80,step=1)
28 #the variable prediction1 predicts by the health state by passing the 4 features to the model
29 prediction2=model2.predict([[glucose,insulin,bmi,age]])[0]
30
31 #prediction part predicts whether the person is affected by Diabetes or not by the help of features taken as input
32 #on the basis of prediction the results are displayed
33 if st.button("Predict"):
34     if prediction2==1:
35         st.warning("You Might Be Affected By Diabetes")
36     elif prediction2==0:
37         st.success("You Are Safe")
38
Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  Python  3.10.8 64-bit
```

```
File Edit Selection View Go Run Terminal Help  ← →  all_disease

EXPLORER
ALL DISEASE
  pages
    diabetes.py
    heart_disease.py
    kidney.py
    liver.py
  1.png
  app.py
  cnn-parameters-improvement-23-091.model
  covid_classification.h5
  Diabetes Predictions.csv
  disease_dataset.csv
  heart.joblib
  indian_liver_patient.csv
  kidney_disease.csv
  my_model2.h5
  test.jpg

heart_disease.py ×
pages > heart_disease.py > ...
34  '4':4
35  }
36  }
37  thall_map = {
38      'Normal':0,
39      'Fixed Defect':1,
40      'Reversible Defect':2
41  }
42  age = st.number_input("Enter The age")
43  gender = st.selectbox("Enter your gender", ['male', 'female'])
44  cpt = st.selectbox("Enter the chest pain type", ["Typical Angina", "Atypical Angina", "Non-Anginal Pain", "Asymptomatic"])
45  rbp = st.number_input("Enter Blood Pressure")
46  bloodsugar = st.selectbox("Fasting blood sugar", ["less than 120 mg/dl", "greater than or equal to 120 mg/dl"])
47  restecg = st.selectbox("Resting ECG result", ["Normal", "ST-T Wave Abnormality", "Left Ventricular Hypertrophy"])
48  hra = st.number_input("Heart Rate Achieved")
49  chl = st.number_input("Cholesterol")
50  eia = st.selectbox("Exercise Induces Angina", ['yes', 'no'])
51  oldpeak = st.number_input("ST Depression Induced by Exercise Relative to Rest")
52  slp = st.selectbox("Slope of the Peak Exercise ST Segment", ['Upsloping', 'Flat', 'Downsloping'])
53  caa = st.selectbox("Number of Major Vessels Colored by Fluoroscopy", ['0','1','2','3','4'])
54  thall = st.selectbox("Thalassemia:", ['Normal', 'Fixed Defect', 'Reversible Defect'])
55  pred = st.button("Submit")
56
57  if pred:
58      model = joblib.load('heart.joblib')
59      gender = gender_map[gender]
60      cpt = cpt_map[cpt]
61      bloodsugar = bloodsugar_map[bloodsugar]
62      restecg = restecg_map[restecg]
63      eia = eia_map[eia]
64      slp = slp_map[slp]
65      caa = caa_map[caa]
66      thall = thall_map[thall]
67      res = model.predict([[age, gender, cpt, rbp, bloodsugar, restecg, hra, chl, eia, oldpeak, slp, caa, thall]])
68      if res[0] == 0:
69          st.write('Yes')
70      else:
71          st.write('NO')
72
Ln 7, Col 1 (25 selected)  Spaces: 4  UTF-8  CRLF  Python  3.10.8 64-bit
```

```

1 import streamlit as st
2 import pandas as pd
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.preprocessing import LabelEncoder
5 df = pd.read_csv('kidney_disease.csv')
6 df = df.drop_duplicates()
7 df = df.dropna(how='any')
8 X = df[['age', 'bp', 'al']]
9 Y = df['ane']
10 le = LabelEncoder()
11 le.fit_transform(Y)
12 model = LogisticRegression()
13 model.fit(X, Y)
14
15 age = st.number_input('Enter the age')
16 bp = st.number_input('Enter the bp')
17 al = st.number_input('Enter the al value')
18
19 pred = st.button('submit')
20
21 if pred:
22     predict = model.predict([[age, bp, al]])
23     if predict[0] == 0:
24         st.write('Yes You have kidney issues')
25     else:
26         st.write('You dont have any kidney issues')
27

```

```

1 import streamlit as st
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 import numpy as np
5 from sklearn.svm import SVC
6 df4 = pd.read_csv('indian_liver_patient.csv')
7 df4 = df4.drop_duplicates()
8 df4 = df4.dropna(how='any')
9
10 df4 = df4.drop(['Age', 'Gender'], axis=1)
11 y = df4['Dataset']
12 x = df4.drop(['Dataset'], axis=1)
13 X_train, X_test, y_train, y_test = train_test_split(x, y,
14                                                     test_size=0.2,
15                                                     random_state=0,
16                                                     stratify=df4['Dataset'])
17 model4 = SVC()
18 model4.fit(X_train, y_train)
19
20 st.header("Know If You Are Affected By Liver")
21 st.write("All The Values Should Be In Range Mentioned")
22
23 total_bilirubin = st.number_input("Enter Your total bilirubin (0.4-75)", min_value=0, max_value=75, step=1)
24 direct_bilirubin = st.number_input("Enter direct bilirubin (0-20)", min_value=0, max_value=20, step=1)
25 alkaline_phosphatase = st.number_input("Enter alkaline phosphatase (60-2200)", min_value=60, max_value=2200, step=1)
26 alaine_aminotransferase = st.number_input("Enter Your blood pressure Rate (10-2000)", min_value=10, max_value=2000, step=1)
27 aspartate_aminotransferase = st.number_input("aspartate aminotransferase (10-5000)", min_value=10, max_value=5000, step=1)
28 total_protein = st.number_input("total protein (2-10)", min_value=2, max_value=10, step=1)
29 albumin = st.number_input("albumin", min_value=0, max_value=6, step=1)
30 albumin_and_globulin_ratio = st.number_input("Enter ratio ", min_value=0, max_value=3, step=1)
31
32 prediction2 = model4.predict([[total_bilirubin, direct_bilirubin, alkaline_phosphatase, alaine_aminotransferase, aspartate_aminotransferase, total_protein, albumin, albumin_and_globulin_ratio]])
33
34 #prediction part predicts whether the person is affected by Diabetes or not by the help of features taken as input
35 #on the basis of prediction the results are displayed
36 if st.button("Predict"):
37     if prediction2 == 1:
38         st.warning("You Might Be Affected By Liver disease")
39     elif prediction2 == 0:
40         st.success("You are not affected by Liver disease")

```

