UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Module	Assessment Type
Big Data	Report

Report On Lung Cancer Dataset and Prediction

Submitted By: Mahan Timalsena and Prabhat Gautam

Submitted On: May 02, 2022

Group: L6CG2

***Abstract*— This report discussed the prediction of lung cancer on the basis of different factors and also gave some minimal information about similar research and Big data and its applications. The primary goal of using this dataset is to figure out what ages and gender get most of the lung cancer. Besides, that is what causes and symptoms that lead to lung cancer to them.**

Introduction to Big Data

The phrase "Big Data" has lately been given to datasets that have become so large that working with them using typical database management systems is challenging. They are data sets that are too enormous to collect, store, manage, and process in a reasonable length of time using frequently used software tools and storage systems. (Nada Elgendy, 2018)

Big Data Processing

Big data processing is a collection of approaches or programming models for accessing massive volumes of data and extracting meaningful information for decision-making. The next sections go through some of the big data analysis tools and approaches accessible in server farms. Standard programming paradigms such as message passing interfaces are inadequate since vast volumes of data are often kept on hundreds of commodity servers. As a result, data centres are using new parallel programming techniques to increase the efficiency of NoSQL databases. (Farhad Mehdipour, 2017)

Sentiment Analysis

The technology of sentiment analysis is used to derive emotions from text. Learning how to extract and categorize valuable ideas from user-generated internet writings will assist individuals, corporate and government intelligence, and decision-making. In general, sentiment analysis approaches are classified into two types: lexicon- based and machine learning-based. Machine learning techniques employ learning algorithms and classification classifiers trained on a given dataset. (Woldemariam, 2019)

No SQL

Massive data (structured, semi-structured, and unstructured) has grown in popularity in recent years, posing a growing number of challenges in terms of the 3Vs (volume, variety, and velocity). A vast or complicated set of information is referred to as big data. Relational data processing methods and applications are incapable of dealing with it. To address this problem, NoSQL (rather than only SQL) databases were developed. NoSQL databases are non-relational database management systems (DBMS) that don't have a querying language. NoSQL databases can store and processing large amounts of structured and unstructured data. Metadata database management systems, key-value files, column family datastores, and graph datastores are all schema-free and capable of storing vast volumes of data. (Jitender Kumar, 2017)

Search Engine Technology

Image, text, video, news, academic, and industry-related search results are all possibilities. Search Engine Optimization (SEO) considers online marketing strategy and offers the most relevant results to users in a faster and more accessible manner. SEO is a collection of tactics and procedures for improving website traffic by putting a website at the top of search results pages. Search engine optimization is based on several ideas. The results from these search engines are displayed as hyperlinks to their respective websites. Using SEO tactics, the most relevant and useful website is placed at the top of a search results page. Before getting into SEO tactics, it's vital to understand how search engine's function, which involves crawling, indexing, processing, sorting, and retrieving results. (Varsha, 2021)

Data Warehousing

Data warehousing is a collection of decision-making tools aimed at assisting experienced professionals in making better, faster judgments. In the previous three years, the quantity of goods and services accessible, as well as the utilization of these technologies by business, has expanded rapidly. Hardware, database software, and tools are all part of data warehousing. Manufacturing, retail, financial services and healthcare have all successfully implemented data warehousing technologies. (Chaudhuri, 2019)

Data Mining

In the realm of information technology, data mining is a logical step. It's straightforward to describe how to find, retrieve, filter, and evaluate data. It's a method for extracting valuable data from vast volumes of data stored in databases, data centres, and other data storage places. Data mining is the process of collecting relevant knowledge, regularities, or high-level information from databases so that it may be seen or assessed from many angles. (Ogunleye, 2021)

Hadoop

Hadoop is a distributed database that uses the MapReduce image, text, video, news, academic, and industry computing paradigm to analyse enormous amounts of data. Hadoop is a scalable, processing-intensive, and storage-intensive distributed database. HDFS, Hadoop's file system component, saves metadata as file system blocks. HDFS has the name node and data nodes. HDFS is built on a master-slave architecture. There is only one name node in an HDFS cluster, which is a master server that maintains the file system namespace and directories hierarchically. There are two files in the data node. The first file contains the data, while the second contains the block's creation stamp. (Toshifa, 2019)

Background of the study

The dataset was taken from Kaggle which was last updated by Mysar Ahmad Bhat. The values of the target column 'lung cancer' are given in '1' and '2'. 1 refers to not affected by lung cancer and 2 refers to affected by lung cancer. In the gender column, 0 refers to male, and 1 refers to female. In the age column, the age of people is given. Remaining column, the data is in 0 and 1 format which means 1 refers to False and 2 refers to True.

Lung cancer has repeatedly emerged as one of the most fatal diseases that humanity has ever known. It is also one of the most common cancers and one of the leading causes of death. The disease tends to be asymptomatic in its early stages, making detection nearly impossible. As a result, early cancer detection is critical in saving lives. Early detection can improve a patient's chances of recovery and cure. Technology plays a critical role in accurately detecting cancer. Based on their research, several researchers have proposed various methods.

The main motive behind using this dataset is to figure out what ages and gender get most of the lung cancer. Figure out the different age groups people and gender cancer rates and compare that result with the other factors like symptoms and causes like gender, age, smoking, yellow fingers, anxiety, peer pressure, chronic disease, fatigue, allergy, wheezing, alcohol consuming, coughing, shortness of breath, shallowing difficulty, chest pain based on the similar age and gender which would help to figure out the reason of lung cancer and which might be useful to predict the lung cancer and might be helpful decreasing the cancer rate.

Related Work

Lung cancer prediction by Deep Learning to identify benign lung nodules

Deep learning has been introduced as a strategy for recognizing and arranging carcinogenic nodules. Our objective was to test our Lung Cancer Prediction Convolutional Neural Network, which had been prepared on US screening information, on an approaching information of uncertain knobs in an European multi-focus examination to preclude harmless nodules while keeping up with high cellular breakdown in the lungs awareness. The LCP-CNN was prepared to create a harm score for every nodule utilizing CT information from the US National Lung Screening Trial and approved involving CT scan from people in the Initial Lung Cancer Detection Utilizing AI and Big Data project.

Prediction of lung cancer risk based on age and smoking history

The CISNET models predict cancer death in any year of life, but their predictions are very uncertain, so it is difficult to use the models themselves to predict cancer death. We wanted to create a model that could produce reliable estimates of the probability of cancer death based on current age, smoking start age, smoking stop age, and smoking intensity. To test the model's estimates of cancer mortality risk vs age, the model's projections of cancer mortality risk were fitted to the mean of published CISNET model projections for the never smoker and six alternate lifetime smoking load scenarios.

Methodology

The dataset was found on Kaggle. From the description given, the data was collected from an online lung cancer prediction system. It contains 55394 records and 16 attributes. The attributes are gender, age, smoking, yellow fingers, anxiety, peer pressure, chronic disease, fatigue, allergy, wheezing, alcohol consuming, coughing, shortness of breath, shallowing difficulty, chest pain, and lung cancer. The block diagram of the entire work is given below.

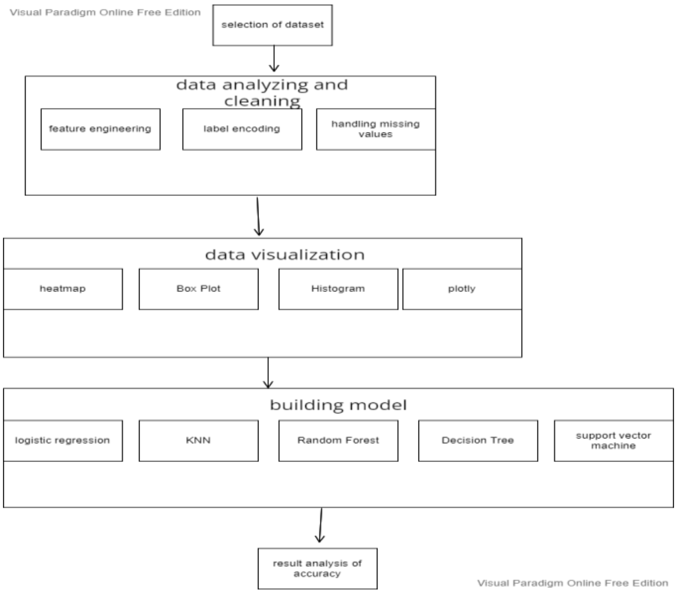


fig: Methodology flow diagram

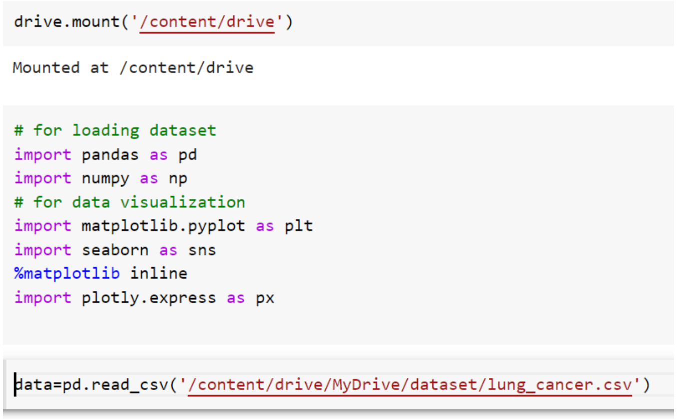
The task is completed as shown in the diagram above. The first step is to choose a dataset. We used Kaggle to find a lung cancer prediction dataset. Following that, we cleaned and analysed the data. We've worked on feature engineering, label encoding, and missing value handling. There were no null values in the dataset, according to the results. To convert a string to an integer, we used label encoding. We also changed the name of the column.

We use various types of charts to visualize data. Heatmap, histogram, box plot, and Plotly were used. These graphs aided us in gaining a better understanding of the data before constructing the model.

A different model is used by us. Random forest yielded the best results, while support vector machine (SVM) yielded less accurate results.

Data reading and exploring

We have used the pandas library to read data.



```
drive.mount('/content/drive')
```

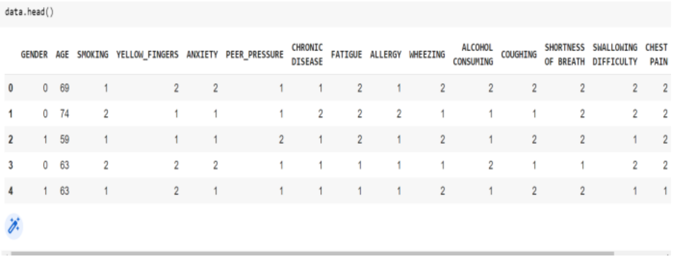
```
Mounted at /content/drive
```

```
# for loading dataset
import pandas as pd
import numpy as np
# for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly.express as px
```

```
data=pd.read_csv('/content/drive/MyDrive/dataset/lung_cancer.csv')
```

fig: Reading the data using panda library

Showing data first five rows



```
data.head()
```

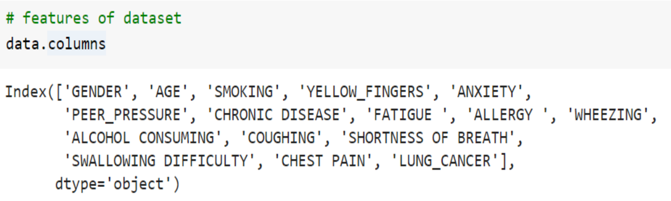
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
0	0	69	1	2	2	1	1	2	1	2	2	2	2	2	2
1	0	74	2	1	1	1	2	2	2	1	1	1	2	2	2
2	1	59	1	1	1	2	1	2	1	2	1	2	2	1	2
3	0	63	2	2	2	1	1	1	1	1	2	1	1	2	2
4	1	63	1	2	1	1	1	1	1	2	1	2	2	1	1

fig: Showing the first five-row of data

Data analysis and cleaning

Data must be cleaned for analysis. One can clean a dataset using different pre-processing processes.

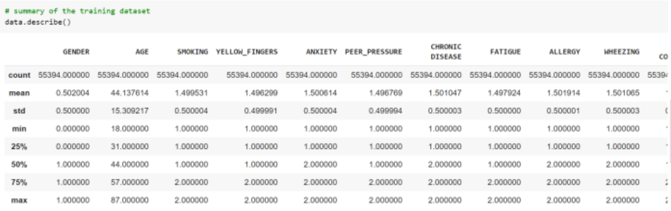
- View columns



```
# features of dataset
data.columns
```

```
Index(['GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY',
       'PEER_PRESSURE', 'CHRONIC DISEASE', 'FATIGUE ', 'ALLERGY ', 'WHEEZING',
       'ALCOHOL CONSUMING', 'COUGHING', 'SHORTNESS OF BREATH',
       'SWALLOWING DIFFICULTY', 'CHEST PAIN', 'LUNG_CANCER'],
      dtype='object')
```

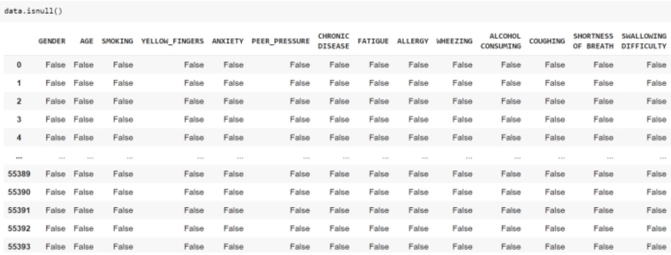
- view summary



```
# summary of the training dataset
data.describe()
```

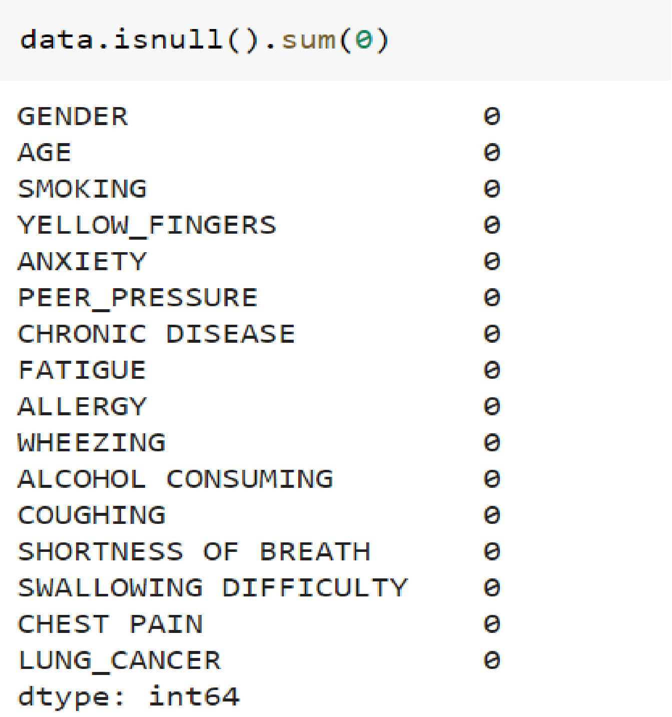
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	CO
count	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	55394.000000	5539
mean	0.502004	44.137614	1.499531	1.496299	1.500614	1.496769	1.501047	1.497924	1.501914	1.501065	
std	0.500000	15.309217	0.500004	0.499991	0.500004	0.499994	0.500003	0.500000	0.500001	0.500003	
min	0.000000	18.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
25%	0.000000	31.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	
50%	1.000000	44.000000	1.000000	1.000000	2.000000	1.000000	2.000000	1.000000	2.000000	2.000000	
75%	1.000000	57.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	
max	1.000000	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	

- Check null values



```
data.isnull()
```

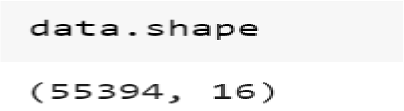
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
55389	False	False	False	False	False	False	False	False	False	False	False	False	False	False
55390	False	False	False	False	False	False	False	False	False	False	False	False	False	False
55391	False	False	False	False	False	False	False	False	False	False	False	False	False	False
55392	False	False	False	False	False	False	False	False	False	False	False	False	False	False
55393	False	False	False	False	False	False	False	False	False	False	False	False	False	False



```
data.isnull().sum(0)
```

```
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
```

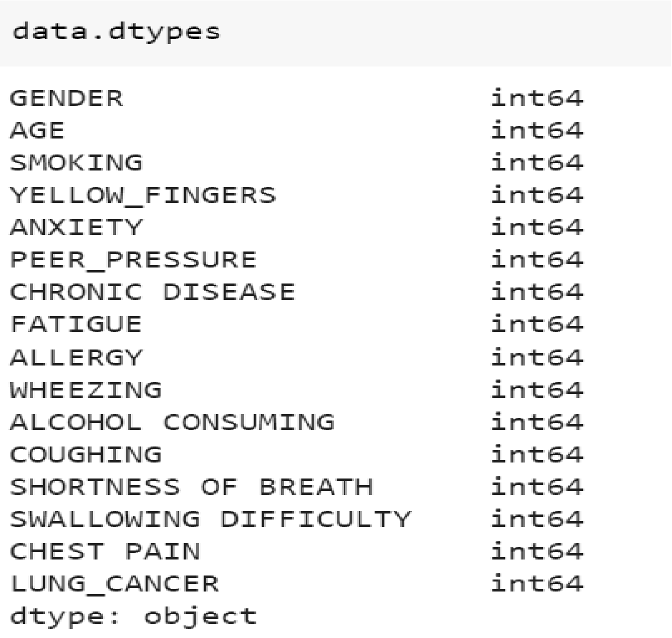
- Count total rows and columns



```
data.shape
```

```
(55394, 16)
```

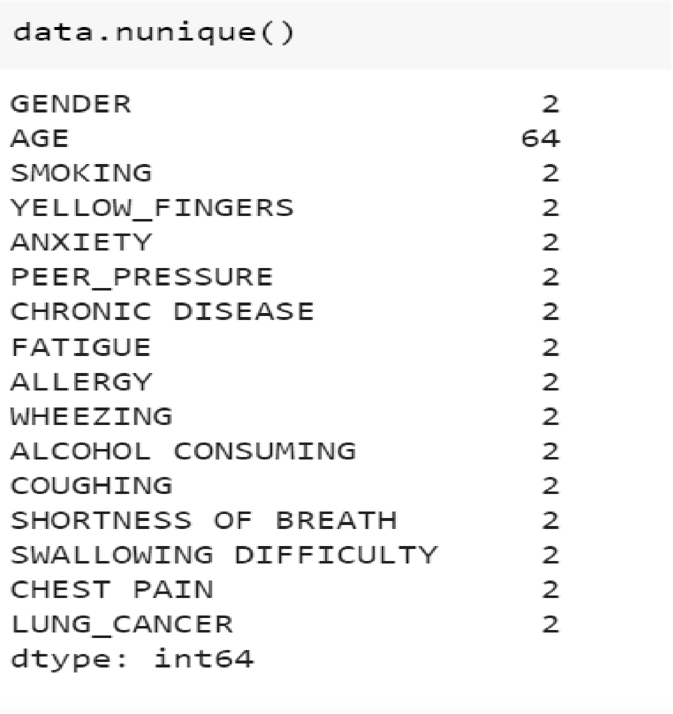
- Find the data types of columns value



```
data.dtypes
```

```
GENDER                   int64
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              int64
dtype: object
```

- Find unique values



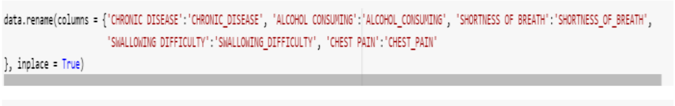
```
data.nunique()
```

```
GENDER                    2
AGE                      64
SMOKING                   2
YELLOW_FINGERS            2
ANXIETY                   2
PEER_PRESSURE             2
CHRONIC DISEASE           2
FATIGUE                   2
ALLERGY                   2
WHEEZING                  2
ALCOHOL CONSUMING         2
COUGHING                  2
SHORTNESS OF BREATH       2
SWALLOWING DIFFICULTY     2
CHEST PAIN                2
LUNG_CANCER               2
dtype: int64
```

From the above figures, from the dataset, we have checked null values using IsNull() function. The results show there is no row that contains null values. As all data are in numeric format, we checked data types and found int type. We also check the unique values presented in the dataset.

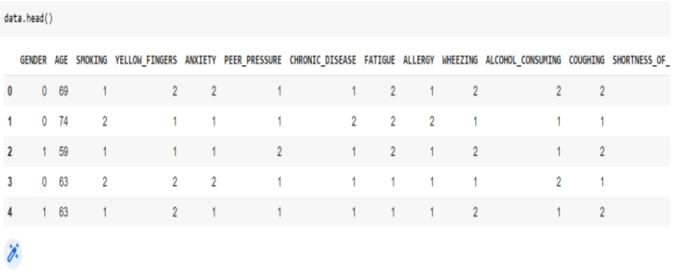
Data cleaning and analysis

- Rename column name which has “two words” column name.



```
data.rename(columns = {'CHRONIC DISEASE':'CHRONIC_DISEASE', 'ALCOHOL CONSUMING':'ALCOHOL_CONSUMING', 'SHORTNESS OF BREATH':'SHORTNESS_OF_BREATH',
                'SWALLOWING DIFFICULTY':'SWALLOWING_DIFFICULTY', 'CHEST PAIN':'CHEST_PAIN'
}, inplace = True)
```

Result,

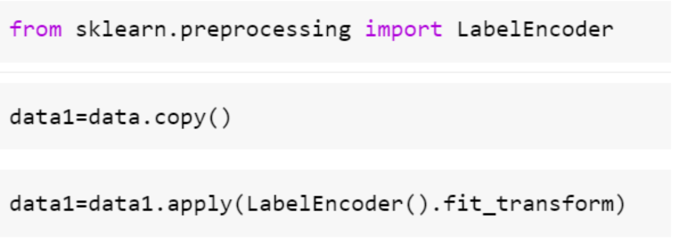


```
data.head()
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	SHORTNESS_OF_
0	0	69	1	2	2	1	1	2	1	2	2	2	
1	0	74	2	1	1	1	2	2	2	1	1	1	
2	1	59	1	1	1	2	1	2	1	2	1	2	
3	0	63	2	2	2	1	1	1	1	1	2	1	
4	1	63	1	2	1	1	1	1	1	2	1	2	

From the above figures, previous data set columns were not inappropriate format, so we convert it into a meaningful format. we rename the column name by adding an underscore (_) between two words.

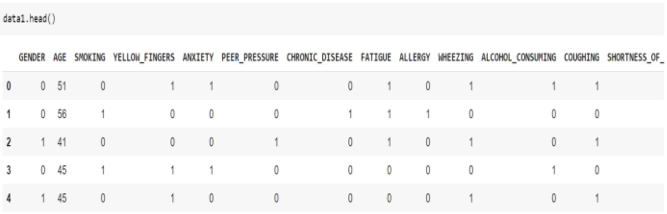
all values are in a numeric format so don’t need to convert them into 0 and 1 format for getting better accuracy but we can convert label names like this.



```
from sklearn.preprocessing import LabelEncoder
```

```
data1=data.copy()
```

```
data1=data1.apply(LabelEncoder().fit_transform)
```



```
data1.head()
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	SHORTNESS_OF_
0	0	51	0	1	1	0	0	1	0	1	1	1	
1	0	56	1	0	0	0	1	1	1	0	0	0	
2	1	41	0	0	0	1	0	1	0	1	0	1	
3	0	45	1	1	1	0	0	0	0	0	1	0	
4	1	45	0	1	0	0	0	0	0	1	0	1	

The computer only understands numeric values, so we convert them into numeric form. It’s difficult to convert string to integer so we have used a label Encoder.

Data visualization

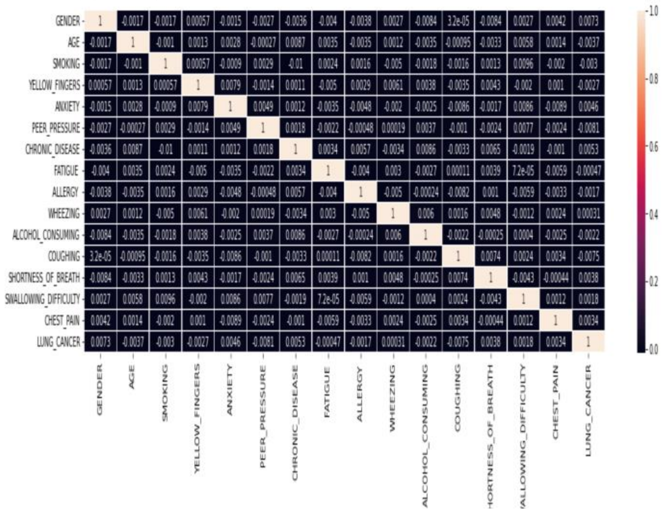
After data cleaning and pre-processing next step is to visualize data into different charts. We have visualized data in 4 different types of charts for better understanding. Proper data visualization helps to understand the data properly so that it helps to build a model better. To find the relationship between features we used the correlation function. Heatmap shows the relationship between feature variable and histogram to understand a feature more. Box plot is used to find the outliers present in features which should be removed to build a better model. Plotly described the total cases of infected and not infected people.

```
data.corr()
```

- Heatmap

```
fig, ax = plt.subplots(figsize=(20,5))
sns.heatmap(data.corr(), linewidths=.5, annot=True,ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6d0b572610>
```

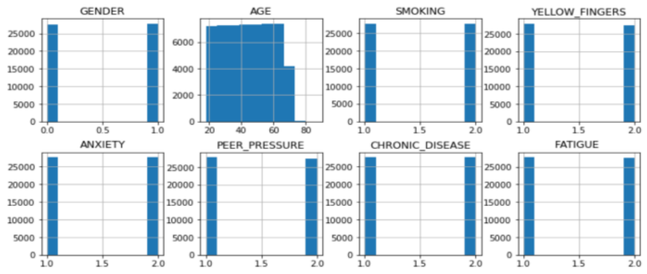


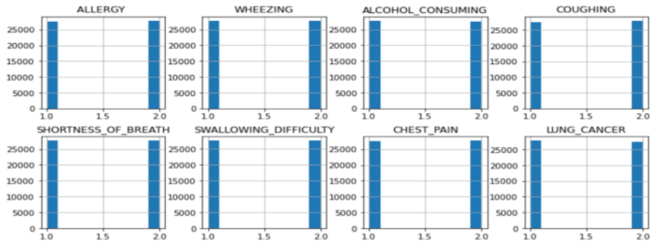
From the above figure, we can find the relation between features and the importance of features. We can see that gender, anxiety and chronic disease are of higher importance in lung cancer from the above figure.

- Histogram

```
data.hist(figsize=(12,12), layout=(4,4), sharex=False)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f64730e9410>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6475034990>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6472dcfb10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6472c27f10>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f6472ce3790>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f647280df90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6472831c50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f64727f4290>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f6472786b90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6472749310>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f64727244d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f64726d9ad0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f647269c110>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f6472654710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f647260dd10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f64725ce350>]],
      dtype=object)
```



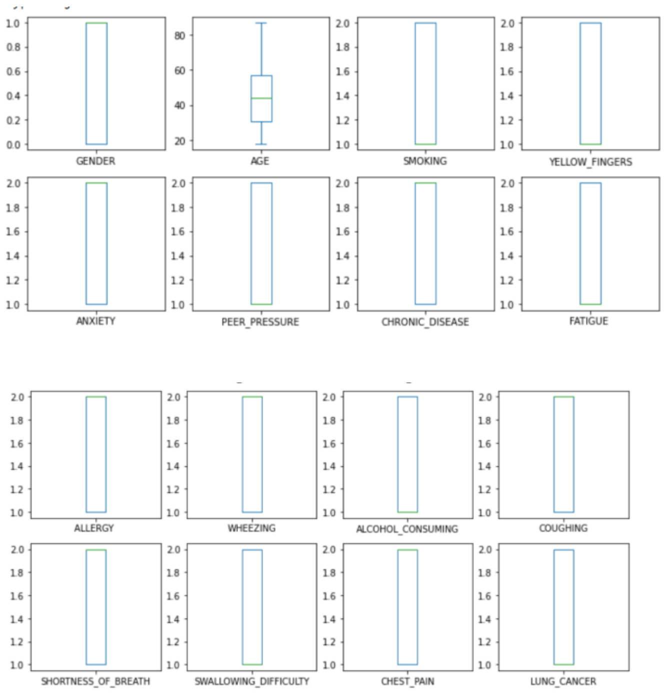


The above figure, it shows the count of feature values like gender, age, smoking, and yellow_fingers, etc. as you can see, ages having 50-65 have high lung cancer data.

- Boxplot

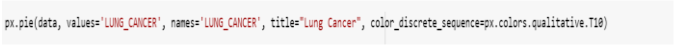
```
data.plot(kind="box", figsize=(12,12), layout=(4,4), sharex=False, subplots=True)
```

```
GENDER                      AxesSubplot(0.125,0.71587;0.168478x0.16413)
AGE                      AxesSubplot(0.327174,0.71587;0.168478x0.16413)
SMOKING                  AxesSubplot(0.529348,0.71587;0.168478x0.16413)
YELLOW_FINGERS           AxesSubplot(0.731522,0.71587;0.168478x0.16413)
ANXIETY                    AxesSubplot(0.125,0.518913;0.168478x0.16413)
PEER_PRESSURE           AxesSubplot(0.327174,0.518913;0.168478x0.16413)
CHRONIC_DISEASE         AxesSubplot(0.529348,0.518913;0.168478x0.16413)
FATIGUE                 AxesSubplot(0.731522,0.518913;0.168478x0.16413)
ALLERGY                    AxesSubplot(0.125,0.321957;0.168478x0.16413)
WHEEZING                AxesSubplot(0.327174,0.321957;0.168478x0.16413)
ALCOHOL_CONSUMING       AxesSubplot(0.529348,0.321957;0.168478x0.16413)
COUGHING                AxesSubplot(0.731522,0.321957;0.168478x0.16413)
SHORTNESS_OF_BREATH           AxesSubplot(0.125,0.125;0.168478x0.16413)
SWALLOWING_DIFFICULTY      AxesSubplot(0.327174,0.125;0.168478x0.16413)
CHEST_PAIN                 AxesSubplot(0.529348,0.125;0.168478x0.16413)
LUNG_CANCER                AxesSubplot(0.731522,0.125;0.168478x0.16413)
dtype: object
```

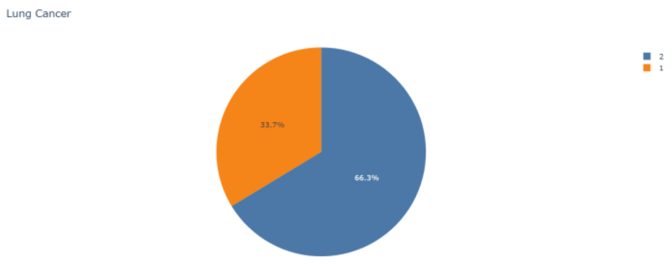


From the above figure, we can see there is no outliers except age. Outliers simply refer to wrong data that reduce model accuracy like negative age which doesn’t have to be negative.

- Plotly

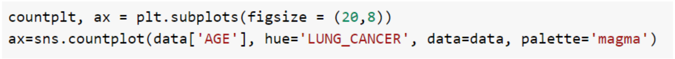


```
px.pie(data, values='LUNG_CANCER', names='LUNG_CANCER', title="Lung Cancer", color_discrete_sequence=px.colors.qualitative.T10)
```

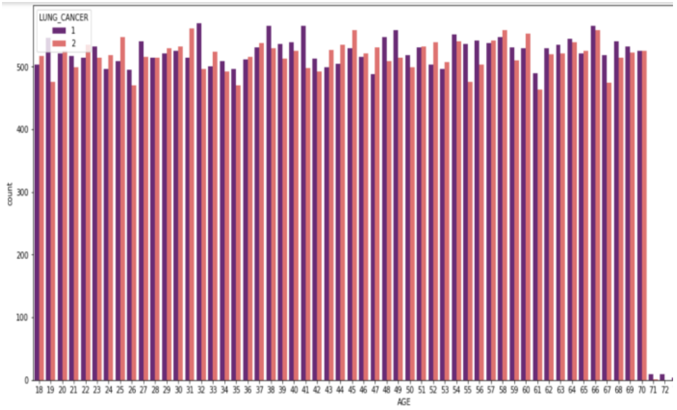


From the above figure, those infected with lung cancer are 66.3% and others are 33.7%.

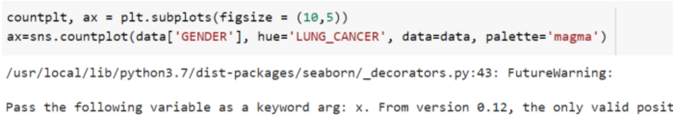
- Count plot



```
countplt, ax = plt.subplots(figsize = (20,8))
ax=sns.countplot(data['AGE'], hue='LUNG_CANCER', data=data, palette='magma')
```



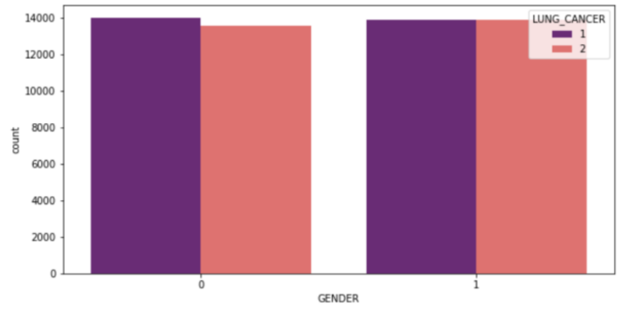
From the above figure, we can see that the people age having 31-to 66 are infected with lung cancer is higher than other.



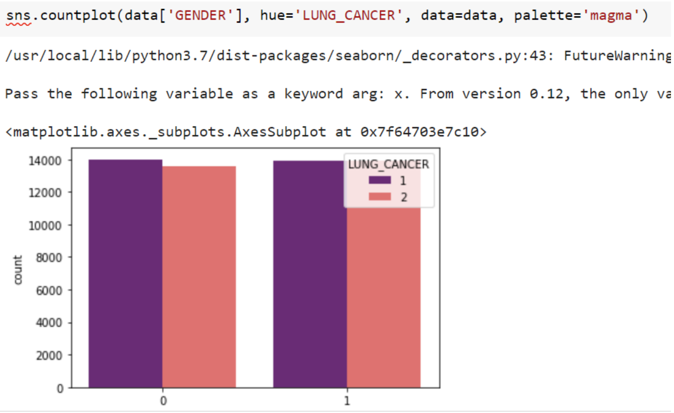
```
countplt, ax = plt.subplots(figsize = (10,5))
ax=sns.countplot(data['GENDER'], hue='LUNG_CANCER', data=data, palette='magma')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only valid posit
```



From the above figure, 0 gender is male and 1 is Female, and infected from lung_cancer is 2 and not infected is 1. As you can see, the number of males infected with lung cancer is lower than women.

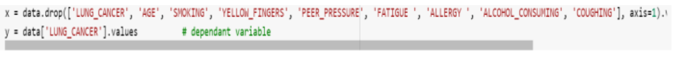


```
sns.countplot(data['GENDER'], hue='LUNG_CANCER', data=data, palette='magma')
```

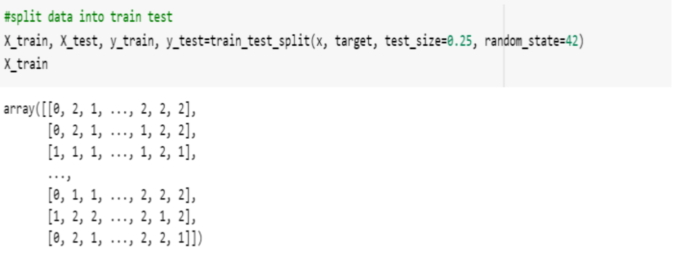
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning
Pass the following variable as a keyword arg: x. From version 0.12, the only va
<matplotlib.axes._subplots.AxesSubplot at 0x7f64703e7c10>
```

Model building

For building the model, at first, we separate the dependent variable and independent variable. In this, the dependent variable is the target variable “lung_cancer” column whereas the independent variable are remaining other columns. After that, we split dataset into training and testing. We used 25% of data for testing and 75% data for training. After that we built model like logistic regression, support vector machine, decision tree, random forest and KNN. After implementing different model, we found the accuracy predicted by models. And finally, we analyze the best model depending on their accuracy.

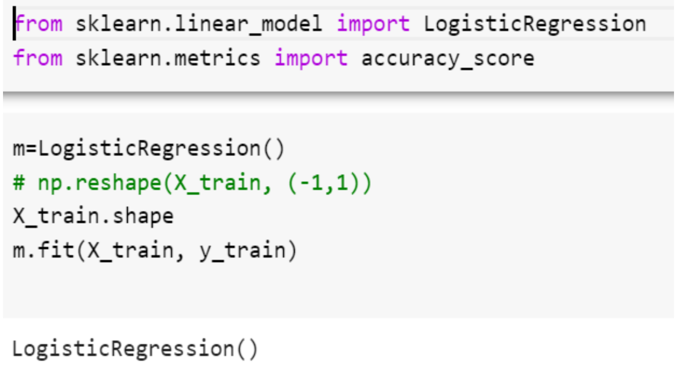


```
x = data.drop(['LUNG_CANCER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'PEER_PRESSURE', 'FATIGUE ', 'ALLERGY ', 'ALCOHOL_CONSUMING', 'COUGHING'], axis=1).
y = data['LUNG_CANCER'].values         # dependant variable
```



```
#split data into train test
X_train, X_test, y_train, y_test=train_test_split(x, target, test_size=0.25, random_state=42)
X_train
```

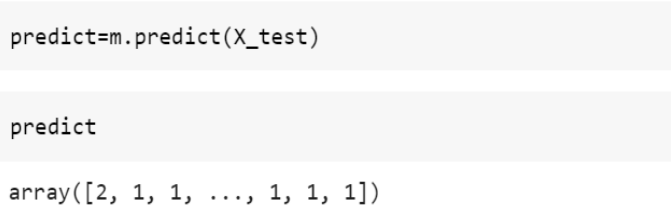
```
array([[0, 2, 1, ..., 2, 2, 2],
       [0, 2, 1, ..., 1, 2, 2],
       [1, 1, 1, ..., 1, 2, 1],
       ...,
       [0, 1, 1, ..., 2, 2, 2],
       [1, 2, 2, ..., 2, 1, 2],
       [0, 2, 1, ..., 2, 2, 1]])
```



```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
m=LogisticRegression()
# np.reshape(X_train, (-1,1))
X_train.shape
m.fit(X_train, y_train)
```

```
LogisticRegression()
```

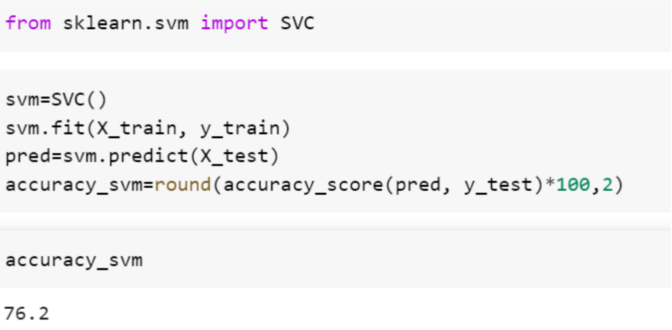


```
predict=m.predict(X_test)
```

```
predict
```

```
array([2, 1, 1, ..., 1, 1, 1])
```

From the above figures, we used 75% data for the train and the remaining 25% for tests. From the logistic regression, we got 80.15% accuracy.



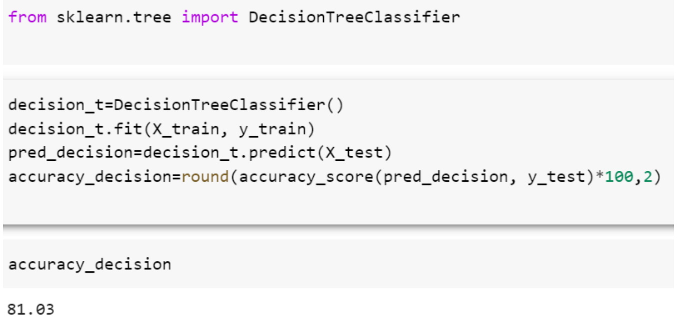
```
from sklearn.svm import SVC
```

```
svm=SVC()
svm.fit(X_train, y_train)
pred=svm.predict(X_test)
accuracy_svm=round(accuracy_score(pred, y_test)*100,2)
```

```
accuracy_svm
```

```
76.2
```

From the figure above, we got only 76.2% accuracy from the support vector machine.



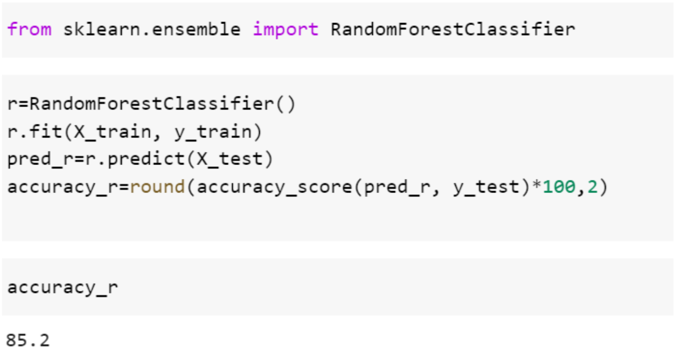
```
from sklearn.tree import DecisionTreeClassifier
```

```
decision_t=DecisionTreeClassifier()
decision_t.fit(X_train, y_train)
pred_decision=decision_t.predict(X_test)
accuracy_decision=round(accuracy_score(pred_decision, y_test)*100,2)
```

```
accuracy_decision
```

```
81.03
```

From the above figure, when implementing the decision tree we got 81.03% accuracy.



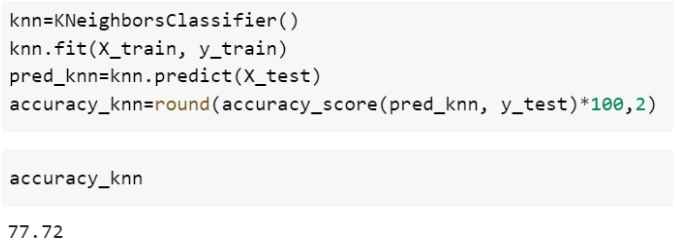
```
from sklearn.ensemble import RandomForestClassifier
```

```
r=RandomForestClassifier()
r.fit(X_train, y_train)
pred_r=r.predict(X_test)
accuracy_r=round(accuracy_score(pred_r, y_test)*100,2)
```

```
accuracy_r
```

```
85.2
```

From the figure above, implementing random forest we got 85.2% accuracy.



```
knn=KNeighborsClassifier()
knn.fit(X_train, y_train)
pred_knn=knn.predict(X_test)
accuracy_knn=round(accuracy_score(pred_knn, y_test)*100,2)
```

```
accuracy_knn
```

```
77.72
```

From the above figure, implementing KNN we got 77.72% accuracy.

Conclusion

From the research, we see there are different factors affecting lung cancer. We conclude that as compared to male, female infected with lung cancer is high. Age between 31-and 66 is most infected from lung cancer. We have implemented different models to view outcomes. Among them, the random forest gave the most accurate (85.2% accuracy) results as compared to others.

References

Auten, G., 2021. Recent Research on Income Distribution: An Overview of the Field. Capitalism &amp; Society, 15(1).

Chakrabarty, N., 2018. A Statistical Approach to Adult Census Income Level Prediction. Greater Noida, India, IEEE.

Chaudhuri, S., 2019. An Overview of Data Warehousing and OLAPTechnology, s.l.: s.n.

Ding, F., 2019. Retiring Adult: s.l.: s.n.

Farhad Mehdipour, B. J., 2017. Big Data Processing. s.l.:ScienceDirect.

Jayavarthini, C., 2018. ANALYSIS AND PREDICTION OF ADULT INCOME. International Journal of Pure and Applied Mathematics, 118(22), pp. 587-590.

Jitender Kumar, V. G., 2017. Security analysis of unstructured data in NoSQL MongoDB database.