A SMART HEALTH DISEASE EVALUATOR

RAVISH KUMAR DUBEY [1], ASHI VERMA [2], BHARAT BAJAJ [3], DHRUV BHATT [4] & GEETANJALI WADHWA [5]

[1]Assistant Professor, Dept. of Computer Science & Engineering, Moradabad Institute of Technology, Uttar Pradesh, India [2]Student, Dept. of Computer Science & Engineering, Moradabad Institute of Technology, Uttar Pradesh, India [3]Student, Dept. of Computer Science & Engineering, Moradabad Institute of Technology, Uttar Pradesh, India [4]Student, Dept. of Computer Science & Engineering, Moradabad Institute of Technology, Uttar Pradesh, India [5]Student, Dept. of Computer Science & Engineering, Moradabad Institute of Technology, Uttar Pradesh, India

ABSTRACT:

In the absurd situations such as lockdown or due to some urgency, busy schedule etc. Most people do not go for regular medical checkups. Even if they have certain health problems such as coughing, runny nose etc. on regular intervals they still do not prefer referring a doctor. Keeping in mind the gap between the doctors and patients, and the ability of AI to take human comparable decisions, An Application that can diagnose a person's disease based on his/her symptoms, external doctors online feedback and consultancy, and fairly accurate Machine Learning algorithms is thought of a first step to bridge the gap between the people and the health services. Diagnosis process like many other processes is ready to be digitalized and this will all be possible because of digital system's ability to learn using dataset, and when this process of machine learning along with a large dataset of limited diseases is used, it becomes practical to have a diagnosis without any doctor involved. And so this research paper contains the prerequisites, techniques and the past researches in this field to explain the idea and its feasibility in the practical world. This research paper also contains the working technologies of essential components of the application such as its AI engine, APIs, database, middle layers, UI, etc. Along with their most suitable form of implementation in the application, e.g.: Apache server could be used instead of SQL server that is being used but it would have required an additional middleware technology such as PHP or Java.

KEYWORDS: AI (Artificial Intelligence), API (Application Programming Interface), UI (User Interface), SQL (Structured Query Language).

I. INTRODUCTION

With the increasing success of automated tasks and less requirements of visiting a doctor for small and simple diseases but which if not taken care of can ruin your daily life, it is the need of an hour to create a fully automated diagnostic system that will work on two sub-domains of Machine Learning (ML). Machine Learning is the process of making machines capable of doing various different tasks. It is a part of Data Mining which helps to work on different kinds of data & hence making machines doing various tasks. It has made our lives easier by day to day applications from taking care of our health to weather forecasting. The system works on one such application of ML having a wide usage in the field of Medical Diagnosis. Disease Evaluator is an application that works on ML & hence making Users/Patients making able to interact with the doctors immediately while facing various body symptoms.

It might have happened so many times that you or someone yours need doctors help immediately, but they are not available due to some reason. Here we propose an android application that allows users to get instant guidance on their health issues through an intelligent health care application online. The application is fed with various symptoms and the disease/illness associated with those systems. The system allows user to share their symptoms and issues. Then processes user's symptoms to check for various illnesses that could be associated with it using some intelligent data mining techniques. The application also has a doctor login, these details are now sent to respective doctor along with patient contact details. The doctors may now contact the patient for further process.

The Smart Health Disease Evaluator is the System consisting of three modules; patient, doctor & admin, which are interconnected with each other & hence patients can easily be able to book appointments with the predicted doctor by the system. Patients only have to enter the symptoms they are facing & the system will automatically predict the related disease along with the respective doctor's list in their own city itself. The system consists of three modules where two of them are android applications & other is the website. The doctor & the patient modules are the android applications whereas the admin module is the website. Let's understand the project design & project description briefly.

II. PROJECT ARCHITECTURE

Creating/developing something is not as simple as it looks, especially when we talk of developing self learning structures and using them in diagnosis processes, this software tool have to be simple Enough to be easily used by common people and accurate to a extent that it do not give erroneous diagnosis or wrong medical consultancy to anyone, So in general, the backbone of the software i.e. its

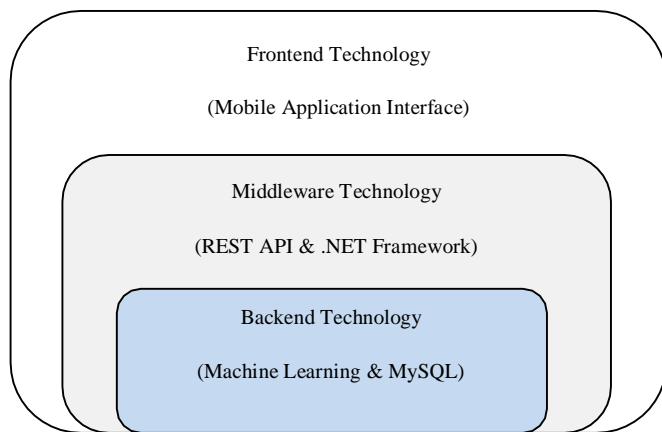architecture has to be really robust. So the following will be the parts of its robust architecture.

Fig 1: Architecture of "Smart Health Disease Predictor"

A. Machine Learning

Machine Learning had played a major role in the designing of the Project. As discussed earlier the project contains three modules, in which the patient side application is the disease prediction system. Here, the patient/user can enter the symptoms & will perhaps get the predicted disease he might be facing with. These predictions are done with the help of various ML algorithms that are; Naïve Bayes (NB), Random Forest (RF). Decision Tree Classifier (DTC) & K Nearest Neighbors (KNN). [1]

B. Application Programming Interface (API)

The module of the Patient side is then connected with the admin module as well as the doctor's application using the REST API. [6] REST API allows us to interact or interconnect with various applications & machine learning models along with the website simultaneously at the same time.

For every data transfer to be secure and fast , every application should have a dedicated API that handles data transfer, so this application have an API that is composed of C# and .Net components[3] which first converts a set of data into a single JSON string then transfers it from one point to another.

C. Mobile Application Interface

For a technology to be usable by common people, it should be simple and easy to use, in other words it should have a really simple and effective user interface to connect with people and when the technology has to be used in smart phones it must be integrated well. [5]

III. PROJECT IMPLEMENTATION

The implementation of the system can be better understood by diving it into different parts. The parts are as follows

Part A: Predictions using Machine Learning

This part of the Project implementation deals with collecting the dataset, fitting machine learning algorithms to get the best fit out of them, splitting of the dataset using machine learning techniques, single value predictions & perhaps representing the system on Graphical User Interface (GUI).

In this part we have used the diabetes dataset & hence the disease-symptoms dataset for the predictions to be made. The framework which is used here is the Scikit Learn & GUI is made with the help of tkinter. Python is the basic programming language which is used the system along with the application of data science with the help of PANDAS. [1]

Data Collection: It is the process where information is gathered perhaps the targeted variables of the establishing system are measured. The Process of data collection also enables one to answer questions relevant to the system been established hence evaluating the outcomes of the system. Various tools are used for the collection or the acquisition of the data such as; case studies, checklists, surveys, questionnaires, report, feedback, online resources, etc. [7]

We have used two datasets in the process of project implementation. First is the Diabetes dataset which we got from kaggle.com [8]. Kaggle allows users to find & publish datasets & hence explore & build the models in data science environment. Second one is the disease- symptoms dataset that we have acquired from the "Unified Medical Library System" [2] as it is the library which is used by doctors itself worldwide.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1

Fig 2: Collected Diabetes Dataset [8]

	Disease	Symptom
0	UMLS:C0020538_hypertensive disease	UMLS:C0008031_pain chest
1	UMLS:C0020538_hypertensive disease	UMLS:C0392680_shortness of breath
2	UMLS:C0020538_hypertensive disease	UMLS:C0012833_dizziness
3	UMLS:C0020538_hypertensive disease	UMLS:C0004093_asthenia
4	UMLS:C0020538_hypertensive disease	UMLS:C0085639_fall

Fig 3: Collected Disease Symptoms Dataset [2]

Splitting the Dataset: After the collection of the dataset, the first thing we did was splitting up the datasets into two parts i.e.; training dataset & testing dataset respectively. This process of splitting the dataset, it enables one to build the best machine learning model. There is no fixed proportion of splitting the dataset into a fixed proportion but usually what applies here is splitting in such a way that training dataset is much bigger than the testing one.[1]

Training Dataset	Testing Dataset

Fig 4: Datasets Approximated Ratio

The idea behind splitting of datasets comes in such a way that the establishing model or the system can work efficiently by working more time on the bigger dataset & hence can work better on the unseen data as well. This is

why the two sets that are training & testing datasets are called as "holdout sets".

Data Preprocessing: This is the technique which is used in data mining to transform raw data into an efficient & useful form. There are majorly three techniques of data preprocessing, they are; a) Data Cleaning, b) Data Transformation & c) Data Reduction. On the practical perspective the preprocessing can be done by methods such as; backward fill (bfill()), forward fill (ffill()), mean fill (mean ()), median fill (median()), etc.[4][7] Here, we have used ffill() method to process the raw data. The final data obtained is as follows

	Disease	Symptom
0	hypertensive disease	[pain chest, shortness of breath, dizziness, a...
1	diabetes	[polyuria, polydypsia, shortness of breath, pa...
2	depression mental	[feeling suicidal, suicidal, hallucinations au...
3	depressive disorder	[feeling suicidal, suicidal, hallucinations au...
4	coronary arteriosclerosis	[pain chest, angina pectoris, shortness of bre...
5	coronary heart disease	[pain chest, angina pectoris, shortness of bre...
6	pneumonia	[cough, fever, decreased translucency, shortne...
7	failure heart congestive	[shortness of breath, orthopnea, jugular venou...
8	accident cerebrovascular	[dysarthria, asthenia, speech slurred, facial ...
9	asthma	[wheezing, cough, shortness of breath, chest t...
10	myocardial infarction	[pain chest, st segment elevation, sweat, swea...
11	hypercholesterolemia	[pain, pain chest, sweat, sweating increased, ...

Fig 5: Preprocessed Dataset

Model Deployment: In this part we have used the Diabetes dataset to find out the best fit algorithms. Let's discuss them in brief.

Decision Tree Classifier (DTC)

It is a type of supervised learning algorithm that is mostly used in classification problem. A decision tree is an acyclic graph or a tree which could be used to make decisions as each branch node of the tree represents a choice between a

number of alternatives & each leaf represents the decision.

In the system, we have the collection of dataset that belong to set {0, 1}. We need to establish the model where the feature vector is to be predicted. There are various formulations of the implementation but here we have used the method called ID3 using the Information Gain.

Attribute Selection: If the dataset consists of N no. of attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a difficult task. So, for the sake of this purpose, various criterions have been decided. Here in this system, we have used one such criterion that is ID3 whereas others can be gini index or chi square also.

Entropy (H): It can be defined as the measure of randomness or the unpredictability in the dataset. As the Dataset splits, entropy decreases simultaneously.

$$H(P_1, P_2, \ldots P_n) = -P_1 \log P_1 - P_2 \log P_2 \ldots \ldots -P_n \log P_n$$

$$\text{Or } H(x) = -\sum_x P(x) \cdot \log P(x) \qquad \text{- (eq. 1)}$$

where P is the Probability & 1,2,….n are the no. of occurring events.

Information Gain (IG): It is the measure of decrease in entropy after the dataset is split. The core algorithm for building dataset is ID3 which was given by J.R. Quinlan, which employs a top down greedy search through the space of possible branches with no backtracking. IG=H(parent)-[Weighted avg.}·H(children)   - (eq. 2)

Steps in ID3 algorithm:

1) The Algorithm starts with the root node of new dataset S.

2) For each iteration, Entropy (H) using eq. 1 and Information gain (IG) using eq. 2 of the unused attribute of the set S is calculated.

3) It selects that attribute which is having the smallest Entropy (H) or Largest Information gain (IG).

4) Set S is then split by the selected attribute & produces a subset of the data.

5) The algorithm continues to iterate on each subset, considering only attributes never used before.

We have used DTC algorithm because it allows one to analyze the possible consequences of a decision & provide framework to quantify the values of outcomes.[1] From here we got the feature importance along with training & testing accuracies as 79% & 78% respectively.

K Nearest Neighbors (KNN)

Abbreviated as KNN, this algorithm is most significantly used in classification situations & can also be used for regression that is for continuous data. It is a non – parametric learning algorithm. This algorithm is also said

to be the "Lazy Algorithm" because it is instance based. Instance based refers to the algorithm that explicitly learns a model that is it chooses to memorize the training instances.

The KNN algorithm was proposed by Fix & Hodges in 1951 for pattern classification task. The major role is played by the distance & the value of 'k' in this algorithm which can be defined as follows

The Distance: Euclidean Distance is calculated as the square root of the squared differences between a new point & the existing point. Let the new point be p & existing point be q respectively. The Euclidean Distance, d can be calculated as

$$d(p, q)= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \qquad \text{- (eq. 3)}$$

Value of 'k': Finding the value of k becomes a really very difficult task. Sometimes, it becomes best to run through each point or the position value for k & hence decide by yourself.

Steps in KNN Algorithm: Let m is the no. of training data samples & p is the unknown point.

1) Start with storing the data samples into the array of data points (say arr[]).
2) Calculate the Euclidean Distance from the point p to all the data points stored in arr[].
3) Store the 'k' small distances into a set (say S).
4) Return the majority label from the stored dataset, S.

We have used KNN because of it predictions that are made just-in-time by simply calculating the similarities. [1] From here we got training & testing accuracies as 74% & 73% respectively.

Naïve Bayes (NB)

In Machine Learning, a bayes' classifier is simple & probabilistic which can be implemented using the Bayes' Theorem. The Naïve Bayes algorithm makes a strong & independent assumption meaning that an existing feature or the attribute of the class is independent to the other existing feature or attribute of the class.

Bayes' Theorem: The bayes' theorem finds out the probability of an event occurring, given that the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$P(A|B)= \frac{P(B|A) \cdot P(A)}{P(B)} \qquad \text{- (eq.}$$

4)

This theorem can be divided into 3 categories; a) Gaussian Naïve Bayes, b) Bernoulli Naive Bayes & c) Multinomial Naïve Bayes. In this system, we have used Gaussian Naïve Bayes as it is used for binary classification. [1]

Naïve Bayes is basically used in this system because of its

suitability in binary & multiclass classification along with independent assumption. From here we got training & testing accuracies as 76% & 75% respectively.

Random Forest (RF)

This algorithm is supervised & is sample learning which is used for classification (discrete data) as well as regression (continuous data) techniques. This algorithm/technique builds multiple Decision Trees & hence mergers them together to get the most accurate results along with the stable prediction.

Ensemble Learning aggregates multiple Machine Learning methods/models for overall better performance of the system.

Steps for the Algorithm:

1) Select 'k' features from n no. of features randomly such that n < k.
2) From the selected features, calculate the node using best split point.
3) Split the nodes into internal nodes using best split method.
4) Repeat the above steps (1, 2, 3) until the leaf node is found.
5) Build a 'Forest' by repeating all of the above steps.

We used RF due to its increased predictive power & feature that prevents the model from 'over fitting'. [1] From here we got training & testing accuracies as 80% & 75% respectively.

After applying all the above discussed algorithms on the same dataset (Diabetes), we selected 3 best fit algorithms that is; 1) Naïve Bayes, 2) Decision Tree Classifier & 3) Random Forest using k fold cross validation technique (process of training using one set of data & test on other set of data) by comparing with each other. These algorithms are then applied to the disease-symptom dataset where it predicts disease based on the symptoms. A Graphical User Interface (GUI) of the predictor is then built using tkinter as follows
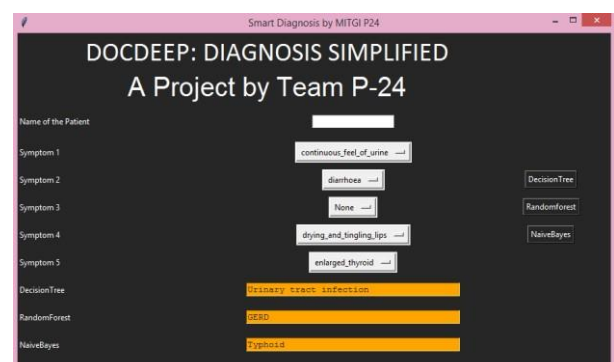

Fig 6: Backend of Smart Health Disease Evaluator System

Part B: Frontend Technology

.Net Framework

.Net is a framework designed by Microsoft to develop software applications. There are two essential core components of .net framework.

Framework Class Library (FCL): It contains a large number of class libraries to build an application.

Common Language Runtime: It provides an environrnent for the execution of programs.

Modules of .Net Framework

ASP.NET

ASP stands for Active Server Pages. It is a module of .net framework designed by Microsoft. It is basically used for the development of websites, web apps and web services. It is a integration of HTML, CSS and JavaScript.

Asp.net web forms: We use Visual studio to create Asp.net web forms. These web forms are implemented on server and displays output to the browser. Although visual studio provides a facility of drag and drop to display various objects but we can write code in editor also.

HTML Server controls: These are the HTML elements. By default, HTML elements in Asp.net are treated as simple text. To make them programmable we add the attribute runat = "server" which indicates that elements should be treated as server controls T here are various server controls that we use to create web forms each having its predefined function. It includes Label, Textbox, Radio Button, Checkbox, Hyperlink, Dropdown List, Tables, Calendars, etc. [3]

C#

C# is pronounced as C Sharp. Although, .net framework support more than 60 languages like COBOL, C#, C++,Visual Basic and many more we have used C# due to its salient features; Simple, Object Oriented,, Interoperability, Modern, Fast Speed & Scalability.

The syntax of C# includes following; Namespace declaration, Class, Class methods, Class attributes, Main methods, Statements & Expressions, & Comments. [3]

Basic Syntax;

Using system;

namespace namespace _name

{Class class_ name

    {Static void Main (string [ ] args)

        {Console .writeline("message");

 }}}

In this:

"*Using system*" is a keyword to include namespace system.

"*Namespace*" is a collection of classes. "*Writeline*" is a method of writing text on console.

The code of C# can be written in any text editor such as Notepad and then compile in console or use visual studio. The steps are as follows

1)   Start visual studio.
2)   Select File > New >Project.
3)   Choose visual C# from templates and then windows.
4)   Choose Console application.
5)   Specify the name and click Ok.

ADO.NET

ADO stands for Active Data Objects. [3] It is a module in .net framework to establish connection between applications and data sources (SQL server). ADO.net provides the built in functions which help us to connect to the database. It consists of classes that can be used to create, retrieve, insert and delete data. It has two basic components which can be stated as follows

Data Provider: Data Provider requires System.data.SqlClient to provide data access for data. There are four main objects in data provider listed as follows

1)   Connection: It has the inbuilt function sqlconnection () to connect the application with database. This connection does not close by itself so we use another inbuilt function close () method.
2)   Commands: The commands are used to execute queries to perform database operations.
3)   Data Reader: This is used to read data from data source. It is used to read the rows from a SQL server database.
4)   Data Adapter: It represents the set of data commands and a database connection that are used to fill datasets and update it.

Datasets: It is a collection of data tabled that contain data. It helps in fetching data without interacting with data sources.

ADO.net data table: It represents the data on tabular form in the form of rows and columns. Initially it does not have any schema. We can create schema by adding columns and constraints. ADO.net allows us to store the submitted values into SQL server database in web forms.

Part C: Middleware Technology
Rest API

The core of every dynamic application require some essential tasks/functionalities to be carried out called CRUD (CREATE, READ, UPDATE, DELETE) functionalities.

Along with this it requires to manage sessions, build cookies, checks for security protocols every time the routing has been done. [6]

The components of an application that communicate with other components of the application are called APIs as shown as follows
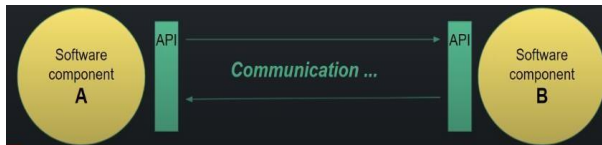

Fig 7: Communication between the two Systems

When we talk about back-end of an application API is that component of application that is responsible for complete routing and data flow ensuring its security and the state of consistency in database. When an API confirms to restful web services it is called REST API. A restful web service is a one that confirms to restful routing. For a routing to be restful most important thing to be known is that can has to be sent everywhere in the form of JSON strings. Thus it requires converting the data into JSON string before transfer and converting the JSON string into original data back after transfer.

Our diagnosis application has a huge role of rest API as it contains a good amount of transfer of data from patients, doctors, AI engine, databases, etc. Back and forth and therefore it was mandatory to build an API that may built as a class so as to inherit its properties and make code more reusable as the application is very data intensive and without code reusability the complexity of the application will reach to heights. Here's an example of how various patient details are firstly converted to JSON string and then routed. [6]

```
82
83
84    function patient_personal_details(name,phn,address,previous_medical_conditions,gender){
85        this.name = name;
86        this.phn = phn;
87        this.address = address;
88        this.previous_medical_conditions = previous_medical_conditions;
89        this.gender = gender;
90    }
91
92
93    let a = new patient_personal_details(x,y,z,s,d);
94    person.push(a);
95
96
97
98
99    SQLserver.setItem('patient_personal_details', JSON.stringify(person));
100
101
102   express.closeConnection();
103
104
105
106
107   /
```

Fig 8: REST API function using JSON

SQL server

To avoid the additional middleware technologies such as PHP, java etc. It was thought as SQL server to replace traditional Apache, XAMMP servers in the project which gives us GUI features along with the features that a traditional MySQL server provides, and hence using SQL server we not only got rid of additional technologies and additional lines of code but also got GUI features which made our query processing quite easy, now with the use of little extension we can manipulate data without additional lines of code, making our application efficient in terms of response time.

As SQL server eliminates the need of server, so in the absence of any external server, it itself contains a protocol layer connecting external interface to the SQL server.

Query processing is still one of the important parts of the application management as to all the modules; Doctor, Patient, Admin are distinguished in the data base with the help of SQL views creation, creating different views for different users ensuring complete integrity of data and Consistency in database after every operation.

PART D: Introduction to Android

As we know that Android is a mobile operating system developed by Google. It is used by several smart phones and tablets. The Android operating system (OS) is based on the Linux kernel. Unlike Apple's iOS, Android is open source, meaning developers can modify and customize the OS for each phone. [5]

Android System Architecture Details: Most of the Android user are using their Android phone just for calls, SMS, browsing and basic apps, But form the development prospective, we should know about Android internal structure. Android uses several partitions (like boot, system, recovery, data etc) to organize files and folders on the device just like Windows OS. Each of these partitions has its own functionality.

About applications: The first layer of the android architecture is an application layer, in this layer your written application which to be installed are available on this layer only. Example like applications Contacts books, Browser, Games etc.

Application framework: This layer provides many high-level services to applications in the form of java classes. Application developers are allowed to make use of these services in their applications

Android libraries: Android libraries are available on top of Linux kernel there is a set of libraries including open-source Web browser engine web kit, well known library libc, SqLite database which is a useful option for storage and sharing of application data, libraries to play and record and video, SSL libraries responsible for internet security etc.

Android Runtime system: This is the third part of android architecture .this section provides a key component called

Dalvik Virtual Machine (DVM) specially designed and optimized for android. The Dalvik Virtual Machine makes use of Linux core features like memory management and multi-threaded, which is intrinsic in the java language.

Android Linux Kernel: In the Linux layers Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc.

Gradle Build System: As we know that Android Studio uses Gradle as the foundation of the build system, with more Android - specific capabilities provided by the Android plug in for Gradle.

This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. We can definitely use the features of this build system to do as follows

1) We have very well customized, configured, and extended the build process.
2) Created multiple APKs for our app, with different features used in our project and modules.

Debug and Profile Tools: Android Studio assists us in debugging and improving the performance of our code, including inline debugging and performance analysis tools.

Inline debugging: We use inline debugging to enhance our code walk-through in the debugger view with inline verification of references, expressions, and variable values. Inline debug information includes the following processes

1) Inline variable values
2) Referring objects that reference a selected object
3) Method return values
4) Lambda and operator expressions
5) Tooltip values

Performance monitors: Android Studio provides performance monitors so we can more easily track our app's memory and CPU usage, and can find de-allocated objects, locate memory leaks, optimize graphics performance, and analyze network requests.

Allocation tracker: Android Studio allows is to track memory allocation as it monitors memory use. Tracking memory allocation allows us to monitor where objects are being allocated when we perform certain actions. Knowing these allocations enables us to optimize our app's performance and memory use by adjusting the method calls related to those actions.

Code inspections: Whenever we compile our program, Android Studio automatically runs configured Lint and other IDE inspections to help us easily identify and correct problems with the structural quality of our code.

The Lint tool has been used to check our Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.[5]

IV. APPLICATIONS OF THE SYSTEM

The application enables a client to share their symptoms and issues. It at that point forms client's side effects to check for different disease that could be related with it. Here we utilize some keen information mining procedures to figure the most exact sickness that could be related to the patient's symptoms.

In the event that the application can't give appropriate results, it urges clients to go for blood test, x-Ray, CITI scan or whichever report it feels client's side effects are related with, so next time client might have the capacity to login and transfer a picture of those reports. This application can be used by all patients or their family members who need help in emergency.

The modern technique to healthcare is to prevent the disease with early symptoms rather than go for a treatment after diagnosis. Traditionally, physicians or doctors use a risk calculator to assess the possibility of disease occurrence. These calculators use basic information such as demographics, medical conditions, life routines, and more to calculate the probability of developing a certain disease.

The highest applications of the system established are as follows

1) *Accurate Diagnosis:* Present-day human doctors are not able to recall all the information necessary to make an observation but an accurate diagnosis. With the abundance of resources such as; textbooks, research papers and case studies, no doctor could even master every aspect of medical care and recall every detail of the familiar as well as similar cases. Perhaps, it is possible to train an AI-based system with relevant data and let the computer predict through the extensive database instead of relying on human knowledge.

2) *Disease Identification:* Personalized medical treatments can provide for a more effective individual health care by data paired with predictions & analysis.

3) *Early Drug Discovery:* Machine learning is a tool in early-stage drug discovery which may be used in a range of activities, from initial screening to success rate prediction based on biological factors of drug discovery and to research & development discovery technologies.

4) *Monitoring epidemic outbreaks:* The sources of data acquisition include satellites, historical information on the web, real-time social media updates, and more. Predicting epidemic outbreak is particularly crucial in countries that often lack medical infrastructure, educational opportunities, expertise, and access to treatments. The application can work better there such.

V. ADVANTAGES & DISADVANTAGES OF THE SYSTEM

Advantages

Traditionally, there was no such system developed from which we would be able to get the details of various illness/disease on a single application. Someone needs to contact doctor to get the idea about the illness/disease or also may need to visit doctor's dispensary. People need to travel to doctor's dispensary wherever or how far it would be. No. of pages used to maintain the records of n no. of peoples.

Considering the anomalies in the existing system computerization of the whole activity is being suggested after initial analysis. It might have happened so many times that you or someone yours need doctors help immediately, but they are not available due to some reason. The Health Prediction application is an end user support and online consultation project.

Here, we propose an android application that allows users to get instant guidance on their health issues through an intelligent health care system online. The application is fed with training data and the disease associated with those symptoms. The application allows user to check their health just by entering multiple values into system to predict health related issues. The main advantages of the project are as follows

1) User can search for doctor's help at any point of time.
2) User can talk about their illness and get instant diagnosis.
3) Doctors get more clients online.

Other advantages of the project are as follows

1) *Load Balancing:* Since the system will be available only the admin logs in the amount of load on server will be limited to time period of admin access.
2) *Easy Accessibility:* Records can be easily accessed and store and other information respectively.
3) *User Friendly:* The system will be giving a very user friendly approach for all user.
4) *Efficient and reliable:* Maintaining the all secured and database on the server which will be accessible according the user requirement without any maintenance cost will be a very efficient as compared to storing all the customer data on the spreadsheet or in physically in the record books.
5) *Easy maintenance:* This system is design as easy way. So maintenance is also easy.

Disadvantages

Traditional machine learning algorithms such as naïve bayes and decision trees have been employed to design prediction

models for medical data sets. These approaches can provide better accuracy of prediction. As machine learning algorithms become popular and accessible, we have tempted to apply ensemble learning method to the data sets in an attempt to improve classification prediction accuracy. However, this type of data is often constricted by smaller sets which are usually preferred to allow for sufficient training and testing of machine learning models. Without sufficiently measured data sets, it is very difficult to determine if a model is best fit to previously unseen sets of data. Major disadvantages of the project are as follows

1) The system is not fully automated.
2) It needs doctors for full diagnosis.
3) It requires active internet connection.

VI. CONCLUSION

The Smart Health Disease Evaluator, historically viewed as a necessary burden in medical offices, healthcare facilities and wellness centers, can be completely automated through an inefficient online software program. The benefits of implementing this technology touch everyone involved in the scheduling process, as administrators and staff can conduct their tasks more efficiently and accurately, while customers and clients have the ability to book their appointments and reservations quickly and more conveniently. Furthermore, the project has a high rate which makes it a handy for doctors to screen patients with high probability of having the disease and transfer those patients to hospitals if required. This project has established that data mining techniques applicable in the prediction of common diseases.

REFERENCES

[1] ANDRIY BURKOV., ed.; The Hundred-Page Machine Learning Book, 1st edition (2019).

[2] Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004 Jan 1;32 (Database issue):D267-70. doi: 10.1093/nar/gkh061. PubMed PMID: 14681409; PubMed Central PMCID: PMC308795.

[3] http://quickstart.developerfusion.co.uk/QuickStart/howto/doc/adoplus/overviewcontents.aspx

[4] https://research.ijcaonline.org/volume121/number3/pxc3904505.pdf.

[5] https://stuff.mit.edu/afs/sipb/project/android/docs/tools/debugging/improving-w-lint.html.

[6] https://www.datacamp.com/community/tutorials/machine-learning-models-api-python.

[7] https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[8] https://www.kaggle.com/edubrq/diabetes.