One Year Life Expectancy Post Thoracic Surgery Using Watson Studio Machine Learning

1.INTRODUCTION

Lung cancer is the leading cause of cancer-related deaths in the world. In the United States, lung cancer claims more lives every year than colon cancer, prostate cancer, and breast cancer combined.

The American Cancer Society's estimates for lung cancer in the United States for 2018 are:

- About 234,030 new cases of lung cancer (121,680 in men and 112,350 in women)
- About 154,050 deaths from lung cancer (83,550 in men and 70,500 in women)

Despite the very serious prognosis (outlook) of lung cancer, some people with earlier-stage cancers are cured. More than 430,000 people alive today have been diagnosed with lung cancer at some point. The data is dedicated to classification problems related to the post-operative life expectancy in lung cancer patients: class 1 - death within one year after surgery, class 2 - survival.

We will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration and IBM deployment.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

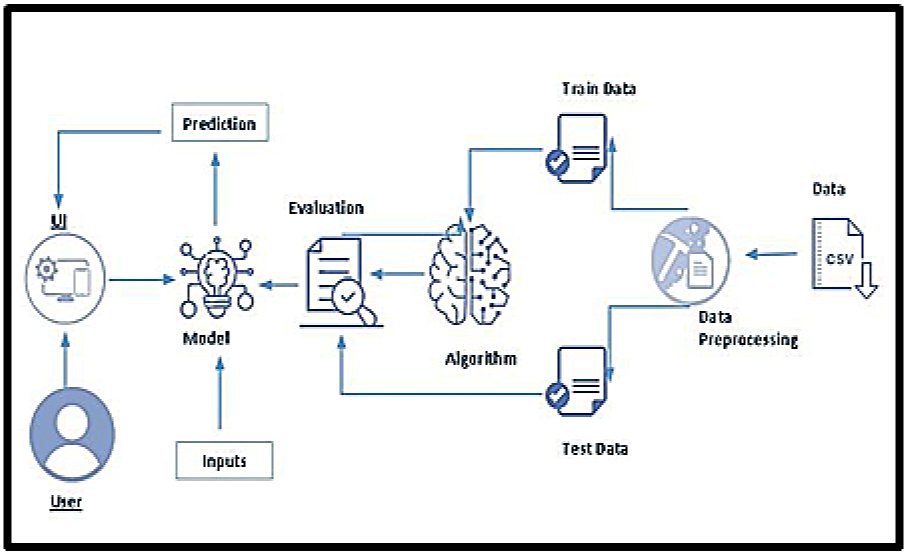
Operative mortality rates have been a topic of great interest among surgeons, patients, lawyers, and health policy administrators. Postoperative respiratory complications are the most common fatality following any type of thoracic surgery. The exact incidence is most contingent upon the preoperative health and lung function of the patient, and we would like to explore and understand how those conditions can drive these complications. One particular metric that has been used to quantify mortality rates in the past has been the thirty-day mortality rate. This metric, however, may not be entirely comprehensive because many patients die shortly after this time period or become very weak, having to be taken to another facility before passing away there. As a result, many of these deaths are severely underreported.

2.2 Proposed Solution

The scope of our project is to examine the mortality of patients within a full year after the surgery. More specifically, we are examining the underlying health factors of patients that could potentially be a powerful predictor for surgically related deaths. As mentioned, our feature set includes both continuous and classification data regarding to the patient's health conditions at the time of the surgery. Each patient has 16 variables associated with them Some of the continuous data includes a patients' forced vital capacity, the maximum volume their lungs exhaled, size of original tumor, and age at surgery. In addition we have several classification features such as presence of pain before surgery, haemoptysis before surgery, cough before surgery, whether the patient is a smoker, whether the patient has asthma, and a few others. The classification predicts whether the patient survived the following year long period.

3.THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware/Software designing

To complete this project, you must required following software's, concepts and packages

1. **Anaconda navigator and pharm:**

   a. download anaconda navigator

2. **Python packages:**
   a. Open anaconda prompt as administrator
   b. Type "pip install numpy" and click enter.
   c. Type "pip install pandas" and click enter.
   d. Type "pip install scikit-learn" and click enter.
   e. Type"pip install matplotlib" and click enter.
   f. Type"pip install scipy" and click enter.
   g. Type"pip install pickle-mixin" and click enter.
   h. Type"pip install seaborn" and click enter.
   i. Type "pip install Flask" and click enter.

4.EXPERIMENTAL INVESTIGATIONS

1. Know fundamental concepts and techniques used for machine learning.
2. Gain a broad understanding about data.
3. Have knowledge on pre-processing the data/transformation techniques on outlier and some visualization concepts.

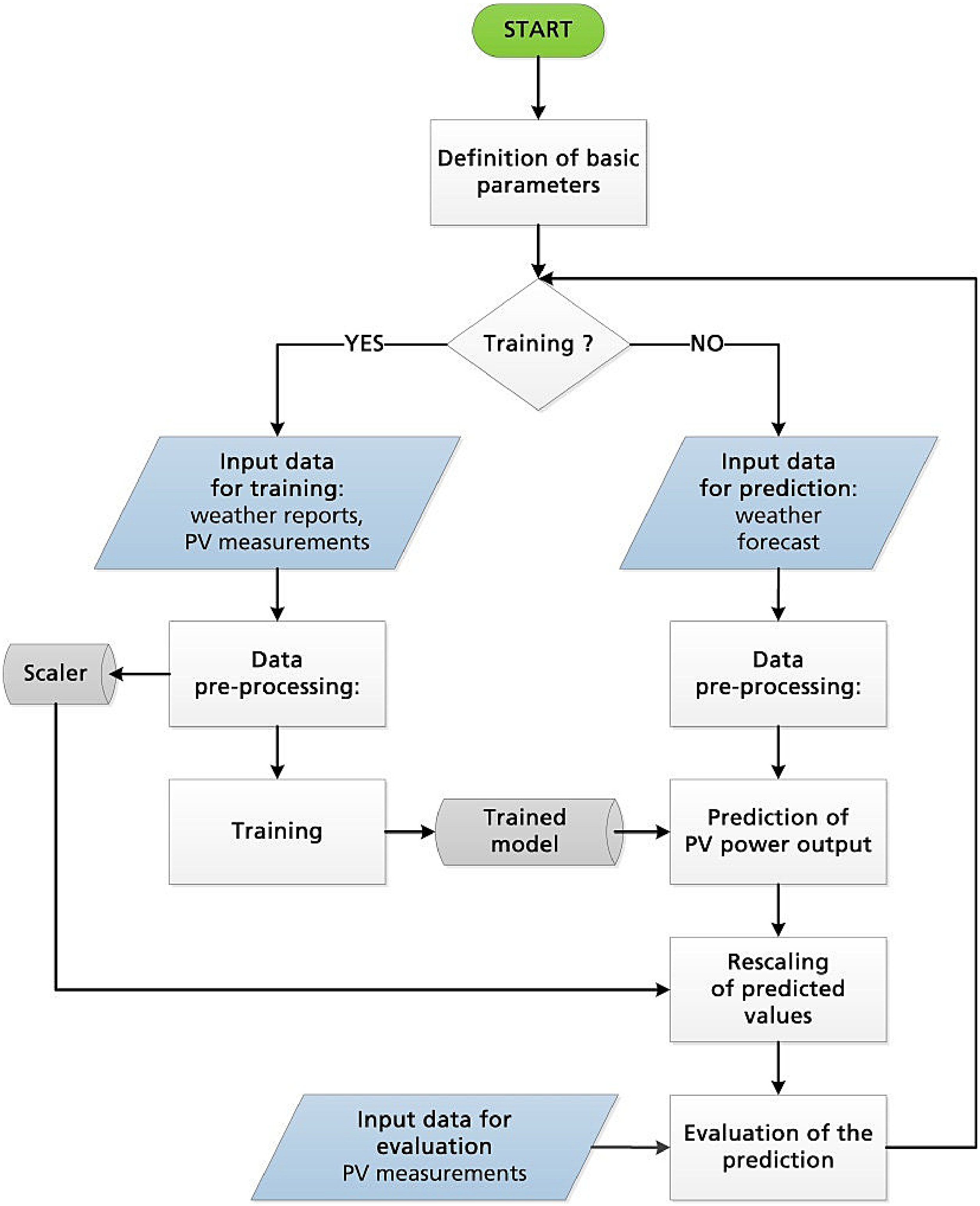
Project Flow:

1. User interacts with the UI to enter the input.
2. Entered input is analyzed by the model which is integrated.
3. Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

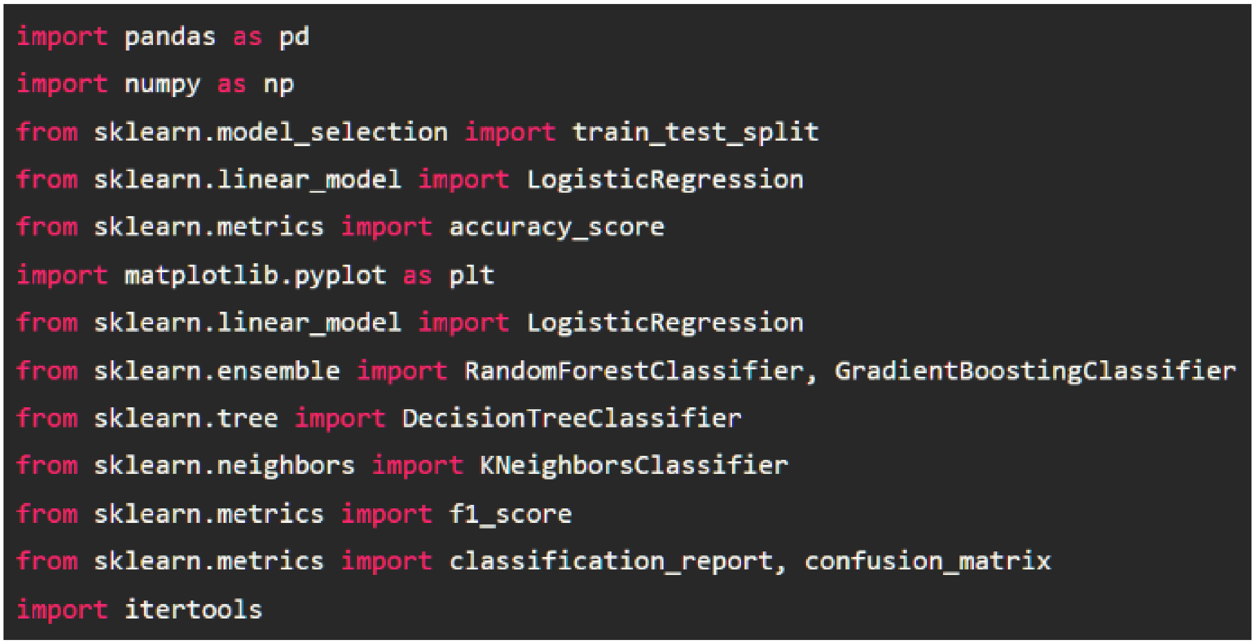
1. Data collection
   a. Collect the dataset or create the dataset
2. Visualizing and analyzing data
   a. Univariate analysis
   b. Bivariate analysis
   c. Multivariate analysis
   d. Descriptive analysis
3. Data pre-processing
   a. Checking for null values
   b. Handling outlier
   c. Handling categorical data
   d. Splitting data into train and test

5. FLOWCHART


START
Definition of basic parameters
Training ?
YES
NO
Input data for training: weather reports, PV measurements
Input data for prediction: weather forecast
Scaler
Data pre-processing:
Data pre-processing:
Training
Trained model
Prediction of PV power output
Rescaling of predicted values
Input data for evaluation PV measurements
Evaluation of the prediction

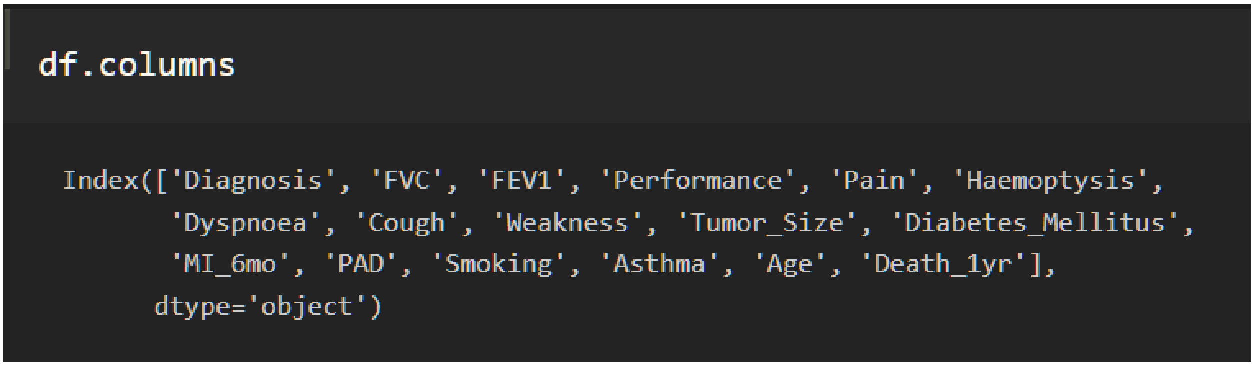
6.RESULTS

Activity 1: Importing the libraries



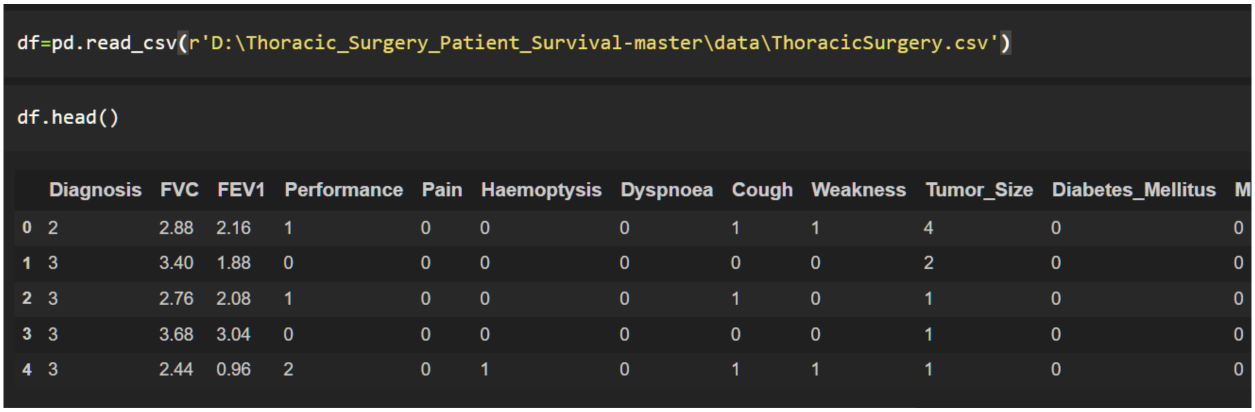
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report, confusion_matrix
import itertools
```

Activity 2: Read the Dataset



```
df.columns
```

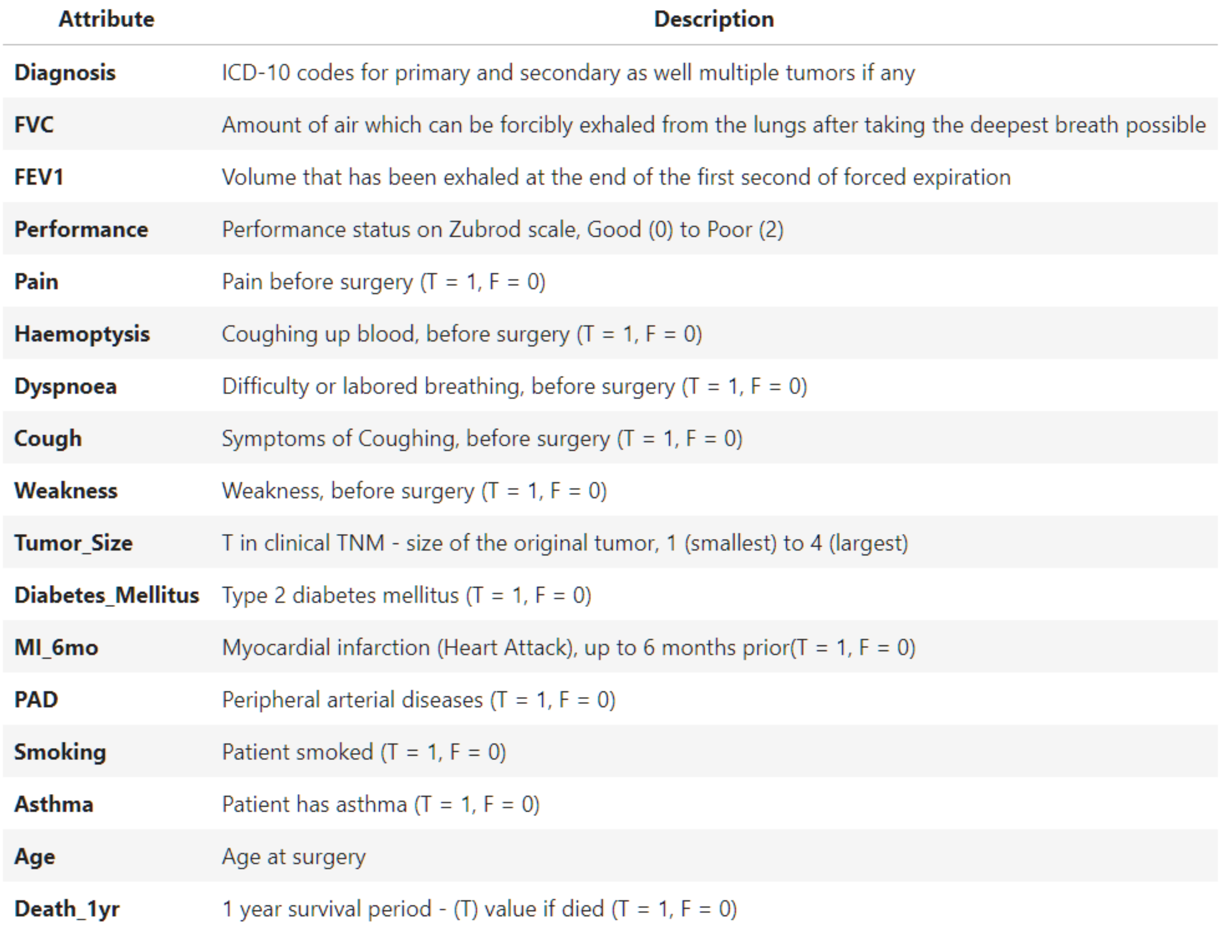
```
Index(['Diagnosis', 'FVC', 'FEV1', 'Performance', 'Pain', 'Haemoptysis',
       'Dyspnoea', 'Cough', 'Weakness', 'Tumor_Size', 'Diabetes_Mellitus',
       'MI_6mo', 'PAD', 'Smoking', 'Asthma', 'Age', 'Death_1yr'],
      dtype='object')
```



```
df=pd.read_csv(r'D:\Thoracic_Surgery_Patient_Survival-master\data\ThoracicSurgery.csv')
```

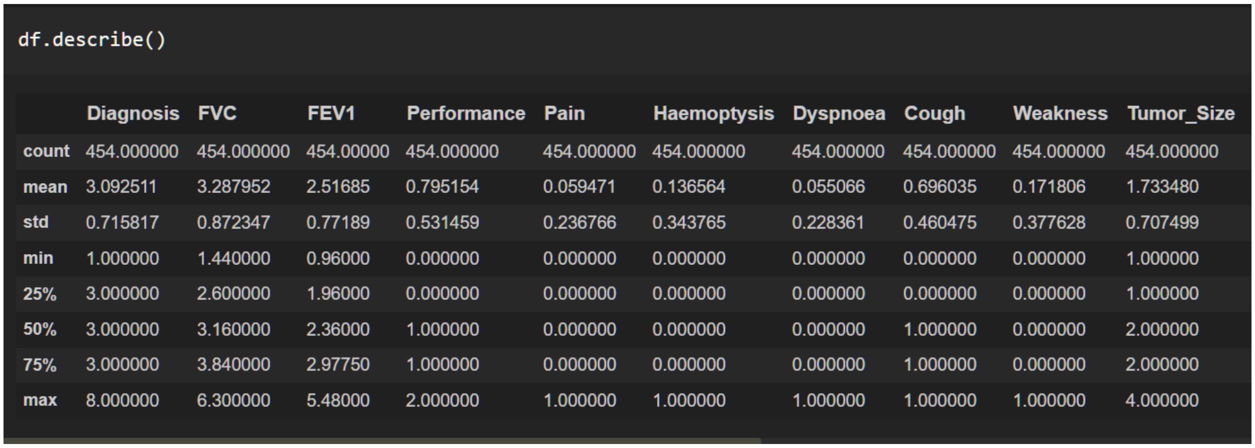
```
df.head()
```

	Diagnosis	FVC	FEV1	Performance	Pain	Haemoptysis	Dyspnoea	Cough	Weakness	Tumor_Size	Diabetes_Mellitus	M
0	2	2.88	2.16	1	0	0	0	1	1	4	0	0
1	3	3.40	1.88	0	0	0	0	0	0	2	0	0
2	3	2.76	2.08	1	0	0	0	1	0	1	0	0
3	3	3.68	3.04	0	0	0	0	0	0	1	0	0
4	3	2.44	0.96	2	0	1	0	1	1	1	0	0



Attribute	Description
Diagnosis	ICD-10 codes for primary and secondary as well multiple tumors if any
FVC	Amount of air which can be forcibly exhaled from the lungs after taking the deepest breath possible
FEV1	Volume that has been exhaled at the end of the first second of forced expiration
Performance	Performance status on Zubrod scale, Good (0) to Poor (2)
Pain	Pain before surgery (T = 1, F = 0)
Haemoptysis	Coughing up blood, before surgery (T = 1, F = 0)
Dyspnoea	Difficulty or labored breathing, before surgery (T = 1, F = 0)
Cough	Symptoms of Coughing, before surgery (T = 1, F = 0)
Weakness	Weakness, before surgery (T = 1, F = 0)
Tumor_Size	T in clinical TNM - size of the original tumor, 1 (smallest) to 4 (largest)
Diabetes_Mellitus	Type 2 diabetes mellitus (T = 1, F = 0)
MI_6mo	Myocardial infarction (Heart Attack), up to 6 months prior(T = 1, F = 0)
PAD	Peripheral arterial diseases (T = 1, F = 0)
Smoking	Patient smoked (T = 1, F = 0)
Asthma	Patient has asthma (T = 1, F = 0)
Age	Age at surgery
Death_1yr	1 year survival period - (T) value if died (T = 1, F = 0)

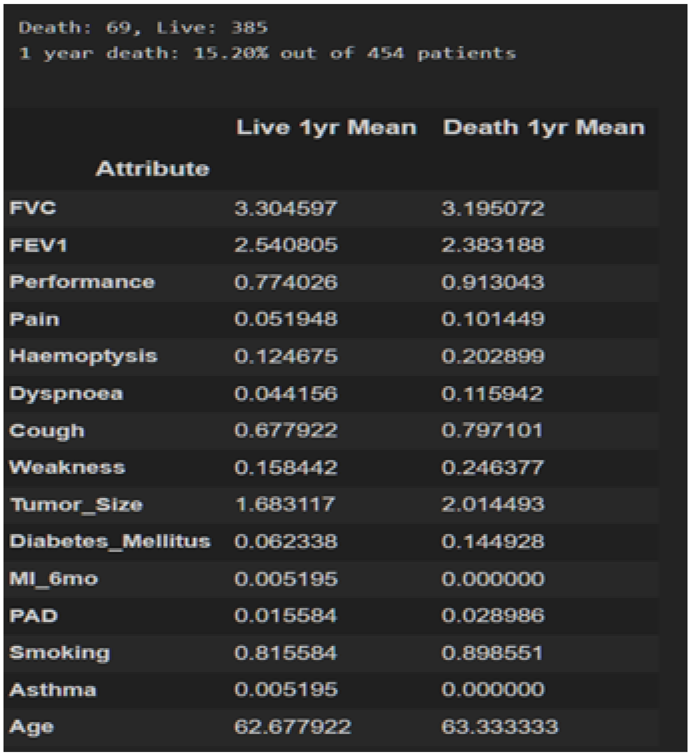
Activity 3: Descriptive Analysis



```
df.describe()
```

	Diagnosis	FVC	FEV1	Performance	Pain	Haemoptysis	Dyspnoea	Cough	Weakness	Tumor_Size
count	454.000000	454.000000	454.00000	454.000000	454.000000	454.000000	454.000000	454.000000	454.000000	454.000000
mean	3.092511	3.287952	2.51685	0.795154	0.059471	0.136564	0.055066	0.696035	0.171806	1.733480
std	0.715817	0.872347	0.77189	0.531459	0.236766	0.343765	0.228361	0.460475	0.377628	0.707499
min	1.000000	1.440000	0.96000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	3.000000	2.600000	1.96000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
50%	3.000000	3.160000	2.36000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	2.000000
75%	3.000000	3.840000	2.97750	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	2.000000
max	8.000000	6.300000	5.48000	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	4.000000

Activity 4: Exploratory Data Analysis



```
Death: 69, Live: 385
1 year death: 15.20% out of 454 patients
```

	Live 1yr Mean	Death 1yr Mean
Attribute		
FVC	3.304597	3.195072
FEV1	2.540805	2.383188
Performance	0.774026	0.913043
Pain	0.051948	0.101449
Haemoptysis	0.124675	0.202899
Dyspnoea	0.044156	0.115942
Cough	0.677922	0.797101
Weakness	0.158442	0.246377
Tumor_Size	1.683117	2.014493
Diabetes_Mellitus	0.062338	0.144928
MI_6mo	0.005195	0.000000
PAD	0.015584	0.028986
Smoking	0.815584	0.898551
Asthma	0.005195	0.000000
Age	62.677922	63.333333

Shows the prediction

6 : Data Pre-Processing

Checking For Null Values

Let’s find the shape of our dataset first, To find the shape of our data, df.shape method is used. To find the data type, df.info() function is used.

For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that there are no null values present in our dataset. So we can skip handling of missing values step.

Drop Unwanted Features

drop() is used to drop specified labels from rows or columns.

Remove rows or columns by specifying label names and corresponding axis, or by specifying directly index or column names.

We are building the model to predict the Life Expectancy FVC are very less related to the dependent variable. so if we remove this column the accuracy won't be affected that much.

Splitting Data Into Test And Train

Now let’s split the Dataset into train and test sets.

Changes: first split the dataset into x and y and then split the data set.

Here x and y variables are created. On the x variable, df is passed by dropping the target variable. And on y target variable is passed. For splitting training and testing data, we are using the train_test_split() function from sklearn. As parameters, we are passing x, y, test_size, random_state.

Here We can check the shape of the x_train,x_test,y_train & y_test data shape.

Feature Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data. Standard scaler() is initialized. Independent training data is passed in the fit_transform() method and independent test data is passed in the transform() function.

7 : Model Building

Decision Tree Model

A function named decision tree is created and train and test data are passed as the parameters. Inside the function, the DecisionTreeClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in the new variable. For evaluating the model, a confusion matrix and classification report is done.

Random Forest Model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, the RandomForestClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in the new variable. For evaluating the model, a confusion matrix and classification report are done.

KNN Model

A function named KNN is created and train and test data are passed as the parameters. Inside the function, the KNeighborsClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with the .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

Xgboost Model

A function named xgboost is created and train and test data are passed as the parameters. Inside the function, GradientBoostingClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, confusion matrix and classification report is done.

Compare The Models

For comparing the above four models compare model function is defined.

After calling the function, the results of models are displayed as output. From the four model random forest and KNeighbors is performing well. From the below image, We can see the accuracy of the model. Random forest model have 88% accuracy & KNeighbors model have 86% accuracy. In confusion matrix we have check the results. Training time of KNeighbors is faster than random forest. In such case we have to select KNeighbors model (time saving & cost wise profitable). But, here random forest is selected and evaluated with cross validation. Additionally, we can tune the model with hyper parameter tuning techniques.

Evaluating Performance Of The Model And Saving The Model

From sklearn, cross_val_score is used to evaluate the score of the model. On the parameters, we have given rf (model name), x, y, cv (as 5 folds). Our model is performing well. So, we are saving the model by pickle.dump().

Save The Model

After building the model we have to save the model.

Import Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

7 : Application Building

we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building serverside script

For this project create three HTML files namely

- index.html
- form.html
- result.html

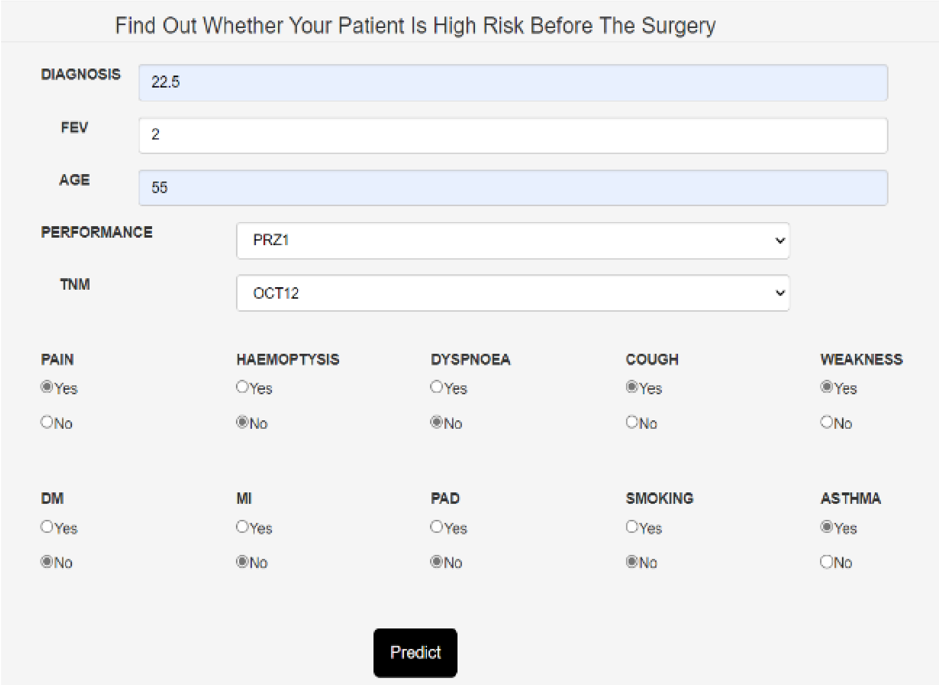
We have to build the python code to run the application.

At last we have to train this model in the IBM.

8: Result

8.1 : Local Deployment Result

Input1



Find Out Whether Your Patient Is High Risk Before The Surgery

DIAGNOSIS 22.5

FEV 2

AGE 55

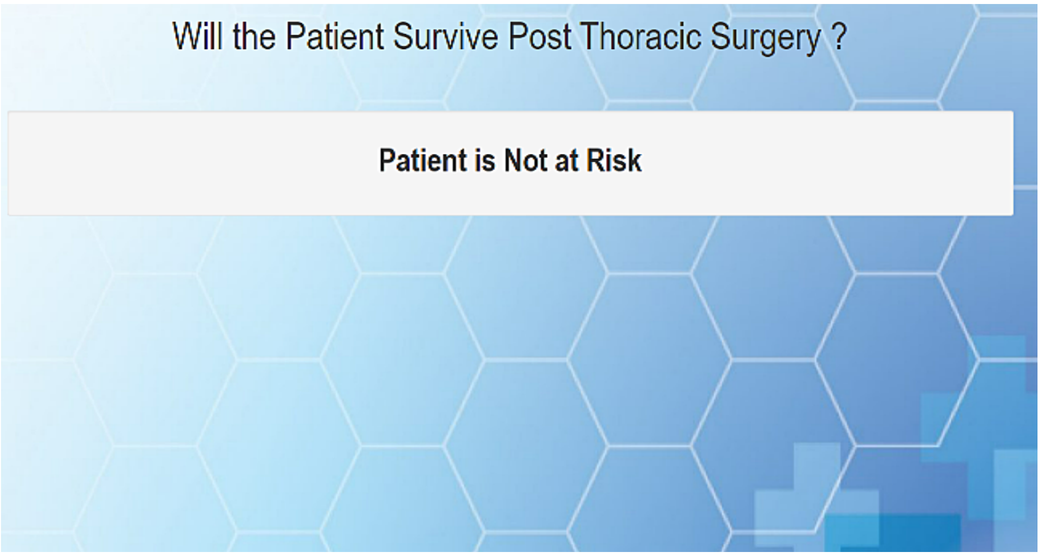
PERFORMANCE PRZ1

TNM OCT12

PAIN	HAEMOPTYSIS	DYSPNOEA	COUGH	WEAKNESS
●Yes	○Yes	○Yes	●Yes	●Yes
○No	●No	●No	○No	○No

DM	MI	PAD	SMOKING	ASTHMA
○Yes	○Yes	○Yes	○Yes	●Yes
●No	●No	●No	●No	○No

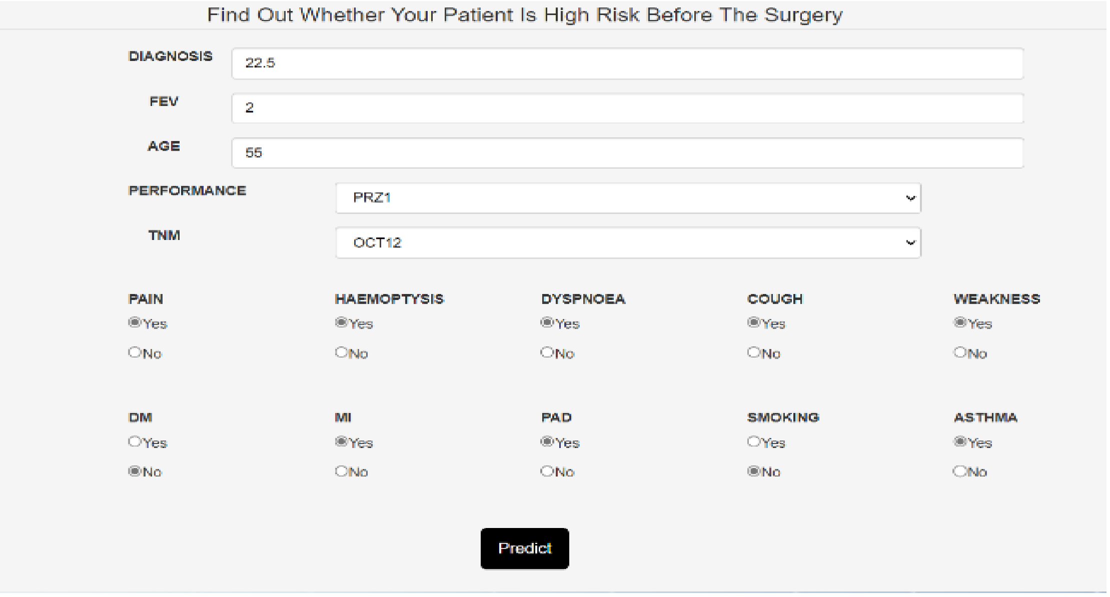
Output1



Will the Patient Survive Post Thoracic Surgery ?

Patient is Not at Risk

Input2



Find Out Whether Your Patient Is High Risk Before The Surgery

DIAGNOSIS 22.5

FEV 2

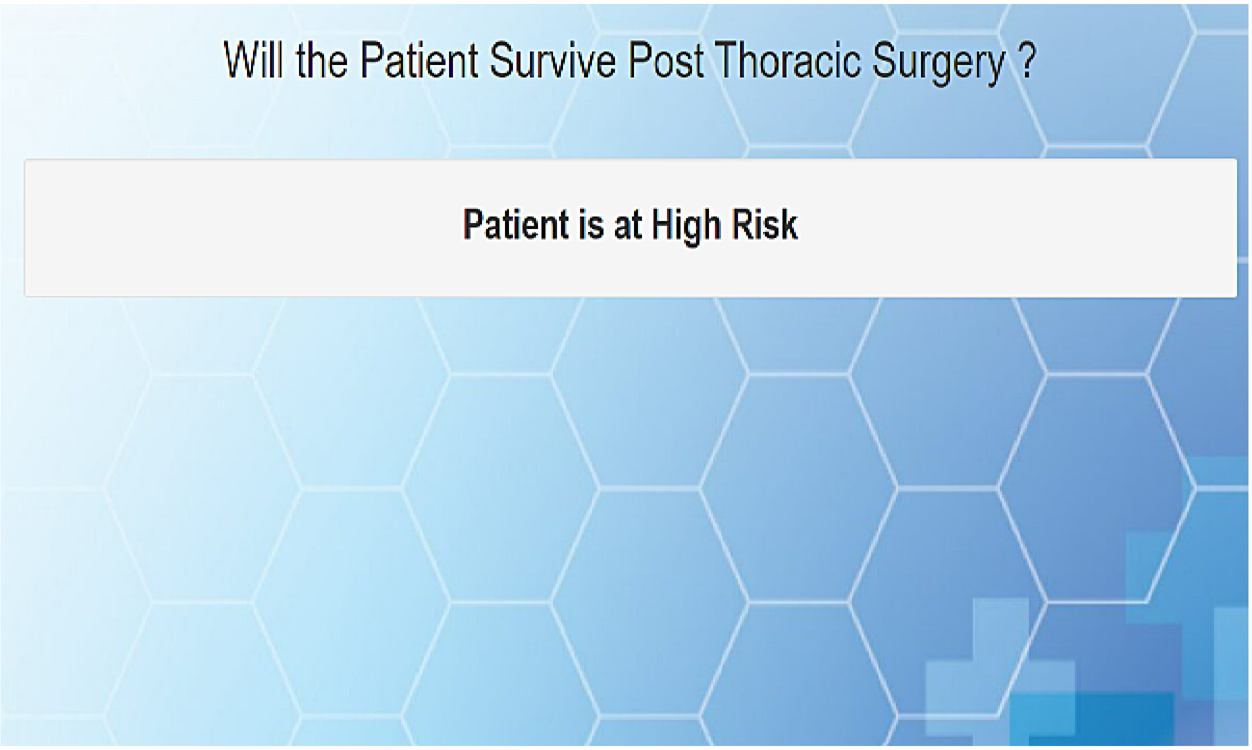
AGE 55

PERFORMANCE PRZ1

TNM OCT12

PAIN	HAEMOPTYSIS	DYSPNOEA	COUGH	WEAKNESS
●Yes	●Yes	●Yes	●Yes	●Yes
○No	○No	○No	○No	○No
DM	MI	PAD	SMOKING	ASTHMA
○Yes	●Yes	●Yes	○Yes	●Yes
●No	○No	○No	●No	○No

Output2



Will the Patient Survive Post Thoracic Surgery ?

Patient is at High Risk

10.Conclusion

Lung cancer is one of the challenging problems in medical field due to structure of cancer cells. Therefore, the proper medication has to be given to the patient for increasing the survival chances of the patient. Once the cancer is detected, the Thoracic Surgery is one of the best treatment options for the diagnosis of Lung Cancer. The project involves the analysis of the patient’s dataset who underwent Thoracic Surgery and an attempt is made to model a classifier that will predict the survival of the patient post the surgery. The dataset will be trained using four Supervised The Algorithms of Machine Learning that are Decision Tree, Random Forest, KNN, Xgboost. Among the four algorithms used, its observed that all the algorithms gives the highest accuracy of 91% compared to the other algorithms data in the future to analyse the system.