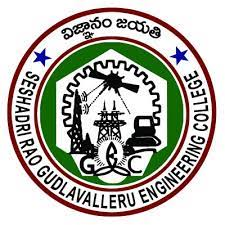
**Seshadri Rao Gudlavalleru Engineering College**

****

**Project report**

****

****

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

Submitted by :

Gudivada Janardhan

Roll Number : 19481A0470

Mentor Name

Srikanth Kummari



**Contents**

1. Abstract

2. Technical Architecture

3. Project Flow

3.1 : Data Collection

3.2 : Visualizing and Analyzing data

3.3 : Data Pre-Processing

3.4 : Model Building

3.5 : Application Building

4. Result

4.1 : Local Deployment Result

4.2 : IBM Deployment Result

5. Conclusion

****

**1. Abstract**

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:

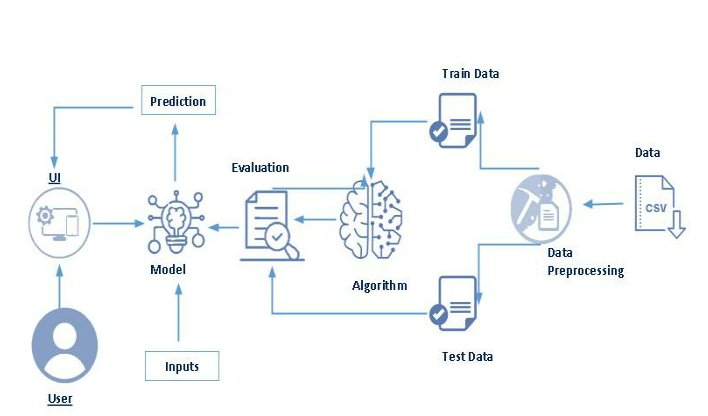
* About 234,030 new cases of lung cancer (121,680 in men and 112,350 in women)
* About 154,050 deaths from lung cancer (83,550 in men and 70,500 in women)

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. Technical Architecture** :



****

**3. Project Flow**

* The user interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once the model analyses the input the prediction is showcased on the UI

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

* Data collection
  + Collect the dataset or create the dataset
* Visualizing and analyzing data
  + Import Libraries
  + Read The dataset
  + Descriptive analysis
  + Exploratory Data Analysis
* Data pre-processing
  + Handling Missing Values
  + Drop the unwanted columns
  + Splitting data into train and test
  + Feature Scaling
* Model building
  + Import the model building libraries
  + Initializing the model
  + Training and testing the model
  + Evaluating the performance of the model
  + Save the model
* Application Building
  + Create an HTML file
  + Build python code

****

**3.1 : Data Collection**

ML depends heavily on data, It is most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

**Download the dataset**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

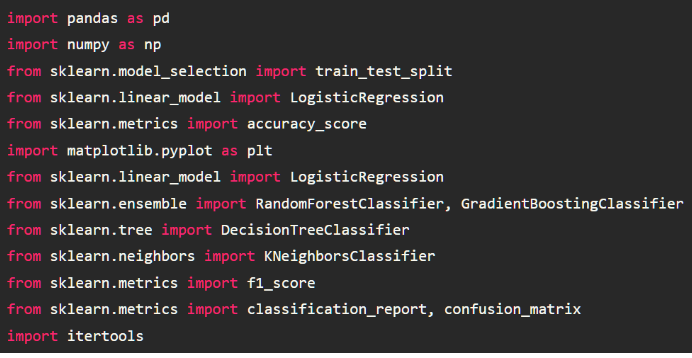
In this project, we have used ThoracicSurgery.csv data. This data is downloaded from the drive. Please refer to the link given below to download the dataset.

**https://drive.google.com/file/d/1TqBklVuZluzklQNf0HfKS87LL2ebLshp/view?usp=sharing**

****

**3.2 : Visualizing and Analyzing data**

. **Improting the libraries**

****

**. Read The Dataset**

df.head()

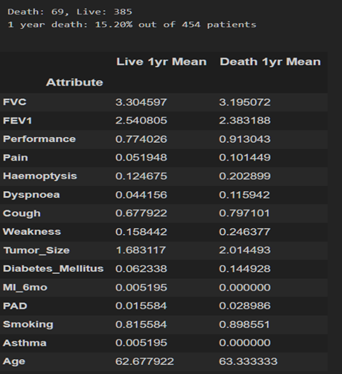
df.columns

. **Descriptive Analysis**

df.describe()

. **Exploratory Data Analysis**



Shows the prediction  ****

**3.3 : 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.

**3.4 : 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.

****

**3.5 : 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.

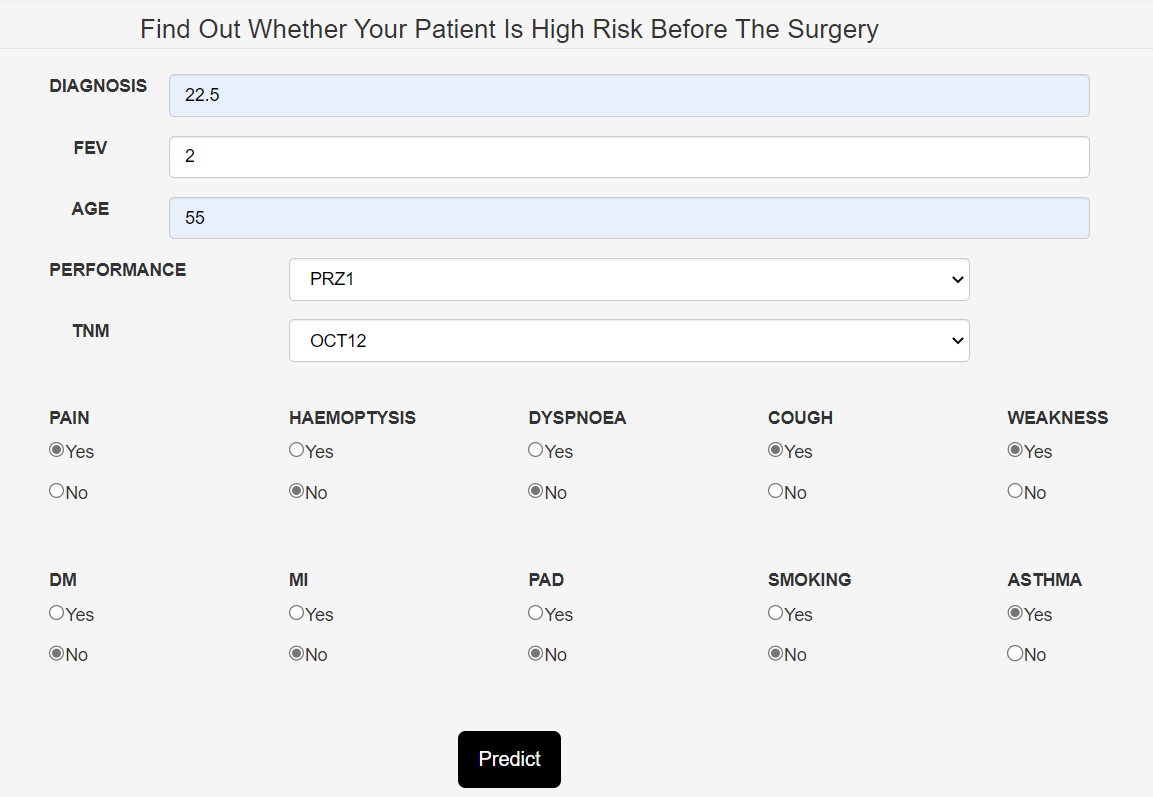
At last we have to train this model in the IBM.

****

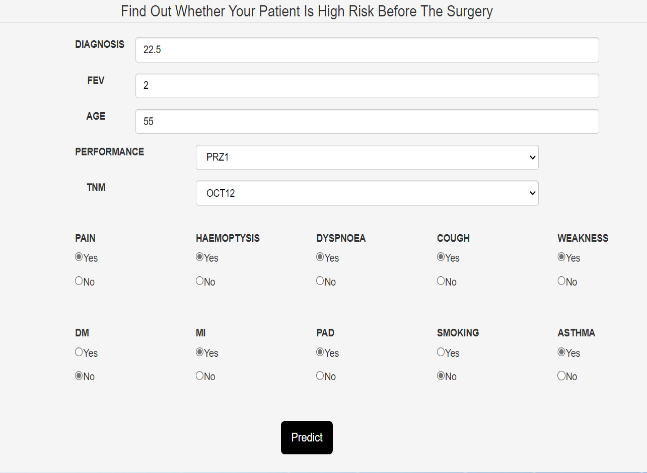
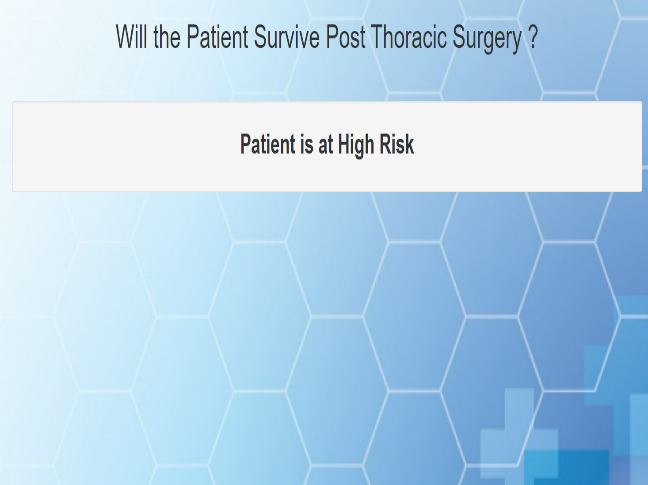
**4 : Result**

**4.1 : Local Deployment Result**

Input1 Output1

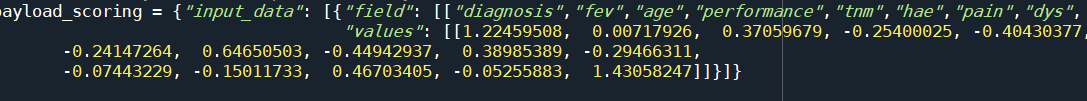
 

Input2 Output2

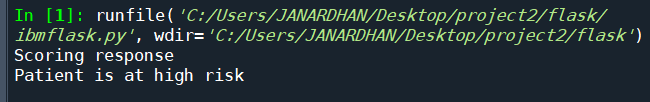
 

**4.1 : IBM Deployment Result**

Input



Output



**5.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.