Lab Project 1 : Breast Cancer Classification

March 23, 2022

1 UE19EC353: Machine Learning · Jan - May 2022 · Lab Project 1

Given the features and the target, build a Machine Learning Classification model that can classify from a given set of features, if the cancer is Benign or Malignant. You can use any classification algorithm.

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\venka\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[3]:
```
df=pd.read_csv('C:
↪\\Users\\venka\\Desktop\\TAMachineLearning\\Project_Assignments\\Solutions\\data.
↪csv')
#https://www.kaggle.com/uciml/breast-cancer-wisconsin-data -> get data from here
```

[4]:
```
df.head()
```

[4]:
```
         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
```

```
   ...  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]
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 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
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

[6]:
```
df.describe()
```

[6]:
```
                 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]
```

[7]:
```
df['diagnosis']=pd.get_dummies(df['diagnosis']) #Convert string to labeled␣
↪numbers
```

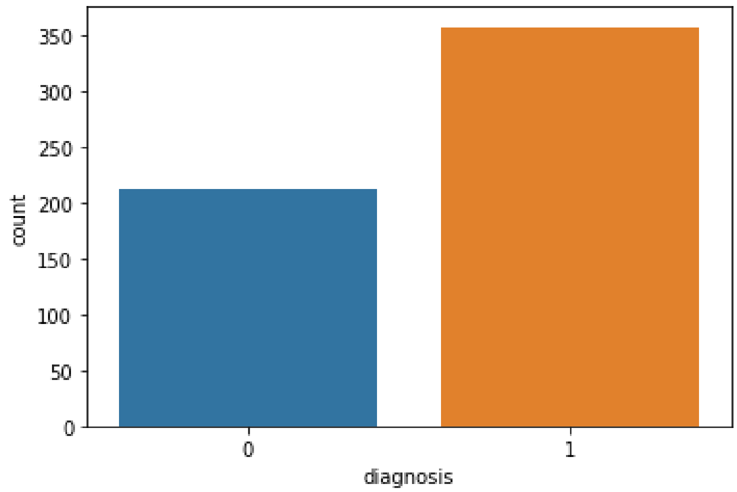
[8]:
```
df
```

[8]:
```
           id  diagnosis  radius_mean  texture_mean  perimeter_mean  \
0      842302          0        17.99         10.38          122.80
1      842517          0        20.57         17.77          132.90
2    84300903          0        19.69         21.25          130.00
3    84348301          0        11.42         20.38           77.58
4    84358402          0        20.29         14.34          135.10
..        ...        ...          ...           ...             ...
564    926424          0        21.56         22.39          142.00
565    926682          0        20.13         28.25          131.20
566    926954          0        16.60         28.08          108.30
567    927241          0        20.60         29.33          140.10
568     92751          1         7.76         24.54           47.92

     area_mean  smoothness_mean  compactness_mean  concavity_mean  \
0       1001.0          0.11840           0.27760         0.30010
1       1326.0          0.08474           0.07864         0.08690
2       1203.0          0.10960           0.15990         0.19740
3        386.1          0.14250           0.28390         0.24140
4       1297.0          0.10030           0.13280         0.19800
..         ...              ...               ...             ...
```

```
564     1479.0          0.11100          0.11590          0.24390
565     1261.0          0.09780          0.10340          0.14400
566      858.1          0.08455          0.10230          0.09251
567     1265.0          0.11780          0.27700          0.35140
568      181.0          0.05263          0.04362          0.00000

     concave points_mean  ...  radius_worst  texture_worst  perimeter_worst  \
0                0.14710  ...        25.380          17.33           184.60
1                0.07017  ...        24.990          23.41           158.80
2                0.12790  ...        23.570          25.53           152.50
3                0.10520  ...        14.910          26.50            98.87
4                0.10430  ...        22.540          16.67           152.20
..                   ...  ...           ...            ...              ...
564              0.13890  ...        25.450          26.40           166.10
565              0.09791  ...        23.690          38.25           155.00
566              0.05302  ...        18.980          34.12           126.70
567              0.15200  ...        25.740          39.42           184.60
568              0.00000  ...         9.456          30.37            59.16

     area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0        2019.0           0.16220            0.66560           0.7119
1        1956.0           0.12380            0.18660           0.2416
2        1709.0           0.14440            0.42450           0.4504
3         567.7           0.20980            0.86630           0.6869
4        1575.0           0.13740            0.20500           0.4000
..          ...               ...                ...              ...
564      2027.0           0.14100            0.21130           0.4107
565      1731.0           0.11660            0.19220           0.3215
566      1124.0           0.11390            0.30940           0.3403
567      1821.0           0.16500            0.86810           0.9387
568       268.6           0.08996            0.06444           0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst
0                  0.2654          0.4601                  0.11890
1                  0.1860          0.2750                  0.08902
2                  0.2430          0.3613                  0.08758
3                  0.2575          0.6638                  0.17300
4                  0.1625          0.2364                  0.07678
..                    ...             ...                      ...
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[569 rows x 32 columns]
```

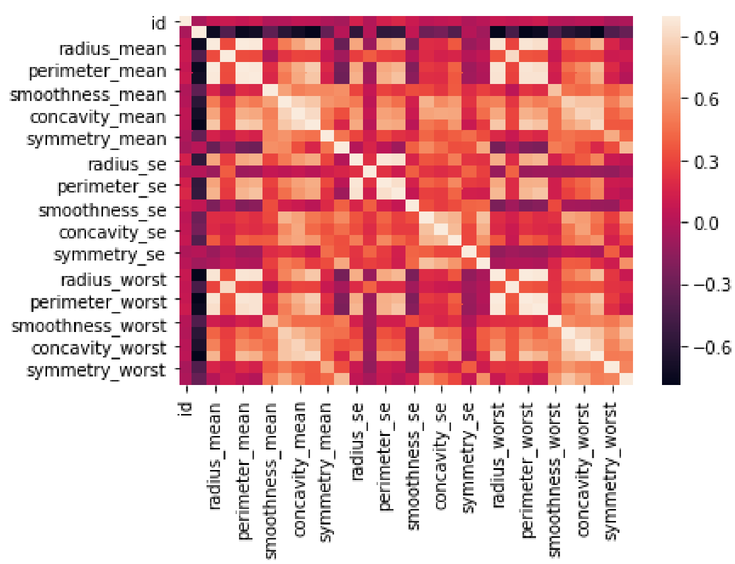
[9]:
```
# See number of cases of Malignant and Benign
sns.countplot(x=df['diagnosis'],data=df)
```

[9]: `<matplotlib.axes._subplots.AxesSubplot at 0x260cbed1b38>`



[10]:
```
sns.heatmap(df.corr()) # Correlation Map
```

[10]: `<matplotlib.axes._subplots.AxesSubplot at 0x260cdf858d0>`



[11]:
```
X=df.drop('diagnosis',axis=1).values # Everything except the 'Diagnosis' column
↪makes up the Features
```

[12]:
```
y=df['diagnosis'].values # The target is the 'Diagnosis' Column
```

[13]:
```
from sklearn.model_selection import train_test_split
```

[14]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=1)␣
↪# Splitting data into training 80% and testing 20%
```

[15]:
```
from sklearn.preprocessing import MinMaxScaler
```

[16]:
```
scaler=MinMaxScaler() # Min Max Scaling, other scalers can be used as well
```

[17]:
```
X_train=scaler.fit_transform(X_train)
```

[18]:
```
X_test=scaler.fit_transform(X_test)
```

[19]:
```
from sklearn.linear_model import LogisticRegression
```

[20]:
```
model_LR = LogisticRegression(random_state=0).fit(X_train, y_train) # Logistic␣
↪Regression Model
```

[21]:
```
# Predicting values
predictions=model_LR.predict(X_test)
```

[22]:
```
print(model_LR.score(X_test,y_test))
```

```
0.9736842105263158
```

[23]:
```
from sklearn.metrics import classification_report
```

[24]:
```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        42
           1       0.96      1.00      0.98        72

    accuracy                           0.97       114
   macro avg       0.98      0.96      0.97       114
weighted avg       0.97      0.97      0.97       114
```

2 Contact for Doubts:

- Dr. B N Krupa : bnkrupa@pes.edu
- K Venkat Ramnan (TA) : venkatramnank@pesu.pes.edu