breast-cancer-diagnosis

February 14, 2024

0.0.1 Breast Cancer Diagnosis Using Python

[43]:
```
# importing libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
```

[44]:
```
# reading data from the file
df=pd.read_csv("C:\\Users\\HP\\Downloads\\Breast_Cancer_prediction-main\\data.
 ↪csv")
```

[45]:
```
df.head()
```

[45]:
```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  …          17.33           184.60      2019.0            0.1622
1  …          23.41           158.80      1956.0            0.1238
```

```
2  …          25.53           152.50      1709.0            0.1444
3  …          26.50            98.87       567.7            0.2098
4  …          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

[46]:
```
df.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
```

[47]:
```
# return all the columns with null values count
df.isna().sum()
```

[47]:
```
id                          0
diagnosis                   0
radius_mean                 0
texture_mean                0
perimeter_mean              0
area_mean                   0
smoothness_mean             0
compactness_mean            0
concavity_mean              0
concave points_mean         0
symmetry_mean               0
fractal_dimension_mean      0
radius_se                   0
texture_se                  0
perimeter_se                0
area_se                     0
smoothness_se               0
compactness_se              0
concavity_se                0
concave points_se           0
symmetry_se                 0
fractal_dimension_se        0
radius_worst                0
texture_worst               0
perimeter_worst             0
area_worst                  0
smoothness_worst            0
compactness_worst           0
```

```
concavity_worst              0
concave points_worst         0
symmetry_worst               0
fractal_dimension_worst      0
Unnamed: 32                569
dtype: int64
```

[48]:
```
# return the size of dataset
df.shape
```

[48]:
```
(569, 33)
```

[49]:
```
# remove the column
df=df.dropna(axis=1)
```

[50]:
```
# shape of dataset after removing the null column
df.shape
```

[50]:
```
(569, 32)
```

[51]:
```
# describe the dataset
df.describe()
```

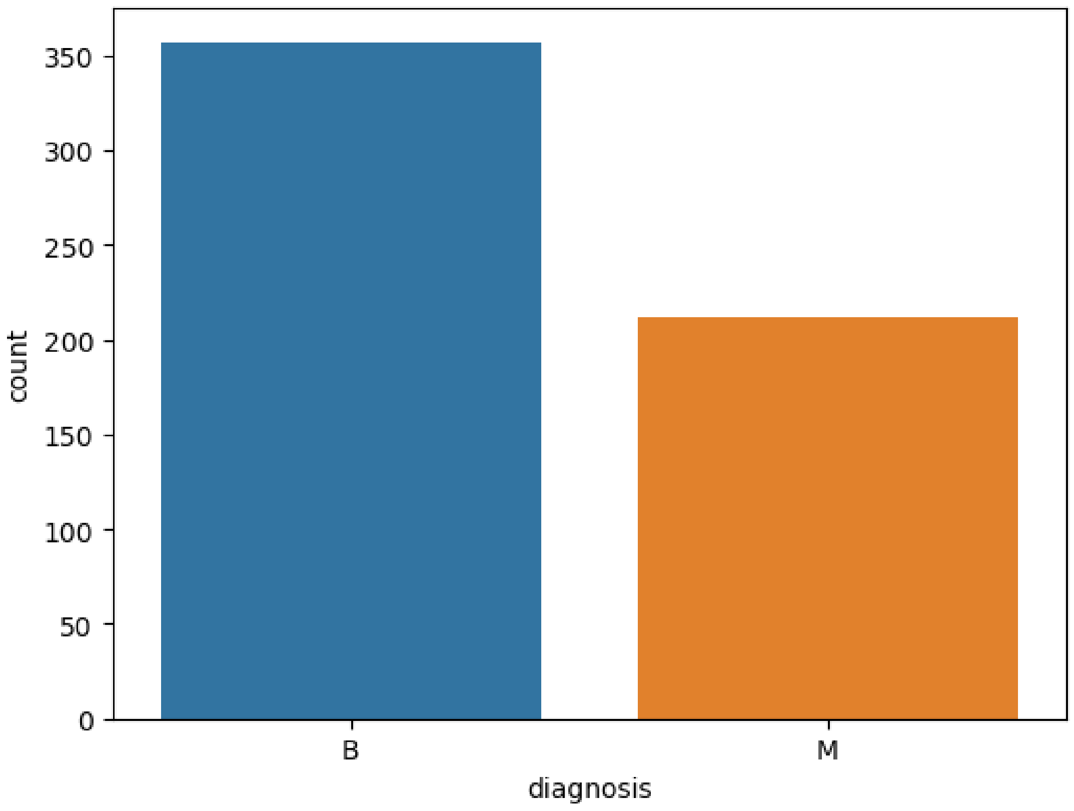
[51]:
```
                 id  radius_mean  texture_mean  perimeter_mean    area_mean  \
count  5.690000e+02   569.000000    569.000000      569.000000   569.000000
mean   3.037183e+07    14.127292     19.289649       91.969033   654.889104
std    1.250206e+08     3.524049      4.301036       24.298981   351.914129
min    8.670000e+03     6.981000      9.710000       43.790000   143.500000
25%    8.692180e+05    11.700000     16.170000       75.170000   420.300000
50%    9.060240e+05    13.370000     18.840000       86.240000   551.100000
75%    8.813129e+06    15.780000     21.800000      104.100000   782.700000
max    9.113205e+08    28.110000     39.280000      188.500000  2501.000000

       smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
count     569.000000  …    569.000000     569.000000       569.000000
mean        0.181162  …     16.269190      25.677223       107.261213
std         0.027414  …      4.833242       6.146258        33.602542
```

```
min         0.106000  …     7.930000     12.020000        50.410000
25%         0.161900  …    13.010000     21.080000        84.110000
50%         0.179200  …    14.970000     25.410000        97.660000
75%         0.195700  …    18.790000     29.720000       125.400000
max         0.304000  …    36.040000     49.540000       251.200000

        area_worst  smoothness_worst  compactness_worst  concavity_worst  \
count   569.000000        569.000000         569.000000       569.000000
mean    880.583128          0.132369           0.254265         0.272188
std     569.356993          0.022832           0.157336         0.208624
min     185.200000          0.071170           0.027290         0.000000
25%     515.300000          0.116600           0.147200         0.114500
50%     686.500000          0.131300           0.211900         0.226700
75%    1084.000000          0.146000           0.339100         0.382900
max    4254.000000          0.222600           1.058000         1.252000

       concave points_worst  symmetry_worst  fractal_dimension_worst
count            569.000000      569.000000               569.000000
mean               0.114606        0.290076                 0.083946
std                0.065732        0.061867                 0.018061
min                0.000000        0.156500                 0.055040
25%                0.064930        0.250400                 0.071460
50%                0.099930        0.282200                 0.080040
75%                0.161400        0.317900                 0.092080
max                0.291000        0.663800                 0.207500

[8 rows x 31 columns]
```

[52]:
```python
# Get the count of malignant (M) and benign (B) cells
diagnosis_counts = df['diagnosis'].value_counts()

# Display the counts
print(diagnosis_counts)
```

```
B    357
M    212
Name: diagnosis, dtype: int64
```

[53]:
```python
# Plot the count using seaborn
sns.countplot(x='diagnosis', data=df, label="count", order=diagnosis_counts.
 ↪index)
plt.show()
```



[55]:
```
# label encoding (convert the value of M and B into 1 and 0)
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
df['diagnosis'] = labelencoder_Y.fit_transform(df['diagnosis'].values)
```

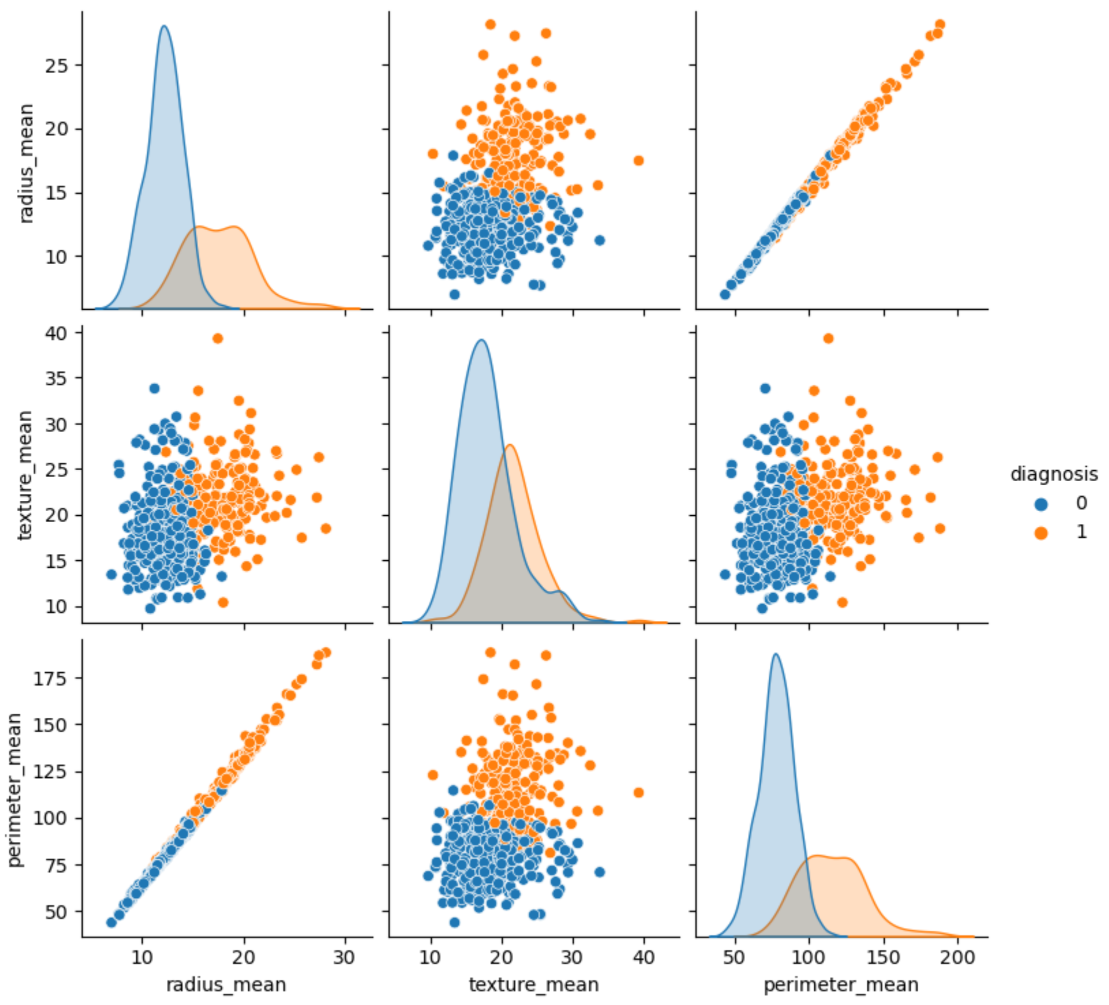
[56]:
```
df.head()
```

[56]:
```
         id  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302          1        17.99         10.38          122.80     1001.0
1    842517          1        20.57         17.77          132.90     1326.0
2  84300903          1        19.69         21.25          130.00     1203.0
3  84348301          1        11.42         20.38           77.58      386.1
4  84358402          1        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430
```

```
   …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  …         25.38          17.33           184.60      2019.0
1  …         24.99          23.41           158.80      1956.0
2  …         23.57          25.53           152.50      1709.0
3  …         14.91          26.50            98.87       567.7
4  …         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

[57]:
```
sns.pairplot(df.iloc[:,1:5],hue="diagnosis")
```

[57]:
```
<seaborn.axisgrid.PairGrid at 0x1a7c1976a90>
```



```
[58]: # get the correlation
      df.iloc[:,1:32].corr()
```

```
[58]:                          diagnosis  radius_mean  texture_mean  perimeter_mean  \
      diagnosis                 1.000000     0.730029      0.415185        0.742636
      radius_mean               0.730029     1.000000      0.323782        0.997855
      texture_mean              0.415185     0.323782      1.000000        0.329533
      perimeter_mean            0.742636     0.997855      0.329533        1.000000
      area_mean                 0.708984     0.987357      0.321086        0.986507
      smoothness_mean           0.358560     0.170581     -0.023389        0.207278
      compactness_mean          0.596534     0.506124      0.236702        0.556936
      concavity_mean            0.696360     0.676764      0.302418        0.716136
      concave points_mean       0.776614     0.822529      0.293464        0.850977
      symmetry_mean             0.330499     0.147741      0.071401        0.183027
      fractal_dimension_mean   -0.012838    -0.311631     -0.076437       -0.261477
      radius_se                 0.567134     0.679090      0.275869        0.691765
```

```
texture_se               -0.008303    -0.097317      0.386358        -0.086761
perimeter_se              0.556141     0.674172      0.281673         0.693135
area_se                   0.548236     0.735864      0.259845         0.744983
smoothness_se            -0.067016    -0.222600      0.006614        -0.202694
compactness_se            0.292999     0.206000      0.191975         0.250744
concavity_se              0.253730     0.194204      0.143293         0.228082
concave points_se         0.408042     0.376169      0.163851         0.407217
symmetry_se              -0.006522    -0.104321      0.009127        -0.081629
fractal_dimension_se      0.077972    -0.042641      0.054458        -0.005523
radius_worst              0.776454     0.969539      0.352573         0.969476
texture_worst             0.456903     0.297008      0.912045         0.303038
perimeter_worst           0.782914     0.965137      0.358040         0.970387
area_worst                0.733825     0.941082      0.343546         0.941550
smoothness_worst          0.421465     0.119616      0.077503         0.150549
compactness_worst         0.590998     0.413463      0.277830         0.455774
concavity_worst           0.659610     0.526911      0.301025         0.563879
concave points_worst      0.793566     0.744214      0.295316         0.771241
symmetry_worst            0.416294     0.163953      0.105008         0.189115
fractal_dimension_worst   0.323872     0.007066      0.119205         0.051019

                         area_mean  smoothness_mean  compactness_mean  \
diagnosis                 0.708984         0.358560          0.596534
radius_mean               0.987357         0.170581          0.506124
texture_mean              0.321086        -0.023389          0.236702
perimeter_mean            0.986507         0.207278          0.556936
area_mean                 1.000000         0.177028          0.498502
smoothness_mean           0.177028         1.000000          0.659123
compactness_mean          0.498502         0.659123          1.000000
concavity_mean            0.685983         0.521984          0.883121
concave points_mean       0.823269         0.553695          0.831135
symmetry_mean             0.151293         0.557775          0.602641
fractal_dimension_mean   -0.283110         0.584792          0.565369
radius_se                 0.732562         0.301467          0.497473
texture_se               -0.066280         0.068406          0.046205
perimeter_se              0.726628         0.296092          0.548905
area_se                   0.800086         0.246552          0.455653
smoothness_se            -0.166777         0.332375          0.135299
compactness_se            0.212583         0.318943          0.738722
concavity_se              0.207660         0.248396          0.570517
concave points_se         0.372320         0.380676          0.642262
symmetry_se              -0.072497         0.200774          0.229977
fractal_dimension_se     -0.019887         0.283607          0.507318
radius_worst              0.962746         0.213120          0.535315
texture_worst             0.287489         0.036072          0.248133
perimeter_worst           0.959120         0.238853          0.590210
area_worst                0.959213         0.206718          0.509604
smoothness_worst          0.123523         0.805324          0.565541
```

```
compactness_worst        0.390410        0.472468          0.865809
concavity_worst          0.512606        0.434926          0.816275
concave points_worst     0.722017        0.503053          0.815573
symmetry_worst           0.143570        0.394309          0.510223
fractal_dimension_worst  0.003738        0.499316          0.687382

                        concavity_mean  concave points_mean  symmetry_mean  \
diagnosis                     0.696360             0.776614       0.330499
radius_mean                   0.676764             0.822529       0.147741
texture_mean                  0.302418             0.293464       0.071401
perimeter_mean                0.716136             0.850977       0.183027
area_mean                     0.685983             0.823269       0.151293
smoothness_mean               0.521984             0.553695       0.557775
compactness_mean              0.883121             0.831135       0.602641
concavity_mean                1.000000             0.921391       0.500667
concave points_mean           0.921391             1.000000       0.462497
symmetry_mean                 0.500667             0.462497       1.000000
fractal_dimension_mean        0.336783             0.166917       0.479921
radius_se                     0.631925             0.698050       0.303379
texture_se                    0.076218             0.021480       0.128053
perimeter_se                  0.660391             0.710650       0.313893
area_se                       0.617427             0.690299       0.223970
smoothness_se                 0.098564             0.027653       0.187321
compactness_se                0.670279             0.490424       0.421659
concavity_se                  0.691270             0.439167       0.342627
concave points_se             0.683260             0.615634       0.393298
symmetry_se                   0.178009             0.095351       0.449137
fractal_dimension_se          0.449301             0.257584       0.331786
radius_worst                  0.688236             0.830318       0.185728
texture_worst                 0.299879             0.292752       0.090651
perimeter_worst               0.729565             0.855923       0.219169
area_worst                    0.675987             0.809630       0.177193
smoothness_worst              0.448822             0.452753       0.426675
compactness_worst             0.754968             0.667454       0.473200
concavity_worst               0.884103             0.752399       0.433721
concave points_worst          0.861323             0.910155       0.430297
symmetry_worst                0.409464             0.375744       0.699826
fractal_dimension_worst       0.514930             0.368661       0.438413

                        …  radius_worst  texture_worst  perimeter_worst  \
diagnosis               …      0.776454       0.456903         0.782914
radius_mean             …      0.969539       0.297008         0.965137
texture_mean            …      0.352573       0.912045         0.358040
perimeter_mean          …      0.969476       0.303038         0.970387
area_mean               …      0.962746       0.287489         0.959120
smoothness_mean         …      0.213120       0.036072         0.238853
compactness_mean        …      0.535315       0.248133         0.590210
```

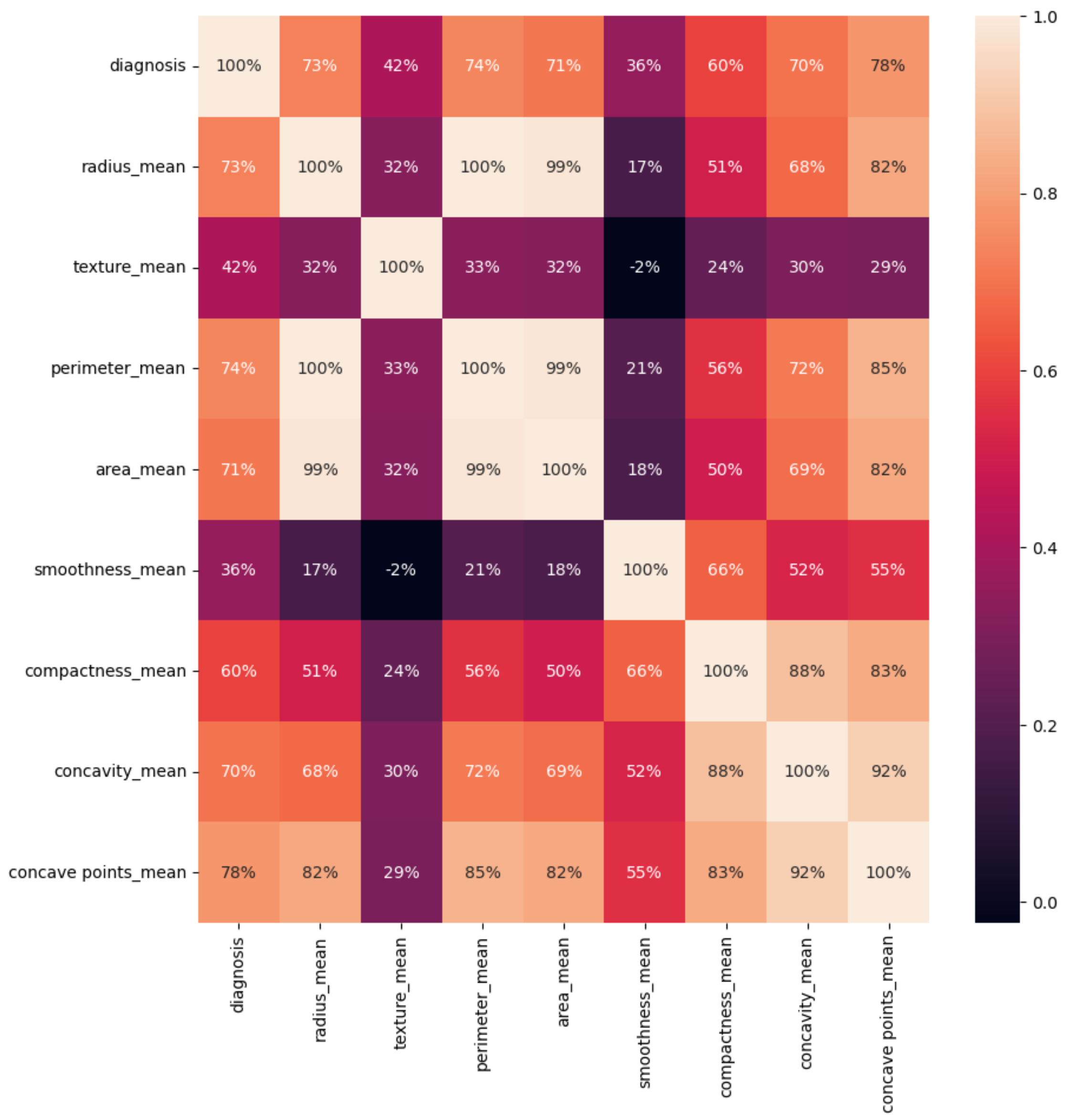
```
concavity_mean          …    0.688236       0.299879          0.729565
concave points_mean     …    0.830318       0.292752          0.855923
symmetry_mean           …    0.185728       0.090651          0.219169
fractal_dimension_mean  …   -0.253691      -0.051269         -0.205151
radius_se               …    0.715065       0.194799          0.719684
texture_se              …   -0.111690       0.409003         -0.102242
perimeter_se            …    0.697201       0.200371          0.721031
area_se                 …    0.757373       0.196497          0.761213
smoothness_se           …   -0.230691      -0.074743         -0.217304
compactness_se          …    0.204607       0.143003          0.260516
concavity_se            …    0.186904       0.100241          0.226680
concave points_se       …    0.358127       0.086741          0.394999
symmetry_se             …   -0.128121      -0.077473         -0.103753
fractal_dimension_se    …   -0.037488      -0.003195         -0.001000
radius_worst            …    1.000000       0.359921          0.993708
texture_worst           …    0.359921       1.000000          0.365098
perimeter_worst         …    0.993708       0.365098          1.000000
area_worst              …    0.984015       0.345842          0.977578
smoothness_worst        …    0.216574       0.225429          0.236775
compactness_worst       …    0.475820       0.360832          0.529408
concavity_worst         …    0.573975       0.368366          0.618344
concave points_worst    …    0.787424       0.359755          0.816322
symmetry_worst          …    0.243529       0.233027          0.269493
fractal_dimension_worst …    0.093492       0.219122          0.138957

                         area_worst  smoothness_worst  compactness_worst  \
diagnosis                  0.733825          0.421465           0.590998
radius_mean                0.941082          0.119616           0.413463
texture_mean               0.343546          0.077503           0.277830
perimeter_mean             0.941550          0.150549           0.455774
area_mean                  0.959213          0.123523           0.390410
smoothness_mean            0.206718          0.805324           0.472468
compactness_mean           0.509604          0.565541           0.865809
concavity_mean             0.675987          0.448822           0.754968
concave points_mean        0.809630          0.452753           0.667454
symmetry_mean              0.177193          0.426675           0.473200
fractal_dimension_mean    -0.231854          0.504942           0.458798
radius_se                  0.751548          0.141919           0.287103
texture_se                -0.083195         -0.073658          -0.092439
perimeter_se               0.730713          0.130054           0.341919
area_se                    0.811408          0.125389           0.283257
smoothness_se             -0.182195          0.314457          -0.055558
compactness_se             0.199371          0.227394           0.678780
concavity_se               0.188353          0.168481           0.484858
concave points_se          0.342271          0.215351           0.452888
symmetry_se               -0.110343         -0.012662           0.060255
fractal_dimension_se      -0.022736          0.170568           0.390159
```

```
radius_worst             0.984015         0.216574          0.475820
texture_worst            0.345842         0.225429          0.360832
perimeter_worst          0.977578         0.236775          0.529408
area_worst               1.000000         0.209145          0.438296
smoothness_worst         0.209145         1.000000          0.568187
compactness_worst        0.438296         0.568187          1.000000
concavity_worst          0.543331         0.518523          0.892261
concave points_worst     0.747419         0.547691          0.801080
symmetry_worst           0.209146         0.493838          0.614441
fractal_dimension_worst  0.079647         0.617624          0.810455

                         concavity_worst  concave points_worst  \
diagnosis                       0.659610              0.793566
radius_mean                     0.526911              0.744214
texture_mean                    0.301025              0.295316
perimeter_mean                  0.563879              0.771241
area_mean                       0.512606              0.722017
smoothness_mean                 0.434926              0.503053
compactness_mean                0.816275              0.815573
concavity_mean                  0.884103              0.861323
concave points_mean             0.752399              0.910155
symmetry_mean                   0.433721              0.430297
fractal_dimension_mean          0.346234              0.175325
radius_se                       0.380585              0.531062
texture_se                     -0.068956             -0.119638
perimeter_se                    0.418899              0.554897
area_se                         0.385100              0.538166
smoothness_se                  -0.058298             -0.102007
compactness_se                  0.639147              0.483208
concavity_se                    0.662564              0.440472
concave points_se               0.549592              0.602450
symmetry_se                     0.037119             -0.030413
fractal_dimension_se            0.379975              0.215204
radius_worst                    0.573975              0.787424
texture_worst                   0.368366              0.359755
perimeter_worst                 0.618344              0.816322
area_worst                      0.543331              0.747419
smoothness_worst                0.518523              0.547691
compactness_worst               0.892261              0.801080
concavity_worst                 1.000000              0.855434
concave points_worst            0.855434              1.000000
symmetry_worst                  0.532520              0.502528
fractal_dimension_worst         0.686511              0.511114

                         symmetry_worst  fractal_dimension_worst
diagnosis                      0.416294                 0.323872
radius_mean                    0.163953                 0.007066
```

```
texture_mean                   0.105008                 0.119205
perimeter_mean                 0.189115                 0.051019
area_mean                      0.143570                 0.003738
smoothness_mean                0.394309                 0.499316
compactness_mean               0.510223                 0.687382
concavity_mean                 0.409464                 0.514930
concave points_mean            0.375744                 0.368661
symmetry_mean                  0.699826                 0.438413
fractal_dimension_mean         0.334019                 0.767297
radius_se                      0.094543                 0.049559
texture_se                    -0.128215                -0.045655
perimeter_se                   0.109930                 0.085433
area_se                        0.074126                 0.017539
smoothness_se                 -0.107342                 0.101480
compactness_se                 0.277878                 0.590973
concavity_se                   0.197788                 0.439329
concave points_se              0.143116                 0.310655
symmetry_se                    0.389402                 0.078079
fractal_dimension_se           0.111094                 0.591328
radius_worst                   0.243529                 0.093492
texture_worst                  0.233027                 0.219122
perimeter_worst                0.269493                 0.138957
area_worst                     0.209146                 0.079647
smoothness_worst               0.493838                 0.617624
compactness_worst              0.614441                 0.810455
concavity_worst                0.532520                 0.686511
concave points_worst           0.502528                 0.511114
symmetry_worst                 1.000000                 0.537848
fractal_dimension_worst        0.537848                 1.000000

[31 rows x 31 columns]
```

[59]:
```
# visualize the correlation
plt.figure(figsize=(10,10))
sns.heatmap(df.iloc[:,1:10].corr(),annot=True,fmt=".0%")
```

[59]:
```
<Axes: >
```



	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
diagnosis	100%	73%	42%	74%	71%	36%	60%	70%	78%
radius_mean	73%	100%	32%	100%	99%	17%	51%	68%	82%
texture_mean	42%	32%	100%	33%	32%	-2%	24%	30%	29%
perimeter_mean	74%	100%	33%	100%	99%	21%	56%	72%	85%
area_mean	71%	99%	32%	99%	100%	18%	50%	69%	82%
smoothness_mean	36%	17%	-2%	21%	18%	100%	66%	52%	55%
compactness_mean	60%	51%	24%	56%	50%	66%	100%	88%	83%
concavity_mean	70%	68%	30%	72%	69%	52%	88%	100%	92%
concave points_mean	78%	82%	29%	85%	82%	55%	83%	92%	100%

```
[77]: # split the dataset into dependent(X) and Independent(Y) datasets
      X=df.iloc[:,2:31].values
      Y=df.iloc[:,1].values
```

```
[78]: # spliting the data into trainning and test dateset
      from sklearn.model_selection import train_test_split
      X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.
       ↪20,random_state=0)
```

```
[79]: # feature scaling
      scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[90]:
```
# models/ Algorithms

def models(X_train, Y_train):
    # Logistic Regression
    from sklearn.linear_model import LogisticRegression
    log = LogisticRegression(random_state=0)
    log.fit(X_train, Y_train)

    # Decision Tree
    from sklearn.tree import DecisionTreeClassifier
    tree = DecisionTreeClassifier(random_state=0, criterion="entropy")
    tree.fit(X_train, Y_train)

    # Random Forest
    from sklearn.ensemble import RandomForestClassifier
    forest = RandomForestClassifier(random_state=0, criterion="entropy",␣
 ↪n_estimators=10)
    forest.fit(X_train, Y_train)

    print('[0] Logistic Regression accuracy:', log.score(X_train, Y_train))
    print('[1] Decision Tree accuracy:', tree.score(X_train, Y_train))
    print('[2] Random Forest accuracy:', forest.score(X_train, Y_train))

    return log, tree, forest
```

[120]:
```
# k-fold cross-validation
def cross_validation(models, X, Y, k=5):
    for i, model in enumerate(models):
        scores = cross_val_score(model, X, Y, cv=k, scoring='accuracy')
        print(f'Model {i} Cross-Validation Accuracy: {np.mean(scores):.4f} (+/-␣
 ↪{np.std(scores):.4f})')
```

[130]:
```
# testing the models/result
models_list = models(X_train, Y_train)
cross_validation(models_list, X_train, Y_train)
```

```
[0] Logistic Regression accuracy: 0.9912087912087912
[1] Decision Tree accuracy: 1.0
[2] Random Forest accuracy: 0.9978021978021978
Model 0 Cross-Validation Accuracy: 0.9824 (+/- 0.0149)
Model 1 Cross-Validation Accuracy: 0.9209 (+/- 0.0189)
Model 2 Cross-Validation Accuracy: 0.9516 (+/- 0.0247)
```

[146]:
```
# k-fold cross-validation for the test dataset
def cross_validation_test(models, X_test, Y_test, k=5):
    for i, model in enumerate(models):
        scores = cross_val_score(model, X_test, Y_test, cv=k,␣
 ↪scoring='accuracy')
        print(f'Model {i} Cross-Validation Accuracy on Test Dataset: {np.
 ↪mean(scores):.4f} (+/- {np.std(scores):.4f})')

cross_validation_test(models_list, X_test, Y_test)
```

```
Model 0 Cross-Validation Accuracy on Test Dataset: 0.9644 (+/- 0.0338)
Model 1 Cross-Validation Accuracy on Test Dataset: 0.9289 (+/- 0.0618)
Model 2 Cross-Validation Accuracy on Test Dataset: 0.9209 (+/- 0.0798)
```

[147]:
```
# grid search for hyperparameter tuning
def grid_search(model, param_grid, X_train, Y_train):
    grid_search = GridSearchCV(model, param_grid, cv=5, scoring='accuracy',␣
 ↪n_jobs=-1)
    grid_search.fit(X_train, Y_train)
    best_params = grid_search.best_params_
    best_model = grid_search.best_estimator_
    return best_model, best_params
```

[144]:
```
# Grid search for hyperparameter tuning for all three classifiers
param_grid_logreg = {'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}
param_grid_tree = {'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5,␣
 ↪10], 'min_samples_leaf': [1, 2, 4]}
param_grid_forest = {'n_estimators': [10, 50, 100, 200], 'max_depth': [None,␣
 ↪10, 20, 30], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4]}

best_logreg_model, best_logreg_params = grid_search(models_list[0],␣
 ↪param_grid_logreg, X_train, Y_train)
best_tree_model, best_tree_params = grid_search(models_list[1],␣
 ↪param_grid_tree, X_train, Y_train)
best_forest_model, best_forest_params = grid_search(models_list[2],␣
 ↪param_grid_forest, X_train, Y_train)

# Display the best parameters for each model
print("Best Logistic Regression Model:")
print(best_logreg_params)
print("Best Decision Tree Model:")
print(best_tree_params)
print("Best Random Forest Model:")
print(best_forest_params)
```

```
Best Logistic Regression Model:
{'C': 1}
```

```
Best Decision Tree Model:
{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}
Best Random Forest Model:
{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2,
'n_estimators': 50}
```

[145]:
```
# Report the final performance on the test dataset
def report_final_performance(models, X_test, Y_test, model_names):
    for model, name in zip(models, model_names):
        print(f"\n{name} Test Performance:")
        y_pred = model.predict(X_test)
        print(classification_report(Y_test, y_pred))
        print('Accuracy : ', accuracy_score(Y_test, y_pred))

# Usage
model_names = ['Logistic Regression', 'Decision Tree', 'Random Forest']
report_final_performance(models_list, X_test, Y_test, model_names)
```

```
Logistic Regression Test Performance:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96        67
           1       0.96      0.94      0.95        47

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114

Accuracy :  0.956140350877193

Decision Tree Test Performance:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94        67
           1       0.90      0.94      0.92        47

    accuracy                           0.93       114
   macro avg       0.93      0.93      0.93       114
weighted avg       0.93      0.93      0.93       114

Accuracy :  0.9298245614035088

Random Forest Test Performance:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        67
           1       0.98      0.96      0.97        47
    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Accuracy :  0.9736842105263158
```