**CHAPTER 1**

**INTRODUCTION**

* 1. **BACKGROUND**

The use of computer systems in decision-making, prediction and recommendation has been a trending topic of research for more than a decade. The recent advances in medical science can be attributed to advances in computer technology. However, the prediction of medical behaviour is still a very challenging task, which is done with the help of a medical professional. The occurrence of every disease shows a pattern based on its symptoms. The focus of this project is to propose a system to exploit these patterns for predicting the associated diseases and facilitating doctors for their treatment. The core idea behind this was that every symptom of a disease has a unique impact on severity .By prediction we mean to forecast an occurrence of a condition based on some mathematical calculation. For implementing this prediction, we need a recommender system. A recommender system is a system, which reads an input, finds a pattern in it, which is based on the dataset given to train the system. Based on the pattern it figures out a solution for the problem.

A naïve solution can be to create a database of every possible disease and its symptoms and predicting diseases based on that. The biggest drawback about this solution is that the efficiency and speed of this solution are very less and the size of this dataset would be very large. The solution that we suggest is, of using the symptoms with the ratings given by the patient, to predict the possible diseases. The main objective of this project is to develop an Intelligent System using data mining modelling technique, namely, Naive Bayes. It is implemented as web based application in this user can enter the disease symptoms and update the past medical history if any. It retrieves hidden data from stored database and compares the user values with trained data set. It can answer complex queries for diagnosing disease and thus assist patients to make intelligent clinical decisions, which traditional decision support systems cannot. By providing effective treatments with authorised doctors, it also helps to reduce treatment costs.

* 1. **PROJECT SCOPE**

Here the scope of the project is that integration of clinical appointments support with computer-based patient records could reduce time consumption, patient’s risk, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modelling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment, which can help to significantly improve the quality of clinical choices .The System is developed using Naive Bayesian Classification technique. The system extracts hidden knowledge from a disease database. This is the most effective model to predict patient’s disease of patients according to the provided symptoms. This model develop a prototype Disease Prediction System using data mining modelling techniques, namely, Naïve Bayes. So it provides effective treatments, it also helps to reduce treatment costs and enhances visualization and ease of interpretation. With immense knowledge and accurate data in that field. Large corporations invest heavily in this kind of activity to help focus attention on possible events and risks that are involved. Such work brings together all available past and current data, as a basis on which to develop reasonable expectations about the future.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 NOVEL METHOD FOR DISEASE RECOGNITION**

**AND CURE TIME PREDICTION BASED ON SYMPTOMS**

Healthcare is a sector where decisions usually have very high-risk and high-cost associated with them. One bad choice can cost a person’s life. With diseases like Swine Flu on the rise, which have symptoms quite similar to common cold, it is very difficult for people to differentiate between medical conditions. Here propose a novel method for recognition of diseases and prediction of their cure time based on the symptoms. We do this by assigning different coefficients to each symptom of a disease, and filtering the dataset with the severity score assigned to each symptom by the user. The diseases are identified based on a numerical value calculated in the fashion mentioned above. For predicting the cure time of a disease, we use reinforcement learning.

The algorithm takes into account the similarity between the condition of the current user and other users who have suffered from the same disease, and uses the similarity scores as weights in prediction of cure time. We also predict the current medical condition of user relative to people who have suffered from same disease by prediction we mean to forecast an occurrence of a condition based on some mathematical calculation. For implementing this prediction, we need a recommender system. A recommender system is a system, which reads an input, finds a pattern in it, which is based on the dataset given to train the system. Based on the pattern it figures out a solution for the problem.

A naive solution can be used to create a database of every possible disease and its symptoms and predicting diseases based on that. The biggest drawback about this solution is that the efficiency and speed of this solution are very less and the size of this dataset would be very large. The solution that we suggest is, of using the symptoms with the ratings given by the patient, to predict the possible diseases and the possible cure time of these diseases.

Our solution is novel and better because we predict the diseases based on the severity of the patients symptoms and the cure time prediction is based on real-life data given by other patients. For an accurate prediction, we give different coefficients to all the symptoms possible for a disease.

**2.2 PREDICTION SYSTEM FOR DISEASE USING NAIVE BAYES**

In this fast moving world people want to live a very luxurious life so they work like a machine in order to earn lot of money and live a comfortable life therefore in this race they forget to take care of themselves, because of this their food habits change, their entire lifestyle change, in this type of lifestyle they are more tensed they have blood pressure, sugar at a very young age and they don’t give enough rest for themselves and eat what they get and they even don’t bother about the quality of the food if sick then go for their own medication as a result of all these small negligence it leads to a major threat that is the heart disease. It is a world known fact that heart is the most essential organ in human body if that organ gets affected then it also affects the other vital parts of the body. Therefore, it is very important for people to go for a heart disease diagnosis. Because of this, people go to healthcare practitioners but the prediction made by them is not 100% accurate.

Quality service implies diagnosing patients correctly and administering treatments are effective. Poor clinical decisions can lead to disastrous consequences, which are therefore unacceptable. Hospitals must also minimize the cost of clinical tests. They can achieve these results by employing appropriate computer-based information and/or decision support systems. The healthcare industry collects huge amounts of healthcare data, which, unfortunately, are not “mined” to discover hidden information for effective decision-making. Discovery of hidden patterns and relationships often goes unexploited. Advanced data mining techniques can help as a remedy for this situation. This research has developed a prototype Heart Disease Prediction System (HDPS) using data mining techniques, namely, Decision Trees, Naïve Bayes and Neural Network. Results show that each technique has its unique strength in realizing the objectives of the defined mining goals. HDPS can answer complex what if queries, which traditional decision support systems cannot.

**CHAPTER 3**

**DESIGN METHOD**

**3.1 SEARCH DOCTOR**

This application enables the user to search for the doctors available depending upon the requirement i.e. the symptoms/disease the patient has. Various doctors for various specializations related to important parts of the body like heart, kidney, brain, liver etc. could be found on this application with the doctor’s availability, contact details.

**3.2 ONLINE APPOINTMENT**

After undergoing the registration, login procedures the user can find the required doctor with his scheduled timings, contact details the user can get an appointment from the doctor easily by this application.

**3.3 ADD DOCTORS**

An admin is the one who can view both the patients and the doctors with this application. The admin can add further more doctors depending upon the requirement and feedback given by the users. The details given by the user and the doctors can be edited or updated anytime.

**3.4 COLLECT FEEDBACK**

Every user of the application can provide certain feedback of the doctor/application after logging in this helps to continuously monitor and improve the efficiency of the application and it can satisfy every user depending upon the requirements. Admin can check the feedbacks provided by the user and act accordingly.

**3.5 SYSTEM DESIGN**

**3.5.1 SYSTEM ARCHITECTURE DESIGN**

In this project, it is discussed majorly about the requirements, architecture, and usage of the proposed android mobile application. The patients, doctors, use the application and the admin so the system can be viewed in three perspectives from various users (patients, doctors and the admin) the system architectures are depicted below.

**3.5.1.1 Patient’s Perspective**

****

**Fig 3.1 Patient’s Perspective**

The user can access or use the various features integrated by installing the android application into the mobile, the home page of the application consists of the register and login options, where the new user needs to register by providing few details like name, phone number, e-mail id. In addition, a registered user will be directed to the options page where various options like his/her details which can be edited, Diseases can be searched with the symptoms seen, doctor’s ID or name or address can search various doctors. A user can give feedback after the usage and logging out from this page directs to the home page i.e. the page, which is seen immediately after opening the application.

**3.5.1.2 Admin Perspective**

****

**Fig 3.2 Admin Perspective**

An admin is the one who coordinates both the users and the doctors by providing necessary details for both of them. Admin can add the doctors to the application depending on the requirement and the feedback sent by the users, Admin is the only one who can add doctors to the application. The details of ID, name, phone number and address of the doctor must be entered while adding a doctor into the application. Further modules of the application are view various doctors, patients, diseases given in the form of symptoms by the user, a disease can be added into the application too that enables the user to find out the doctors specialized to treat that disease, Admin can check the details by using view feedback module in the application.

**3.5.1.3. Doctor’s Perspective**

****

**Fig 3.3 Doctor’s Perspective**

Another user of the application is a doctor. A doctor module consists of doctors details, which can be updated by the doctor in case any contact info/address of the doctor is changed. Notifications enable the doctor to check the id of the patients who need a doctor’s help. View disease module enables the doctor to view the patient ID and the symptoms provided by the user, View patient module enable doctor to view various patient’s details who have used the application including their details like ID, name, contact details etc. using the application.

**3.6 METHODOLOGY**

**3.6.1 SEARCH HOSPITAL**

In this paper we mainly focus on the how to find the doctors who are specialized to treat a particular disease that too in a shorter span. The doctors who are added to the application can be viewed by various patients including their contact information, availability of the doctor etc. this whole searching process is said to be carried out by the data which has been stored earlier by the admin which can be also be updated at any desired time.

**3.6.2 IMPROVISE APPLICATION**

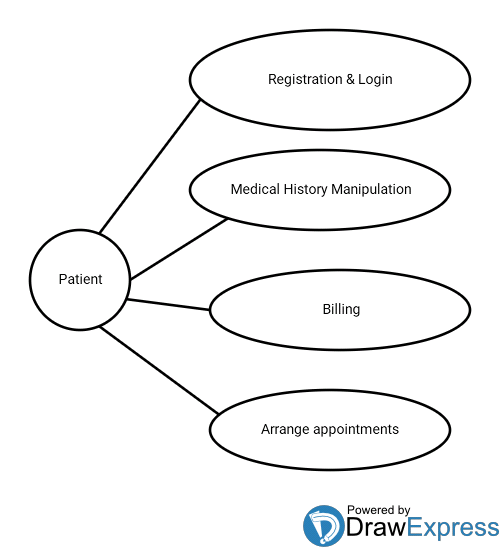
The feedback module, which is provided to the user or the patients, enables to gather various feedbacks given by vivid users of the application. Both the doctors and the admin. Can view this module. The doctors can improve their services. Whereas, an admin can further more add doctors as per the requirement of the users (patients).

**3.6.3 SQLITE DATABASE**

SQLite database is a Relational database management system (RDBMS) which is used to both add and retrieve the data from the database or the stored location (In our proposed application SD card acts as the storage location), SQLite database can be used both as both web and android application.

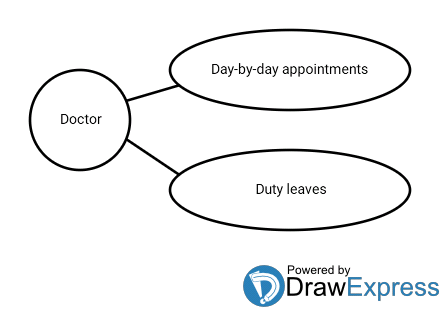
**3.7 USE CASE DIAGRAMS**

**3.7.1 PATIENT USE CASE DIAGRAM**



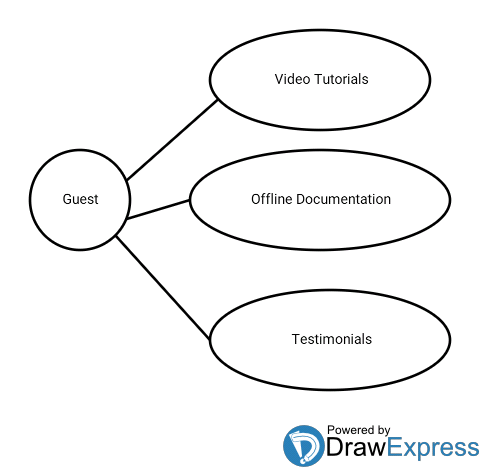
**Fig 3.4 Patient UCD**

**3.7.2 DOCTOR USE CASE DIAGRAM**



**Fig 3.5 Doctor UCD**

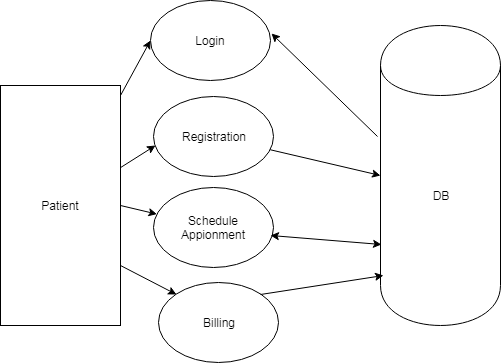
**3.7.3 GUEST USE CASE DIAGRAM**



**Fig 3.6 Guest UCD**

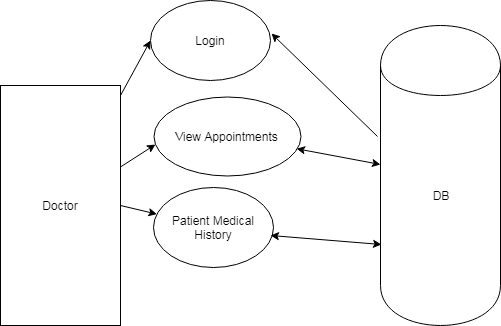
**3.8 DATA FLOW DIAGRAMS**

**3.8.1 PATIENT DATA FLOW DIAGRAM**



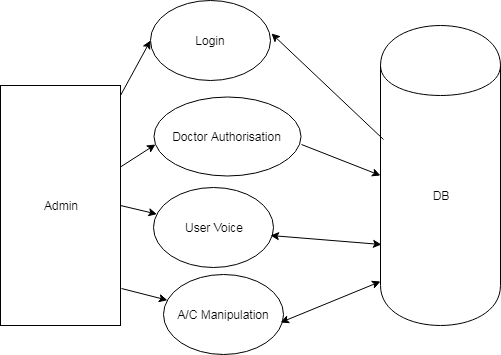
**Fig 3.7 Patient DFD**

**3.8.2 DOCTOR DATA FLOW DIAGRAM**



**Fig 3.8 Doctor DFD**

**3.8.3 ADMIN DATA FLOWDIAGRAM**



**Fig 3.9 Admin DFD**

**3.9 SYSTEM REQUIREMENTS**

**3.9.1 HARDWARE REQUIREMENTS**

Client Side

* Android Smart Phone

Server Side

* 2 GHZ Processor
* 4 GB RAM
* 160 GB HDD
* Monitor, Keyboard & Mouse
  + 1. **SOFTWARE REQUIREMENTS**

Client Side

* At least API 19 (KitKat)

Server Side

* Python with ML Libraries (Naive - Bias)
* MySQL
* Google Datasets
* Any Server OS
  + - 1. **Android KitKat Mobile OS**



**Fig 3.10 Android KitKat Logo**

Android is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) developed by [Google](https://en.wikipedia.org/wiki/Google), based on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source](https://en.wikipedia.org/wiki/Open-source_software) software and designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). In addition, Google has further developed [Android TV](https://en.wikipedia.org/wiki/Android_TV) for televisions, [Android Auto](https://en.wikipedia.org/wiki/Android_Auto) for cars, and [Wear OS](https://en.wikipedia.org/wiki/Wear_OS) for wristwatches, each with a specialized user interface. Variants of Android are also used on [game consoles](https://en.wikipedia.org/wiki/Video_game_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [PCs](https://en.wikipedia.org/wiki/Personal_computer) and other electronics. Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the [first commercial Android device](https://en.wikipedia.org/wiki/HTC_Dream) launched in September 2008. The operating system has since gone through multiple major releases, with the current version being [9 "Pie"](https://en.wikipedia.org/wiki/Android_Pie), released in August 2018. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the [Apache License](https://en.wikipedia.org/wiki/Apache_License).



**Fig 3.11 Android KitKat UI**

Android is also associated with a suite of [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) developed by Google, called [Google Mobile Services](https://en.wikipedia.org/wiki/Google_Mobile_Services) (GMS) that very frequently comes pre-installed in devices, which usually includes the [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) web browser and [Google Search](https://en.wikipedia.org/wiki/Google_Search_(mobile_app)) and always includes core apps for services such as [Gmail](https://en.wikipedia.org/wiki/Gmail), as well as the [application store](https://en.wikipedia.org/wiki/Application_store) and [digital distribution](https://en.wikipedia.org/wiki/Digital_distribution) platform [Google Play](https://en.wikipedia.org/wiki/Google_Play), and associated [development platform](https://en.wikipedia.org/wiki/Google_Play_Services). These apps are licensed by manufacturers of Android devices certified under standards imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com)'s [Fire OS](https://en.wikipedia.org/wiki/Fire_OS), which use their own equivalents to GMS. Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest [installed base](https://en.wikipedia.org/wiki/Installed_base) of any operating system, and as of June 2018, the [Google Play](https://en.wikipedia.org/wiki/Google_Play) store features over 3.3 million apps.

Android "KitKat" is a codename for the [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) and the eleventh version of Android. Unveiled on September 3, 2013, KitKat focused primarily on optimizing the [operating system](https://en.wikipedia.org/wiki/Operating_system) for improved performance on entry-level devices with limited resources. As of November 2018, statistics issued by Google indicate that 7.6% of all Android devices accessing Google Play run KitKat.

The overall interface of KitKat further downplays the "Holo" interface appearance introduced on 4.0, replacing remaining instances of blue accenting with greys and white (such as the status bar icons). Apps may trigger a translucent status and navigation bar appearance, or trigger a full screen mode ("Immersive mode") to hide them entirely. The launcher also received a refreshed appearance, with the implementation of the translucent navigation bars, and the replacement of the black backdrop in the application drawer with a translucent backdrop. Additionally, action overflow menu buttons in apps are always visible, even on devices with the deprecated "Menu" navigation key. In the Settings menu, users can now specify a default Home (launcher) and [text messaging](https://en.wikipedia.org/wiki/Short_Message_Service) app. On stock devices, the Messaging and Movie Studio apps were removed; the former was replaced by [Google Hangouts](https://en.wikipedia.org/wiki/Google_Hangouts), which supported SMS. The AOSP Gallery app was also deprecated in favour of [Google+ Photos](https://en.wikipedia.org/wiki/Google_Photos). In certain apps, a new "Documents" file picker is available.

A new [runtime environment](https://en.wikipedia.org/wiki/Runtime_environment) known as the [Android Runtime](https://en.wikipedia.org/wiki/Android_Runtime) (ART), intended to replace the [Dalvik virtual machine](https://en.wikipedia.org/wiki/Dalvik_virtual_machine), was introduced as a [technology preview](https://en.wikipedia.org/wiki/Technology_preview) in KitKat. ART is a cross-platform runtime which supports the [x86](https://en.wikipedia.org/wiki/X86), [ARM](https://en.wikipedia.org/wiki/ARM_architecture), and [MIPS](https://en.wikipedia.org/wiki/MIPS_architecture) architectures in both [32-bit](https://en.wikipedia.org/wiki/32-bit) and [64-bit](https://en.wikipedia.org/wiki/64-bit) environments. Unlike Dalvik, which uses [just-in-time compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation) (JIT), ART compiles apps [upon installation](https://en.wikipedia.org/wiki/Ahead-of-time_compilation), which are then run exclusively from the compiled version from then on. This technique removes the processing overhead associated with the JIT process, improving system performance. Devices with 512 MB of RAM or less report themselves as "low RAM" devices. Using an API, apps may detect low RAM devices and modify their functionality accordingly. KitKat also supports [zram](https://en.wikipedia.org/wiki/Zram). WebView components were updated to utilize a version of the [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) [rendering engine](https://en.wikipedia.org/wiki/Blink_(layout_engine)).

* + - 1. **Python Programming Language**

Python logo and wordmark.svg

**Fig 3.12 Python Logo**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level programming and scripting language](https://en.wikipedia.org/wiki/High-level_programming_language) for [general-purpose programming](https://en.wikipedia.org/wiki/General-purpose_programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python has a design philosophy that emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability), notably using [significant whitespace](https://en.wikipedia.org/wiki/Significant_whitespace). It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years. Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), named variables, default parameters including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming) [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library). Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). [CPython](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation).

**Python Machine Learning Libraries**

Machine learning is the study of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [mathematical models](https://en.wikipedia.org/wiki/Mathematical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to progressively improve their performance on a specific task. Machine learning algorithms build a mathematical model of sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in the applications of [email filtering](https://en.wikipedia.org/wiki/Email_filtering), detection of network intruders, and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.wikipedia.org/wiki/Data_mining) is a field of study within machine learning, and focuses on [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) through [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning). In its application across business problems, machine learning is also referred to as [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics).

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **Naive Bayes Classifiers** are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classifier)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naive) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features. Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the [text retrieval](https://en.wikipedia.org/wiki/Information_retrieval) community in the early 1960s, and remains a popular (baseline) method for [text categorization](https://en.wikipedia.org/wiki/Text_categorization), the problem of judging documents as belonging to one category or the other (such as [spam or legitimate](https://en.wikipedia.org/wiki/Spam_filtering), sports or politics, etc.) with [word frequencies](https://en.wikipedia.org/wiki/Bag_of_words) as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine). It also finds application in automatic [medical diagnosis](https://en.wikipedia.org/wiki/Medical_diagnosis). Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. [Maximum-likelihood](https://en.wikipedia.org/wiki/Maximum-likelihood_estimation) training can be done by evaluating a [closed-form expression](https://en.wikipedia.org/wiki/Closed-form_expression), which takes [linear time](https://en.wikipedia.org/wiki/Linear_time), rather than by expensive [iterative approximation](https://en.wikipedia.org/wiki/Iterative_method) as used for many other types of classifiers. In the [statistics](https://en.wikipedia.org/wiki/Statistics) and [computer science](https://en.wikipedia.org/wiki/Computer_science) literature, naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a [Bayesian](https://en.wikipedia.org/wiki/Bayesian_probability) method.

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. Bayes’ theorem states the following relationship, given class variable y and dependent feature vector x1 through xn:



The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of P (xi∣y). In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering. They require a small amount of training data to estimate the necessary parameters. (For theoretical reasons why naive Bayes works well, and on which types of data it does, see the references below.) Naive Bayes learners and classifiers can be extremely fast compared to methods that are more sophisticated. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. On the other side, although naive Bayes is known as a decent classifier, it is known to be a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.

[**Scikit-learn**](http://scikit-learn.org/stable/user_guide.html) is one the most popular ML libraries. It supports many supervised and unsupervised learning algorithms. Examples include linear and logistic regressions, decision trees, clustering, and k-means and so on. It builds on two basic libraries of Python, NumPy and SciPy. It adds a set of algorithms for common machine learning and data mining tasks, including clustering, regression and classification. Even tasks like transforming data, feature selection and ensemble methods can be implemented in a few lines. For a novice in ML, Scikit-learn is a more-than-sufficient tool to work with, until you start implementing algorithms that are more complex.

The interesting thing about [**Tensorflow**](https://www.tensorflow.org/)is that when you write a program in Python, you can compile and run on either your CPU or GPU. Therefore, you do not have to write at the C++ or CUDA level to run on GPUs. It uses a system of multi-layered nodes that allows you to quickly set up, train, and deploy artificial neural networks with large datasets. This is what allows Google to identify objects in photos or understand spoken words in its voice-recognition app.

[**Theano**](http://www.deeplearning.net/software/theano/)is another good Python library for numerical computation, and is similar to NumPy. Theano allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. What sets Theano apart is that it takes advantage of the computer’s GPU. This allows it to make data-intensive calculations up to 100 times faster than when run on the CPU alone. Theano’s speed makes it especially valuable for deep learning and other computationally complex tasks. The final release of Theano library was last year — 2017, version 1.0.0 with a lot of new features, interface changes and improvements.

**Pandas** is a very popular library that provides high-level data structures, which are simple to use as well as intuitive. It has many inbuilt methods for grouping, combining data and filtering as well as performing time series analysis. Pandas can easily fetch data from different sources like SQL databases, CSV, Excel, JSON files and manipulate the data to perform operations on it.

The best and most sophisticated ML is meaningless if you cannot communicate it to other people. So how do you actually turn around value from all data that you have? How do you inspire your business analysts and tell them “stories” full of “insights”? This is where [**Matplotlib**](https://matplotlib.org/tutorials/index.html)comes to the rescue. It is a standard Python library used by every data scientist for creating 2D plots and graphs. It is low-level, meaning it requires more commands to generate nice-looking graphs and figures than with some advanced libraries. However, the other side of that is flexibility. With enough commands, you can make just about any kind of graph you want with Matplotlib. You can build diverse charts, from histograms and scatterplots to non-Cartesian coordinates graphs. It supports different GUI back ends on all operating systems, and can export graphics to common vector and graphic formats like PDF, SVG, JPG, PNG, BMP, GIF, etc.

* + - 1. **MySQL**

MySQL the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius's daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, and Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

**BASIC COMMANDS**

**i) USE database\_name**

Change to this database. You need to change to some database when you first connect to MySQL.

**ii) SHOW DATABASES**

Lists all MySQL databases on the system.

**iii) SHOW TABLES [FROM database\_name]**

Lists all tables from the current database or from the database given in the command.

**iv) CREATE Tables**

CREATE TABLE table\_name (column\_name1 data\_type (size), column\_name2 data\_type (size), column\_name3 data\_type (size) ...);

**v) DROP TABLE table\_name**

Removes the table from the database. Permanently! So be careful with this command!

**vi) INSERT INTO Tables**

INSERT [INTO] table\_name (column\_name1, column\_name2 ...)

VALUES (value1, value2...);

**vii) DELETE FROM table\_name WHERE where\_clause**

Delete rows that meet the conditions of the where\_clause. If the WHERE statement is omitted, the table is emptied, although its structure remains intact.

**viii) UPDATE Tables**

UPDATE table\_name SET column\_name1=value1, column\_name2=value2 ... [WHERE where\_clause];

Alters the data within a column based on the conditions in the where\_clause.

**ix) SELECT Table Data**

SELECT column\_name1, column\_name2, /\* ... FROM table\_name;

* + - 1. **Google Datasets**

Google’s goal has always been to organize the world’s information, and its first target was the commercial web. Now, it wants to do the same for the scientific community with a new search engine for datasets. The service, called [Dataset Search](https://www.blog.google/products/search/making-it-easier-discover-datasets/), launches today, and will be a companion of sorts to [Google Scholar](https://scholar.google.com/), the company’s popular search engine for academic studies and reports. Institutions that publish their data online, like universities and governments, will need to include [metadata tags](https://schema.org/) in their webpages that describe their data, including who created it, when it was published, how it was collected, and so on. This information will then be indexed by Dataset Search and combined with input from Google’s Knowledge Graph. (That is the name for those boxes that pop up for common searches.

So if dataset X was published by CERN, some info about the institute will also be included in the results.) The initial release of Dataset Search will cover the environmental and social sciences, government data, and datasets from news organizations like *ProPublica*. However, if the service becomes popular, the amount of data it indexes should quickly snowball as institutions and scientists scramble to make their information accessible. The recent flourishing of open data initiatives around the world should help this. “I do think in the last several years the number of repositories has exploded,” says Noy. She credits this to the increasing importance of data in scientific literature, which means journals ask authors to publish datasets, as well as “government regulations in the US and Europe and the general rise of the open data movement.”

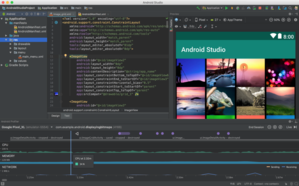
Having Google involved should help make this project a success, says Jeni Tennison, CEO of the Open Data Institute (ODI). “Dataset search has always been a difficult thing to support, and I’m hopeful that Google stepping in will make it easier,” she says. To create a decent search engine, you need to know how to build user-friendly systems and understand what people mean when they type in certain phrases, says Tennison. Google knows what it is doing in both of those departments. In fact, says Tennison, ideally Google will publish its own dataset on how Dataset Search gets used. Although the metadata tags the company is using to make datasets visible to its search crawlers are an open standard (meaning that any competitor, like Bing or Yandex, can use them to build their own competing service), search engines improve most quickly when a critical mass of users is there to provide data on what they’re doing. “Simply understanding how people search is important... what kind of terms they use, how they express them,” says Tennison. “If we want to get to grips with how people search for data and make it more accessible, it would be great if Google opened up its *own* data on this.” In other words: Google should publish a dataset about dataset search that would be indexed by Dataset Search.

* + - 1. **Server OS**

A server operating system, also called a server OS, is an [operating system](https://www.webopedia.com/TERM/O/operating_system.html) specifically designed to run on [servers](https://www.webopedia.com/TERM/S/server.html), which are specialized computers that operate within a [client/server architecture](https://www.webopedia.com/TERM/C/client_server_architecture.html) to serve the requests of client computers on the network. The server operating system, or server OS, is the software layer on top of which other software programs, or [applications](https://www.webopedia.com/TERM/A/application.html), can run on the server [hardware](https://www.webopedia.com/TERM/H/hardware.html). Server operating systems help enable and facilitate typical server roles such as Web server, mail server, file server, database server, application server and print server.

**3.9.3 DEVELOPMENT REQUIREMENTS**

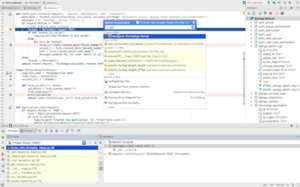
**Android Studio**



**Fig 3.13 Android Studio UI**

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [Jet Brains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development. Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8, which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.2.1, which was released in October 2018.

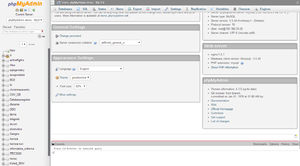
**PyCharm**



**Fig 3.14 PyCharm UI**

PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) language. The Czech company develops it [Jet Brains](https://en.wikipedia.org/wiki/JetBrains). It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)). PyCharm is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), with [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) versions. The Community Edition is released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License), and there is Professional Edition with extra features, released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software). The beta version was released in July 2010, with the 1.0 arriving 3 months later. Version 2.0 was released on 13 December 2011, version 3.0 on 24 September 2013, and version 4.0 on November 19, 2014. PyCharm Community Edition, the open source version of PyCharm, became available on 22 October 2013.

**phpMyAdmin**



**Fig 3.15 phpMyAdmin UI**

phpMyAdmin is a [free and open source](https://en.wikipedia.org/wiki/Free_and_open_source) administration tool for [MySQL](https://en.wikipedia.org/wiki/MySQL) and [MariaDB](https://en.wikipedia.org/wiki/MariaDB). As a portable [web application](https://en.wikipedia.org/wiki/Web_application) written primarily in [PHP](https://en.wikipedia.org/wiki/PHP), it has become one of the most popular MySQL administration tools, especially for [web hosting services](https://en.wikipedia.org/wiki/Web_hosting_service). Tobias Ratschiller, then an IT consultant and later founder of the software company Maguma, started to work on a PHP-based web [front-end](https://en.wikipedia.org/wiki/Front-end_and_back-end) to MySQL in 1998, inspired by MySQL-Webadmin. He gave up the project (and [phpAdsNew](https://en.wikipedia.org/wiki/OpenX_(software)), of which he was also the original author) in 2000 because of lack of time. By that time, phpMyAdmin had already become one of the most popular PHP applications and MySQL administration tools, with a large community of users and contributors. In order to coordinate the growing number of patches, a group of three developers registered the phpMyAdmin Project at [Source Forge](https://en.wikipedia.org/wiki/SourceForge) & took the development in 2001. 

**MySQL Workbench**



**Fig 3.16 MySQL Workbench UI**

It  is a visual [database design](https://en.wikipedia.org/wiki/Database_design) tool that integrates [SQL](https://en.wikipedia.org/wiki/SQL) [development](https://en.wikipedia.org/wiki/Software_development), [database design](https://en.wikipedia.org/wiki/Database_design), creation and maintenance into a single IDE for the [MySQL](https://en.wikipedia.org/wiki/MySQL) database system. It is the successor to DB Designer 4 from fabFORCE.net, and replaces the previous package of software, [MySQL GUI Tools Bundle](https://en.wikipedia.org/wiki/MySQL_Workbench#MySQL_GUI_Tools_Bundle). MySQL Workbench is the first MySQL family of products that offer two different editions - an open source and a proprietary edition. The "Community Edition" is a full featured product that is not crippled in any way. Being the foundation for all other editions it will benefit from all future development efforts. The proprietary "Standard Edition" extends the Community Edition with a series of modules and plugins. As Sun Microsystems announced this business decision soon after the takeover of MySQL, this has caused speculation in the press about the future licensing of the MySQL database.

**CHAPTER 4**

**DESIGN EXAMPLE**



**Fig 4.1 User activities**

User activities is the page displayed immediately after logging in by user where he can get details about various diseases and doctor, doctors can be searched based on id, name & address . Here the user can provide feedback.



**Fig 4.2 Admin activities**

Here, the admin can view various patients who used the application, Can view various doctors, Admin can also add doctors to the application depending on the requirement and feedback given by the users. Admin can access the details of both the doctors and the patients.



**Fig 4.3 Doctor Activities**

Here, the doctor can view various patients who used the application; the doctor can check the notifications to know about the disease of the patients. The details of the doctor can be updated/edited anytime to ensure the appointment of patients.

**CHAPTER 5**

**CONCLUSION**

The “Smart health consultation” Android Application is helpful for patient to search the hospital based on specialist. This application is simplify the task of patient and doctor. This application facilitates the interaction between patient and doctor. It helps to optimize the work of patient and doctor. Installation of the app in the Smartphone is quite simple and more useful to patients who have normal idea of android mobile. Smart health consulting android system is an effortless, efficient and influential mobile application for the society.

As we have already mentioned, we address the problem of predicting diseases using “Naïve Bayes algorithm” and their respective Doctor’s appointment time based on the symptoms. The focus was on the classification of symptoms based on their severity and importance and using this knowledge to calculate a numerical value to identify diseases. Although the method was tested in a limited environment with high accuracy, it can be extended to larger settings. We also provide a severity rating for the current condition, relative to the other users with similar symptoms. The test results for various medical conditions can be used to further improve the reliability of the system. Since the results are dependent on the experience of previous users, it is important to isolate genuine experiences from fake ones.

**REFERENCES**

1. Wikipedia. List of medical symptoms, 2015. [Online; accessed 22- January-2015].
2. WebMD. Disease symptoms, 2015. [Online; accessed 22-January- 2015].
3. Siri Krishan Wasan, Vasudha Bhatnagar, and Harleen Kaur. The impact of data mining techniques on medical diagnostics. Data Science Journal: 119–126, 2006.
4. Roshawnna Scales and Mark Embrechts. Computational intelligence techniques for medical diagnostics. In Proceedings of Walter Lincoln Hawkins, Graduate Research Conference from the World Wide Web, 2002.
5. Yanwei, X.; Wang, J.; Zhao, Z.; Gao, Y., “Combination data mining models with new medical data to predict outcome of coronary heart disease”. Proceedings International Conference on Convergence Information Technology 2007.
6. Sellappan Palaniappan, Rafiah Awang, "Intelligent Heart Disease Prediction System Using Data Mining Techniques", IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.8, August 2008.
7. Niti Guru, Anil Dahiya, Navin Rajpal, "Decision Support System for Heart Disease Diagnosis Using Neural Network", Delhi Business Review, Vol. 8 (June 2007).
8. HeonGyu Lee, Ki Yong Noh, KeunHoRyu, “Mining Bio signal Data: Coronary Artery Disease Diagnosis using Linear and Nonlinear Features of HRV,” LNAI 4819: Emerging Technologies in Knowledge Discovery and Data Mining, May 2007.
9. Shantakumar B.Patil, Y.S.Kumaraswamy “Intelligent and Effective Heart Attack Prediction System Using Data Mining and Artificial Neural Network”. ISSN 1450-216X Vol.31 No.4 (2009), pp.642-656.