SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

Symbiosis School for Online and Digital Learning

A
DISSERTATION REPORT
ON

"LUNG CANCER PREDICTION USING MACHINE LEARNING"

SUBMITTED

BY

Ayush Sanjiv Sanghavi (23039141504)

M.Sc. (Computer Application)

Semester – II

July – 2023 Batch

DECLARATION

I hereby declare that the Dissertation Work entitled "**Lung Cancer Prediction Using Machine Learning**" submitted for the M.Sc. (Computer Application) Semester – II degree is my original work and the Dissertation work has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Students

Place: Mumbai, Maharashtra, India

Date: 16/08/2024

ACKNOWLEDGEMENT

It gives us immense pleasure to present this report on **"Dissertation Title"**. The dissertation work has brought out significance of sincere efforts, teamwork, guidance and support that makes a dissertation work successful. I take this opportunity to acknowledge the guidance and encouragement of all those with whom I have interacted during the course of this Dissertation.

I would like to thanks to our Director of the Symbiosis School for Online and Digital Learning **Dr. Parimala Veluvali** for their support and encouragement. I would like to thanks to my project guide **Dr. Niket Tajne** for valuable suggestions during the Dissertation work.

Ayush Sanjiv Sanghavi (23039141504)

Date: 16/08/2024

CERTIFICATE

This is to certify that the Dissertation titled "**Lung Cancer Prediction Using Machine Learning**" is the bona fide work carried out by **AYUSH SANJIV SANGHAVI (23039141504)** a student of M.Sc. Computer Application Semester - II of Symbiosis School for Online and Digital Learning, Pune, India, affiliated to Symbiosis International (Deemed University) during the academic year 2023-24, in partial fulfillment of the requirements for the award of the degree of M.Sc. Computer Application.

Signature of the Guide

Place: Mumbai, Maharashtra, India
Date: 16/08/2024

INDEX

Chapter 1

Introduction

1.1 OVERVIEW

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Machine learning, as a powerful approach to achieve Artificial Intelligence, has been widely used in pattern recognition, a very basic skill for humans but a challenge for machines. Nowadays, with the development of computer technology, pattern recognition has become an essential and important technique in the field of Artificial Intelligence. The pattern recognition can identify letters, images, voice or other objects and also can identify status, extent or other abstractions.

Since the computer was invented, it has begun to affect our daily life. It improves the quality of our lives; it makes our life more convenient and more efficient. A fascinating idea is to let a computer think and learn as a human. Basically, machine learning is to let a computer develop learning skills by itself with given knowledge. Pattern recognition can be treated like computer being able to recognize different species of objects. Therefore, machine learning has close connection with pattern recognition.

Machine Learning is scientific research of statistical procedures and methods which they are used by computer systems designed to perform such functions without specific instructions, rather than trusting in the models and conclusions. This is believed to be part of an artificial intelligence. Machine Learning algorithms sets up a mathematical model based on data examples called "training data" to make predictions without the completion of a task being explicitly programmed.

2 1.2 PURPOSE OF THE MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Nowadays, with the development of computer technology, pattern recognition has become an essential and important technique in the field of Artificial Intelligence. The pattern recognition can identify letters, images, voice or other objects and also can identify status, extent or other abstractions.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.3 PROBLEM STATEMENT

With the rapid increase in population rate, the rate of diseases like cancer, chikungunya, cholera etc., are also increasing. Among all of them, cancer is becoming a common cause of death. Cancer can start almost anywhere in the human body, which is made up of trillions of cells. Normally, human cells grow and divide to form new cells as the body needs them. When cells grow older or become damaged, they die, and new cells take their place. When cancer cells develop, however, this orderly process breaks down. As cells become more and more abnormal, old or damaged cells survive when they should die, and new cells form when they are not needed. These extra cells can divide without stopping and may form growths called tumor. This tumor starts spreading to different of body. Tumors are of two types benign and malignant where benign (noncancerous) is the mass of cell which lack in ability to spread to other part of the body and malignant (cancerous) is the growth of cell which has ability to spread in other part 3 of body this spreading of infection is called metastasis. There is various type of cancer like Lung cancer, leukemia, and colon cancer etc. The incidence of lung cancer has significantly increased from the early 19th century. There is various cause of lung cancer like smoking, exposure to radon gas, secondhand smoking, and exposure to asbestos etc.

1.4 OBJECTIVE

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow

CHAPTER 2

Literature review, Objective, Scope

2.1 LITERATURE REVIEW

In the 21st century, cancer is still considered a serious disease as the mortality rates are high. Among all cancer types, lung cancer ranks first regarding morbidity and mortality [1, 2]. There are two main categories of lung cancer: non-small-cell lung cancer (NSCLC) and small cell lung cancer (SCLC). For non-small-cell lung cancer, a subcategorization into lung squamous cell carcinoma (LUSC) and lung adenocarcinoma (LUAD) is further used. These types of cancers account for approximately 85% of lung cancer cases [3]. Compared with the diagnosis of benign and malignant, further fine-grained classification of lung cancers such as LUSC, LUAD, and SCLC is of great significance for the prognosis of lung cancer. Accurately determining the category of lung cancer in the early diagnosis directly influences the effect of the treatment and thus the patients' survival rate [1, 4]. Positron emission tomography (PET) and computed tomography (CT) are both widely used noninvasive diagnostic imaging techniques for clinical diagnosis in general and for the diagnosis of lung cancer in particular [4]. Immunohistochemical evaluation is considered the gold standard for lung cancer classification. However, this procedure requires a tissue biopsy, an invasive procedure with the inherent risk of a delayed diagnosis and thus exacerbation of the patient's pain. Advances in artificial intelligence research enabled numerous studies on the automatic diagnosis of lung cancer. The use of data in lung cancer-type classification is roughly divided into three categories: CT and PET image data as well as pathological images [5]. The well-known data science community Kaggle provides high-quality CT images for participants with the task to distinguish malignant or benign nodules from pulmonary nodules. Kaggle competitions repeatedly produce excellent deep learning approaches for these tasks [6, 7]. With the progresses in the research of automatic lung cancer diagnosis, studies are no longer limited to the classification of benign and malignant nodules and data sets are no longer limited to CT images [8–12]. Wu et al. [9] use quantitative imaging characteristics such as statistical, histogram-related, morphological, and textural features from PET images to predict the distance metastasis of NSCLC, which shows that quantitative features based on PET images can effectively characterize intratumor heterogeneity and complexity. Two recent publications propose the application of deep learning to pathological images to classify NSCLC and SCLC [10] and to classify transcriptome subtypes of LUAD [11]. The complexity of the clinical diagnosis of lung cancer is also characterized by the wide range of imaging modality, which is employed in the diagnosis [13, 14]. 5 Previous research already proved that deep learning approaches can not only use the feature distribution patterns from different pulmonary imaging modalities but even merging different features to achieve the computer-aided diagnosis. Liang et al. [15] employ multichannel techniques to predict the IDH genotype from PET/CT data using a convolutional neural network (CNN), while other approaches use a parallel CNN architecture to extract several features of different imaging modalities [16, 17]. Compared with the classification of the benign and malignant, the classification of the three types of lung cancer from medical images are more suitable to constitute a fine-grained image recognition problem as diverse distributions of features and potential pathological features need to be considered. Because the fine-grained features which need to extract in images, and meanwhile the lesion region is a small part of the whole image, the deep learning framework is susceptible to feature noise. At present, most methods based on various deep learning frameworks have proved to have certain bottleneck in fine-grained problems. In order to solve this problem, the previous research mainly implements the attention mechanism from the two dimensions (channel and

spatial) of the feature representation. The channel attention mechanism models the relationship between feature channels [18], while the spatial attention mechanism ensures that noise is suppressed by weighting feature representation spatially [19–21]. So far, spatial attention mechanism has been used in medical image processing to enhance extracted features [20, 21]. The channel attention mechanism has been used in the detection and classification of pulmonary disease [22, 23]. The presentation of these attention mechanisms illustrates the source of characteristic noise from different perspectives. There are few related studies on how to use the attention mechanism more effectively on images with different imaging modalities, so the deep learning model based on the multimodality dataset still has problems in fine-grained problems. Many works has already been proposed for prediction of cancer by various researchers among then Palani et al., [5] has proposed IoT based predictive modeling by using fuzzy C mean clustering for segmentation and incremental classification algorithm using association rule mining and decision tree for classification for classifying the tumor sets and based on the output generated by incremental classification model convolutional neural network has been applied with other features for predicting benign or malignant. Lynch et al., [6] Various machine learning algorithm are implemented for predicting the survivability rate of person, performance is measured based on root mean square error. Each model is trained using 10-fold cross validation, as the parameters are preprocessed by assigning default value so cross 6 validation is used for avoiding over fitting. FENWA et al., [3] proposed a model whether feature like contrast, brightness from the image dataset is extracted using texture-based feature extraction and on that two type of machine learning algorithm are applied one is artificial neural network another one is support vector machine and then performance has been evaluated on both the algorithm to compare which algorithm is giving more accuracy. Öztürk et al., [7] proposed a model where five types of feature extraction techniques were used in individual classification algorithm to predict at which features extraction technique which machine learning algorithm is giving more accuracy. Jin et al., [8] proposed a model where the original image is first converted into binary image the erosion and dilution has been operated on that image after that image has been segmented on the segmented image region of interest extraction is applied to identify volume or size of the tumor and after extraction convolutional neural network is applied with softmax classification layer to recognize the tumor is cancerous or not. Sumathipala et al., [9] proposed a model where the image data are taken from LIDC-IDRI, after collecting the image data image filtration has been implemented, filtration is done based on the patient who went through biopsy and module level is equal to 30 and then images whose module level is equal to 30 is segmented and then Logistic regression and random forest has been applied for prediction.

2.2 OBJECTIVE

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow

2.3 SCOPE

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. Nowadays, with the development of computer technology, pattern recognition has become an essential and important technique in the field of Artificial Intelligence. The pattern recognition can identify letters, images, voice or other objects and also can identify status, extent or other abstractions. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

CHAPTER 3

Research Methodology

3.1. INTRODUCTION

This research aims to develop a robust framework for early-stage lung cancer prediction by leveraging Digital Image Processing (DIP) and Machine Learning techniques. The objective is to accurately classify lung tumors as benign or malignant, enabling timely medical intervention and potentially improving patient outcomes. Lung cancer remains one of the leading causes of cancer-related mortality worldwide. Therefore, early and accurate prediction of malignancy can significantly reduce mortality rates by allowing clinicians to initiate treatment at the earliest possible stage.

3.2. DATA COLLECTION

The data used in this research will come from two primary sources. CT scan images of the lungs, obtained from reputable sources such as the LIDC-IDRI (Lung Image Database Consortium and Image Database Resource Initiative) or hospital archives, will be used. These images are pre-labeled as benign or malignant based on histopathological reports. In addition to the images, relevant patient demographics and clinical history, such as age, gender, smoking history, and family history of cancer, will also be collected. All data collection will adhere to strict ethical guidelines, ensuring patient confidentiality and compliance with regulations like HIPAA or GDPR.

3.3. DIGITAL IMAGE PROCESSING (DIP)

The DIP process begins with image preprocessing, where advanced filters like Gaussian or median filters will be applied to minimize noise and enhance image clarity. Contrast enhancement techniques, such as histogram equalization, will be employed to improve the visibility of lung structures and tumors, and image intensity values will be normalized to a standard range to ensure consistency across the dataset. Following preprocessing, image segmentation will be conducted using state-of-the-art algorithms like watershed segmentation, level set methods, or U-Net to accurately delineate tumor boundaries. The accuracy of segmentation will be validated against expert radiologist annotations to ensure reliability.

3.4. FEATURE EXTRATION

Feature extraction will focus on three main types: morphological features related to the size, shape, and edge sharpness of the tumor; texture features using methods like Gray Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP) to capture texture information critical in differentiating benign from malignant tumors; and positional features that analyze the tumor's location within the lung to account for the spatial distribution of different cancer

types. To reduce the feature set and minimize computational complexity, dimensionality reduction techniques such as Principal Component Analysis (PCA) or t-SNE (t-Distributed Stochastic Neighbor Embedding) will be used.

3.5. MACHINE LEARNING MODEL DEVELOPMENT

For the machine learning model development, three algorithms have been selected: Support Vector Machines (SVM) for its ability to handle high-dimensional feature spaces and create optimal decision boundaries; Random Forest for its robustness and ability to handle non-linear relationships through ensemble learning; and Artificial Neural Network (ANN) for its capacity to model complex, non-linear relationships between input features and the target variable. The model will be trained on a stratified split of the dataset, typically 70%, ensuring a balanced representation of benign and malignant cases. To assess model generalizability and avoid overfitting, 10-fold cross-validation will be implemented. Hyperparameter tuning will be performed using methods like Grid Search or Bayesian Optimization to fine-tune model parameters for optimal performance.

3.6. MODEL EVALUTION

The evaluation of the model's performance will focus on key metrics such as accuracy, recall (sensitivity), precision, and F1 score, which combine precision and recall into a single metric to balance the trade-off between them. A confusion matrix will be used to visualize the model's performance in distinguishing between benign and malignant cases. A comparative analysis of the SVM, Random Forest, and ANN models will be conducted to determine the most effective algorithm for lung cancer prediction.

3.7. RESULT INTERPERATION

The final step in the research will involve interpreting the results, including analyzing the contribution of each feature to the model’s decision-making process using techniques like SHAP (SHapley Additive exPlanations) to enhance model interpretability. The model's robustness will be tested on an independent test set and, if available, external datasets to evaluate its generalizability. Performance will be visualized using ROC curves, precision-recall curves, and confusion matrices to provide a clear understanding of the model’s strengths and weaknesses.

CHAPTER 4

Data Analysis and Findings/ Need of Software

4.1 DRAWBACKS OF EXISTING SYSTEMS:

In some cases, the application still does not have accurate results. Further optimization is needed. Priority information is needed for segmentation. Database extension is required for greater accuracy. Only a few diseases are covered. Therefore, the work must be expanded to cover more diseases. Possible causes that can cause misclassification can be: Symptoms of the disease vary from cigarettes, optimizing the characteristics needed, more training patterns are needed to cover and predict more cases - the actual disease.

4.2 PROPOSED SYSTEM:

The main theme of this project, is to detect the lung cancer and to take precautions to avoid or clear that diseases. It overcomes the drawbacks of existing system. The implementation phase begins with smoking as input and do the following steps: ¬ Pre-Processing layer. ¬ Segmentation layer. ¬ Feature Extraction of layer. ¬ Machine learning classifier. After performing all the above steps, we can detect the disease of lung cancer.

4.3 SYSTEM REQUIREMENTS

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

4.4 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

1. Intle® Core™ I3 11th Gen Processor
2. 18 GB RAM
3. 1 TB Hard Disk
4. SSD
5. 4 GB Nvidia GPU
6. Monitor

7. Web camera

4.5 SOFTWARE REQUIREMENTS

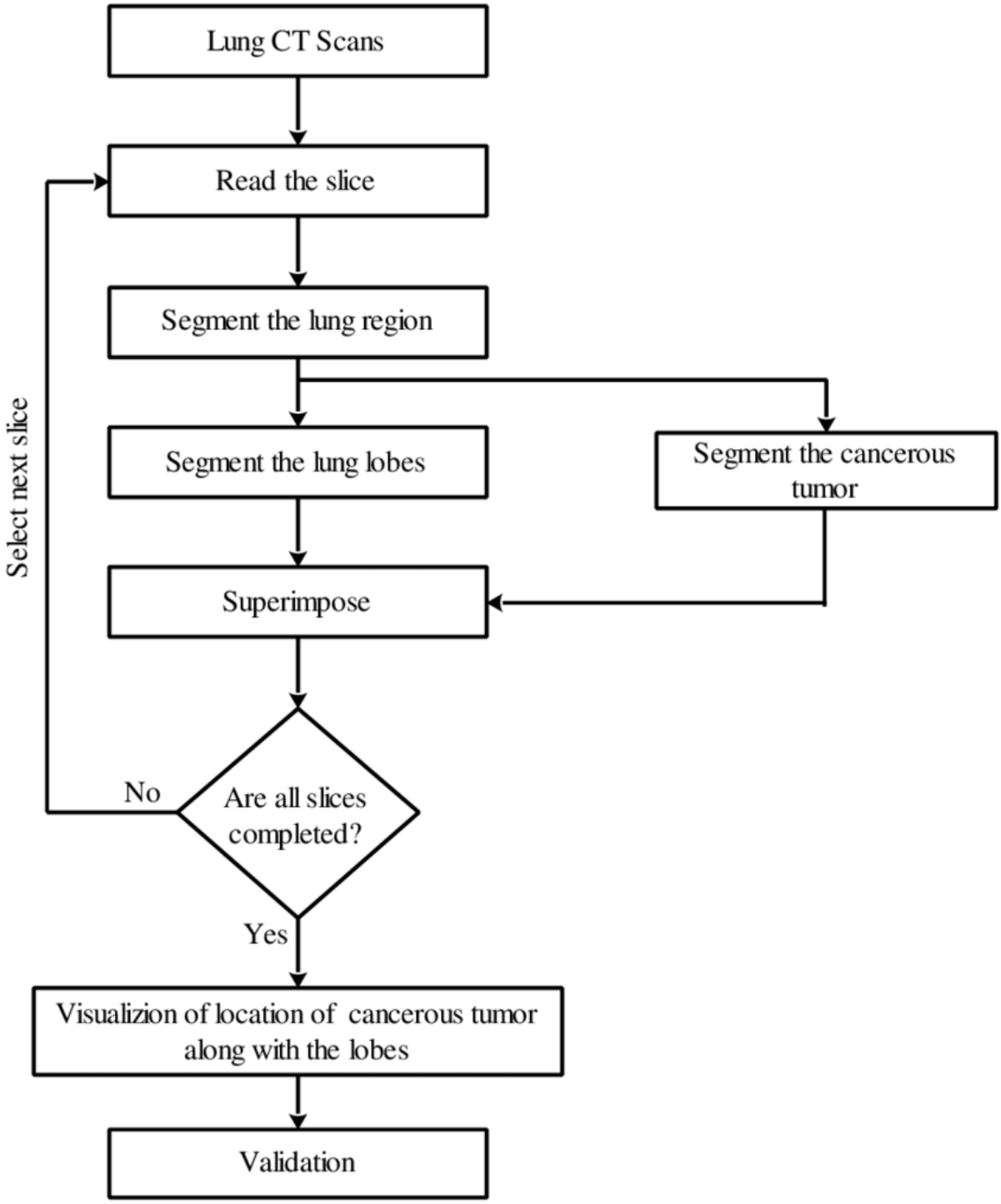
The software requirements are the specification of the system. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

1. Windows 11
2. Web Browser
3. Python Package Manager
4. IDE
5. Jupyter Notebook

4.6 PROPOSED WORK

Based on the literature survey a novel model has been proposed which consist of preprocessing block, segmentation block, feature extraction block and then classification block. In prediction of cancer CT scan report is basically used. But CT scan report is full of noise which cannot be seen by human eye for that reason various digital image processing plays a important role to get a noise free image. Digital image processing is the process where the analysis and manipulation of image is used to extract some useful information from the image. Digital image processing involve various step like image pre-processing where we can enhance the image using histogram equalization, spatial filter etc. Then image restoration can be done where various kind of noise like salt and pepper noise, Gaussian noise etc are applied and filter like median filter, mean filter can be applied on the pre-processed image. After that color conversions is applied only if the image is colored image then convert it to gray level. Fig. shows the proposed novel framework. Image segmentation is a process which divides the image into several segment based on the pixel, once the image segmentation is over the feature extraction can be applied. Feature extraction is a type of dimensionality reduction where a set of raw data is reduced to more manageable group image data for extracting the feature like region and texture. After extracting the feature different machine learning technique is used to classify the image.

4.6.1 UML DIAGRAM



Fg:1

CHAPTER 5

Findings, Suggestions and Conclusion

5.1 FINDINGS

A web application has been created using Python, where users can input data for predicting whether they have cancer or not. The machine learning model used in this application is logistic regression, which has demonstrated the highest accuracy. Based on the input data provided by the user, the model will predict the likelihood of the person having cancer and assign a severity scale of either, which displays various user-inputted parameters such as gender, age, smoking habits, yellow fingers, anxiety, peer pressure, chronic disease, fatigue, allergy, wheezing, alcohol consuming, coughing, shortness of breath, swallowing difficulty, chest pain, and lung cancer. Once the user fills in these parameters, they can click on the Predict button, which transfers the data to the backend. Here, our chosen machine learning algorithm, Logistic Regression, Confusion Matrix, KNeighbors Classifier, DecisionTree Classifier, Bagging Classifier, OneVsRest Classifier, GaussianNB, RandomForest Classifier, Correlation, processes the input data to get every variety of prediction and on that basis we can get the diagnostic of lung cancer. Based on the analysis, the output will indicate whether the patient is diagnosed with lung cancer or not.

5.2 SUGGESTIONS

After reviewing the description, several areas can be improved for clarity and conciseness. The explanation of the "severity scale" needs more detail. While it's mentioned that the model predicts the likelihood of cancer and assigns a severity scale, the scale itself isn't clearly defined. It would be helpful to specify what the scale indicates—whether it's a range (e.g., low, medium, high) or a numerical score—and explain its significance in the context of the prediction. The current description lists several machine learning models and metrics (like Logistic Regression, Confusion Matrix, KNeighbors Classifier, etc.), which could be confusing to readers. It might be more effective to streamline this by focusing on the primary model, Logistic Regression, and then briefly mentioning that other models are used for validation or comparison. This would make the text more approachable while still conveying the necessary technical information.

Consistency in terminology is important, particularly regarding "cancer" and "lung cancer." The text alternates between these terms, which might create some confusion. Sticking to "lung cancer" throughout the description would make the content clearer and more specific. The flow of the explanation could also be improved. Currently, the text jumps between describing the input parameters and discussing the machine learning models. A better structure might involve first outlining how users input their data and then explaining how the machine learning model processes this information to produce a prediction. This would create a more logical sequence that is easier to follow.

Additionally, the description mentions that the data is transferred to the backend without explaining what happens there. It would be helpful to include a brief overview of the backend process, highlighting how the various models and algorithms work together to analyze the input data and generate a prediction. There is some redundancy in the list of classifiers, which could be streamlined to focus on the key models central to the prediction process. This would make the text more concise and focused on the most important aspects of the application. Lastly, it might be beneficial to include a note about the user interface, specifically how the application presents results to the user. Mentioning that the interface is user-friendly or explaining how the results are displayed (e.g., as a simple yes/no diagnosis or a probability score) would give readers a better understanding of the user experience.

5.3 CONCLUSION

In summary, the development of a web application utilizing Logistic Regression for the prediction of lung cancer represents a significant step towards leveraging machine learning for early diagnosis. The application effectively allows users to input a range of personal and health-related data, which the model then processes to predict the likelihood of lung cancer and provides an indication of severity. This tool not only offers a user-friendly interface but also demonstrates high accuracy in its predictions, making it a valuable resource for both patients and healthcare providers.

By focusing on Logistic Regression as the primary model, the application achieves a balance between simplicity and accuracy, ensuring that users receive reliable predictions. The use of additional models for validation further enhances the robustness of the diagnostic process. Moreover, the clear and structured presentation of results allows users to easily understand their health status, potentially prompting timely medical consultations.

While the application shows promise, future enhancements could include the integration of more advanced machine learning models, such as deep learning techniques, to further improve predictive accuracy. Additionally, expanding the input parameters to include more comprehensive medical data could provide a more nuanced analysis, leading to even more precise predictions. Overall, this project underscores the potential of machine learning in healthcare, paving the way for more personalized and proactive approaches to lung cancer diagnosis.

CHAPTER 6

References

6.1 BOOKS

[1] Krishnaiah, V., G. Narsimha, and Dr N. Subhash Chandra. "Diagnosis of lung cancer prediction system using data mining classification techniques." International Journal of Computer Science and Information Technologies 4.1 (2013): 39-45.

[2] Zhang, Junjie, et al. "Pulmonary nodule detection in medical images: a survey." Biomedical Signal Processing and Control 43 (2018): 138- 147.

[3] Fenwa, Olusayo D., Funmilola A. Ajala, and A. Adigun. "Classification of cancer of the lungs using SVM and ANN." Int. J. Comput. Technol. 15.1 (2016): 6418-6426.

[4] Daoud, Maisa, and Michael Mayo. "A survey of neural network-based cancer prediction models from microarray data." Artificial intelligence in medicine (2019).

[5] Palani, D., and K. Venkatalakshmi. "An IoT based predictive modelling for predicting lung cancer using fuzzy cluster-based segmentation and classification." Journal of medical systems 43.2 (2019): 21.

6.2 WEBSITES

[1] PubMed - https://pubmed.ncbi.nlm.nih.gov/

[2] IEEE Xplore Digital Library - https://ieeexplore.ieee.org/

[3] Google Schola - https://scholar.google.com/

[4] National Cancer Institute (NCI) - https://www.cancer.gov/

[5] Kaggle - https://www.kaggle.com/

[6] ResearchGate - https://www.researchgate.net/

[7] The Lancet - https://www.thelancet.com/

[8] ScienceDirect - https://www.sciencedirect.com/

[9] SpringerLink - https://link.springer.com/

[10] PLOS ON - https://journals.plos.org/plosone/

CHAPTER 7

APPENDIX

7.1 SOURCE CODE

```
import pandas as pd
import matplotlib. pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

lung_data = pd.read_csv("survey lung cancer.csv")
lung_data
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHOR OF B
0	M	69	1	2	2	1	1	2	1	2	2	2	
1	M	74	2	1	1	1	2	2	2	1	1	1	
2	F	59	1	1	1	2	1	2	1	2	1	2	
3	M	63	2	2	2	1	1	1	1	1	2	1	
4	F	63	1	2	1	1	1	1	1	2	1	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
304	F	56	1	1	1	2	2	2	1	1	2	2	
305	M	70	2	1	1	1	1	2	2	2	2	2	
306	M	58	2	1	1	1	1	1	2	2	2	2	
307	M	67	2	1	2	1	1	2	2	1	2	2	
308	M	62	1	1	1	2	1	2	2	2	2	1	

309 rows × 16 columns

```
lung_data.head()
lung_data.tail()

#dependent_variable
x = lung_data.iloc[:,0:-1]
print(x)
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	\
0	M	69	1	2	2	1	
1	M	74	2	1	1	1	
2	F	59	1	1	1	2	
3	M	63	2	2	2	1	
4	F	63	1	2	1	1	
..	...	...	...	...	...	...	
304	F	56	1	1	1	2	
305	M	70	2	1	1	1	
306	M	58	2	1	1	1	
307	M	67	2	1	2	1	
308	M	62	1	1	1	2	

	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	\
0	1	2	1	2	2	2	
1	2	2	2	1	1	1	
2	1	2	1	2	1	2	
3	1	1	1	1	2	1	
4	1	1	1	2	1	2	
..	...	...	...	...	...	...	
304	2	2	1	1	2	2	
305	1	2	2	2	2	2	
306	1	1	2	2	2	2	
307	1	2	2	1	2	2	
308	1	2	2	2	2	1	

	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
0	2	2	2
1	2	2	2
2	2	1	2
3	1	2	2
4	2	1	1
..	...	...	...
304	2	2	1
305	2	1	2
306	1	1	2
307	2	1	2
308	1	2	1

[309 rows x 15 columns]

#independent_variable
y = lung_data. iloc[:,-1:]
print(y)

	LUNG_CANCER
0	YES
1	YES
2	NO
3	NO
4	NO
..	...
304	YES
305	YES
306	YES
307	YES
308	YES

[309 rows x 1 columns]

lung_data.GENDER = lung_data.GENDER.map({"M":1,"F":2})
lung_data.LUNG_CANCER = lung_data.LUNG_CANCER.map({"YES":1,"NO":2})

lung_data.shape

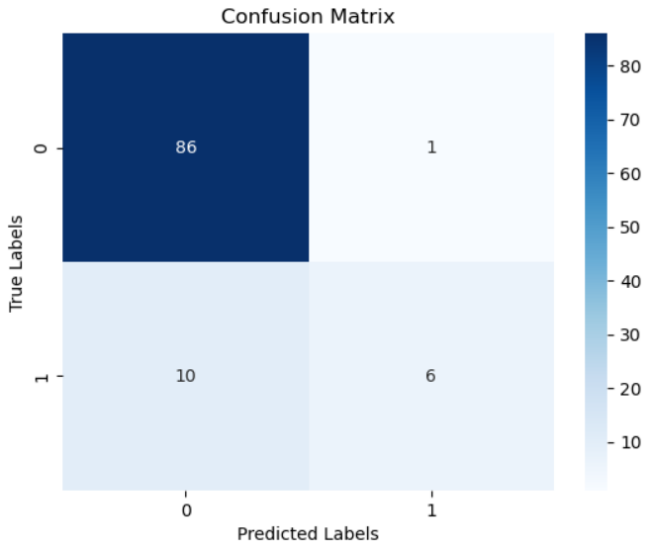
lung_data.isnull().sum()
lung_data.dtypes
lung_data.head()
lung_data.tail()

```
#the describe() method returns description of data in DataFrame
lung_data.describe()

#the info() method prints information of the database
lung_data.info()

#Splitting the Dataset: Training and Testing
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/3,random_state=0)

lung_data['LUNG_CANCER'].value_counts()

len(lung_data)
len(x_test)
len(x_train)

#dependent_variable
x = lung_data.iloc[:,0:-1]
x

#independent_variable
y = lung_data.iloc[:,-1:]
y

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

from sklearn.linear_model import LogisticRegression
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/3,random_state=0)

Logistic Regression

#Fitting simple linear regression to the training test
Model1 = LogisticRegression()
Model1.fit(x_train, y_train)
#Predicting the test set results
prediction1 = Model1.predict(x_test)

prediction1
```

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1,
       1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn.metrics import accuracy_score
confusion_matrix(y_test,prediction1)

accuracy_score(y_test,prediction1)
0.8932038834951457

from sklearn.metrics import precision_score
probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction1, average = None)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction1)
precision = precision_score(y_test, prediction1)
recall = recall_score(y_test, prediction1)
f1 = f1_score(y_test, prediction1)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)

Accuracy: 0.8932038834951457
Precision: 0.8958333333333334
Recall: 0.9885057471264368
F1 score: 0.9398907103825137

from sklearn.metrics import recall_score
from sklearn.metrics import f1_score

recall_score(y_test, prediction1, average = None)

f1_score(y_test, prediction1, average = None)

cm = confusion_matrix(y_true = y_test, y_pred = prediction1)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



KNN

```
from sklearn.neighbors import KNeighborsClassifier
#Fitting K-NN to the Training set
classifier = KNeighborsClassifier(n_neighbors = 3, metric = "minkowski", p = 2)
classifier.fit(x_train, y_train)

#Predicting the Test set result
prediction2 = classifier.predict(x_test)
```

prediction2

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
confusion_matrix(y_test,prediction2)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction2)
precision = precision_score(y_test, prediction2)
recall = recall_score(y_test, prediction2)
f1 = f1_score(y_test, prediction2)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

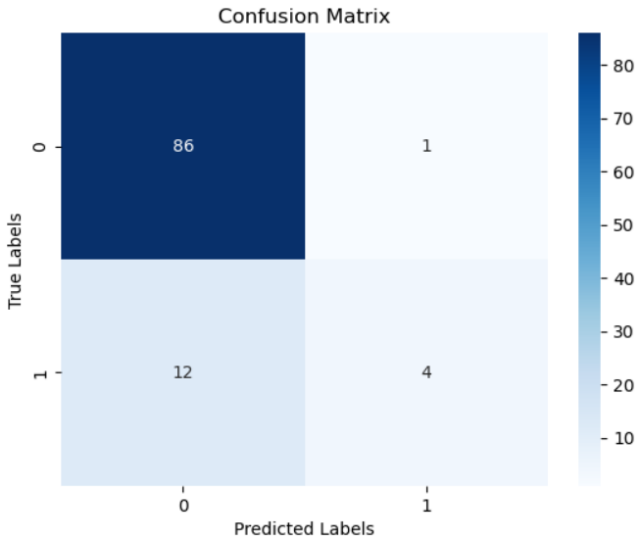
```
Accuracy: 0.8737864077669902
Precision: 0.8775510204081632
Recall: 0.9885057471264368
F1 score: 0.9297297297297297
```

accuracy_score(y_test,prediction2)

probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction2, average = None)

recall_score(y_test, prediction2, average = None)
f1_score(y_test, prediction2, average = None)

cm = confusion_matrix(y_true = y_test, y_pred = prediction2)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()



Decision Tree

#Decision Tree
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state = 0,criterion = "entropy")
tree.fit(x_train, y_train)
prediction3 = classifier.predict(x_test)

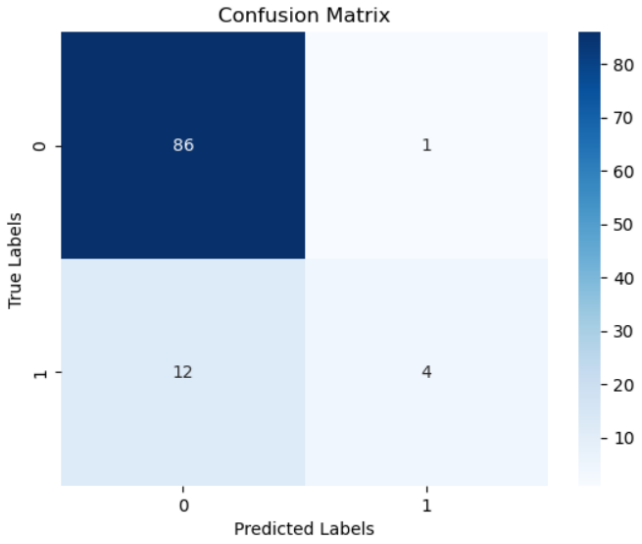
prediction3

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

```
confusion_matrix(y_test,prediction3)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction3)
precision = precision_score(y_test, prediction3)
recall = recall_score(y_test, prediction3)
f1 = f1_score(y_test, prediction3)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8737864077669902
Precision: 0.8775510204081632
Recall: 0.9885057471264368
F1 score: 0.9297297297297297
```

```
accuracy_score(y_test,prediction3)

probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction3, average = None)

recall_score(y_test, prediction3, average = None)
f1_score(y_test, prediction3, average = None)

cm = confusion_matrix(y_true = y_test, y_pred = prediction3)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```

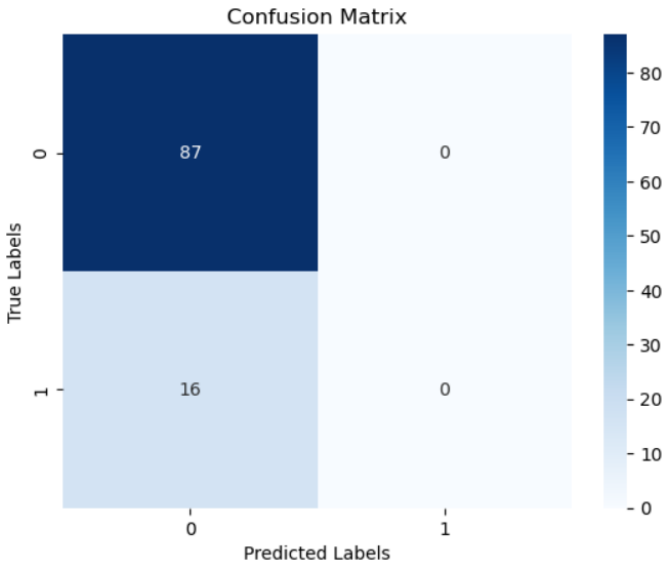


SVM

```
#Support Vector Machine
from sklearn.ensemble import BaggingClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
svm =
OneVsRestClassifier(BaggingClassifier(SVC(C=10,kernel='rbf',random_state=9,probability=
True),n_jobs=-1))
svm.fit(x_train, y_train)
prediction4 = svm.predict(x_test)

confusion_matrix(y_test,prediction4)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction4)
precision = precision_score(y_test, prediction4)
recall = recall_score(y_test, prediction4)
f1 = f1_score(y_test, prediction4)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8446601941747572
Precision: 0.8446601941747572
Recall: 1.0
F1 score: 0.9157894736842105
```

```
accuracy_score(y_test,prediction4)

probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction4, average = None)

recall_score(y_test, prediction4, average = None)
f1_score(y_test, prediction4, average = None)

cm = confusion_matrix(y_true = y_test, y_pred = prediction4)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



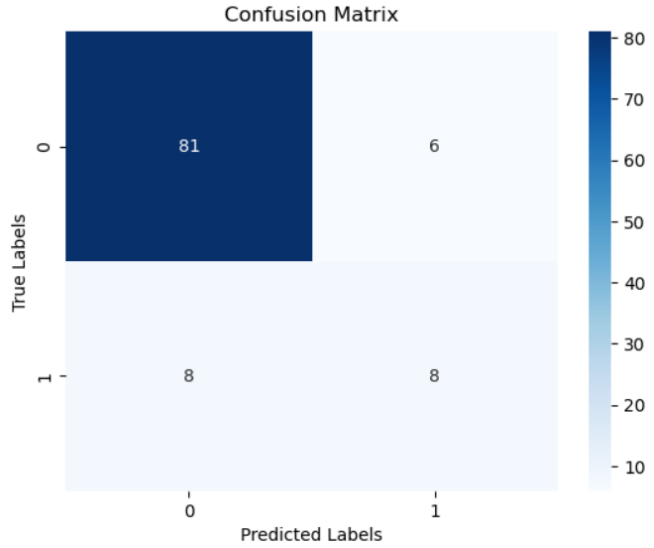
Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
nbcla = GaussianNB()
nbcla.fit(x_train, y_train)
prediction5 = nbcla.predict(x_test)

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
confusion_matrix(y_test,prediction5)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction5)
precision = precision_score(y_test, prediction5)
recall = recall_score(y_test, prediction5)
f1 = f1_score(y_test, prediction5)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Accuracy: 0.8640776699029126
Precision: 0.9101123595505618
Recall: 0.9310344827586207
F1 score: 0.9204545454545454
```

```
accuracy_score(y_test,prediction5)

probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction5, average = None)

recall_score(y_test, prediction5, average = None)
```

```
f1_score(y_test, prediction5, average = None)

cm = confusion_matrix(y_true = y_test, y_pred = prediction5)
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier

# Initialize the classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model using training dataset
rf_classifier.fit(x_train, y_train)

# Make predictions on test dataset
prediction6 = rf_classifier.predict(x_test)

confusion_matrix(y_test,prediction6)

from sklearn.metrics import precision_score, recall_score, f1_score

accuracy = accuracy_score(y_test, prediction6)
precision = precision_score(y_test, prediction6)
recall = recall_score(y_test, prediction6)
f1 = f1_score(y_test, prediction6)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
```

print("F1 score:", f1)

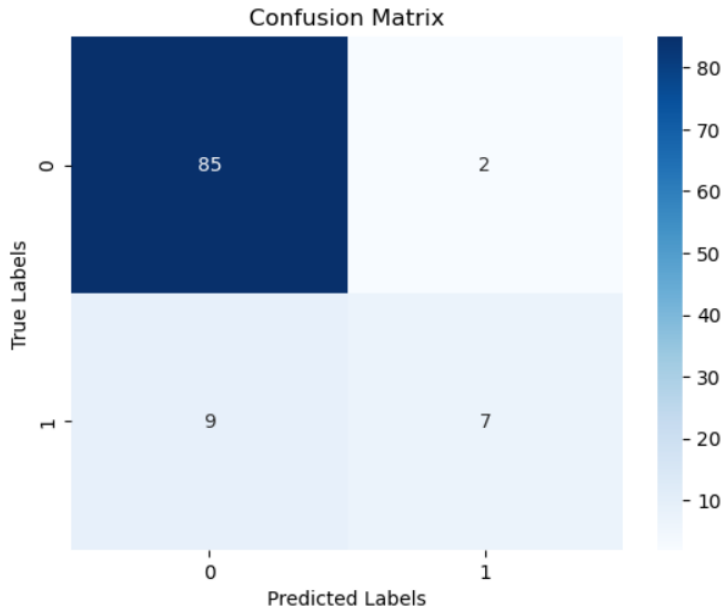
```
Accuracy: 0.8932038834951457
Precision: 0.9042553191489362
Recall: 0.9770114942528736
F1 score: 0.9392265193370166
```

accuracy_score(y_test,prediction6)

probs = Model1.predict_proba(x_test)
precision_score(y_test, prediction6, average = None)

recall_score(y_test, prediction6, average = None)
f1_score(y_test, prediction6, average = None)

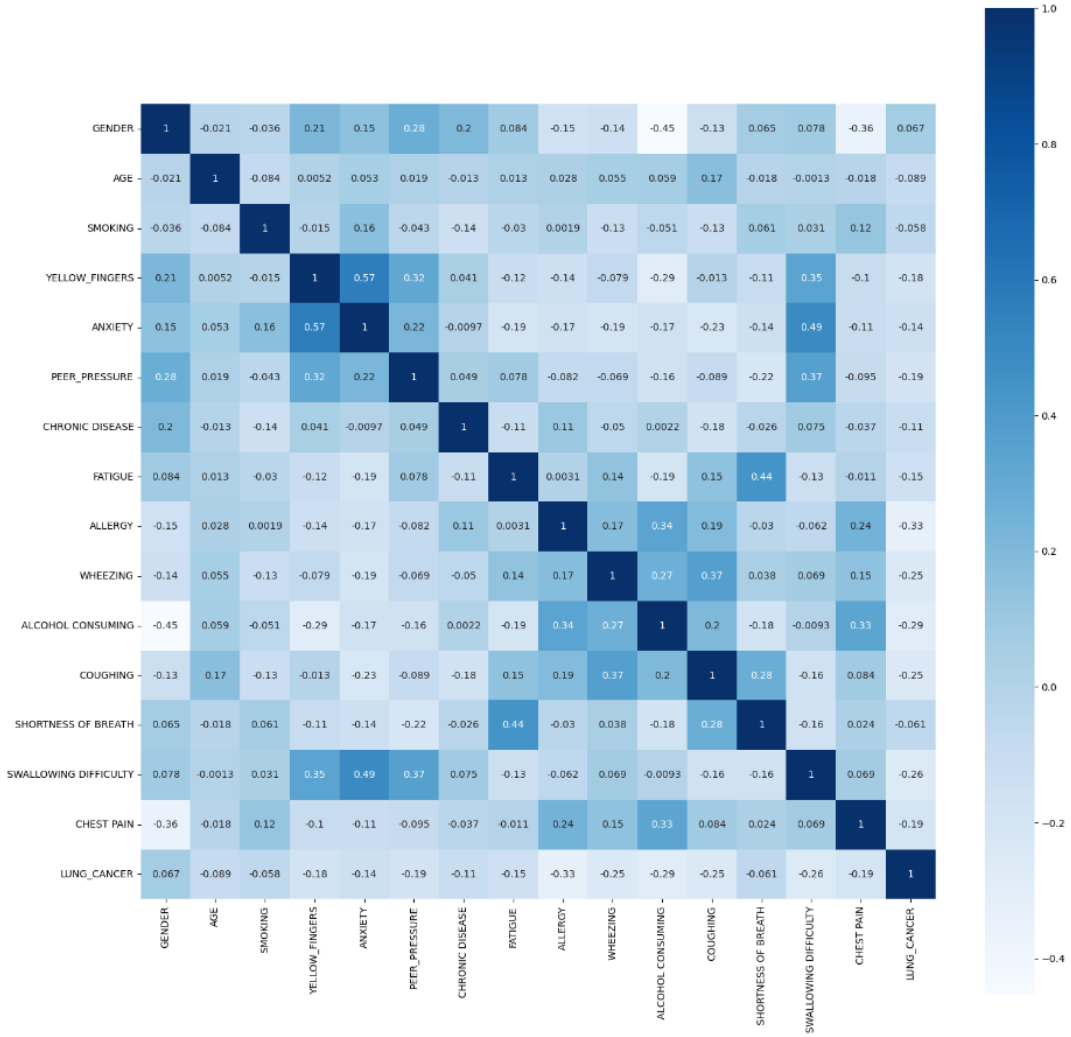
cm = confusion_matrix(y_true = y_test, y_pred = prediction6)
#plot_confusion_matrix(cm,level,title = "confusion_matrix")
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()



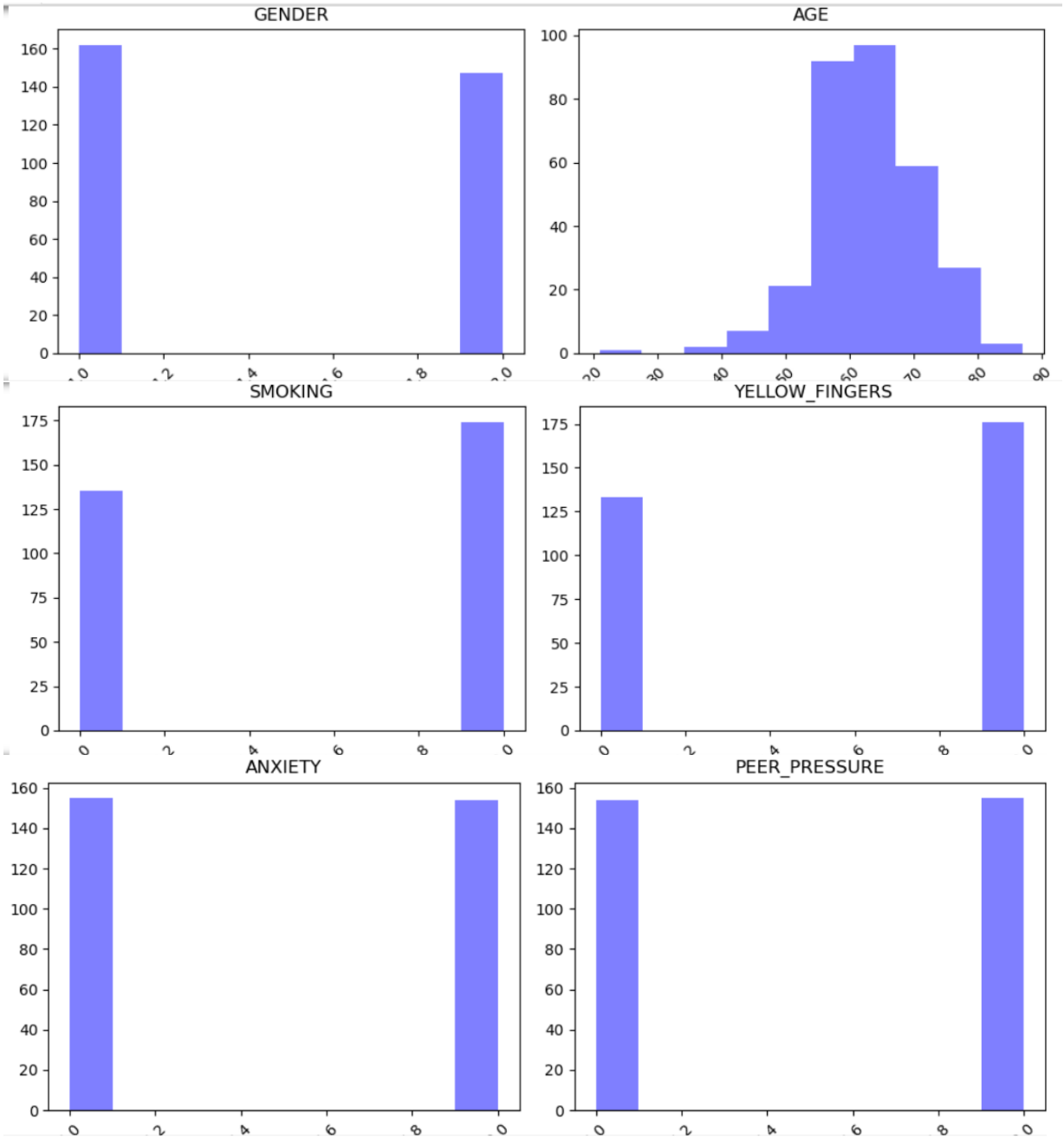
#Finding Correlation
cn=lung_data.corr()
cn

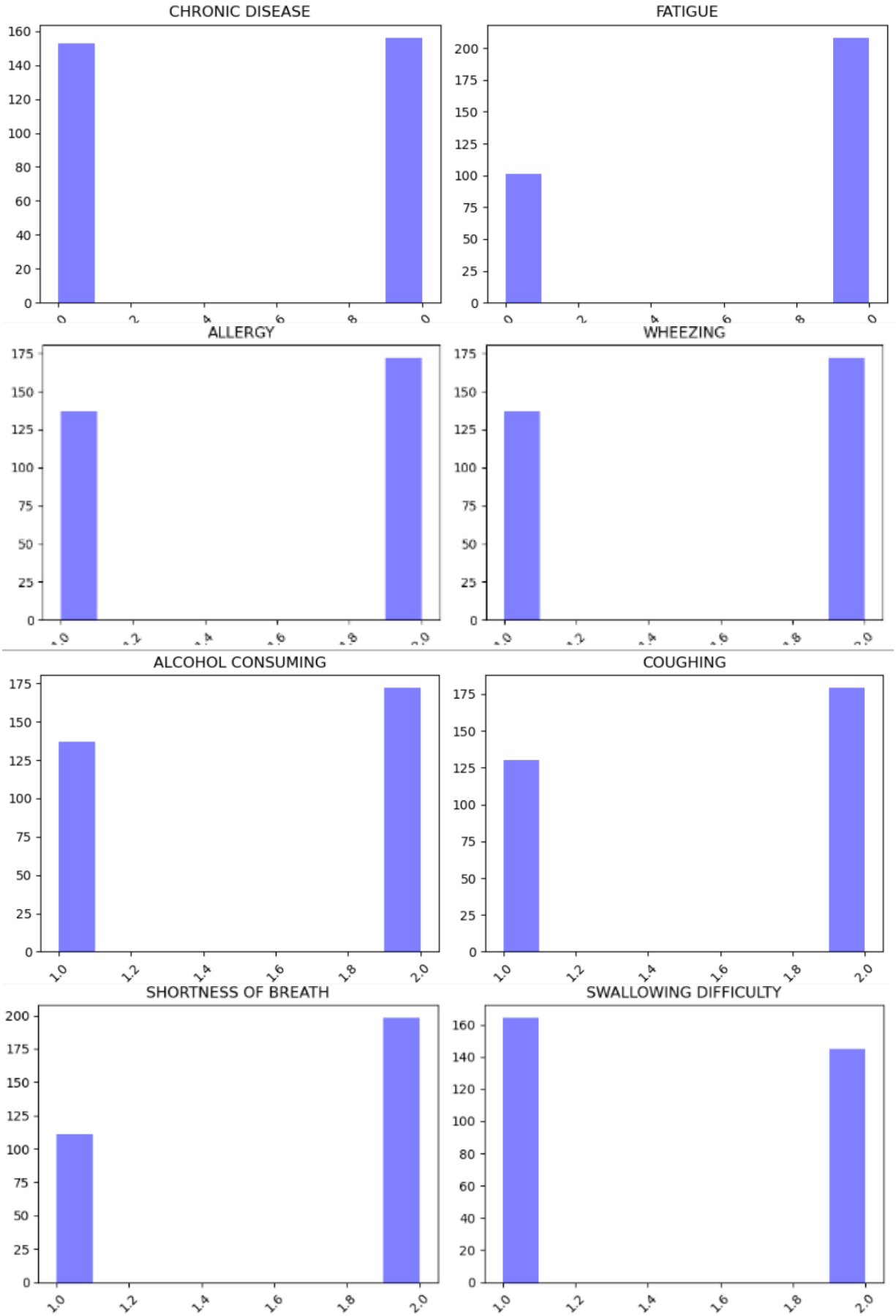
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING
GENDER	1.000000	-0.021306	-0.036277	0.212959	0.152127	0.275564	0.204606	0.083560	-0.154251	-0.141207	-0.45426
AGE	-0.021306	1.000000	-0.084475	0.005205	0.053170	0.018685	-0.012642	0.012614	0.027990	0.055011	0.05898
SMOKING	-0.036277	-0.084475	1.000000	-0.014585	0.160267	-0.042822	-0.141522	-0.029575	0.001913	-0.129426	-0.05062
YELLOW_FINGERS	0.212959	0.005205	-0.014585	1.000000	0.565829	0.323083	0.041122	-0.118058	-0.144300	-0.078515	-0.28902
ANXIETY	0.152127	0.053170	0.160267	0.565829	1.000000	0.216841	-0.009678	-0.188538	-0.165750	-0.191807	-0.16575
PEER_PRESSURE	0.275564	0.018685	-0.042822	0.323083	0.216841	1.000000	0.048515	0.078148	-0.081800	-0.068771	-0.15997
CHRONIC DISEASE	0.204606	-0.012642	-0.141522	0.041122	-0.009678	0.048515	1.000000	-0.110529	0.106386	-0.049967	0.00215
FATIGUE	0.083560	0.012614	-0.029575	-0.118058	-0.188538	0.078148	-0.110529	1.000000	0.003056	0.141937	-0.19137
ALLERGY	-0.154251	0.027990	0.001913	-0.144300	-0.165750	-0.081800	0.106386	0.003056	1.000000	0.173867	0.34433
WHEEZING	-0.141207	0.055011	-0.129426	-0.078515	-0.191807	-0.068771	-0.049967	0.141937	0.173867	1.000000	0.26565
ALCOHOL CONSUMING	-0.454268	0.058985	-0.050623	-0.289025	-0.165750	-0.159973	0.002150	-0.191377	0.344339	0.265659	1.00000
COUGHING	-0.133303	0.169950	-0.129471	-0.012640	-0.225644	-0.089019	-0.175287	0.146856	0.189524	0.374265	0.20272
SHORTNESS OF BREATH	0.064911	-0.017513	0.061264	-0.105944	-0.144077	-0.220175	-0.026459	0.441745	-0.030056	0.037834	-0.17941
SWALLOWING DIFFICULTY	0.078161	-0.001270	0.030718	0.345904	0.489403	0.366590	0.075176	-0.132790	-0.061508	0.069027	-0.00929
CHEST PAIN	-0.362958	-0.018104	0.120117	-0.104829	-0.113634	-0.094828	-0.036938	-0.010832	0.239433	0.147640	0.33122
LUNG_CANCER	0.067254	-0.089465	-0.058179	-0.181339	-0.144947	-0.186388	-0.110891	-0.150673	-0.327766	-0.249300	-0.28853

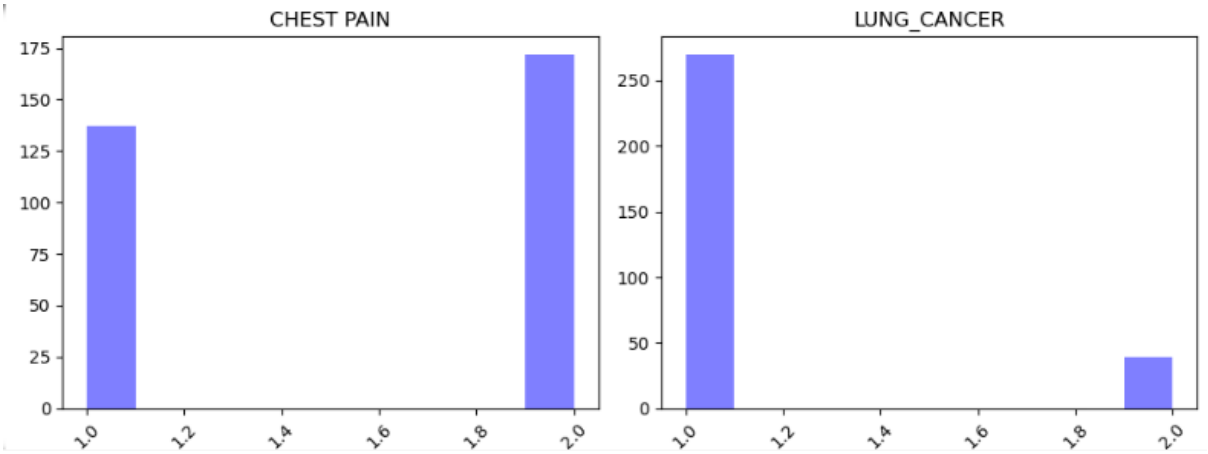
```
#Correlation
cmap=sns.diverging_palette(260,-10,s=50, l=75, n=6,
as_cmap=True)
plt.subplots(figsize=(18,18))
sns.heatmap(cn,cmap="Blues",annot=True, square=True)
plt.show()
```



```
num_list = list(lung_data.columns)

fig = plt.figure(figsize=(10,100))

for i in range(len(num_list)):
    plt.subplot(8,2,i+1)
    plt.title(num_list[i])
    plt.xticks(rotation=45)
    plt.hist(lung_data[num_list[i]],color='blue',alpha=0.5)

plt.tight_layout()
```




CHRONIC DISEASE
FATIGUE
ALLERGY
WHEEZING
ALCOHOL CONSUMING
COUGHING
SHORTNESS OF BREATH
SWALLOWING DIFFICULTY


CHEST PAIN
175
150
125
100
75
50
25
0
1.0
1.2
1.4
1.6
1.8
2.0
LUNG_CANCER
250
200
150
100
50
0
1.0
1.2
1.4
1.6
1.8
2.0