

IIT Kanpur

Prutor.ai

A Project Report

On

Breast Cancer Detection

Using Machine Learning

Course – AIML 5 Batch July 2021

Submitted By:

Harsh Gupta

United College of Engineering and Management

Abstract

Breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, a newly-inverted nipple, or a red or scaly patch of skin. In those with distant spread of the disease, there may be bone pain, swollen lymph nodes, shortness of breath, or yellow skin. Breast cancer most commonly develops in cells from the lining of milk ducts and the lobules that supply these ducts with milk. Cancers developing from the ducts are known as ductal carcinomas, while those developing from lobules are known as lobular carcinomas. There are more than 18 other sub-types of breast cancer. Some, such as ductal carcinoma in situ, develop from pre-invasive lesions. The diagnosis of breast cancer is confirmed by taking a biopsy of the concerning tissue. Once the diagnosis is made, further tests are done to determine if the cancer has spread beyond the breast and which treatments are most likely to be effective.

The project is based on machine learning performed on jupyter notebook

✓ First we Import our libraries

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

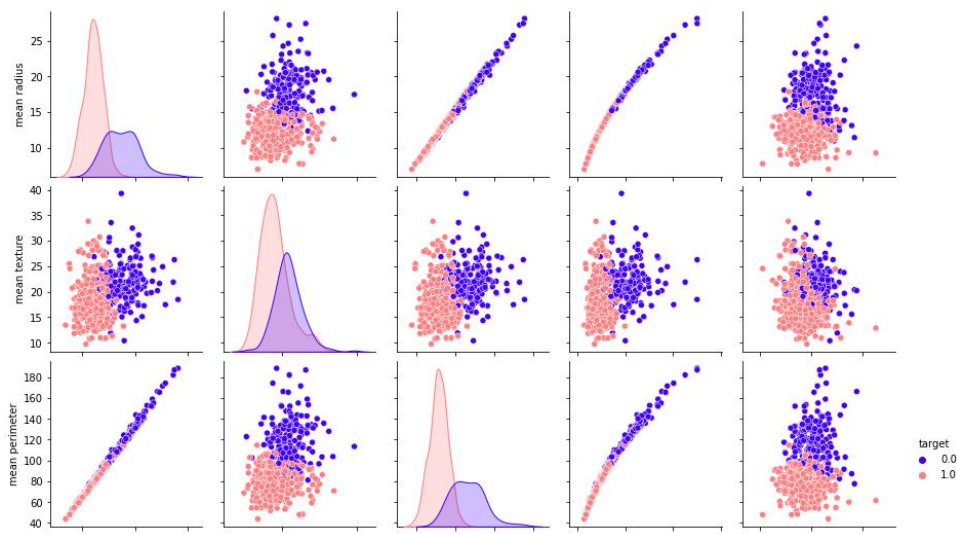
import seaborn as sns

from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

%matplotlib inline

from sklearn.datasets import load_breast_cancer
Breast_cancer = load_breast_cancer()
```

- ✓ The we imported our csv file from sklearn.dateset
- ✓ Then we load the data set
- ✓ Then we created different graphs to analys the data



- ✓ Then we split the data
 - ✓ Then feature scaling is done
 - ✓ Then we used two Algorithm to analys the data set
1. Svm

svm

```
In [16]: from sklearn.svm import SVC  
svc_model = SVC()
```

```
In [17]: svc_model.fit(X_train_scaled, y_train)
```

```
Out[17]: SVC()
```

```
In [18]: Prediction = svc_model.predict(X_test_scaled)  
print(accuracy_score(y_train, svc_model.predict(X_train)))
```

```
0.36043956043956044
```

```
In [19]: from sklearn.metrics import classification_report, confusion_matrix
```

```
In [20]: cm = np.array(confusion_matrix(y_test, Prediction, labels=[1,0]))  
confusion = pd.DataFrame(cm, index=['is_cancer', 'is_healthy'],  
                          columns=['predicted_cancer', 'predicted_healthy'])  
confusion
```

```
Out[20]:
```

	predicted_cancer	predicted_healthy
is_cancer	66	0
is_healthy	1	47

```
In [28]: print(classification_report(y_test, Prediction))
```

	precision	recall	f1-score	support
0.0	0.89	0.93	0.91	45
1.0	0.96	0.93	0.94	69

2. Knn

KNN

```
In [25]: knn = KNeighborsClassifier()  
knn.fit(X_train, y_train)
```

```
Out[25]: KNeighborsClassifier()
```

```
In [26]: y_test = knn.predict(X_test)  
print(accuracy_score(y_train, knn.predict(X_train)))  
  
0.9516483516483516
```

```
In [27]: knn_acc = accuracy_score(y_test, knn.predict(X_test))  
print(classification_report(y_test, knn.predict(X_test)))  
print('The score is', knn_acc*100)
```

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	45
1.0	1.00	1.00	1.00	69
accuracy			1.00	114
macro avg	1.00	1.00	1.00	114
weighted avg	1.00	1.00	1.00	114

The score is 100.0

3. Random Forest

Random_forest

```
In [25]: rf = RandomForestClassifier(n_estimators=20)  
  
rf.fit(X_train,y_train)  
#dt.fit(xtrain,ytrain)  
  
rf_test_score = rf.score(X_test,y_test)  
rf_train_score = rf.score(X_train,y_train)  
  
print('rf train score:',rf_train_score)  
print('rf test score:',rf_test_score)
```

```
rf train score: 1.0  
rf test score: 0.9473684210526315
```

Confusion Matrix

Confusion matrix

```
In [23]: Prediction = pickle_model.predict(X_test_scaled)
cm = confusion_matrix(y_test, Prediction)
```

```
In [24]: cm = np.array(confusion_matrix(y_test, Prediction, labels=[1,0]))
confusion = pd.DataFrame(cm, index=['is_cancer', 'is_healthy'],
                          columns=['predicted_cancer', 'predicted_healthy'])
confusion
```

Out[24]:

	predicted_cancer	predicted_healthy
is_cancer	66	0
is_healthy	1	47

****END OF PROJECT****