

CHAPTER 1

INTRODUCTION

1.1 What is Machine learning?

Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithms it uses to determine correct answer.

1.2 How Machine Learning works?

Machine learning is a form of artificial intelligence (AI) that teaches computers to think in a similar way to how humans do: learning and improving upon past experiences. It works by exploring data, identifying patterns, and involves minimal human intervention. Almost any task that can be completed with a data-defined pattern or set of rules can be automated with machine learning.

Machine learning uses two main techniques:

- **Supervised learning** allows you to collect data or produce a data output from a previous ML deployment. Supervised learning is exciting because it works in much the same way humans actually learn.

In supervised tasks, we present the computer with a collection of labeled data points called a training set.

Examples:

- **Neural Networks:** A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
- **Decision Trees:** The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).
- **Naive Bayes:** Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.
- **K Nearest Neighbor:** The 'K' in KNN indicates the number of nearest neighbors, which are used to classify or predict outputs in a data set. It is a non-parametric algorithm which means it does not make any assumptions on underlying data.

- **Unsupervised machine learning** helps you find all kinds of unknown patterns in data. In unsupervised learning, the algorithm tries to learn some inherent structure to the data with only unlabeled examples. Two common unsupervised learning tasks are clustering and dimensionality reduction. In clustering, we attempt to group data points into meaningful clusters such that elements within a given cluster are similar to each other but dissimilar to those from other clusters.

Examples:

- **K-Mean Clustering:** It is a method of vector quantization originally from signal processing that ends to partition n observations into k clusters in which each observations belongs to the cluster with the nearest mean.
- **Principal Component Analysis(PCA):** It is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using the first few principal components.
- **Apriori Algorithm:** It is an algorithm for frequent item set mining and association rule learning over relational database. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

1.3 Characteristics of Machine learning

- (i) **The ability to perform automated data visualization:** Machine learning offers a number of tools that provide rich snippets of data which can be applied to both unstructured and structured data. With the help of user-friendly automated data visualization platforms in machine learning, businesses can obtain a wealth of new insights in an effort to increase productivity in their processes.
- (ii) **Automation at its best:** One of the biggest characteristics of machine learning is its ability to automate repetitive tasks and thus, increasing productivity. A huge number of organizations are already using machine learning-powered paperwork and email automation.
- (iii) **Customer engagement like never before:** Machine learning plays a critical role in enabling businesses and brands to spark more valuable conversations in terms of

customer engagement. The technology analyzes particular phrases, words, sentences, idioms, and content formats which resonate with certain audience members.

(iv) **The ability to take efficiency to the next level when merged with IoT:**

machine learning is probably the best technology that can be used to attain higher levels of efficiency. By merging machine learning with IoT, businesses can boost the efficiency of their entire production processes.

(v) **The ability to change the mortgage market:** With the help of machine learning, lenders can now obtain a more comprehensive consumer picture. They can now predict whether the customer is a low spender or a high spender and understand his/her tipping point of spending. Apart from mortgage lending, financial institutions are using the same techniques for other types of consumer loans.

(vi) **Accurate data analysis:** By developing efficient and fast algorithms, as well as, data- driven models for processing of data in real-time, machine learning is able to generate accurate analysis and results.

(vii) **Business intelligence at its best:** Machine learning characteristics, when merged with big data analytical work, can generate extreme levels of business intelligence with the help of which several different industries are making strategic initiative. Mature ML operators take their processes to a significantly higher level. Models have access restrictions. Code naturally should be tested, but it's clear to the mature ML operator that some amount of data testing is also critical to a well-functioning system. Mature ML operators perform data testing that monitors changes in the distribution of the data. The mature environment enables current objectives to reuse existing models. Rather than start from scratch, the MMLO can add features to distinguish a new model. ML processes include the use of a repository for models and robust model packaging, deployment, serving, and monitoring.

1.4 Python Programming Language

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy.

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

1.4.1 What can Python technology do?

1) Readable and Maintainable Code: The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional

code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

2) Multiple Programming Paradigms: Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming.

3) Compatible with Major Platforms and Systems: At present, Python is supporting many operating systems. Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, not required to

recompile the code after making any alteration. Can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

4) Robust Standard Library: Its large and robust standard library makes Python score over other programming languages. The standard library allows to choose from a wide range of modules according to user precise needs. Each module further enables to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols.

5) Many Open Source Frameworks and Tools: As an open source programming language, Python helps to curtail software development cost significantly. Can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. For instance, can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and CherryPy.

The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

6) Simplify Complex Software Development: Python is a general-purpose programming language. Hence, can use the programming language for developing both desktop and web applications. Also, can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization.

7) Adopt Test Driven Development: Python can be used to create prototype of the software application rapidly. Also, can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier to perform coding and testing simultaneously by adopting test driven development (TDD) approach.

CHAPTER 2

LITERATURE SURVEY

[1] “Prediction of Cardiovascular Disease Using Machine Learning Algorithms”, Kumar G Dinesh, K Arumugaraj, Kumar D Santhosh, V Mareeswari, June 2017

It contributes the correlative application and analysis of distinct machine learning algorithms in the R software which gives an immediate mechanism for the user to use the machine learning algorithms in R software for forecasting the cardiovascular diseases. The results show that the system has great potential in predicting the heart disease risk level more accurately. ID3 has some features like removing outliers, handling missing values and but their major disadvantage is to over-fitting. And it's not so easy to implement as that of Naïve Bayes algorithm.

[2] “Multi Disease Prediction Using Data Mining Techniques”, K. Gomathi Kamaraj, D. Shanmuga Priyaa, February 2019

In this study two different data mining classification techniques was used for the prediction of various diseases and their performance was compared in order to evaluate the best classifier. The Artificial Neural Network, K Means Clustering Algorithm and Frequent Item Set generation using Apriori Techniques are used to classify whether a patient suffers from heart disease or not. Up to now, several studies have been reported that have focused on cardiovascular disease diagnosis. These dies have applied different approaches to the given problem and achieved high classification vacancies of 77% or higher.

[3] “Prediction of Heart Disease Using Machine Learning Algorithms”, Santhana Krishnan J, Geetha S, April 2019

Here two supervised data mining algorithms was applied on the dataset to predict the possibilities of having heart disease of a patient, were analyzed with classification model namely Naïve Bayes Classifier and Decision tree classification. The Decision tree model has predicted the heart disease patient with an accuracy level of 91% and Naïve Bayes classifier has predicted heart disease patient with an accuracy level of 87%.

[4] “A Proposed Model for Lifestyle Disease Predict Vectorion Using Support Machine”, Rajeev D M, Shadab Arshad, January 2019

This study aims to understand support vector machine and use it to predict lifestyle diseases that an individual might be susceptible to. In the existing system the info set is often little, for patients and diseases with specific conditions. These systems are principally designed for the additional prodigious diseases like cardiovascular disease, Cancer etc. The pre-selected characteristics could generally not satisfy the changes within the malady and its influencing factors that may lead to quality in results. The main feature will be the machine learning, in which algorithm use are such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will help us in getting accurate predictions.

[5] “Review of Medical Disease Symptoms Prediction Using Data Mining Technique”, Rahul Deo Sah, Dr Jitendra Sheetalani, September 2018

It evaluates the performance of medical disease prediction based on data mining technique. Classification proceed based on classifier selection to medical disease data and propose a clustering-based classifier selection method. In the method, many clusters are selected for a ensemble process. Then, the standard presentation of each classifier on selected clusters is calculated and the classifier with the best average performance is chosen to classify the given data. Data Mining Technique: KNN Classifier, Rough Set Theory, SVM (Support Vector Machine).

[6] “A. L. Predicting Individual Disease Risk Based On Medical History ”, A.Davis, D, V.Chawla, Blumm, Christakis, & Barabasi (2008)

Darcy A. Davis, Nitesh V. Chawla, Nicholas Blumm, Nicholas Christakis, Albert-Laszlo Barabasi have found that global treatment of chronic disease is neither time or cost efficient. So the authors conducted this research to predict future disease risk. For this CARE was used (which relies only on a patient's medical history using ICD- 9-CM codes in order to predict future diseases risks). CARE combines collaborative filtering methods with clustering to predict each patient's greatest disease risks based on their own medical history and that of similar patients.

[7] “Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction.” Jyoti Soni, Ansari,U Sharma & Soni S March 2011

Jyoti Soni, Ujma Ansari, Dipesh Sharma and Sunita Soni have done this research paper into provide a survey of current techniques of knowledge discovery in databases using data mining techniques that are in use in today's medical research particularly in Heart Disease Prediction. Number of experiment has been conducted to compare the performance of predictive data mining technique on the same dataset and the outcome reveals that Decision Tree outperforms and some time Bayesian classification is having similar accuracy as of decision tree but other predictive methods like KNN, Neural Networks, Classification based on clustering is not performing well.

[8] “ Disease Predicting System Using Data Mining Techniques.”, Nishar Banu, MA Gomathy (2013)

M.A. Nishara Banu, B Gomathy used medical data mining techniques like association rule mining, classification, clustering I to analyze the different kinds of heart based problems. Decision tree is made to illustrate every possible outcome of a decision. Different rules are made to get the best outcome. In this research age, sex, smoking, overweight, alcohol intake, blood sugar, hear rate, blood pressure are the parameters used for making the decisions. Risk level for different parameter.

2.1 EXISTING SYSTEM

Existing system reviews concepts of Machine learning (ML), ML prediction model and applications, describes implementation of ML applications in health care.

The existing system proposes various features to be implemented in health care with various benefits. At first Machine learning algorithms are used to analyze medical data sets effectively in present. The main aim was to implement supervised machine learning concept by using datasets regarding blood cells collected from blood cells detecting and counting sensors, of a human as the input. This input was trained by artificial neural network algorithm and decision tree classification learning algorithm to perform classification. It resulted in recognizing and also possibly predict the disease based on the nature of blood cells and classify accordingly. Later The application of machine learning in the field of medical diagnosis is increasing gradually. This contributed primarily to the improvement in the classification and recognition systems used in disease diagnosis. Different classification algorithms are applied, on three separate databases of disease (Heart, Breast cancer, Diabetes) available in UCI repository for disease prediction. The feature selection for each dataset was accomplished by backward modeling using the p-value test. ML is playing a vital role in health care.

Implementation ML technologies to in health care as a part of prediction model helps doctors and patients to analyze the disease.

DISADVANTAGES OF EXISTING SYSTEM:

- The sub diseases fall under the same category. So, it becomes difficult to differentiate between diseases of the same category but of different types.
- The data sets includes the data of mostly the frequently occurring diseases so it leaves out rare diseases or less occurring diseases.
- No sorts of precautions available to patients in case they cannot immediately pay a visit to the doctors.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROPOSED SYSTEM

The proposed system focuses on overcoming all the disadvantages of existing system. Proposed system predicts all diseases and sub diseases which occur in society. The system combines the results using multi classifier into subsets and increase the accuracy. Collecting data from user using disease prediction model, where the user selects the symptoms which the user is experiencing. Dataset is obtained from Kaggle.com and various health related websites. Dataset consists of about 5000 rows, with consisting of 140 symptoms and around 50 diseases which can be predicted. We use different algorithms which increase the overall operational efficiency which include:

Decision Tree: generate pattern of disease and define disease and sub disease

Map Reduce: partition the medical data based on output of Decision tree.

Neural Networks: recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

The scheming efficiency of our system is quicker than that of the existing systems. when the user symptoms are received the symptoms are given to the trained model which consists of decision tree and Naïve Bayes algorithm in order to predict the output i.e. the predicted disease. After successful completion of machine learning algorithm, the disease predicted is displayed on the user screen with the probability of its occurrence. The system also recommends the specialist doctors of the disease and platform where patient can text to doctor and take appointments.

3.2 PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCS), and supports web development with Django as well as Data Science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux

versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

3.2.1 How to Create a New Project.

1. Select the file from the menu bar.
2. Select New Project.
3. Give the File Name.
4. Press create button. A project is created.

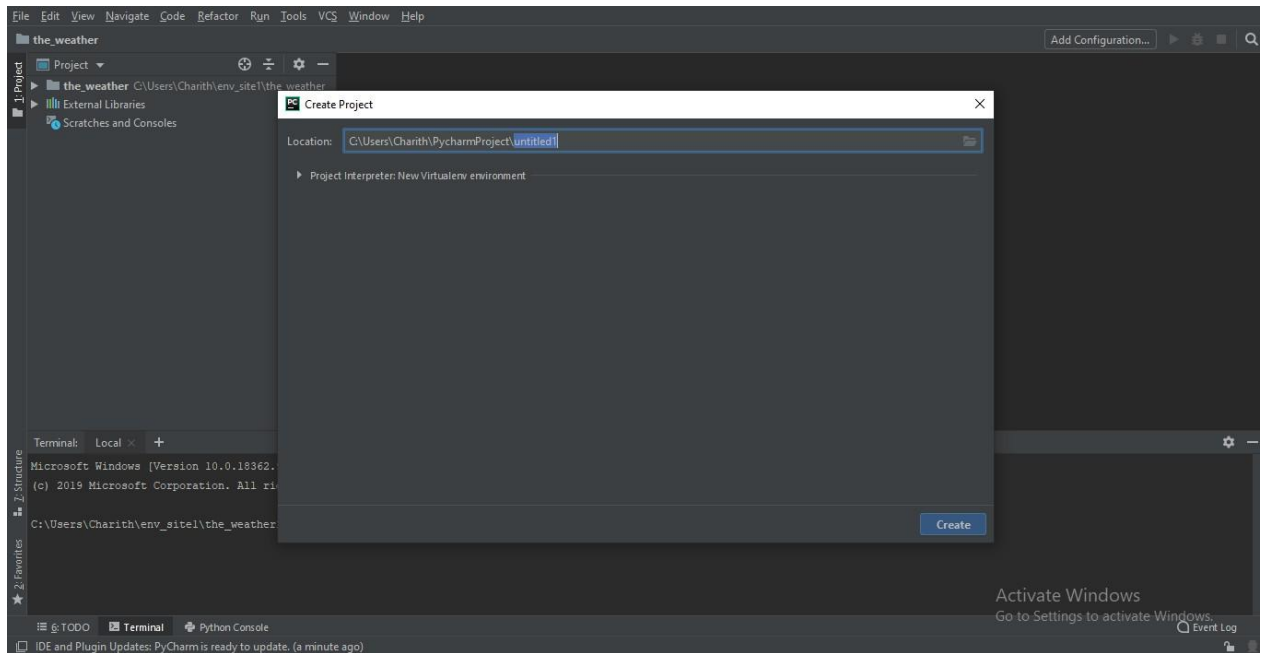


FIG 3.1: PROJECT CREATION

3.2.2 Running the program

1. open the terminal.
2. run the command “python manage.py runserver”.
3. An url will be displayed on clicking it opens the web page of the project.

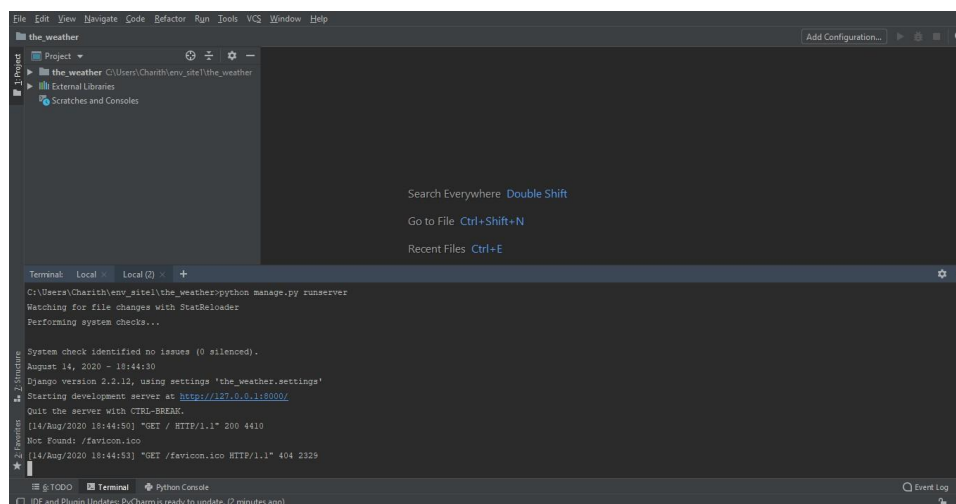


Fig 3.2: RUNNING THE PROGRAM

3.3 Django

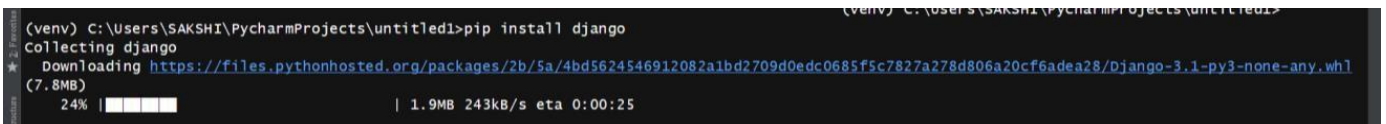
Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django is a Python-based free and open-source web framework that follows the model-view-controller (MVC) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

3.3.1 How to install Django

1. open pycharm and run the terminal
2. type the command “pip install django”

A screenshot of a terminal window with a dark background. The prompt is '(venv) C:\Users\SAKSHI\PycharmProjects\untitled1>'. The command entered is 'pip install django'. The output shows 'Collecting django' followed by a star icon and 'Downloading https://files.pythonhosted.org/packages/2b/5a/4bd5624546912082a1bd2709d0edc0685f5c7827a278d806a20cf6adea28/Django-3.1-py3-none-any.whl (7.8MB)'. Below this is a progress bar at 24% and the text '| 1.9MB 243kB/s eta 0:00:25'.

```
(venv) C:\Users\SAKSHI\PycharmProjects\untitled1>pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/2b/5a/4bd5624546912082a1bd2709d0edc0685f5c7827a278d806a20cf6adea28/Django-3.1-py3-none-any.whl (7.8MB)
    24% |██████████| 1.9MB 243kB/s eta 0:00:25
```

FIG 3.3: INSTALLATION OF DJANGO

3.3.2 Setting up the project

1. Once Django is installed open the terminal
2. Type the command Django- admin startproject Project_name
3. Django by default creates some files like settings.py views.py urls.py and manage.py model.py Which we will use to create our project.

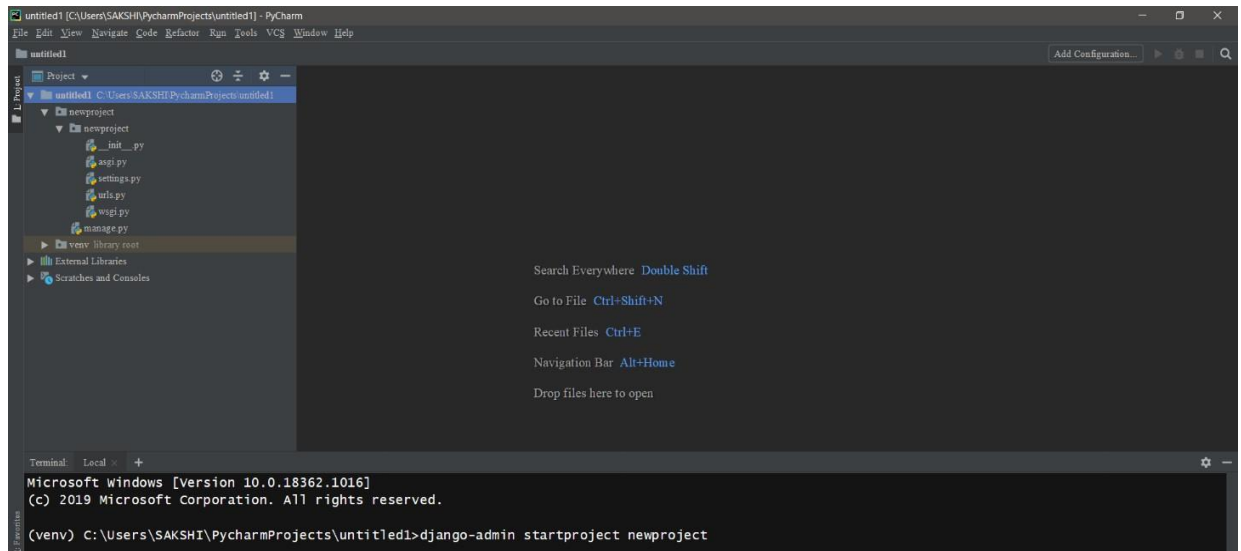


FIG 3.4: SETTING UP PROJECT

3.4 PostgreSQL

PostgreSQL also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley. In 1996, the project was renamed to PostgreSQL to reflect its support for SQL. After a review in 2007, the development team decided to keep the name PostgreSQL and the alias Postgres.

PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the default database for macOS Server, and is also available for Linux, FreeBSD, OpenBSD, and Windows.

PostgreSQL manages concurrency through multi version concurrency control (MVCC), which gives each transaction a "snapshot" of the database, allowing changes to be made without affecting other transactions. This largely eliminates the need for read locks, and ensures the

database maintains ACID principles. PostgreSQL offers three levels of transaction isolation: Read Committed, Repeatable Read and Serializable. Because PostgreSQL is immune to dirty reads, requesting a Read Uncommitted transaction isolation level provides read committed instead. PostgreSQL supports full serializability via the serializable snapshot isolation (SSI) method.

PostgreSQL includes built-in synchronous replication that ensures that, for each write transaction, the master waits until at least one replica node has written the data to its transaction log. Unlike other database systems, the durability of a transaction (whether it is asynchronous or synchronous) can be specified per-database, per-user, per-session or even per-transaction. This can be useful for workloads that do not require such guarantees, and may not be wanted for all data as it slows down performance due to the requirement of the confirmation of the transaction reaching the synchronous standby.

3.5 pgAdmin

pgAdmin is a free software project released under the PostgreSQL/Artistic license. The software is available in source and binary format from the PostgreSQL mirror network. Because compiling from source requires technical knowledge, it is recommended to install binary packages whenever possible.

pgAdmin 4 is a complete rewrite of pgAdmin, built using Python and Javascript/jQuery. A desktop runtime written in C++ with Qt allows it to run standalone for individual users, or the web application code may be deployed directly on a webserver for use by one or more users through their web browser. The software has the look and feel of a desktop application whatever the runtime environment is, and vastly improves on pgAdmin III with updated user interface elements, multi- user/web deployment options, dashboards and a more modern design.

3.5.1 Connection to Database

To connect to database, we write the code in the settings.py file of the project

- In the settings.py identify the database connection area.
- Add the name of database, username and password.
- Run the command “python manage.py make migrations”
- The database is created and can be viewed in the PgAdmin browser.

```
# cat myproject/settings.py
. . .
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': '<db_name>',
        'USER': '<db_username>',
        'PASSWORD': '<password>',
        'HOST': '<db_hostname_or_ip>',
        'PORT': '<db_port>',
    }
}
. . .
```

FIG 3.5: CONNECTION TO DATABASE

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

In architectural model it contains two databases: Patient Records database and Disease/Symptoms database. Four web services are used to implement the SOA. They are Pattern matching, recent trends, differential diagnosis and recent differential diagnosis. The patient Record database contains all the patient information from all the hospitals in the network.

Diseases/Symptoms database is a centralized database. It contains the list of existing known diseases and their corresponding symptoms along with their weights. These databases are replicated across various servers and these replicated servers are used to achieve the fault tolerance with concurrency protocols to achieve atomic transactions. First the doctor retrieves the symptoms from the patient record database. After retrieving the symptoms, the doctor identify whether any symptom related diseases contains in the Diseases/Symptoms database. Here the pattern matching service is activated. If any diseases match with Symptoms means list out all the possible matched symptoms and presents the result to the doctors. If the doctors not satisfied with results, compare to recent history and recent trend service must be activated. This service makes use of the Diseases/symptoms database and Patient Record database and the result obtained from pattern matching service to get results. After comparing the diseases to the recent history, cluster the shortlisted diseases.

The probability may be computed based on the distance vector. The highest priority cluster produces the accurate result. Finally, to avoid the vagueness in decisions, the doctor use differential diagnosis and recent diagnosis features use Diseases/symptoms database and Patient record database and result acquired from recent trend services to gain the results. Since the large medical data, using simple client server architecture would not produce the effective aforesaid services and would increase the response time of the system. Finally we conclude that SOA was well suited to apply this system because it improve the delivery of important information and sharing of data across the community of healthcare professionals more practical in cost, security and risk deployment. In various existing EHRs, SOA is more essential for data providers to this system, are already using this very successful and efficient architecture.

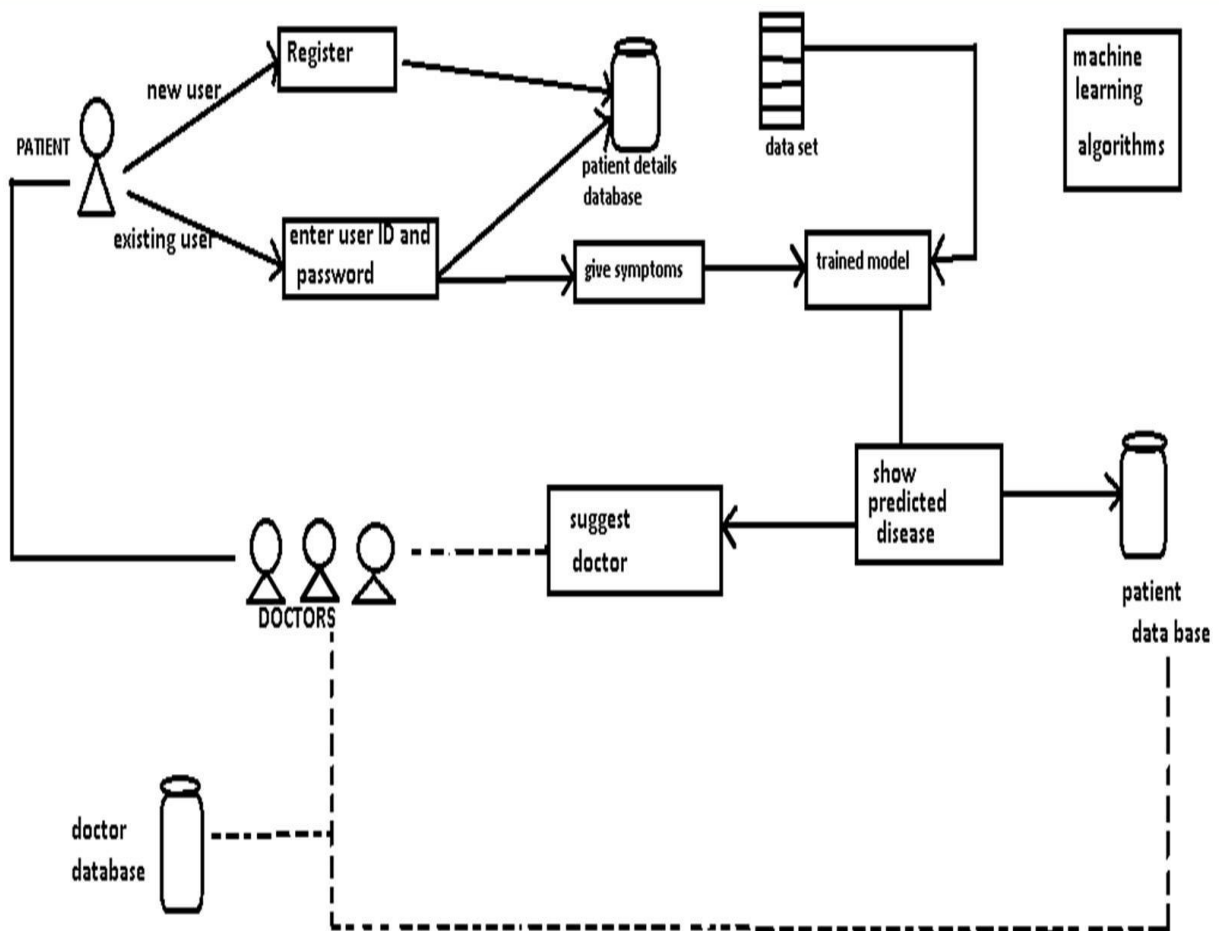


FIG 4.1: SYSTEM ARCHITECTURE

4.2 Use case diagram

The Use Case diagram of the project disease prediction using machine learning consist of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities are linked to each other where the user gets started with the system.

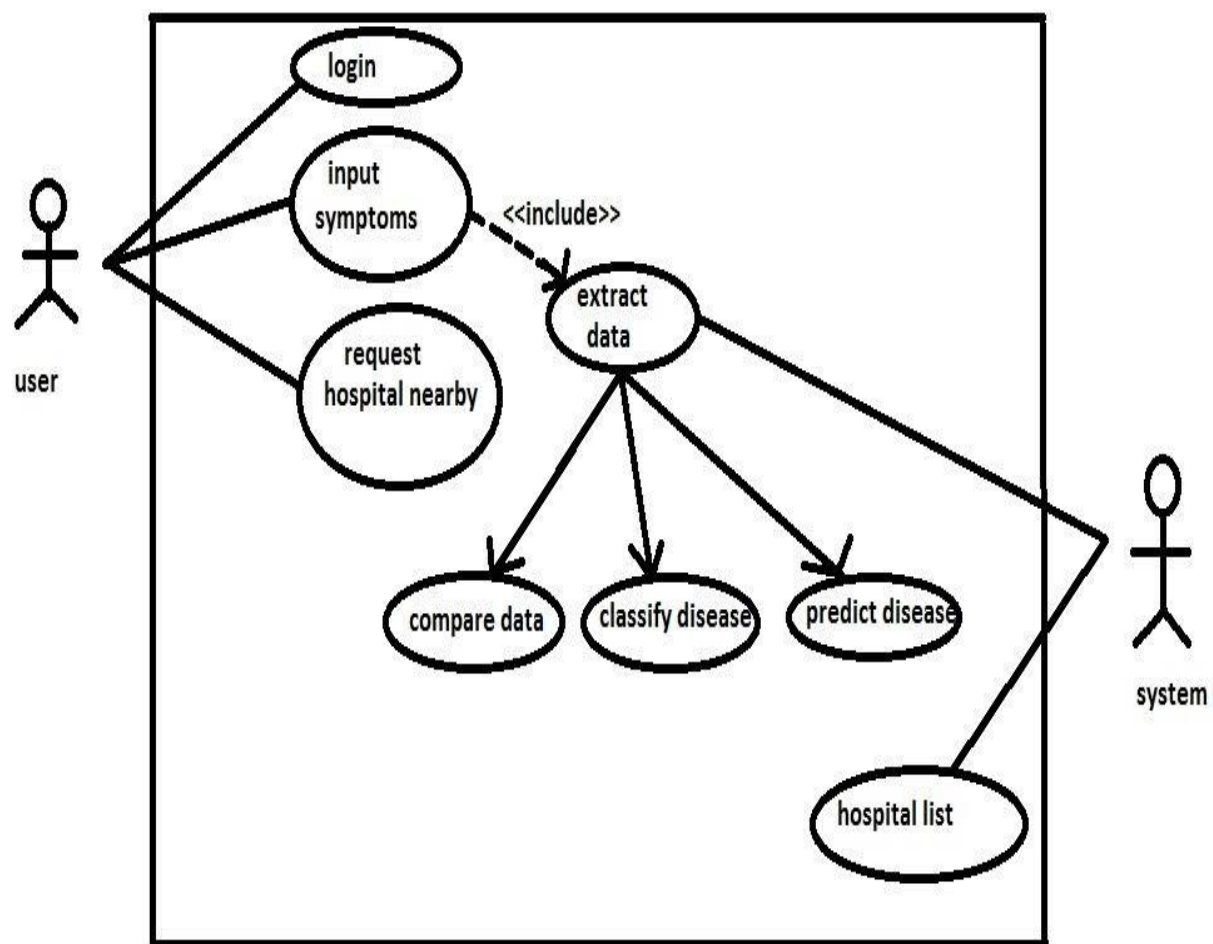


FIG 4.2: USE CASE DIAGRAM

CHAPTER 5

REQUIREMENTS SPECIFICATION

5.1 Hardware requirements

- 4Gb RAM
- I5 Processor ranging 1.90GHz -3.80GHz
- Hard Disk 500Gb recommended
- 512 KB cache memory or more

5.2 Software requirements

- PyCharm
- pgAdmin4
- Django
- PostgreSQL
- Operating System: Windows 10

CHAPTER 6

IMPLEMENTATION

6.1 Modules implementation

Implementation of a system must be done step by step in order to achieve the object of the developed system. Identifying the modules makes it easier to work on developing the system. In the proposed system, there are six modules that have been identified. These six modules are the six different features which put together will form a whole system. Six modules that have been identified are:

- Input module
- Doctor
- User
- Admin
- Prediction system
- Output model

Above mentioned modules are explained briefly in the below sections.

6.1.1 Input module

Input module is the user module, where user can enter the symptoms from the given set of predefined symptoms. User can select multiple symptoms which user is suffering from. These entered symptoms are analyzed and the disease is predicted based on the entered symptoms.

This input module is the important module for the system, where user enters symptoms and prediction is done by the input module. Whole prediction processes is done through the Input module. User can be a patient or any user or even doctor, where doctors can use this system to analyze the disease in a better way.

6.1.2 Doctors

In this module the doctors can create their account by sing in to the system with their specialization, which will be displayed to the user after prediction of disease. User can see doctor's details by clicking in to the doctor recommended in the prediction page.

This module also support user to contact with the patient, were patient can take appointment from the specialized doctor. If any issues regarding health then patient can text to doctors were doctor suggest some medications for the patient during emergency.

Doctor is authenticated in the sign-up form with the username and password, with successful authentication doctor can check for appointments or any patient text. Patient can clarify any issue through chat box available. This doctors' details are stored in the database for authentication, which increases security.

6.1.3 User

User module plays an important role in this system. User can access to the system by sign up to the system, with user details after successful account creation user is directed to prediction page were user should select the symptoms from the set of predefined symptoms, after selection the system predicts the disease associated and provides link which leads to more elaborate explanation about the particular disease. User is also be provided with the doctor specialized in that field.

User is authenticated in the sign-up form with the username and password, with successful authentication user is directed to prediction page. User can take appointments from doctors and even clarify any issues regarding health through chat box provided. User's details are stored in the database for authentication, which increases security.

User is also provided with doctors and hospitals for unpredicted diseases also were user can refer to any hospital or doctor according to their convenience

6.1.4 Admin

Users and doctors have unrestricted access to system, where as Admin can access all the modules in the system. Admin can update or remove modules based on the feedback provided by user and doctor. Admin modules mainly adds improved version of the system. Admin can control unrestricted access done by any user or doctor, and appropriate action can be taken accordingly. Admin can access databases like doctor, patient etc. where Admin can add or delete user or doctor.

6.1.5 Prediction system

Prediction system predicts the disease for the user based on the given symptoms. This is the main processing system where the user's data is compared with the trained data and various algorithms like decision tree, Naïve Bayes are applied in order to predict the diseases of the patient is based on the symptoms given by the user. This prediction system involves many various algorithms like:

Decision Tree: Decision tree uses tree structure and the tree begins with a single node representing the training samples. If the symptoms are all of the same disease, then node becomes the leaf and the class marks it. Otherwise, the algorithm chooses the discriminatory attribute as the current node of the decision tree. According to the value of the current decision node attribute, the training samples are divided into several subsets, each of which forms a branch, and there are several values that form several branches. For each subset or branch obtained in the previous step, the previous steps are repeated, recursively forming a decision tree on each of the partitioned samples.

Naïve Bayesian: Naïve Bayes classifiers are a collection of algorithms based on Bayes' theorem. The dataset is divided into two parts namely Feature matrix and Response vector. Feature matrix contains all vectors(rows) of the dataset in which each vector consists of value of dependent features. In Our data set, features are the symptoms such as 'Fever', 'Cold', 'Headache' etc. Sponse vector contains the values of Class variable (prediction or the output) for each row of feature matrix. in our dataset, the class variables are the diseases like 'malaria', 'dengue' etc. Bayes' theorem will also find the probability of an event occurring and hence in our model it not only helps in predicting the disease but also gives the probability of the occurrence of the disease.

6.1.6 Output module

This is the output which the user receives where it displays the predicted disease and the recommended doctor. This is the last module of the system which provides result page. This module provides output to user after selection of symptoms, and analyzation of given symptoms sending trained set in to various algorithm and output is predicted to the user.

