Report-Breast Cancer Prediction

Abraham Tony

B.Tech Semester Six Computer Science-Mar Baselios College of Engineering & Technology Trivandrum

INTRODUCTION

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society.

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of BC and classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modelling.

PURPOSE

Classification and data mining methods are an effective way to classify data. Especially in the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

So it's amazing to be able to possibly help save lives just by using data, python, and machine learning!

RISK FACTORS FOR BREAST CANCER

The following are some of the known risk factors for breast cancer. However, most cases of breast cancer cannot be linked to a specific cause. Talk to your doctor about your specific risk.

Age. The chance of getting breast cancer increases as women age. Nearly 80 percent of breast cancers are found in women over the age of 50.

Personal history of breast cancer. A woman who has had breast cancer in one breast is at an increased risk of developing cancer in her other breast.

Family history of breast cancer. A woman has a higher risk of breast cancer if her mother, sister or daughter had breast cancer, especially at a young age (before 40). Having other relatives with breast cancer may also raise the risk.

Genetic factors. Women with certain genetic mutations, including changes to the BRCA1 and BRCA2 genes, are at higher risk of developing breast cancer during their lifetime. Other gene changes may raise breast cancer risk as well.

Childbearing and menstrual history. The older a woman is when she has her first child, the greater her risk of breast cancer. Also at higher risk are:

- Women who menstruate for the first time at an early age (before 12)
- Women who go through menopause late (after age 55)
- Women who've never had children

PROPOSED SOLUTION

Using this model, breast cancer among women can be found by classifying whether it's malignant or benign i.e mild or at high risk. To achieve this I have used machine learning classification methods to fit a function that can predict the discrete class of new input.

This prediction uses the above features:

- ID number
- Diagnosis (M = malignant, B = benign)
- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- Perimeter
- Area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area — 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- Symmetry
- fractal dimension ("coastline approximation" — 1)

EXPERIMENTAL INVESTIGATION

Data was collected from
http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29

and then pre-processed so that it is understood by the Machine Learning Algorithms properly.

```
In [3]: df.head()
```

Out[3]:

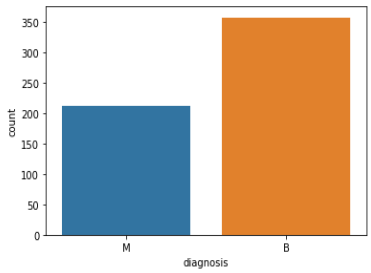
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

5 rows × 32 columns

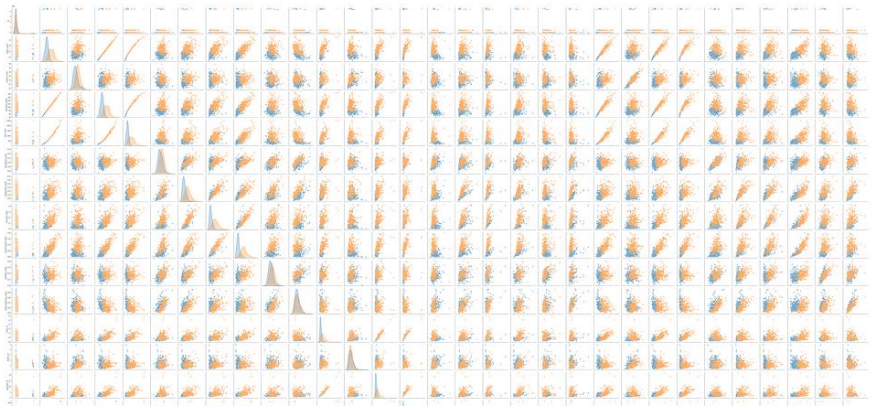
Exploratory Data analysis was done, in order to visualise the dataset, and to check the correlation of different parameters on the ‘Breast Cancer’

```
In [13]: sns.countplot(df['diagnosis'],label="Count")
```

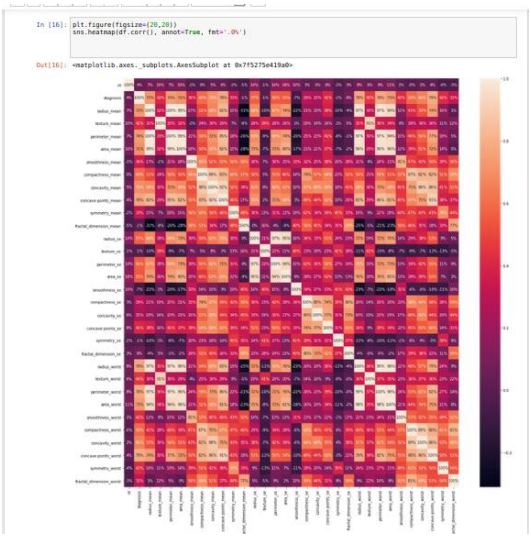
```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f52905f0cd0>
```



```
In [15]: sns.pairplot(df, hue="diagnosis")
Out[15]: <seaborn.axisgrid.PairGrid at 0x7f528e51fb50>
```

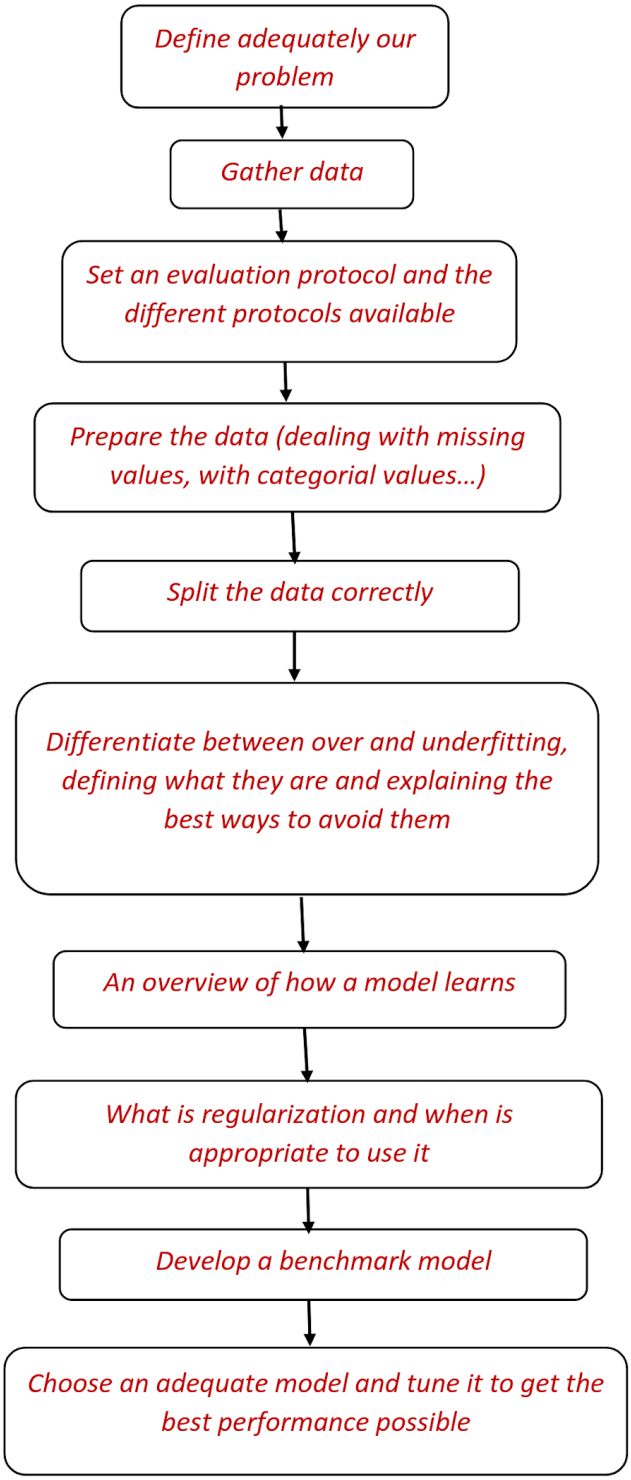


```
In [16]: plt.figure(figsize=(20,20))
         sns.heatmap(df.corr(), annot=True, fmt='.0%')
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7f5275e419a0>
```



After done with data visualisation, then Machine Learning algorithms were applied and then accuracy is checked.

FLOWCHART


Define adequately our problem
Gather data
Set an evaluation protocol and the different protocols available
Prepare the data (dealing with missing values, with categorial values…)
Split the data correctly
Differentiate between over and underfitting, defining what they are and explaining the best ways to avoid them
An overview of how a model learns
What is regularization and when is appropriate to use it
Develop a benchmark model
Choose an adequate model and tune it to get the best performance possible

RESULT

```
In [36]: from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score

         print('Model')
         print( classification_report(Y_test, model.predict(X_test)) )
         print( accuracy_score(Y_test, model.predict(X_test)))
```

```
Model
              precision    recall  f1-score   support

           0       0.98      0.97      0.97        90
           1       0.94      0.96      0.95        53

    accuracy                           0.97       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.97      0.97      0.97       143

0.965034965034965
```

So finally we have built our classification model and we can see that the Random Forest Classification algorithm gives the best results for our dataset with a great accuracy score of 96.5%.

ADVANTAGES & DISADVANTAGES

ADVANTAGES

1. Breast cancer can be predicted depending on certain parameters with great accuracy.
2. Knowing this information can help you make a more informed choice regarding whether you have this disease and go for further treatment.

DISADVANTAGES

1. Though, the accuracy of the model is very high, still there is some chance that it does not give the exact prediction. It depends on input data.
2. It may create some tension and mental stress when people got to know their wrong results.

CONCLUSION

This is a basic Machine Learning model to predict Breast Cancer and classifying as benign or malignant. Random Forest Classification algorithm gives the best results for our dataset. Well it's not always applicable to every dataset. To choose our model we always need to analyze our dataset and then apply our machine learning model.

BIBLIOGRAPHY

- http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29
- https://www.kaggle.com/

APPENDIX

Source Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df=pd.read_csv('data.csv')
df.head()
df.columns
df.shape
df.isnull().sum()
cor=df.corr()
cor.shape
df.describe()
df['diagnosis'].value_counts()
sns.countplot(df['diagnosis'],label="Count")
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
df.iloc[:,1]= labelencoder_Y.fit_transform(df.iloc[:,1].values)
print(labelencoder_Y.fit_transform(df.iloc[:,1].values))
sns.pairplot(df, hue="diagnosis")
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(), annot=True, fmt='.0%')
X = df.iloc[:, 2:31].values
Y = df.iloc[:, 1].values
X
Y
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
def models(X_train,Y_train):
        from sklearn.ensemble import RandomForestClassifier
        forest = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
        forest.fit(X_train, Y_train)
        print('Random Forest Classifier Training Accuracy:', forest.score(X_train, Y_train))
        return forest
model = models(X_train,Y_train)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, model.predict(X_test))
TN = cm[0][0]
TP = cm[1][1]
```

```
FN = cm[1][0]
FP = cm[0][1]

print(cm)
print('Model Testing Accuracy = "{}!"'.format((TP + TN)/(TP + TN + FN + FP)))
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

print('Model')
print( classification_report(Y_test, model.predict(X_test)) )
print( accuracy_score(Y_test, model.predict(X_test)))
pred = model[6].predict(X_test)
print(pred)

#Print a space
print()

#Print the actual values
print(Y_test)
```