In [1]:
```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data = pd.read_csv("breast cancer data.csv")
```

In [3]:
```
data.drop(['Unnamed: 32',"id"], axis=1, inplace=True)
data.diagnosis = [1 if each == "M" else 0 for each in data.diagnosis]
y = data.diagnosis.values
x_data = data.drop(['diagnosis'], axis=1)
```

In [4]:
```
x_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   radius_mean              569 non-null    float64
 1   texture_mean             569 non-null    float64
 2   perimeter_mean           569 non-null    float64
 3   area_mean                569 non-null    float64
 4   smoothness_mean          569 non-null    float64
 5   compactness_mean         569 non-null    float64
 6   concavity_mean           569 non-null    float64
 7   concave points_mean      569 non-null    float64
 8   symmetry_mean            569 non-null    float64
 9   fractal_dimension_mean   569 non-null    float64
 10  radius_se                569 non-null    float64
 11  texture_se               569 non-null    float64
 12  perimeter_se             569 non-null    float64
 13  area_se                  569 non-null    float64
 14  smoothness_se            569 non-null    float64
 15  compactness_se           569 non-null    float64
 16  concavity_se             569 non-null    float64
 17  concave points_se        569 non-null    float64
 18  symmetry_se              569 non-null    float64
 19  fractal_dimension_se     569 non-null    float64
 20  radius_worst             569 non-null    float64
 21  texture_worst            569 non-null    float64
 22  perimeter_worst          569 non-null    float64
 23  area_worst               569 non-null    float64
 24  smoothness_worst         569 non-null    float64
 25  compactness_worst        569 non-null    float64
 26  concavity_worst          569 non-null    float64
 27  concave points_worst     569 non-null    float64
 28  symmetry_worst           569 non-null    float64
 29  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30)
memory usage: 133.5 KB
```

In [6]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [7]:
```
scaler = MinMaxScaler(feature_range=(0, 1))
x_scaled=scaler.fit_transform(x_data)
x_scaled
```

Out[7]:
```
array([[0.52103744, 0.0226581 , 0.54598853, ..., 0.91202749, 0.59846245,
        0.41886396],
       [0.64314449, 0.27257355, 0.61578329, ..., 0.63917526, 0.23358959,
        0.22287813],
       [0.60149557, 0.3902604 , 0.59574321, ..., 0.83505155, 0.40370589,
        0.21343303],
       ...,
       [0.45525108, 0.62123774, 0.44578813, ..., 0.48728522, 0.12872068,
        0.1519087 ],
       [0.64456434, 0.66351031, 0.66553797, ..., 0.91065292, 0.49714173,
        0.45231536],
       [0.03686876, 0.50152181, 0.02853984, ..., 0.        , 0.25744136,
        0.10068215]])
```

In [8]:
```
scaled_data = pd.DataFrame(x_scaled, columns=x_data.columns)
```

In [9]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(scaled_data, y, test_size=0.15,
```

In [18]:
```
## Implementing Randomforest model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(max_depth=6, random_state=0, n_estimators=10)
model.fit(x_train, y_train)
```

Out[18]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=6, n_estimators=10, random_state=0)
```

In [19]:
```
y_pred = model.predict(x_test)
```

In [20]:
```
y_pred
```

Out[20]:
```
array([0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1],
      dtype=int64)
```

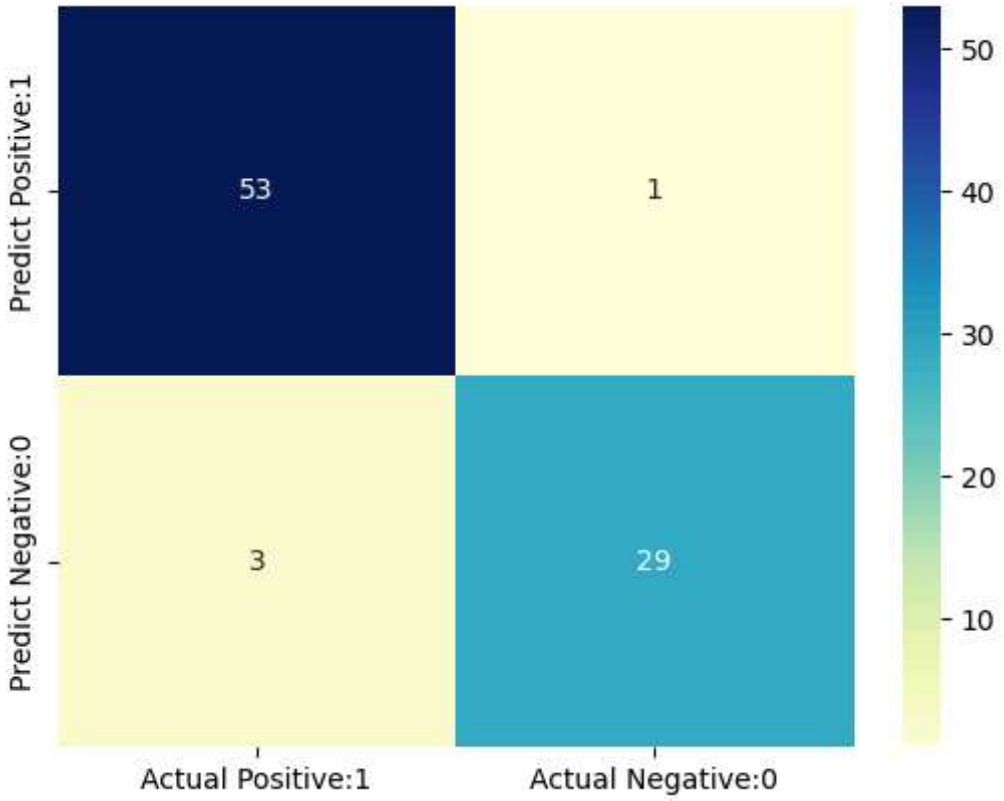
In [21]:
```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
```

In [22]:
```
cm_matrix = pd.DataFrame(data=cm, columns=['Actual Positive:1', 'Actual Negative:0'
                                   index=['Predict Positive:1', 'Predict Negative:0']

sns.heatmap(cm_matrix, annot=True, fmt='d', cmap='YlGnBu')
```

Out[22]:
```
<Axes: >
```



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

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      0.98      0.96        54
           1       0.97      0.91      0.94        32

    accuracy                           0.95        86
   macro avg       0.96      0.94      0.95        86
weighted avg       0.95      0.95      0.95        86
```

In [24]:
```
###Importing lime for model explainiability
```

In [25]:
```
pip install lime
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
     ---------------------------------------- 0.0/275.7 kB ? eta -:--:--
     - -------------------------------------- 10.2/275.7 kB ? eta -:--:--
     ----- ---------------------------------- 41.0/275.7 kB 487.6 kB/s eta 0:00:01
     ------------------------------------ - 266.2/275.7 kB 2.3 MB/s eta 0:00:01
     ---------------------------------------- 275.7/275.7 kB 2.1 MB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: matplotlib in c:\programdata\anaconda3\lib\site-packa
ges (from lime) (3.8.4)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages
(from lime) (1.26.4)
Requirement already satisfied: scipy in c:\programdata\anaconda3\lib\site-packages
(from lime) (1.13.1)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packages (f
rom lime) (4.66.4)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\user\appdata\roaming\p
ython\python312\site-packages (from lime) (1.3.1)
Requirement already satisfied: scikit-image>=0.12 in c:\programdata\anaconda3\lib\si
te-packages (from lime) (0.23.2)
Requirement already satisfied: networkx>=2.8 in c:\programdata\anaconda3\lib\site-pa
ckages (from scikit-image>=0.12->lime) (3.2.1)
Requirement already satisfied: pillow>=9.1 in c:\programdata\anaconda3\lib\site-pack
ages (from scikit-image>=0.12->lime) (10.3.0)
Requirement already satisfied: imageio>=2.33 in c:\programdata\anaconda3\lib\site-pa
ckages (from scikit-image>=0.12->lime) (2.33.1)
Requirement already satisfied: tifffile>=2022.8.12 in c:\programdata\anaconda3\lib\s
ite-packages (from scikit-image>=0.12->lime) (2023.4.12)
Requirement already satisfied: packaging>=21 in c:\programdata\anaconda3\lib\site-pa
ckages (from scikit-image>=0.12->lime) (23.2)
Requirement already satisfied: lazy-loader>=0.4 in c:\programdata\anaconda3\lib\site
-packages (from scikit-image>=0.12->lime) (0.4)
Requirement already satisfied: joblib>=1.1.1 in c:\programdata\anaconda3\lib\site-pa
ckages (from scikit-learn>=0.18->lime) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\user\appdata\roaming
\python\python312\site-packages (from scikit-learn>=0.18->lime) (3.5.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib->lime) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\programdata\anaconda3\lib\site-pac
kages (from matplotlib->lime) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->lime) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\programdata\anaconda3\lib\sit
e-packages (from matplotlib->lime) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in c:\programdata\anaconda3\lib\site
-packages (from matplotlib->lime) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\programdata\anaconda3\lib
\site-packages (from matplotlib->lime) (2.9.0.post0)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-package
s (from tqdm->lime) (0.4.6)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-package
s (from python-dateutil>=2.7->matplotlib->lime) (1.16.0)
Building wheels for collected packages: lime
  Building wheel for lime (setup.py): started
```

```
  Building wheel for lime (setup.py): finished with status 'done'
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283842 sha256=
1409a7b35c005904f568c5887cb761092bc01bef6c462f1048fafb285e6e8509
  Stored in directory: c:\users\user\appdata\local\pip\cache\wheels\e7\5d\0e\4b4fff9
a47468fed5633211fb3b76d1db43fe806a17fb7486a
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [26]:
```
import lime
import lime.lime_tabular
```

In [28]:
```
explainer = lime.lime_tabular.LimeTabularExplainer(x_train.values, feature_names=x_
                                                   class_names=['diagnosis'], verbos
```

In [29]:
```
###NOW let us look for how it explains
#1 take a sample where we have same prediction
```

In [33]:
```
a=[y_test,y_pred]
```

In [44]:
```
for i in np.arange(0,len(a[0])):
    if a[0][i]==a[1][i]:
        print(f'{i} is case matched')
    else:
        print(f'{i} is case not matched')
```

```
0 is case matched
1 is case matched
2 is case matched
3 is case matched
4 is case matched
5 is case matched
6 is case matched
7 is case matched
8 is case not matched
9 is case matched
10 is case matched
11 is case matched
12 is case matched
13 is case matched
14 is case matched
15 is case matched
16 is case matched
17 is case matched
18 is case matched
19 is case matched
20 is case not matched
21 is case matched
22 is case matched
23 is case matched
24 is case matched
25 is case matched
26 is case matched
27 is case matched
28 is case matched
29 is case matched
30 is case matched
31 is case matched
32 is case matched
33 is case matched
34 is case matched
35 is case matched
36 is case matched
37 is case matched
38 is case matched
39 is case matched
40 is case matched
41 is case matched
42 is case matched
43 is case matched
44 is case matched
45 is case matched
46 is case matched
47 is case matched
48 is case matched
49 is case matched
50 is case matched
51 is case matched
52 is case matched
53 is case matched
54 is case matched
55 is case matched
```

```
56 is case matched
57 is case matched
58 is case matched
59 is case matched
60 is case matched
61 is case matched
62 is case matched
63 is case matched
64 is case matched
65 is case matched
66 is case matched
67 is case matched
68 is case matched
69 is case matched
70 is case matched
71 is case matched
72 is case matched
73 is case matched
74 is case matched
75 is case matched
76 is case matched
77 is case not matched
78 is case matched
79 is case matched
80 is case matched
81 is case matched
82 is case not matched
83 is case matched
84 is case matched
85 is case matched
```

In [46]:
```
a[0][0],a[1][0]
```

Out[46]:
```
(0, 0)
```

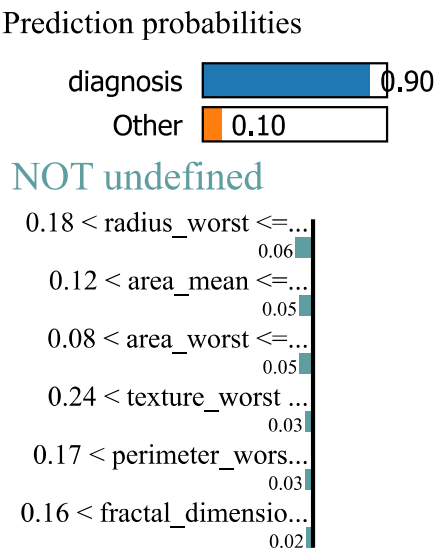
In [48]:
```
# Choose the 0th instance and use it to predict the results where there is 0 as tar
j = 0
exp = explainer.explain_instance(x_test.values[j], model.predict_proba, num_feature
```

```
Intercept 0.5217175384118667
Prediction_local [0.29172666]
Right: 0.10114315481390108
```

In [54]:
```
x_test.iloc[j,:]
```

Out[54]:
```
radius_mean               0.259785
texture_mean              0.300643
perimeter_mean            0.257757
area_mean                 0.143542
smoothness_mean           0.424483
compactness_mean          0.265076
concavity_mean            0.187559
concave points_mean       0.189911
symmetry_mean             0.436869
fractal_dimension_mean    0.290017
radius_se                 0.103060
texture_se                0.151123
perimeter_se              0.081987
area_se                   0.043870
smoothness_se             0.178128
compactness_se            0.126611
concavity_se              0.068207
concave points_se         0.196439
symmetry_se               0.139838
fractal_dimension_se      0.092976
radius_worst              0.250445
texture_worst             0.336354
perimeter_worst           0.227302
area_worst                0.121092
smoothness_worst          0.471703
compactness_worst         0.204238
concavity_worst           0.213339
concave points_worst      0.348797
symmetry_worst            0.285630
fractal_dimension_worst   0.212908
Name: 204, dtype: float64
```

In [49]:
```
exp.show_in_notebook(show_table=True)
```



Feature	Value
radius_worst	0.25
area_mean	0.14
area_worst	0.12
texture_worst	0.34
perimeter_worst	0.23
fractal_dimension_worst	0.21

In [56]:
```
a[0][1],a[1][1]
```

Out[56]: (1, 1)

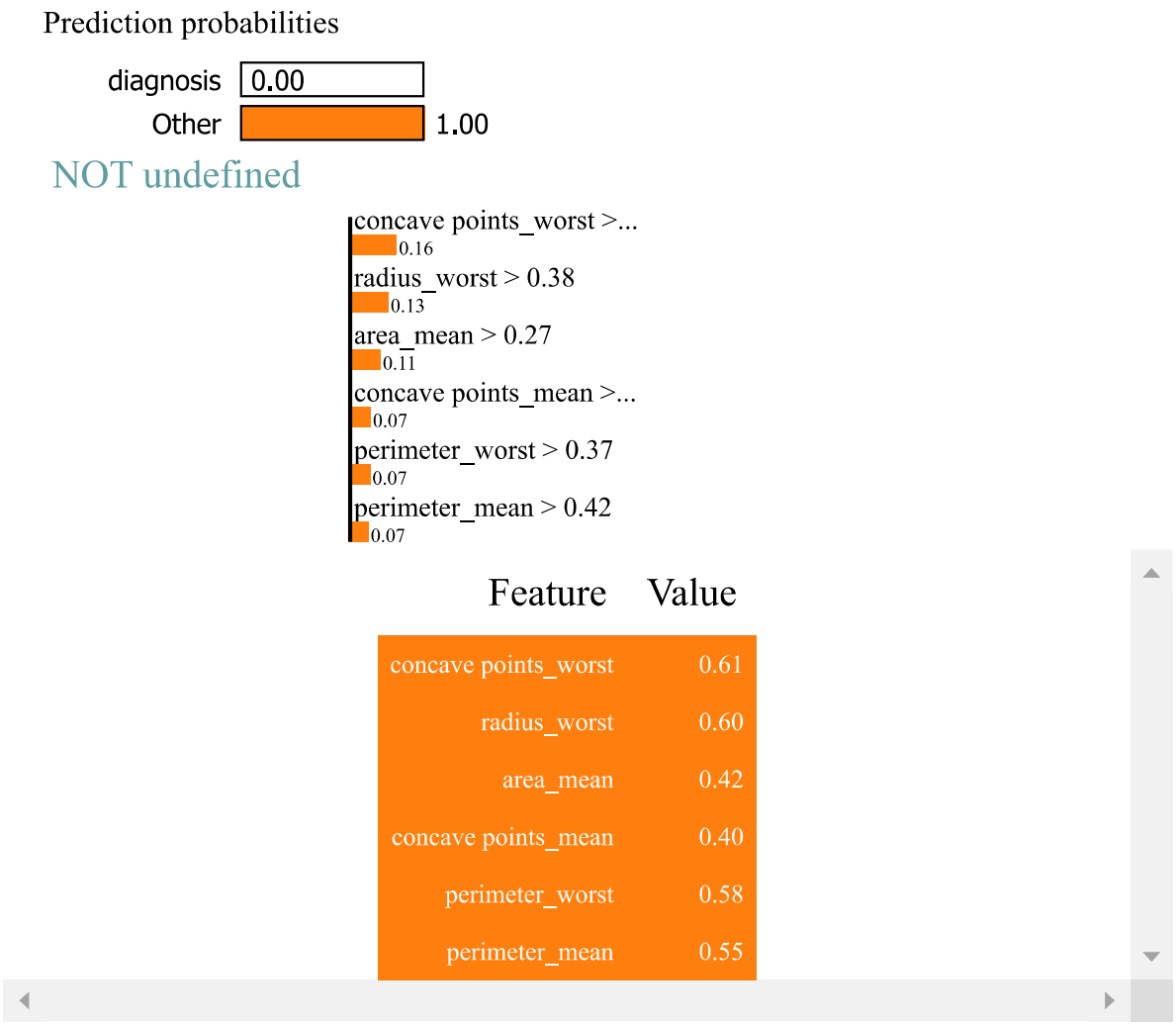
In [57]:
```
# Choose the 1st instance and use it to predict the results where there is 0 as tar
j = 1
exp = explainer.explain_instance(x_test.values[j], model.predict_proba, num_feature
```

```
Intercept 0.3179814342603986
Prediction_local [0.93108202]
Right: 1.0
```

In [58]:
```
x_test.iloc[j,:]
```

Out[58]:
```
radius_mean               0.565999
texture_mean              0.392289
perimeter_mean            0.551517
area_mean                 0.418452
smoothness_mean           0.338178
compactness_mean          0.256181
concavity_mean            0.253046
concave points_mean       0.395179
symmetry_mean             0.263636
fractal_dimension_mean    0.097936
radius_se                 0.245265
texture_se                0.096645
perimeter_se              0.222824
area_se                   0.166695
smoothness_se             0.092837
compactness_se            0.107159
concavity_se              0.057298
concave points_se         0.259519
symmetry_se               0.084117
fractal_dimension_se      0.027749
radius_worst              0.602277
texture_worst             0.388060
perimeter_worst           0.575178
area_worst                0.413095
smoothness_worst          0.317837
compactness_worst         0.200163
concavity_worst           0.214617
concave points_worst      0.614777
symmetry_worst            0.194362
fractal_dimension_worst   0.071166
Name: 70, dtype: float64
```

In [59]:
```
exp.show_in_notebook(show_table=True)
```


Prediction probabilities
diagnosis
0.00
Other
1.00
NOT undefined
concave points_worst >...
0.16
radius_worst > 0.38
0.13
area_mean > 0.27
0.11
concave points_mean >...
0.07
perimeter_worst > 0.37
0.07
perimeter_mean > 0.42
0.07
Feature
Value
concave points_worst
0.61
radius_worst
0.60
area_mean
0.42
concave points_mean
0.40
perimeter_worst
0.58
perimeter_mean
0.55