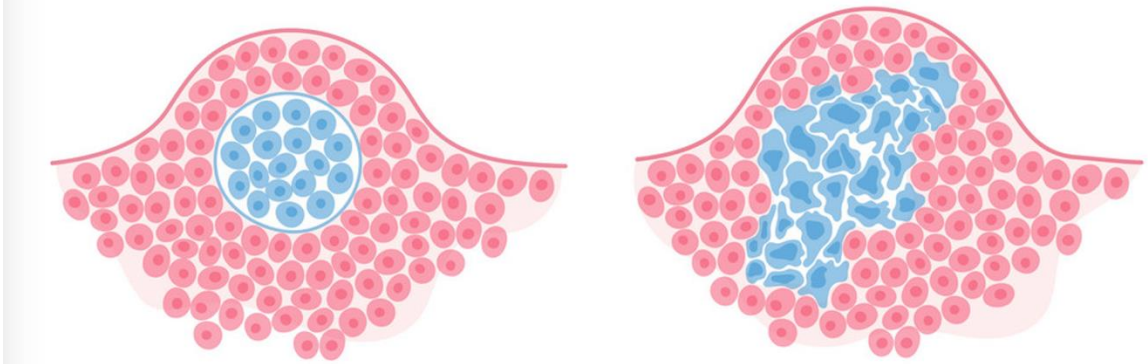
King Saud University
College of Computer and Information Sciences
Software Engineering department

SWE 485: Selected topics in software engineeringCourse

Project

Cancer Data Analysis and Classification

Phase #1



Project Report

Group 5:

Name	ID	Section
Razan Barakat	441203808	60120
Reema Almutairi	441200838	60120
Eman Fattah	441203742	60120

Table of content:

Table of figures:

1. Introduction:

Cancer data analysis and classification is a crucial area of re*search that involves the analysis and interpretation of large datasets related to cancer. This analysis aims to identify whether a cancer is benign or malignant according to its individual characteristics, One key aspect of cancer data analysis is classification, which involves grouping cancer patients based on various factors such as their Cancer id, Cancer Types diagnosis wither its malignant cancer (m) or benign cancer (B) and other Visual Characteristics of cancer and correlations within the data that can provide insights into the underlying types of cancer, as well as prevention strategies. Machine learning algorithms and other advanced analytical techniques are commonly used in cancer data analysis and classification. These methods can help researchers uncover new insights into the complex mechanisms underlying cancer development and progression, paving the way for new treatments and improved patient outcomes. The result of the diagnose (m) is malignant cancer and (B) benign cancer. We expect a variety of results due to the amount of data set being used.*

2. Motivation:

The motivation behind cancer data analysis and classification is to improve the understanding of complex diseases and ultimately improve patient outcomes. Researchers can develop potential biomarkers for diagnosis and treatment by analyzing large amounts of cancer data. By classifying cancer patients based on their molecular and clinical characteristics, one can predict prognosis, select appropriate treatment options, and monitor the progression of the disease. Ultimately, the goal is to develop personalized therapies tailored to individual patients, leading to increased survival rates and quality of life.

3. Machine learning Tasks

The objective of our study is to identify the relationship between traits and the impact of each trait on cancer. By detecting the disease early, patients can become more aware of their situation and researchers can develop more effective treatments in the future.
In order to remove and detect extreme values, we will use clustering to divide the data into a set of similar features.
It is an unsupervised machine learning technique that identifies and groups similar data points in a larger dataset without regard to the specific outcome.
Our goal is to create a model that can predict whether someone has cancer or not based on the attributes of the data object. If so, which type of cancer it is, using the category attribute set.
In order to classify data correctly, it is imperative to determine accurately what the target class is for each case. We must also consider the accuracy of the model, as well as the reliability of the data and its sources. Finally, we need to select the best machine learning algorithm that will be used to build the model.

4. Data:

It is the cancer data that has been chosen to be the data set that we will be using because the goal of our project is to determine whether there are benign or malignant cancer cells in the data we have chosen.
Our cancer data contains 2 types of cancers: 1. benign cancer (B) and 2. malignant cancer (M). The dataset is called cancer - Cancer Data Dataset which we got from kaggle.com by downloading the csv file.

URL: https://www.kaggle.com/datasets/eremtaha/cancer-data?resource=download&select=Cancer_Data.csv

It consists of exactly 569 observations(records) that each consisting of 32 variables (Id, diagnosis, radius mean, texture mean, perimeter mean, area mean, smoothness mean, compactness mean, concavity mean, concave points mean, symmetry mean, fractal dimension mean, radius se, texture se, perimeter se, area se, smoothness se, compactness se, concavity se, concave points se, symmetry se, fractal dimension se, radius worst, texture worst, perimeter worst, area worst, smoothness worst, compactness worst, concavity worst, concave points worst, symmetry worst, fractal dimension worst).

Attribute name	Attribute description	Attribute type
Id	Cancer id	int64
diagnosis	Cancer type	Object
radius mean	Visual Characteristics of cancer	float64
texture mean	Visual Characteristics of cancer	float64
perimeter mean	Visual Characteristics of cancer	float64
area mean	Visual Characteristics of cancer	float64
smoothness mean	Visual Characteristics of cancer	float64
Compactness mean	Visual Characteristics of cancer	float64
Concavity mean	Visual Characteristics of cancer	float64
concave points mean	Visual Characteristics of cancer	float64
symmetry mean	Visual Characteristics of cancer	float64
fractal dimension mean	Visual Characteristics of cancer	float64
radius se	Visual Characteristics of cancer	float64
texture se	Visual Characteristics of cancer	float64

perimeter se	Visual Characteristics of cancer	float64
area se	Visual Characteristics of cancer	float64
smoothness se	Visual Characteristics of cancer	float64
compactness se	Visual Characteristics of cancer	float64
concavity se	Visual Characteristics of cancer	float64
concave points se	Visual Characteristics of cancer	float64
symmetry se	Visual Characteristics of cancer	float64
fractal dimension se	Visual Characteristics of cancer	float64
radius worst	Visual Characteristics of cancer	float64
texture worst	Visual Characteristics of cancer	float64
perimeter worst	Visual Characteristics of cancer	float64
area worst	Visual Characteristics of cancer	float64
smoothness worst	Visual Characteristics of cancer	float64
compactness worst	Visual Characteristics of cancer	float64
concavity worst	Visual Characteristics of cancer	float64
concave points worst	Visual Characteristics of cancer	float64
symmetry worst	Visual Characteristics of cancer	float64
fractal dimension worst	Visual Characteristics of cancer	float64
Unnamed: 32	Visual Characteristics of cancer	float64

Table1: data variables.

The data type of each variable is displayed using the .info() method.

```
cancer1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

Figure 1. Variables and their data types in the dataset

A sample from the dataset was displayed using the following. head () method. The function displayed the first 10 rows of the dataset.

```
[53]: #the first 10 rows of the dataset.
      cancer1.head(10)
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	conc points_w
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.15780	0.08089	...	23.75	103.40	741.6	0.1791	0.5249	0.5355	0.
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.11270	0.07400	...	27.66	153.20	1606.0	0.1442	0.2576	0.3784	0.1
7	84458202	M	13.71	20.83	90.20	577.9	0.11890	0.16450	0.09366	0.05985	...	28.14	110.60	897.0	0.1654	0.3682	0.2678	0.1
8	844981	M	13.00	21.82	87.50	519.8	0.12730	0.19320	0.18590	0.09353	...	30.73	106.20	739.3	0.1703	0.5401	0.5390	0.2
9	84501001	M	12.46	24.04	83.97	475.9	0.11860	0.23960	0.22730	0.08543	...	40.68	97.65	711.4	0.1853	1.0580	1.1050	0.1

10 rows × 33 columns

Figure 2. The 10 displayed rows of the datasets.

concave points_worst	symmetry_worst	fractal_dimension_worst	Unnamed: 32
0.2654	0.4601	0.11890	NaN
0.1860	0.2750	0.08902	NaN
0.2430	0.3613	0.08758	NaN
0.2575	0.6638	0.17300	NaN
0.1625	0.2364	0.07678	NaN
0.1741	0.3985	0.12440	NaN
0.1932	0.3063	0.08368	NaN
0.1556	0.3196	0.11510	NaN
0.2060	0.4378	0.10720	NaN
0.2210	0.4366	0.20750	NaN

Figure 3. The rest columns of the displayed rows in the dataset.

For ease of use, we mapped diagnosis attributes from objects to integers

```
[54]: cancer1['diagnosis'].replace(['B', 'M'],
                                   [0, 1], inplace=True)
```

Figure 4: mapping diagnosis into an integer.

5. Variables distribution:

- **Distribution plot**

This Distribution plot depicts the overall distribution and shows the impact of texture mean (Visual Characteristics of cancer) and texture se (Visual Characteristics of cancer) on cancer disease.

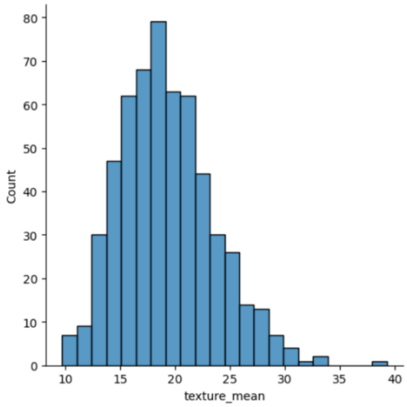


Figure 5: distribution plot1.

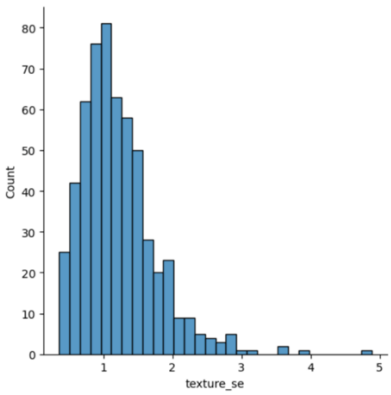


Figure 6: distribution plot2.

- **Scatter graphs**

The scatter graphs show the distribution of the three Visual Characteristics of cancer which are radius mean, radius se and radius worst according to diagnosis.

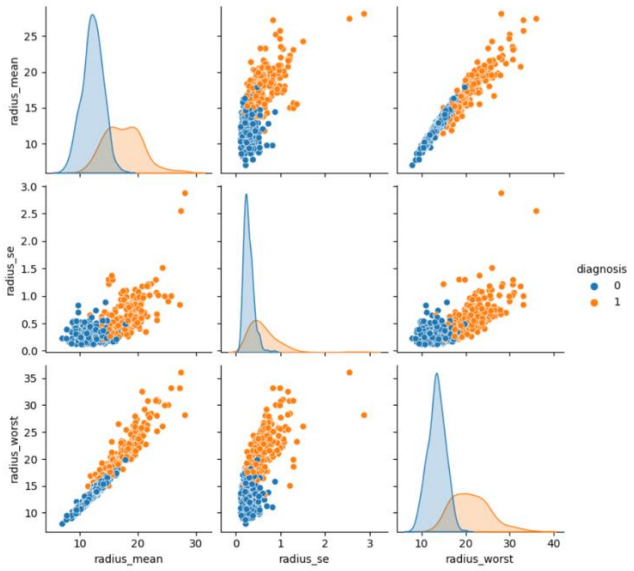


Figure 7: scatter graphs.

- **Bar plot**

  The bar plot shows that the majority of cancer diagnoses are benign with 357 total cases, while the minority is malignant with 212 cases.

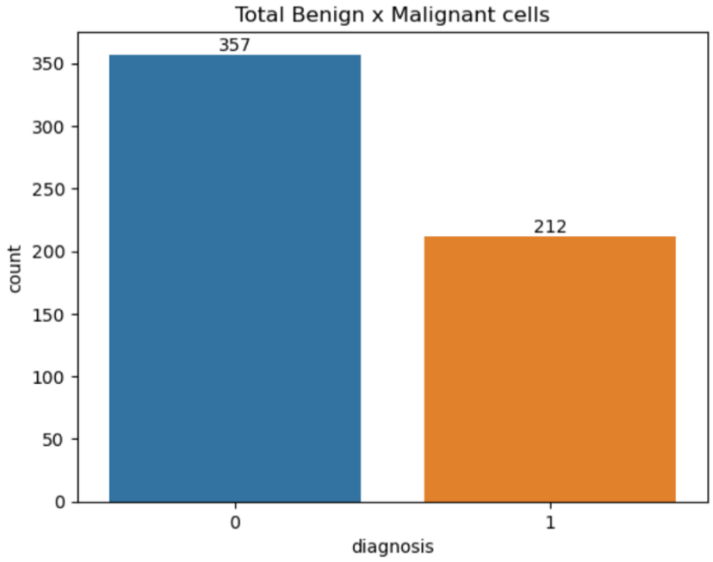


Figure 8: bar plot

- **Missing values(null):**

To Spot any missing values (null values) used, isnull() function which indicts that there are no null values:

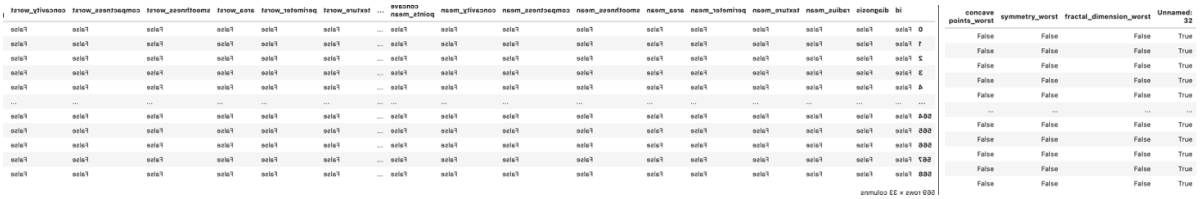
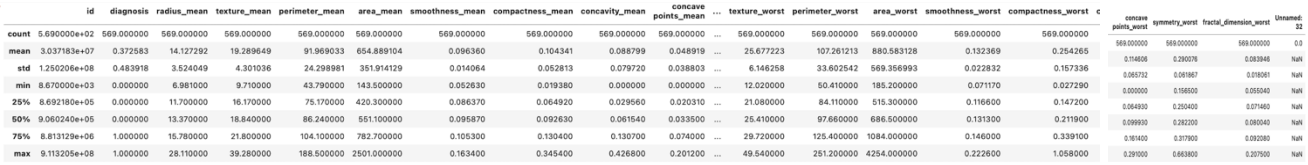


Figure 9: Missing values result.

- **Statistical summaries:**

.describe() method used to calculate Mean and variance and other statistical calculations.



	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	c
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	0.372583	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	...	25.677223	107.261213	880.583128	0.132369	0.254265	
std	1.250206e+08	0.483918	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	...	6.146258	33.602542	569.356993	0.022832	0.157336	
min	8.670000e+03	0.000000	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	...	12.020000	50.410000	185.200000	0.071170	0.027290	
25%	8.692180e+05	0.000000	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	...	21.080000	84.110000	515.300000	0.116600	0.147200	
50%	9.060240e+05	0.000000	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	...	25.410000	97.660000	686.500000	0.131300	0.211900	
75%	8.813129e+06	1.000000	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	...	29.720000	125.400000	1084.000000	0.146000	0.339100	
max	9.113205e+08	1.000000	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	...	49.540000	251.200000	4254.000000	0.222600	1.058000	

concave points_worst	symmetry_worst	fractal_dimension_worst	Unnamed: 32
569.000000	569.000000	569.000000	0.0
0.114606	0.290076	0.083946	NaN
0.065732	0.061867	0.018061	NaN
0.000000	0.156500	0.055040	NaN
0.064930	0.250400	0.071460	NaN
0.099930	0.282200	0.080040	NaN
0.161400	0.317900	0.092080	NaN
0.291000	0.663800	0.207500	NaN

Figure 10: statistical calculations.

6. Data preprocessing

6.1 Data Transformation

- **Normalization**

```
In [9]: data['area_mean'].max

Out[9]: <bound method Series.max of 0      1001.0
        1      1326.0
        2      1203.0
        3       386.1
        4      1297.0
               ...
        564    1479.0
        565    1261.0
        566     858.1
        567    1265.0
        568     181.0
        Name: area_mean, Length: 569, dtype: float64>
```

Figure 11: the max of area-mean.

After exploring the method MAX shown in the figure we noticed that the area_mean attribute

has a large scale of data from 181.0 to 1479.0 Hence, we normalize it to easily compare the results by exclusively controlling its range

```
# Separate the aimed variable from the Variables
aim1 = cancer1 ['area_mean']
var1 = cancer1.drop('area_mean', axis=1)
# Normalize the Variables using the z-score method
var1 = (var1 - var1.mean()) / var1.std()
# Combine the normalized features with the target variable
normalized_cancer1 = pdd.concat ([var1, aim1], axis=1)
# Print the normalized data
print (normalized_cancer1.head())
```

```
         id  diagnosis  radius_mean  texture_mean  perimeter_mean  \
0 -0.236197   1.296535     1.096100     -2.071512        1.268817
1 -0.236196   1.296535     1.828212     -0.353322        1.684473
2  0.431362   1.296535     1.578499      0.455786        1.565126
3  0.431741   1.296535    -0.768233      0.253509       -0.592166
4  0.431821   1.296535     1.748758     -1.150804        1.775011

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0         1.567087          3.280628        2.650542             2.530249
1        -0.826235         -0.486643       -0.023825             0.547662
2         0.941382          1.052000        1.362280             2.035440
3         3.280667          3.399917        1.914213             1.450431
4         0.280125          0.538866        1.369806             1.427237

   symmetry_mean  ...  perimeter_worst  area_worst  smoothness_worst  \
0       2.215566  ...         2.301575    1.999478          1.306537
1       0.001391  ...         1.533776    1.888827         -0.375282
2       0.938859  ...         1.346291    1.455004          0.526944
3       2.864862  ...        -0.249720   -0.549538          3.391291
4      -0.009552  ...         1.337363    1.219651          0.220362

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0           2.614365         2.107672              2.294058        2.748204
1          -0.430066        -0.146620              1.086129       -0.243675
2           1.081980         0.854222              1.953282        1.151242
3           3.889975         1.987839              2.173873        6.040726
4          -0.313119         0.612640              0.728618       -0.867590

   fractal_dimension_worst  Unnamed: 32  area_mean
0                 1.935312          NaN     1001.0
1                 0.280943          NaN     1326.0
2                 0.201214          NaN     1203.0
3                 4.930672          NaN      386.1
4                -0.396751          NaN     1297.0

[5 rows x 33 columns]
```

Figure 12: the normalization of area_mean.

- **Discretization**

To increase the model performance, and reduce memory usage, we can simplify the radius_mean by splitting them into three intervals are:

1- Low, which holds any values from 0 to 9,
2-Middle, which holds any values from 10 to 19,
3- high, which holds any values from 20 to 29.

The code in figure takes radius_mean column, the minimum, and maximin values, and replaced its values with the corresponding labels.

```
In [148]: data['radius_mean'] = pd.cut(x = data['radius_mean'], bins=[0,10,20,30], right= False,
                                       labels=[1, 2, 3])

In [149]: data.head()
```

Out[149]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	tex
0	842302	M	2	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	3	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	2	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	2	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	3	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	

5 rows × 33 columns

Figure 12: results of discretization.

6.2 Data Cleaning

- **Remove the Outlier**

The table shows the boxplot of the attributes below.

```
In [40]: diagnosis = ['diagnosis']
radius_mean = ['radius_mean']
texture_mean = ['texture_mean']
perimeter_mean = ['perimeter_mean']
area_mean = ['area_mean']
texture_worst = ['texture_worst']
perimeter_worst = ['perimeter_worst']
area_worst = ['area_worst']
smoothness_worst = ['smoothness_worst']
compactness_worst = ['compactness_worst']
concavity_worst = ['concavity_worst']
concavePoints_worst = ['concave points_worst']
symmetry_worst=['symmetry_worst']
fractal_dimension_worst= ['fractal_dimension_worst']
```

Figure 13: removing outlier.

The box plot visually shows the distribution of numerical data and skewness through their quartiles, the points that go beyond the whiskers are treated as outliers by the box plot layout.

boxplot	Overview
	This boxplot shows that most cancer patients have the radius_mean between12 - 16

```
In [46]: data.boxplot(radius_mean)

Out[46]: <AxesSubplot:>
```

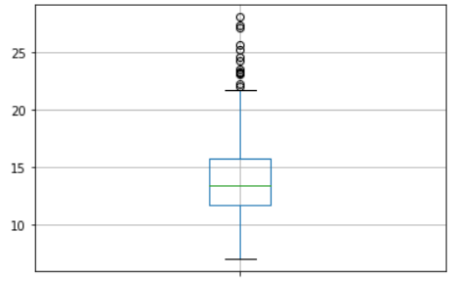


Figure 14: radius mean boxplot.

```
In [47]: data.boxplot(texture_mean)

Out[47]: <AxesSubplot:>
```

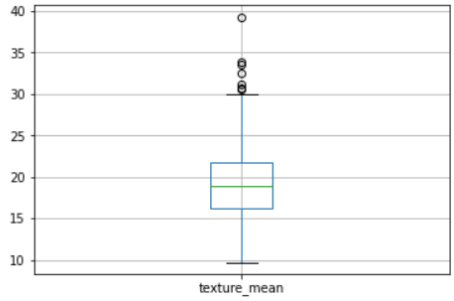


Figure 15: texture mean boxplot.

This boxplot shows that most cancer patients have the texture _mean between 16 - 22

```
In [52]: data.boxplot(perimeter_mean)

Out[52]: <AxesSubplot:>
```

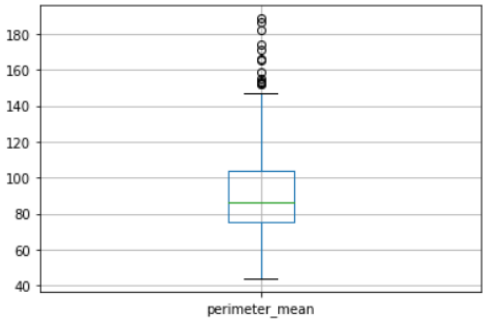


Figure 16: perimeter mean boxplot.

This boxplot shows that most cancer patients have the perimeter_mean between 75 - 105

```
In [53]: data.boxplot(area_mean)

Out[53]: <AxesSubplot:>
```

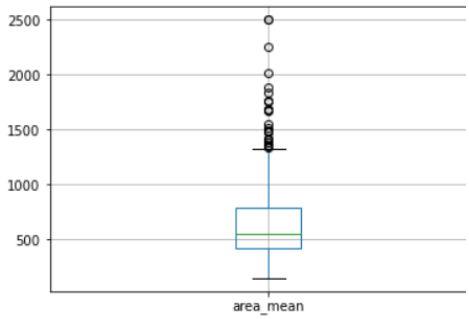


Figure 17: area mean boxplot.

This boxplot shows that most cancer patients have the area_mean between 400 - 775

```
In [57]: data.boxplot(texture_worst)

Out[57]: <AxesSubplot:>
```

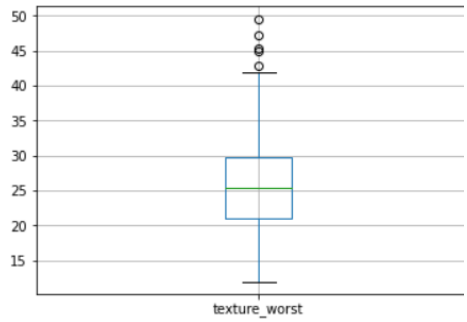


Figure 17:texture worst boxplot.

This boxplot shows that most cancer patients have the texture_worast between 21 - 30

```
In [59]: data.boxplot(perimeter_worst)

Out[59]: <AxesSubplot:>
```

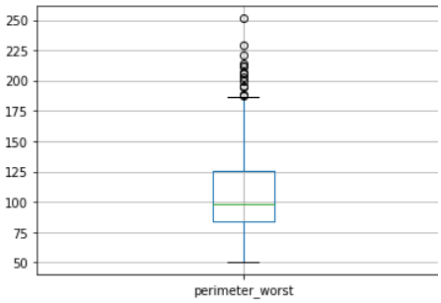


Figure 18:perimeter worst boxplot.

This boxplot shows that most cancer patients have the perimeter_worst between 85 - 125

```
[60]: data.boxplot(area_worst)

Out[60]: <AxesSubplot:>
```

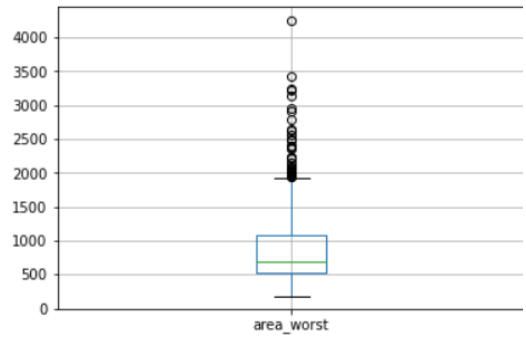


Figure 19: area worst boxplot.

This boxplot shows that most cancer patients have the area_worst between 500 - 1100

```
In [61]: data.boxplot(smoothness_worst)

Out[61]: <AxesSubplot:>
```

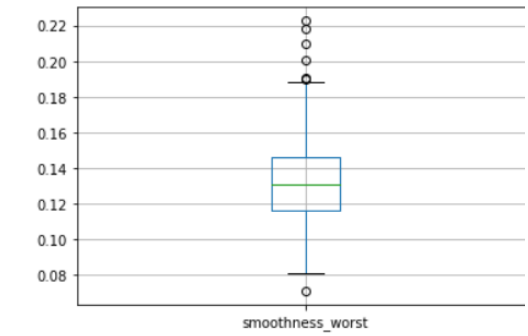


Figure 20: smoothness worst boxplot.

This boxplot shows that most cancer patients have the smoothness_worst between 0.169 - 0.1489

```
In [62]: data.boxplot(compactness_worst)

Out[62]: <AxesSubplot:>
```

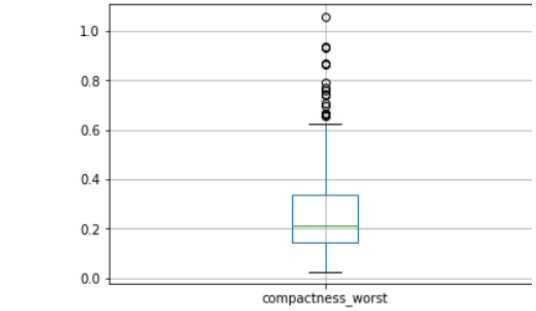


Figure 21: compactness worst boxplot.

This boxplot shows that most cancer patients have the compactness_worst between 0.15 - 0.35

```
In [63]: data.boxplot(concavity_worst)

Out[63]: <AxesSubplot:>
```

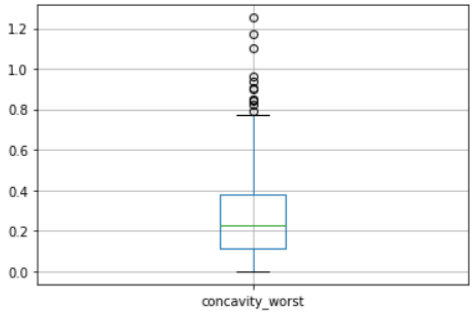


Figure 22: concavity worst boxplot.

This boxplot shows that most cancer patients have the concavity_worst between 0.12 - 0.38

```
In [64]: data.boxplot(concavePoints_worst)

Out[64]: <AxesSubplot:>
```

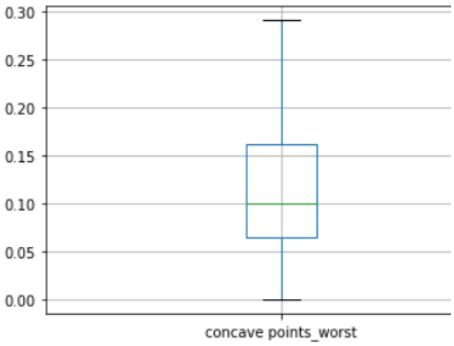


Figure 23: concave points worst boxplot.

This boxplot shows that most cancer patients have the concavePoints_worst between 0.67 - 0.167

```
In [65]: data.boxplot(symmetry_worst)

Out[65]: <AxesSubplot:>
```

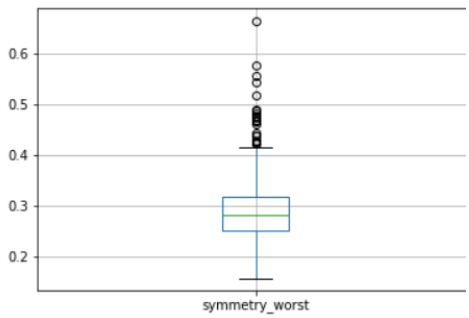


Figure 24: symmetry worst boxplot.

This boxplot shows that most cancer patients have the symmetry_worst between 0.25 - 0.35

```
In [66]: data.boxplot(fractal_dimension_worst)

Out[66]: <AxesSubplot:>
```

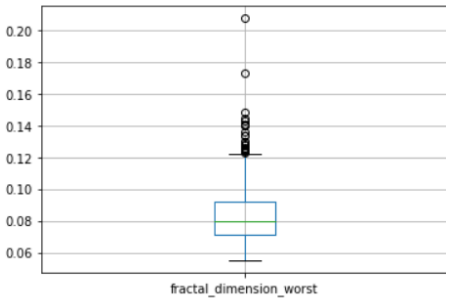


Figure 25: fractal dimension worst boxplot.

This boxplot shows that most cancer patients have the fractal_dimension_worst between 0.0725 - 0.0925

7. Recourses:

[1]Cancer Data. (2023, March 22). Kaggle. https://www.kaggle.com/datasets/erdemtaha/cancer-data

[2]AskPython. 2022. Detection and Removal of Outliers in Python - An Easy to Understand Guide - AskPython.[online] Available at: <https://www.askpython.com/python/examples/detectionremoval-outliers-
in-python>

[3] Introduction to Python
https://www.w3schools.com/python/python_intro.asp#:~:text=Python%20has%20a%20simple%20syntax,prototyping%20can%20be%20very%20quick.