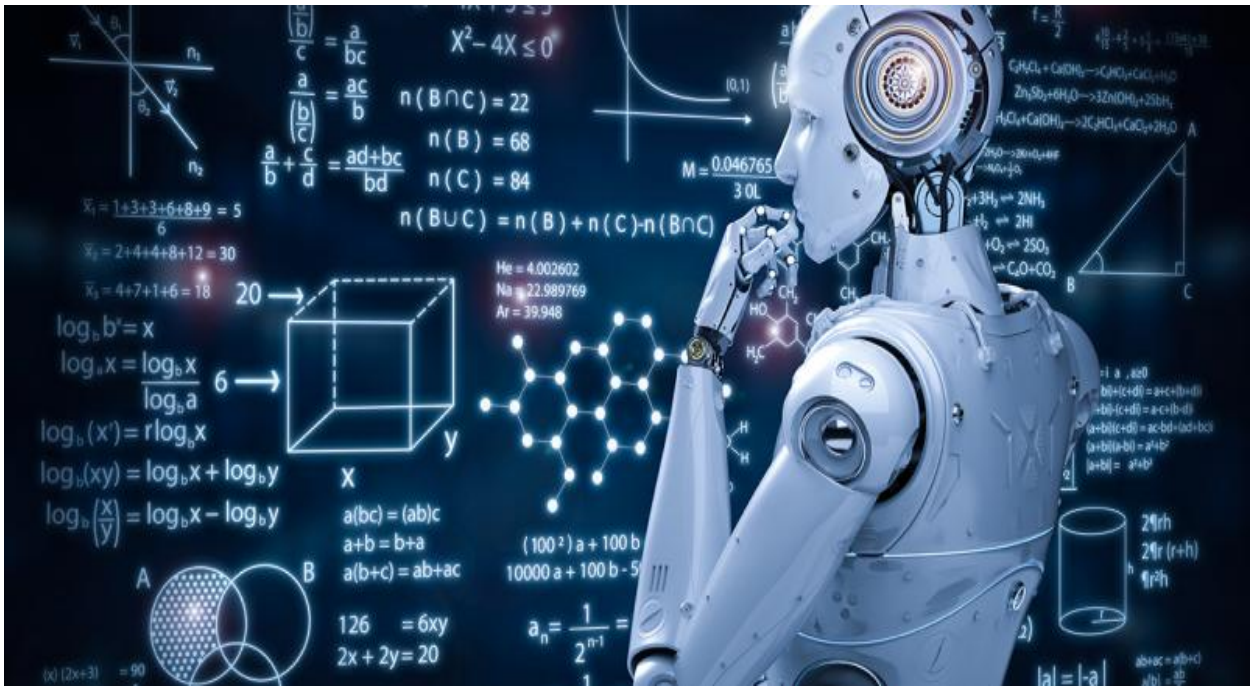
COMPUTER SCIENCE AND ENGINEERING

BREAST CANCER PREDICTION USING ML

IMPACTEERS CLUB



GROUP:

198W1A0503 - ATHULURI BHHARATHEE

198W1A0504 - BADUGU BHAVANA

198W1A0528 - KOMMINENI NISHITHA

198W1A0564 - YALAMANCHILI MANASA

Abstract

Breast cancer is one of the most common and leading causes of cancer among women. Prior identification is the best way to manage breast cancer results. Computer-aided detection or diagnosis (CAD) systems play a major role in the prior identification of breast cancer and can be used for the reduction of the death rate among women. The goal is to increase the proportion of breast cancers identified at an early stage, allowing for more effective treatment to be used and reducing the risks of death from breast cancer. Since early detection of cancer is key to effective treatment of breast cancer we use various machine learning algorithms to predict if a tumor is benign or malignant, based on the features provided by the data.

In this project, we use the concept of 'LOGISTIC REGRESSION' to detect 'Breast Cancer' based on various parameters given in the dataset.

Problem Statement

Breast Cancer cases have been increasing steadily over the past ten years in our country. A 2018 report of Breast Cancer statistics recorded that cancer survival becomes more difficult in higher stages of its growth, and more than 50% of Indian women suffer from stages 3 and 4 of breast cancer. There have been some effective attempts to detect the cancer cells at an early stage so as to provide a hope of survival.

We intend to use Machine Learning techniques to assist in the detection of breast cancer cells in this project.

Methodology

This project predicts whether the person has breast cancer using the diagnosis that is taken from the biopsy and magnetic resonance imaging(MRI) tests. In this project, we use the package "sklearn" to divide the information in the dataset into two parts for training and testing. Then we built a Logistic Regression model. We trained the model using 80% of the data from the dataset and tested the predictions made by it using the latter 20%.

Hardware and Software used

Hardware:

Simulated on Google COLAB.

Software:

Programming language: Python

Machine Learning packages used:

1. Pandas
2. Seaborn
3. Matplotlib
4. Sklearn

Test dataset

The link for the dataset we used is:

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

In this dataset 32 attributes are considered to train the model and then predict the diagnosis.

Attribute Information:

1. ID number
2. Diagnosis (M = malignant, B = benign)
3. From row-3 to row-32:

   Ten real-valued features are computed for each cell nucleus:

   a. radius (mean of distances from the center to points on the perimeter)
   b. texture (standard deviation of gray-scale values)
   c. perimeter

d. area
e. smoothness (local variation in radius lengths)
f. compactness (perimeter^2 / area - 1.0)
g. concavity (severity of concave portions of the contour)
h. concave points (number of concave portions of the contour)
i. symmetry
j. fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recorded with four significant digits.

Missing attribute values: none

Class distribution: 357 benign, 212 malignant

Source Code

IMPORTING REQUIRED LIBRARIES

```
#import libraries

import pandas as pd

import seaborn as sns

import matplotlib.mlab as mlab

import matplotlib.pyplot as plt

%matplotlib inline
```

DOWNLOADING DATASET FROM KAGGLE

```
#set kaggle API credentials

import os

os.environ['KAGGLE_USERNAME'] = "bhharatheeathuluri"

os.environ['KAGGLE_KEY'] = "b87b0317ed102f25ac8c9673ab58a488"
```

```
#download dataset from kaggle
! kaggle datasets download -d uciml/breast-cancer-wisconsin-data


#unzip the downloaded zip file
! unzip /content/breast-cancer-wisconsin-data.zip
```

LOAD AND EXPLORE DATA

```
#load data on dataframe
df = pd.read_csv('/content/data.csv')


#display dataframe
df.head()


#count of rows and columns
df.shape


#count number of null(empty) values
df.isna().sum()


# Drop the column with null values
df.dropna(axis=1,inplace=True)


# count of rows and columns
df.shape
```

```
#Get count of number of M or B cells in diagnosis

df['diagnosis'].value_counts()
```

LABEL ENCODING

```
#Get Datatypes of each column in our dataset

df.dtypes


#Encode the diagnosis values

from sklearn.preprocessing import LabelEncoder

labelencoder = LabelEncoder()

df.iloc[:,1] = labelencoder.fit_transform(df.iloc[:,1].values)
```

DATA VISUALISATION

```
#graph showing the values of 'diagnosis' column

sns.countplot(x='diagnosis',data=df)


#graphs showing the values various parameters with respect to diagnosis values

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


#graphs showing all the values of the parameters taken

def draw_histogram(df, features, rows, cols):

  fig = plt.figure(figsize=(20,20))

  for i, feature in enumerate(features):

    ax = fig.add_subplot(rows,cols,i+1)
```

```
    df[feature].hist(bins=20,ax=ax,facecolor='pink')
    ax.set_title(feature + " Distribution",color='midnightblue')
  fig.tight_layout()
  plt.show()
  print('\n\n')
draw_histogram(df,df.columns,8,4)
```

SPLIT DATASET AND FEATURE SCALING

```
#Splitting the dataset into independent and dependent datasets
X = df.iloc[:,2:].values      #Independent Set
Y = df.iloc[:,1].values       #Dependent Set

#Splitting datasets into training(80%) and testing(20%)
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)

#Scaling the data(feature scaling)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)

#print data
X_train
```

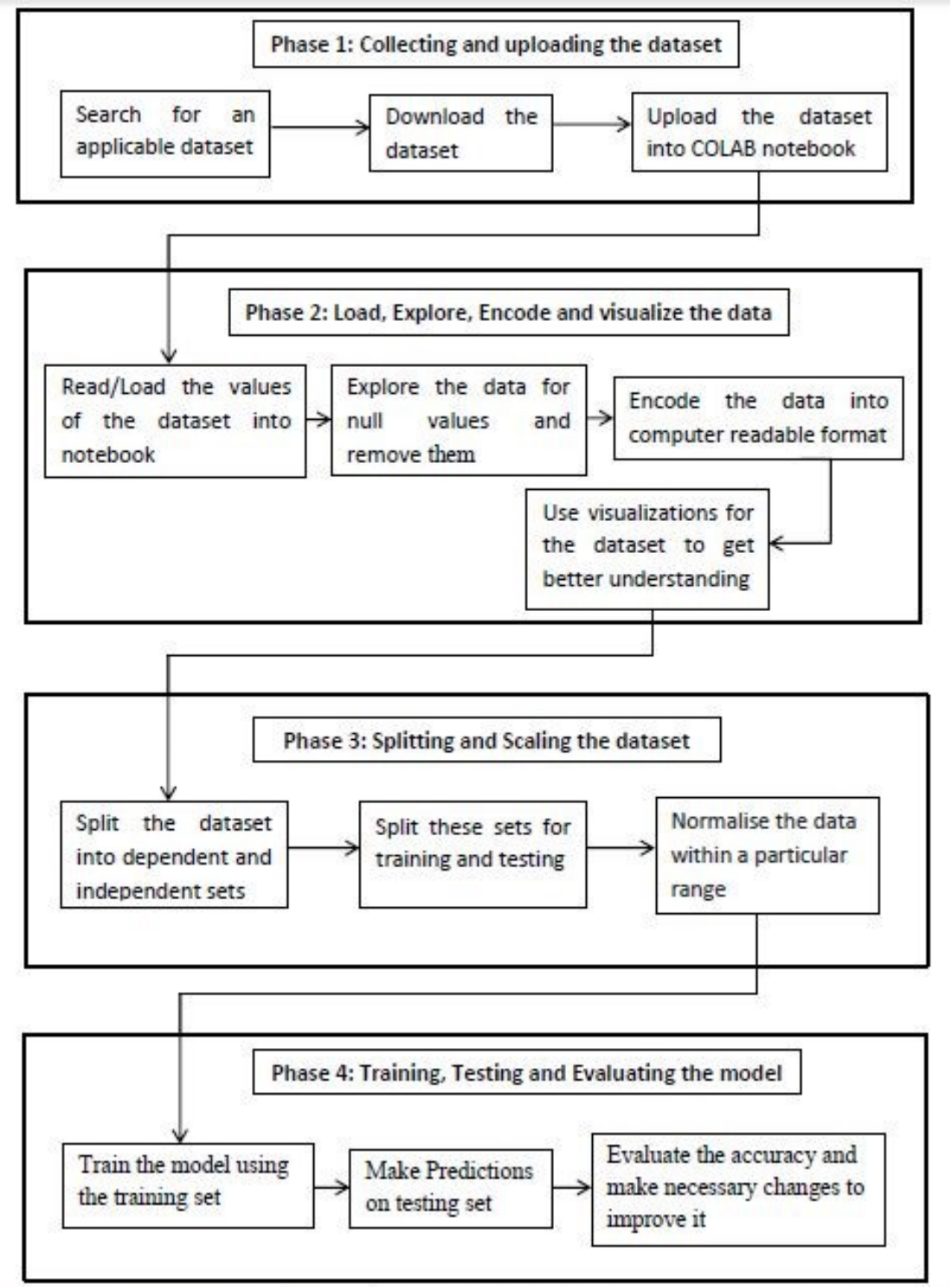
BUILD A LOGISTIC REGRESSION MODEL

```
#build a logistic regression classifier

from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression()

classifier.fit(X_train,Y_train)


#make use of trained model to make predictions on test data

predictions = classifier.predict(X_test)
```

PERFORMANCE EVALUATION

```
#plot confusion matrix

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(Y_test,predictions)

print(cm)

sns.heatmap(cm,annot=True)


#get accuracy score for model

from sklearn.metrics import accuracy_score

print('Accuracy of this prediction model
is{0:.2f}%'.format(accuracy_score(Y_test,predictions)*100))


#printing the actual values in the dataset

print(Y_test)

#printing the values predicted by the model

print(predictions)
```

Process Diagram



Performance

Here is the screenshot of the accuracy of the model (It varies each time you run it).

```
[58] #get accuracy score for model
     from sklearn.metrics import accuracy_score
     print('Accuracy of this prediction model is {0:.2f}%'.format(accuracy_score(Y_test,predictions)*100))

     Accuracy of this prediction model is 97.37%
```

Result

Here is the screenshot of the output in which the accuracy of the model, the actual values in the dataset, and the predicted values are printed.

```
[58] #get accuracy score for model
     from sklearn.metrics import accuracy_score
     print('Accuracy of this prediction model is {0:.2f}%'.format(accuracy_score(Y_test,predictions)*100))

     Accuracy of this prediction model is 97.37%

     print(Y_test)

     [0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0
      0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 1
      0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1
      1 1 0]

[60] print(predictions)

     [0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1
      0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0 0 1 1
      1 1 0]
```

Github Link of our project:

https://github.com/Bhharathee-Athuluri/Breast-Cancer-Prediction-using-ML

The files uploaded in the Github repository are updated files with the best results obtained. The accuracy of the model changes every time we run the notebook.