International Journal of Engineering and Management Research
e-ISSN: 2250-0758 | p-ISSN: 2394-6962
https://ijemr.vandanapublications.com

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

Tuberculosis Prediction using KNN Algorithm

Viswanatha V[1], Ramachandra A.C[2], Ankita R Togaleri[3] and Nisarga S Gowda[4]
[1]Assistant Professor, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, INDIA
[2]Professor, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, INDIA
[3]Student, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, INDIA
[4]Student, Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, INDIA

[1]Corresponding Author: viswas779@gmail.com

Received: 04-07-2023 Revised: 20-07-2023 Accepted: 04-08-2023

ABSTRACT

In this paper, a machine learning model is used to develop a model that is used for tuberculosis prediction. Tuberculosis is known to be one of the top reasons for death from an infectious agent that affects the lungs and continues to threaten the human population on a wider basis. According to WHO, tuberculosis is a serious threat to the human population after HIV/AIDS. It is estimated by the World Health Organization (WHO) that 1/3rd of the global population is infected with TB and that seven to eight million new cases of TB occur each year across the globe Because the disease is difficult to differentiate between the common cold, it takes a long time to decide the patient is affected by the disease. So we use the detection of tuberculosis by utilizing the K-NN algorithm method for classification and HOG as feature extraction. K-NN abbreviated as K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.

The data provided K-NN model should be labelled one. Then these datasets are given to a training model where the training process of the model is being undergone. Once the training is completed, the next step is to predict the output. For this process, we have to provide new data that may or may not belong to the dataset, so that the model can predict the output of it. If the prediction is wrong, again the training is done until we get the actual output matching with the desired output given by the designer for verification purposes. This is the basic working process under the K-NN algorithm. The data that is used for this separation is a Tuberculosis dataset that contains various information about the different symptoms that are helpful in detecting tuberculosis effectively. Here it is used in the early detection of tuberculosis which helps save millions of people which might otherwise lead to death because of lack of detection. ML model helps to improve the efficiency in detecting by considering various symptoms. ML models are more accurate at differentiating even the slightest difference that deviates from the data that was used to train the model. Unlike the manpower we fail to detect the slightest as we notice the symptoms only after they become more severe. The accuracy of this model was found to be 98%. The following model uses a dataset consisting of data that contrasts between males and females and the various symptoms are shown in them. It also contrasts the severity of these two.

Keywords-- Lung Disease, Tuberculosis, k-NN Algorithm

I. INTRODUCTION

Tuberculosis is an infectious disease caused by bacteria whose scientific name is Mycobacterium tuberculosis. It was first isolated in 1882 by a German physician named Robert Koch who received the Nobel Prize for this discovery.

There are mainly three types of TB: Active TB, Miliary TB, and Latent TB Infection. The major symptoms of this disease include coughing, chest pain, and the coughing up of blood. TB is treated using antibiotics and can be fatal without treatment. BCG is a vaccine for tuberculosis disease. TB most commonly affects the lungs Fig.1, Fig.2, but also can involve almost any organ of the body. Tuberculosis (TB) is one of the most prevalent infections of human beings and contributes considerably to illness and death around the world [1]-[2]. Tuberculosis is more prone to lower socioeconomic sectors of the population and marginalized sections of the community. Tuberculosis is not a hereditary disease. This is a direct communicable disease. Whenever a patient having active tuberculosis coughs or sneezes into an open space, bacteria cause TB to come out in the aerosol [3]. This aerosol can infect any person who happens to inhale it. Symptoms of tuberculosis can vary depending on the stage of the complaint. In the early stages, there may be no conspicuous symptoms, or the symptoms may be mild and non-specific, similar to persistent cough, weight, fatigue, fever, loss, night sweats, and loss of appetite. As the disease progresses, symptoms may become more severe and include coughing up blood, casket pain, and difficulty in breathing

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

Diagnosis of tuberculosis involves colorful tests, including a tuberculin skin test, blood tests, chest X-rays, and foam analysis. Treatment generally involves a combination of antibiotics taken over several months to exclude the bacteria and help the development of medicine-resistant strains, generally used specifics for TB treatment include isoniazid, rifampin, pyrazinamide, and ethambutol [4]-[5].

Patterns in the data by connecting existing patterns of data with the new data.

Figure 1: This image shows the X-ray of healthy lungs

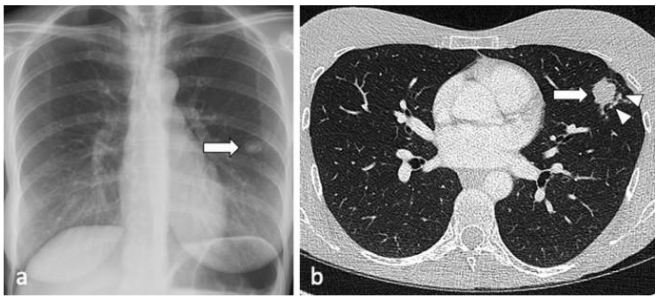

Figure 2: This figure shows the X-ray image of lungs infected by Tuberculosis

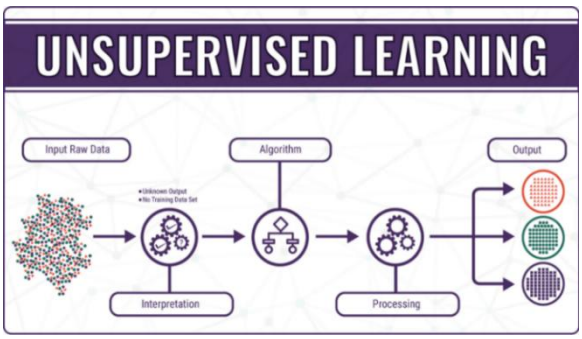

Figure 3a: shows an unsupervised model

Fig.3a While in unsupervised learning, data does not yet have any pattern, and the unsupervised learning objective is to find patterns in data Fig.3b. The goal of the KNN algorithm is to classify new objects by attribute and training samples

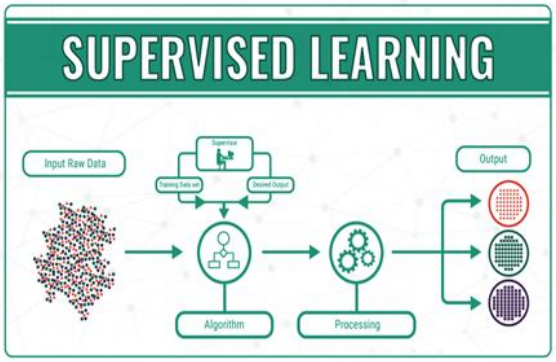

Figure 3b: indicates a supervised model

The KNN algorithm includes methods that use the supervised algorithm. The difference between supervised learning and unsupervised learning is that supervised attribute and training sample.

The K-Nearest Neighbour (KNN) algorithm is a method of classifying the object based on the learning data that is closest to the object. Learning data is projected into multiple dimensioning spaces, each of which dimensions represent the features of the data [6]-[7].

The k-NN algorithm assumes the similarity between new data and available data cases. It puts the new data case into a category that is almost similar to the available categories. K-NN is a non-parametric algorithm. It doesn't make any assumption on underlying data. K-NN is also referred to as a lazy learner algorithm as it doesn't learn from the training set immediately instead it stores the dataset and performs action at the time of classification [8]-[9].

The k-NN is a non-linear algorithm, which means it doesn't capture complex relationships between features and the presence of TB. The k-NN algorithm assumes similarity between the new case and available cases and puts the new case into the category that is most similar to the available categories. The k-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears, it can be easily classified into a good suite category using the K- NN algorithm. Fig.4 KNN is a flexible algorithm that can handle different types of data, including categorical and numerical variables. This flexibility is valuable in TB prediction, where a variety of data types, such as demographics, symptoms, and test results, may need to be considered [10]-[11].

KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

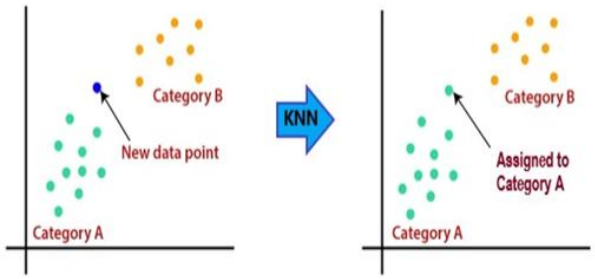

Figure 4: indicates the application of KNN

II. LITERATURE REVIEW

Tuberculosis (TB) is a kind of irresistible sickness brought about by Mycobacterium tuberculosis, which goes after the lungs but can likewise go after the bones, digestion tracts, or organs. During the Coronavirus pandemic, TB cases in Indonesia likewise expanded. TB and Coronavirus had comparative side effects like hack, fever, and breathing trouble, with the goal that TB victims should be given serious treatment to stay away from Coronavirus [12]. In anticipating a sickness, well-being laborers should simply decide, hence it is important to do an early finding to lessen the transmission of TB locally. There are numerous calculation techniques utilized in directing information examination, for this study the creators utilize K-Closest Neighbor (K-NN) calculation and Strategic Relapse as correlation. Trial results involving accessible dataset gathered from wellbeing focuses in the Maura Enim Area of South Sumatra Territory show that the K-NN calculation gives the best exactness of 89% on the dataset with preparing to test information proportion of 80%:20%, while the Strategic Relapse gives the best precision of 96% on 70%:30% proportion. The examination system talked about in this paper might be considered as a device for the power to anticipate and make essential moves to forestall the TB spreading [13].
Albeit the death pace of Tuberculosis (TB) has decreased emphatically over the course of the last ten years, recognition of TB stays a test and a functioning exploration issue lately. Contrasted with further developed imaging frameworks like SPECT, CT, and PET sweep, the standard Chest X-Beam (CXR) imaging is common, modest, quick, and with substantially fewer radiation dosages for identifying TB patients. Since aspiratory sicknesses can make a few mathematical minor deviations from lung shape, examination of the lung shape assumes a viable part in deciding the kind of lung illness. The objective of this study is to find the ideal element vectors for TB discovery in CXR pictures. In this review, utilizing form-based shape descriptors and Two Layered Head Part Examination (2DPCA), we proposed a clever way to deal with highlight determination of the lung shape to TB identification. In our insight, surface-based highlights are utilized in all of the earlier examinations on TB identification rather than form-based highlights. Utilizing the K-Closest Neighbor classifier on one of the free datasets (specifically the Montgomery dataset), we accomplished the most extreme exactness (ACC) and AUC of 92.86% and 91.67% separately.

Tuberculosis (TB), an irresistible infection brought about by Mycobacterium tuberculosis (M.tb), causes the biggest number of passing worldwide for any bacterial illness requiring novel determination and treatment procedures. High-throughput sequencing techniques produce a lot of information that could be taken advantage of in deciding multi-drug safe (MDR-TB) related transformations. The current work is a computational system that utilizes man-made brainpower (artificial intelligence) based AI (ML) approaches for anticipating opposition in the qualities rpoB, inhA, katG, pncA, gyrA, and gyrB for the medications rifampicin, isoniazid, pyrazinamide, and fluoroquinolones. The single nucleotide varieties were addressed by a few grouping and underlying elements that show the impact of transformations on the objective protein coded by every quality. We utilized ML calculations - guileless Bayes, k closest Neighbor, support vector machine, and counterfeit brain organization, to fabricate the forecast models. The arrangement models had a typical exactness of 85% across undeniably inspected qualities and were assessed on an outside inconspicuous dataset to exhibit their application. Further, sub-atomic docking and atomic element recreations were performed for wild sort and anticipated opposition causing freak protein and hostile to TB drug edifices to concentrate on their effect on the adaptation of proteins to affirm the noticed aggregate [14].

With current innovative turns of events, AI has become one of the most famous techniques, one of the famous AI calculations is k-closest neighbors (KNN). AI has been generally utilized in the clinical field to dissect clinical datasets, in this study the k-closest neighbors (KNN) machine learning calculation will be involved in view of its great degree of exactness in acknowledgment and is remembered for the administered learning calculation bunch. The outcomes showed the k-closest neighbors (KNN) technique in perceiving x-beam pictures of tuberculosis (TB) utilizing SURF including extraction with a typical exactness of 73% [15].

[5] Software engineering assumes a significant part in current powerful well-being frameworks. Given the cooperative idea of the demonstrative interaction, PC innovation offers significant types of assistance to medical care experts and associations, as well as to patients and their families, specialists, and leaders. In this manner, any advancements that work on the demonstrative cycle while keeping up with quality and security are essential to the improvement of the medical services field. Numerous sicknesses can be probably analyzed during their underlying stages. In this review, all created methods were applied to tuberculosis (TB). Hence, we propose an enhanced AI-based model that extricates ideal surface elements from TB-related

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

pictures and chooses the hyper-boundaries of the classifiers. Expanding the precision rate and limiting the number of attributes separated are our objectives. All in all, this is a perform multiple tasks streamlining issue. A hereditary calculation (GA) is utilized to pick the best elements, which are then taken care of into a help vector machine (SVM) classifier. Utilizing the Picture CLEF 2020 informational collection, we led tests utilizing the proposed approach and accomplished fundamentally higher precision and improved results in examination with cutting-edge works. The acquired exploratory outcomes feature the proficiency of altered SVM classifier contrasted and other standard ones.

This examination work depends on the different trials performed for the discovery of lung tuberculosis utilizing different strategies like sifting, and division, including extraction and order. The outcomes got from these investigations are talked about in this paper. Lung tuberculosis is a bacterial disease that causes a greater number of death on the planet than some other irresistible illnesses. Two billion individuals are contaminated with tuberculosis from one side of the planet to the other. Lung tuberculosis is an illness brought about by microscopic organisms known as Mycobacterium tuberculosis or Tubercle bacillus. This exploration work endeavors to recognize strategies by which patients, who require a second assessment for a generally distinguished outcome, can save a ton of cash. When we get an X-beam picture and info, pre-handling strategies like the Gaussian channel, and the middle channel is applied. These channels help to eliminate undesirable clamor and help to get fine textural highlights. The result got from this is taken as input and applied to the watershed division and dark level division which assists with zeroing in on the lung region of the acquired results. Yield from these division techniques is melded to get a District of Interest (return for money invested). From the return for money invested, the measurable elements like region, significant hub, minor pivot, whimsy, mean, kurtosis, skewness, and entropy are separated. At long last, we use KNN, Successive insignificant streamlining (SMO), and basic straight relapse order techniques to identify lung tuberculosis. The outcomes acquired in this paper propose KNN classifier performs well than the other two classifiers.

World Health Association (WHO) has recorded Tuberculosis (TB) as one of the main 10 explanations behind death and an early determination will assist with restoring the patient by giving appropriate treatment. TB typically influences the lungs and an exact bio-imaging plan will be adept to analyse the contamination. This exploration expects to carry out a mechanized plan to identify TB contamination in chest radiographs (X-beam) utilizing a picked Profound Learning (DL) approach. The essential target of the proposed conspire is to accomplish better grouping exactness while identifying TB in X-beam pictures. The proposed conspire comprises of the accompanying stages to be specific, (1) picture assortment and pre-handling, (2) highlight extraction with pre-prepared VGG16 and VGG19, (3) Mayfly-calculation (Mama) based ideal element determination, (4) sequential component connection and (5) twofold characterization with a 5-crease cross approval. In this work, the exhibition of the proposed DL conspire is independently approved for (1) VGG16 with regular highlights, (2) VGG19 with traditional elements, (3) VGG16 with ideal elements, (4) VGG19 with ideal elements and (5) linked double profound elements (DDF). All exploratory examinations are led and accomplished utilizing the MATLAB® program. The trial result affirms that the proposed framework with DDF yields a characterization precision of 97.8%using a K Closest Neighbor (KNN) classifier.

In this paper, we have focused on AI (ML) highlight choice (FS) calculations for distinguishing and diagnosing multidrug-safe (MDR) tuberculosis (TB). MDR-TB is an inclusive general medical condition, and its initial location has been one of the consuming issues. The current review has been led in the Malakand Division of Khyber Pakhtunkhwa, Pakistan, to additional add to information on the illness and to manage the issues of ID and early recognition of MDR-TB by ML calculations. These models likewise recognize the main elements causing MDR-TB disease whose study gives extra bits of knowledge into the matter. ML calculations, for example, irregular woodland, k-closest neighbors, support vector machine, strategic relapse, least outright shrinkage and choice administrator (Rope), counterfeit brain organizations (ANNs), and choice trees are applied to investigate the case-control dataset. This study uncovers that nearby contacts of MDR-TB patients, smoking, despondency, past TB history, ill-advised treatment, and break-in first-line TB treatment extraordinarily affect the situation with MDR. As needs be, weight reduction, chest torment, hemoptysis, and exhaustion are significant side effects. In view of exactness, awareness, and explicitness, SVM and RF are the recommended models to be utilized for patients' characterizations.

Marginal interferon-gamma (IFN-γ) results (close to the cut-off level of 0.35 IU/ml) happen in QuantiFERON (QFT) examines. We explored the presentation of elective biomarkers for the arrangement of dormant tuberculosis contamination (LTBI) status in pregnant ladies with fringe QFT IFN-γ reactions. Pregnant ladies (n = 96) were distinguished from a companion concentrate on in Ethiopia, in view of QFT-In addition to IFN-γ results (QFT-low: <0.20 IU/ml, n = 33; QFT-fringe: 0.20-0.70 IU/ml, n = 31; QFT-high: >0.70 IU/ml, n = 32), remembering 12 HIV-positive people for each gathering and with 20 HIV-pessimistic non-pregnant ladies from a similar partner with QFT IFN-γ <0.20 IU/ml as controls. Centralizations of 8 markers (IL-1ra, IL-6, IL-8, IP-10, MCP-1, MCP-2, osteopontin, and resistin) were estimated in entire blood QFT supernatants, animated independently with TB1 and TB2 antigens. K-closest Neighbor investigation (KNN) was utilized to arrange members with respect to the

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

probability of LTBI. Convergences of MCP-2, IP-10, and IL-1ra were higher in QFT-fringe contrasted with QFT-low members in both antigen excitements ($p < 0.001$). KNN characterization demonstrated a high probability of LTBI in 13/31 (42%) ladies with QFT-fringe IFN-γ results. MCP-2, IP-10, and IL-1ra communicated in entire blood after TB antigen excitement might be considered as elective biomarkers for grouping of LTBI status in pregnant ladies with fringe QFT IFN-γ results.
Pneumonic tuberculosis is an irresistible sickness brought about by Micro bacterial tuberculosis, which is one of the lower respiratory plot illnesses, which is to a great extent in the aspiratory tissue of the lung disease and afterward goes through a cycle known as the essential focal point of Ghon. Since the infection is troublesome and consumes most of the day to conclude the patient is impacted by the sickness Tuberculosis, then, at that point, the discovery of the patient influences Tuberculosis by using the K-NN strategy as order and Hoard as component extraction. Consequences of the characterization of positive determination with a sum of 234 examples from 330 examples or effectively.

III. METHODOLOGY

A. Tool Used

Anaconda Navigator is a desktop graphical user interface (GUI) that is included with the Anaconda distribution. Anaconda is a popular open-source distribution of Python and R programming languages, which is widely used in the field of data science and machine learning. Anaconda Navigator provides an intuitive interface that allows users to easily manage and launch applications, environments, and packages within the Anaconda ecosystem. It provides a central hub where users can access different tools, such as Jupyter Notebook, JupyterLab, Spyder, and RStudio, among others. These tools are commonly used for data analysis, scientific computing, and developing machine learning models. Through Anaconda Navigator, users can create and manage separate environments, which are isolated spaces where specific packages and dependencies can be installed. This allows users to work on different projects with their own unique configurations, preventing conflicts between packages. Additionally, Anaconda Navigator includes a package manager that simplifies the installation and updating of packages. Users can easily search for and install packages from Anaconda's extensive library, which contains numerous scientific computing and data analysis libraries. In summary, Anaconda Navigator provides a user-friendly interface for managing and accessing various tools, environments, and packages within the Anaconda distribution. It simplifies the process of setting up and working with data science and machine learning environments, making it a valuable tool for both beginners and experienced practitioners. Jupyter Notebook, formerly known as I Python Notebook, is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It is a popular tool among data scientists, researchers, and developers for interactive computing and data analysis. The main idea behind Jupyter Notebook is to provide an interactive environment where you can write and execute code in small sections called cells. Each cell can contain code written in languages such as Python, R, Julia, and many others, depending on the available kernels. This allows you to experiment, iterate, and visualize your data in a step-by-step manner. Jupyter Notebook supports a wide range of features and functionalities that make it a powerful tool for data analysis and exploration. Some of its key features include Code Execution: You can execute code cells individually or all at once, and the output is displayed inline with the code. Rich Output: Jupyter Notebook supports the display of rich media, including images, videos, interactive plots, and HTML elements, enhancing the presentation of your analysis. Markdown Support: You can write formatted text using Markdown syntax within Markdown cells, allowing you to add explanations, headings, lists, and other formatting elements. Collaboration and Sharing: Notebooks can be easily shared with others, making it convenient for collaborative work. Notebooks can be exported to different formats, such as HTML, PDF, and Markdown, facilitating sharing and publishing. Interactive Widgets: Jupyter Notebook supports interactive widgets that enable the creation of interactive user interfaces, enhancing the interactivity and user experience of your notebooks. Jupyter Notebook provides an environment that encourages reproducible and exploratory data analysis. It allows you to document your workflow, share your findings, and create interactive presentations. Whether you are developing machine learning models, conducting data analysis, or presenting your research, Jupyter Notebook is a versatile and powerful tool in the data science ecosystem.

B. Algorithm

The K-Closest Neighbors (KNN) calculation is a regulated grouping calculation that is usually utilized in AI. It is a nonparametric calculation, meaning it doesn't make any suspicions about the fundamental circulation of the information. It's significant that the progress of KNN in tuberculosis expectation relies upon the quality furthermore, representativeness of the preparation information, the decision of fitting highlights, and the legitimate setting of the hyperparameter K. Furthermore, KNN may not be the most reasonable calculation for all cases, and other AI procedures, for example, choice trees, support vector machines, or then again profound learning models, can likewise be utilized for The KNN calculation works in view of the rule that pieces of information with comparative elements tend to have a place with a similar class. It characterizes another information point by finding the K closest neighbors in the preparing dataset and doling out it to the greater part class among those neighbors K-closest neighbors (KNN) is an AI

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

calculation that can be utilized for foreseeing tuberculosis. KNN is a non-parametric calculation that doesn't make suppositions about the hidden information conveyance. A straightforward and natural calculation can be compelling in specific situations. This is the way KNN can be utilized for anticipating tuberculosis: 1) Data Preparation: Collect and preprocess the data relevant to tuberculosis prediction. This may include demographic information, medical history, symptoms, and diagnostic test results. 2) Feature Selection: Recognize the most applicable highlights that are probably going to add to the tuberculosis forecast. This step assists with lessening the dimensionality of the information and working on the effectiveness of the calculation.3) Training Data: Prepare a labeled dataset with known cases of tuberculosis, where each instance has associated features and a tuberculosis label (positive or negative). 4) Distance Calculation: The KNN algorithm uses a distance metric (e.g., Euclidean distance) to measure the similarity between instances in the feature space. Compute the distance between the new instance (for which tuberculosis prediction is required) and all instances in the training dataset.5) Finding K Neighbors: Select the K nearest neighbors to the new instance based on the calculated distances. K is a hyperparameter that needs to be set beforehand and determines the number of neighbors considered for prediction. 6) Voting: Among the K nearest neighbors, count the number of tuberculosis-positive and tuberculosis-negative instances. The class with the majority vote will be predicted as the label for the new instance. 7) Prediction: Assign the predicted tuberculosis label (positive or negative) to the new instance based on the majority vote. 8) Evaluation: Survey the presentation of the KNN model utilizing assessment measurements like exactness, accuracy, review, and F1-score. This step assists with deciding the viability of the calculation in foreseeing tuberculosis. It's worth noting that the success of KNN in tuberculosis prediction depends on the quality and representativeness of the training data, the choice of appropriate features, and the proper setting of the hyperparameter K. Additionally, KNN may not be the most suitable algorithm for all cases, and other machine learning techniques, such as decision trees, support vector machines, or deep learning models, can also be used for tuberculosis prediction depending on the specific requirements and characteristics of the dataset. K-nearest neighbors (KNN) algorithm can be useful in predicting tuberculosis (TB) in several ways: 1) Non-linearity: KNN is a non-linear algorithm, which means it can capture complex relationships between features and the presence of TB. TB diagnosis involves considering multiple factors, such as symptoms, medical history, and test results. KNN's ability to handle non-linear relationships makes it suitable for capturing the complex interactions between these factors.2) Pattern Recognition: KNN excels at pattern recognition. By comparing the feature vectors of a new patient with those of known TB cases in the training dataset, KNN can identify similarities and patterns that indicate the likelihood of TB. The algorithm makes predictions based on the characteristics of similar patients from the training set. 3) Flexibility: KNN is a flexible algorithm that can handle different types of data, including categorical and numerical variables. This flexibility is valuable in TB prediction, where a variety of data types, such as demographics, symptoms, and test results, may need to be considered.4) Lack of Assumptions: KNN does not assume any underlying data distribution, making it applicable to a wide range of datasets, including those with complex and unknown distributions. This is particularly beneficial in TB prediction, where the data may exhibit diverse patterns and distributions.5) Interpretable Results: KNN provides interpretable results since predictions are based on the majority vote of the nearest neighbors. It allows healthcare professionals to understand the reasoning behind the prediction and potentially identify which features contribute the most to TB prediction. (6) Incremental Learning: KNN supports incremental learning, meaning that new data can be added to the existing model without retraining the entire algorithm. This capability is useful in TB prediction, as new cases and data become available over time, allowing the model to continuously improve and adapt to changing conditions. 7) Availability of Training Data: KNN performs well when the training dataset contains a sufficient number of representative cases, including both TB-positive and TB-negative instances. If a comprehensive and diverse dataset is available, KNN can effectively learn patterns and make accurate predictions. It's important to note that while KNN can be useful in TB prediction, the performance and accuracy of the model depend on various factors, including the quality and representativeness of the training data, the choice of appropriate features, and the determination of an optimal value for the hyperparameter K. Additionally, KNN should be used in conjunction with other diagnostic tools and medical expertise to make informed decisions about TB diagnosis and treatment.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

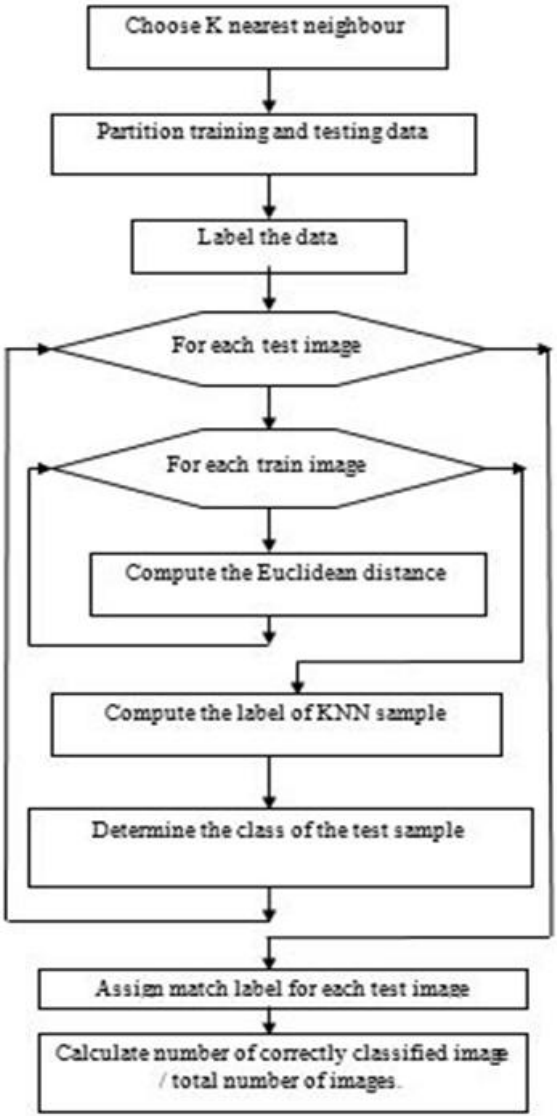

Figure 5: shows the flow of KNN

C. Preprocessing

Data preprocessing refers to the procedures we must follow to alter or encode data so that a machine can quickly and readily decode it. The algorithm's ability to quickly analyze the properties of the data is essential for a model to be accurate and exact in its predictions. Due to their heterogeneous origin, the bulk of real-world datasets used for machine learning are very likely to contain missing data, inconsistent results, and noise. Data mining methods would not produce high-quality results when applied to this noisy data because they would be unable to successfully find patterns. Therefore, data processing is crucial to raising the general level of data quality. Missing or duplicate values could present an inaccurate picture of the data's overall statistics. Inconsistent data points and outliers are often tending to disturb the model by affecting overall learning and hence leading to false predictions.

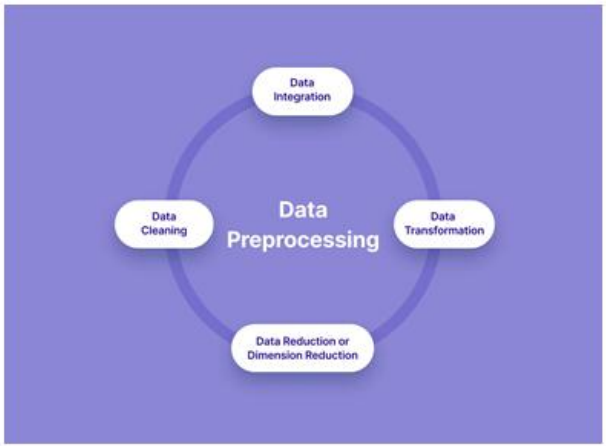

Figure 6: indicates the steps involved in data preprocessing

D. Dataset

The dataset includes features such as coughing blood, fever for two weeks, sputum mixed with blood, chest pain, night sweats, back pain, shortness of breath, weight loss, the body feeling tired, lumps that appear around the armpit and neck, cough and phlegm for continuously two weeks, swollen lymph nodes and loss of appetite.

IV. RESULTS AND DISCUSSION

The law you handed significant several libraries generally used for data analysis and visualization numpy is a library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of fine functions to operate on these arrays. panda is a library for data manipulation and analysis. It provides data structures like DataFrames, which allow you to organize and dissect data effectively. matplotlib is a conniving library for Python. It provides a variety of functions to produce different types of plots and visualizations. rcParams is a module within matplotlib that allows you to customize the parcels of the generated plots. rainbow is a color chart module within matplotlib. cm which provides a range of color schemes for conniving. matplotlib inline enables. It suppresses advising affairs, which can be useful in some cases, but it's generally recommended to handle warnings meetly. This line imports the KNeighborsClassifier class from the neighbor's module of Scikit-learn. The KNeighborsClassifier is the perpetration of the k- nearest neighbor's algorithm for the bracket. This line imports the DecisionTreeClassifier class from the tree module of Scikit- learn. The DecisionTreeClassifier is the perpetration of the decision tree algorithm for a bracket. The law you handed is trying to read a CSV train using the pandas library in Python. It seems that you are trying to read a train located at" C/ druggies/ nisar/ OneDrive/ Documents/ Tb complaint symptoms without id anddatetime.csv". () the function is used to get a terse summary of the data frame.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

It comes in really handy when doing exploratory analysis of the data. To get a quick overview of the dataset we use thedataframe.info() function. describe () is used to view some introductory statistical details like percentile, mean, std, etc. of a data frame or a series of numeric values. When this system is applied to a series of strings, it returns a different affair. Seaborn is a library for making statistical plates in Python. It builds on top of matplotlib and integrates nearly with pandas data structures. Seaborn helps you explore and understand your data. Its conniving functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce instructional plots. Its dataset-acquainted, declarative. The law you handed is using the seaborn library to plot a heatmap of the correlation matrix of a data frame called df. The heatmap visualizes the correlation between different features in the dataset. A heatmap is a graphical portrayal of information where values are portrayed involving colors in a two-layered framework. It is frequently used to imagine and investigate information examples, connections, and dispersions. In a heatmap, every cell in the framework relates to a particular worth or estimation. The shade of every cell addresses the extent of the worth, with various tones demonstrating various ranges or levels. Commonly, a variety scale or colormap is utilized to relegate tones to the qualities, going from low to high. Heatmaps are normally utilized in different fields, including information examination, measurements, AI, and perception. They can be utilized to investigate connections between factors, recognize bunches or examples in information, feature patterns or exceptions, and present complex data in an outwardly engaging and straightforward configuration. With regards to the model I gave before, a heatmap is utilized to envision the connection between TB sickness side effects and their seriousness levels. Every side effect is addressed by a line, every seriousness level is addressed by a section, and the count of events is addressed by the variety of forces in every cell. The heatmap assists with distinguishing which side effects are more pervasive and how they are appropriated across various seriousness levels. Using a heatmap to visualize a confusion matrix, time-series movements, temperature changes, correlation matrix, and SHAP interaction values. Heatmaps can bring your data to life. Versatile and eye-catching. There are many situations where they can highlight important relationships in your data.

Syntax: seaborn.Heatmap(data, *, vmin=None, vmax=None, cmap=None, center=None, annot_kws=None, linewidths=0, linecolor='white', cbar=True, **kwargs) Important Parameters: data: 2D dataset that can be coerced into an narray. vmin, vmax: Values to anchor the colormap, otherwise they are inferred from the data and other keyword arguments. cmap: The mapping from data values to color space. center: The value at which to center the colormap when plotting divergent data. annot: If True, write the data value in each cell. fmt: String formatting code to use when adding annotations. linewidths: Width of the lines that will divide each cell. Line color: Color of the lines that will divide each cell. cbar: Whether to draw a color bar. All the parameters except the data are optional. Returns: An object of type matplotlib.axes._subplots.AxesSubplot Correlation heatmaps can be used to find potential relationships between variables and to understand the strength of these relationships. In addition, correlation plots can be used to identify outliers and detect linear and nonlinear relationships. The color-coding of the cells makes it easy to identify relationships between variables at a glance. Correlation heatmaps can be used to find both linear and nonlinear relationships between variables. A correlation heatmap is a graphical representation of a correlation matrix representing the correlation between different variables. The value of correlation can take any value from -1 to 1. A correlation between two random variables or bivariate data does not necessarily imply a causal relationship. Correlation between two variables can also be determined using a scatter plot between these two variables.

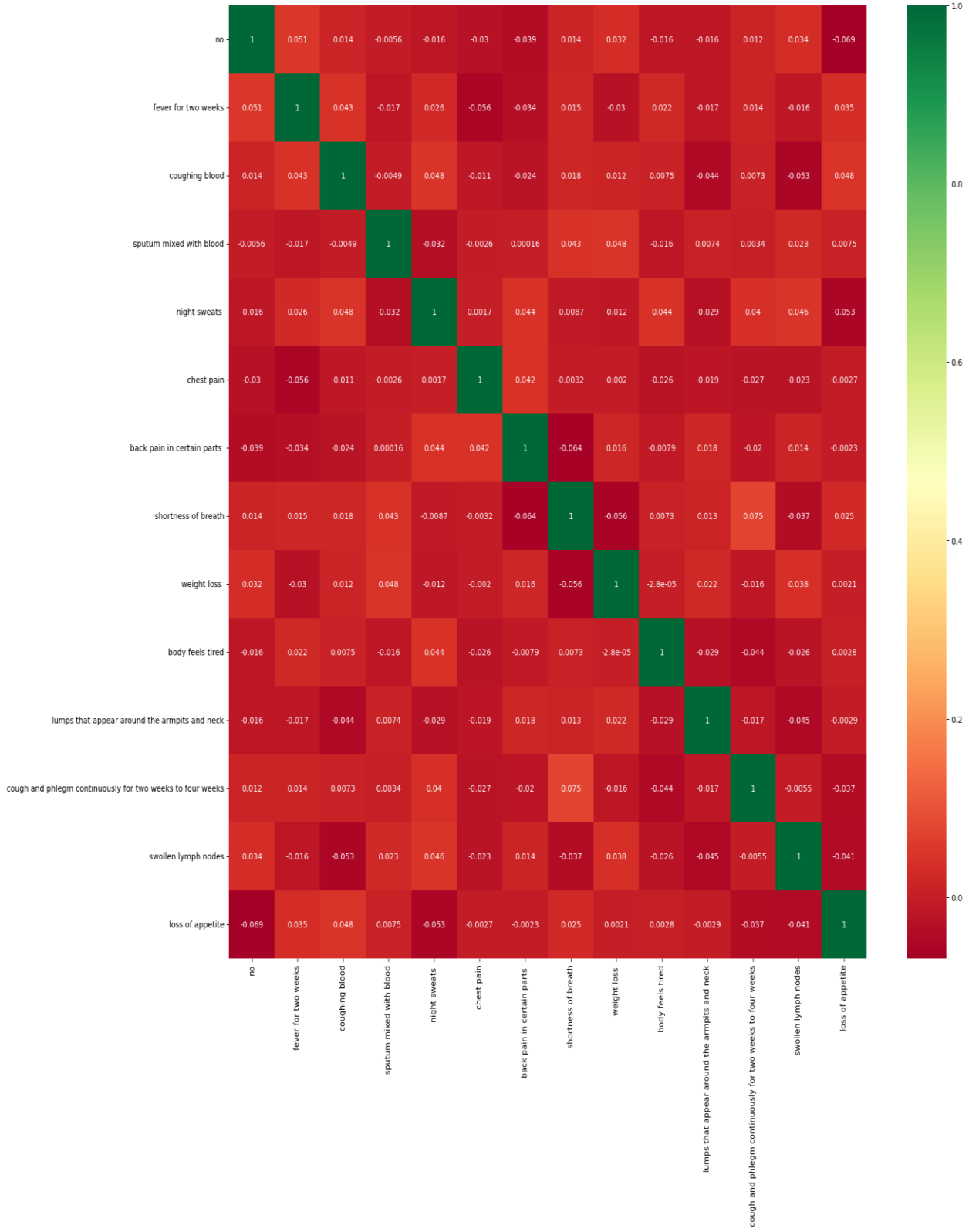

Figure 7: Correlation heatmap of all features

DataFrame.hist() function is useful in understanding the distribution of numeric variables. This function splits up the values into the numeric variables. Its main functionality is to make the Histogram of a given Data frame. The distribution of data is represented by Histogram. When Function Pandas DataFrame.hist() is used, it automatically calls the

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

function matplotlib.pyplot.hist() on each series in the DataFrame.

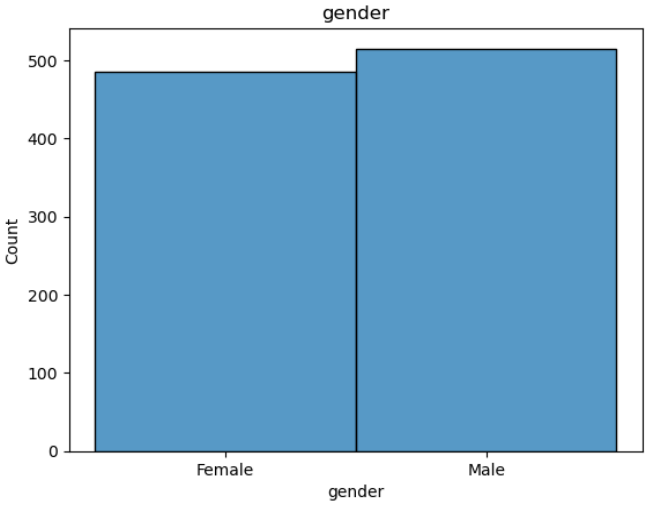

Figure 8: Number of males and females

Fig.8 tells us that the dataset contains different people having different symptoms. It includes that there are more than 500 males under which the dataset was created. Dataset includes less than 500 females.

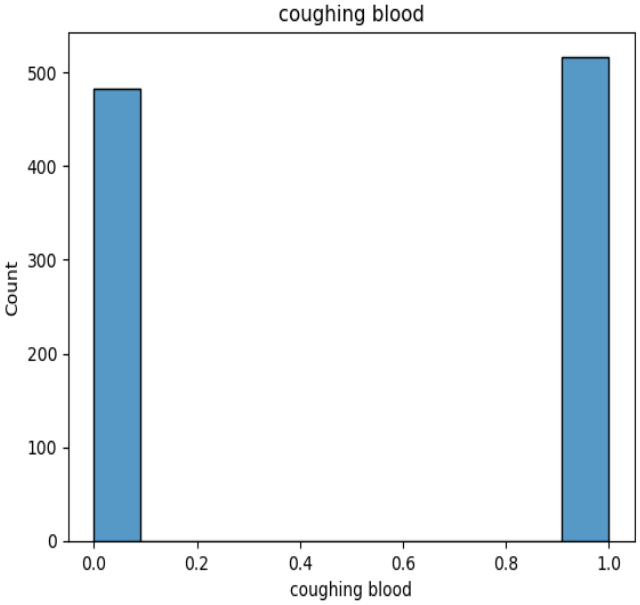

Figure 9: Coughing blood feature histogram

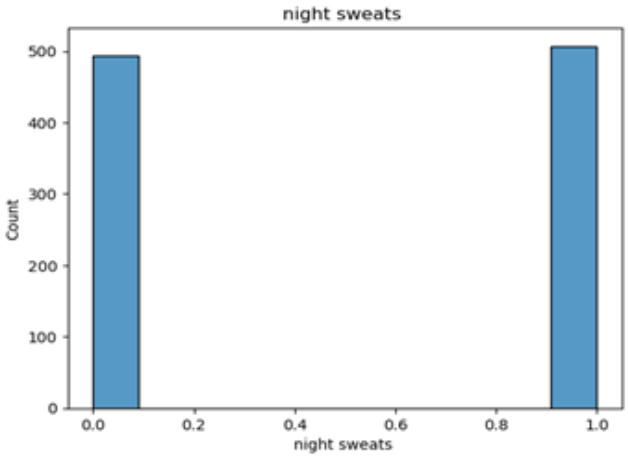

Figure 10: Night sweats feature bar graph

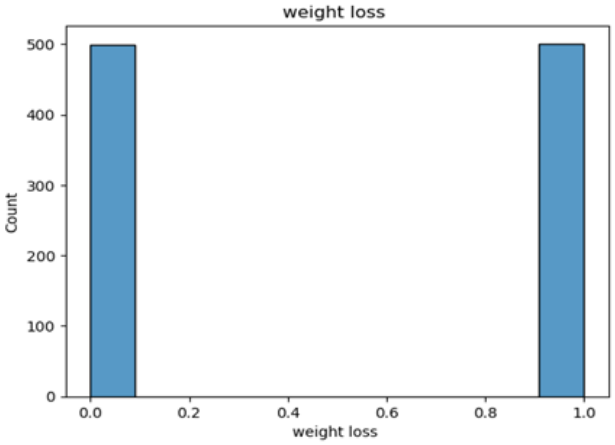

Figure 11: Weight loss feature histogram

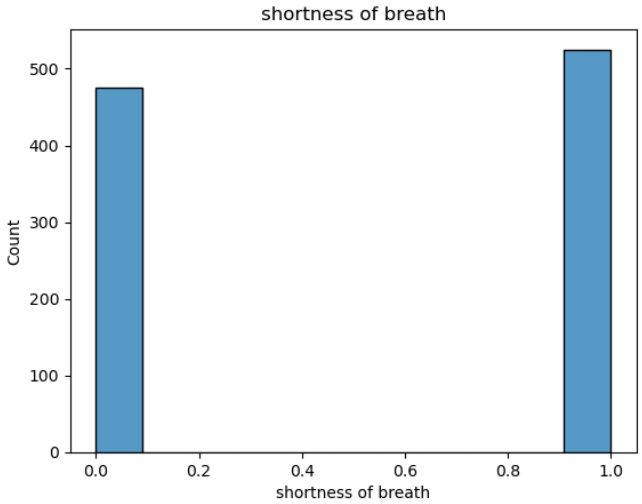

Figure 12: Shortness of breadth histogram

Fig.9, Fig.10, and Fig. 11 signify that the majority of the males and females in the dataset show symptoms of coughing blood. More males show symptoms of coughing blood than compared to females.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

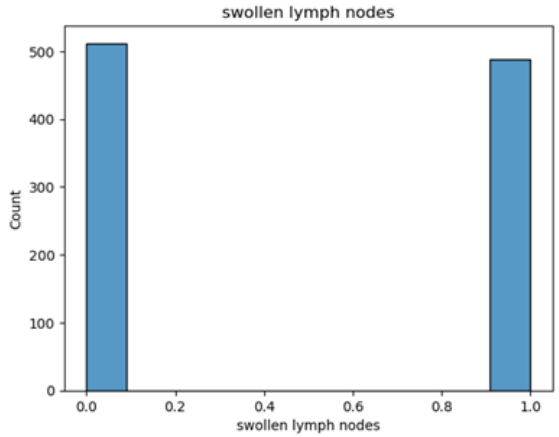

Figure 13: Swollen lymph nodes feature histogram

Fig.12 shows us that an equal number of males and females in the dataset have the symptom of weight loss. While Fig.13 signifies that more females have the symptom of swollen lymph nodes as compared to males. The code you provided is using the seaborn library to create a count plot based on the column 'loss of appetite' in the DataFrame df. It sets the style to 'white grid' and uses the color palette 'RdBu_r' for the plot.

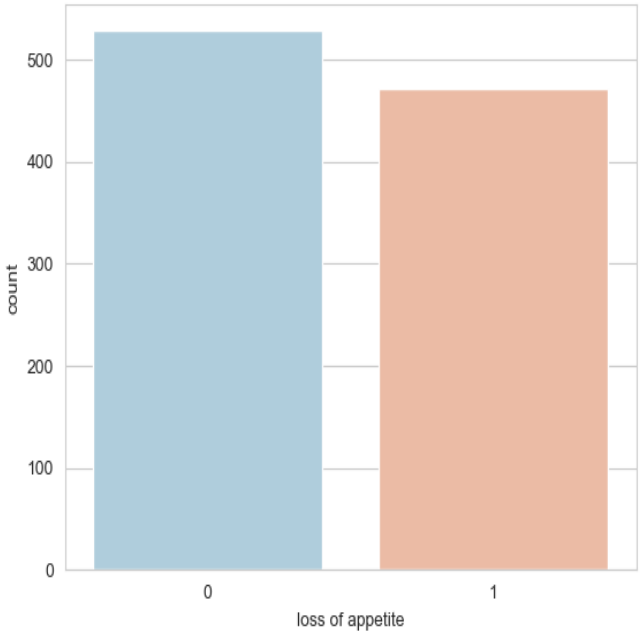

Figure 14: Loss of appetite bar graph

In Figure (14), 0 marks for males while 1 mark for females. The count is represented on the y-axis while loss of appetite is represented on the x-axis. It signifies that more males have the symptom of ‘loss of appetite’ than females.

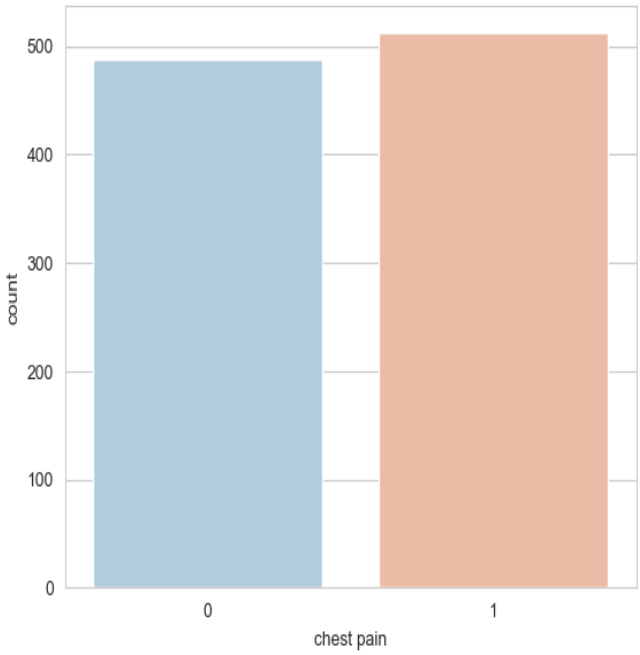

Figure 15: Chest pain bar graph

Fig.15, clearly shows that more females have the symptom ‘chest pain’ as compared to that of females. The pd.get_dummies() function from the pandas library to perform one-hot encoding on specific columns of the DataFrame df. The columns being one-hot encoded are 'gender', 'fever for two weeks', 'coughing blood', 'sputum mixed with blood', 'night sweats', 'chest pain', 'back pain in certain parts', and 'shortness of breath'. dataset.head(): For the first few rows of the dataset DataFrame after performing one-hot encoding, you can use the head() method. This will print the first few rows of the dataset DataFrame, showing the encoded columns and their corresponding values. Adjust the number of rows displayed by specifying the desired number inside the head() method, such as the head(10) to display the first 10 rows. df.dtypes: The data types of each column in the DataFrame df, you can use the dtypes attribute. This will print the data types of each column in the DataFrame df. The output will show the column names along with their corresponding data types. dataset[columns_to_scale]=dataset[columns_to_scale].replace('active_tb', np. nan): The code you provided is replacing the occurrences of the string 'active_tb' with np. nan (NumPy’s representation of missing or undefined values) in the specified columns of the dataset DataFrame. The columns_to_scale variable should contain a list of column names in the dataset DataFrame that you want to replace the string 'active_tb' with np. nan. The replace() method is used to perform the replacement operation. After executing this code, the occurrences of 'active_tb' in the specified columns will be replaced with np.nan. Splitting the dataset into training and testing sets:
The train_test_split function from a machine learning library is used to split a dataset into training and testing

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

subsets. X: This variable refers to the input features or independent variables of your dataset. y: This variable represents the target variable or dependent variable of your dataset. random_state: It is an optional parameter that sets the random seed for reproducibility. When the same random state is used, the data split will be the same across multiple runs of the program. The train_test_split function takes X and y as input and returns four sets of data: X_train: This variable contains a subset of the input features that will be used for training your machine learning model. X_test: This variable contains a subset of the input features that will be used for testing the trained model's performance. y_train: This variable contains the corresponding subset of target values or labels for the training data. y_test: This variable contains the corresponding subset of target values or labels for the testing data. By splitting the data into training and testing sets, you can train your model on the training data and evaluate its performance on the testing data, which helps in assessing how well the model generalizes to unseen data. Printing the training set data: The shape of the X_train and y_train arrays is printed. The shape of an array represents the dimensions or the number of rows and columns in the array. This line of code prints the shape of the X_train array. The format() function is used to insert the value of X_train.shape into the string. The X_train.shape attribute returns a tuple representing the shape of the array, where the first element of the tuple represents the number of rows and the second element represents the number of columns in the array. This line of code prints the shape of the y_train array in a similar way. It uses the format() function to insert the value of y_train. shape into the string. By printing the shapes of X_train and y_train, you can verify the dimensions of your training data, which can be useful for understanding the structure of your dataset and ensuring it aligns correctly with your machine-learning model. X_train.shape:(750, 23). y_train.shape:(750,). Printing test dataset: It prints the shapes of the X_test and y_test arrays, similarly to how the shapes of X_train and y_train were printed in the previous code snippet. The X_test.shape attribute returns the shape of the X_test array, where the first element represents the number of rows (samples) and the second element represents the number of columns (features) in the array. Similarly, the y_test.shape attribute returns the shape of the y_test array, which represents the shape of the target variable or labels corresponding to the X_test samples. By printing the shapes of X_test and y_test, you can verify the dimensions of your testing data, which is important for evaluating the performance of your trained machine learning model on unseen data. X_test.shape:(250, 23) y_test.shape:(250,) Later a range object neighbors_settings contains the values from 1 to 20 (inclusive). This range will be used to iterate over different values of the n_neighbors parameter in the k-nearest neighbors algorithm. In other words, it sets up a loop to try different values of n_neighbors from 1 to 20. The purpose of the code snippet you provided is likely to be part of a larger code block that trains and evaluates a k-nearest neighbors model with different values of n_neighbors. The accuracies obtained during training and testing will be stored in the training_accuracy and test_accuracy lists, respectively, for further analysis or visualization. Implementation of the KNN Algorithm: KNeighborsClassifier class from the sci-kit-learn library. This class provides the implementation of the KNN algorithm for classification tasks. A KNeighborsClassifier object is created with the current n_neighbors value using the n_neighbors parameter. This object is assigned to the variable clf. Then, the fit method is called on clf to train the KNN model. The score method of the KNeighborsClassifier object clf calculates the accuracies of the model on the training and testing data. The score method takes two arguments: the input features and the corresponding true labels. In this case, X_train.isnull() and y_train.isnull() are passed as arguments to calculate the accuracy of the training data, while X_test.isnull() and y_test.isnull() is used to calculate the accuracy of the testing data. The calculated accuracies are then appended to the training_accuracy and test_accuracy lists using the append method. This allows you to keep track of the model's performance on both the training and testing sets for each value of n_neighbors. By appending the accuracies to these lists, you can later analyze or visualize the performance of the KNN model across different values of n_neighbors on both the training and testing data.

The plot of the accuracy of the model:

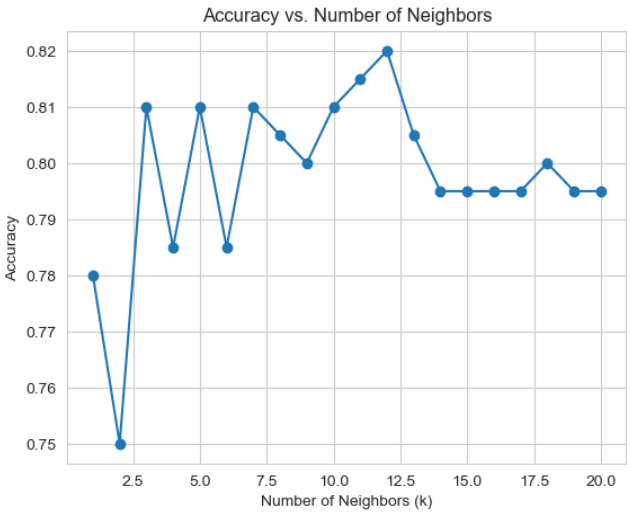

Figure 16: Plot of accuracy v/s number of neighbors

The plt.plot() function is used to create the plot, and it takes two arguments: the x-coordinates and the y-coordinates of the data points. In this case, the x-coordinates are generated using a list comprehension [k for k in range(1, 21)], representing the values 1 to 20 for K. The y-coordinates are represented by the knn_scores variable, which presumably contains the accuracy scores for each K value. The color='blue' argument sets the color of the line to blue.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

V. CONCLUSION AND FUTURE SCOPE

Overall, ML models offer a valuable complement to traditional methods in TB prediction. Continued research, collaborations, and advancements in ML techniques can contribute to more effective TB control and prevention strategies, ultimately leading to improved patient outcomes and global health impact. The above code provides a foundation for analyzing and predicting tuberculosis using K-NN classifiers. It includes data preprocessing, feature selection, model training, evaluation, and visualization techniques. Researchers and practitioners can gain insights into the dataset, understand feature importance, and build predictive models for rheumatoid arthritis. The future scope of this code includes enhancing feature engineering, exploring different machine learning algorithms, optimizing model hyperparameters, and implementing ensemble methods. Integration of biomarkers and genomic data can provide deeper insights and improve predictions. The code can be integrated into clinical decision support systems to assist healthcare professionals in making informed decisions. Real-time monitoring and prediction methods can enable early intervention and personalized treatment strategies. Further research can focus on patient stratification, personalized medicine, and developing interpretable models. The application of machine learning techniques to rheumatoid arthritis holds the potential for improving patient care, treatment outcomes, and our understanding of the disease. Continued advancements can lead to. The future scope of using machine learning (ML) models for tuberculosis (TB) prediction is promising and offers several areas of potential advancement: 1. Integration of Multi-Omics Data: ML models can be enhanced by integrating multi-omics data, including genomics, transcriptomics, proteomics, and metabolomics. By analyzing these diverse data sources, ML models can provide a comprehensive understanding of the host-pathogen interaction, identify biomarkers for TB diagnosis, predict treatment outcomes, and potentially discover novel therapeutic targets. 2. Utilization of Imaging Data: ML algorithms can leverage imaging techniques such as chest X-rays and computed tomography (CT) scans to aid in TB diagnosis and monitoring. ML models can learn to identify specific radiographic patterns associated with TB and contribute to the development of automated image analysis tools for early detection and monitoring of the disease. 3. Harnessing Wearable Devices and IoT: The integration of wearable devices, such as smartwatches and activity trackers, along with other Internet of Things (IoT) devices, can provide real-time monitoring of patient health parameters. ML models can utilize this data to track disease progression, monitor treatment adherence, and identify early signs of TB-related complications. 4. Application of Deep Learning: Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown great potential in various medical applications. These models can be employed to extract intricate patterns and features from large-scale TB datasets, leading to improved accuracy in predicting TB and enhancing personalized treatment strategies. 5. Integration of Social Determinants of Health: ML models can incorporate social determinants of health, such as socioeconomic status, access to healthcare, and living conditions, into TB prediction. By considering these factors, ML models can identify high-risk populations, target interventions, and help policymakers make informed decisions to mitigate the TB burden. 6. Explainable AI and Interpretability: Efforts to enhance the interpretability and transparency of ML models are crucial. The development of explainable AI techniques can help healthcare professionals understand how ML models arrive at their predictions, fostering trust and facilitating the adoption of ML-driven TB prediction tools in clinical practice. 7. Global Collaboration and Data Sharing: Collaborative initiatives and data-sharing platforms can facilitate the development of robust ML models for TB prediction. By pooling diverse datasets from different regions, ML models can be trained on more representative and comprehensive data, enabling better generalization and addressing geographical variations in TB characteristics. It is important to note that the successful implementation of ML models in TB prediction requires addressing challenges related to data quality, privacy, and bias. Additionally, clinical validation and integration into healthcare systems are crucial steps to ensure the practical and effective use of ML for TB control and management. By leveraging the potential of ML and addressing these challenges, we can improve the accuracy, efficiency, and impact of TB prediction, ultimately leading to better patient outcomes and a reduction in the global burden of TB.

REFERENCES

[1] Budiarto, Rahmat, Hilwa Lelisa & Yaya Sudarya Triana. (2022). Analysis of tuberculosis disease spreading pattern in muara enim district using knn algorithm. *International Conference on Engineering and Information Technology for Sustainable Industry*.

[2] Afzali, Ali, Farshid Babapour Mofrad & Majid Pouladian. (2019). Feature selection for contour-based tuberculosis detection from chest x-ray images. *26th National and 4th International Iranian Conference on Biomedical Engineering (ICBME)*. IEEE.

[3] Jamal, Salma, et al. (2020). Artificial intelligence and machine learning based prediction of resistant and susceptible mutations in mycobacterium tuberculosis. *Scientific Reports*, *10*(1), 5487.

Peer Reviewed & Refereed Journal
Volume-13, Issue-4 (August 2023)
https://doi.org/10.31033/ijemr.13.4.7

[4] Muhathir, Muhathir, Theofil Tri Saputra Sibarani & Al-Khowarizmi Al-Khowarizmi. (2020). Analysis K-Nearest Neighbors (KNN) in identifying Tuberculosis bisease (Tb) by utilizing hog feature extraction. *Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal*, *1*(1).

[5] Hrizi, Olfa, et al. (2022). Tuberculosis disease diagnosis based on an optimized machine learning model. *Journal of Healthcare Engineering*.

[6] Antony, Betsy & K. NB. (2017). Lung tuberculosis detection using x-ray images. *International Journal of Applied Engineering Research*, *12*(24), 15196-15201.

[7] Rajakumar, M. P., et al. (2021). Tuberculosis detection in chest X-ray using Mayfly-algorithm optimized dual-deep-learning features. *Journal of X-Ray Science and Technology*, *29*(6), 961-974.

[8] Ali, Mian Haider, et al. (2021). Prediction of multidrug-resistant tuberculosis using machine learning algorithms in swat, Pakistan. *Journal of Healthcare Engineering*.

[9] Tesfaye, Fregenet, et al. (2020). Alternative biomarkers for classification of latent tuberculosis infection status in pregnant women with borderline Quantiferon plus results. *Tuberculosis*, *124*, 101984.

[10] Muhathir, Muhathir, Theofil Tri Saputra Sibarani & Al-Khowarizmi Al-Khowarizmi. (2020). Analysis K-Nearest Neighbors (KNN) in identifying Tuberculosis disease (Tb) by utilizing hog feature extraction. *Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal*, *1*(1).

[11] AC, Ramachandra & Venkata Siva Reddy. (2022). *Bidirectional dc-dc converter circuits and smart control algorithms: a review*.

[12] Kumari, Ashwini, et al. (2018). Multilevel home security system using arduino & gsm. *Journal for Research*, *4*.

[13] Viswanatha, V., et al. (2020). Intelligent line follower robot using MSP430G2ET for industrial applications. *Helix-The Scientific Explorer*, *10*(02), 232-237.

[14] Viswanatha, V. & R. Reddy. (2020). Characterization of analog and digital control loops for bidirectional buck–boost converter using PID/PIDN algorithms. *Journal of Electrical Systems and Information Technology*, *7*(1), 1-25.

[15] Viswanatha, V., R. K. Chandana & A. C. Ramachandra. (2022). *IoT based smart mirror using raspberry pi 4 and yolo algorithm: A novel framework for interactive display*.