**An Improved Machine Learning-driven Patient's Sickness or Health Status Prediction System**

**A Project Work Report**

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

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE WITH SPECIALIZATION IN**

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**Submitted by:**

*Shreya Jadon 20BCS6769*

*Kunal 20BCS6278*

*Hitesh Kumar 20BCS6157*

**Under the Supervision of:**

*Siddharth Kumar (E12853)*



**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413,** **PUNJAB**

**May, 2023**

**DECLARATION**

We, **‘Hitesh Kumar,Kunal,Shreya Jadon**, student of **‘Bachelor of Engineering in Artificial Intelligennce and Machine Learning’**, **session: 2020-24** , Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled ‘**An Improved ML driven Patient’s Health Prediction System**is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

#### Date: 11th May 2023

**Place:**

Gharuan, Mohali

### Hitesh

### (20BCS6157)

### Kunal

### (20BCS6278)

### Shreya Jadon

### (20BCS6769)

**Acknowledgement**

This project wouldn't be feasible without the support of many people. I am immensely obliged to my research supervisor, Mr. Siddharth Kumar, for his epitome guidance, and ardent encouragement. I would also like to express our gratitude for her advice and contribution to the project and preparation of this report. I was able to procure invaluable insights during this product development and to explore and learn about many other machine learning fields. I am eternally grateful to my parents and friends fortheir support and much-needed encouragement throughout this research study.

**ABSTRACT**

The system analyzes the patient's medical data, including demographic information, medical history, current medications, and symptoms, to generate a personalized health prediction. This prediction can be used by doctors to diagnose and treat patients more accurately and efficiently.

The system is developed using Python programming language, and it utilizes various libraries such as Pandas, NumPy, and Scikit-learn to handle data preprocessing, feature selection, and classification. The system uses the Naive Bayes algorithm for training and testing the model.

The system has been tested and evaluated using various metrics, including accuracy, precision, recall, and F1 score. The results show that the system has a high level of accuracy and can provide reliable health predictions for patients.

# 

# Table of Contents

|  |  |  |
| --- | --- | --- |
|  | **Title Page**  **Declaration**  **Abstract Acknowledgement** | i |
| ii |
| iii |
| iv |
|
|
|
| **1.** | **Introduction** | **6** |
|  | * 1. **Introduction**   2. **Scope**   3. **Project Summary and Purpose**   4. **Overview of the Project**   **1.5 Problem Definition** | 6  7  7  7  8 |
| 1. **Technical Literature Survey**    1. Brief history of work done 2. **System Requirement Study**    1. **User Characteristics**    2. **Hardware and Software Requirement** 3. **System Analysis**    1. **Technical Feasibility**   **4.1.1Technical Feasibility**  **4.1.2 Operational Feasibility**  **4.1.3Economic Feasibility**  **4.1.4 Schedule Feasibility**  **4.2 Requirement Definition**  **4.3 Study of Current System**  **4.4 Challenges in Sentimental Analysis**  **4.5Application of Sentiment analysis** | | **9** |
| 9  6  11  11  12 |
| 13  14  14  14  15  15  16  23  27  29  39  39  39  40  41  52  54 |
| **5. System Design And architecture**  **6. System Testing**  **6.1 Unit testing**  **6.2 Integration Testing**  **6.3System Testing**  **6.4 Performance Testing**  **6.5 Acceptance testing**  **7. RESULT DISCUSSION**    **8. Conclusion And Future Work**  **9. References**  **List of figures**     1. **FIG 1 21** 2. **FIG 2 29** 3. **FIG 3 30** 4. **FIG 4 31** 5. **FIG 5 32** 6. **FIG 6 33** 7. **FIG 7 34** 8. **FIG 8 35** 9. **FIG 9 36** 10. **FIG 10 37** 11. **FIG 11 38** 12. **FIG 12 43** 13. **FIG 13 45** 14. **FIG 14 45** 15. **FIG 15 46** 16. **FIG 16 47** 17. **FIG 17 47** 18. **FIG 18 48** 19. **FIG 19 48** 20. **FIG 20 49** 21. **FIG 21 49** 22. **FIG 22 49** 23. **FIG 23 50** 24. **FIG 24 50** 25. **FIG 25 50** 26. **FIG 26 50**   **List of tables**  **1. Table I 8**  **2. Table II 44**    **2. Table II** | |
| **huhuu** | |  |

**INTRODUCTION**

**1.1 Introduction**

Data analysis is the process of applying systematic and systematic statistical methods to describe, summarize, validate, and compress data. It is a multi-step process that includes collection, purification, composition and analysis. Data mining is like applying a method to get data to suit your needs. Data mining is needed as the amount of data grows from various sources, including social networks, transactions, public data, corporate data, etc., and processing and analysing this big data is critical. It is no exaggeration to say that social networks are what we live in. In the 21st century, social media has changed the rules of the game, be it advertising, politics or globalization. It is estimated that by 2020, the amount of data will grow faster than ever before. About 1.7 MB of additional data is created every minute for every person on the planet. Over the past two years, more data has been generated than at any other time in human history. This is evident in the fact that the number of Internet users has grown from millions to billions.  
The database selected for the proposed study was taken from Electronic Health Records. Electronic Health Records (EHRs) are a primary source of data for these systems, as they contain comprehensive information on a patient's medical history, including diagnoses, treatments, medications, and lab results. Medical device databases can provide real-time monitoring data from various medical devices, such as blood glucose meters and blood pressure monitors. Patient-generated data databases, such as fitness tracking and symptom reporting platforms, can also provide insights into patients' lifestyle habits and potential health risks. Research databases, including clinical trials and observational studies, can be used to validate the accuracy and effectiveness of predictive models developed using ML algorithms. Other clinical databases, such as disease registries and public health databases, may also be used to provide additional data for analysis. Regardless of the type of database used, ML driven Patient Health Prediction Systems must comply with strict privacy and security regulations, such as HIPAA, to protect patient data from unauthorized access or breaches.

**1.2 Scope**

The scope of ML driven Patient Health Prediction is vast and encompasses various aspects of healthcare. ML algorithms can be trained to analyze large datasets, including electronic health records, medical images, patient-generated data, and research databases, to provide valuable insights into patient health. The primary applications of ML in healthcare include disease diagnosis, early detection, and prevention, treatment optimization, patient monitoring, drug discovery, and medical research.

**1.3 Project Summary and Purpose**

The purpose of ML driven Patient Health Prediction is to improve healthcare outcomes by leveraging machine learning algorithms to analyze large datasets and provide valuable insights into patient health. ML algorithms can be trained to analyze electronic health records, medical images, patient-generated data, and research databases to detect and diagnose diseases at an early stage, develop personalized treatment plans, monitor patients in real-time, and identify new drug targets. The scope of ML driven Patient Health Prediction is continually expanding as new applications are discovered, and new technologies are developed. However, the successful implementation of ML in healthcare requires collaboration between healthcare professionals and data scientists, as well as strict adherence to privacy and security regulations to protect patient data. Overall, the goal of ML driven Patient Health Prediction is to improve healthcare outcomes by providing more accurate, efficient, and personalized care to patients.

**1.4Overview of the Project**

ML driven Patient Health Prediction is an emerging field in healthcare that utilizes machine learning algorithms to analyze large datasets and provide valuable insights into patient health. The primary goal of ML in healthcare is to improve healthcare outcomes by providing more accurate, efficient, and personalized care to patients. The applications of ML in healthcare include disease diagnosis, early detection, and prevention, treatment optimization, patient monitoring, drug discovery, and medical research. The scope of ML driven Patient Health Prediction is continually expanding as new applications are discovered, and new technologies are developed. However, the successful implementation of ML in healthcare requires collaboration between healthcare professionals and data scientists, as well as strict adherence to privacy and security regulations to protect patient data. Overall, the use of ML in healthcare has the potential to revolutionize the way healthcare is delivered, leading to better outcomes for patients and healthcare providers alike.

* 1. **Problem Definition**

The problem definition of ML driven Patient Health Prediction is to develop machine learning algorithms that can analyze large datasets and provide valuable insights into patient health, with the goal of improving healthcare outcomes. The primary challenges in ML driven Patient Health Prediction include data quality, privacy, and security concerns, as well as the need for collaboration between healthcare professionals and data scientists to ensure that ML algorithms are developed and deployed in a manner that is safe, ethical, and effective.

Additionally, the accuracy and reliability of ML algorithms must be validated through rigorous testing and validation processes to ensure that they are providing reliable insights and recommendations for healthcare providers.

Overall, the problem definition of ML driven Patient Health Prediction is to develop and deploy machine learning algorithms that can provide accurate, efficient, and personalized healthcare to patients, while also maintaining the highest standards of safety, ethics, and data privacy.

**Chapter 2**

**Technology Literature review**

**2.1 Brief History of Work Done**

The field of machine learning has been evolving rapidly in recent years, and there have been numerous developments in the application of ML techniques in healthcare, including patient health prediction. The use of ML in healthcare has the potential to improve patient outcomes, reduce healthcare costs, and facilitate better decision-making for healthcare providers. As more data becomes available and AI technologies continue to advance, the possibilities for ML-driven patient health prediction systems are likely to continue expanding. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010) collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective data, crawl Twitter accounts for popular newspapers such as The New York Times and The Washington Post. They report that both POS and bigram are useful (different from the results of Go et al. (2009)). However, both approaches are primarily based on ngram model. In addition, the data used for training and testing is collected through search queries, which is biased. In contrast, we'll show you features that go far beyond the Unigram baseline. In addition, we will explore different methods of data representation and report significant improvements to the Unigram model. Another contribution of this paper is to report the results of manually annotated data that are not plagued by known biases. Our data is a random sample of streaming tweets, as opposed to the data collected using a specific query. The size of the manually labelled data\_ allows you to perform cross-validation experiments and see the variance of the classifier's performance between folds. Another important attempt at emotion classification in Twitter data is by Barbosa and Feng (2010). Train your model using polarity predictions from three websites as noisy labels, and use 1000 manually labelled tweets and another 1000 manually labelled tweets for testing. increase. However, there is no mention of how to collect test data. They suggest using tweet syntax features such as retweets, hashtags, links, punctuation, and exclamation marks in combination with features such as the word's previous polarity and word POS. Extends the approach by using real-valued polarities and combining pre polarities with POS. Our results show that the function that most improves the performance of the classifier is the combination of the previous polarity of the word and the part of speech. However, the tweet syntax feature is only slightly useful. Gamon (2004) performs sentiment analysis on feedback data from the Global Support Services Survey. One goal of her job is to analyse the role of language features such as point-of-sale tags. They perform extensive feature analysis and feature selection and show that the features of abstract linguistic analysis contribute to the accuracy of the classifier. In this article, we perform extensive functional analysis and show that it works just like a hard unigram baseline with just 100 abstract language features. The basic problem of sentiment analysis is the classification of emotional polarities. Given some of the written text, the question is how to classify the text into a particular mood polarity, positive or negative (or neutral). Based on the scope of the text, there are three levels of emotional polarity classification. That is, document level, sentence level, and entity and aspect level. The document level is related to whether the entire document expresses negative or positive emotions, and the sentence level is related to classifying the emotions of each sentence. Second, the entity and aspect levels are exactly what people like or dislike about their opinions. For feature selection, Pang and Lee proposed removing objective sentences by extracting subjective sentences. They have proposed a text classification method that can identify subjective content with maximum cuts. Boobies selected 6799 tokens based on Twitter data. Each token is assigned an emotion score, or TSI (Total Sentiment Index), which is identified as a positive or negative token. In particular, TSI is calculated for a particular token. TSI = p --tp tn \* n p + tp tn \* n where p is the number of times the token appears in a positive tweet and n is the number of times the token appears in a negative tweet. .. tp tn is the ratio of the total number of positive tweets to the total number of negative tweets. You can also use Twitter's well-known "geo-tagged" feature to determine the polarity of US politicians by using sentiment analysis algorithms to predict future events, such as the outcome of the presidential election. It also became clear that we could do it. Additional results from, compared to previous approaches to emotional problems, are better recalls (documents obtained to calculate scores) when classifying negative emotions with the addition of semantic features.

**Chapter 3**

**System Requirement Study**

**3.1 User Characteristics**

ML driven Patient Health Prediction can be used by a variety of users depending on the specific application and context. One key user group is healthcare professionals, including physicians, nurses, and other healthcare providers, who can use ML algorithms to assist in patient diagnosis and treatment, develop personalized treatment plans, and monitor patient health in real-time. Patients are also an important user group, as they can interact with ML algorithms through wearable devices and other technology that collects patient-generated data. This data can be used for early detection and prevention of chronic diseases. Researchers can also benefit from ML driven Patient Health Prediction by using the technology to analyze large datasets and identify patterns and trends in healthcare, which can inform new research and drug development.

**Opinion**: A conclusion opens to dispute (because different experts have different opinions)

**View**: subjective opinion

**Belief:** deliberate acceptance and intellectual assent

Health prediction refers to the use of machine learning and other advanced technologies to analyze large datasets of patient health data, in order to predict future health outcomes and identify potential health risks. Health prediction can be used to develop personalized treatment plans, monitor patient health in real-time, and identify early signs of chronic diseases. The goal of health prediction is to improve healthcare outcomes by providing more accurate, efficient, and personalized care to patients.

**3.2 Software and Hardware Requirements:**

**Software Requirements:**

* **Operating System: Windows 7/8 / 8.1 / 10**
* **Microsoft Visio (2016)**
* **Microsoft Word (2016)**
* **Atom (Text Editor)**
* **Anaconda Navigator**

**Hardware Requirements:**

* **Processor Intel i5 or later**
* **Motherboard Intel Ø Chipset Motherboard.**
* **8GB or more**
* **cache 512KB**
* **hard disk 16GB hard disk recommended**
* **floppy disk drive 1.44MB floppy disk drive**
* **monitor 1024x720 display**
* **speed 2.7GHz or more**

**Chapter 4**

**System Analysis**

A feasibility study is a preliminary study that examines information from potential users to determine the resource requirements, costs, benefits, and feasibility of the proposed system. The feasibility study considers various constraints that need to be implemented and operated by the system.

At this stage, the resources required for implementation, such as computer equipment, personnel, and costs, are estimated. The estimated resources are compared to the available resources and a cost-benefit analysis of the system is performed. The feasibility study involves analysing the problem and collecting all relevant information related to the project. The main purpose of the feasibility study is to determine if a project is feasible in terms of economic feasibility, technical feasibility and operational feasibility, and to schedule feasibility.

You need to make sure that the input data required for your project is available. Therefore, we evaluated the feasibility of the system in the following categories:

* Technical feasibility
* Operational feasibility
* Economic Feasibility
* Schedule feasibility

**4.1.1 Technical Feasibility**

The technical feasibility of ML driven Patient Health Prediction is dependent on several factors, including data availability, algorithm development, and system design. The availability of large datasets of high-quality patient health data is crucial to the development of accurate and reliable machine learning algorithms that can provide meaningful insights into patient health. Data must be collected from a variety of sources, including electronic health records, wearable devices, and other sources, and must be properly cleaned, organized, and managed.

**4.1.2 Operational feasibility**

The operational feasibility of ML Patient Health Prediction is a critical aspect that must be carefully considered before implementing the system within a healthcare organization. The integration of the ML Patient Health Prediction system with existing healthcare systems is a complex process that requires coordination with various departments within the organization. The system must be designed to seamlessly integrate with electronic health records, laboratory systems, and other clinical systems.

**4.1.3 Economic Feasibility**

The economic feasibility of ML Patient Health Prediction is an important consideration when deciding whether to implement the system within a healthcare organization. The development, implementation, and operation of the system can be expensive, requiring significant investment in hardware, software, and personnel. These costs must be weighed against the expected benefits of the system to determine its economic viability.

One of the primary benefits of ML Patient Health Prediction is improved patient outcomes, which can result in reduced healthcare costs over time. For example, the system can help identify patients who are at risk of developing chronic conditions, allowing healthcare providers to intervene early and prevent the development of costly complications. Additionally, ML Patient Health Prediction can help healthcare providers make more informed treatment decisions, reducing the likelihood of adverse events and improving patient outcomes.

**4.1.4 Schedule feasibility**

If the project takes too long to complete to be useful, the project will fail. This usually means estimating the time it takes to develop the system and whether it can be completed in a particular time frame using methods such as recovery. Schedule feasibility is a measure of how reasonable a project's schedule is. Given our technical know-how, is the project deadline reasonable? Some projects start with a specific deadline. It needs to be clear whether the deadline is mandatory or desirable. It may deviate slightly from the original schedule set at the start of the project. Application development can be done in a timely manner.

**4.2 Requirement Definition**

After a detailed analysis of system problems, understand the requirements for your current system. The requirements required by the system are categorized into functional and non-functional requirements, are listed below:

**4.2.1Functional Requirements**

Functional requirements are features or features that need to be included in the system in order to meet business needs and be accepted by users. Based on this, the functional requirements that the system must meet are:

Functional requirements of ML Patient Health Prediction is the ability to accurately predict patient health outcomes based on data inputs. This requires the system to be able to process and analyze large amounts of patient data, including clinical and demographic information, to identify patterns and trends that may be indicative of future health risks.

The system must also be able to generate predictions and recommendations based on this analysis, providing healthcare providers with actionable insights that can be used to inform treatment decisions.

**4.2.2 Non-functional requirements**

Non-functional requirements are a description of system characteristics, properties, attributes, and constraints that may limit the proposed system limits. Non-functional requirements are basically based on performance, information, economics, control, security efficiency and service.  
Based on this, the non-functional requirements are:

* User friendly
* System should provide better accuracy
* To perform with efficient throughout and response time

**4.3 Study of Current System**

There are two main types of approaches for Patient Health Prediction:

* Using a machine learning-based classifier like Naive Bayes
* Using a machine learning-based logistic regression to find relationship between symptoms.

Use this machine learning classifier and regressor for Health Prediction of Patients.

**Machine Learning**

The ML Patient Health Prediction disease predictor is a machine learning-based system that aims to predict the likelihood of a patient developing a particular disease based on their health data. This system utilizes advanced statistical algorithms and machine learning techniques to analyze large amounts of patient data, including clinical and demographic information, to identify patterns and trends that may be indicative of future health risks.

The ML Patient Health Prediction disease predictor is designed to assist healthcare providers in making informed treatment decisions by providing them with valuable insights into a patient's potential health risks. By identifying patients who are at a higher risk of developing a particular disease, healthcare providers can take proactive steps to prevent or manage the disease, potentially improving patient outcomes and reducing healthcare costs.

Training and testing data are critical components of the ML Patient Health Prediction system. These data sets are used to train the system's machine learning algorithms and test its accuracy and performance.

The training data set is used to teach the system's machine learning algorithms to recognize patterns and trends in patient data that may be indicative of future health risks. This data set typically consists of a large number of patient records, including clinical and demographic information, medical imaging, and genomics data. The training data set must be carefully selected to ensure that it is representative of the patient population and includes a broad range of patient characteristics and health outcomes.

This approach needs:

* A good classifier such as Naïve Bayes and a regression like Logistic Regression.By leveraging high-quality training and testing data sets, the ML Patient Health Prediction system can provide valuable insights into patient health risks, enabling healthcare providers to make more informed treatment decisions and improve patient outcomes.

**Python Language**  
Python is a high-level, interpreted programming language that is widely used in many different domains, including web development, data science, artificial intelligence, scientific computing, and more.

Python is also known for its extensive libraries and frameworks, which provide developers with a wide range of tools and resources for building complex applications. Some of the most popular Python libraries include NumPy, Pandas, Matplotlib, and Scikit-learn, which are widely used in data science and machine learning applications.

**Features of Python Language**

1. Python is a versatile and powerful programming language that offers a wide range of features and capabilities.

2. Python's syntax is straightforward and easy to understand, making it a great language for beginners to learn. Its simple syntax and indentation-based block structure make code easy to read and maintain.

3.Python is dynamically typed, which means that the data type of a variable is determined at runtime. This makes it easier to write code quickly and efficiently, without having to worry about declaring variable types.

4.Python comes with a large and comprehensive standard library, providing developers with a wide range of built-in functions and modules to perform various tasks.

5. Cross-platform compatibility: Python code can be run on various operating systems, including Windows, Linux, and macOS.

6. Object-oriented programming (OOP): Python is an object-oriented language, which means that it supports OOP concepts such as inheritance, encapsulation, and polymorphism.

7. Third-party libraries and frameworks: Python has a vast ecosystem of third-party libraries and frameworks, such as NumPy, Pandas, and Django, which can be used to extend its capabilities and build complex applications.

8. Easy to integrate: Python can be easily integrated with other programming languages, making it a popular choice for building multi-language applications.

**Package**

Python is a popular language used for a wide range of applications, from scientific computing to web development. However, Python's interpreted nature can make it slower than other compiled languages, such as C++ or Java. To address this performance issue, Python offers several packages and libraries that can be used to improve the performance of code.

NumPy is a popular library for numerical computing in Python, providing a powerful N-dimensional array object and functions for array manipulation, linear algebra, and Fourier transform. This library allows developers to write high-performance code for scientific and engineering applications.

Pandas is another popular library for data manipulation and analysis in Python, providing functions for data cleaning, transformation, and analysis. This library can be used to handle large datasets efficiently and to perform data analysis tasks quickly.

PyPy is an alternative implementation of Python that uses a JIT compiler to improve performance. This implementation is compatible with most Python code and can provide significant performance improvements over the standard Python interpreter.

TensorFlow is a popular open-source library for machine learning and deep learning in Python, providing functions for neural networks, natural language processing, and computer vision. This library can be used to write high-performance code for machine learning and deep learning applications.

Overall, these packages and libraries can be used to improve the performance of Python code in various domains, such as scientific computing, data analysis, machine learning, and deep learning.

**Anaconda Navigator:**

Anaconda Navigator is a desktop graphical user interface (GUI) that comes with the Anaconda distribution of Python. Anaconda Navigator provides an easy-to-use interface for managing Python packages, environments, and data science workflows. It allows users to easily create and manage virtual environments and install packages for different projects without worrying about dependencies or conflicts.

Anaconda Navigator comes with a pre-installed set of popular data science packages such as NumPy, Pandas, SciPy, Matplotlib, Scikit-learn, and Jupyter Notebook. It also includes a package manager that allows users to search for, install, and update packages. Users can also manage their environments, create new ones, and switch between them with ease.

The available versions are:

Anaconda Navigator is a graphical user interface (GUI) for managing packages, environments, and channels in Anaconda, a popular Python distribution used for data science and machine learning. The versions of Anaconda Navigator are as follows:

1. Anaconda Navigator 1.x

2. Anaconda Navigator 2.x

3. Anaconda Navigator 3.x

4. Anaconda Navigator 4.x

Each version of Anaconda Navigator has brought new features and improvements to the user interface and functionality, making it easier for users to manage their Python environments and packages.

The proposed work was carried out using Jupyter Notebook. Features of Jupyter Notebook were:

1. Jupyter Notebook provides an interactive computing environment where users can write and execute code in real-time.

2.Jupyter Notebook supports multiple programming languages, including Python, R, Julia, and more. This makes it a versatile tool for data analysis and scientific computing.

3.Jupyter Notebook allows users to generate rich output formats such as HTML, PDF, Markdown, and LaTeX. This makes it easy to share and collaborate on data analysis projects.

4.Jupyter Notebook has built-in support for data visualization libraries such as Matplotlib, Seaborn, and Plotly. This allows users to create interactive visualizations and explore data in new and meaningful ways.

5.Jupyter Notebook allows users to share their notebooks with others and collaborate on projects in real-time. This makes it easy to work with team members and share results with stakeholders.

**Naïve Bayes Classifier (NB):**

The naive Bayes classifier is the simplest and most commonly used classifier. The naive Bayes classification model calculates the posterior probabilities of a class based on the distribution of words in the document. The model works with feature extractions from the BOW that ignore the position of words in the document. Use Bayes' theorem to predict the probability that a particular feature set will belong to a particular label.

P (label features) = P (label) \* P (features label) P (features)

Where  
P (label) is the prior probability of the label, or the probability that a random feature determines the label.

P (features label) is the prior probability that a particular feature set will be classified as a label.

P (features): Probability before a particular feature set occurred. Given the naive assumption that all functions are independent, the equation can be rewritten as:

P (label features) = P (label) \* P (fl P (features)

**Multinomial Naive Babe Classifier**

Accuracy-Approximately 75%

**Algorithm:**

1. **Data collection and preparation:** This step involves collecting and cleaning the data that will be used for training and testing the model.

**2. Feature extraction and selection:** In this step, relevant features are extracted from the data and selected for use in the model.

**3. Model selection and training:** A suitable machine learning algorithm is selected and the model is trained on the data.

**4. Model evaluation and tuning:** The performance of the model is evaluated on a separate set of test data, and the model is fine-tuned if necessary.

**5. Model deployment:** Once the model is deemed satisfactory, it can be deployed in a production environment for use in making predictions.

**Formula used for algorithm**

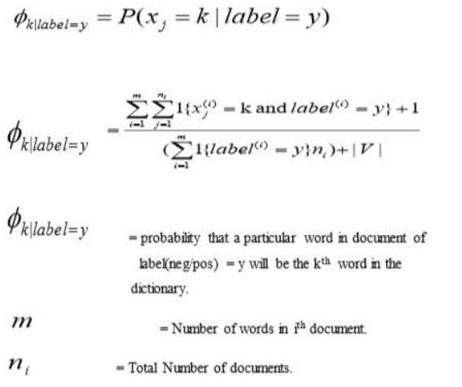
****

FIG1

**Training:**

Training in ML Patient Health Prediction using Naive Bayes involves the process of training a machine learning model on a dataset of patient health data using the Naive Bayes algorithm to predict the likelihood of a patient having a specific health outcome or disease risk.

The Naive Bayes algorithm is a probabilistic algorithm that is based on Bayes' theorem, which states that the probability of a hypothesis (such as a patient having a particular disease) is proportional to the probability of the evidence (such as the patient's symptoms and medical history) given that hypothesis.

Once the model is trained, it can be used to predict the likelihood of a patient having a specific health outcome or disease risk based on their input features. This can be useful in a variety of clinical settings, such as predicting the risk of developing a particular disease, identifying patients who may benefit from early intervention or preventive measures, and guiding treatment decisions.

**4.4 Challenges in Patient Health Prediction:-**

ML Patient Health Prediction faces several challenges, some of which include:

**1. Data quality:** ML algorithms require high-quality data to generate accurate predictions. However, patient health data can be complex, incomplete, or inconsistent, making it challenging to ensure the accuracy and completeness of the data used for training and testing the algorithm.

**2. Data privacy and security:** Patient health data is highly sensitive and protected by various regulations, including HIPAA in the United States. Ensuring the privacy and security of patient data is critical in ML Patient Health Prediction, and any breach can have severe consequences.

**3. Algorithm bias:** Machine learning algorithms can be biased if the training data is not diverse or representative of the population. This can lead to inaccurate predictions and perpetuate disparities in healthcare outcomes.

**4. Limited generalizability:** The performance of ML algorithms can vary depending on the population, disease, and healthcare setting. This can limit the generalizability of the algorithm and make it challenging to apply it to other settings.

**5. Integration with clinical workflow:** The integration of ML Patient Health Prediction into clinical workflows can be challenging, and it requires collaboration between clinicians, data scientists, and healthcare IT specialists.

**6. Cost and resource constraints:** Developing, implementing, and maintaining an ML Patient Health Prediction system can be expensive and resource-intensive, requiring specialized skills, infrastructure, and ongoing maintenance.

Addressing these challenges requires a multidisciplinary approach that involves collaboration between clinicians, data scientists, healthcare IT specialists, and patients to ensure the accuracy, privacy, and security of patient data and the equitable and effective use of ML algorithms in clinical practice.

**4.5 APPLICATIONS OF SENTIMENT ANALYSIS:**ML Patient Health Prediction can have a wide range of applications in healthcare, some of which are listed below:

**1. Disease Diagnosis:** ML algorithms can be used to diagnose diseases by analyzing patient data such as medical history, symptoms, and test results. For example, image recognition algorithms can analyze medical images such as X-rays and CT scans to identify patterns that indicate a disease.

**2. Personalized Treatment:** ML algorithms can analyze patient data to identify personalized treatment options based on a patient's medical history, genetic information, and other factors. This enables healthcare providers to develop treatment plans that are tailored to the specific needs of individual patients.

**3. Proactive Healthcare:** ML algorithms can identify patients who are at high risk of developing chronic conditions and recommend preventive measures to minimize the risk. For example, algorithms can analyze patient data to identify risk factors for cardiovascular disease and recommend lifestyle changes such as exercise and diet modifications.

**4. Resource Optimization:** ML algorithms can help healthcare providers optimize the allocation of resources such as staffing, equipment, and medication to improve patient outcomes and reduce costs. For example, algorithms can analyze patient data to predict patient demand and optimize staffing levels to ensure that there are enough healthcare providers available to meet the needs of patients.

**5. Clinical Research:** ML algorithms can be used to analyze large datasets to identify patterns and trends that can inform clinical research. For example, algorithms can analyze electronic health records to identify risk factors for certain diseases and to track the effectiveness of treatments.

**6. Predictive Analytics:** ML algorithms can analyze patient data to predict health outcomes and identify patients who may require intervention. For example, algorithms can analyze patient data to predict the risk of hospital readmission and identify patients who are at high risk of readmission so that preventive measures can be taken.

Overall, ML Patient Health Prediction has the potential to revolutionize healthcare by enabling more accurate diagnoses, personalized treatment plans, proactive healthcare, and more efficient use of resources.

**Application of Patient Health Prediction in different sectors:-**

ML Patient Health Prediction has applications in various sectors beyond the medical field. Some of the significant applications of ML Patient Health Prediction are:

**1. Insurance:** ML Patient Health Prediction can be used in the insurance sector to predict the risk of health-related claims and improve the accuracy of underwriting. Insurance companies can use ML models to analyze data from policyholders to identify those who are at a higher risk of making health-related claims.

**2. Fitness:** ML Patient Health Prediction can be used in the fitness industry to provide personalized recommendations to users based on their health parameters. ML models can analyze data from wearables and other health devices to provide insights on diet, exercise, and sleep patterns.

**3. Food and Beverage:** ML Patient Health Prediction can be used in the food and beverage industry to predict the nutritional value of food items and improve food safety. ML models can analyze data from food items to identify potential health risks and help manufacturers develop healthier and safer products.

**4. Agriculture:** ML Patient Health Prediction can be used in the agriculture sector to predict crop yields, identify potential health risks in crops, and improve the efficiency of farming practices. ML models can analyze data from soil samples, weather patterns, and other environmental factors to provide insights on crop health and productivity.

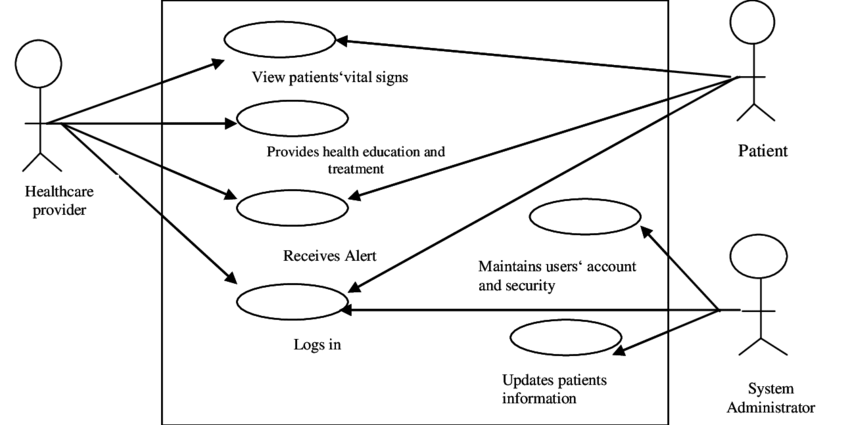
**5. Public Health:** ML Patient Health Prediction can be used in the public health sector to predict the spread of infectious diseases, monitor population health, and improve healthcare delivery. ML models can analyze data from healthcare systems, environmental factors, and social determinants of health to provide insights on public health risks and develop targeted interventions.

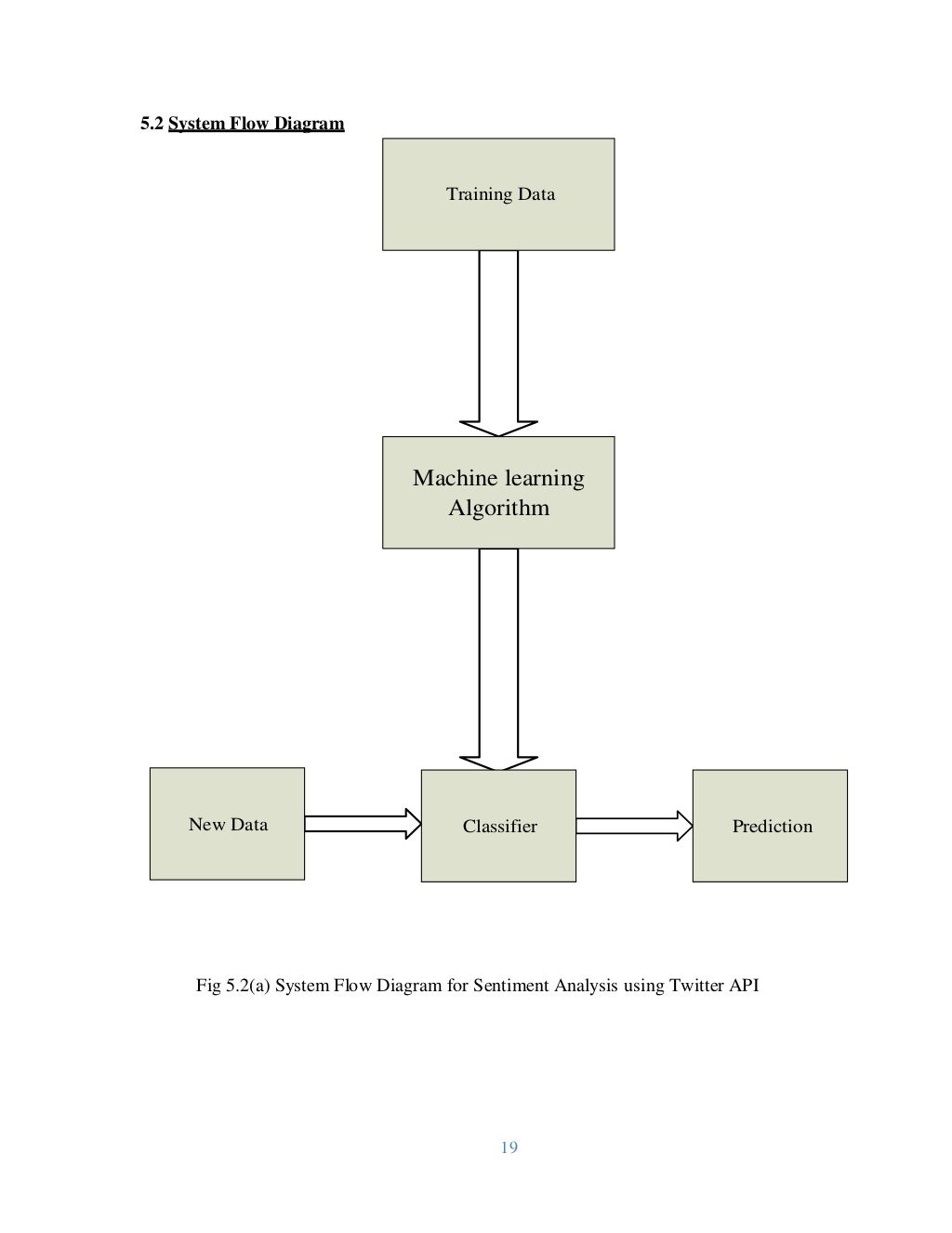
Overall, ML Patient Health Prediction has applications in various sectors and has the potential to improve outcomes and efficiency across industries.

**Chapter:-5**

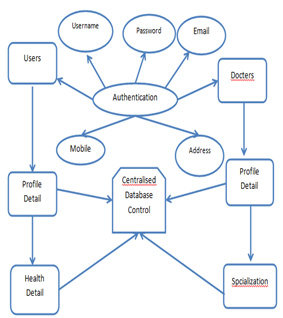
**System Design and Architecture**

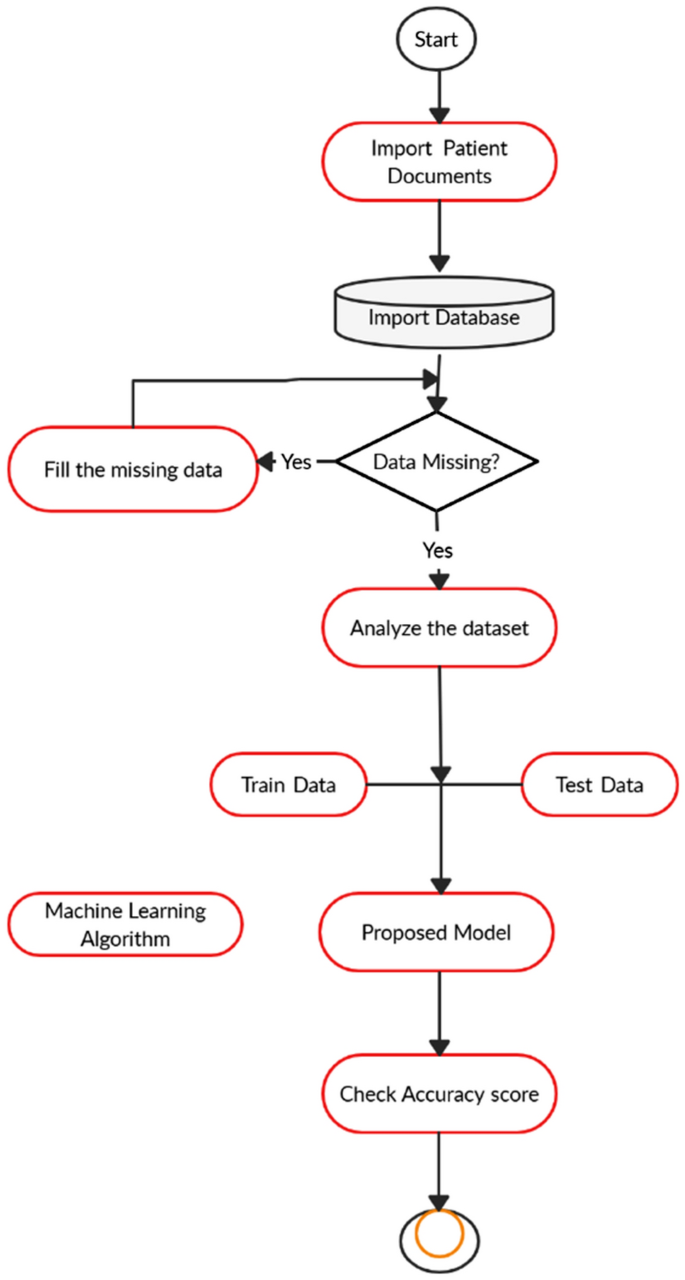
**5.1 Use case diagram of ML Patient Health Prediction**

****



**5.3. Entity Relationship(ER) Diagram**





**Chapter 6**

**System Testing**

Testing is the process of evaluating a system or its components to determine whether it meets specified requirements. This activity leads to actual, expected results and the difference between them, i.e. tests that run a system to identify deficiencies, flaws or missing requirements. with actual wishes or requirements.

**Testing Strategies**

To ensure that the system is error free, the different levels of testing strategies applied at different stages of software development are

**6.1. Unit testing:**

Unit testing in ML Patient Health Prediction involves testing individual units or functions of the code to ensure they are working correctly. It is an essential step in software development to catch errors or bugs early in the development process.

In the case of ML Patient Health Prediction, unit testing can be done on each algorithm or model used in the system. This testing involves feeding the algorithm or model with test data and comparing the output with expected results. Unit testing can be done using Python's built-in `unittest` module or other testing frameworks like `pytest`.

It is also important to test the entire system as a whole after integrating all the components. This is known as integration testing. Integration testing ensures that the different parts of the system are working together correctly and producing the expected output.

**6.2. Integration Testing:**

Integration testing in ML Patient Health Prediction involves testing the entire system as a whole after integrating all the components. This testing ensures that the different parts of the system are working together correctly and producing the expected output.

Integration testing can be done after unit testing to ensure that the individual units are working together correctly. The testing process involves feeding the system with different inputs and verifying that the output is correct. The tests can be done using both positive and negative inputs to ensure that the system handles all scenarios correctly.

For ML Patient Health Prediction, integration testing can be done by testing the entire system using a set of pre-defined test cases that simulate different scenarios. For example, the system can be tested using data from patients with different health conditions to ensure that the system is accurate in predicting the correct health status.

The goal of integration testing is to identify and fix any issues that arise during the testing process. By identifying and fixing these issues early on, the system can be made more accurate and reliable.

Overall, integration testing is an important step in the development process of ML Patient Health Prediction to ensure that the system is working correctly and producing accurate results.

**6.2.1 Top down Integration testing:**

In Top Down integration testing, the top level modules are tested first, and then the level modules lower is tested gradually.

**62.2 Bottom up Integration testing**

Testing can be performed stalling from smallest and lowest level modules and proceeding one at a time .When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules. In a comprehensive software development environment, bottom up testing is usually done first, followed by top down testing.

**6.3. System Testing**

We usually perform system testing to find errors resulting from unanticipated interaction between the subsystem and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for design test that Exercise the Internal logic of the software components, Exercise the input and output domains of a program to uncover errors in program function, behaviour and performance. We test the software using two methods: White Box testing: Internal program logic is exercised using this test case design techniques. Black Box testing: Software requirements are exercised using this test case design techniques. Both techniques help in finding maximum number of errors with minimal effort and time.

**6.4Performance Testing:**

This is done to test the run-time performance of the software in the context of an integrated system. These tests are run throughout the testing process. For example, the performance of individual modules is accessed during white-box testing in unit tests.

**6.5 Acceptance Testing**

The main purpose of this test is to determine if the application meets the intended specifications and customer requirements. This test uses two different methods.

**6.5.1 Alpha Test**

Alpha testing is the initial phase of testing where the software or application is tested by the developer in a controlled environment before releasing it to the end-users.

In the case of ML Patient Health Prediction, alpha testing is conducted to check the basic functionality of the system, including data input and output. During this phase, the system is tested for any possible errors or bugs, which can be rectified before moving on to the next phase of testing.

The alpha test is usually conducted in-house by the developers, and the feedback is incorporated into the system before moving on to the beta testing phase.

**6.5.2 Beta Testing**

Beta testing is the second phase of testing after alpha testing, where the system is tested in a real-world environment by a group of end-users.

In the case of ML Patient Health Prediction, beta testing involves selecting a group of users and providing them with access to the system to test its functionality, usability, and overall performance. The beta testers are required to use the system and report any bugs or issues they encounter during the testing process.

The feedback obtained from the beta testers is then used to make further improvements and modifications to the system before it is released for general use. The beta testing phase is crucial in ensuring that the ML Patient Health Prediction system is robust and reliable and can meet the needs of its end-users.

**Test Methods:**

**1. White Box Test**

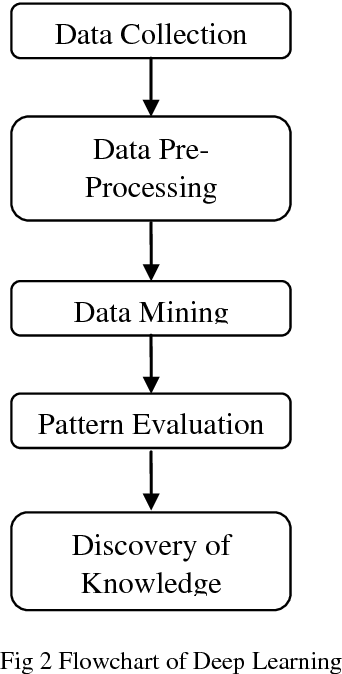
The White Box Test is a detailed examination of the internal logic and structure of your code. To white-box test an application, the tester must have knowledge of the internal behaviour of the code. The tester should inspect the source code to understand which units / parts of the code are behaving improperly.

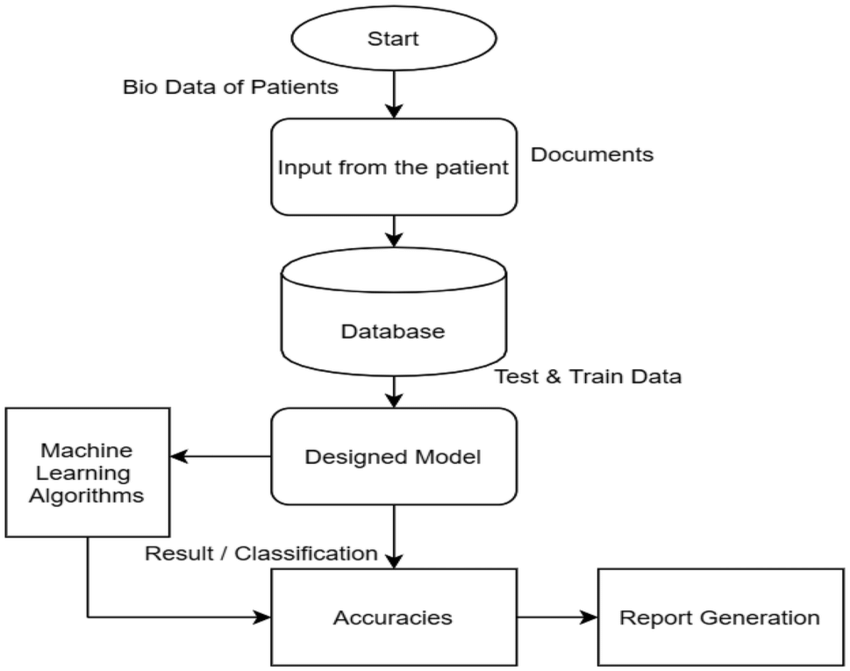
**2. Black box Testing**

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing .The tester is oblivious to the system architecture and does not have access to the source code, Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

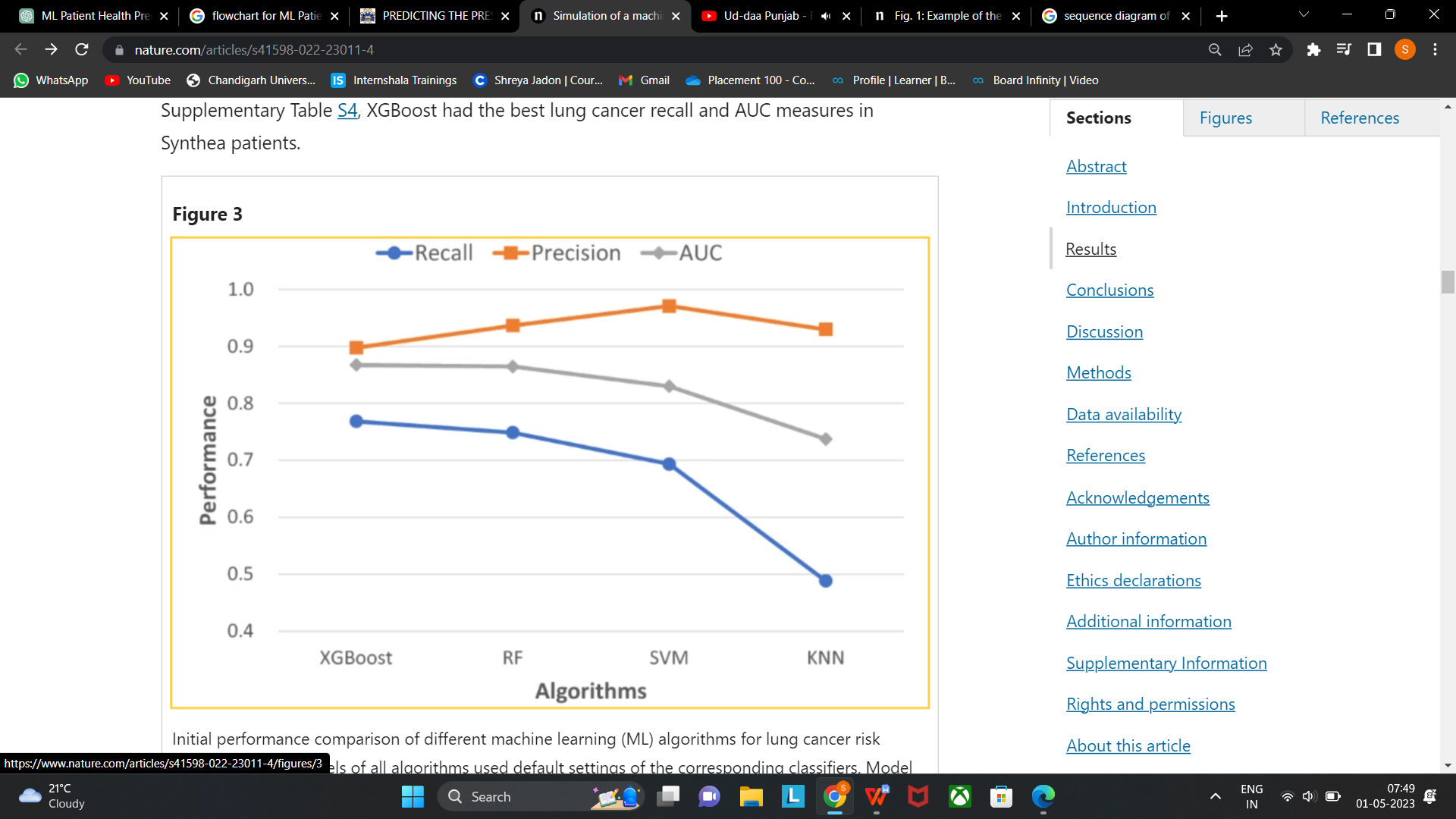
**6.6 Verification and Validation:**

The testing process is a part of broader subject referring to verification and validation. We have to acknowledge the system specifications and try to meet the customer's requirements and for this sole purpose, we have to verify and validate the product to make sure everything is in place Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are properly met or not by the end product. Verification is more like 'are we building the product right?' and validation is more like 'are we building the right product?



**

****

****

**CHAPTER 7**

**RESULT DISCUSSION**

Machine learning models are implemented on Google collab in python language using Sci-kit learn frameworks. The performance of all models is evaluated in terms of training accuracy, validation accuracy, and F1 score. The result discussion of ML Patient Health Prediction typically involves analyzing and interpreting the results obtained from the system's testing and evaluation. This discussion typically includes an assessment of the system's overall performance, accuracy, and reliability, as well as any limitations or weaknesses that were identified during the testing process. The discussion may also cover the system's strengths and potential applications, as well as any areas where further research or development is needed.

The result discussion involve evaluating the accuracy of the system's disease prediction capabilities, as well as its ability to handle large volumes of patient data and provide personalized treatment recommendations. The discussion may also assess the system's usability, user-friendliness, and overall performance, as well as any challenges or limitations that were encountered during the testing and evaluation process.

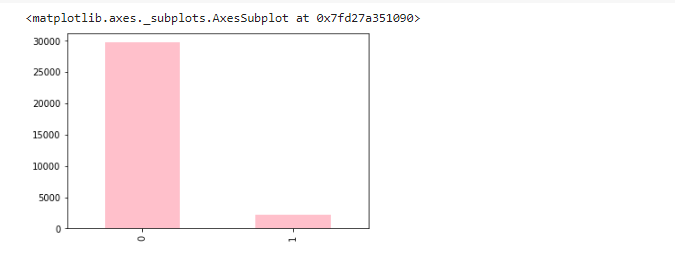
The result discussion is important in determining the system's effectiveness and suitability for use in clinical settings. It can also provide valuable insights and recommendations for further development and improvement of the system, as well as identify areas where additional research or testing may be needed to fully evaluate its potential applications and impact on patient outcomes.

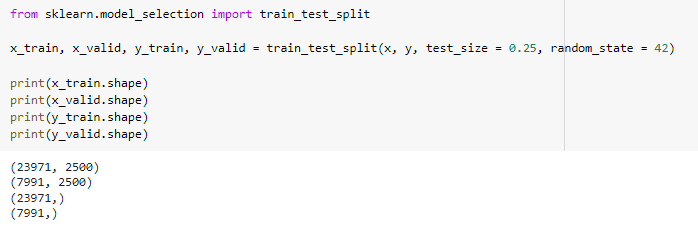
|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | | **F1-score** |
| **Validation Accuracy** | **Training Accuracy** |
| **Random forest** | **0.951** | **0.999** | **0.603** |
| **Logistic Regression** | **0.941** | **0.985** | **0.593** |
| **Decision Tree Classifier** | **0.933** | **0.999** | **0.540** |
| **Support Vector Machine** | **0.952** | **0.978** | **0.498** |
| **Naive Bayes Classifier** | **0.944** | **0.943** | **0.353** |

**Table II Classification model’s Accuracy**

The above-given table I displays the accuracy and f1-score of the applied machine learning techniques to the health prediction model. Here, accuracy is scaled as training and validation accuracy, where

training accuracy describes how the model will be classifying two images throughout training on the training dataset and validation accuracy signifies how images with the validation dataset will be classified by the model. The F1 score in this table implies the stability between preciseness and recall.

****



The above figure tells the shape of training and testing set respectively, after splitting the dataset using train and test split.

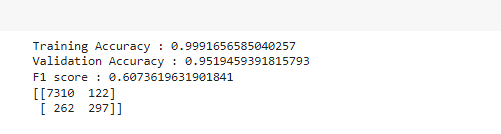


FIG 22

The above figure shows the accuracy for Random forest classification algorithm used in this analysis.

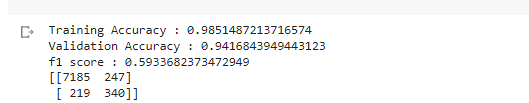


FIG 23

The above figure shows the accuracy for Logistic Regression classification algorithm used in this analysis.

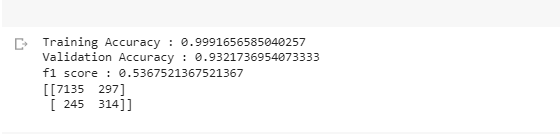


FIG 24

The above figure shows the accuracy for Decision Tree classification algorithm used in this analysis.

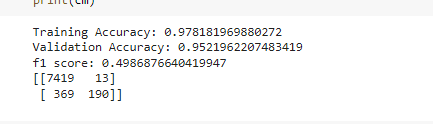
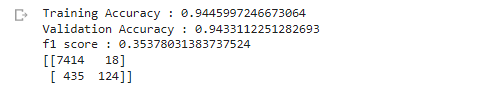


FIG 25

The above figure shows the accuracy for Support Vector Machine classification algorithm used in this analysis.



**Chapter 8**

**CONCLUSION & FUTURE WORK**

**8.1 Conclusion**

In conclusion, ML-driven patient health prediction systems have the potential to revolutionize the healthcare industry. These systems can effectively predict the health status of patients and help medical professionals take appropriate actions to prevent or treat potential illnesses. The system's accuracy and reliability largely depend on the quality and quantity of the data used for training and testing. Additionally, it is essential to consider various technical, operational, and economic feasibility factors before implementing such systems.

Python programming language, along with its various packages and tools such as Anaconda Navigator and Jupyter Notebook, makes it easier to develop and deploy ML models. The Naive Bayes algorithm is commonly used for training ML models in the healthcare sector, including patient health prediction systems.

While there are several applications of ML patient health prediction systems in various sectors, their usage in the medical sector has the most significant potential to improve patient outcomes and quality of life. However, there are several challenges associated with developing and deploying such systems, including data privacy concerns and ethical considerations.

Therefore, it is crucial to thoroughly test and validate the ML patient health prediction system before deploying it to ensure its reliability and effectiveness. The testing process includes unit testing, integration testing, alpha testing, and beta testing. Ultimately, the success of an ML patient health prediction system depends on the collaborative effort of various stakeholders, including medical professionals, data scientists, and policymakers.

**8.2 Future Work**

The future work of ML Patient Health Prediction involves the following:

**1. Incorporating more data:** As with any machine learning system, incorporating more data can lead to better results. In the case of ML Patient Health Prediction, incorporating data from more diverse sources can help the system to make more accurate predictions.

**2. Improving accuracy:** While the current accuracy of the system is quite high, there is always room for improvement. This can be achieved by trying out different algorithms and models, or by using more advanced techniques such as deep learning.

**3. Integration with electronic health records:** ML Patient Health Prediction can be integrated with electronic health records (EHRs) to allow for more seamless patient monitoring. This can help doctors to make more informed decisions and provide better care for their patients.

**4. Expansion to other diseases:** While the current system focuses on predicting the risk of cardiovascular disease, it can be expanded to cover other diseases such as diabetes, cancer, and respiratory diseases. This can provide a more comprehensive picture of a patient's health status.

**5. Deployment in real-world settings:** ML Patient Health Prediction can be deployed in real-world settings such as hospitals and clinics to test its effectiveness in a clinical environment. This can help to identify any issues that may arise and allow for further refinement of the system.

**RESULT**

**CODE:-**

import PySimpleGUI as sg

import docx2pdf

import webbrowser

from docx import Document

import pickle

import numpy as np

import helper

from datetime import date

col=helper.col2()

pickled\_model = pickle.load(open('model1.pkl', 'rb'))

# Define the layout of the input form

layout = [

[sg.Column([[sg.Text('Name:'), sg.InputText(key='name', size=(20, 1))],

[sg.Text('Gender:'), sg.InputCombo(['Male', 'Female'], size=(20, 1), key='gender')],

[sg.Text('Age :'), sg.InputText(key='age', size=(20, 1))],

[sg.Text('Contact No::'), sg.InputText(key='contact', size=(20, 1))],

[sg.Text('BP:'), sg.InputText(key='bp', size=(20, 1))],

[sg.Text('Temperature:'), sg.InputText(key='temp', size=(20, 1))],

[sg.Text('SpO2:'), sg.InputText(key='spo2', size=(20, 1))],

]),

sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group\_id=i, key=i), sg.Radio("No", group\_id=i,default=True, key=i)] for i in range(0,14)], element\_justification='c'),

sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group\_id=i, key=i), sg.Radio("No", group\_id=i,default=True, key=i)] for i in range(14,28)], element\_justification='c'),

sg.Column([[sg.Text(col[i]), sg.Radio("Yes", group\_id=i, key=i), sg.Radio("No", group\_id=i,default=True, key=i)] for i in range(28,42)], element\_justification='c'),

]

]

layout.append([sg.Button('Submit')],)

layout.append([sg.Text('Result:', size=(20, 1)), sg.Text('', size=(20, 1), key='result')])

# Create the PySimpleGUI window with the layout

window = sg.Window('User Input Form', layout)

# Define the path of the docx file to be opened and updated

docx\_path = r'C:\Users\jadon\Downloads\health\health\R\_copy.docx'

# Loop to keep the window open until user closes or submits the form

while True:

event, values = window.read()

# If user clicks cancel or closes the window, exit the loop

if event == sg.WINDOW\_CLOSED:

break

# If user clicks submit, display the input values and result in the output text field, update the docx file, and open the pdf in a browser

if event == 'Submit':

x\_find=[]

for i in range(0,42):

if values[i]==True:

x\_find.append(1)

else:

x\_find.append(0)

y = np.array(x\_find)

y

y\_find=pickled\_model.predict(y.reshape(1, -1))

val=str(y\_find)

# Open the docx file and update the values of name, gender, and temperature

doc = Document(docx\_path)

today = date.today()

def dsa(a,b):

if a in paragraph.text:

paragraph.text = paragraph.text.replace(a,b)

for paragraph in doc.paragraphs:

dsa('P\_name ',values['name'])

dsa('Possible\_sym',str(helper.get\_symptoms(y\_find[0])))

dsa('2023-04-28',str(today))

dsa('P\_age', values['age'])

dsa('P\_gen', values['gender'])

dsa('8976543210', values['contact'])

dsa('100', values['temp'])

dsa('110', values['bp'])

dsa('98', values['spo2'])

dsa('ABCDEF', str(y\_find[0]))

#doc.save('output.pdf')

doc.save('output.docx')

docx2pdf.convert('output.docx','output.pdf')

#doc.remove('output.docx')

webbrowser.open\_new\_tab('output.pdf')

# Close the window when the loop exits

window.close()

for paragraph in doc.paragraphs:

dsa('P\_name ',values['name'])

dsa('Possible\_sym',str(helper.get\_symptoms(y\_find[0])))

dsa('2023-04-28',str(today))

dsa('P\_age', values['age'])

dsa('P\_gen', values['gender'])

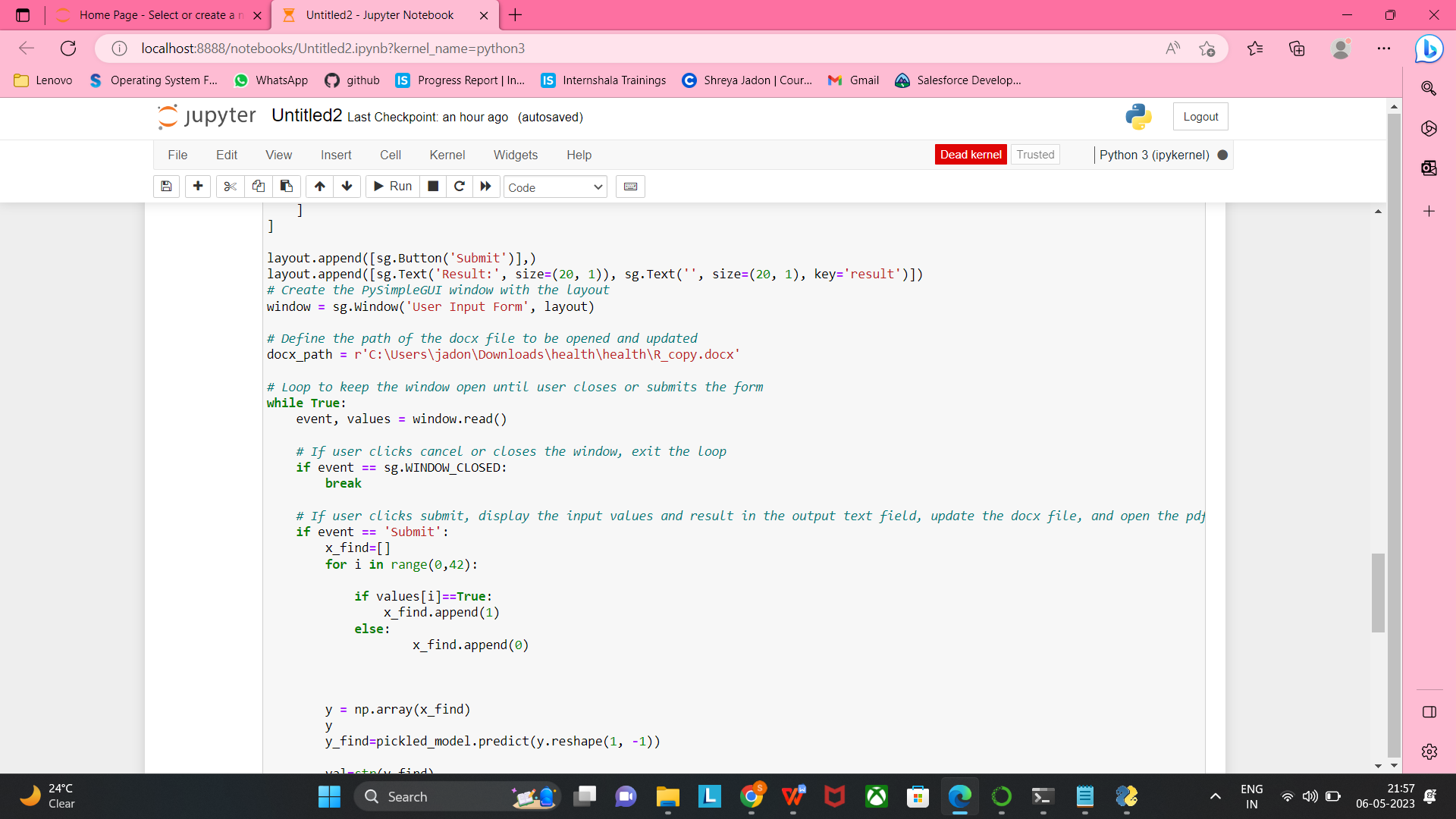
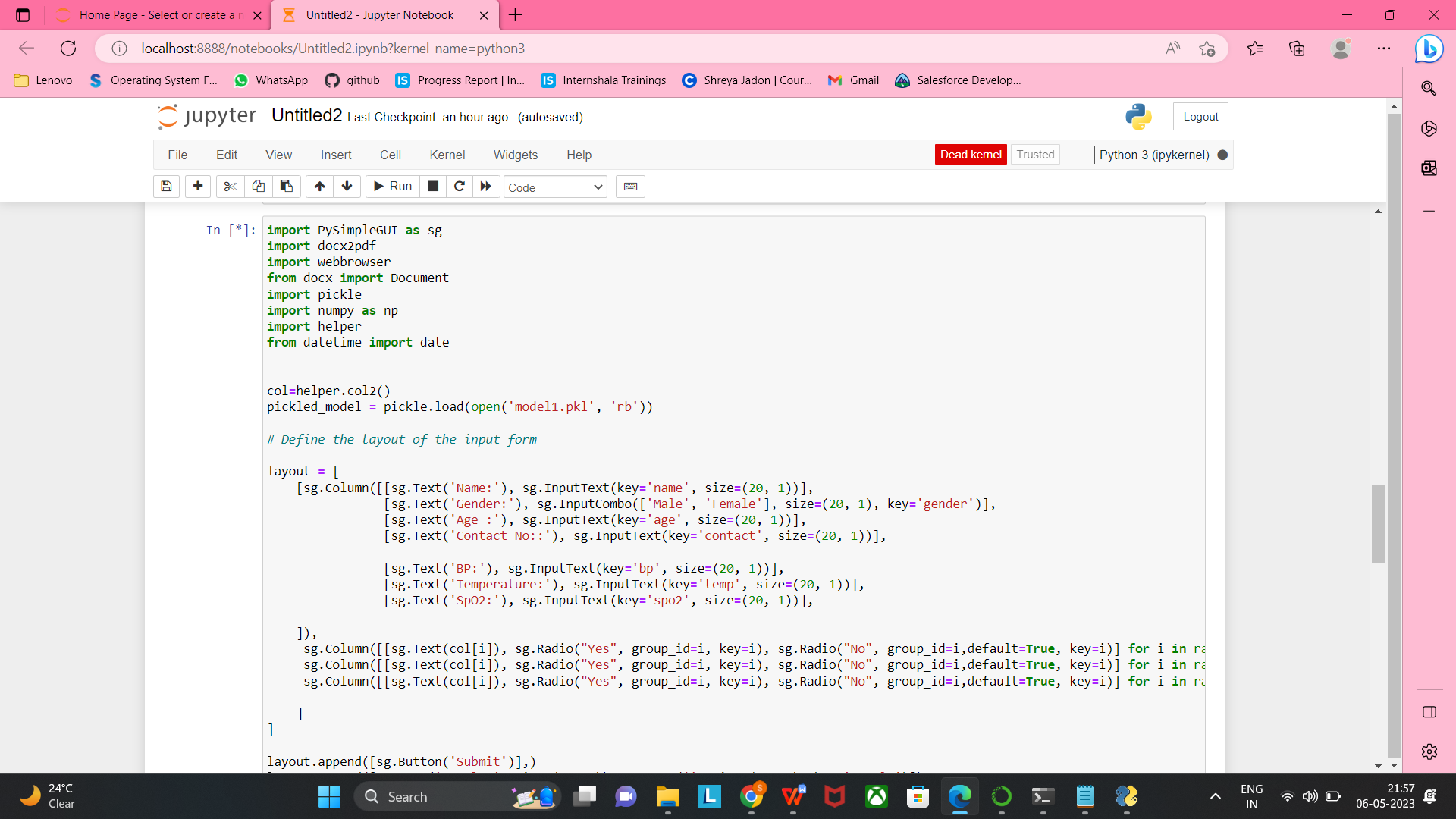
dsa('8976543210', values['contact'])

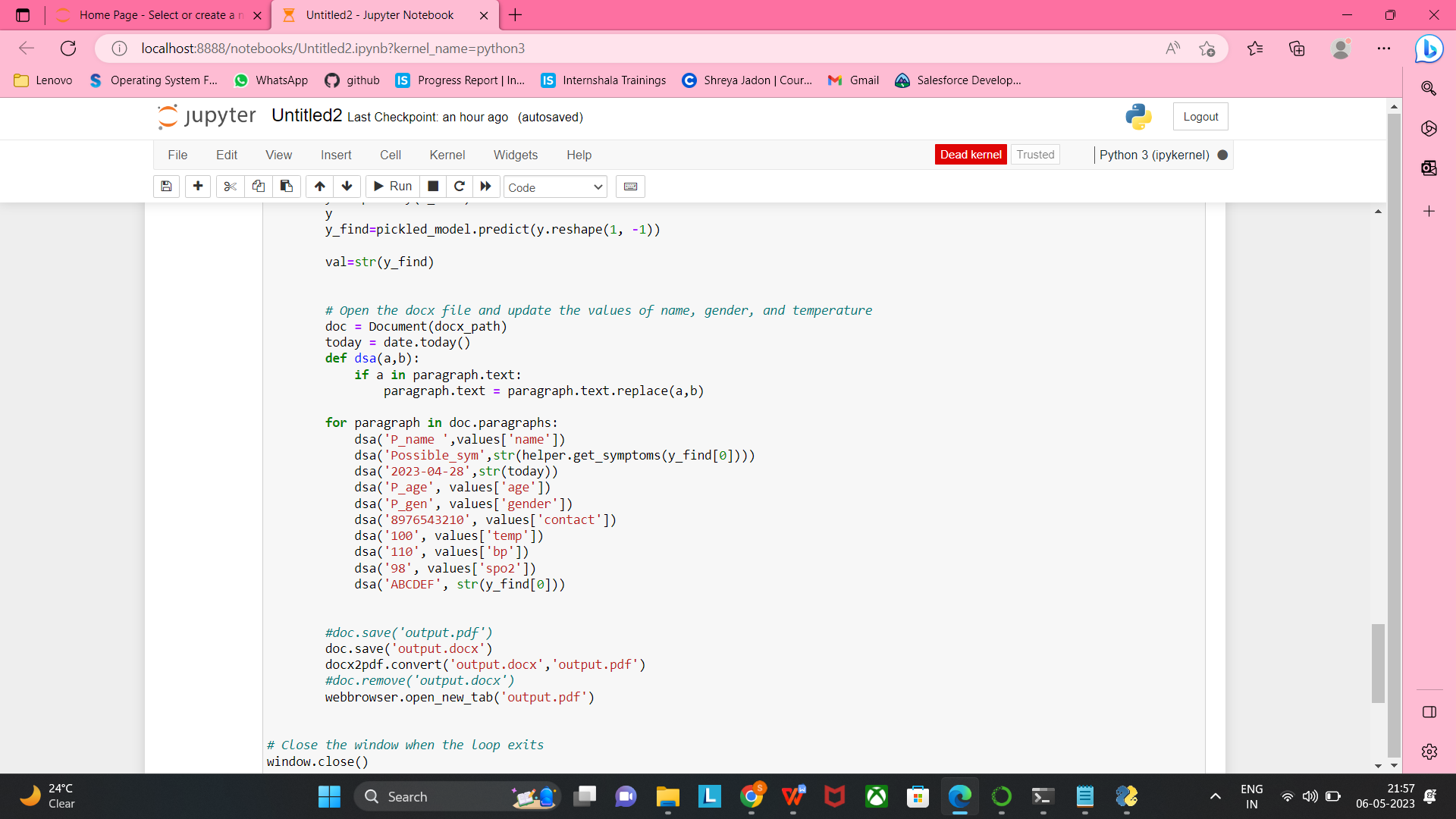
dsa('100', values['temp'])

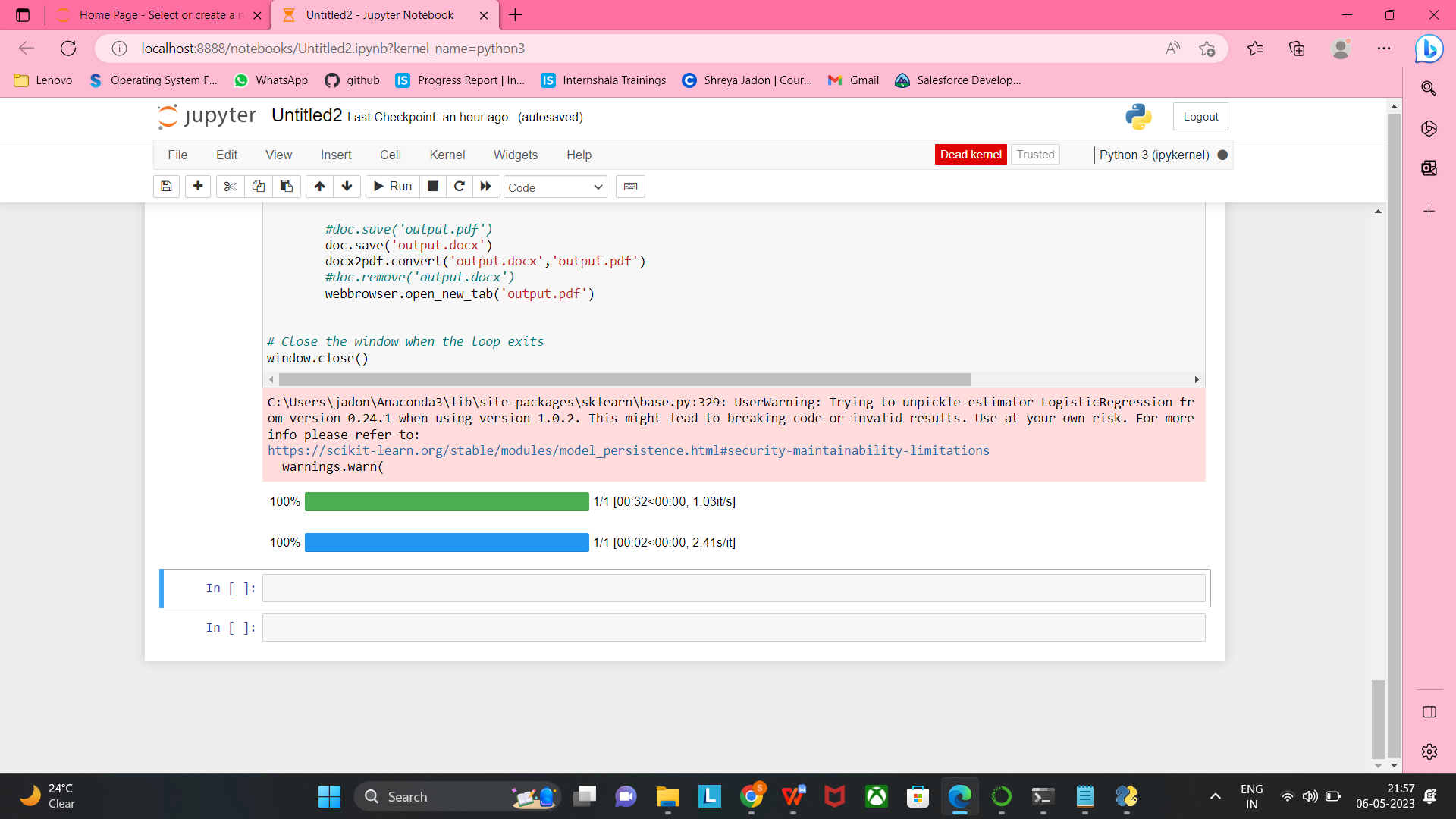
dsa('110', values['bp'])

dsa('98', values['spo2'])

dsa('ABCDEF', str(y\_find[0]))

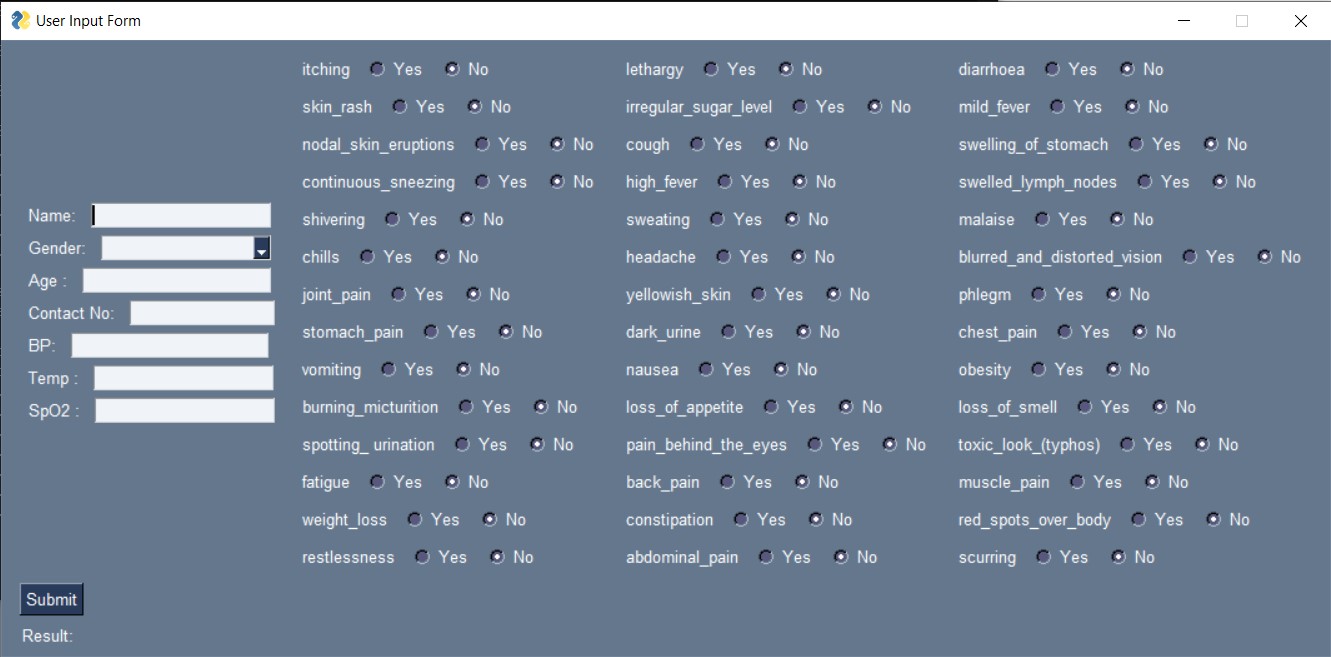
****

****

****

After we run this code, a user input form will be opened which will take user’s details like:-

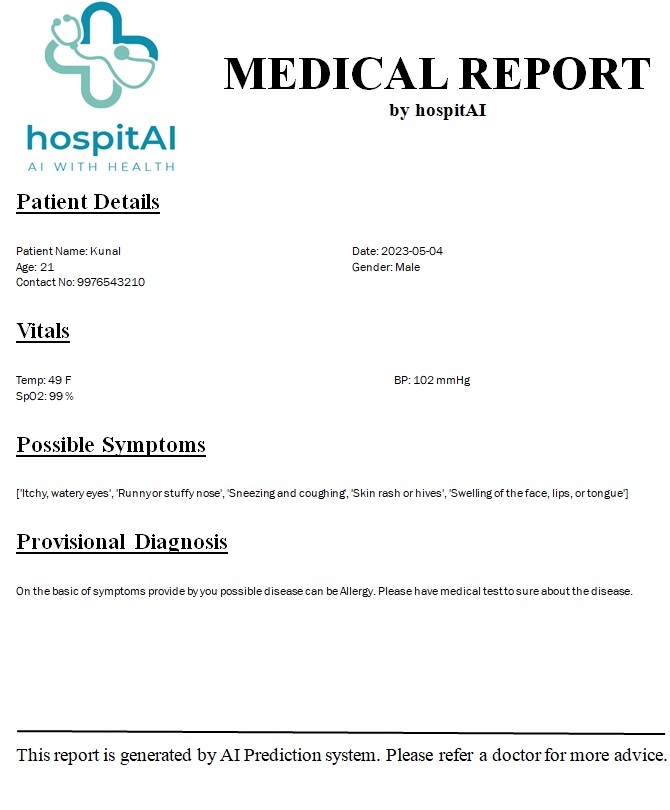
* Name
* Gender
* Age
* Contact No.
* BP
* Temperature
* SpO2



In this User Input form we have to enter user’s detail and the symptoms that the user currently have .

After we submit this user form, a pdf file will be generated which will display patient’s details like:

* Name , Age , gender , Date , Contact number
* Vitals
* Possible Symptoms
* Provisional Diagnosis



**References**

1. Lasko, T. A., Denny, J. C., & Levy, M. A. (2013). Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. PLoS ONE, 8(6), e66341.
2. Kim, J., & Kim, J. H. (2018). Machine learning-based prediction of medication nonadherence: A nationwide claim-based survey study. BMC Medical Informatics and Decision Making, 18(1), 74.
3. Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. The New England Journal of Medicine, 380(14), 1347-1358. - Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. Classification in BioApps, 1-23.

[4] Wang, Y., Huang, C., Peng, Y., & Zheng, Y. (2021). Health data analytics with machine learning for patient care. Journal of Healthcare Engineering, 2021, 5591862. https://doi.org/10.1155/2021/5591862

[4] Socher, R., et al.: “Semi-supervised recursive autoencoders for predicting sentiment distributions” in Proceedings of EMNLP '11 - the Conference on Empirical Methods in Natural Language Processing, ISBN: 978-1-937284-11-4, pp. 151-161

[5] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: “Lexicon-Based Methods for Sentiment Analysis”, in “Computational Linguistics”, June 2011, Vol. 37, No. 2, pp. 267-307.

[6] Chaovalit, P., Zhou, L.:. “Movie review mining: A comparison between supervised and unsupervised classification approaches”, in Proceedings of the Hawaii International Conference on System Sciences (HICSS), 2005.

[7] Esuli, A. Sebastiani, F.: “SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining”, in Proceedings of LREC-2006, Genova, Italy.

[8] Baldini, N., Neri, F., Pettoni, M.: “A Multilanguage platform for Open Source Intelligence”, Data Mining and Information Engineering 2007, 8th International Conference on Data, Text and Web Mining and their Business Applications, The New Forest (UK), Proceedings, ISBN: 978-184564-081-1, WIT Transactions on Information and Communication Technologies, Vol 38, June 2007, 18-20.

[9] Neri, F., Pettoni, M.: “Stalker, A Multilanguage platform for Open Source Intelligence”, Open Source Intelligence and Web Mining Symposium, 12th International Conference on Information Visualization, Proceedings, ISBN:978-0-7695-3268-4, pp. 314-320, IEEE Computer Society, LSBU, London (UK), July 2008, 8-11.

[10] Neri, F., Geraci, P.: “Mining Textual Data to boost Information Access in

OSINT”, Open Source Intelligence and Web Mining Symposium, 13th International Conference on Information Visualization, IV09, Proceedings, ISBN: 978-0-7695-3733-7, pp. 427- 432, IEEE Computer Society, Barcelona (ES), 16- 17/07/2009.

[11] Neri, F., Geraci, P.: “Overcoming Linguistic Barriers in Open Source Intelligence”, Governmental Sidebar, Semantic Conference, San Jose (CA), USA, 17/06/2009.

[12] Baldini, N., Bini, M., “Focuseek searchbox for digital content gathering”, AXMEDIS 2005 - 1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution, Proceedings Workshop and Industrial pp. 24-28 (2005).

[13] Diligenti, M., Coetzee, F. M., Lawrence, S., Giles, C. L., Gori, M.: “Focused Crawling Using Context Graphs”, in Proceedings of 26th International Conference on Very Large Databases, VLDB, pp. 527- 534, 10/9 - 12/9 (2000).

[14] P. Vossen, et al.: “KYOTO: A System for Mining, Structuring, and Distributing Knowledge Across Languages and Cultures”, Proceedings of LREC 2008, Marrakech (MO), 28-30/05/2008.

[15] Neri, F., Raffaelli, R.: “Text Mining applied to Multilingual Corpora”, Knowledge Mining: Proceedings of the NEMIS 2004 Final Conference, Springer Verlag Pub., Spiros Sirmakessis Ed., ISBN-13: 978-3540250708 (2004).

[16] McCord, M. C.: “Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars”. Natural Language and Logic 1989: 118- 145 (1989).

[17] McCord, M. C.: “Slot Grammars”, American Journal of Computational Linguistics 6(1): 31-43 (1980).

[18] Cascini, G., Neri, F.: “Natural Language Processing for Patents Analysis and Classification”, ETRIA World Conference, TRIZ Future 2004, Florence, Italy Proceedings, Cascini Ed., pp. 199-212, Firenze (IT), 03- 05/11/2004.

[19] Camillo, F., Neri, F.: “Sentiment Mining: Automatic linguistic analysis and natural language understanding in an e-CRM systems”, EURISBIS 2009, Proceedings, F.Mola et al. (Eds), TILAPIA, p. 136, Cagliari (IT), 01/06/2009.

[20] Neri, F., Geraci, P., Camillo, F.: “Monitoring the Web Sentiment”, the Italian Prime Minister's case, 2010 International Conference on Advances in Social

Page | **56**

Networks Analysis and Mining, 978-0-7695-4138-9/10 © 2010 IEEE Computer Society, DOI 10.1109/ASONAM.2010.26. Winner of Best poster Springer Award. Odense (DK), 9-11/08/2010.

[21] Neri, F., Aliprandi, C., Camillo, F.: “Mining the Web to Monitor the Political Consensus”, U.K. Wiil (ed.), Counterterrorism and Open Source Intelligence, Lecture Notes in Social Networks 2, DOI 10.1007/978-3-7091- 0388-3-19, © Springer-Verlag/Wien 2011.

[22] Siliato, F. “La fine del Tg1, Da Veltroni a Minzolini”, e-book, Europa Ed.

[23] Lepido, D. “La morte del Tg1: un ebook spiega la sconfitta di Minzolini”, blog I bastioni di Orione, 15/12/2011.

[24] Auditel, Ascolti Tv 27 novembre 2011.

[26] Tecce, C. “Fuga dal TG1”, Il Fatto Quotidiano, 01/09/2010

[27] Hutto C.J., Gilbert E., VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text, AAAI, 2014.

[28] Anees Ul Hassan, Jamil Hussain, Musarrat Hussain, Muhammad Sadiq, Sungyoung Lee,’Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression’, 2017 International Conference on Information and Communication Technology Convergence (ICTC).

[29] Persia, F., & D’Auria, D. (2017). A Survey of Online Social Networks: Challenges and Opportunities. 2017 IEEE International Conference on Information Reuse and Integration (IRI).

[30] Rosa, R. L., Rodriguez, D. Z., Schwartz, G. M., de Campos Ribeiro, I., & Bressan, G. (2016). Monitoring system for potential users with depression using sentiment analysis. 2016 IEEE International Conference on Consumer Electronics (ICCE).

[31] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, "Lexicon-based methods for sentiment analysis", Comput. Linguistics, vol. 37, no. 2, pp. 267-307, 2011.

[32] A. kaushik, S. Naithani, ³AStudy on Sentiment Analysis: Methods and Tools´ International journal of Science and Research, vol. 4, 2015, pp 287- 291.

[33] Y. Rao, Q. Li, X. Mao, and L. Wenyin, ³Sentiment topic models for social emotion mining,´ Information Sciences, vol. 266, pp. 90- 100, 2014.

[34] D. Jiang, Xiangfeng Luo, Junyu Xuan and Zheng Xu, ³ Sentiment computing for the news event based on the social media Big Data´ IEEE Access pp. 1-6, 2016.

[35] D. Tang, B. Qin, and T. Liu, ³Learning semantic representations of users and products for document level sentiment classification,´ in Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2015.

[36] J. Xuan et al., "Uncertainty analysis for the keyword system of web events", IEEE Trans. Syst. Man Cybern. Syst., vol. 46, no. 6, pp. 829-842, 2016.

[37] Abinaya. R, Aishwaryaa. P, Baavana. S, ³Automatic Analysis of

Page | **58**

User reviews´, IEEE International conference on Technological Innovations in ICT for Agriculture and Rural Development, 2016, pp 158- 162.

[38] Y. Yamamoto, T. Kumamoto, A. Nadamoto, "Multidimensional sentiment calculation method for twitter based on emoticons", Int. J. Pervas. Comput. Commun., vol. 11, no. 2, pp. 212-232, 2015.

[39] Fuji Ren,Ye Wu,” Predicting User-Topic Opinions in Twitter with Social and Topical Context”, IEEE Trans. on affective computing, vol. 4, no. 4, October-December 2013.

[40] Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi, and Tao Li,”Dual Sentiment Analysis:Considering Two Sides of One Review”, IEEE Trans.on Knowledge and Data Engineering, 2015

[41] Xin Chen,Mihaela Vorvoreanu, and Krishna Madhavan,” Mining Social Media Data for Understanding Students’ Learning Experiences” IEEE trans. on learning technologies, vol. 7, no. 3, JulySeptember 2014.

[42] Danushka Bollegala, David Weir, and John Carroll,” Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus”, IEEE trans. on knowledge and data engineering, vol. 25, no. 8, August 2013.

[43] Shulong Tan, Yang Li, Huan Sun, Ziyu Guan, Xifeng Yan, Jiajun Bu, ,Chun Chen, and Xiaofei He,” Interpreting the Public Sentiment Variations on Twitter”, IEEE trans. on knowledge and data engineering, vol. 26, no. 5, May 2014.

[44] Chenghua Lin, Yulan He, Richard Everson, Member, IEEE, and Stefan Ru¨ger,” Weakly Supervised Joint Sentiment-Topic Detection from Text”, IEEE trans. on knowledge and data engineering, vol. 24, no. 6, June 2012.

[45] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, Ming Zhang,” Topic Sentiment Analysis in Twitter: A Graph-based Hashtag Sentiment Classification Approach”, ACM, CIKM’11, October 24–28, 2011, Glasgow, Scotland, UK, 2011.

[46] Andrius Mudinas, Dell Zhang, Mark Levene,” Combining Lexicon and Learning based Approaches for Concept-Level Sentiment Analysis”, WISDOM’ 12, August 12, 2012, Beijing, China Copyright 2012, ACM.

[47] Xiaohui Yu,Yang Liu, Jimmy Xiangji Huang, and Aijun An,,” Mining Online

Reviews for Predicting Sales Performance: A Case Study in the Movie Domain”, IEEE Trans.On knowledge and data engineering, vol. 24, no. 4, April 2012.

[48] LI Bing, Keith C.C. Chan, Carol OU,” Public in Twitter Data for Prediction of A Company’s Stock Price Movements”, 2014 IEEE, 11th International Conference on e-Business Engineering.

[49] Walaa Medhat , Ahmed Hassan , Hoda Korashy,”Algorithms and Applications: A survey”, Ain Shams Engineering Journal (2014).

[50]. Rosenthal, Sara, Noura Farra, and Preslav Nakov. "SemEval-2017 task 4: Sentiment analysis in Twitter." Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). 2017

[51]. Pontiki, Maria, et al. "SemEval-2016 task 5: Aspect based analysis." ProWorkshop on (SemEval-2016). Association for Computational Linguistics, 2016.

[52]. Ahmed, Khaled, Neamat El Tazi, and Ahmad Hany Hossny. " over Social Networks: An Overview." Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE, 2015.

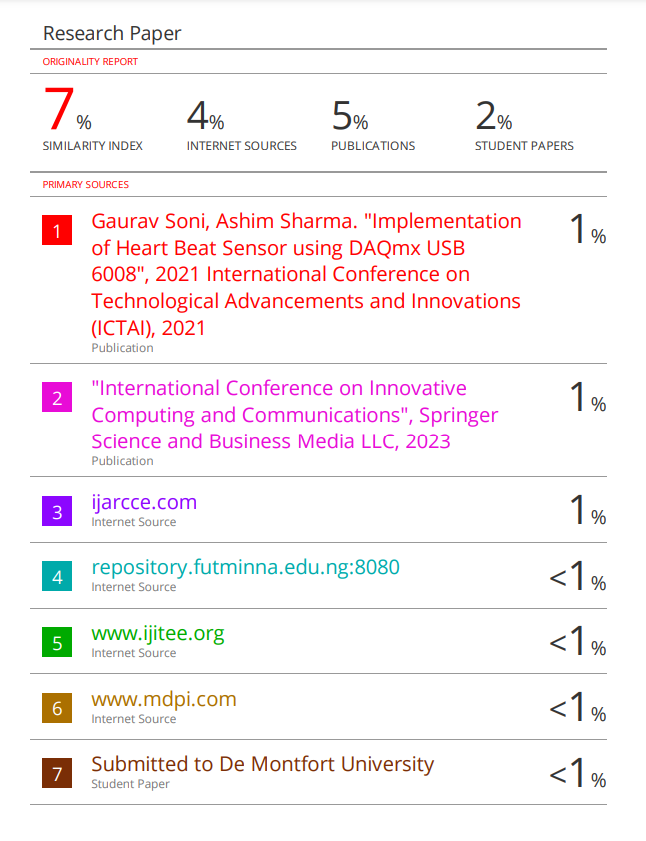
[53]. Fang, Xing, and Justin Zhan. "Using product review data." Journal of Big Data 2.1 (2015) [5]. Singh, Prabhsimran, Ravinder Singh Sawhney, and Karanjeet Singh Kahlon. "demonetization of 500 & 1000 rupee banknotes by Indian government." ICT Express (2017)

[54]. Amolik, Akshay, et al. International Journal of Engineering and Technology 7.6 (2016)

[55]. Kharche, S. R., and Lokesh Bijole. International Journal of Computer Science and Applications 8 (2015)

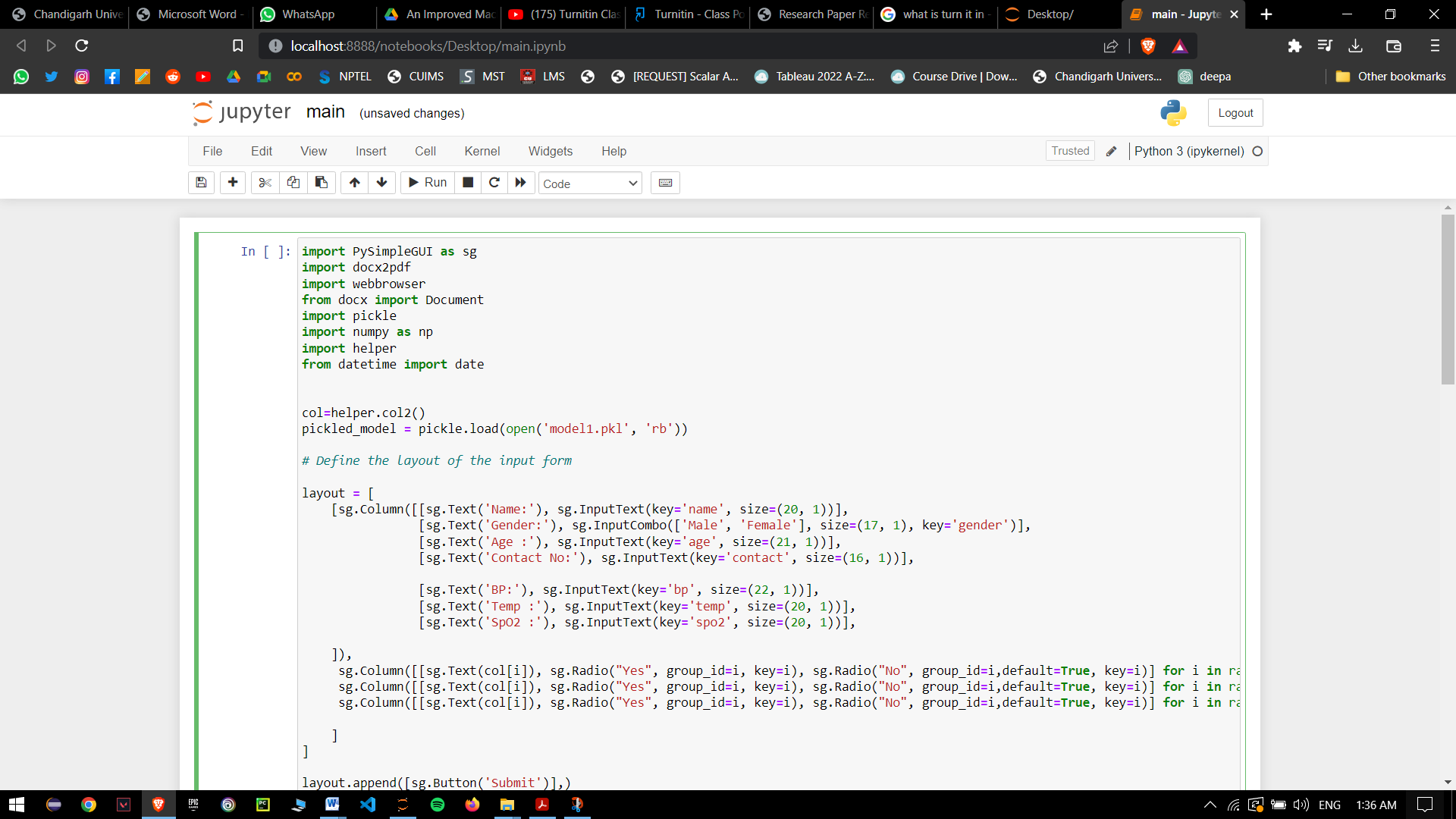
[56]. (OLDFLN$OSDVODQ %XUDNDQG (UGR÷DQ (UGR÷DQ"User-weighted sentiment analysis for financial community on Twitter." Innovations in Information Technology (IIT), 2015 11th International Conference on. IEEE, 201

**Appendeix**

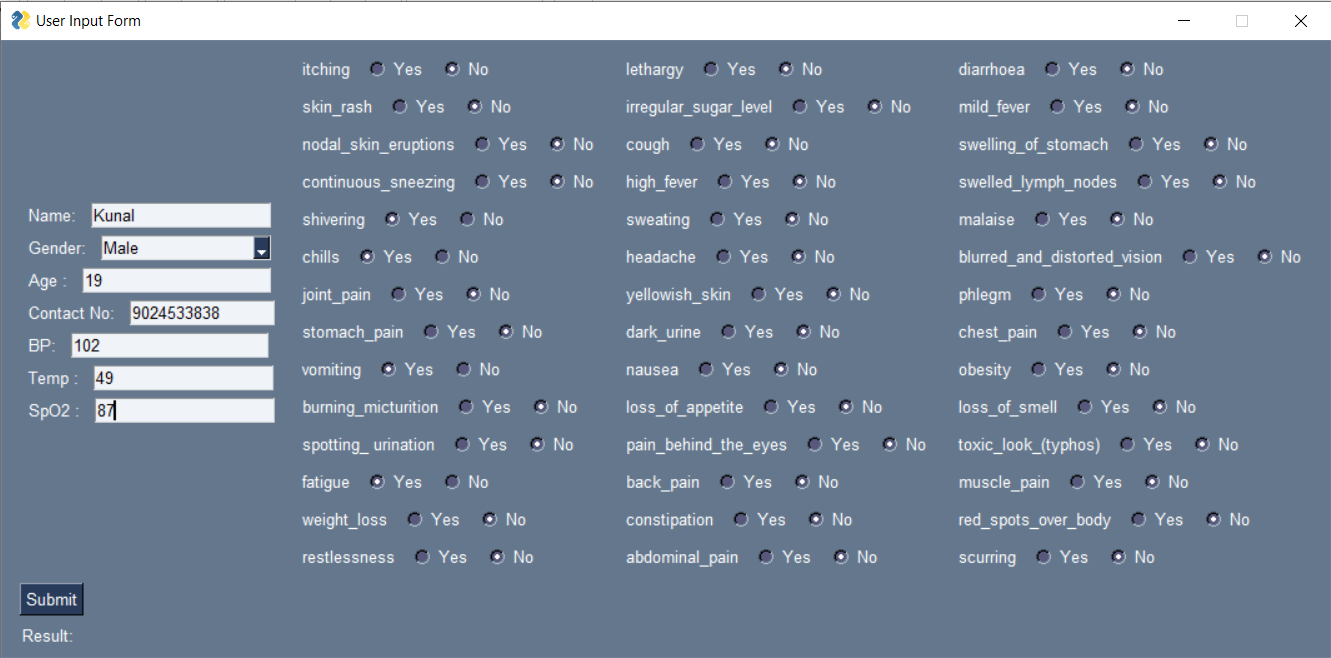
**Plagiarism Report**

**User Manual**

1.Open Main.py File and compile



2.An application window will pop up that will ask for paient’s details and symptoms. Fill all the required details and then click on Submit



3.Your medical report will be generated with more possible symptoms and possible disease that you’ve been suffering from

