PRODICTION

PREDICTION AND DETECTION OF CANCER DISEASE

A project report submitted to
MALLA REDDY UNIVERSITY
in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLGY
in
COMPUTER SCIENCE & ENGINEERING (AI & ML)

Submitted by

Y.Chandra Shekar 2011CS020435

Under the Guidance of

N.V.P.R. Rajeswari

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI & ML)

MALLA REDDY UNIVERSITY
(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

2023

MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

COLLEGE CERTIFICATE

This is to certify that this is the bonafide record of the application development entitled, **Prediction and Detection of Cancer disease in Human beings Using Machine Learning** Submitted by Y. Chandra Shekar (2011CS020435), B. Tech III year I semester, Department of CSE(AI&ML) during the year 2022-23. The results embodied in the report have not been submitted to any other university or institute for the award of any degree or diploma

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
N.V.P.R. **Rajeswari**	**Dr.Thayyaba Khatoon**
Assistant **Professor**	**CSE(AI&ML)**

ABSTRACT

Machine learning is used in almost all the medical fields by the diagnostics and doctors especially in predicting and detecting the risk of cancer. This growing trend of machine learning utilization in this approach enables the researchers to survey on the various types and approaches of machine learning . Machine learning is increasingly being employed in cancer detection and diagnosis. Cancer prediction will become quite easy in the future and we can predict it without the need of going to the hospitals. As we can see many technologies are being used and tested in the medical field. So, by this we can say that this will make us easier in the future to detect cancer.

CONTENTS

Chapter 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Cancer is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body .When cells grow old or become damaged, they die, and new cells take their place. Sometimes this orderly process breaks down, and abnormal or damaged cells grow and multiply when they shouldn't. These cells may form tumors, which are lumps of tissue. Tumors can be cancerous or not cancerous (benign) .Using Machine Learning applications, we try to detect Cancer disease symptoms at early stages.

1.2 OBJECTIVE OF PROJECT

- Disease will be predicted using K Neighbors Classifier which works on probabilistic approach
- The goal is to classify whether the cancer is benign or malignant. To achieve this we have used machine learning classification methods to fit a function that can predict the discrete class of new input.
- To review on various state-of-the-art cancer prediction models and develop a new feature extraction model.
- Compare the symptoms of cancer for early notification.
- To design and develop a Machine learning model to predict the cancer.
- To validate the proposed model by comparing it with other conventional models.

1.3 LIMITATIONS OF PROJECT

- Time consuming
- Complicated process

Chapter 2

ANALYSIS

2.1 INTODUCTION

- Cancer is a tumor which refers to the abnormal growth of new tissues. Tumor can be classified benign or malignant. Different kinds of cancer such as breast cancer, skin cancer, lung cancer, colon cancer etc. can be diagnosed based on the part of the body which results in growth of abnormal cells.

- Today many conventional methods are available to detect the cancer but these are expensive and time consuming leading to death of patient due to late diagnosis. So it is very significant to diagnose the cancer in its preliminary stage which is the only way to save a patient life.

- To deal with these issues, K Neighbors Classifier has been appliedto medical applications which are more intelligent than conventional techniques. It has become a robust tool in medical world in solving different acute diseases

2.2 SOFTWARE REQUIREMENT SPECIFICATION

2.2.1 Software Requirement

- Jupyter Notebook
- VS Code

2.2.2 Hardware Requirement

- 8 GB RAM
- 128 GB ROM
- PROCESSOR ABOVE 1.4 GHz

2.3 EXISTING SYSTEM

- Because of high quantity data in CT images and blurred boundaries, tumor segmentation and classification is very hard.

- In MR images, the amount of data is too much for manual interpretation and analysis.

- Issue is extracted and then analyzed to see if the tissue is cancerous or not, now this is very invasive, very uncomfortable and very costly for everyone

2.4 PROPOSED SYSTEM

- This work as automatic cancer detection method to increase the accuracy and yield and decrease the diagnosis time.

- The goal is classifying the tissues to three classes of normal, benign and malignant.

- Accurate detection of size and location of cancer plays a vital role in the diagnosis of cancer.

2.5 MODULES

Data Pre-processing Module

Data Pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data Pre- processing task

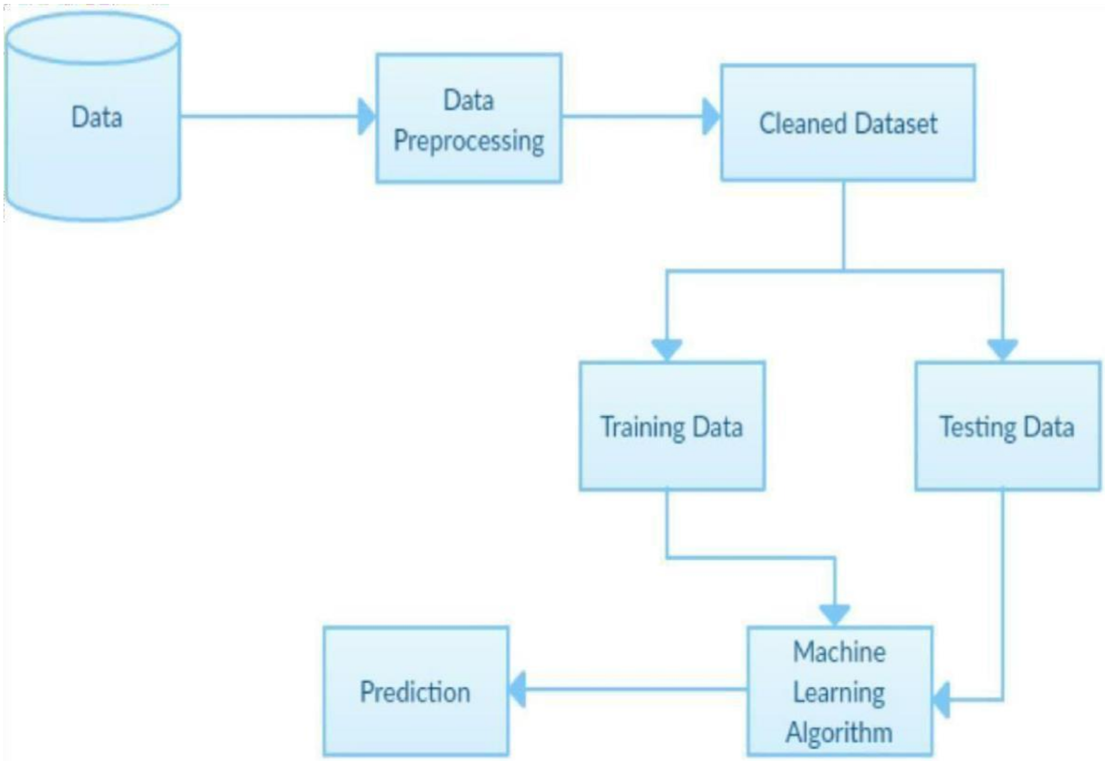
Data Visualization Module

Visualising the data for data analysis. We can find the relations between the attributes and can work on them to make any necessary changes on our training data.

Training and Testing

In this Module we will train the system using K Neighbors Classifier . Using the training Model the system will produce the classifies the testing data

2.6 ARCHITECTURE



Chapter 3

DESIGN

3.1 INTRODUCTION

Cancer is a class of diseases characterized by out-of-control cell growth. Cancer harms the body when damaged cells divide uncontrollably to form lumps or masses of tissue called tumors (except in the case of leukemia where cancer prohibits normal blood function by abnormal cell division in the blood stream). Tumors can grow and interfere with the digestive, nervous, and circulatory systems and they can release hormones that alter body function. Tumors that stay in one spot and demonstrate limited growth are generally considered to be benign.

There are over 100 different types of cancer, and each is classified by the type of cell that is initially affected.

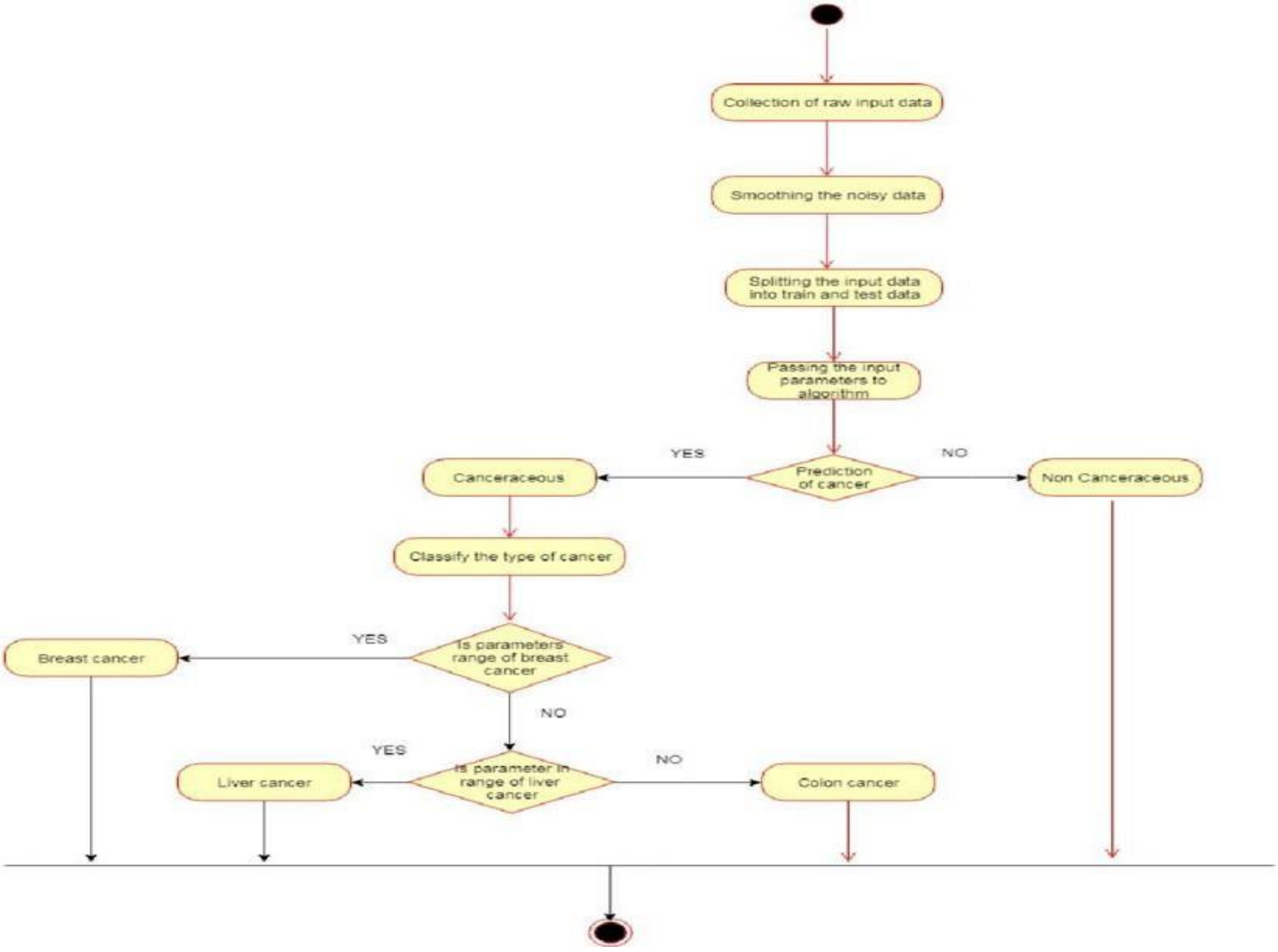
More dangerous, or malignant, tumors form when two things occur:

1. A cancerous cell manages to move throughout the body using the blood or lymph systems, destroying healthy tissue in a process called invasion

2. That cell manages to divide and grow, making new blood vessels to feed itself in a process called angiogenesis.

Many things are known to increase the risk of cancer, including tobacco use, dietary factors, certain infections, exposure to radiation, lack of physical activity, obesity, and environmental pollutants .These factors can directly damage genes or combine with existing genetic faults within cells to cause cancerous mutations. Approximately 510% of cancers can be traced directly to inherited genetic defects. Many cancers could be prevented by not smoking, eating more vegetables, fruits and whole grains, eating less meat and refined carbohydrates, maintaining a healthy weight, exercising, minimizing sunlight exposure, and being vaccinated against some infectious diseases.

3.2 UML diagram


Collection of raw input data
Smoothing the noisy data
Splitting the input data into train and test data
Passing the input parameters to algorithm
YES
NO
Prediction of cancer
Canceraceous
Non Canceraceous
Classify the type of cancer
YES
Is parameters range of breast cancer
Breast cancer
NO
YES
Is parameter in range of liver cancer
NO
Liver cancer
Colon cancer

3.3 Data set description

ML, a branch of Artificial Intelligence, relates the problem of learning from data samples to the general concept of inference .Every learning process consists of two phases:
(i) Estimation of unknown dependencies in a system from a given dataset and
(ii) Use of estimated dependencies to predict new outputs of the system.

ML has also been proven an interesting area in biomedical research with many applications, where an acceptable generalization is obtained by searching through an *n-dimensional* space for a given set of biological samples, using different techniques and algorithms. There are two main common types of ML methods known as
(i) Supervised learning
(ii) Unsupervised learning
In supervised learning a labeled set of training data is used to estimate or map the input data to the desired output. In contrast, under the unsupervised learning methods no labeled examples are provided and there is no notion of the output during the learning process. As a result, it is up to the learning scheme/model to find patterns or discover the groups of the input data.

3.4 Data Pre Processing Techniques:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

3.5 Methods and Algorithm:

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another. The k value in the k-NN algorithm defines how many neighbors will be checked to determine the classification of a specific query point. For example, if k=1, the instance will be assigned to the same class as its single nearest neighbor. Defining k can be a balancing act as different values can lead to overfitting or under fitting . Lower values of k can have high variance, but low bias, and larger values of k may lead to high bias and lower variance. The choice of k will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k. Overall, it is recommended to have an odd number for k to avoid ties in classification, and cross-validation tactics can help you choose the optimal k for your dataset.

3.6 Building a model:

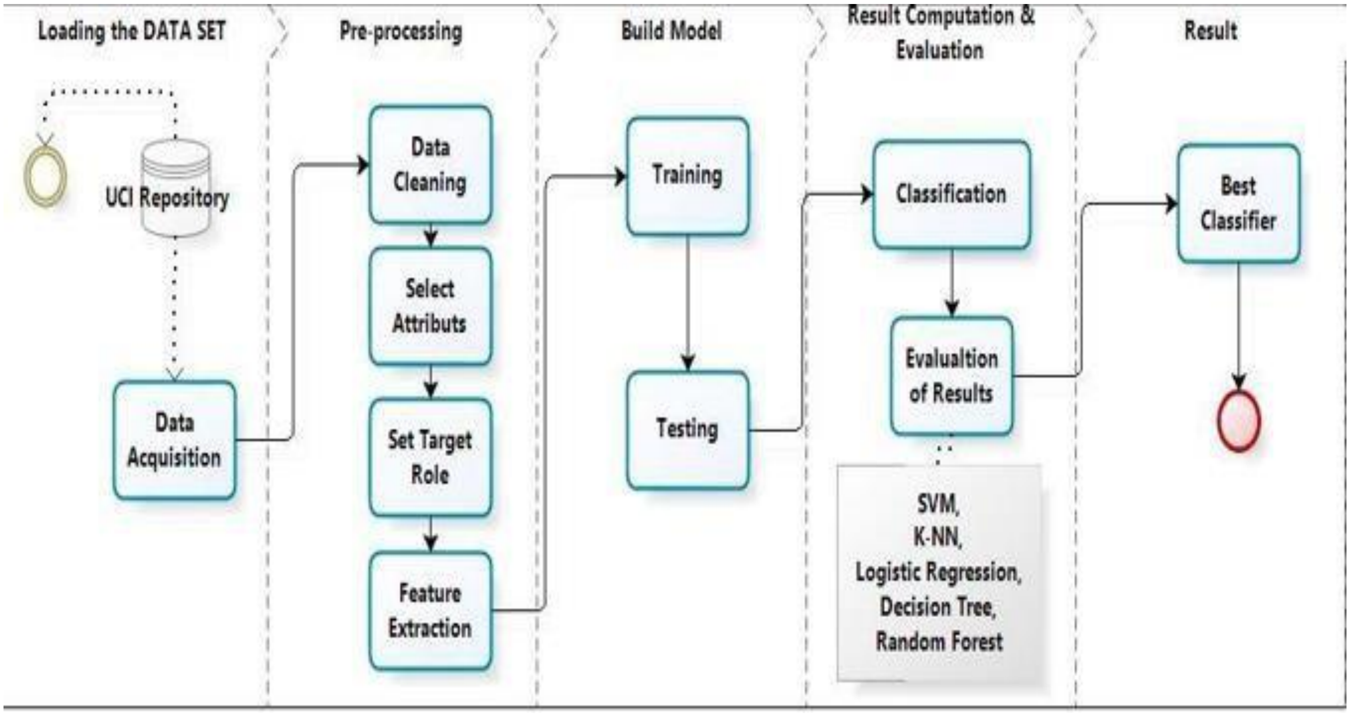


Figure 3.6

3.7 Evalution

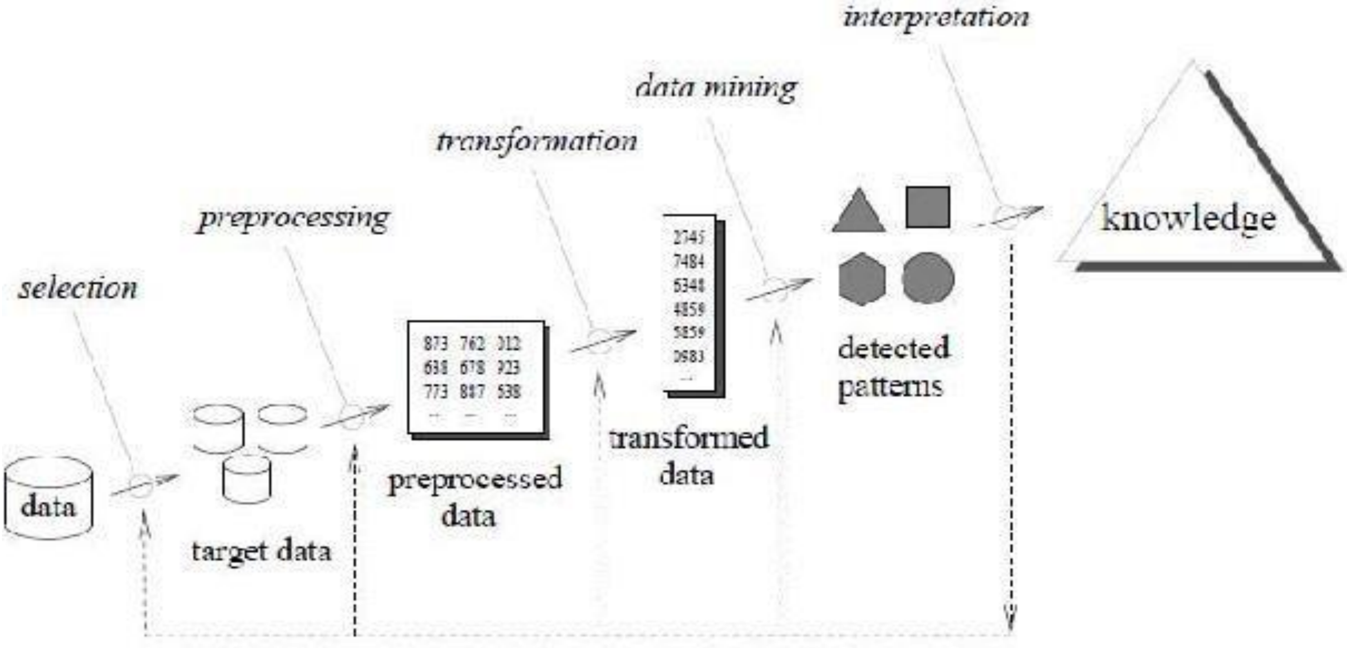


Figure 3.7

It's important to use new data when evaluating our model to prevent the likelihood of overfitting to the training set. However, sometimes it's useful to evaluate our model as we're building it to find that best parameters of a model - but we can't use the test set for this evaluation or else we'll end up selecting the parameters that perform best on the test data but maybe not the parameters that generalize best. To evaluate the model while still building and tuning the model, we create a third subset of the data known as the validation set. A typical train/test/validation split would be to use 60% of the data for training, 20% of the data for validation, and 20% of the data for testing

The Holdout method is used to evaluate the model performance and uses two types of data for testing and training. The test data is used to calculate the performance of the model whereas it is trained using the training data set. This method is used to check how well the machine learning model developed using different algorithm techniques performs on unseen samples of data. This approach is simple, flexible and fast.

Cross-validation is a procedure of dividing the whole dataset into data samples, and then evaluating the machine learning model using the other samples of data to known

Chapter 4

DEPLOYMENT AND RESULTS

4.1 Introduction

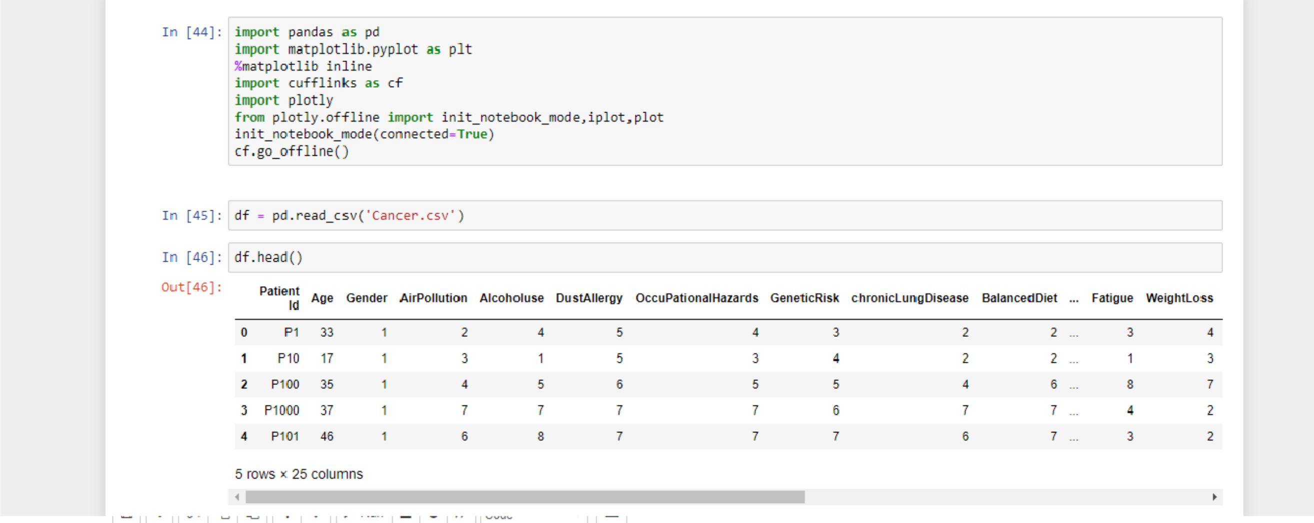
Cancer is a genetic disease—that is, it is caused by changes to genes that control the way our cells function, especially how they grow and divide.

Genetic changes that cause cancer can happen because:

- Errors that occur as cells divide.
- Damage to DNA caused by harmful substances in the environment, such as the chemicals in tobacco smoke and ultraviolet rays from the sun. (Our Cancer Causes and Prevention section has more information.)
- They were inherited from our parents.

The body normally eliminates cells with damaged DNA before they turn cancerous. But the body's ability to do so goes down as we age. This is part of the reason why there is a higher risk of cancer later in life.

4.2 Source code



In [44]:
```
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()
```

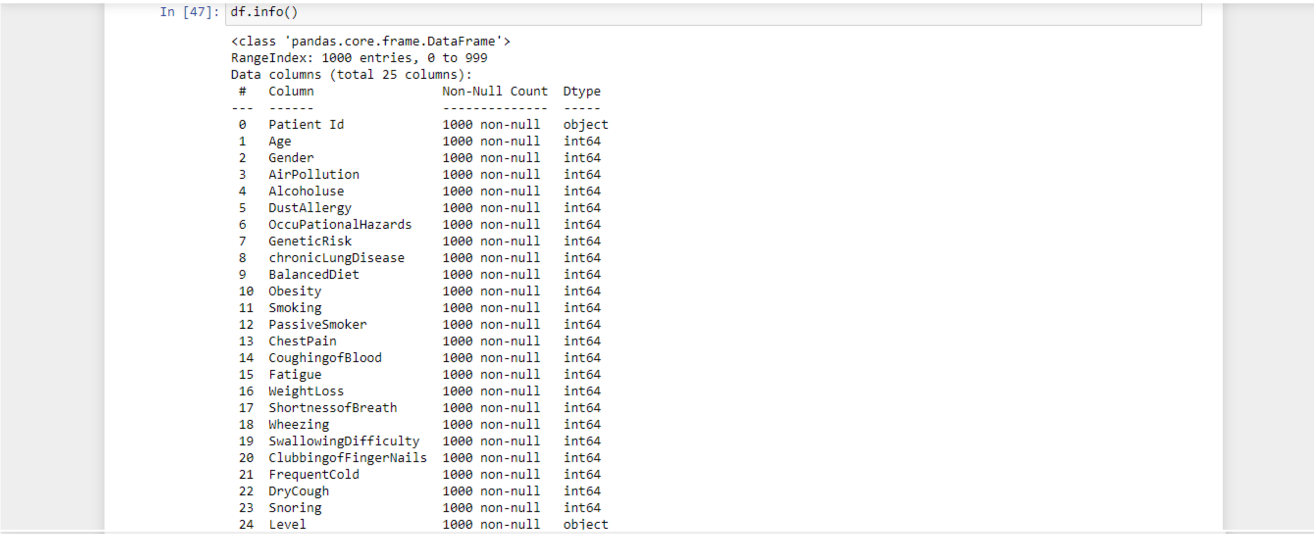
In [45]:
```
df = pd.read_csv('Cancer.csv')
```

In [46]:
```
df.head()
```

Out[46]:

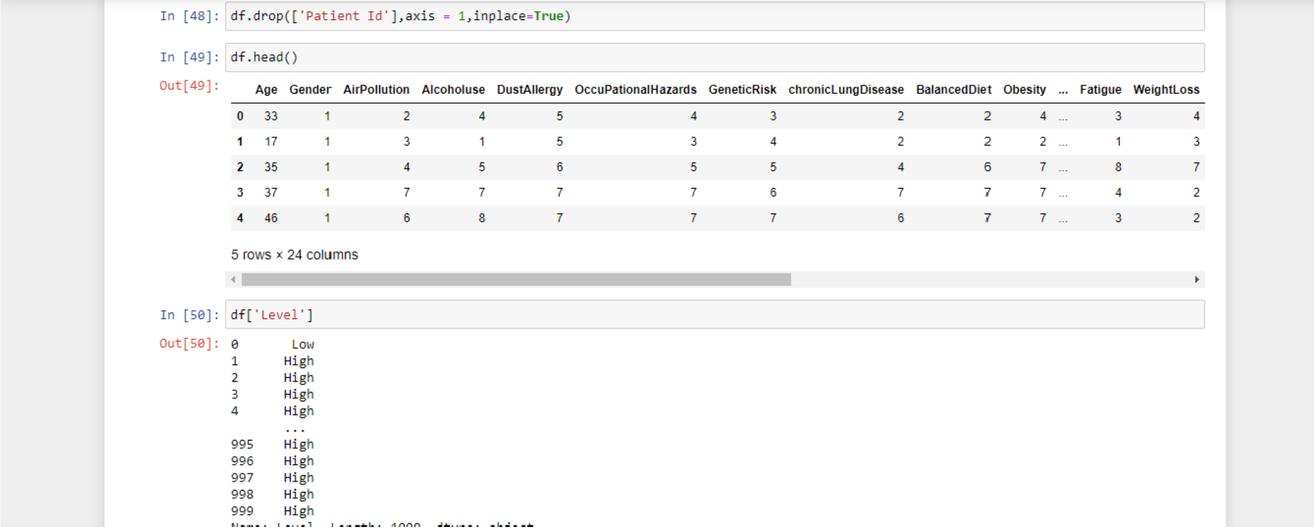
	Patient Id	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	...	Fatigue	WeightLoss
0	P1	33	1	2	4	5	4	3	2	2	...	3	4
1	P10	17	1	3	1	5	3	4	2	2	...	1	3
2	P100	35	1	4	5	6	5	5	4	6	...	8	7
3	P1000	37	1	7	7	7	7	6	7	7	...	4	2
4	P101	46	1	6	8	7	7	7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

5 rows × 25 columns



In [47]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Patient Id            1000 non-null   object
 1   Age                   1000 non-null   int64 
 2   Gender                1000 non-null   int64 
 3   AirPollution          1000 non-null   int64 
 4   Alcoholuse            1000 non-null   int64 
 5   DustAllergy           1000 non-null   int64 
 6   OccuPationalHazards   1000 non-null   int64 
 7   GeneticRisk           1000 non-null   int64 
 8   chronicLungDisease    1000 non-null   int64 
 9   BalancedDiet          1000 non-null   int64 
 10  Obesity               1000 non-null   int64 
 11  Smoking               1000 non-null   int64 
 12  PassiveSmoker         1000 non-null   int64 
 13  ChestPain             1000 non-null   int64 
 14  CoughingofBlood       1000 non-null   int64 
 15  Fatigue               1000 non-null   int64 
 16  WeightLoss            1000 non-null   int64 
 17  ShortnessofBreath     1000 non-null   int64 
 18  Wheezing              1000 non-null   int64 
 19  SwallowingDifficulty  1000 non-null   int64 
 20  ClubbingofFingerNails 1000 non-null   int64 
 21  FrequentCold          1000 non-null   int64 
 22  DryCough              1000 non-null   int64 
 23  Snoring               1000 non-null   int64 
 24  Level                 1000 non-null   object
```



In [48]:
```
df.drop(['Patient Id'],axis = 1,inplace=True)
```

In [49]:
```
df.head()
```

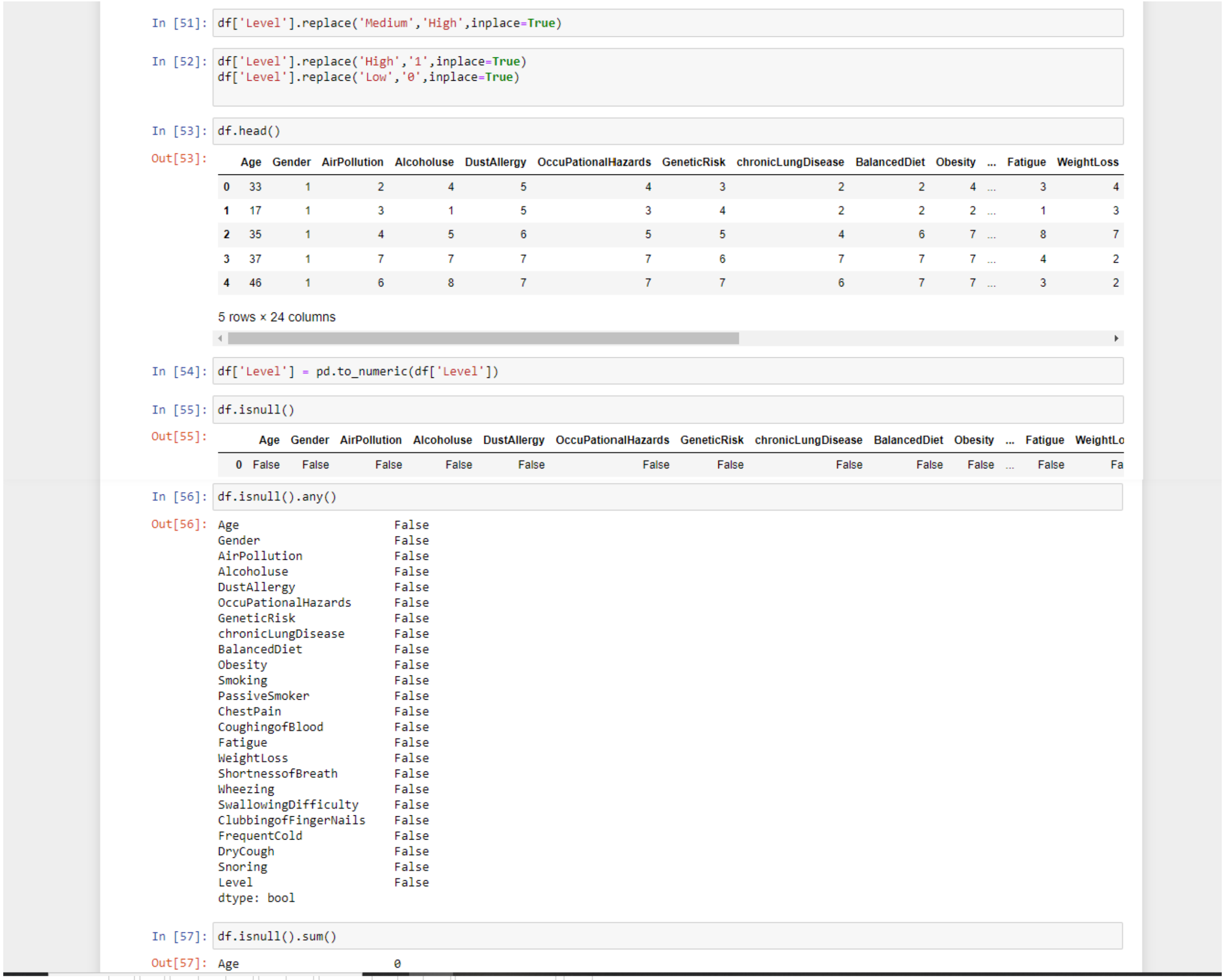
Out[49]:

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

5 rows × 24 columns

In [50]:
```
df['Level']
```

Out[50]:
```
0        Low
1       High
2       High
3       High
4       High
        ... 
995     High
996     High
997     High
998     High
999     High
Name: Level, Length: 1000, dtype: object
```



```
In [51]: df['Level'].replace('Medium','High',inplace=True)
```

```
In [52]: df['Level'].replace('High','1',inplace=True)
         df['Level'].replace('Low','0',inplace=True)
```

```
In [53]: df.head()
```

Out[53]:

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

5 rows × 24 columns

```
In [54]: df['Level'] = pd.to_numeric(df['Level'])
```

```
In [55]: df.isnull()
```

Out[55]:

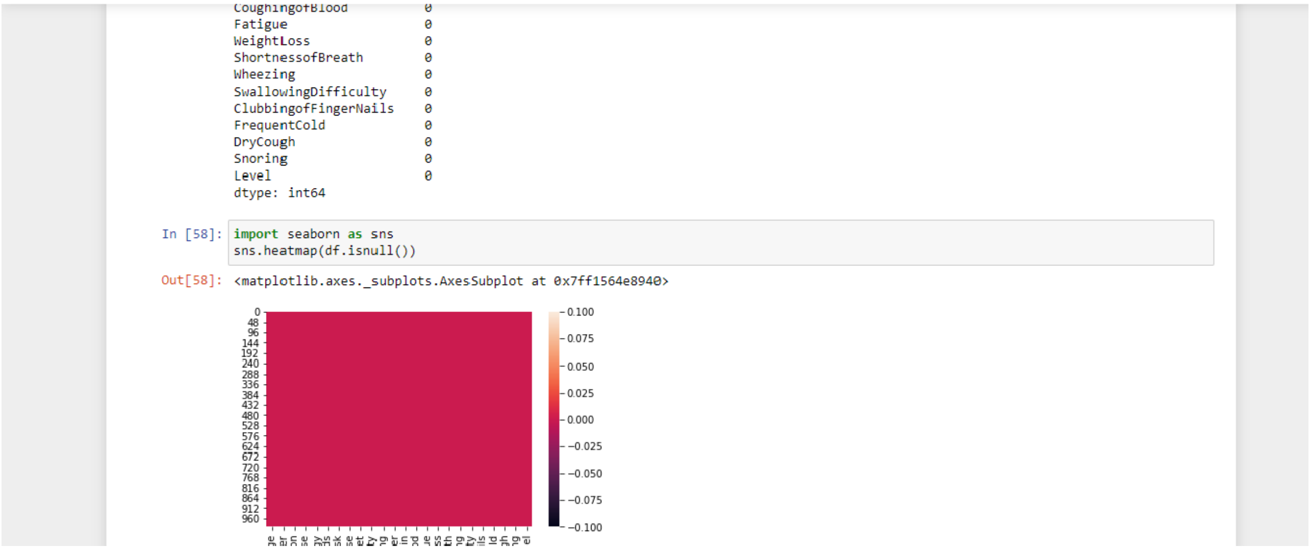
	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	Obesity	...	Fatigue	WeightLo
0	False	False	False	False	False	False	False	False	False	False	...	False	Fa

```
In [56]: df.isnull().any()
```

```
Out[56]: 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
```

```
In [57]: df.isnull().sum()
```

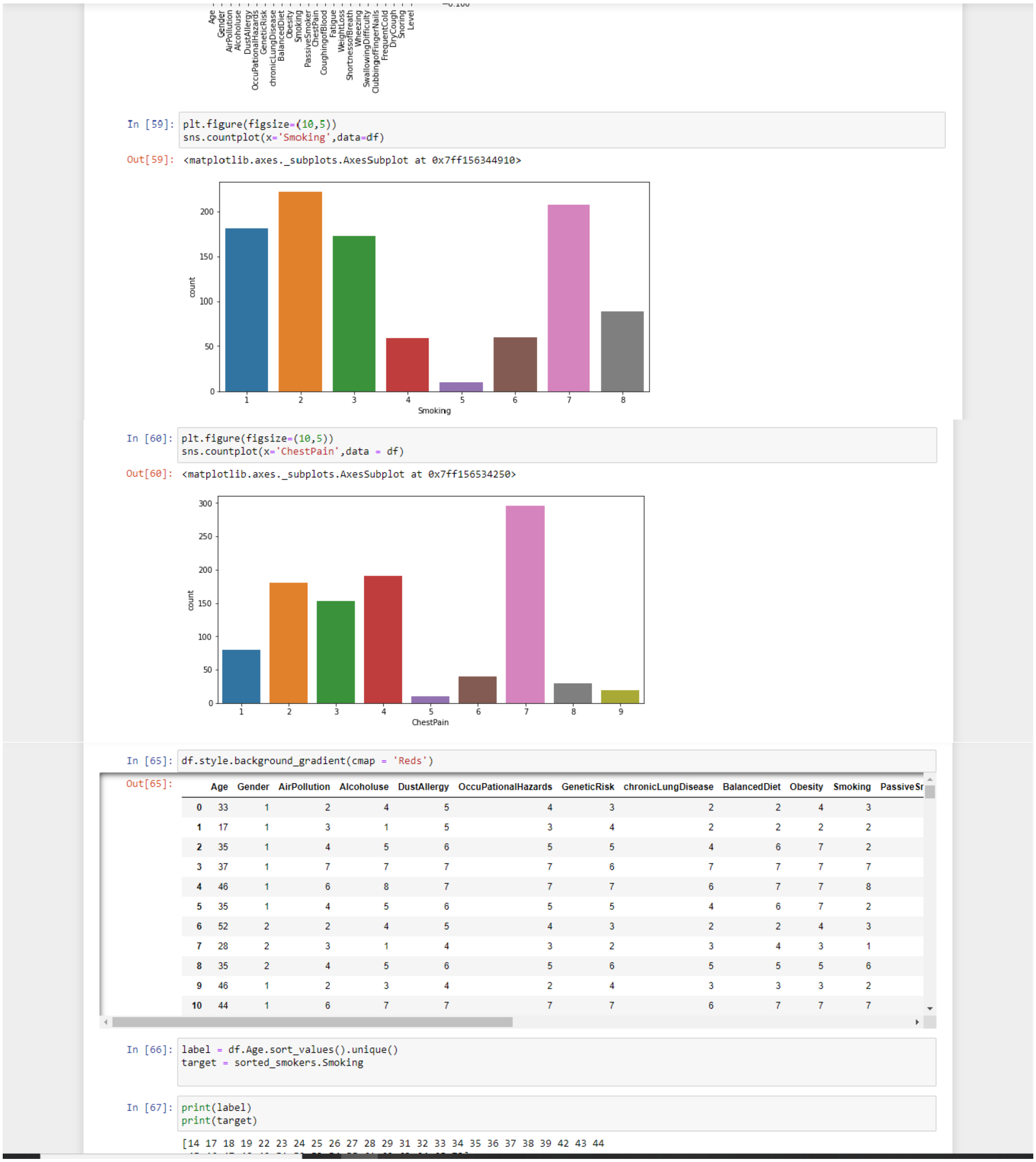
```
Out[57]: Age                         0
```



```
         CoughingofBlood             0
         Fatigue                     0
         WeightLoss                  0
         ShortnessofBreath           0
         Wheezing                    0
         SwallowingDifficulty        0
         ClubbingofFingerNails       0
         FrequentCold                0
         DryCough                    0
         Snoring                     0
         Level                       0
         dtype: int64
```

```
In [58]: import seaborn as sns
         sns.heatmap(df.isnull())
```

```
Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff1564e8940>
```



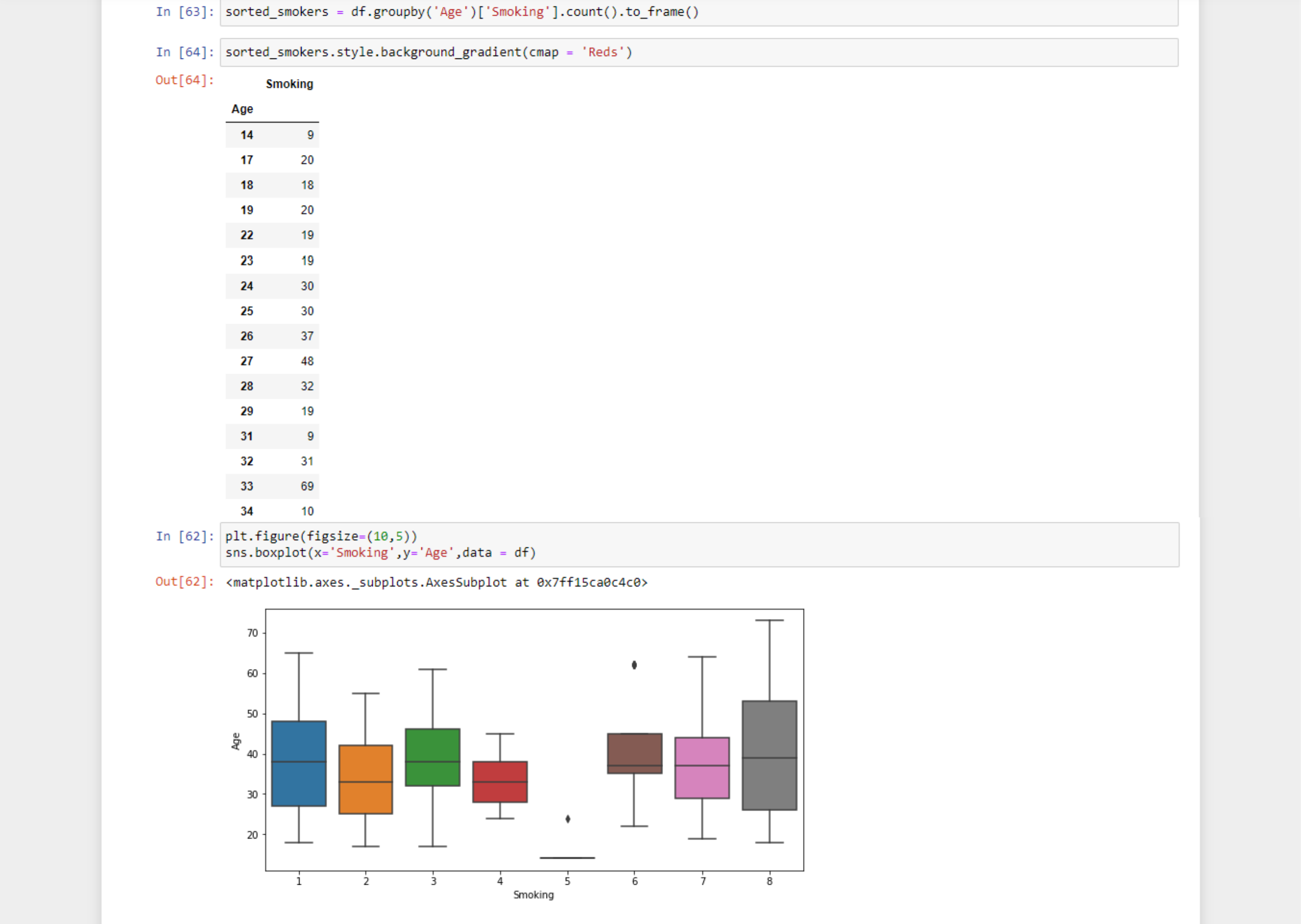
```
In [59]: plt.figure(figsize=(10,5))
         sns.countplot(x='Smoking',data=df)

Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff156344910>
```

```
In [60]: plt.figure(figsize=(10,5))
         sns.countplot(x='ChestPain',data = df)

Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff156534250>
```

```
In [65]: df.style.background_gradient(cmap = 'Reds')

Out[65]:
```

	Age	Gender	AirPollution	Alcoholuse	DustAllergy	OccuPationalHazards	GeneticRisk	chronicLungDisease	BalancedDiet	Obesity	Smoking	PassiveSr
0	33	1	2	4	5	4	3	2	2	4	3	
1	17	1	3	1	5	3	4	2	2	2	2	
2	35	1	4	5	6	5	5	4	6	7	2	
3	37	1	7	7	7	7	6	7	7	7	7	
4	46	1	6	8	7	7	7	6	7	7	8	
5	35	1	4	5	6	5	5	4	6	7	2	
6	52	2	2	4	5	4	3	2	2	4	3	
7	28	2	3	1	4	3	2	3	4	3	1	
8	35	2	4	5	6	5	6	5	5	5	6	
9	46	1	2	3	4	2	4	3	3	3	2	
10	44	1	6	7	7	7	7	6	7	7	7	

```
In [66]: label = df.Age.sort_values().unique()
         target = sorted_smokers.Smoking

In [67]: 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 43 44
```



```
In [63]: sorted_smokers = df.groupby('Age')['Smoking'].count().to_frame()
```

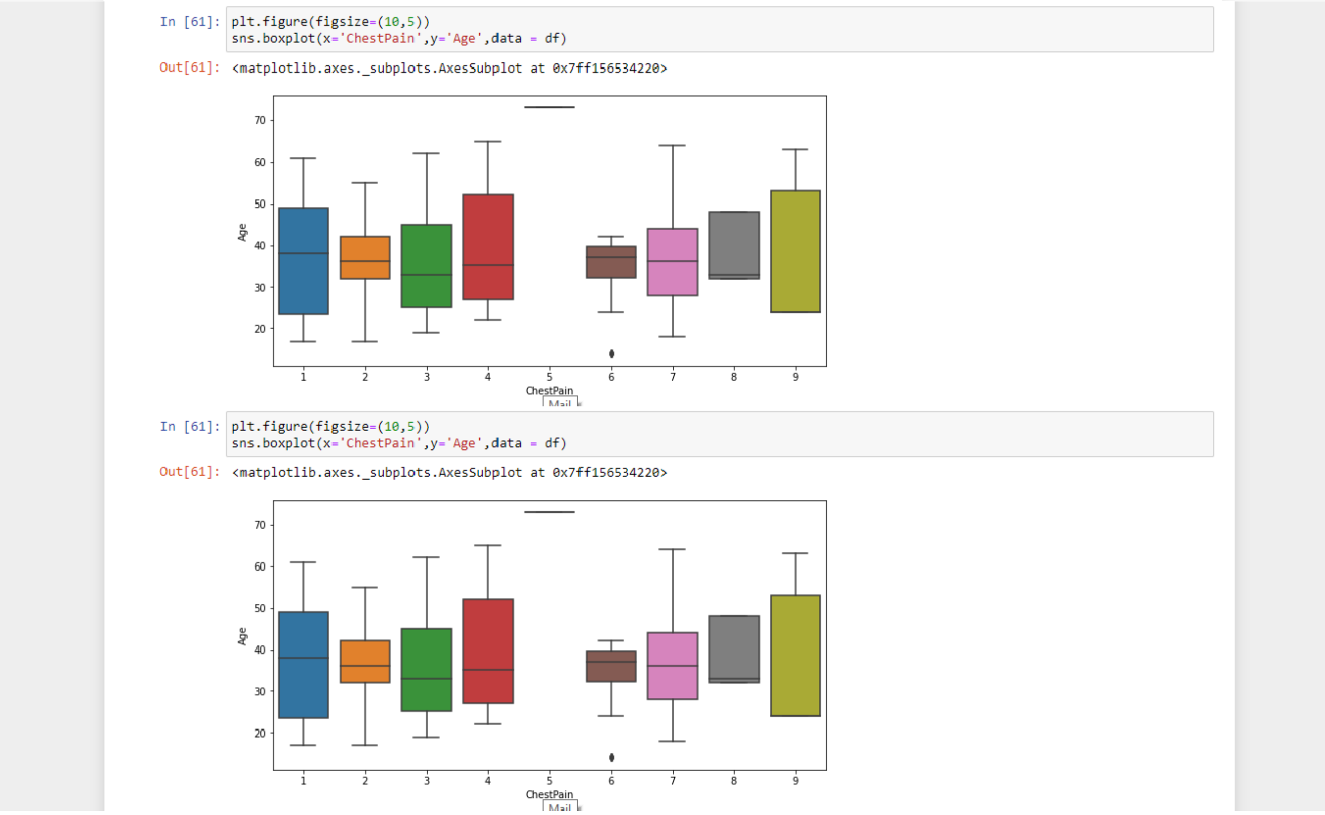
```
In [64]: sorted_smokers.style.background_gradient(cmap = 'Reds')
```

Out[64]:

Age	Smoking
14	9
17	20
18	18
19	20
22	19
23	19
24	30
25	30
26	37
27	48
28	32
29	19
31	9
32	31
33	69
34	10

```
In [62]: plt.figure(figsize=(10,5))
         sns.boxplot(x='Smoking',y='Age',data = df)
```

```
Out[62]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff15ca0c4c0>
```

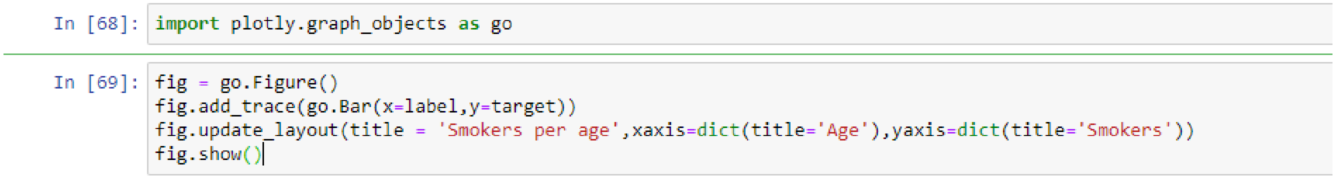


```
In [61]: plt.figure(figsize=(10,5))
         sns.boxplot(x='ChestPain',y='Age',data = df)
```

```
Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff156534220>
```

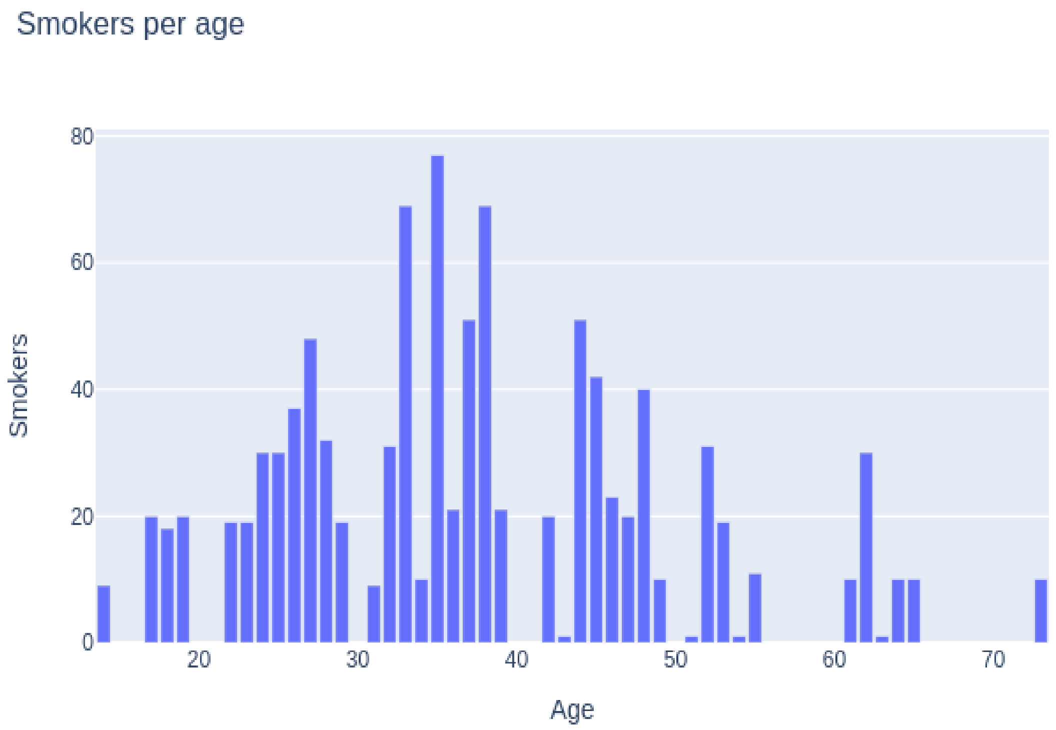
```
In [61]: plt.figure(figsize=(10,5))
         sns.boxplot(x='ChestPain',y='Age',data = df)
```

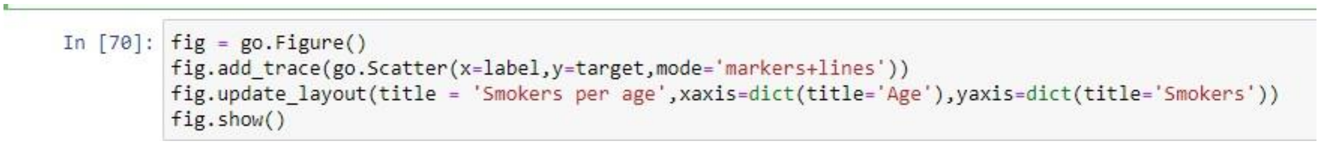
```
Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff156534220>
```



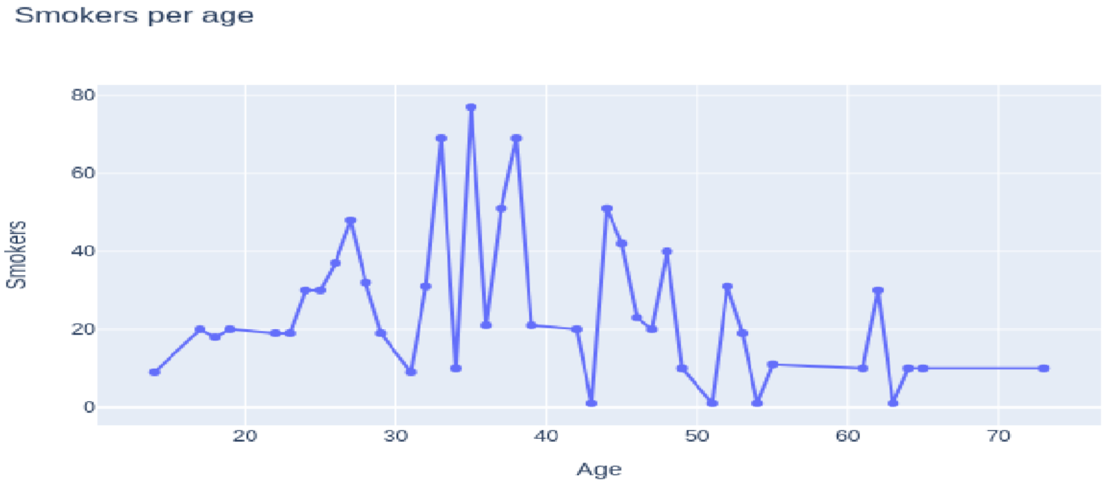
```
In [68]: import plotly.graph_objects as go
```

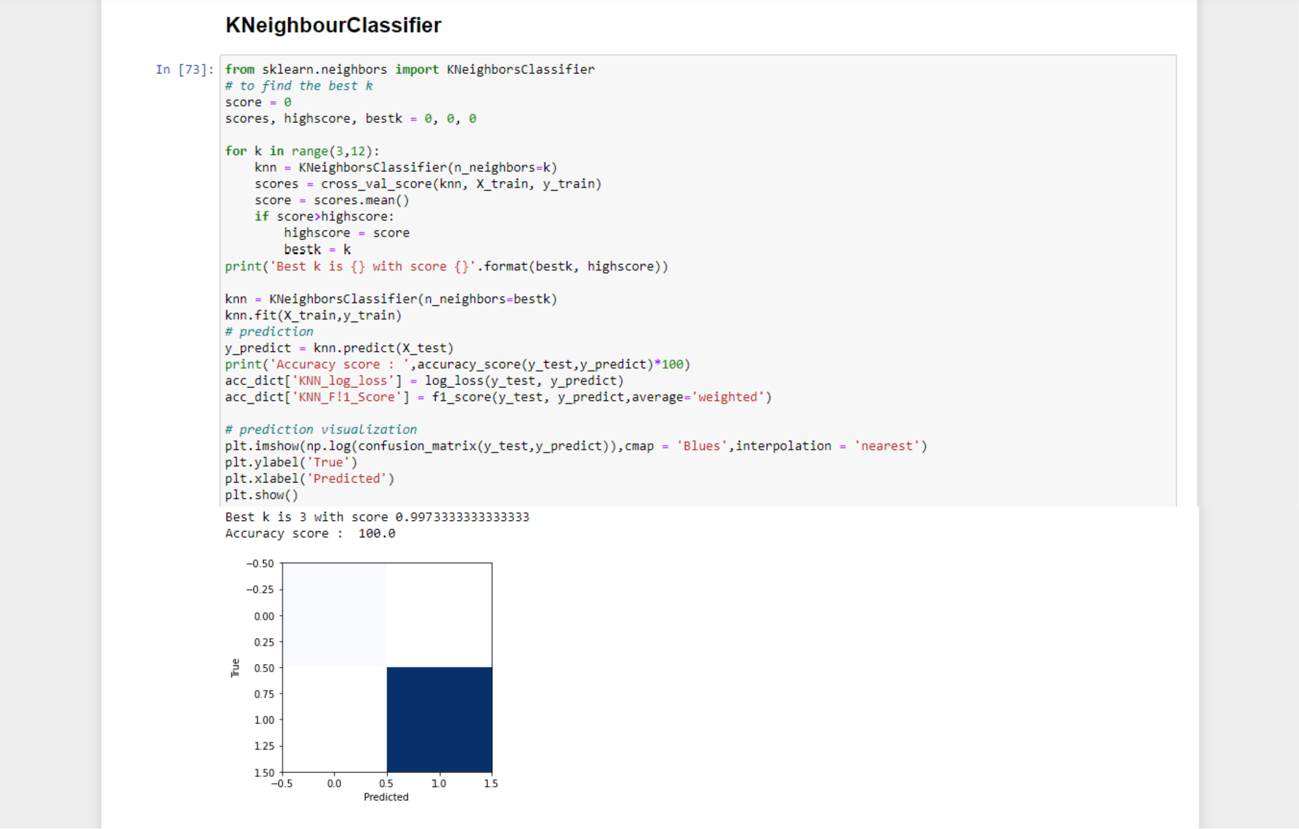
```
In [69]: fig = go.Figure()
         fig.add_trace(go.Bar(x=label,y=target))
         fig.update_layout(title = 'Smokers per age',xaxis=dict(title='Age'),yaxis=dict(title='Smokers'))
         fig.show()
```





```
In [70]: 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=dict(title='Smokers'))
         fig.show()
```



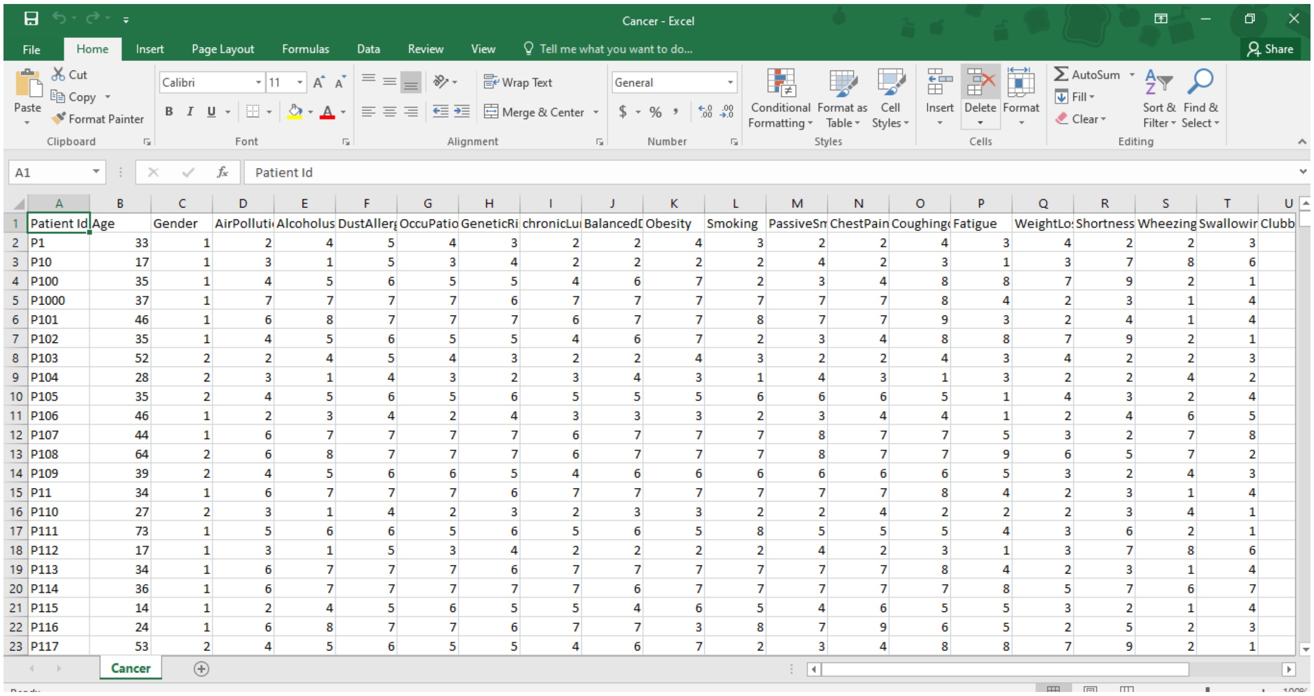


KNeighbourClassifier

```
In [73]: 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',interpolation = 'nearest')
         plt.ylabel('True')
         plt.xlabel('Predicted')
         plt.show()
```

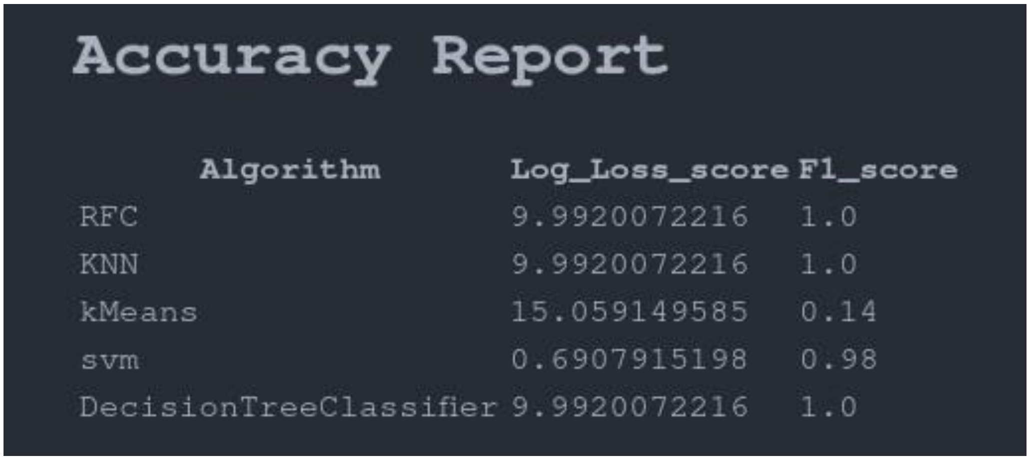
```
Best k is 3 with score 0.9973333333333333
Accuracy score :  100.0
```

Dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Patient Id	Age	Gender	AirPolluti	Alcoholus	DustAllerg	OccuPatio	GeneticRi	chronicLu	BalancedD	Obesity	Smoking	PassiveSm	ChestPain	Coughing	Fatigue	WeightLo:	Shortness	Wheezing	Swallowir	Clubb
2	P1	33	1	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	
3	P10	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	
4	P100	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	
5	P1000	37	1	7	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	
6	P101	46	1	6	8	7	7	7	6	7	7	8	7	7	9	3	2	4	1	4	
7	P102	35	1	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	
8	P103	52	2	2	4	5	4	3	2	2	4	3	2	2	4	3	4	2	2	3	
9	P104	28	2	3	1	4	3	2	3	4	3	1	4	3	1	3	2	2	4	2	
10	P105	35	2	4	5	6	5	6	5	5	5	6	6	6	5	1	4	3	2	4	
11	P106	46	1	2	3	4	2	4	3	3	3	2	3	4	4	1	2	4	6	5	
12	P107	44	1	6	7	7	7	7	6	7	7	7	8	7	7	5	3	2	7	8	
13	P108	64	2	6	8	7	7	7	6	7	7	7	8	7	7	9	6	5	7	2	
14	P109	39	2	4	5	6	6	5	4	6	6	6	6	6	6	5	3	2	4	3	
15	P11	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	
16	P110	27	2	3	1	4	2	3	2	3	3	2	2	4	2	2	2	3	4	1	
17	P111	73	1	5	6	6	5	6	5	6	5	8	5	5	5	4	3	6	2	1	
18	P112	17	1	3	1	5	3	4	2	2	2	2	4	2	3	1	3	7	8	6	
19	P113	34	1	6	7	7	7	6	7	7	7	7	7	7	8	4	2	3	1	4	
20	P114	36	1	6	7	7	7	7	7	6	7	7	7	7	7	8	5	7	6	7	
21	P115	14	1	2	4	5	6	5	5	4	6	5	4	6	5	5	3	2	1	4	
22	P116	24	1	6	8	7	7	6	7	7	3	8	7	9	6	5	2	5	2	3	
23	P117	53	2	4	5	6	5	5	4	6	7	2	3	4	8	8	7	9	2	1	

4.3 Final Result



Accuracy Report

Algorithm	Log_Loss_score	F1_score
RFC	9.9920072216	1.0
KNN	9.9920072216	1.0
kMeans	15.059149585	0.14
svm	0.6907915198	0.98
DecisionTreeClassifier	9.9920072216	1.0

Chapter 5

CONCLUSION

5.1 Project conclusion

In this Project we have attempted to explain, compare and assess the performance of different machine learning that are being applied to cancer prediction . Specifically we identified a number of trends with respect to the types of machine learning methods being used, the types of training data being integrated, the kinds of endpoint predictions being made, the types of cancers being studied and the overall performance of these methods in predicting cancer susceptibility or outcomes

5.2 Future Scope

AI is set to change the medical industry in the coming decades — it wouldn’t make sense for pathology to not be disrupted too.

Currently, ML models are still in the testing and experimentation phase for cancer prognoses. As datasets are getting larger and of higher quality, researchers are building increasingly accurate models.