LUNG CANCER PREDICTION

SUBMITTED BY:

GAUTAM JAISWAL
SOPHIYA SALEEMA QURESHI
KAMAKSHI SHARMA
OJAS KHATAVKAR

INTRODUCTION

Lung cancer is the one of the leading cause of cancer deaths in both women and men. Manifestation of Lung cancer in the body of the patient reveals through early symptoms in most of the cases.Treatment and prognosis depend on the histological type of cancer, the stage (degree of spread), and the patient's performance status. Possible treatments include surgery, chemotherapy , and radiotherapy Survival depends on stage, overall health , and other factors, but overall only 14% of people diagnosed with lung cancer survive five years after the diagnosis.

OBJECTIVE

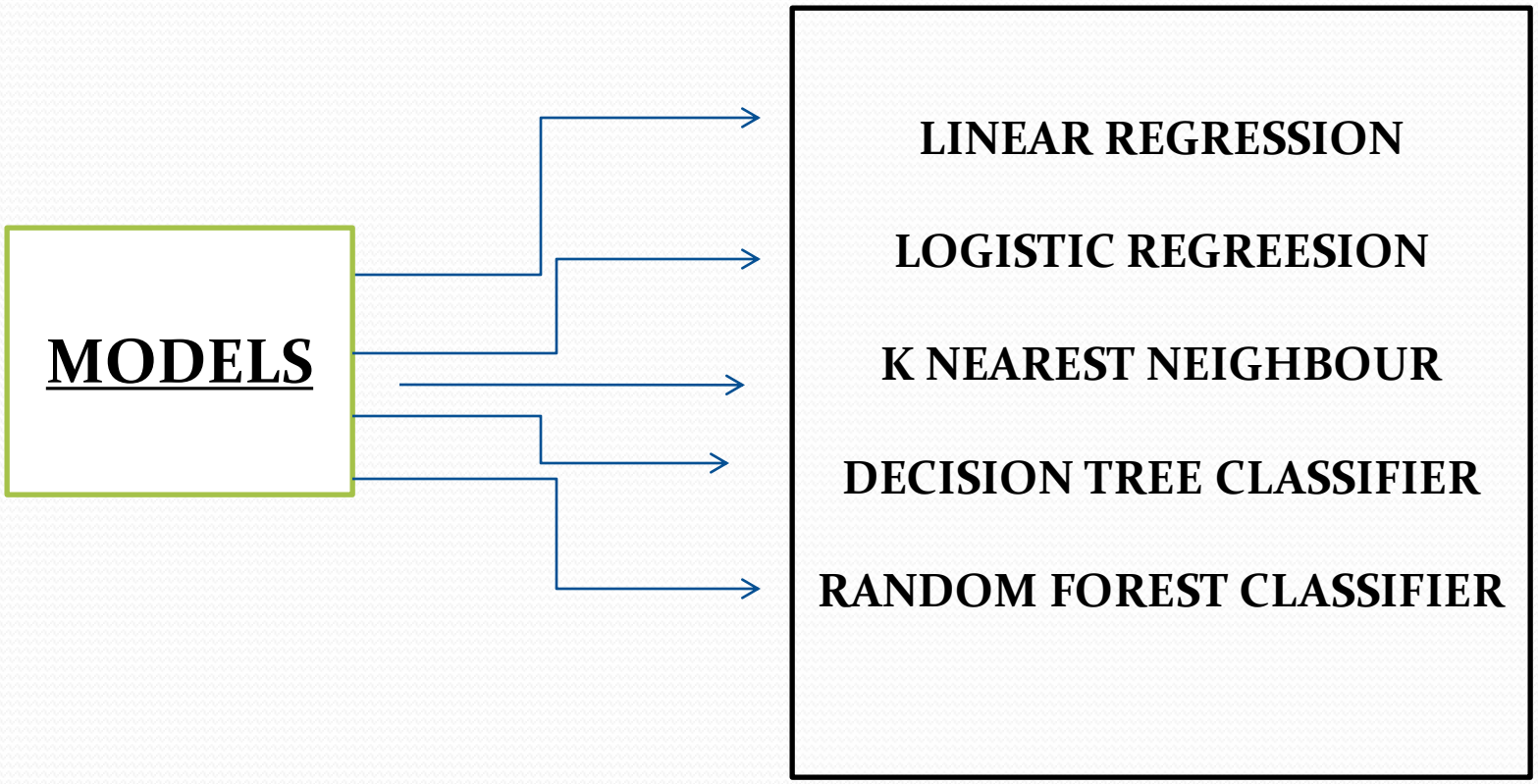
- Implement various machine learning algorithm to Detect Lung Cancer of person based on various symptoms and habits (using the gathered dataset from various source) and find out which algorithm is best(highest accuracy)for the given problem.

Symptoms of Lung Cancer

The following are the generic lung cancer symptoms

- i. A cough that does not go away and gets worse over time
- ii. Coughing up blood (heamoptysis) or bloody mucus.
- iii. Chest, shoulder, or back pain that doesn't go away and often is made worse by deep Hoarseness
- iv. Weight loss and loss of appetite
- v. Increase in volume of sputum.

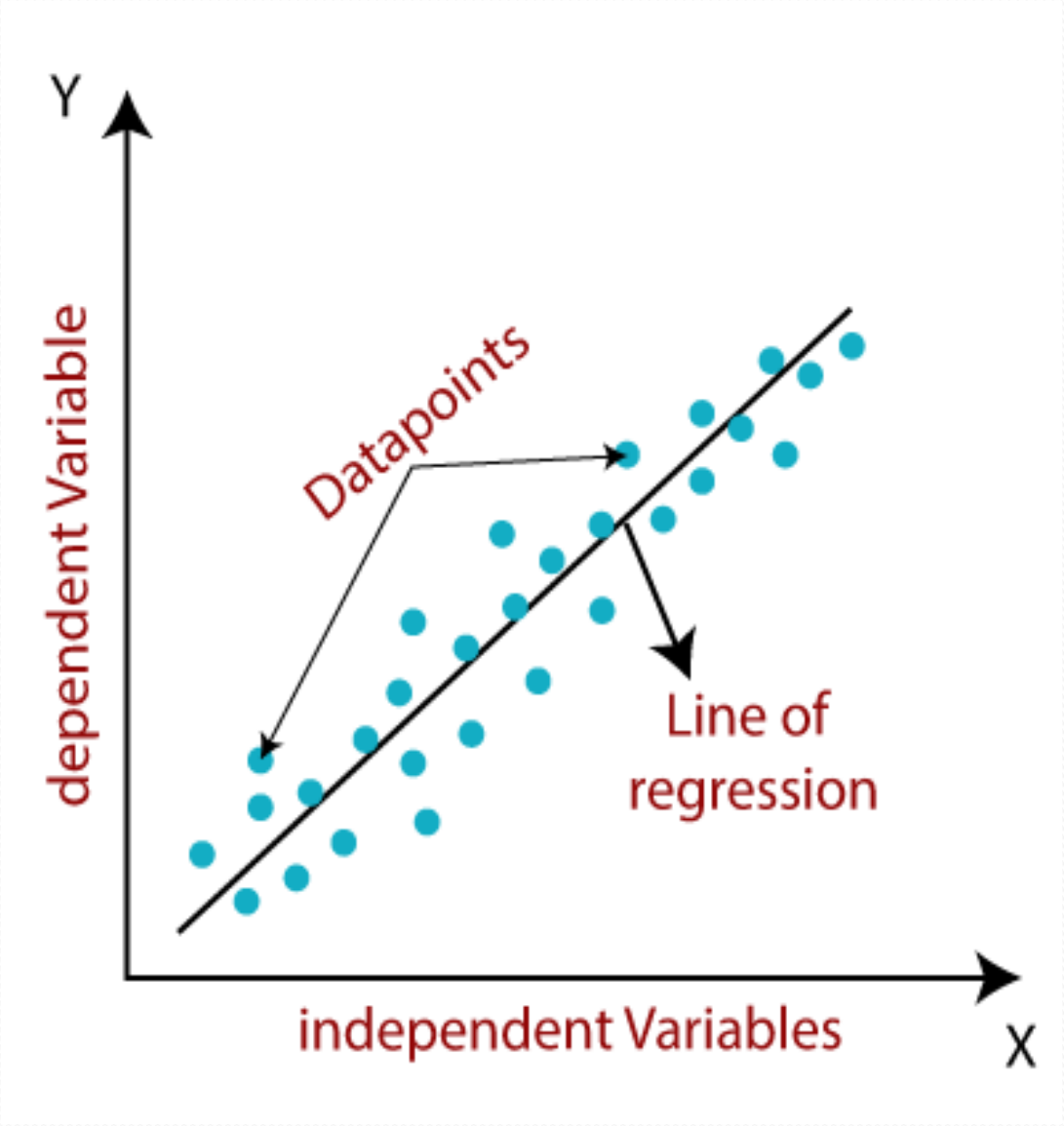
METHODOLODY



LINAER REGRESSION

_Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

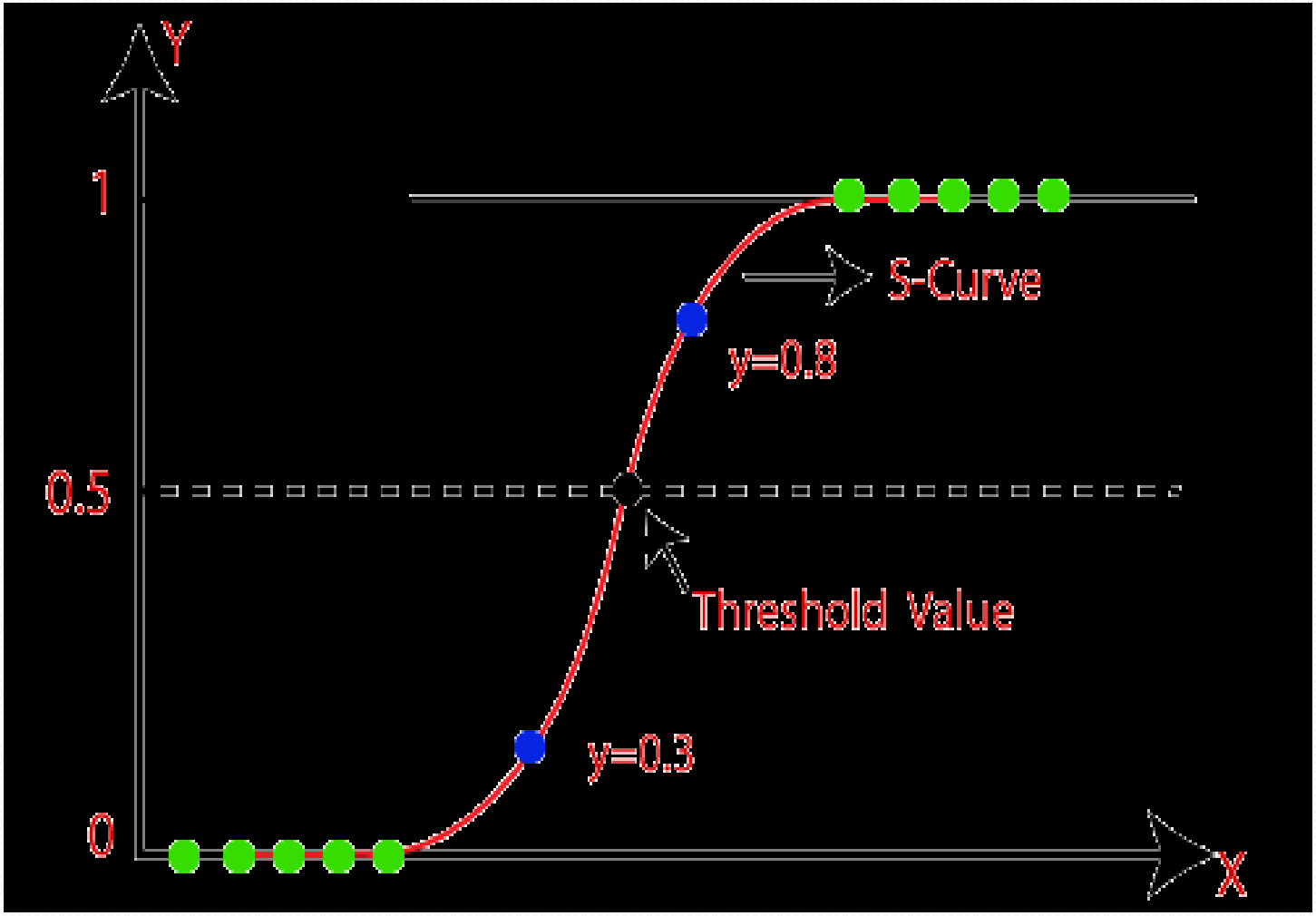

Y
dependent Variable
Datapoints
Line of
regression
independent Variables
X

LOGISTIC REGRESSION

It is a supervised learning classification technique that forecasts the likelihood of a target variable. There will only be a choice between two classes. Data can be coded as either one or yes, representing success, or as 0 or no, representing failure.

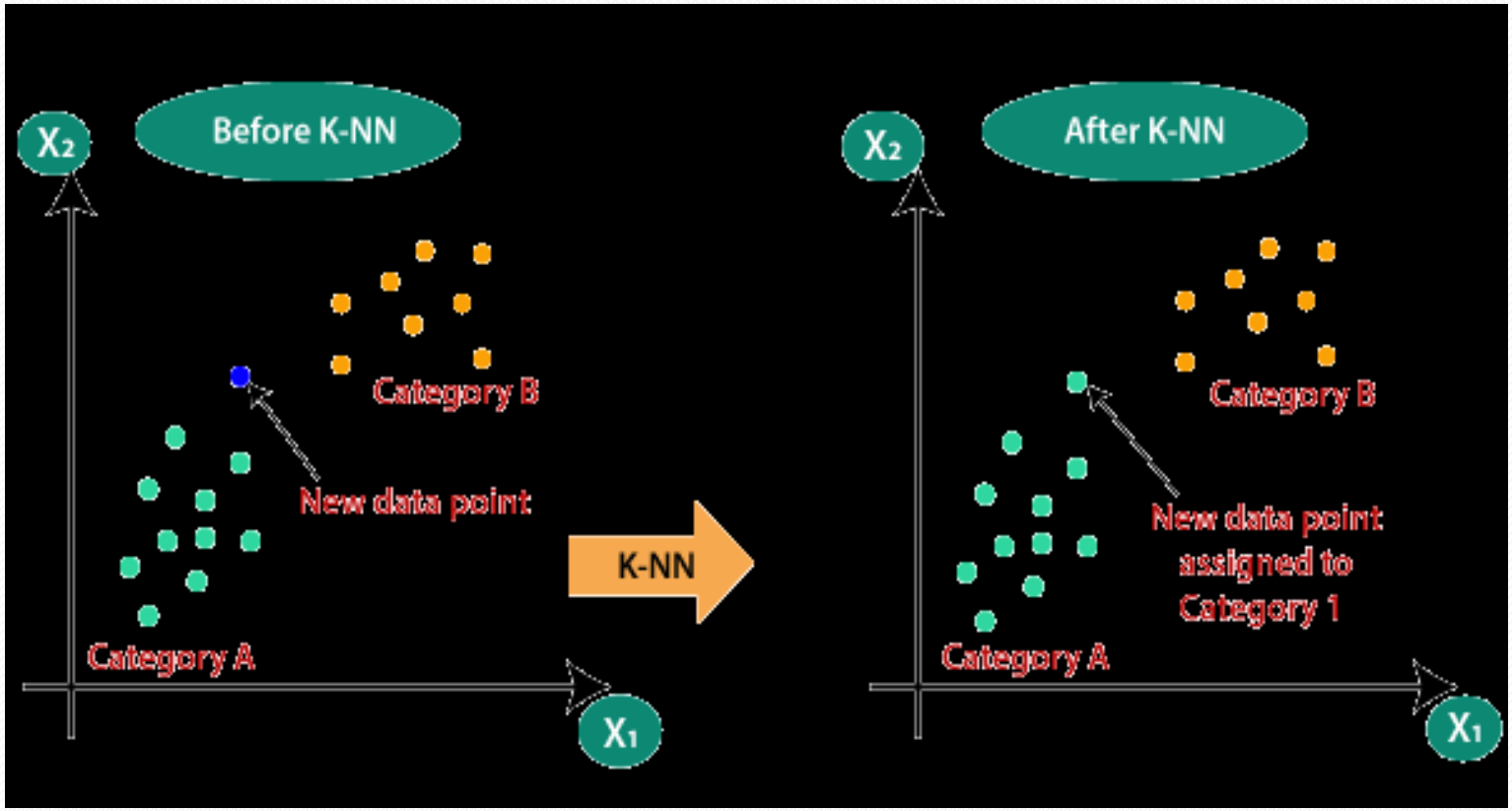
The dependent variable can be predicted most effectively using logistic regression. When the forecast is categorical, such as true or false, yes or no, or a 0 or 1, you can use it. A logistic regression technique can be used to determine whether or not an email is a spam.

Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).


Y
1
0.5
0
S-Curve
y=0.8
Threshold Value
y=0.3
X

K NEAREST NEIGHBOUR(KNN)

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.This algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.


Before K-NN
X2
Category B
New data point
Category A
X1
K-NN
After K-NN
X2
Category B
New data point
assigned to
Category 1
Category A
X1

DECISION TREE CLASSIFIER

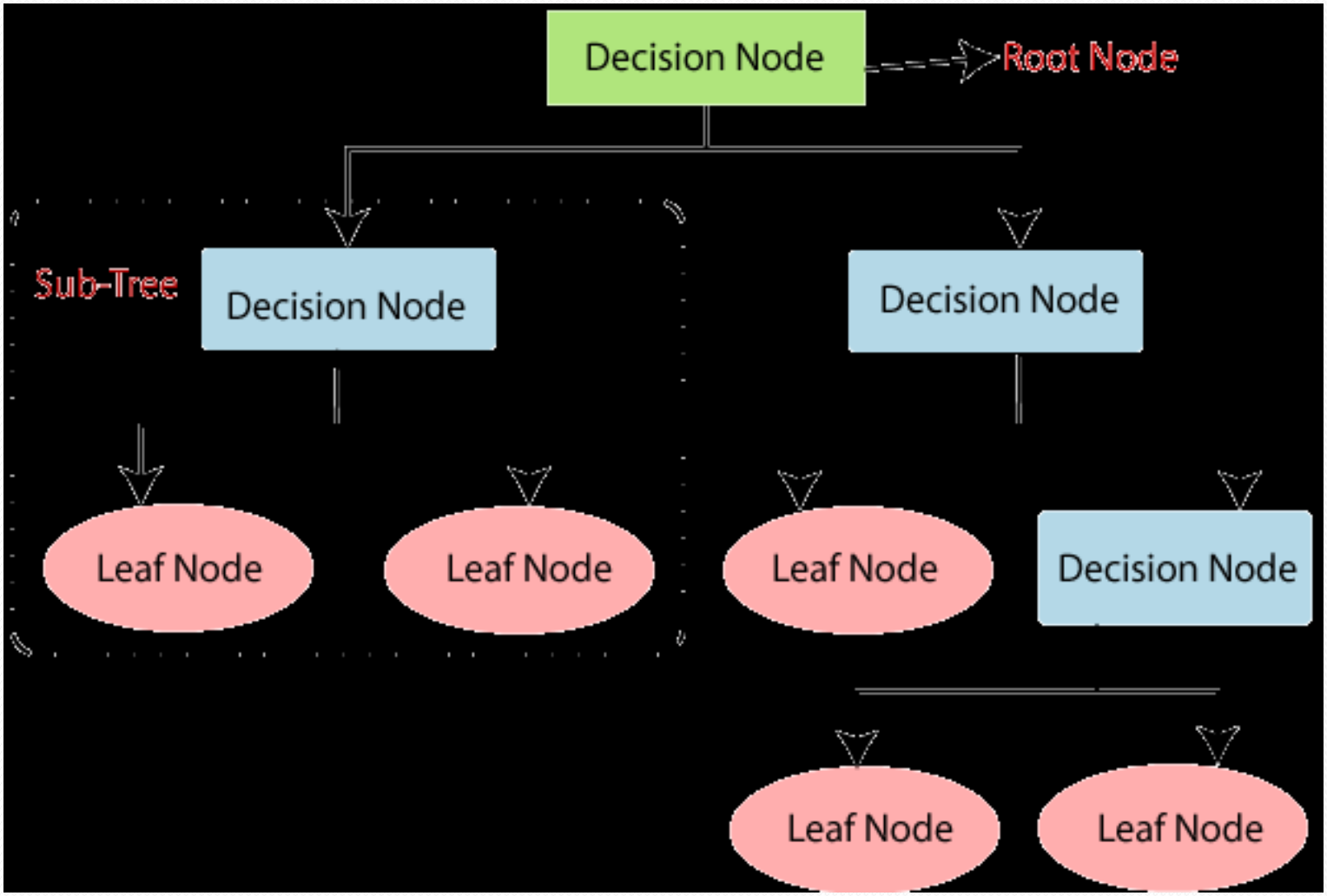
Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

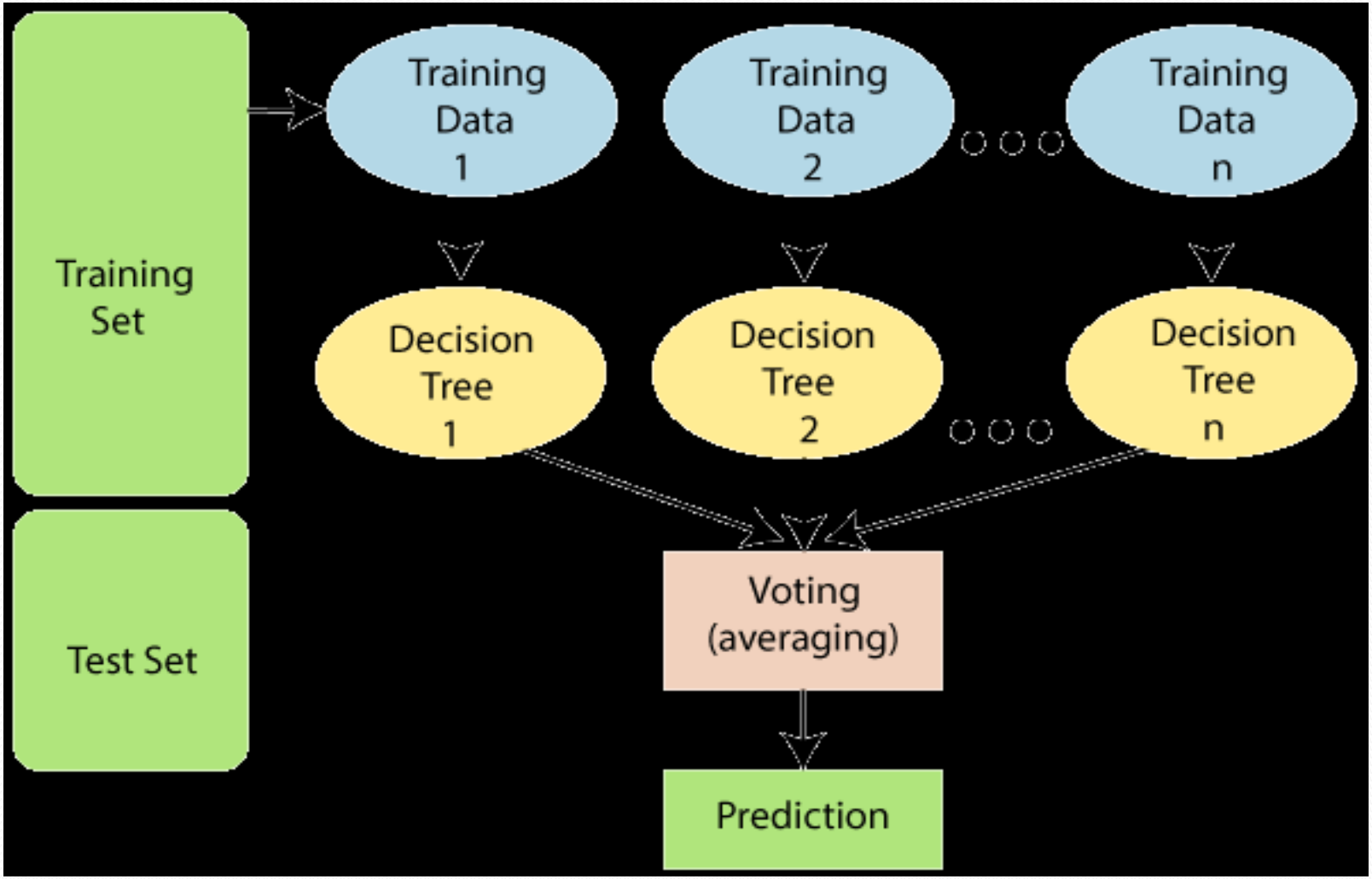

Decision Node
Root Node
Sub-Tree
Decision Node
Decision Node
Leaf Node
Leaf Node
Leaf Node
Decision Node
Leaf Node
Leaf Node

RANDOM FOREST CLASSIFIER

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.


Training Set
Test Set
Training Data 1
Training Data 2
Training Data n
Decision Tree 1
Decision Tree 2
Decision Tree n
Voting (averaging)
Prediction

STEPS

- 1. import library
- 2. read dataset
- 3. data analysis
- 4. Label encoding of the categorical columns
- 5. defining x and y
- 6. splitting x and y into train and test data
- 7. import the model
- 8. train the model with x_train and y_train
- 9. predict with x_test and got predicted y
- 10. evaluation with confusion matrix, classification report , accuracy score.

IMPLEMENTATION

Lung Cancer Detection

About Dataset The effectiveness of the cancer prediction system helps people to know their cancer risk wi a low cost and it also helps the people to take the appropriate decision based on their cancer risk status. The data is collected from the website online lung cancer prediction system.

In [17]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df1=pd.read_csv("lung_cancer.csv")
df1.columns
```

Out[17]:
```
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')
```

In [27]:
```
df.head(2)
```

Out[27]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE
0	M	69	1	2	2	1	1	2
1	M	74	2	1	1	1	2	2

linear Regression

In [26]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df1['LUNG_CANCER']=lb.fit_transform (df1[ 'LUNG_CANCER'])
df1['GENDER']=lb.fit_transform (df1[ 'GENDER'])
x=df1[[ 'GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', 'C
y=df1[ 'LUNG_CANCER']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.25)
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
print (f'first 5 actual_y: {y_test.values[:5]} predicted y: {y_pred[:5]}')

print('Model_score -',lr.score (x_test,y_test))
sns.heatmap(a, annot=True)
plt.show()
```

```
first 5 actual_y: [1 1 1 0 1] predicted y: [0.78383047 0.87888029 0.83485786 0.761
62371 0.79791579]
Model_score - 0.053811716223445605
```

logistic regression

In [18]:

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df1['LUNG_CANCER']=lb.fit_transform (df1[ 'LUNG_CANCER'])
df1['GENDER']=lb.fit_transform (df1[ 'GENDER'])
x=df1[[ 'GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', 'C
y=df1[ 'LUNG_CANCER']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.25)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
print (f'first 5 actual_y: {y_test.values[:5]} predicted y: {y_pred[:5]}')
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
print('confusion_matrix')
a=confusion_matrix (y_test,y_pred)
print(a)
print('classification_report')
print(classification_report (y_test,y_pred))
print('accuracy_score-', accuracy_score (y_test,y_pred))
print('Model_score -',lr.score (x_test,y_test))
sns.heatmap(a, annot=True)
plt.show()
```

k nearest classifier

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df1['LUNG_CANCER']=lb.fit_transform (df1[ 'LUNG_CANCER'])
df1['GENDER']=lb.fit_transform (df1[ 'GENDER'])
x=df1[[ 'GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', '(
```

nvert/html/ds project lung.ipynb?download=false

ds project lung

```
y=df1[ 'LUNG_CANCER']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.25)
#from sklearn.linear_model import LogisticRegression
#lr=LogisticRegression()
from sklearn.neighbors import KNeighborsClassifier
lr=KNeighborsClassifier()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
print (f'first 5 actual_y: {y_test.values[:5]} predicted y: {y_pred[:5]}')
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
print('confusion_matrix')
a=confusion_matrix (y_test,y_pred)
print(a)
print('classification_report')
print(classification_report (y_test,y_pred))
print('accuracy_score-', accuracy_score (y_test,y_pred))
print('Model_score -',lr.score (x_test,y_test))
sns.heatmap(a, annot=True)
plt.show()
```

decision tree classifier

In [20]:

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df1['LUNG_CANCER']=lb.fit_transform (df1[ 'LUNG_CANCER'])
df1['GENDER']=lb.fit_transform (df1[ 'GENDER'])
x=df1[[ 'GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', 'C
y=df1[ 'LUNG_CANCER']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.25)
from sklearn.tree import DecisionTreeClassifier
lr=DecisionTreeClassifier()

lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
print (f'first 5 actual_y: {y_test.values[:5]} predicted y: {y_pred[:5]}')
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
print('confusion_matrix')
a=confusion_matrix (y_test,y_pred)
print(a)
print('classification_report')
print(classification_report (y_test,y_pred))
print('accuracy_score-', accuracy_score (y_test,y_pred))
print('Model_score -',lr.score (x_test,y_test))
sns.heatmap(a, annot=True)
plt.show()
```

Random forest classifier

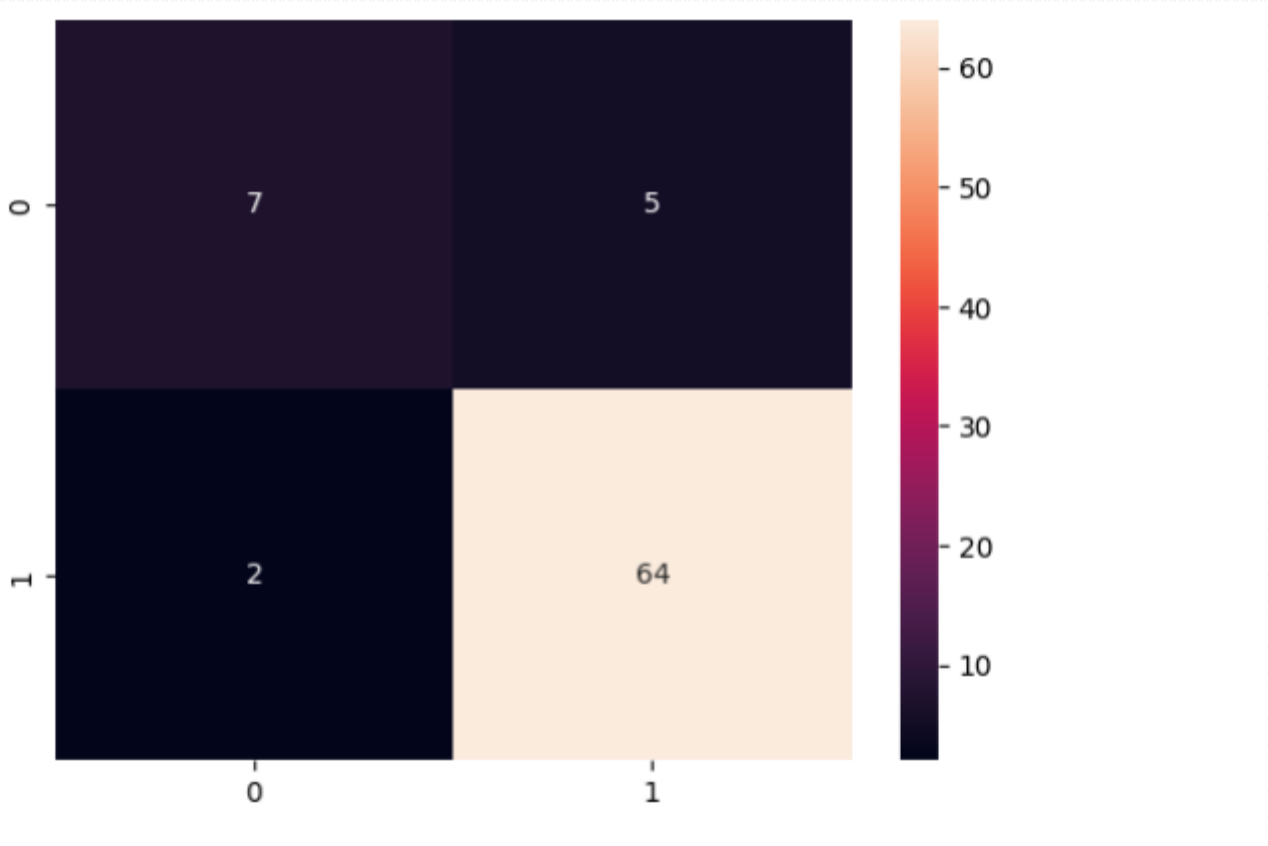
In [22]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df1['LUNG_CANCER']=lb.fit_transform (df1[ 'LUNG_CANCER'])
df1['GENDER']=lb.fit_transform (df1[ 'GENDER'])
x=df1[[ 'GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', 'C
y=df1[ 'LUNG_CANCER']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train, y_test=train_test_split(x,y,test_size=.25)
from sklearn.ensemble import RandomForestClassifier
lr=RandomForestClassifier()

lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
print (f'first 5 actual_y: {y_test.values[:5]} predicted y: {y_pred[:5]}')
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
print('confusion_matrix')
a=confusion_matrix (y_test,y_pred)
print(a)
print('classification_report')
print(classification_report (y_test,y_pred))
```

```
print('accuracy_score-', accuracy_score (y_test,y_pred))
print('Model_score -',lr.score (x_test,y_test))
sns.heatmap(a, annot=True)
plt.show()
```

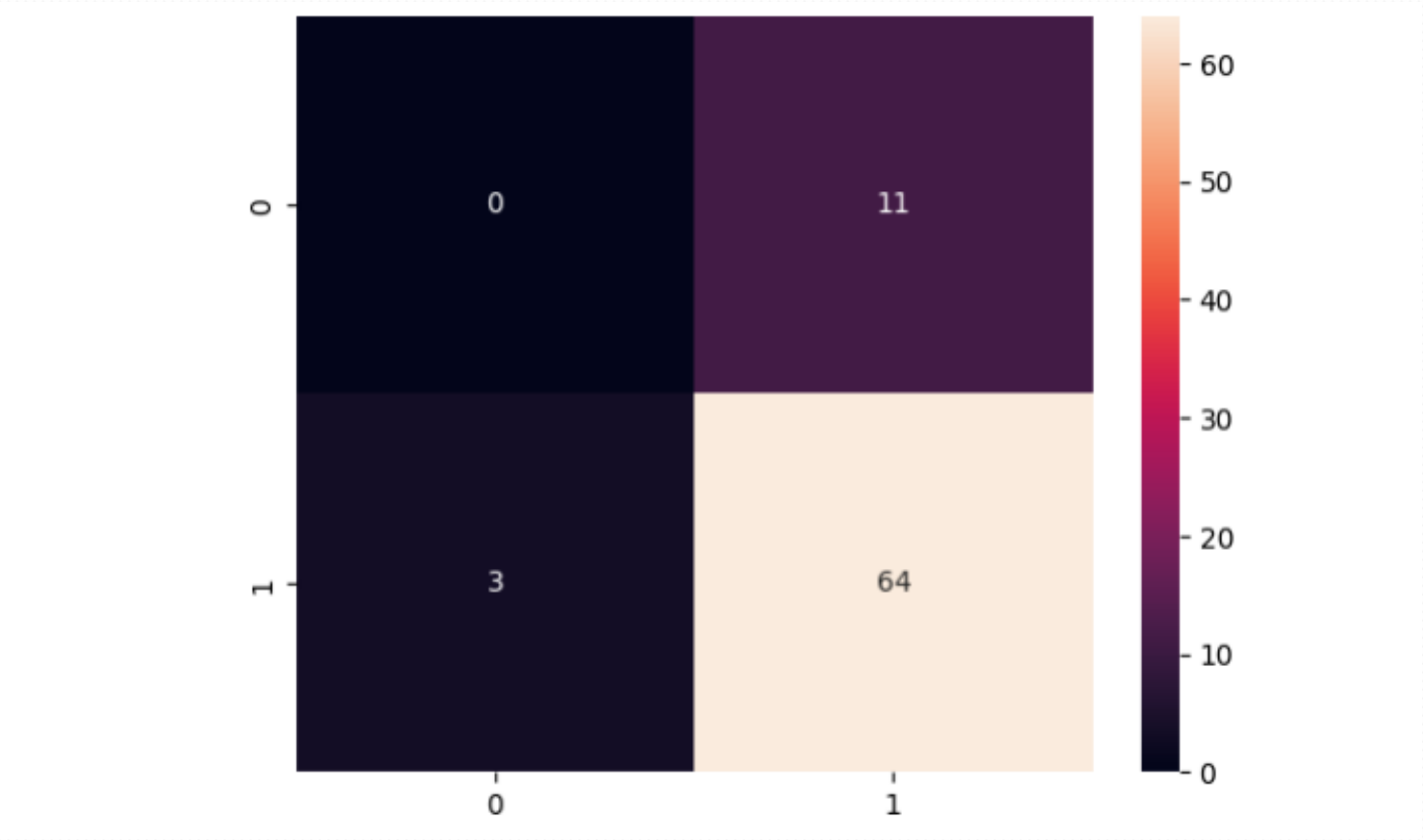
```
first 5 actual_y: [0 0 1 1 0] predicted y: [1 0 1 1 0]
confusion_matrix
[[ 7  5]
 [ 2 64]]
classification_report
              precision    recall  f1-score   support

           0       0.78      0.58      0.67        12
           1       0.93      0.97      0.95        66

    accuracy                           0.91        78
   macro avg       0.85      0.78      0.81        78
weighted avg       0.90      0.91      0.90        78

accuracy_score- 0.9102564102564102
Model_score - 0.9102564102564102
```

RESULT

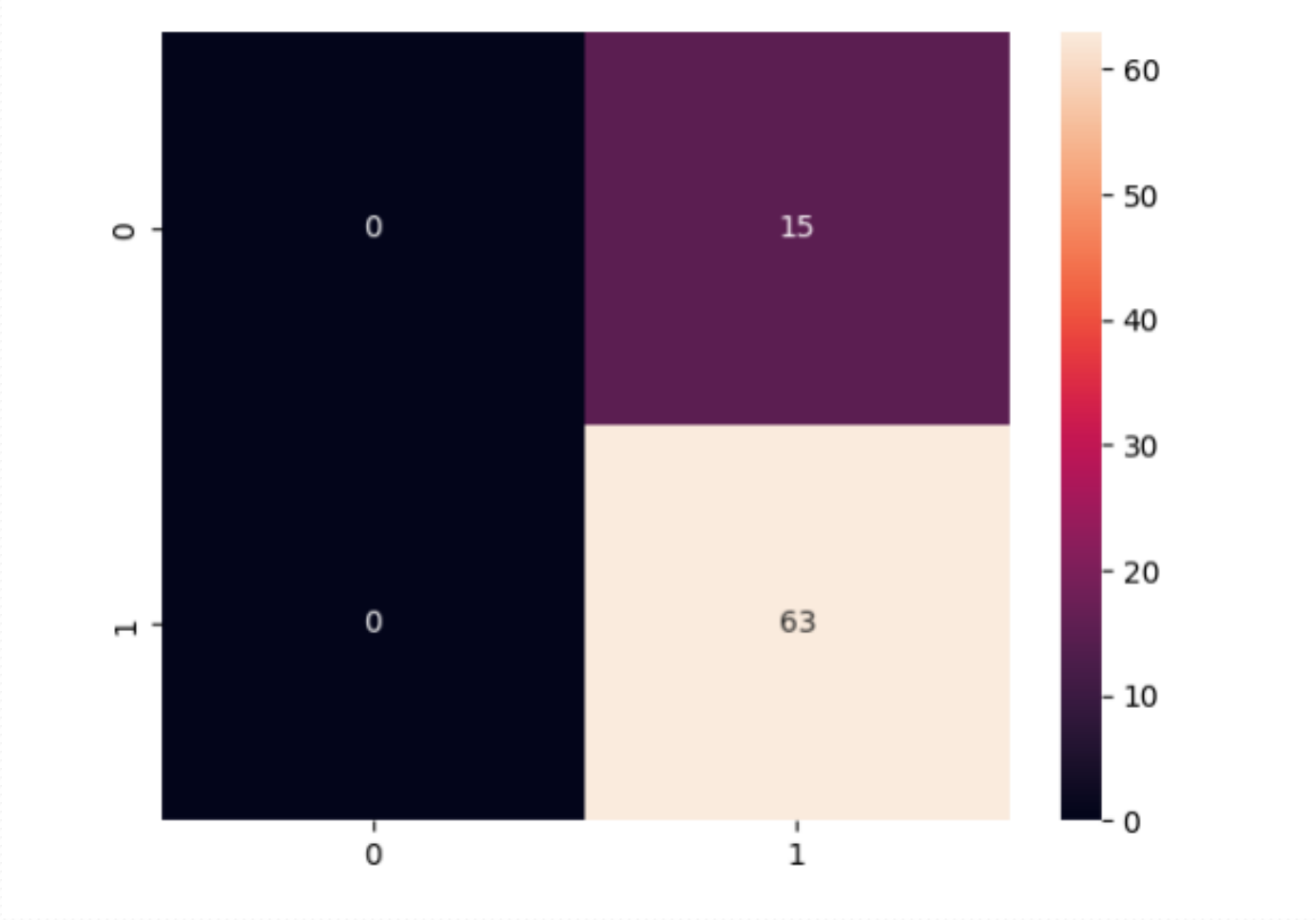
LINEAR REGRESSION:



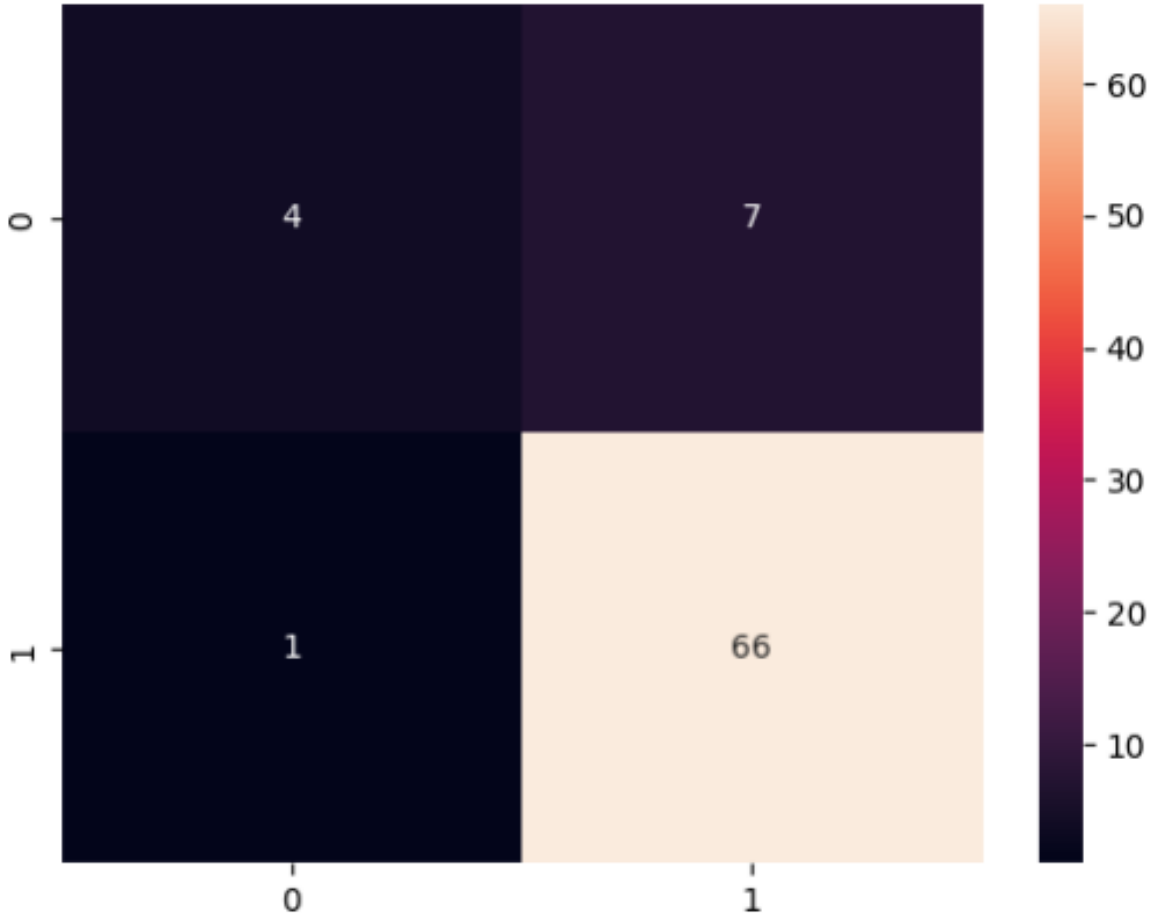
LOGISTIC REGRESSION



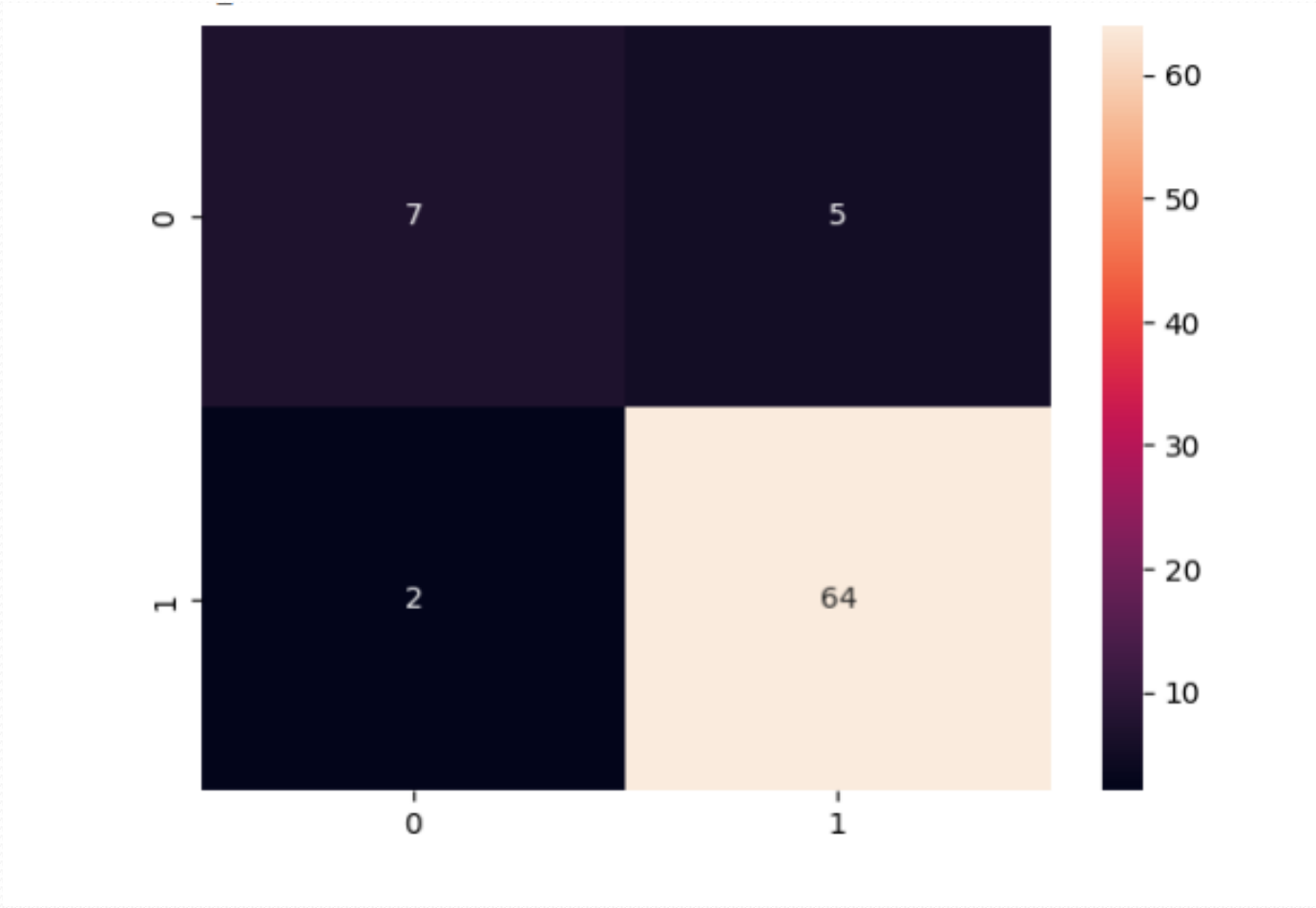
K NEAREST NEIGHBOUR(KNN)



DECISION TREE CLASSIFIER



RANDOM FOREST CLASSIFIER



CONCLUSION:

Here,Random Forest Classifier predict lung cancer with more accuracy. This result will help us to increase the accuracy of lung cancer prediction systems that use strong classification and prediction techniques.