CANCER DETECTION USING MACHINE LEARNING

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

G SANDHYARANI (19BCE7701)
G NANDINI (19BEC7704)

Under the Guidance of

DR. SELVA KUMAR S

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VIT-AP UNIVERSITY AMARAVATI- 522237

JANUARY 2023

CERTIFICATE

This is to certify that the Capstone Project work titled "**CANCER DETECTION USING MACHINE LEARNING**" that is being submitted by G **SANDHYARANI (19BEC7701), G NANDINI (19BCE7704)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Selva Kumar S

Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

PROGRAM CHAIR **DEAN**

B. Tech. CSE School Of Computer Science Engineering

ACKNOWLEDGEMENTS

We would like to express our heart full gratitude to Dr. Selva Kumar for his support and contribution of our project. Without his guidance we may not complete this project. His guidance always gave the confidence to us. He gave truth full reviews when we explained to him regarding this project. He gave confidence to complete this project. Programme chair gave the chance to do this project to improve our knowledge.

ABSTRACT

Cancer is a fatal illness that commonly results from the accumulation of genetic disorders and various pathological changes. Cancerous cells are abnormal areas that often grow in any part of the human body, which can be deadly if not noted and treated promptly. Cancer is also known as tumor and refers to any abnormal growth or lesion that may require medical intervention. The main goal of this research project is to analyze, review, categorize and address the current developments of human body cancer detection using machine learning techniques.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

Cancer refers to any one of numerous conditions characterized by the development of abnormal Cell that divide uncontrollably and can insinuate and destroy normal body towel. It'll spread into girding apkins. Cancer frequently could spread throughout your body. It's caused by inheritable changes in DNA. Cancer is the another leading cause of death in the world. But survival rates are perfecting for numerous types of cancer

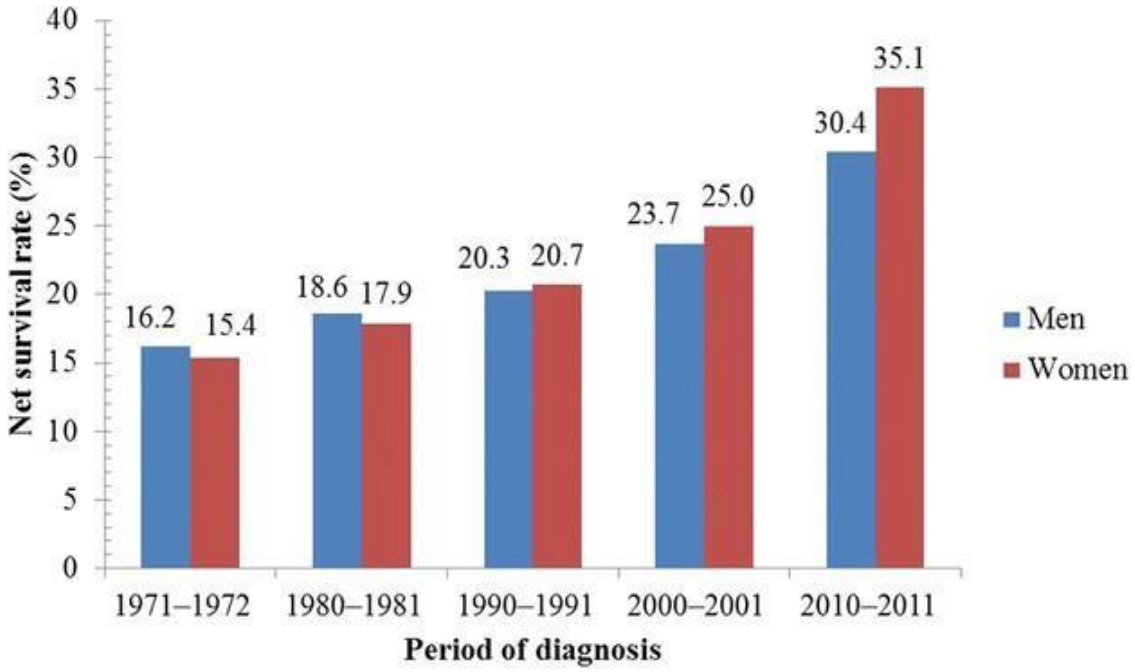


Figure 1 Period of Diagnosis

Detecting characteristic cases as early as possible in original stages is the stylish chance for successful treatment. When cancer care is delayed or inapproachable there's a lower chance of survival. several types of cancers for different body corridor to kill mortal. Cancer can be medical detected beforehand during a webbing examination through mammography or by movable cancer individual tool.

Over the once decades, a nonstop elaboration related to cancer exploration has been performed. With the arrival of innovative technologies in the field of drug, enormous quantities of cancer data have been collected and are available to the medical exploration community to train the data. still, the accurate prediction of a complaint outgrowth is one of the most intriguing and grueling tasks

Machine learning algorithms are trained to prognosticate cancer and its issues. In machine literacy, we use some algorithms which give the stylish delicacy rate by training the data to prognosticate the cancer at original stages. Those ways can discover and identify patterns and connections between them, from complex datasets, while they're suitable to effectively prognosticate unborn issues of a cancer type.

1.1 Objectives

The following are the objectives of this project:

- A machine learning algorithm trained to descry cancer issues zeroed in on the sanitarium where the excrescence image was taken, rather than the case's excrescence biology
- Machine literacy ways can be used to overcome these downsides which are cause due to the high confines of the data.
- In this design I'm using machine literacy algorithms to prognosticate the chances of getting cancer.
- The main goal of this design is to descry the cancer at early stages which helps to drop the cases.

1.2 Background and Literature Survey

It has discussed Cancer Detection analysis with different Machine Learning Algorithms Decision Tree, Random Forest Classification, SVM, KNN, K means Clustering and Naïve Bayes on the Online real time dataset is conducted and their performance was compared.Their experimental results shows that Random Forest Classification, KNN and SVM gives the highest delicacy(96) with high accuracy. They've executed within a simulation terrain and conducted in Python. The main end is to detect whether the cancer is benign or nasty and prognosticate whether it is coming in constantly or non-repeatedly of nasty cases after a certain period.

.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology, software details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

CANCER DETECTION USING MACHINE LEARNING

This Chapter describes the proposed system, working methodology, software details.

2.1 Proposed System

The following block diagram (figure 2) shows the system architecture of this project.

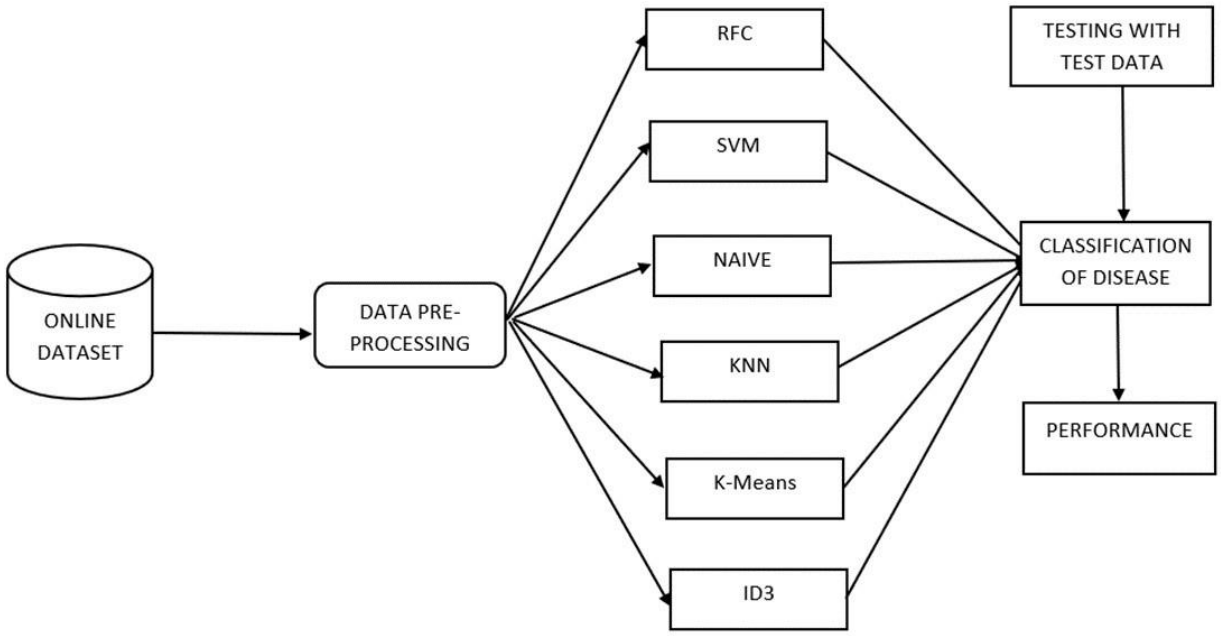


Figure 2 Architecture Diagram

2.2 Working Methodology

The system has only one sections, that is software. Software consists of Google colab which is helps to run the program code of python. Firstly, the dataset needs to be inserted into the google colab after that importing libraries to be done. Load the dataset using the python syntax using the path of dataset that is inserted.

Google Colab Software is monitored using all machine learning algorithms. Accuracy rate is monitored using Google colab and machine learning algorithms. Each algorithm will give different accuracy rates. Based on accuracy rates, we can detect the cancer at early stages with exactness. This google colab is used for monitoring the level of accuracy rates.

Random Forest Classification will give different accuracy rates compared to previous and the next accuracy rates.

2.3Standards

Various standards used in this project are:

- **Decision Tree Algorithm**

Decision tree could be used for both classification and Regression problems, but substantially it's preferred for working classification problems. The opinions or the test are performed based on features of the given dataset. It's a graphical representation for getting all the workable results to a problem/ decision based on given conditions. It generally mimics mortal thinking capability while deciding, so it's easy to understand. The sense behind the decision tree can be fluently understood because it shows a tree- such as structure. Decision trees tend to be the system of choice for prophetic modelling because they're easy to understand and are also largely effective. The introductory thing of a decision tree is to resolve a population of data into lower parts. There are two stages to prediction

- **Random Forest Classification**

Random forest is a combination of decision trees and it ensembles the classification model which is a process of combining multiple classifiers to classification a complex problem and to improve the performance of the model. Random forest model collects trained data from all the tree bumps and separates the weaker bumps training data to get better prognostications. Both classification and regression problems are answered using RF model. It takes the prediction from each tree and grounded on the maturity votes of prognostications it'll prognosticate the final affair. The main purpose we use for this is for the discovery which is used to produces good prognostications that can be understood fluently. It can handle large datasets efficiently. It'll give the stylish and advanced position of accuracy in prognosticating issues over the decision tree algorithm

- **K Means Clustering**

The K- means clustering algorithm is used to find groups which have not been explicitly labelled in the data. This can be used to confirm business hypotheticals about what types of groups live or to identify unknown groups in complex data sets. K Means Clustering is used to conversion of document to image segmentation. It's generally applied to the data that has a lower number of confines, numeric, and nonstop. This clustering helps to understand data in a unique way by grouping or dividing data into groups effects together into clusters. The main ideal of the K- Means is to minimise the sum of distances between the points and their separate cluster centroid.

- **K Nearest Algorithm**

KNN is one of the simplest forms of machine literacy algorithms substantially used for classification. It classifies the data point on how its neighbour is classified. It classifies the new data points based on the similarity measure of the before stored data points. It's distance- based it classifies objects based on their proximate neighbours' classes. KNN is most frequently used for classification, but can be applied to retrogression problems as well. It classifies the data point on how its neighbours are classified. KNN classifies the new data points based on the compatibility measure of the before stored data points

- **Naïve Bayes**

Naïve Bayes algorithms are used in sentiment analysis, spam filtering, recommendation systems, etc. They're quick and easy to apply. It doesn't bear as important training data. It is one of the simple and utmost effective Bracket algorithms which help in building the fast machine literacy models that can make quick prognostications. It's a probabilistic classifier. Naive Bayes predicts because of the probability of an object. It handles both nonstop and separate data. It's largely scalable with the number of predictors and data points. It's fast and can be used to make real- time prognostications

- **Support Vector Machine**

SVM is used in applications like handwriting recognition, face detection etc. It can handle both classification and regression on linear and non-linear data. It is used for both classification and regression. Though we say regression problems as well its best suited for classification also. It distinctly classifies the data points. This is one of the reasons we use SVMs in machine learning.

It is known to not suffer the condition of overfitting. Performance of SVM, and its generalization is better on the dataset. And lastly, SVM is known to have the best results for classification types of problems

2.4 System Details

This section describes the software details of the system:

2.4.1 Software Details

Google colab is used.

i) Google Colab

Google Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

Google Colab is simply an online representation of Jupyter Notebook. While Jupyter Notebook needs installation on a computer and can only use local machine resources, Colab is a full-fledged cloud app for Python coding. You can write Python codes using Colab on your Google Chrome or Mozilla Firefox web browsers.

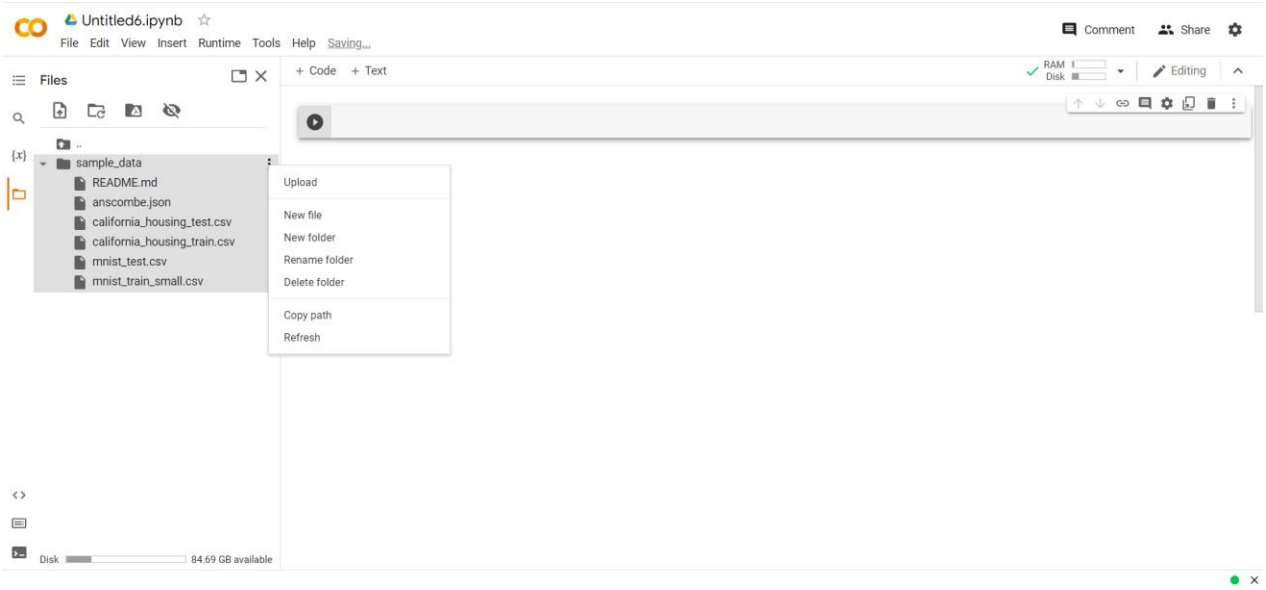


Figure 3 Google colab

- Likewise add dataset according to the algorithm of Machine Learning
- This colab will having features like adding heading, editing, copying the cell of code
- By showing figure 3, upload dataset and can add the code to load dataset and the code for accuracy rate for different algorithms and can add Heading as showing figure 4 and 5.

Figure 4 Google Colab Editor 1

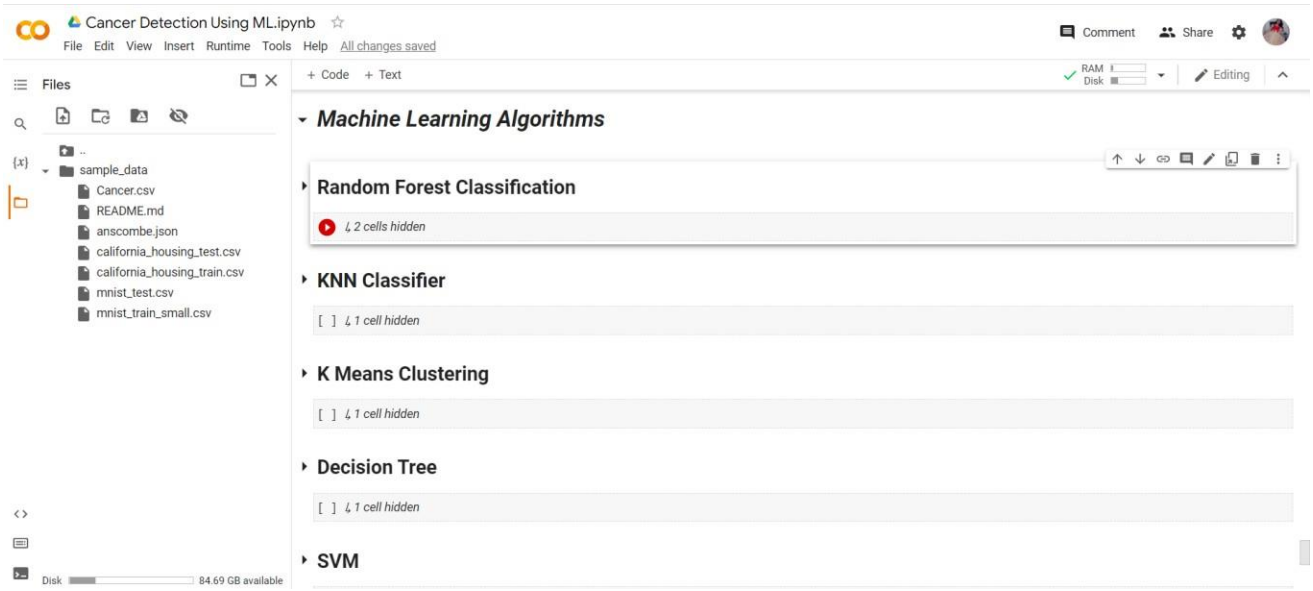


Figure 5 Google Colab Editor 2

ii) Database

Database will consist of 25 column's and 101 entries. It has several symptoms attributes like air pollution, Alcohols use, Dust Allergy, Genetic risk, Fatigue, Chest Pain, Obesity, Balanced Diet etc. which helps to detect the cancer at early stages.

Dataset provides a Realtime database where the user can understand the report and store under the particular attribute. The Realtime dataset provides an application data to be synchronized across the patient and stored on Dataset. The dataset can be accessible with python syntax as mention in the figure 6.

Connecting with Dataset

- After adding the screens, we will add buttons and create an authentication key.
- After appropriate authentication it is linked to the Google Firebase.

Figure 6 Connecting dataset

Now on google colab, make an account.

- It is also linked with the Google Drive app by creating a new colab notebook.
- A new folder is created for opening the google colab notebook on a purpose.

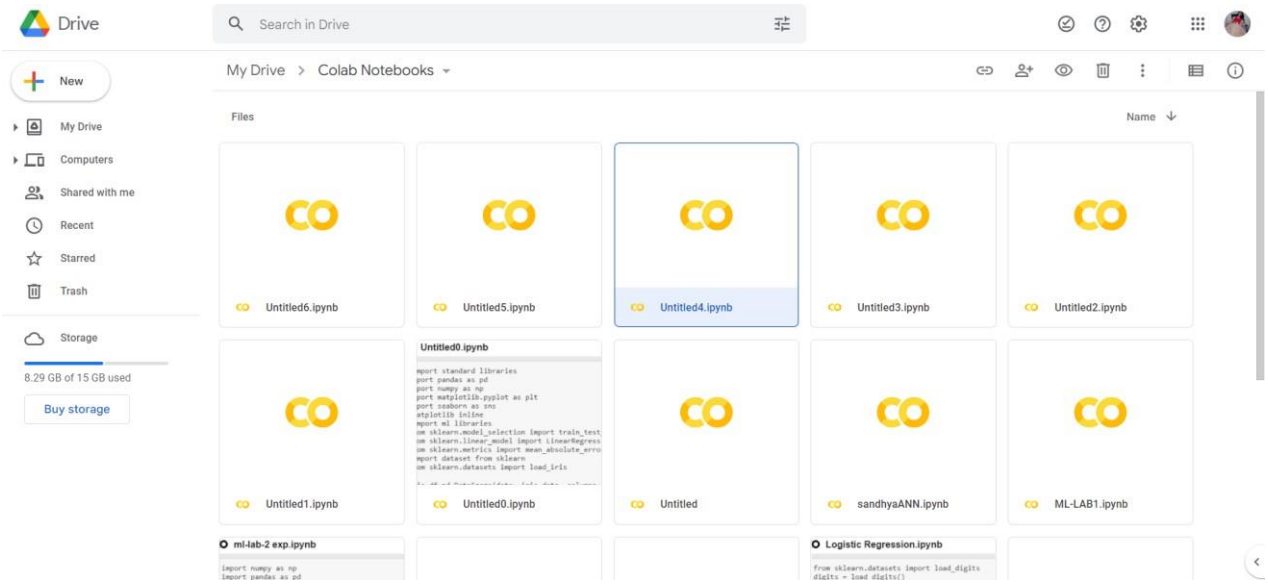


Figure 7 Google colab Notebooks

iii) Integration

Database is used for communication between patient and the system to detect the cancer. As to load or read the dataset is fairly fast, this system get to know very easily as soon as there is any symptom's to come. Whenever any patient requests to know the cancer with the help of dataset **is** appreciable. This system will monitor if there are any upcoming symptoms that are for cancer on Dataset. If it found any symptom's related to cancer System copies the data and let know the patient's account of user. **Python** is really a popular choice when it comes for any detection or prediction with the Machine Learning Algorithms. Python is processing all the data and sending it to the required tag in the Dataset. All the accuracy rate data is recorded at regular way.

CHAPTER 4

RESULTS AND DISCUSSIONS

a. Accuracy Rates

We had discussed that the Cancer Detection analysis was done with different Machine Learning Algorithms as mentioned below and the real time dataset is conducted and their performance was compared.

Algorithms	Accuracy Score
RFC	86.0
KNN	96.0
K Means	0.0
SVM	96.0
Decision Tree	84.0
Naïve Bayes	92.0

Table - 1 Accuracy rates

Their experimental results shows that KNN and SVM gives the highest accuracy (96) with high best accuracy rates. They have executed within a simulation environment and conducted in Python.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Cancer provides a unique context for medical diagnosis by considering the patient's condition and treatment response. It has been helped by machine learning. Despite technological advancements, accurate detection and monitoring of cancer detection remains a challenge. The biological and demographic streams of data must all be combined to improve prediction models. To propose machine learning classification, we examined the performance of basic logistic regression learning, random forest, decision tree, KNN k-means clustering and support vector machine learning with sequential minimum optimization, voting classifier, and convolutional neural network. The performance of these six classifiers was assessed using a variety of performance measures, precision of cancer, including accuracy rates like RFC gave (96%), KNN gave (96%) similarly SVM (96%), Naive Bayes (92%), Decision tree (84%), K- means Clustering (0%) respectively. Other cancer classification datasets could be used to test and improve the performance of the proposed method. These latest results can be used to classify breast tumors using images as a starting point. Cancer detection is important for nowadays. Machine learning algorithms are used to detect the cancer. Above proposed system is not only detected cancer but also predict the cancer by using those machine learning algorithms. Those algorithms detect to helps us to determine the cancer for approximately. By using these algorithms, we can detect the cancer easier so that we can reduce the cancer cases.

CHAPTER 6

APPENDIX

Python Code for Data Set Load

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import cufflinks as cf
import plotly
from plotly.offline import init_notebook_mode,iplot,plot
init_notebook_mode(connected=True)
cf.go_offline()

df = pd.read_csv('/content/sample_data/Cancer.csv')

df.head()
```

```
df.head()
```

	Patient Id	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	...	Fatigue	WeightLoss	ShortnessofBreath	Wheezing
0	P1	33	1	2	4	5	4	3	2	2	...	3	4	2	2
1	P10	17	1	3	1	5	3	4	2	2	...	1	3	7	8
2	P100	35	1	4	5	6	5	5	4	6	...	8	7	9	2
3	P1000	37	1	7	7	7	7	6	7	7	...	4	2	3	1
4	P101	46	1	6	8	7	7	7	6	7	...	3	2	4	1

5 rows × 25 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 25 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Patient Id             100 non-null    object
 1   Age                    100 non-null    int64
 2   Gender                 100 non-null    int64
 3   AirPollution           100 non-null    int64
 4   Alcoholuse             100 non-null    int64
 5   DustAllergy            100 non-null    int64
 6   OccuPationalHazards    100 non-null    int64
 7   GeneticRisk            100 non-null    int64
 8   chronicLungDisease     100 non-null    int64
 9   BalancedDiet           100 non-null    int64
 10  Obesity                100 non-null    int64
 11  Smoking                100 non-null    int64
 12  PassiveSmoker          100 non-null    int64
 13  ChestPain              100 non-null    int64
 14  CoughingofBlood        100 non-null    int64
 15  Fatigue                100 non-null    int64
 16  WeightLoss             100 non-null    int64
 17  ShortnessofBreath      100 non-null    int64
 18  Wheezing               100 non-null    int64
 19  SwallowingDifficulty   100 non-null    int64
 20  ClubbingofFingerNails  100 non-null    int64
 21  FrequentCold           100 non-null    int64
 22  DryCough               100 non-null    int64
 23  Snoring                100 non-null    int64
 24  Level                  100 non-null    object
dtypes: int64(23), object(2)
memory usage: 19.7+ KB
```

```
df.head()
```

	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	Obesity	...	Fatigue	WeightLoss	ShortnessofBreath	Wheezing	SwallowingDifficulty
0	33	1	2	4	5	4	3	2	2	4	...	3	4	2	2	3
1	17	1	3	1	5	3	4	2	2	2	...	1	3	7	8	6
2	35	1	4	5	6	5	5	4	6	7	...	8	7	9	2	1
3	37	1	7	7	7	7	6	7	7	7	...	4	2	3	1	4
4	46	1	6	8	7	7	7	6	7	7	...	3	2	4	1	4

5 rows × 24 columns

```
df['Level'].replace('Medium','High',inplace=True)
```

```
0      Low
1     High
2     High
3     High
4     High
      ...
95    High
96    High
97    High
98    High
99    High
Name: Level, Length: 100, dtype: object
```

```
df['Level'].replace('High','1',inplace=True)
df['Level'].replace('Low','0',inplace=True)

df.head()
df['Level'] = pd.to_numeric(df['Level'])

df.isnull()
```

	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	Obesity	...	Fatigue	WeightLoss	ShortnessofBreath	Wheezing	SwallowingDifficul
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
96	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
97	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
98	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal
99	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	Fal

100 rows × 24 columns

```
df.isnull().any()
```

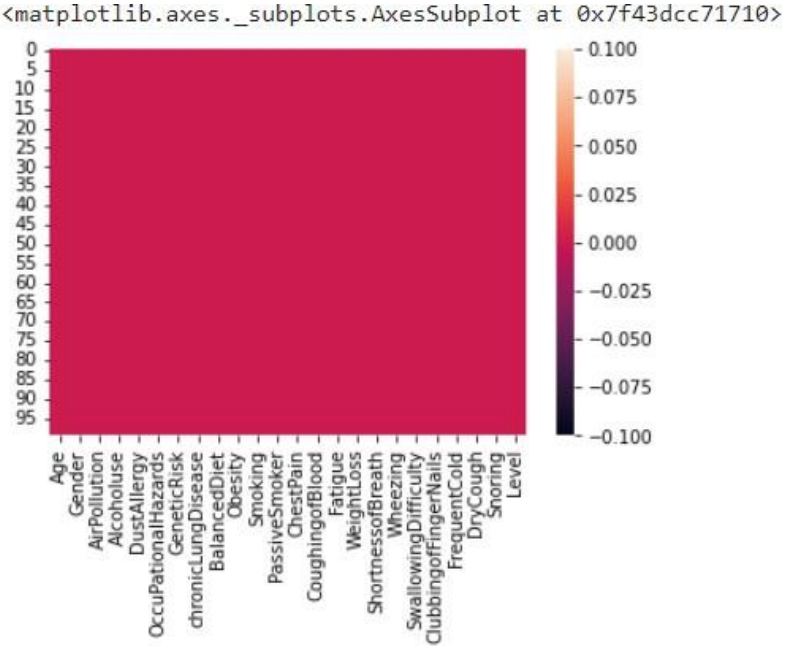
```
Age                     False
Gender                  False
AirPollution            False
Alcoholuse              False
DustAllergy             False
OccuPationalHazards     False
GeneticRisk             False
chronicLungDisease      False
BalancedDiet            False
Obesity                 False
Smoking                 False
PassiveSmoker           False
ChestPain               False
CoughingofBlood         False
Fatigue                 False
WeightLoss              False
ShortnessofBreath       False
Wheezing                False
SwallowingDifficulty    False
ClubbingofFingerNails   False
FrequentCold            False
DryCough                False
Snoring                 False
Level                   False
dtype: bool
```

```
df.isnull().sum()
```

```
Age                       0
Gender                    0
AirPollution              0
Alcoholuse                0
DustAllergy               0
OccuPationalHazards       0
GeneticRisk               0
chronicLungDisease        0
BalancedDiet              0
Obesity                   0
Smoking                   0
PassiveSmoker             0
ChestPain                 0
CoughingofBlood           0
Fatigue                   0
WeightLoss                0
ShortnessofBreath         0
Wheezing                  0
SwallowingDifficulty      0
ClubbingofFingerNails     0
FrequentCold              0
DryCough                  0
Snoring                   0
Level                     0
dtype: int64
```

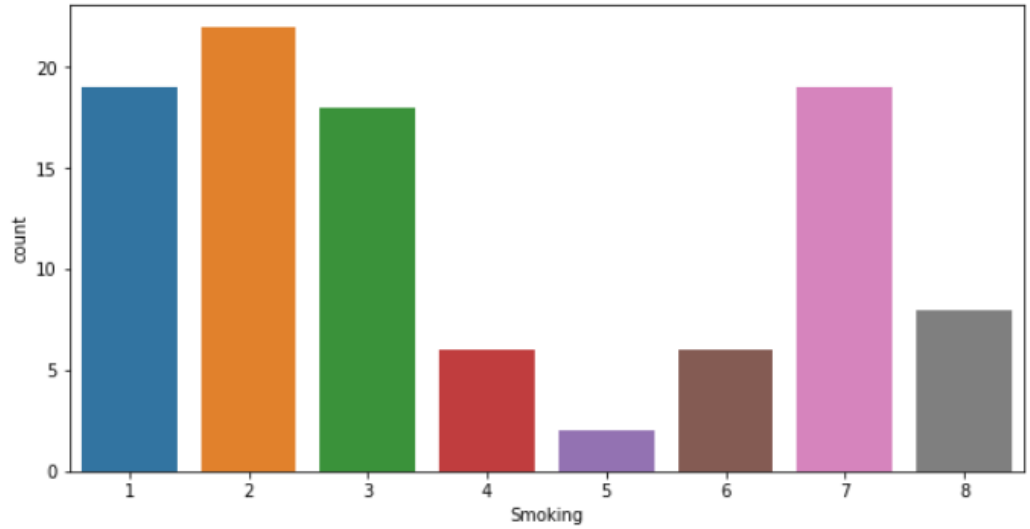
```
import seaborn as sns
sns.heatmap(df.isnull())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f43dcc71710>
```



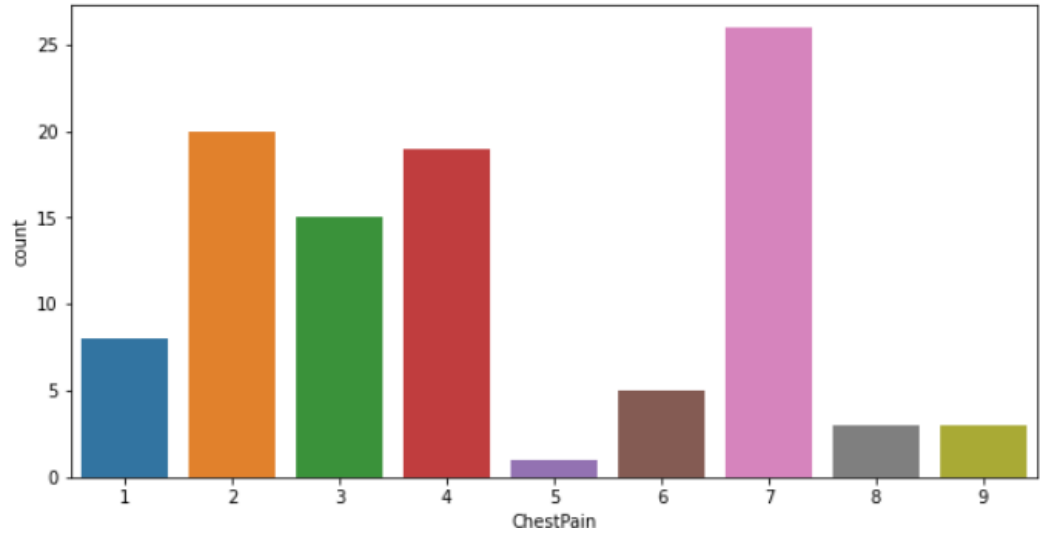
```
plt.figure(figsize=(10,5))
sns.countplot(x='Smoking',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f43dcf83990>
```



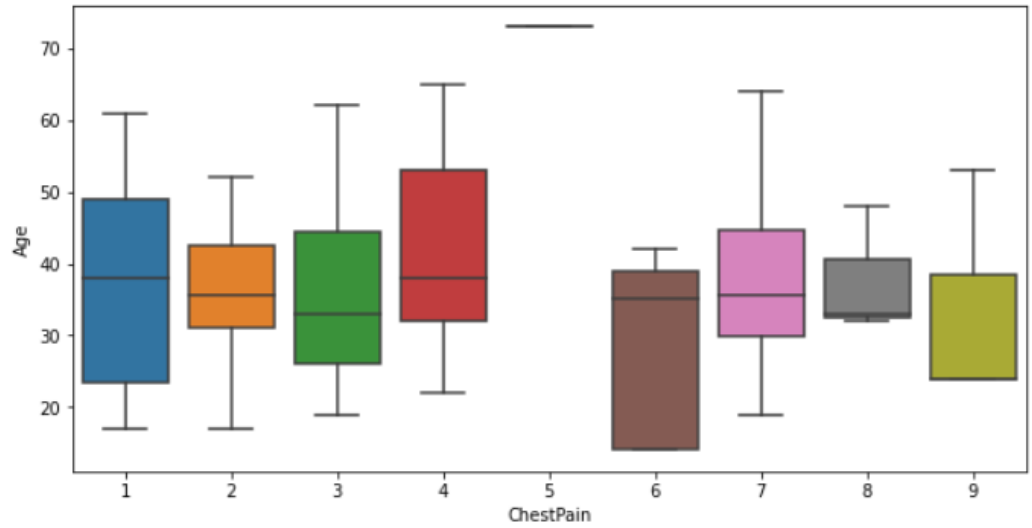

```
plt.figure(figsize=(10,5))
sns.countplot(x='ChestPain',data = df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f43da52b9d0>
```



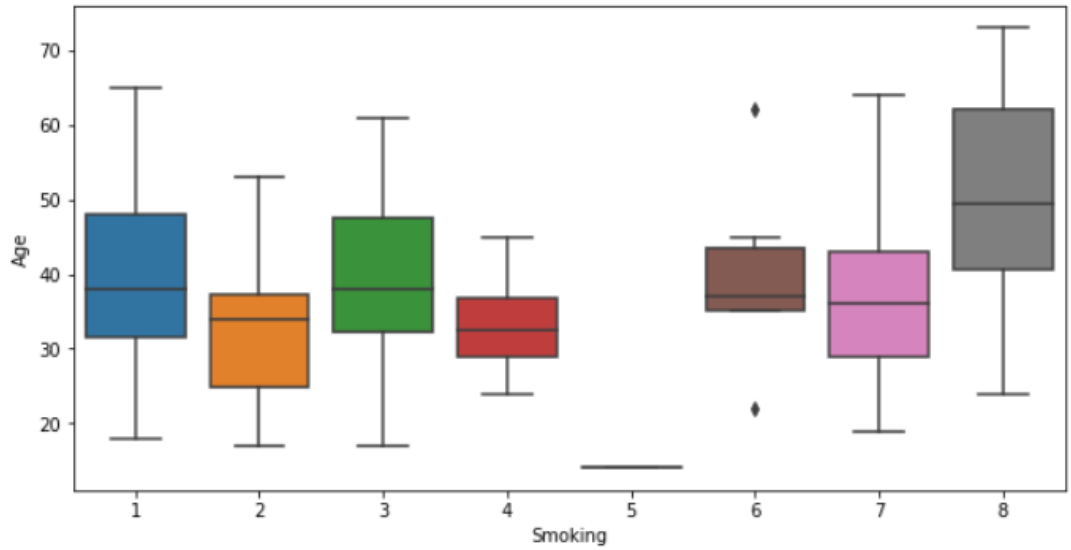
```
plt.figure(figsize=(10,5))
sns.boxplot(x='ChestPain',y='Age',data = df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f43da51fe90>
```




```
plt.figure(figsize=(10,5))
sns.boxplot(x='Smoking',y='Age',data = df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f43daad17d0>
```




```
sorted_smokers = df.groupby('Age')['Smoking'].count().to_frame()
sorted_smokers.style.background_gradient(cmap = 'Reds')
```

Age	
14	2
17	3
18	1
19	2
22	2
23	2
24	4
25	1
26	2
27	4
28	4
29	3
31	1
32	3
33	6
34	2
35	9
36	3
37	2

```
df.style.background_gradient(cmap = 'Reds')
```

	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	Obesity	Smoking	PassiveSmoker	ChestPain	CoughingofBlood	Fatigue	WeightLoss
0	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4
1	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3
2	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7
3	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2
4	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2
5	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7
6	52	2	2	4	5	4	3	2	2	4	3	2	2	4	3	4
7	28	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2
8	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	4
9	46	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2
10	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3
11	64	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6
12	39	2	4	5	6	6	5	4	6	6	6	6	6	6	5	3
13	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2
14	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2
15	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	3
16	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3
17	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2
18	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5
19	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3
20	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2

```
label = df.Age.sort_values().unique()
target = sorted_smokers.Smoking

print(label)
print(target)
```

```
[14 17 18 19 22 23 24 25 26 27 28 29 31 32 33 34 35 36 37 38 39 42 44 45
 46 47 48 52 53 55 61 62 64 65 73]
Age
14    2
17    3
18    1
19    2
22    2
23    2
24    4
25    1
26    2
27    4
28    4
29    3
31    1
32    3
33    6
34    2
35    9
36    3
37    2
38    7
39    1
42    2
44    5
45    4
46    4
47    1
48    3
52    4
53    3
55    1
61    1
62    4
64    1
65    2
73    1
```

```
import plotly.graph_objects as go

fig = go.Figure()
fig.add_trace(go.Bar(x=label,y=target))
fig.update_layout(title = 'Smokers per age',xaxis=dict(title='Age'),yaxis=d
ict(title='Smokers'))
fig.show()
```

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=label,y=target,mode='markers+lines'))
fig.update_layout(title = 'Smokers per age',xaxis=dict(title='Age'),yaxis=d
ict(title='Smokers'))
fig.show()
```

Machine Learning Algorithms

Random Forest Classification

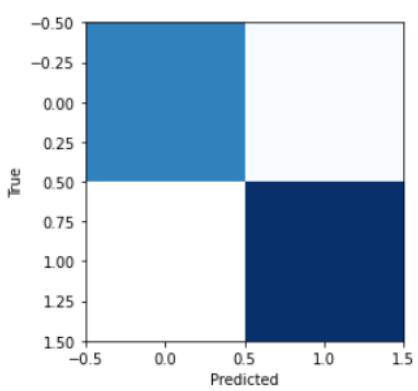
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import log_loss, f1_score
from sklearn.model_selection import cross_val_score
import numpy as np
acc_dict = {}
# create the data
X = df.drop('Level',axis = 1)
y = df['Level']
X_train, X_test, y_train, y_test = train_test_split(X,y)


from sklearn.ensemble import  RandomForestClassifier
# create model
model = RandomForestClassifier()
# fit the data in the model
model.fit(X_train,y_train)
y_pred_randomF = model.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test, y_pred_randomF)*100)
acc_dict['RFC_log_loss'] = log_loss(y_test, y_pred_randomF)
acc_dict['RFC_F!1_Score'] = f1_score(y_test, y_pred_randomF,average='weight
ed')
# prediction visualization
plt.imshow(np.log(confusion_matrix(y_test,y_pred_randomF)),cmap = 'Blues',i
nterpolation = 'nearest')
plt.ylabel('True')
plt.xlabel('Predicted')
plt.show()
```

```
Accuracy score :  84.0
-----------------------------------------
```

KNN Classifier

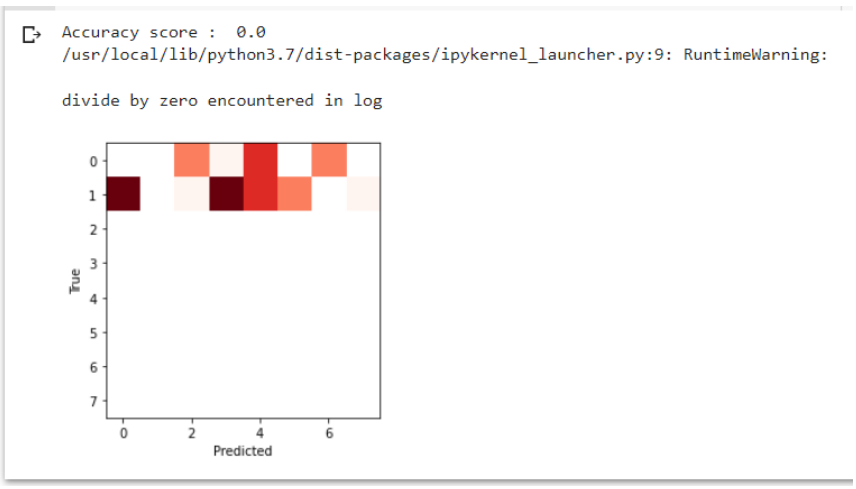
```
from sklearn.neighbors import KNeighborsClassifier
# to find the best k
score = 0
scores, highscore, bestk = 0, 0, 0

for k in range(3,12):
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X_train, y_train)
    score = scores.mean()
    if score>highscore:
        highscore = score
        bestk = k
print('Best k is {} with score {}'.format(bestk, highscore))

knn = KNeighborsClassifier(n_neighbors=bestk)
knn.fit(X_train,y_train)
# prediction
y_predict = knn.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test,y_predict)*100)
acc_dict['KNN_log_loss'] = log_loss(y_test, y_predict)
acc_dict['KNN_F!1_Score'] = f1_score(y_test, y_predict,average='weighted')

# prediction visualization
plt.imshow(np.log(confusion_matrix(y_test,y_predict)),cmap = 'Blues',interp
olation = 'nearest')
plt.ylabel('True')
plt.xlabel('Predicted')
plt.show()
```

```
Best k is 5 with score 0.8933333333333333
Accuracy score :  96.0
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:24: RuntimeWarning:

divide by zero encountered in log
```



K Means Clustering

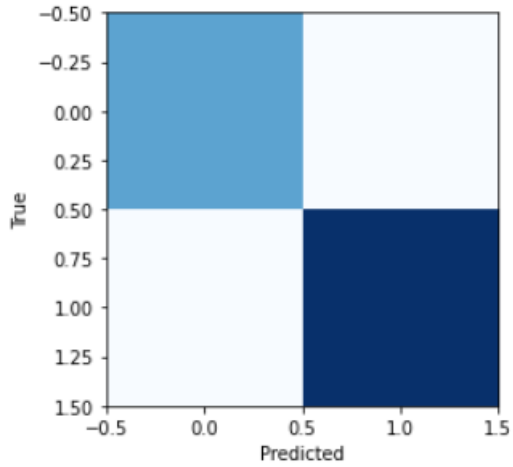
```
from sklearn.cluster import KMeans
clf = KMeans()
clf.fit(X_train)
maxx = clf.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test,maxx)*100)
acc_dict['kMeans_log_loss'] = log_loss(y_test, maxx)
acc_dict['kMeans_F1_Score'] = f1_score(y_test, maxx,average='weighted')

plt.imshow(np.log(confusion_matrix(y_test,maxx)),cmap='Reds', interpolation
 = 'nearest')
plt.ylabel('True')
plt.xlabel('Predicted')

plt.show()
```



Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
tree_ = DecisionTreeClassifier()
tree_.fit(X_train,y_train)
y_pred = tree_.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test, y_pred)*100)
acc_dict['Tree_log_loss'] = log_loss(y_test,y_pred)
acc_dict['Tree_f!1_score'] = f1_score(y_test,y_pred)

# prediction visualization
plt.imshow(np.log(confusion_matrix(y_test,y_pred)),cmap = 'Blues',interpola
tion = 'nearest')
plt.ylabel('True')
plt.xlabel('Predicted')
plt.show()
```

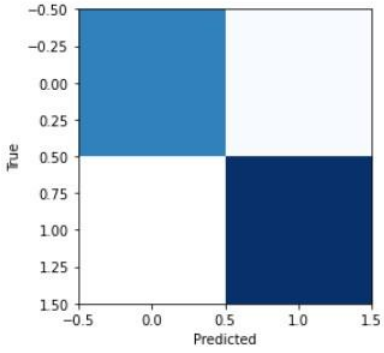
```
Accuracy score :  84.0
```



SVM:

```
from sklearn.svm import SVC
model = SVC()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print('Accuracy score : ',accuracy_score(y_test, y_pred)*100)
acc_dict['svc_log_loss'] = log_loss(y_test,y_pred)
acc_dict['svc_f!1_score'] = f1_score(y_test,y_pred)
# prediction visualization
plt.imshow(np.log(confusion_matrix(y_test,y_pred)),cmap = 'Blues',interpola
tion = 'nearest')
plt.ylabel('True')
plt.xlabel('Predicted')
plt.show()
```

```
Accuracy score :  96.0
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: RuntimeWarning:

divide by zero encountered in log
```



Naïve Bayes:

```
import numpy as np
import pandas as pd
dataset = pd.read_csv('/content/sample_data/Cancer12.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
 random_state =0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
# Classifier model
GaussianNB(priors=None, var_smoothing=1e-09)
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = classifier.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[8 0 1]
 [0 8 0]
 [1 0 7]]
0.92
```

Drive link for the Source Code and for the Dataset

https://drive.google.com/drive/folders/1U1ABgn9-bIOzh60YNsK7Htqi3W9KBWR_?usp=share_link

REFERENCES

[1] S. Gc, R. Kasaudhan, T. K. Heo, and H.D. Choi, “Variability Measurement forBreast Cancer Classification Mammographic adaptive and convergent systems (RACS), Prague, Czech Republic, 2015, pp. 177–182. [2] S. Hafizah, S. Ahmad,R. Sallehuddin, and N. Azizah “CancerDetection Using Artificial Neural Networka Support Vector Machine.

[2] A Comparative Study,” J. Teknol, vol. 65, pp. 73–81, 2013. [3] A.T. Azar, andS.El – Said” Performance analysis of support vector Neural Compute. Appl., vol. 24, no. 5, pp. 1163– 1177, 2014. [4] machines classifiers in breast cancer mammography recognition,” Neural Comput. Appl., vol. 24, no. 5, pp. 1163– 1177, 2014. [5] C. Deng, and M. Perkowski, “A Novel Weighted Hierarchical Adaptive Voting Ensemble Machine Learning Method

[3] Hsu, C.-C., Wang, K.-S. and Chang, S.-H. (2011) Baye- sian decision theory for support vector machines: Imba- lance measurement and feature optimization. Expert Sys- tems with Applications, 38, 4698-4704. doi: 10.1016/j.eswa.2010.08.150

[4] Koch, K.-R. (2007) Introduction to naves Bayesian statistics. Springer, NewYork, 2007. [32] Brase, C.H. and Brase, C.P. (2012) Understanable statis- tics. 10th Edition, Cengage Learning, StamfordFan Z, Ji T, Wan S, Wu Y, Zhu Y, Xiao F, et al. Smoking and risk of meningi-oma: a meta- analysis. Cancer Epidemiol. 2013;37(1):39–45.Ferlay J, Soerjomataram I, Dikshit R, Eser S, Mathers C, Rebelo M, et al. Cancerincidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012. Int J Cancer. 2015;136(5):E359–86Dubey AK, Gupta U, Jain S. Breast cancer statistics and prediction methodology: asystematic review and analysis. Asian Pac J Cancer Prev. 2015;16(10):4237–45.

BIODATA

Name: Gorla Nandini

Mobile No.: 7386438558

E-Mail: gorlanandini8@gmail.com

Address: 2-14, Kalluru (v), Yellanuru (m), Anantapur (Dist.)

Name: Gulivindala Sandhyarani

Mobile No.: 6281493556

E-Mail: sandhya.19bce7701@vitap.ac.in

Address: Kuntibhadra (v). Kotturu (m), Srikakulam (Dist.).