A project report on

LUNG CANCER PREDICTION USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfilment for the award of the degree of

Master of Computer Applications

by

MYTHILY S (24MCA0371)

Under the guidance of

Dr. RATHI R

Associate Professor Grade 1

(SCORE)

May,2024

DECLARATION

I hereby declare that the thesis entitled "**LUNG CANCER PREDICTION USING MACHINE LEARNING ALGORITHMS**" submitted by me, for the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA)** is a record of Bonafide work carried out by me under the supervision of **Dr. RATHI R**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: 30-05-2024 **Signature of the Candidate**

CERTIFICATE

This is to certify that the thesis entitled **"LUNG CANCER PREDICTION USING MACHINE LEARNING ALGORITHMS"** submitted by **MYTHILY S (24MCA0371)**, **SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS (SCORE), SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**,
for the award of the degree **MASTER OF COMPUTER APPLICATIONS (MCA)** is a record of Bonafide work carried out by her under my supervision.

The contents of this report have not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The project report fulfils the requirements and regulations of **SRM INSTITIUE OF SCIENCE AND TECHNOLOGY** and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of the HOD**

Internal Examiner **External Examiner**

ABSTRACT

Lung cancer, one of the dangerous cancers and difficult to diagnose. The effectiveness of cancer prediction helps the people to know their cancer risk, and it is more important to care at immediately. It causes commonly death to both women and the men. Accurate analysis of the stage of nodules is more important for the treatment. Some cancers are reported with genetics makes up. Here comparing the different techniques basedon the ML algorithms to predict the lung cancer presented. Algorithm used, Logistic Regression, k-Nearest Neighbor, Cat Boast, Decision Tree, Random Forest machine learning methods to predict anomaly and to check which algorithm will give the best result and accuracy for the lung cancer prediction. Also creating a website to check the patient has lung cancer or not by using flask web application.

Keywords: Prediction, Lung cancer, algorithms, Machine learning

CONTENTS

CONTENTS…….

LIST OF ACRONYMS.....VII

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION 1

CHAPTER 2

PROJECT DESCRIPTION

2.1 PROJECT DEFINITION.....2

2.2 PROPOSED WORK DIAGRAM..... 3

CHAPTER 3
MODULES DESCRIPTION

3.1 DATASET DESCRIPTION 5

3.2 ALGORITHM DESCRIPTION 5

CHAPTER 4

REQUIREMENT ENGINEERING

4.1 SOFTWARE REQUIREMENT.....7

4.2 HARDWARE REQUIREMENT 7

CHAPTER 5

IMPLEMENTATION

5.1 CODING 8

5.2 MODEL IMPLEMENTATION1 19

5.3 MODEL IMPLEMENTATION2… 27

5.4 COMPARISON OF MODELS 35

5.5 WEB APP IMPLEMENTATION 35

5.6 SCREENSHOT OF WEB APP 38

CHAPTER 6

RESULTS AND DISCUSSION

6.1 RESULT OF THE PROJECT 40

6.2 FRAMEWORK SCREEN SHOT 41

CHAPTER 7

7.1 CONCLUSION 43

7.2 FUTURE WORK 44

7.3 REFERENCES 45

LIST OF ACRONYMS

SCLC	Small Cell Lung Cancer
KNN	K- Nearest Neighbor
AUC	Area Under the Curve
DNN	Deep Neural Network
MLP	Multi-Layer Perceptron
SVM	Support Vector Machine
ACO	Ant Colony Optimization
WHO	World Health Organization
HNN	Hopfield Neural Networks
CAD	Computer Aided Diagnosis
NSCLC	Non-Small Cell Lung Cancer
PCA	Principal Component Analysis
CNN	Convolutional Neural Network
ROC	Receiver Operating Characteristics

Chapter1

Introduction

1.1 INTRODUCTION

Cancers exist in more organs, and simultaneously it has different types. This different type of cancers occurs in various organs of body. This type of disease cannot be noticed by us for long time. WHO reports says, if this disease can be preventedif it is was detected in the early stage. The life span of cancer patient is extended, whether if he/she noticed the disease symptoms and took proper diagnosis at right time.It is different from other cancer, and it is different from the tumor staging from thetime of diagnosis. There are two type of lung cancers, NSCLC and SCLC. Heavy consumption of the Tobacco and the Alcohol is the major reason for affecting the lungs.According to research, the individual person may be affected by nineteen distinct formsof cancer. Lung cancer has the highest death rate among all those tumors. As per theresearch this disease is expected to kill people greater than 1.7 million per year. Machine learning comes to existence to reduce human laborers. Most of the systems lack adequate detection accuracy, and some system must be developed to reach highest accuracy rate of 100%. After the comparison of some machine learningalgorithms, this project provides the best machine learning algorithm to predict the lungcancer with great performance and accuracy rate.

Keywords: Cancer, Machine learning, algorithms, accuracy, tobacco

2.1 PROJECT DEFINITION

The lung cancers prediction is initiated for detecting the lung cancer disease patient from all the other disease patient. This detection helps to give awareness for the peopleabout the danger which is caused by the lung cancer and the information for detection is takenin the form of Kaggle datasets. Kaggle datasets are nothing but data that are already being posted by the hospital and researchers for the purpose of machine learning. Pre- Processing the data such as Checking for missing values, Categorial Values, Feature selection and sampling. Here, we are using ML algorithms like Logistic Regressions, KNN, Cat Boast, Decision Tree, and the Random Forest. We are comparing those algorithms to checkwhether which algorithm gives the great result in predicting the lung cancer. As the result of implemented algorithm which gives Accuracy score, Errors (Outliers or Anomalies) and Classification Report. So, after the process we found the best algorithms for our project, with help of this algorithm we going to implement the website with flask, to check that the patient has lung cancer or not. So, in my project I have done a website using html code, pythoncode to get the details from the user and checking those detail with the founded best algorithm which is stored in pickle files to provide the result for the user. Finally, by getting details from the user my website can provide the result has the patient has lung cancer or not.

2.2 PROPOSED WORK DIAGRAM

This (Figure1) is the flow diagram of my proposed work, in this work first the dataset is taken from the Kaggle, then I am checking whether the dataset has any imbalanced or missing values in it or not. If there is any imbalanced data found, then I must balance it. Then the transformation is takes place here, example if all the values in the dataset has categorical but one attribute has numerical, we must change that numerical value to categorical to normalize it for prediction. Then next step is visualization, if we used to visualize or we can see any attributes we can do with the help of graphs. The next stage is featuring selection, here we are assigning the values for x and y axis, where x as the input values and y is the predictionvalue. The next stage is model training, here I am training the model has 80% and 60%. The next stage is algorithm selection, I have used six algorithms here. Next steps have resulted, and the next stage is comparing the result of one algorithm with another algorithm. Then finally we will get one good algorithm to implement our project. With the help of the best algorithm, I am predicting the output for the data which I am collecting from the user on the website.

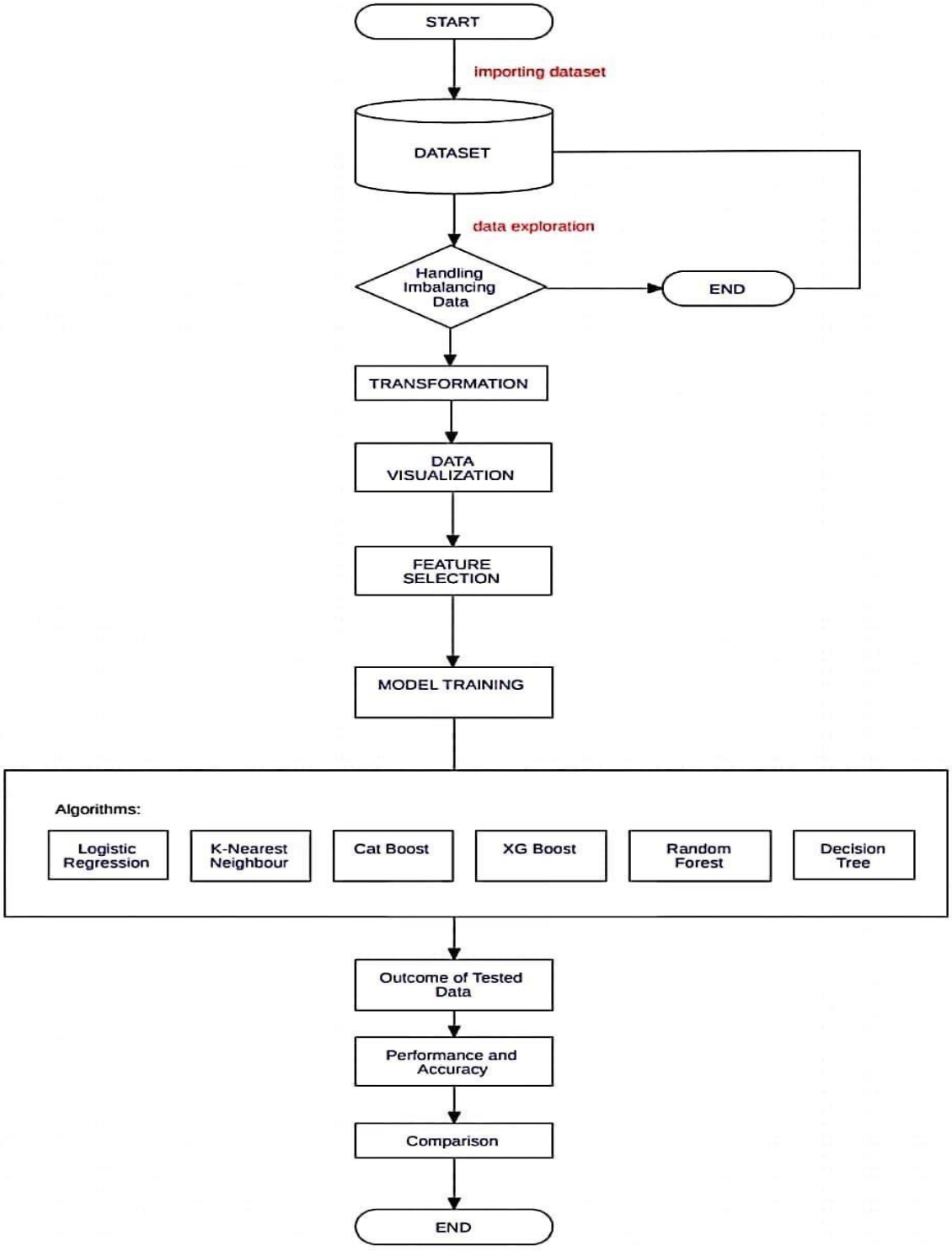

START
importing dataset
DATASET
data exploration
Handling Imbalancing Data
END
TRANSFORMATION
DATA VISUALIZATION
FEATURE SELECTION
MODEL TRAINING
Algorithms:
Logistic Regression
K-Nearest Neighbour
Cat Boost
XG Boost
Random Forest
Decision Tree
Outcome of Tested Data
Performance and Accuracy
Comparison
END

Figure1

Chapter3

Module Description

3.1 DATASET DESCRIPTION

Dataset is taken from Kaggle. The name of this dataset is Lung Cancer which contains 309 values with 16 attributes. Here in this proposed work, we are using all the attributes of the dataset to predict the result. The 16 attributes help the application to find the patient has cancer or not. In this dataset 14 attributes have numerical value and remaining two attributeshas categorical values that is lung cancer and gender has categorical value. I am also changingthe gender value has numerical value for the dataset.

3.2 ALGORITHM DESCRIPTION

In this project, I planned to implement five different ML algorithms to analyze its performance as well as the accuracy of the result on this dataset to predict the result for the data which we are getting from the user. So, by implementing these algorithms I will get one best algorithm, with the best algorithm I am, to predict my resultant values. The algorithms are:

1.LOGISTIC REGRESSION:

Logistic Regression, it is the popular ML Algorithm, it comes under the supervised Learning Technique. It is used to predict the categorial dependent variables with the use of independent variables. It also predicts the output of the categorical dependent variables. In myproject it is helpful to predict the result for the attribute, which is the patient has canceror not. But the output can be a categorical value or discrete value like yes/no ,0 or 1. Mostly recommended for solving classification-based problems.

II. K- NEAREST NEIGHBOUR:

K-NN, it is a Supervised type of M L Algorithm which is widely used for Classification and for Regression problems and mainly used for the Classification problems. It also helps tostores all the avail data. It has good algorithms, which stores all available data, and it will also classifying their new data point with based on their similarity. When we insert new records, it can also find the result for the new records based on their similarity of the old records. It does not make it own assumption but instead it will take decision based on their underlying data.

III. RANDOM FOREST:

Random forest, it is Supervised type of ML Algorithm which is used widely in the Classification and the Regression problems. One of the main parts of these Algorithm is that it can also handle the data set containing the variables which is continuous and also as inthe case of regression. It performs best for classification problems. It is also like the concept ofthe ensemble type learning; this is the process of combining the multiple classifiers for solvingthe complex problem.

IV. CATBOOST:

Cat Boost, it is an algorithm which helps for gradient and boosting on the decision trees. It is like Matrix Net algorithm which is widely in use within their company for ranking the tasks, forecasting and making recommendations. It is the high performance for open-source library for boosting gradient on their decision trees. It also has the feature called categorical feature for supporting to improve our training result with the cat boost, that allows to pre- process our data. It also had the feature called fast prediction and improved accuracy.

V. DECISION TREE:

Decision Tree, it is a Supervised type of machine learning technique, it also used for both the classifications and the Regressions problem, but mainly it prefers for solving the Classification problems. It is the tree-structured classifier, which has internal nodes which represent the feature of the data, branches represent the decisions rules and then each the leaf represents the output/outcome. The test or the decision is performed on their feature basis from the given datasets. It is like a graph representation; it gets all their possible solutions for the problem/decisions based on their given condition.

Chapter4

Requirement Engineering

4.1 SOFTWARE REQUIREMENTS

Programming Languages & Tools for Implementation:

→ Python 3.8 , Jupyter Notebook, Anaconda, vstudio

4.2 HARDWARE REQUIREMENTS

- ➔ Processor: PENTIUM IV 2.6 GHZ, Intel Core 2 Duo
- ➔ RAM: 512 MB DD RAM
- ➔ Monitor: 15” Color
- ➔ Hard Disk: 50GB

Chapter5

Implementation

5.1 CODING

• Importing Packages and Declaring Variables

Here I am importing some of the packages that I need to implement my project, so here I need NumPy to perform the mathematical operations, pandas are used for data analysis, seaborn is used for visualizing data, matplotlib is used for graphical plotting, os is a system library and plotly also used for graphical plotting.

```
In [1]: #IMPORTING PACKAGES AND DECLARING VARIABLES
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        import plotly.graph_objs as go
        import plotly.express as px
        import os
```

• Reading Dataset and Data pre-processing

Here I am reading the dataset to perform the implementation process. With the helpof the python code, I am the dataset from location where the file is stored.

```
In [2]: # READING DATASET

        data = pd.read_csv('slc.csv')
```

- Glimpse the Data by Looking the first 5 rows of the dataset.

Here I am visualizing my dataset first 5 rows to see how many attributes I have andwhat are all the attributes.

```
In [3]: #VIEW OF FIRST FIVE ROWS

        data.head()
```

Out[3]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING
0	M	69	1	2	2	1	1	2	1	2	2
1	M	74	2	1	1	1	2	2	2	1	1
2	F	59	1	1	1	2	1	2	1	2	1
3	M	63	2	2	2	1	1	1	1	1	2
4	F	63	1	2	1	1	1	1	1	2	1

- Looking more details of the data

Here I am looking for the shape of the dataset and for the types of datasets to see which attributes has which data type and also how many rows and columns my dataset has. Also, I used query called data.info to check any data has null values or not and to check dataset information.

In [4]:
```
# LOOKIND DATASET SHAPE AND DATASET DATA TYPES

#DATA SHAPE
print(data.shape)

#DATA TYPE
print(data.dtypes)
```

```
(309, 16)
GENDER                       object
AGE                           int64
SMOKING                       int64
YELLOW_FINGERS                int64
ANXIETY                       int64
PEER_PRESSURE                 int64
CHRONIC DISEASE               int64
FATIGUE                       int64
ALLERGY                       int64
WHEEZING                      int64
ALCOHOL CONSUMING             int64
COUGHING                      int64
SHORTNESS OF BREATH           int64
SWALLOWING DIFFICULTY         int64
CHEST PAIN                    int64
LUNG_CANCER                  object
dtype: object
```

In [5]:
```
# CHECKING DATASET INFORMATION
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   GENDER                  309 non-null    object
 1   AGE                     309 non-null    int64
 2   SMOKING                 309 non-null    int64
 3   YELLOW_FINGERS          309 non-null    int64
 4   ANXIETY                 309 non-null    int64
 5   PEER_PRESSURE           309 non-null    int64
 6   CHRONIC DISEASE         309 non-null    int64
 7   FATIGUE                 309 non-null    int64
 8   ALLERGY                 309 non-null    int64
 9   WHEEZING                309 non-null    int64
 10  ALCOHOL CONSUMING       309 non-null    int64
 11  COUGHING                309 non-null    int64
 12  SHORTNESS OF BREATH     309 non-null    int64
 13  SWALLOWING DIFFICULTY   309 non-null    int64
 14  CHEST PAIN              309 non-null    int64
 15  LUNG_CANCER             309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```

- Checking for any null values:

Here I am checking with the dataset, whether there is any null values presented in this dataset or not, and if presented the count of the null data.

```
In [6]: #CHECKING FOR ANY NULL VALUES

        data.isnull().sum()
```

```
Out[6]: GENDER                    0
        AGE                       0
        SMOKING                   0
        YELLOW_FINGERS            0
        ANXIETY                   0
        PEER_PRESSURE             0
        CHRONIC DISEASE           0
        FATIGUE                   0
        ALLERGY                   0
        WHEEZING                  0
        ALCOHOL CONSUMING         0
        COUGHING                  0
        SHORTNESS OF BREATH       0
        SWALLOWING DIFFICULTY     0
        CHEST PAIN                0
        LUNG_CANCER               0
        dtype: int64
```

- Replacing Gender column values as ‘M’ to 0 and ‘F’ to 1:

Since my dataset has 16 attributes, in that 14 has numerical data and remaining two has categorial values. In those 15 attributes is considered as input data and the last one that is lung cancer attribute is the considered as the output data. So, from the input data 13 has numerical values and only one has categorical values, so I am changing the gender attribute from categorical to numerical data type by using the below python query.

```
In [10]: data.replace({"GENDER":{'M':0,'F':1}},inplace=True)

         #printing first five rows to check whether it is changed or not.
         data.head(5)
```

Out[10]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATI
0	0	69	1	2	2	1	1	
1	0	74	2	1	1	1	2	
2	1	59	1	1	1	2	1	
3	0	63	2	2	2	1	1	
4	1	63	1	2	1	1	1	

```
In [11]: #checking datatype of lung cancer column now:
         data['GENDER'].value_counts()
```

```
Out[11]: 0    162
         1    147
         Name: GENDER, dtype: int64
```

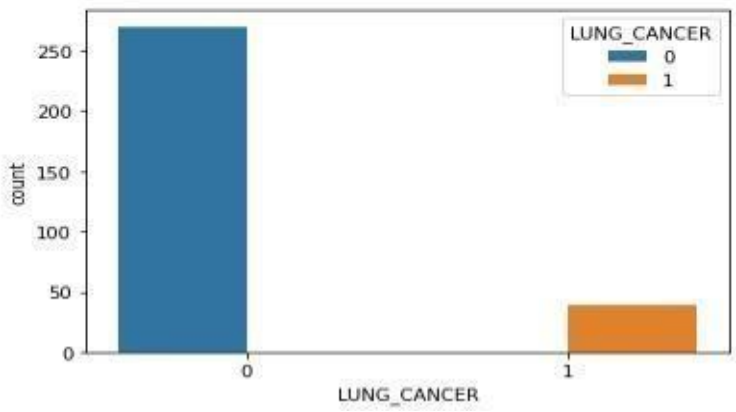
- Diagram representation

This diagram represents the data of the lung cancer patient. This is bar chart type diagram here 0 represent the patient who has lung cancer and 1 represent the patient who did not has lung cancer, and these things are plotted in the form of graph.

```
In [12]: import seaborn as sns
         sns.countplot(x='LUNG_CANCER',hue='LUNG_CANCER',data=data)
```

```
Matplotlib is building the font cache; this may take a moment.
```

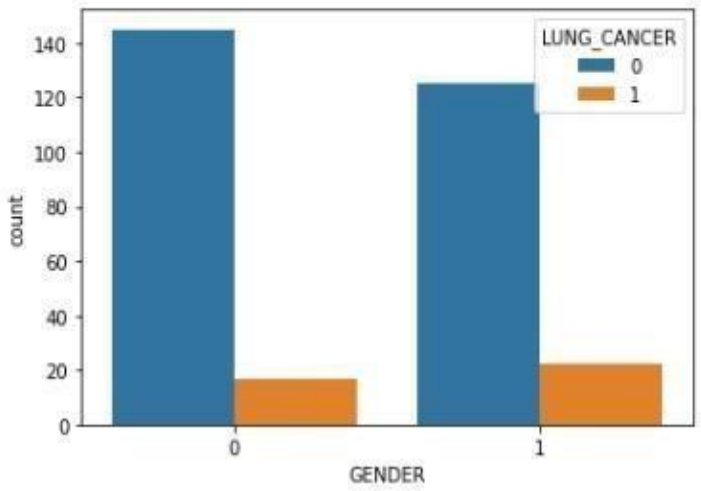
```
Out[12]: <AxesSubplot:xlabel='LUNG_CANCER', ylabel='count'>
```



This diagram represents that how many males has lung cancer that is plotted as 0 (Here blue color represents the count of male which has lung cancer and orange color represents the count of male has no lung cancer). And, how many females has lung cancer that is plotted as 1. (Here blue color represents the count of female which has lung cancer and orange color represents the count of male has no lung cancer)

```
In [13]:
import seaborn as sns
sns.countplot(x='GENDER',hue='LUNG_CANCER',data=data)
```

```
Out[13]: <AxesSubplot:xlabel='GENDER', ylabel='count'>
```

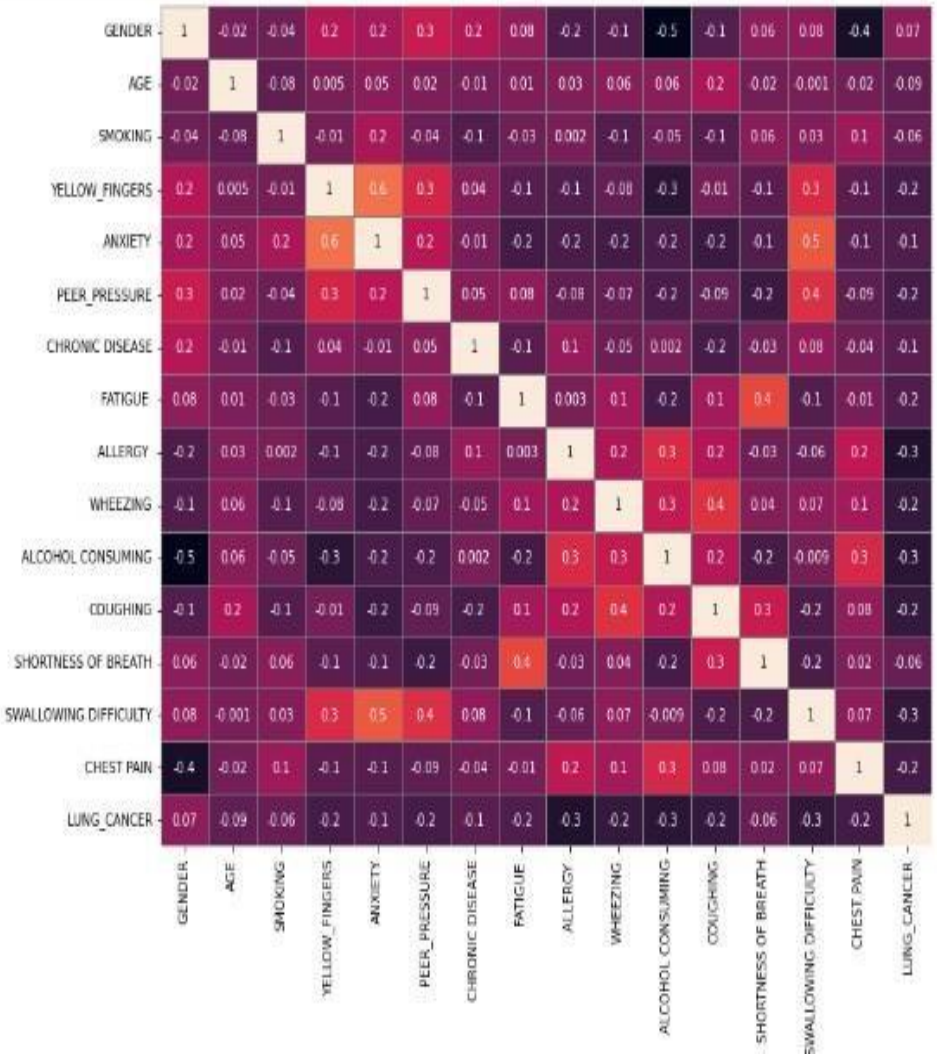


• Heat map representation:

Here we are visualizing all the attributes from our dataset. The entries were standardized by using p- values. Here the p -values is mapped from [-0.02 ,+0.5]. Then the standardization is doing along with rows from the hierarchical clustering algorithm. Here the correlation value is 1. By giving the condition has data.corr() it gives the result has the correlated values, which we can see from the diagram.

```
In [14]: def custom_palette(custom_colors):
             customPalette = sns.set_palette(sns.color_palette(custom_colors))
             sns.palplot(sns.color_palette(custom_colors),size=0.8)
             plt.tick_params(axis='both',labelsize=0,length=0)
```

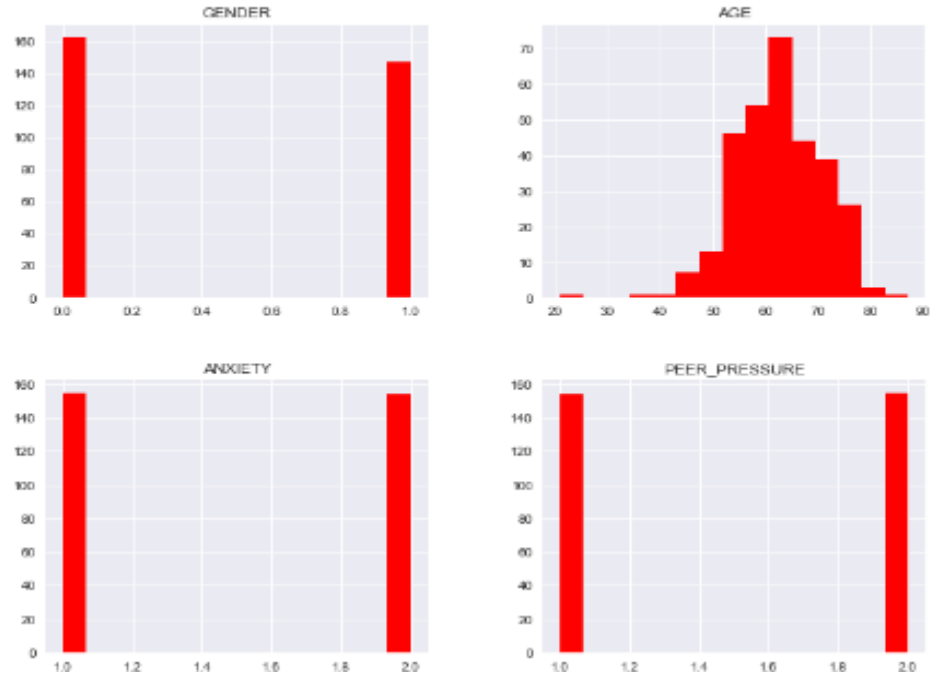
```
In [15]: fig, ax=plt.subplots(figsize=(12,10))
         sns.heatmap(data.corr(),annot=True,fmt='.1g',cbar=False,linewidths=0.5,linecolor='grey');
```

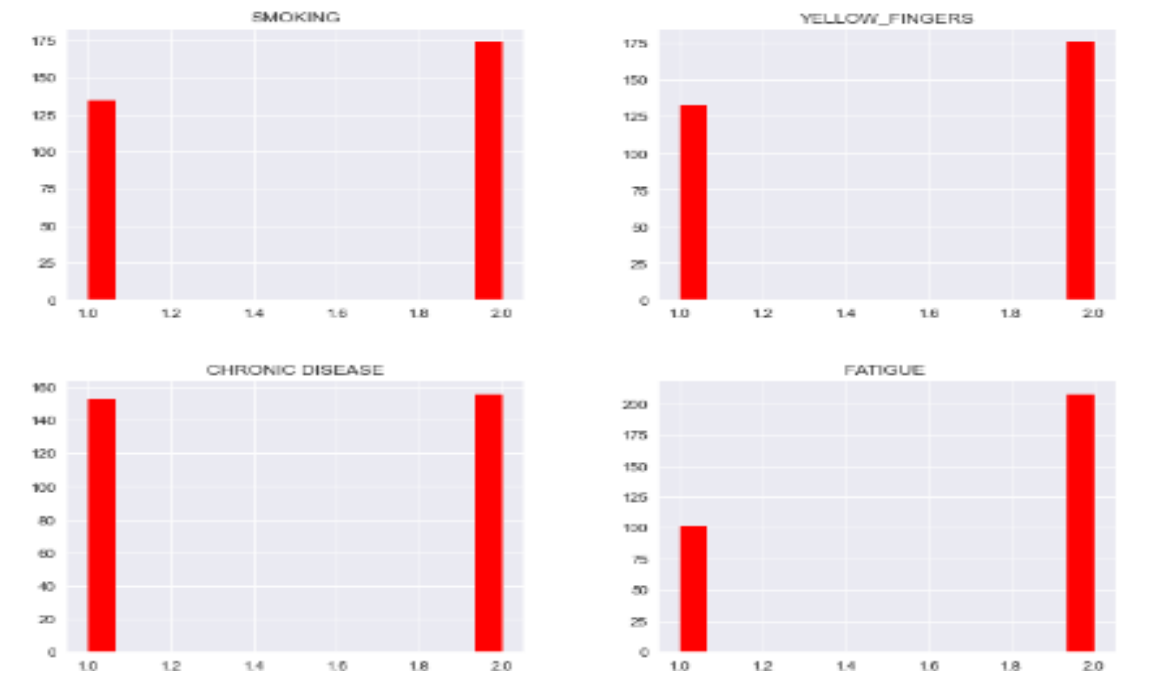


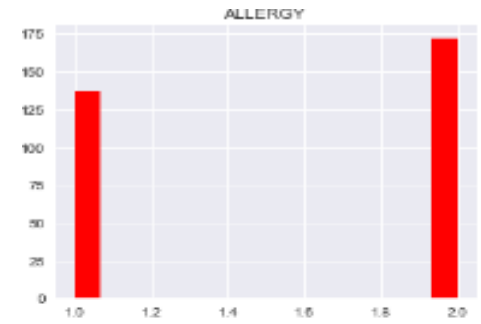
• Overall attributes representation:

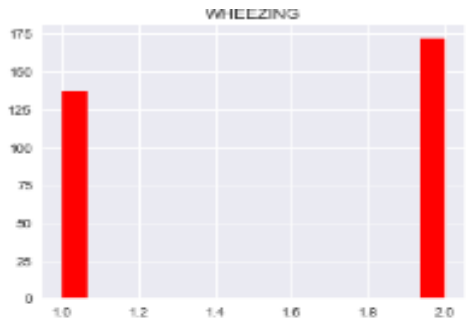
This is histogram type of diagram, here we are visualizing every attribute of our dataset. Example if we take gender how many female (1) counts and male (0) counts are represented in the form of diagram. Likewise, each and all attributes are plotted in histogram.

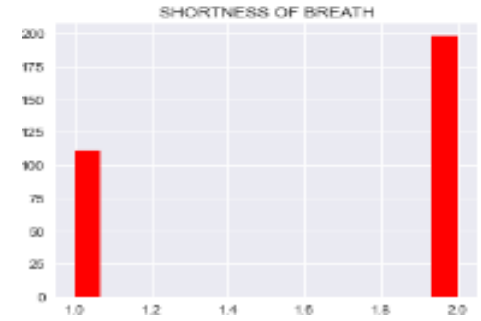
In [16]:
```
plt.style.use("seaborn")
data.hist(figsize=(25,20),color="red",bins=15);
```

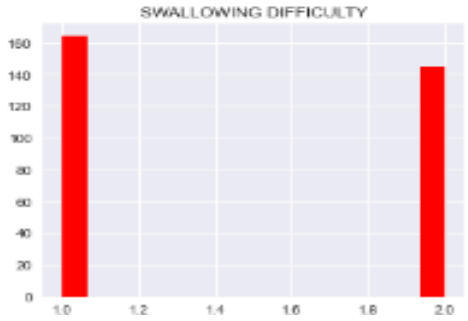


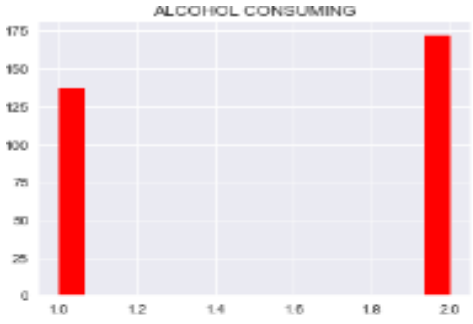


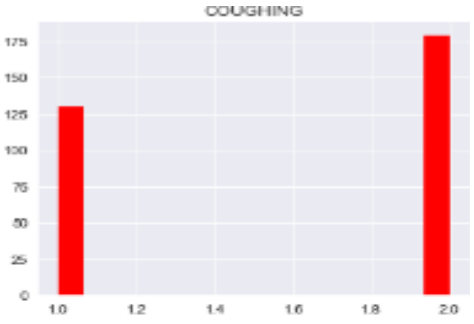


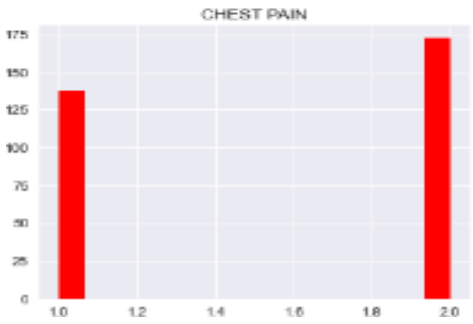


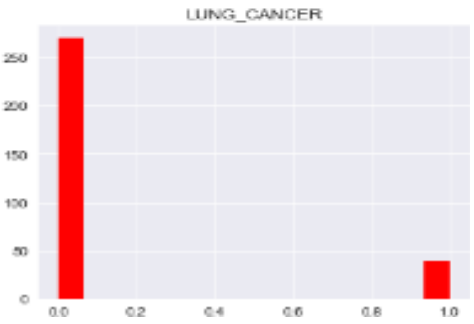












• Splitting Dataset into Training Data (80%) and Test Data (20%)

Here I am splitting my dataset into two division. The one has x and another one has y. In the x division I am assigning all the 15 attributes expect the one lung cancer attributes, because

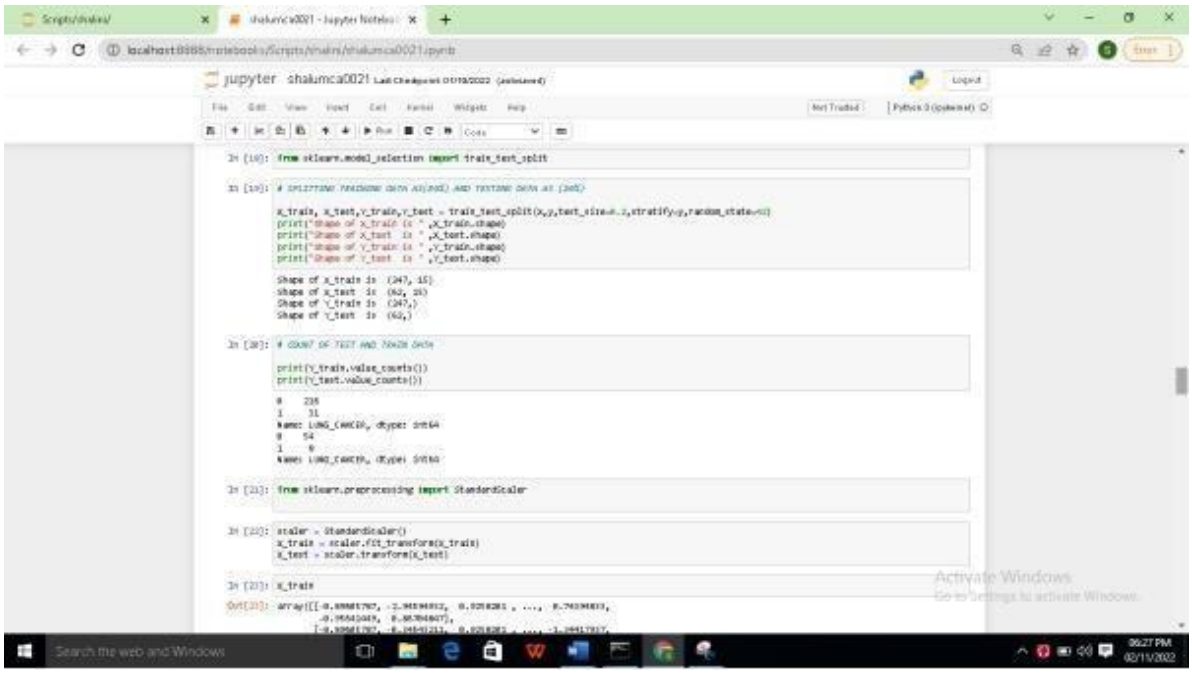
15 attributes are input attributes for our projects. And in y division I am assigning lung cancer attribute alone, because lung cancer is the output attribute. Here also we are splitting the dataset has 80 percent for training our model and 20 percent for testing.

```
In [17]: # SPLITTING THE DATA SET INTO TRAINING SET AND TESTING SET:
         # FEATURING:

         X = data.drop(columns=['LUNG_CANCER'],axis=1)
         y = data['LUNG_CANCER']

         print("The shape of X is " , X.shape)
         print("The shape of Y is " , y.shape)
```

```
The shape of X is  (309, 15)
The shape of Y is  (309,)
```

```
In [19]: # SPLITTING TRAINING DATA AS(80%) AND TESTING DATA AS (20%)

X_train, X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.2,stratify=y,random_state=42)
print("Shape of X_train is " ,X_train.shape)
print("Shape of X_test  is " ,X_test.shape)
print("Shape of Y_train is " ,Y_train.shape)
print("Shape of Y_test  is " ,Y_test.shape)
```

```
Shape of X_train is  (247, 15)
Shape of X_test  is  (62, 15)
Shape of Y_train is  (247,)
Shape of Y_test  is  (62,)
```



• Data Shape of Training Data and Testing Data

Here we are checking the shape of the training and the testing data to verify whether the splitting is perfectly implemented or not. And we are counting the trainingoutput values and testing output values. And we also checking the shape x training and testing and the shape of the y testing and y training.

```
In [18]: from sklearn.model_selection import train_test_split

In [19]: # SPLITTING TRAINING DATA AS(80%) AND TESTING DATA AS (20%)

         X_train, X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.2,stratify=y,random_state=42)
         print("Shape of X_train is " ,X_train.shape)
         print("Shape of X_test  is " ,X_test.shape)
         print("Shape of Y_train is " ,Y_train.shape)
         print("Shape of Y_test  is " ,Y_test.shape)

         Shape of X_train is  (247, 15)
         Shape of X_test  is  (62, 15)
         Shape of Y_train is  (247,)
         Shape of Y_test  is  (62,)

In [20]: # COUNT OF TEST AND TRAIN DATA

         print(Y_train.value_counts())
         print(Y_test.value_counts())

         0    216
         1     31
         Name: LUNG_CANCER, dtype: int64
         0    54
         1     8
         Name: LUNG_CANCER, dtype: int64
```

• Fitting the model to the dataset

Here we are importing the standard scaler library. Standard scaler is used for fitting our training dataset and testing dataset into our model for implementing algorithms. Fitting training and the testing data is more important for algorithm implementation.

```
In [21]: from sklearn.preprocessing import StandardScaler

In [22]: scaler = StandardScaler()
         X_train = scaler.fit_transform(X_train)
         X_test = scaler.transform(X_test)

In [23]: X_train

Out[23]: array([[-0.89605787, -2.94594932,  0.9258201 , ...,  0.74394833,
                 -0.95641449,  0.86704847],
                [-0.89605787, -0.34543211,  0.9258201 , ..., -1.34417937,
                 -0.95641449,  0.86704847],
                [ 1.11599935,  0.1499045 , -1.08012345, ..., -1.34417937,
                 -0.95641449, -1.15333806],
                ...,
                [-0.89605787, -0.34543211, -1.08012345, ..., -1.34417937,
                 -0.95641449, -1.15333806],
                [ 1.11599935,  0.02607035, -1.08012345, ...,  0.74394833,
                 -0.95641449, -1.15333806],
                [ 1.11599935, -0.46926626,  0.9258201 , ...,  0.74394833,
                  1.04557178, -1.15333806]])
```

5.2 MODEL IMPLEMENTATION

Predictive Model Implementation for 80% and 20% data:

Here we are splitting, dataset into the 2 parts one is testing, and the other one is training. So here I am first splitting my dataset has 80 percent has training and 20 percent has testing and implementing all the five algorithms.

Logistic regression:

Here I am using logistic regression ml algorithm, to check whether it gives how much as accuracy rate for lung cancer.

```
In [24]: #  PROPOSED MODEL IMPLEMENTATION

         #LOGISTIC REGRESSION:

         from sklearn.linear_model import LogisticRegression
         from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

         lr = LogisticRegression()
         lr.fit(X_train, Y_train)
         y_pred = lr.predict(X_test)

         lr_train_acc = accuracy_score(Y_train, lr.predict(X_train))
         lr_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result of the experiment the logistic regression gives the training accuracy has 95 percent, and the testing accuracy has 92 percent.

```
print(f"Training Accuracy of Logistic Regression Model is {lr_train_acc}")
print(f"Test Accuracy of Logistic Regression Model is {lr_test_acc}")
```

```
Training Accuracy of Logistic Regression Model is 0.951417004048583
Test Accuracy of Logistic Regression Model is 0.9193548387096774
```

```
In [25]: confusion_matrix(Y_test, y_pred)

Out[25]: array([[53,  1],
                [ 4,  4]], dtype=int64)
```

```
In [26]: print(classification_report(Y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.93	0.98	0.95	54
1	0.80	0.50	0.62	8
accuracy			0.92	62
macro avg	0.86	0.74	0.79	62
weighted avg	0.91	0.92	0.91	62

K- Nearest Neighbor:

Here I am using KNN Classification ml algorithm, to check whether it gives how much as accuracy rate for lung cancer.

```
In [27]: #K-NEAREST NEIGHBOUR CLASSIFIER:

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)

y_pred = knn.predict(X_test)

knn_train_acc = accuracy_score(Y_train, knn.predict(X_train))
knn_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result, the experiment the KNN classifier gives the training accuracy has 92 percent, and the testing accuracy has 90 percent.

```
print(f"Training Accuracy of KNN Model is {knn_train_acc}")
print(f"Test Accuracy of KNN Model is {knn_test_acc}")
```

```
Training Accuracy of KNN Model is 0.9230769230769231
Test Accuracy of KNN Model is 0.9032258064516129
```

```
In [28]: confusion_matrix(Y_test, y_pred)

Out[28]: array([[52,  2],
                [ 4,  4]], dtype=int64)
```

```
In [29]: print(classification_report(Y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.93	0.96	0.95	54
1	0.67	0.50	0.57	8
accuracy			0.90	62
macro avg	0.80	0.73	0.76	62
weighted avg	0.89	0.90	0.90	62

Random Forest:

Here I am using Random Forest Classification ml algorithm, to check whether it gives how much as accuracy rate for lung cancer.

```
In [30]: # RANDOM FOREST CLASSIFIER

         from sklearn.ensemble import RandomForestClassifier

         rand_clf = RandomForestClassifier(criterion = 'gini', max_depth = 3, max_features = 'sqrt', min_samples_leaf = 2, min_samples_
         rand_clf.fit(X_train, Y_train)

         y_pred = rand_clf.predict(X_test)

         rand_clf_train_acc = accuracy_score(Y_train, rand_clf.predict(X_train))
         rand_clf_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result, the experiment the Random Forest model gives the training accuracy has 89 percent,athe testing accuracy has 85 percent.

```
print(f"Training Accuracy of Random Forest Model is {rand_clf_train_acc}")
print(f"Test Accuracy of Random Forest Model is {rand_clf_test_acc}")
```

```
Training Accuracy of Random Forest Model is 0.8947368421052632
Test Accuracy of Random Forest Model is 0.8548387096774194
```

```
In [31]: confusion_matrix(Y_test, y_pred)

Out[31]: array([[53,  1],
                [ 8,  0]], dtype=int64)
```

Cat Boost:

Here I am using Cat Boost Gradient Classifier ml algorithm, to check whether it gives how much as accuracy rates of lung cancer.

```
In [32]: #Cat boost
         from catboost import CatBoostClassifier

         cat = CatBoostClassifier(iterations = 30, learning_rate = 0.1)
         cat.fit(X_train, Y_train)

         y_pred = cat.predict(X_test)
```

```
0:      learn: 0.6645273        total: 54.4ms   remaining: 1.58s
1:      learn: 0.6165877        total: 56.1ms   remaining: 785ms
2:      learn: 0.5919552        total: 60ms     remaining: 540ms
3:      learn: 0.5663639        total: 64.1ms   remaining: 417ms
4:      learn: 0.5449063        total: 68ms     remaining: 340ms
5:      learn: 0.5265795        total: 72.2ms   remaining: 289ms
```

Experimental Results of training and testing dataset:

As a result, the experiment the Cat boost classifier model gives the training accuracy has 93 percent, and the testing accuracy has 88 percent.

```
28:     learn: 0.2741660        total: 133ms    remaining: 4.59ms
29:     learn: 0.2707071        total: 136ms    remaining: 0us
```

In [33]:
```
cat_train_acc = accuracy_score(Y_train, cat.predict(X_train))
cat_test_acc = accuracy_score(Y_test, y_pred)

print(f"Training Accuracy of Cat Boost Classifier Model is {cat_train_acc}")
print(f"Test Accuracy of Cat Boost Classifier Model is {cat_test_acc}")
```

```
Training Accuracy of Cat Boost Classifier Model is 0.9311740890688259
Test Accuracy of Cat Boost Classifier Model is 0.8870967741935484
```

In [34]:
```
confusion_matrix(Y_test, y_pred)
```

Out[34]:
```
array([[52,  2],
       [ 5,  3]], dtype=int64)
```

In [35]:
```
print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.91      0.96      0.94        54
           1       0.60      0.38      0.46         8

    accuracy                           0.89        62
   macro avg       0.76      0.67      0.70        62
weighted avg       0.87      0.89      0.88        62
```

Decision Tree:

Here I am using Decision tree classifier ml algorithm, to check whether it gives how much as accuracy rate for lung cancer. Here we can see the diagrammatic represented of our dataset. We can easily understand with help of these representation.

```
In [36]: from sklearn.tree import DecisionTreeClassifier

         dtc = DecisionTreeClassifier()
         dtc.fit(X_train, Y_train)

         y_pred = dtc.predict(X_test)

         dtc_train_acc = accuracy_score(Y_train, dtc.predict(X_train))
         dtc_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result, the experiment the Decision tree classifier model gives the training accuracy has 100 percent, and the testing accuracy has 89 percent.

```
print(f"Training Accuracy of Decision Tree Model is {dtc_train_acc}")
print(f"Test Accuracy of Decision Tree Model is {dtc_test_acc}")
```

```
Training Accuracy of Decision Tree Model is 1.0
Test Accuracy of Decision Tree Model is 0.8870967741935484
```

```
In [37]: confusion_matrix(Y_test, y_pred)
```

```
Out[37]: array([[49,  5],
                [ 2,  6]], dtype=int64)
```

```
In [38]: print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.91      0.93        54
           1       0.55      0.75      0.63         8

    accuracy                           0.89        62
   macro avg       0.75      0.83      0.78        62
weighted avg       0.91      0.89      0.89        62
```

Result comparison of all models of 80% training data:

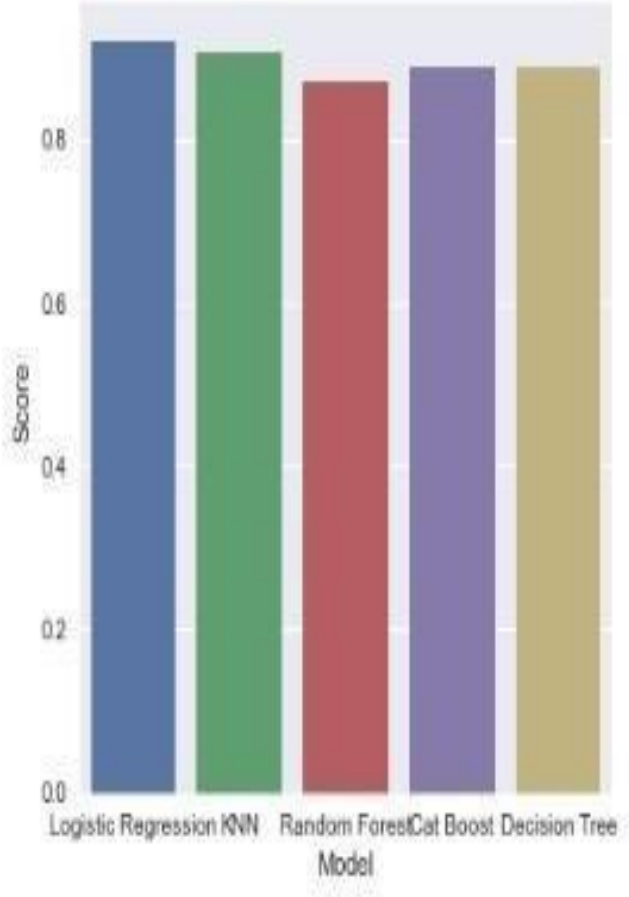
So as per the result comparison of all the five-algorithm logistic regression gives the accuracy has 95%, and KNN gives the accuracy has 92%, Random Forest gives the accuracy has 89%, Cat Boost gives the accuracy as 92% and the Decision tree gives the accuracy has 89%. So, by comparing the training accuracy and the testing accuracy of each algorithm with other algorithm, Logistic regression the good accuracy.

In [39]:
```
models = ['Logistic Regression', 'KNN', 'Random Forest','Cat Boost', 'Decision Tree']
scores = [lr_test_acc, knn_test_acc, rand_clf_test_acc,dtc_test_acc,cat_test_acc]

models = pd.DataFrame({'Model' : models, 'Score' : scores})


models.sort_values(by = 'Score', ascending = True)
```

Out[39]:

	Model	Score
2	Random Forest	0.870968
3	Cat Boost	0.887097
4	Decision Tree	0.887097
1	KNN	0.903226
0	Logistic Regression	0.919355

```
In [40]: plt.figure(figsize = (5, 5))

         sns.barplot(x = 'Model', y = 'Score', data = models)
         plt.show()
```



5.3 MODEL IMPLEMENTATION

Splitting Dataset into Training Data (60%) and Test Data (40%)

Here we are dividing the dataset into two parts one is testing, and the other one is training. So here I am first splitting my dataset has 60 percent has training and 40 percent has testing and implementing all the five algorithms. Because in previous implementation logistic regression gives the good output, so here we are changing the training as 60 and testing as 40 to see which is giving best output.

In [34]:
```
# SPLITTING TRAINING DATA AS(60%) AND TESTING DATA AS (40%)

X_train, X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.4,stratify=y,random_state=42)
print("Shape of X_train is " ,X_train.shape)
print("Shape of X_test  is " ,X_test.shape)
print("Shape of Y_train is " ,Y_train.shape)
print("Shape of Y_test  is " ,Y_test.shape)
```

```
Shape of X_train is  (185, 15)
Shape of X_test  is  (124, 15)
Shape of Y_train is  (185,)
Shape of Y_test  is  (124,)
```

In [35]:
```
print(Y_train.value_counts())
print(Y_test.value_counts())
```

```
0    162
1     23
Name: LUNG_CANCER, dtype: int64
0    108
1     16
Name: LUNG_CANCER, dtype: int64
```

In [36]:
```
from sklearn.preprocessing import StandardScaler
```

In [37]:
```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [38]:
```
X_train
```

Out[38]:
```
array([[ 1.08465229,  1.12165139,  0.92195445, ...,  0.70997538,
         1.03858157, -1.22474487],
       [-0.92195445, -0.35350624,  0.92195445, ..., -1.40849954,
        -0.96285167,  0.81649658],
       [ 1.08465229, -1.09108506,  0.92195445, ..., -1.40849954,
         1.03858157,  0.81649658],
```

Predictive Model Implementation for 60% and 40% data

Logistic regression:

Here I am using logistic regressions ml algorithms, to check whether it gives how much as accuracy rate for lung cancer.

In [39]:
```
#  PROPOSED MODEL IMPLEMENTATION

#LOGISTIC REGRESSION:

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

lr = LogisticRegression()
lr.fit(X_train, Y_train)
y_pred = lr.predict(X_test)

lr_train_acc = accuracy_score(Y_train, lr.predict(X_train))
lr_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result of the experiment the logistic regression gives the training accuracy has 96 percent, and the testing accuracy has 88 percent.

```
print(f"Training Accuracy of Logistic Regression Model is {lr_train_acc}")
print(f"Test Accuracy of Logistic Regression Model is {lr_test_acc}")
```

```
Training Accuracy of Logistic Regression Model is 0.9675675675675676
Test Accuracy of Logistic Regression Model is 0.8870967741935484
```

```
In [40]: confusion_matrix(Y_test, y_pred)
```

```
Out[40]: array([[101,   7],
                [  7,   9]], dtype=int64)
```

```
In [41]: print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94       108
           1       0.56      0.56      0.56        16

    accuracy                           0.89       124
   macro avg       0.75      0.75      0.75       124
weighted avg       0.89      0.89      0.89       124
```

K- Nearest Neighbor:

Here I am using KNN Classification ml algorithm, to check whether it gives how much as accuracy rate for lung cancer.

```
In [42]: #K-NEAREST NEIGHBOUR CLASSIFIER:

         from sklearn.neighbors import KNeighborsClassifier

         knn = KNeighborsClassifier()
         knn.fit(X_train, Y_train)

         y_pred = knn.predict(X_test)

         knn_train_acc = accuracy_score(Y_train, knn.predict(X_train))
         knn_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training and testing dataset:

As a result of the experiment the k-Nearest Neighbor classifier gives the training accuracy has 93 percent, and the testing accuracy has 84 percent.

```
print(f"Training Accuracy of KNN Model is {knn_train_acc}")
print(f"Test Accuracy of KNN Model is {knn_test_acc}")
```

```
Training Accuracy of KNN Model is 0.9351351351351351
Test Accuracy of KNN Model is 0.8467741935483871
```

```
In [43]: confusion_matrix(Y_test, y_pred)
```

```
Out[43]: array([[100,   8],
                [ 11,   5]], dtype=int64)
```

```
In [44]: print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.90      0.93      0.91       108
           1       0.38      0.31      0.34        16

    accuracy                           0.85       124
   macro avg       0.64      0.62      0.63       124
weighted avg       0.83      0.85      0.84       124
```

Random Forest:

Here I am using Random Forest Classification ml algorithm, to check whether it gives how much as accuracy rate for lung cancer.

```
In [45]: # RANDOM FOREST CLASSIFIER

from sklearn.ensemble import RandomForestClassifier

rand_clf = RandomForestClassifier(criterion = 'gini', max_depth = 3, max_features = 'sqrt', min_samples_leaf = 2, min
rand_clf.fit(X_train, Y_train)

y_pred = rand_clf.predict(X_test)

rand_clf_train_acc = accuracy_score(Y_train, rand_clf.predict(X_train))
rand_clf_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training data and testing dataset:

As a result, the experiment the Random Forest algorithm gives the training accuracy has 89 percent, and the testing accuracy has 86 percent.

```
Training Accuracy of Random Forest Model is 0.918918918918919
Test Accuracy of Random Forest Model is 0.8629032258064516
```

```
In [46]: print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.96      0.92       108
           1       0.43      0.19      0.26        16

    accuracy                           0.86       124
   macro avg       0.66      0.58      0.59       124
weighted avg       0.83      0.86      0.84       124
```

Cat Boost:

Here I am using Cat Boost Gradient Classifier ml algorithm, to check whether it gives how much as accuracy rate of the lung cancer.

```
In [47]: from catboost import CatBoostClassifier

cat = CatBoostClassifier(iterations = 30, learning_rate = 0.1)
cat.fit(X_train, Y_train)

y_pred = cat.predict(X_test)
```

```
25:     learn: 0.2790629        total: 450ms    remaining: 69.2ms
26:     learn: 0.2734007        total: 452ms    remaining: 50.2ms
27:     learn: 0.2684866        total: 453ms    remaining: 32.4ms
28:     learn: 0.2639409        total: 455ms    remaining: 15.7ms
29:     learn: 0.2593087        total: 457ms    remaining: 0us
```

```
In [48]: cat_train_acc = accuracy_score(Y_train, cat.predict(X_train))
cat_test_acc = accuracy_score(Y_test, y_pred)
```

Experimental Results of training data and testing dataset:

As a solution, the experiment the Cat boost classifier model gives the training accuracy has 95 percent, and the testing accuracy has 84 percent.

```
print(f"Training Accuracy of Cat Boost Classifier Model is {cat_train_acc}")
print(f"Test Accuracy of Cat Boost Classifier Model is {cat_test_acc}")
```

```
Training Accuracy of Cat Boost Classifier Model is 0.9567567567567568
Test Accuracy of Cat Boost Classifier Model is 0.8467741935483871
```

```
In [49]: confusion_matrix(Y_test, y_pred)
```

```
Out[49]: array([[102,   6],
                [ 13,   3]], dtype=int64)
```

```
In [50]: print(classification_report(Y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.89	0.94	0.91	108
1	0.33	0.19	0.24	16
accuracy			0.85	124
macro avg	0.61	0.57	0.58	124
weighted avg	0.82	0.85	0.83	124

Decision Tree:

Here I am using Decision tree classifier ml algorithm, to check whether it gives how much as accuracy rate for lung cancer. Here we can see the diagrammatic represented of our dataset. We can easily understand with help of these representation.

```
In [51]: from sklearn.tree import DecisionTreeClassifier

         dtc = DecisionTreeClassifier()
         dtc.fit(X_train, Y_train)

         y_pred = dtc.predict(X_test)

         dtc_train_acc = accuracy_score(Y_train, dtc.predict(X_train))
         dtc_test_acc = accuracy_score(Y_test, y_pred)

         print(f"Training Accuracy of Decision Tree Model is {dtc_train_acc}")
         print(f"Test Accuracy of Decision Tree Model is {dtc_test_acc}")

         Training Accuracy of Decision Tree Model is 1.0
         Test Accuracy of Decision Tree Model is 0.8387096774193549
```

Experimental Results of training data and testing dataset:

As a result, the experiment the Decision tree classifier model gives the training accuracy has 100 percent, and the testing accuracy has 83 percent.

```
         print(f"Training Accuracy of Decision Tree Model is {dtc_train_acc}")
         print(f"Test Accuracy of Decision Tree Model is {dtc_test_acc}")

         Training Accuracy of Decision Tree Model is 1.0
         Test Accuracy of Decision Tree Model is 0.8387096774193549

In [52]: confusion_matrix(Y_test, y_pred)

Out[52]: array([[97, 11],
                [ 9,  7]], dtype=int64)

In [53]: print(classification_report(Y_test, y_pred))

                     precision    recall  f1-score   support

                  0       0.92      0.90      0.91       108
                  1       0.39      0.44      0.41        16

           accuracy                           0.84       124
          macro avg       0.65      0.67      0.66       124
       weighted avg       0.85      0.84      0.84       124
```

Result comparison of all models for 60%training:

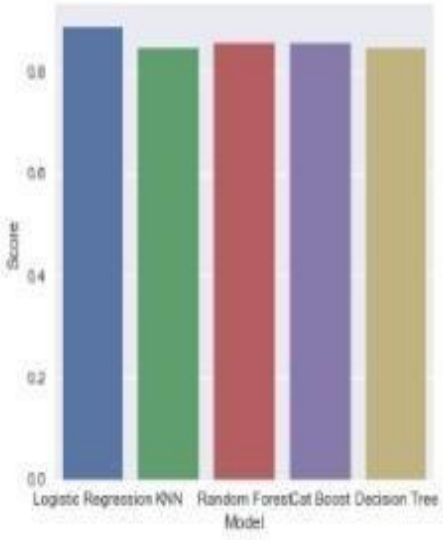
So as per the result comparison of all the five-algorithm logistic regression gives the accuracy has 88%, and KNN gives the accuracy has 85%, Random Forest gives the accuracy has 86%, Cat Boost gives the accuracy as 85% and the Decision tree gives the accuracy has 84%. So, by comparing the training accuracy and the testing accuracy of each algorithm with other algorithm, here also Logistic regression gives the good accuracy.

```
In [54]: models = ['Logistic Regression', 'KNN', 'Random Forest','Cat Boost', 'Decision Tree']
         scores = [lr_test_acc, knn_test_acc, rand_clf_test_acc,dtc_test_acc,cat_test_acc]

         models = pd.DataFrame({'Model' : models, 'Score' : scores})


         models.sort_values(by = 'Score', ascending = True)
```

Out[54]:

	Model	Score
1	KNN	0.846774
4	Decision Tree	0.846774
2	Random Forest	0.854839
3	Cat Boost	0.854839
0	Logistic Regression	0.887097

```
In [55]: plt.figure(figsize = (5, 5))

         sns.barplot(x = 'Model', y = 'Score', data = models)
         plt.show()
```



5.4 COMPARISON OF MODELS

The result for this lung cancer detecting model is based on comparing, all the proposed algorithms, which is going to provide the solution by consideration of the aspects. By the comparison of these Five implemented algorithm logistic regression provides the good result with accuracy of 92%. So, using logistic regression for web app implementation.

5.5 WEB APP IMPLEMENTATION

Here the Anaconda framework is used to provide a GUI. In that Visual Studio acts as a code editor for implementation. I used flask, a python web framework to create a web application, where html is used to design the website and python file is used to get the details and attributes of ourdata's so that if the user enters the value in the web, it will predict the required solution. Pickleused for serializing our model for that pickle dump () method is used to train our algorithms like(ML algorithms etc..). In our project we used Logistic Regression, because it provides good accuracy comparing with other algorithms. To predict the new entry value result I am using logistic regression algorithm which is stored in the pickle file for prediction. So here I created a web page in the form format using html and collected 15 attributes values has a integer value for predicting that the patient has the symptoms for lung cancer or not. The trained algorithm will find the result and provide the result in the same web page in string format.

APP CODE:

```
import numpy as np

import pandas as pd

from flask import Flask, request, render_template

import pickle

app = Flask(__name__)

model = pickle.load(open('model2.pkl', 'rb'))

@app.route('/')

def home():

  return render_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():

  input_features = [int(x) for x in request.form.values()]

  features_value = [np.array(input_features)]

  prediction=model.predict(features_value)

  return render_template('index.html', prediction_text='Patient has {} Lung
cancer'.format(prediction))

if__name___== "_main_":

  app.run(debug=True)
```

HTML CODE:

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
```

```
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

</head>

<body>
 <div class="login">
        <h1>Flower Class Prediction</h1>

    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('predict')}}"method="post">
        <input type="text" name="Sepal_Length" placeholder="Sepal_Length" required="required" />
      <input type="text" name="Sepal_Width" placeholder="Sepal_Width" required="required" />
                <input type="text" name="Petal_Length" placeholder="Petal_Length" required="required" />
                <input type="text" name="Petal_Width" placeholder="Petal_Width" required="required" />

      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
   </form>

  <br>
  <br>
  {{ prediction_text }}

 </div>



</body>
</html>
```

MODEL CODE:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import pickle
data = pd.read_csv("slc.csv")
print(data.head())

X= data[["AGE", "SMOKING" , "YELLOW_FINGERS",
"ANXIETY","PEER_PRESSURE","ALCOHOL
CONSUMING","COUGHING","SHORTNESS OF BREATH","SWALLOWING
DIFFICULTY","CHEST PAIN"]]
Y= data["LUNG_CANCER"]
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.3,random_state=42)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

classifier = RandomForestClassifier()
classifier.fit(X_train,Y_train)
pickle.dump(classifier,open("model2.pkl","wb"))
```

5.6 SCREENSHOT OF WEB APP

This is the web page which I created in order collect the 15 attributes value from the user for prediction of disease. If user enters that value and click the predict button, the python code fetches the value and found the result with the trained model and provide the solution for the use in the same bottom of the same web page.

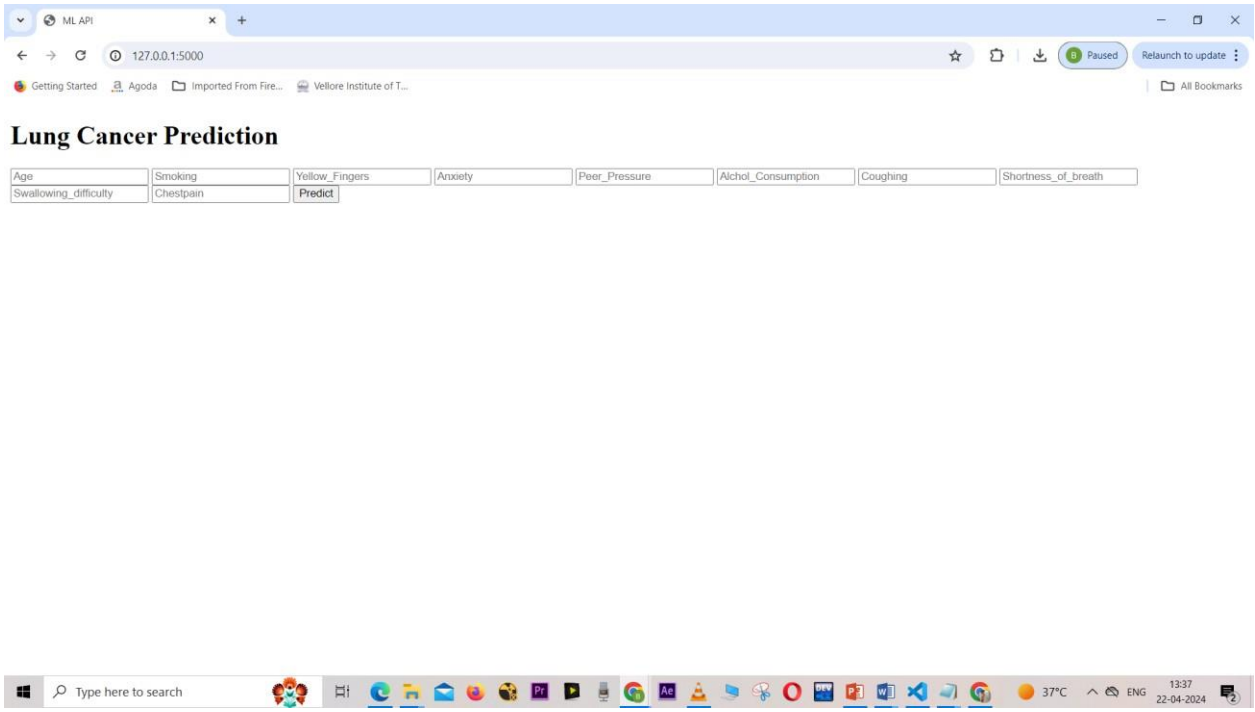


Figure 1

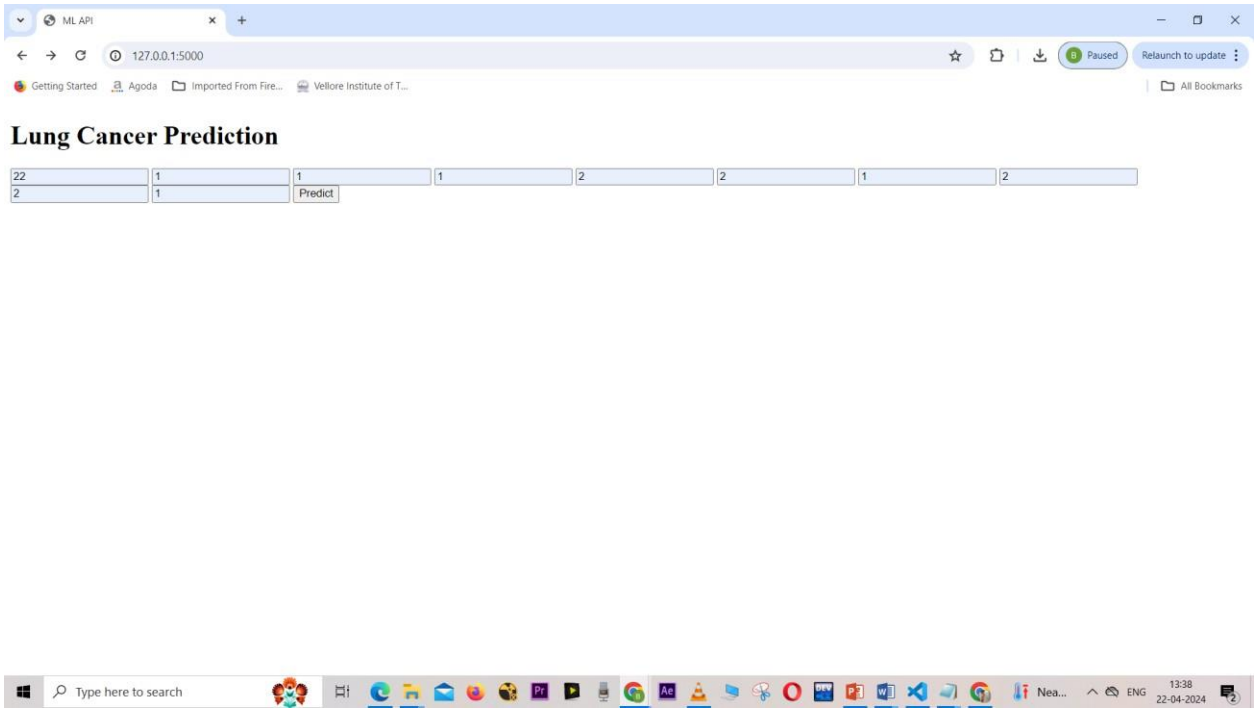


Figure 2

Chapter 6

Result and Discussion

6.1 RESULT OF THE PROJECT

As a result of the project, logistic regression is trained for result prediction. So here by gettingthe 15 attributes new value the trained model provides the test result as the patient has lung cancer or the patient don't have lung cancer. This is done with the help of the trained algorithm.Here we are not directly training the algorithm, the algorithm is trained and serialized has the pickle file. So, in the python code of our project, I am calling the pickle file and assign the test value to it for prediction. Based on the trained algorithm result it will also predict the test resultand provide the result for our assigned values also. Then with help of the python code, I am displaying the result in the web page.

The patient has lung cancer result screenshot:

Here the screen shot provides the input values and also the result for the values has the patientdon't have lung cancer.

New data inputs entry:

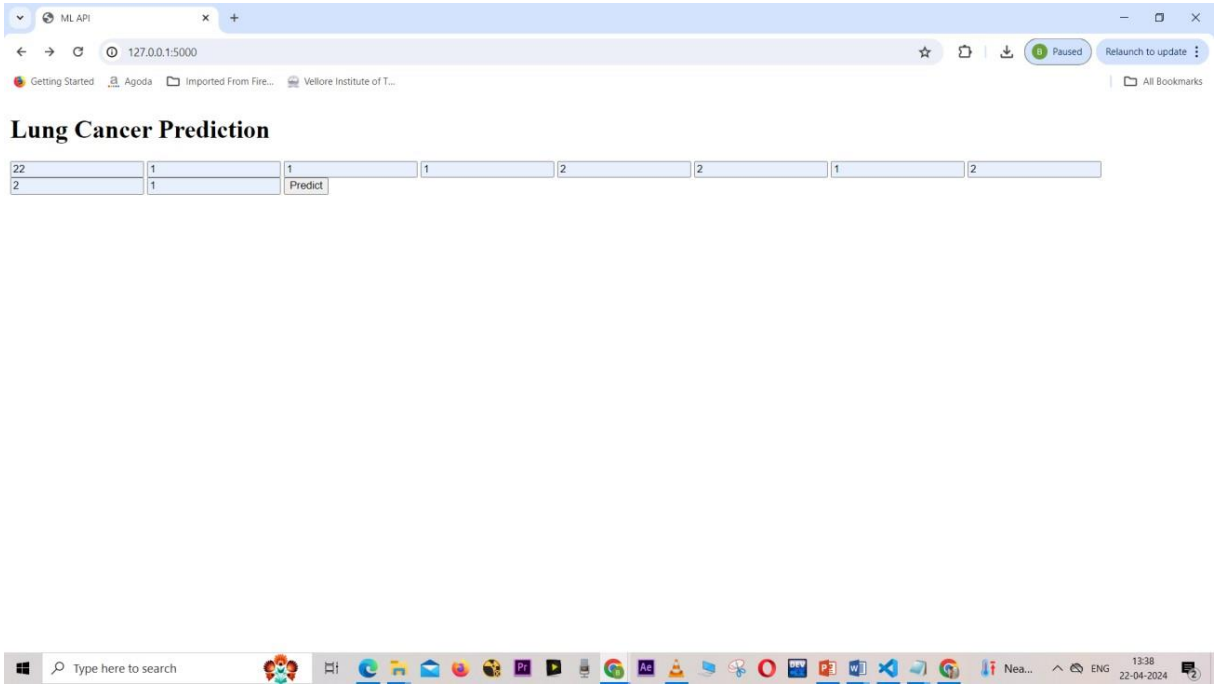


Figure 3

Result for the new values:

By getting the 15 attribute input values, the website provides the result has the patient has the lung cancer.

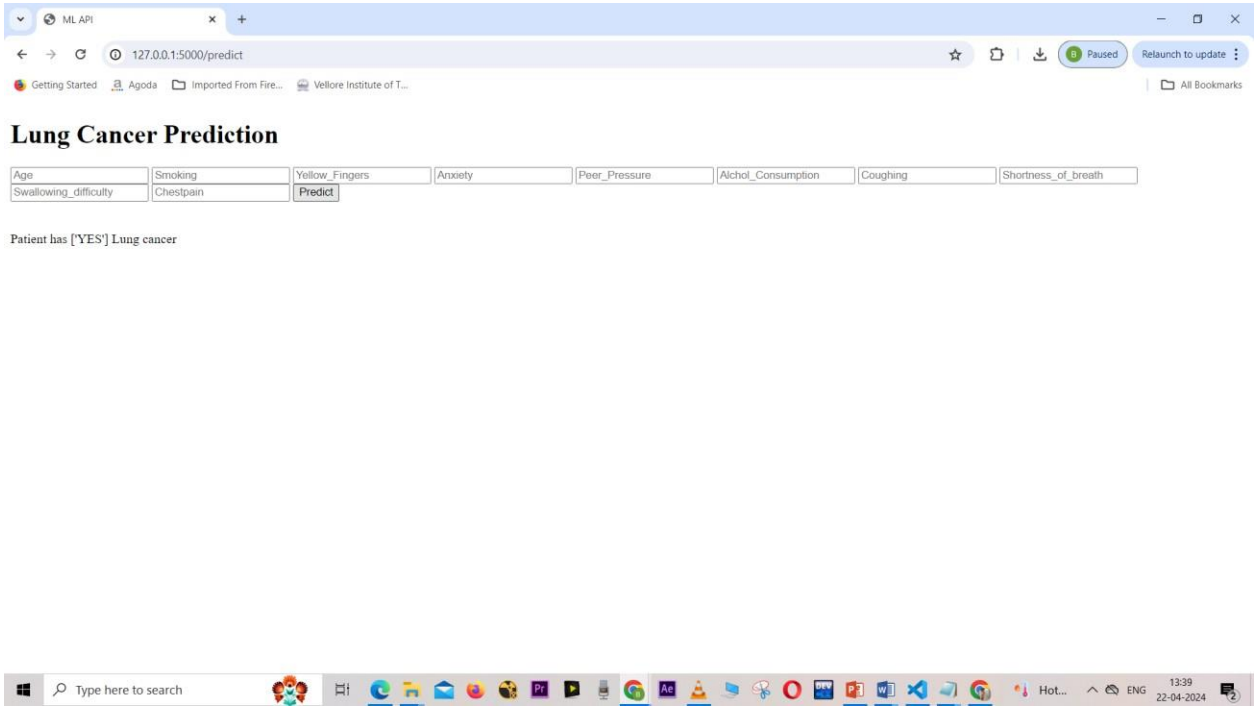


Figure 4

7.2 FRAMEWORK SCREEN SHOT:

For this project I am using Anaconda framework, in that I am using Jupyter Notebook for model implementation, Flask for web implementation, vs code for implementation of html and python for our website.

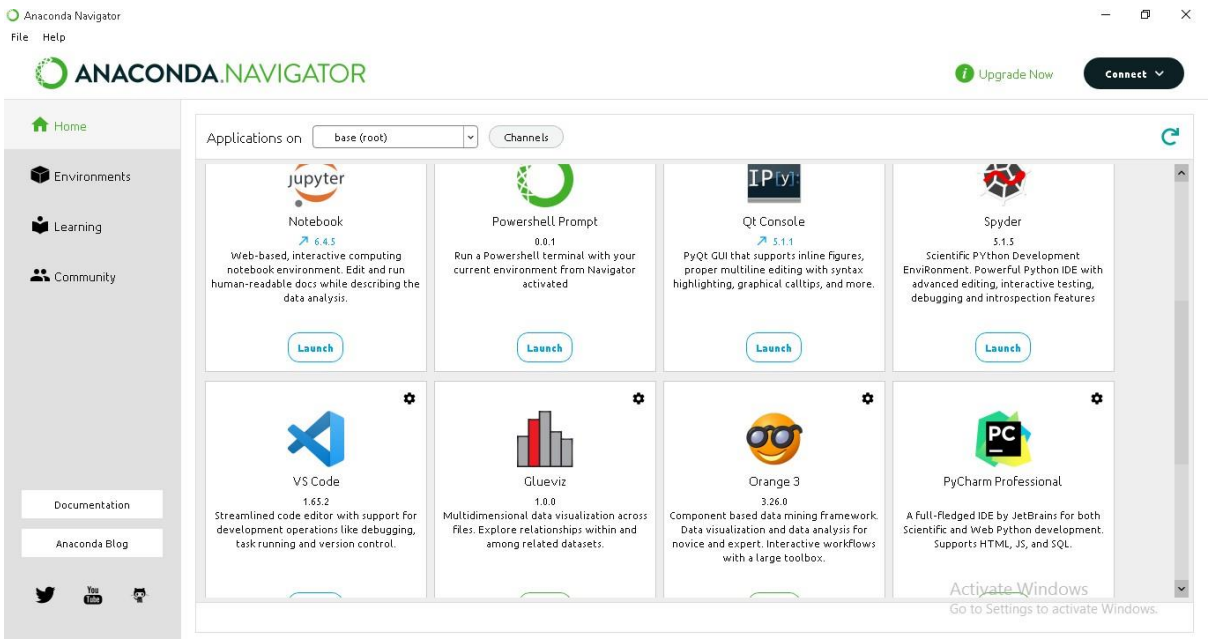


Figure 5

Vs code:

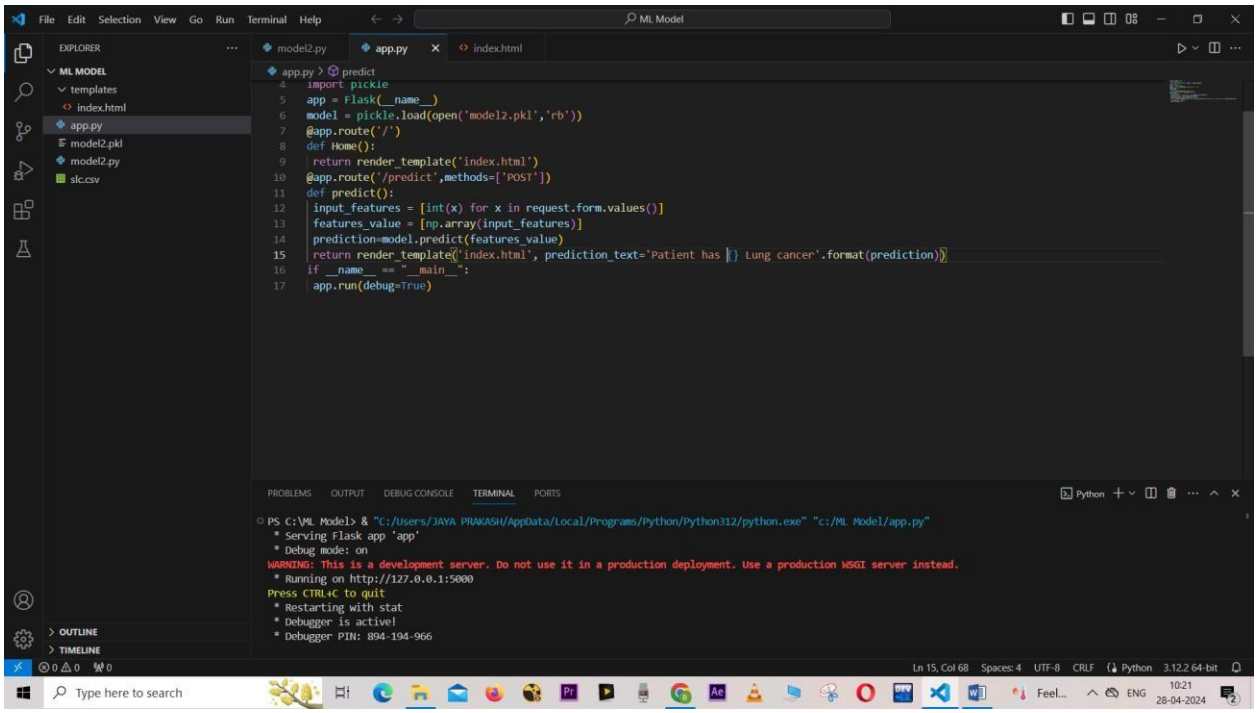


FIGURE 6

Chapter 7

Result & Discussion

7.1 CONCLUSION

This project is designed for the purpose to reduce the time for the prediction. As a result of my project, I compared more than five ML algorithms and checked which algorithm gives the best result in prediction for lung cancer. And I created a web application to check whether the patient has lung cancer or not by collecting the data from the patient. With the help of logistic regression and random forest algorithm the application predicts and providesthe results for the patients. This is done with the help of the trained algorithm. Here we are notdirectly training the algorithm, the algorithm is trained and serialized has the pickle file. So, in the python code of our project, I am calling the pickle file and assign the test value to it for prediction. Based on the trained algorithm result it will also predict the test result and provide the result for our assigned values also. Then with help of the python code, I am displaying the result in the web page.

7.2 FUTURE WORK

In future, we can compare more algorithms to check which algorithm is best suit for lung cancer prediction. Also, we can use another platform for the creation and deployment of the web application. In this project I used Anaconda platform for implementation. In my project mostly I am using classification type of supervised learning algorithms, further we can also implement with unsupervised learning also. we can also use advanced ML type of algorithms or deep learning or also we can use neural networks, fuzzy logics for further implementation and for prediction.

7.3 REFERENCES

[1] Naveen N C, Pradeep KR (2018), “Lung cancer prediction based on ML algorithms for the Healthcare and Analytics “, ScienceDirect, Bengaluru-560060.

[2] Dakhaz Mustafa Abdullah, Adnan Mohsins Abdulazeezz, Amira Bibo Sallow (2020), “Lung cancer detection and Classification based on the Correlation Selection methods Using the ML Techniques”, Qubahan Academic Journal- Duhok, Iraq.

[3] Saadaldeen Rashid Ahmed, Ammar Mhana , Haider Rasheed Abdulshaheed (2019), “Lung cancer classifications using the supervised ML algorithms on the multi-dimensional data set and data mining”, ISSN, Altinbas University, Istanbul, Turkey.

[4] Carol M. Jim, George Dimitoglou,, and JamesA. Adams (2015), “Comparison of the decision tree and a NaiveBayes Classifier for the detecction of Lung Cancer Survivability”, ISSN – International Journal of Applied Engineering and Research, 0973-4562.

[5] Maxim D Podolsky, Anton A Barchuk, Vladimir I Kuznetcov, Natalia F Gusarova , Vadim S Gaidukov, Segrey A Tarakanov (2016), “Evaluation of ML Algorithm Utilizations of the Lungs Cancer & Classification Based on the Gene Expressions & Levels”, Asian pacific journal,835-838 .

[6] Pati J (2019),“Gene Expression Analysis for Early Lung Cancer detection Using Machine Learning Techniques”, January, IEEE Access, 2169-3536.

[7] Wafaa K. Shams, Zaw Z. Htike (2017), “Lung Cancer Prediction Using Gene Expression Profiling and Machine Learning” IEEE Access, 2169-3536.

[8] Melanie Hilario, Alexandros Kalousis, Markus Müller, Christian Pellegrini (2003), "Machine learning approaches to lung cancer prediction", Proteomics 2003, 1716-1719, Weinheim.

[9] Manju B R, Athira V , Athul Rajendran (2020), "Efficient multi-leves of lung cancer detection model using SVM classifier", IOP Publishing Limited, Canada.

[10] H Bharathi, T S Arulananth (2017), "Review of lung cancer detection system using the Data Mining Techniques", November, ISSN – Internationals & Journals of the Applied Engineerings & Research, 0973-4562.

[11] Asha R B, Suresh Kumar K R (2021), "Lung Cancer Prediction using (ANN) artificial neurals & network", IEEE Access, 2169-3536.

[12] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, "Lung Cancer Prediction -Machine LearningMethods", IJCSMC, Vol-4, January 2019.

[13] T Sowmiya, M New Beginn, M N Gopi (2014), "Optimizaations of Lungs Cancers using the data mining techniques", Researcher gate, March 2014.

[14] Ada, Rajneet Kaur (2013), "Early Detecting and Predicting of Lung Cancer Survival using Neural Network Classifier", ISSN – International Journal of Applied Engineering and Research, 0973-4562.

[15] R Thomas Robinson (2014)., "Prediction of Lung Cancer Survival", IJCSMC, Vol-4, January 2014.

[16] Almas Pathan, and Bairu.K.saptalkar (2015), "Detection and Classification of the Lung Cancers Using Artificial Neural Network", IEEE Access, January, Vol-2.

[17] Dasu Ravi Vaman Prasad (2013), “Lung cancer prediction using images processing techniques”, IJSCMC, Vol-4.

[18] S Vishukumar, Pavan Shrivastavab, K Patela (2018), “Lung A Cancer Classifications Using Images Processing”, September, ISSN Publisher.

19] Fatma Taher, Hussain Al-Ahmad, Naoufel Werghi, Rachid Sammouda (2017)., “Lung Cancer Detection Using the help of ANN and the Fuzzy Clustering Methods,” IEEE Access, April, Vol-2.

[20] O Günaydin, M Günay and Ö Şengel (2019), “Comparison of Lung Cancer Detection Algorithms”, ISSN – International Journal of Applied Engineering and Research, 0973-4562.

[21] Zaw Zaw Hitke, IbrahimA, (2014), “Recurrence Cancer Prediction using ML algorithms, International Journals of Computation Science & Information Technlogy (IJCSY), Vol-2.

LUNG CANCER PREDICTION 1

ORIGINALITY REPORT

10%	8%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	gitlab.sliit.lk Internet Source	1%
2	www.ijfrcsce.org Internet Source	1%
3	huggingface.co Internet Source	1%
4	ijrpr.com Internet Source	1%
5	begen.github.io Internet Source	1%
6	www.ijraset.com Internet Source	1%
7	cse.anits.edu.in Internet Source	<1%
8	www.tutorialspoint.com Internet Source	<1%
9	Vaibhav Verdhan. "Supervised Learning with Python", Springer Science and Business Media LLC, 2020	<1%

Publication

<1%

10 dokumen.pub
Internet Source

<1%

11 Neapolitan, Richard, Curt M. Horvath, and Xia Jiang. "Pan-cancer analysis of TCGA data reveals notable signaling pathways", BMC Cancer, 2015.
Publication

<1%

12 Sadaqat Ali, Asifa Noreen, Adeem Qamar, Imran Zafar et al. "Amomum subulatum: A treasure trove of anti-cancer compounds targeting TP53 protein using in vitro and in silico techniques", Frontiers in Chemistry, 2023
Publication

<1%

13 course.ccs.neu.edu
Internet Source

<1%

14 www.jica.go.jp
Internet Source

<1%

15 Jayadeep Pati. "Gene Expression Analysis for Early Lung Cancer Prediction using Machine Learning Techniques: An Eco-Genomics Approach", IEEE Access, 2018
Publication

<1%

16 Trailokya Raj Ojha, Menuka Maharjan. "Machine-Learning Based Prediction of Lung Cancer", SCITECH Nepal, 2023

	Publication	
17	medium.com Internet Source	<1%
18	www.coursehero.com Internet Source	<1%
19	etd.aau.edu.et Internet Source	<1%
20	www.frontiersin.org Internet Source	<1%
21	www.javatpoint.com Internet Source	<1%
22	Adel Mellit, Soteris Kalogirou. "Artificial intelligence techniques: Machine learning and deep learning algorithms", Elsevier BV, 2022 Publication	<1%
23	sumtjnhnmkkmnhnjtmus.stmra.com Internet Source	<1%
24	Bibin Kurian, Ranjith Liyanapathirana. "Chapter 1 Machine Learning Techniques for Structural Health Monitoring", Springer Science and Business Media LLC, 2020 Publication	<1%
25	www.oncology.scientexconference.com Internet Source	
26	effectivehealthcare.ahrq.gov Internet Source	

		<1%
27	spiegato.com Internet Source	<1%
28	www.cs.mta.ac.il Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		