International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

Detecting Breast Cancer with Logistic Regression Model

[1]Sunami Dasgupta, [2]Tusita Sarkar, [3]Rajveer Pandya, [4]Sayani Hazra Pal

[1]*Student, D.A.V. Public School, Kolkata, India.*
[2]*Student,Heritage Institute of Technology, Kolkata, India.*
[3]*Student, Jaypee Institute of Information Technology, Noida, India.*
[4]*Instructor, D.A.V. Public School, Kolkata, India.*

Submitted: 15-04-2021 Revised: 28-04-2021 Accepted: 30-04-2021

ABSTRACT: True Artificial Intelligence is decades away, but one can be achieved through deep learning and machine learning, where the machine learns to do function without being explicitly programmed. They are trained and tested on different datasets. In this research paper our aim will be to construct some machine learning models whose algorithms use Euclidian distance between two data points in their computation to predict malignant cells in a breast cancer patient. The models will be first rained and then tested on the data set which will be split in two one for training another for testing.

KEYWORDS: Breast Cancer, Prediction through Machine Learning, Logistic Regression, Python, Data Science.

I. INTRODUCTION

What is Machine Learning and why are we using it?

Machine Learning (ML) is a field of Artificial Intelligence that utilizes measurable strategies to enable computer system frameworks to 'learn' (e.g., dynamically improve execution on assignment) from information or data, without being expressly customized.

Now, if we talk about well-defined definition of this topic---there isn't. But there two sentences that we considered to be the "definition" of Machine Learning (ML).

1. Arthur Samuel described it as: "the field of study that gives computers the ability to learn without being explicitly programmed." This is an older, informal definition.
2. Tom Mitchell provides a more modern definition: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine Learning Can be classified into
Two groups:

1. **Supervised Learning**: In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.
2. **Unsupervised Learning:** Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.

Breast Cancer Overview

Malignancy happens when changes considered transformations to occur in qualities that direct cell development. The changes let the cells separate and duplicate in an uncontrolled manner.

Breast malignancy is disease that creates in breast cells. Normally, the disease frames in either the lobules or the pipes of the breast. Lobules are the organs that produce milk, and pipes are the pathways that carry the milk from the organs to the areola. Malignancy can likewise happen in the fatty tissue or the fibrous connective tissue inside your breast.

The uncontrolled malignancy cells regularly attack other sound breast tissue and can go to the lymphnodes under the arms. The lymph nodes are an essential pathway that helps the disease cells move to different pieces of the body.

Breast malignancy is the subsequent driving reason for disease demise in ladies, second just to cellular breakdown in the lungs. The main

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

danger factor for breast disease is basically being a lady. Even though breast malignant growth happens in men, the illness is multiple times more normal in ladies. Men can likewise get breast malignancy. In 2017, the American Cancer Society gauges 2,470 new instances of intrusive breast malignancy will be analysed in men in the U.S. A lady has around a one of every eight possibility of being determined to have breast malignant growth during her life, as per the National Cancer Institute. Most ladies (around eight out of 10) who get breast malignancy don't have a family background of the infection. But ladies who have close blood family members with breast malignancy have a higher danger.

Types of Breast Cancer:

1. Angiosarcoma:Angiosarcoma is a rare type of cancer that forms in the lining of the blood vessels and lymph vessels. Your lymph vessels, which are part of your immune system, collect bacteria, viruses and waste products from your body and dispose of them. Angiosarcoma can occur anywhere in your body, but it most often occurs in the skin on your head and neck. Rarely, angiosarcoma may form in the skin on other parts of your body, such as the breast. Or it may form in deeper tissue, such as the liver and the heart. Angiosarcoma can occur in areas previously treated with radiation therapy.
2. Ductile Carcinoma in Situ (DCIS):Ductal carcinoma in situ (DCIS) is the presence of abnormal cells inside a milk duct in the breast. DCIS is considered the earliest form of breast cancer. DCIS is non-invasive, meaning it hasn't spread out of the milk duct and has a low risk of becoming invasive. DCIS is usually found during a mammogram done as part of breast cancer screening or to investigate a breast lump.
3. Inflammatory breast cancer: Inflammatory breast cancer is a rare type of breast cancer that develops rapidly, making the affected breast red, swollen and tender. Inflammatory breast cancer occurs when cancer cells block the lymphatic vessels in skin covering the breast, causing the characteristic red, swollen appearance of the breast.
4. Invasive lobular carcinoma: Invasive lobular carcinoma is a type of breast cancer that begins in the milk-producing glands (lobules) of the breast. Invasive cancer means the cancer cells have broken out of the lobule where they began and have the potential to spread to the lymph nodes and other areas of the body.
5. Lobular carcinoma in situ (LCIS): Lobular carcinoma in situ (LCIS) is an uncommon condition in which abnormal cells form in the milk glands (lobules) in the breast. LCIS isn't cancer. But being diagnosed with LCIS indicates that you have an increased risk of developing breast cancer.
6. Male breast cancer: Male breast cancer is a rare cancer that forms in the breast tissue of men. Though breast cancer is most commonly thought of as a disease that affects women, breast cancer does occur in men. Male breast cancer is most common in older men, though it can occur at any age.
7. Paget's disease of the breast: Paget's (PAJ-its) disease of the breast is a rare form of breast cancer. Paget's disease of the breast starts on the nipple and extends to the dark circle of skin (areola) around the nipple. Paget's disease of the breast isn't related to Paget's disease of the bone, a metabolic bone disease.
8. Recurrent breast Cancer: Recurrent breast cancer is breast cancer that comes back after initial treatment. Although the initial treatment is aimed at eliminating all cancer cells, a few may have evaded treatment and survived. These undetected cancer cells multiply, becoming recurrent breast cancer.

Symptoms:

- A breast lump or thickening that feels different from the surrounding tissue
- Change in the size, shape or appearance of a breast
- Changes to the skin over the breast, such as dimpling
- A newly inverted nipple
- Peeling, scaling, crusting or flaking of the pigmented area of skin surrounding the nipple (areola) or breast skin
- Redness or pitting of the skin over your breast, like the skin of an orange.

Recommended Screening Guidelines:

Mammography: The main evaluating test for bosom disease is the mammogram. A mammogram is a X-beam of the bosom. It can distinguish bosom malignant growth as long as two years before the tumour can be felt by you or your PCP.

Women at high risk should have yearly mammograms along with an MRI starting at age 30.

II. PREREQUISITES

1.Machine Learning Models
2.Dataset to train the models.
3.Dataset to test the models.

The models that we will be using in this experiment are:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

a. LogisticRegression
b. DecisionTreeClassifier
c. RandomForestClassifier

III. METHODOLOGY

The whole experiment will be done step by step, the steps are:

a. Data Preparation
b. Data Exploration
c. Categorical Data
d. Splitting Dataset
e. Feature Scaling
f. Model Selection
g. Accuracy Check

IV. EXPERIMENTATION

Data Preparation:

We will use the Breast Cancer Wisconsin (Diagnostic) Data Set.

The dataset used in this experiment is publicly available.

Attribute Information:

1. ID number 2) Diagnosis (M = malignant, B = benign) 3–32)

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

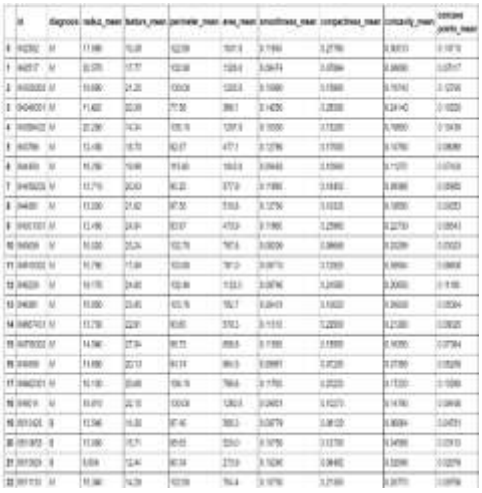
1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness (perimeter² / area — 1.0)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension ("coastline approximation" — 1)

Data Exploration:

We will be using Google colab-Jupiter notebook to work on our dataset. We will be first importing the necessary libraries and import our database:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
uploaded = files.upload()
df = pd.read_csv('data.csv')
df.head(30)
```

The database loaded



We can find the dimension of the database using Shape attribute:

```
print("Cancer data set dimensions : {}".format(dataset.shape))

Cancer data set dimensions : (569, 32)
```

We can see that the database contains 569 rows and 32 columns. 'diagnosis' is the column which we are going to predict, which says if the cancer is malignant (M) or benign (B).

Code to find how many people are malignant:

```
In [12]: #Get a count of the number Malignant (M) or Benign (B) cells(@sunami09)
         df['diagnosis'].value_counts()

Out[12]: B    357
         M    212
         Name: diagnosis, dtype: int64
```

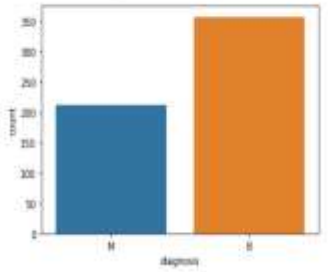
We can see from the output that 212 people are malignant that is they have cancer cells while 357 people don't.

Visualization of Data is an impressive aspect of data science. It helps to understand data and also to explain the data to another person. We will use seaborn and Matplotlib to do this very task.

```
In [13]: #Visualise the count(@sunami09)
         sns.countplot(df['diagnosis'], label='count')
```

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f755cded310>
```



Categorical Data:
This are variables that contain values (label) instead of numeric values. The number possibility is fixed to a particular dataset.
We will be using Label Encoder to label the data.

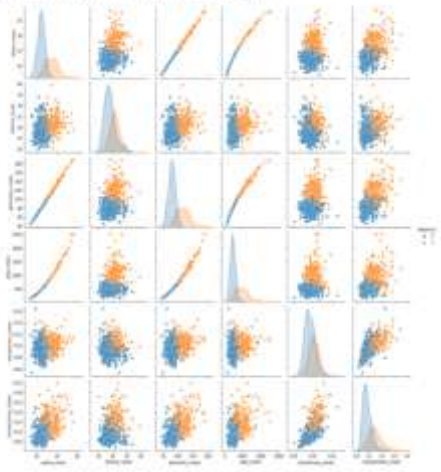
```
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
df.iloc[:,1] = labelencoder_Y.fit_transform(df.iloc[:,1].values)

df.iloc[:,1]
```

```
0      1
1      1
2      1
3      1
4      1
      ..
564    1
565    1
566    1
567    1
568    0
Name: diagnosis, Length: 569, dtype: int64
```

Visualizing the data in pair plot manner:

```
In [16]: #Create a pair plot
         sns.pairplot(df.iloc[:,1:6],hue = 'diagnosis')
Out[16]: <seaborn.axisgrid.PairGrid at 0x7f755bf0150>
```



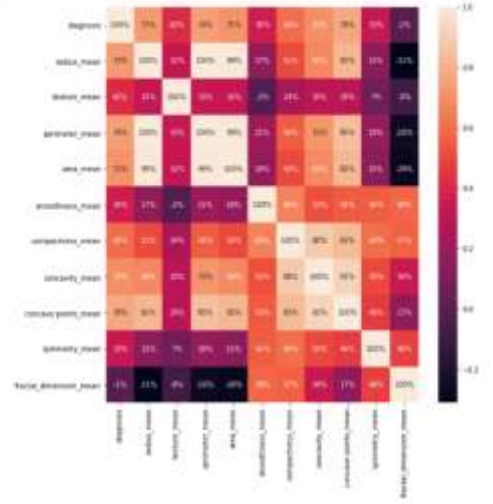
Creating the correlation of the columns using iloc attribute:

```
In [19]: #Get the correlations of the columns
         df.iloc[:,1:12].corr()
Out[19]:
```

	diagnosis	radius_mean	texture_mean	perimeter_mean
diagnosis	1.000000	0.730029	0.415185	0.742636
radius_mean	0.730029	1.000000	0.323782	0.997855
texture_mean	0.415185	0.323782	1.000000	0.329533
perimeter_mean	0.742636	0.997855	0.329533	1.000000
area_mean	0.708984	0.987357	0.321086	0.986507
smoothness_mean	0.358560	0.170581	-0.023389	0.207278
compactness_mean	0.596534	0.506124	0.236702	0.556936
concavity_mean	0.696360	0.676764	0.302418	0.716136
concave points_mean	0.776614	0.822529	0.293464	0.850977
symmetry_mean	0.330499	0.147741	0.071401	0.183027
fractal_dimension_mean	-0.012838	-0.311631	-0.076437	-0.261477

Now, we will visualize the correlation of the data we just got on a heat map:

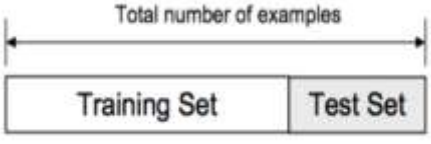
```
In [20]: #Visualize the correlations
         plt.figure(figsize=(10,10))
         sns.heatmap(df.iloc[:,1:12].corr(), annot = True, fmt = '.0%')
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f755a4a8ad0>
```



Splitting the Dataset:

We are using only one dataset so, we will be dividing or splitting the dataset into two parts not equal be will give the training of Models more data compared to testing them. That is

1. Training Data – 75%
2. Testing Data -- 25%



Splitting the data in X and Y set:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
In [21]: #split the data set into independent (X) and dependent (Y) data sets(@su
         manik09)
         X = df.iloc[:,2:31].values
         Y = df.iloc[:,1].values

In [22]: #split the dataset into 75% training and 25% testing (@sunamik09)
         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)
         print(X,Y)
```

Now the data set we created looks like this:

```
[[ 17.99    10.38   122.8    ...   0.7119   0.2654   0.4601]
 [ 20.57    17.77   132.9    ...   0.2416   0.186    0.275 ]
 [ 19.69    21.25   130.     ...   0.4504   0.243    0.3613]
 ...
 [ 16.6     28.08   108.3    ...   0.3403   0.1418   0.2218]
 [ 20.6     29.33   140.1    ...   0.9387   0.265    0.4087]
 [  7.76    24.54    47.92   ...   0.       0.       0.2871]] [1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1
1
 0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0
0
 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 1
0
 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 1 1 1 1 0
1
 0 1 0 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 0 0 1 1 0
0
 0 1 0 0 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1
1
 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0
0
 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 1 0
0
 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1
1
 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
0
 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0
0
 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1
0
 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 0 1 1 0 1 0 1 0 1
1
 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0]
```

'1' stands for TRUE which means the patient have malignant cells. '0' stands for FALSE which means the patient have benign cells.

Feature Scaling:

The values in the dataset have highly varying magnitudes, units and range. But since, most of the machine learning models and algorithms use Euclidian distance between two data points in their computation. We need to bring all features to the same level of magnitudes. And this can be obtained by Scaling like 0-100 or 0-1.

In this experiment we will use StandardScaler method:

```
In [23]: #Scale the data (Feature Scaling) (@sunamik09)
         from sklearn.preprocessing import StandardScaler
         sc = StandardScaler()
         X_train = sc.fit_transform(X_train)
         X_test = sc.fit_transform(X_test)
         print(X_train)

[[-0.65079907 -0.43057322 -0.68024847 ... -0.69592933 -0.36433881
   0.32349851]
 [-0.82835341  0.15226547 -0.82773762 ... -1.29277423 -1.45036679
   0.62563098]
 [ 1.68277234  2.18977235  1.68089756 ...  0.26255563  0.72584581
  -0.51329768]
 ...
 [-1.33114223 -0.22172269 -1.3242844  ... -0.78274313 -0.98806491
  -0.69995543]
 [-1.25110186 -0.24698763 -1.28700242 ... -1.36015587 -1.75887319
  -1.56206114]
 [-0.74662285  1.14066273 -0.72203706 ...  0.47201917 -0.2860679
  -1.34094654]]
```

Model Selection:

There are many machine learning models that can be used to detect the breast cancer they are:

1. Logistic Regression
2. Nearest Neighbor
3. Support Vector Machines
4. Kernel SVM
5. Naïve Bayes
6. Decision Tree Algorithm
7. Random Forest Classification

But we will only use three of them and then check their accuracy during the training period and then test them on real data to predict the cancer cells.

The models that we will be using are:

a. LogisticRegression
b. DecisionTreeClassifier
c. RandomForestClassifier

Accuracy Check:

First, we will be creating a function to execute the model and then return the prediction as an array of Boolean values.

Building the function:

```
In [24]: # Create a function for the models(@sunamik09)
         def models(X_train,Y_train):

           #Logistic Regression(@sunamik09)
           from sklearn.linear_model import LogisticRegression
           log = LogisticRegression(random_state = 0)
           log.fit(X_train,Y_train)

           #Decision Tree(@sunamik09)
           from sklearn.tree import DecisionTreeClassifier
           tree = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
           tree.fit(X_train, Y_train)

           #Random Forest Classifier(@sunamik09)
           from sklearn.ensemble import RandomForestClassifier
           forest = RandomForestClassifier(n_estimators = 10, criterion = 'entrop
         y', random_state = 0)
           forest.fit(X_train,Y_train)

           #print the model's accuracy on the training data(@sunamik09)
           print('[0]Logistic Regression Training Accuracy:', log.score(X_train,
         Y_train))
           print('[1]Decision Tree Classifier Training Accuracy:', tree.score(X_t
         rain, Y_train))
           print('[2]Random Forest Classifier Training Accuracy:',forest.score(X_
         train, Y_train))

           return log, tree, forest
```

Checking the accuracy:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
model = models(X_train, Y_train)

[0]Logistic Regression Training Accuracy: 0.9900103286384976
[1]Decision Tree Classifier Training Accuracy: 1.0
[2]Random Forest Classifier Training Accuracy: 0.9953051643192489
```

V. RESULT

As we have successfully trained our model on the 75% of our data from the dataset now it's time for testing it with rest of the 25% of the data.

We will be testing with all the three models and the models are as follows:

1. Model 0 = LogisticRegression
2. Model 1 = DecisionTreeClassifier
3. Model 2 = RandomForestClassifier

Testing:

```
In [26]: # Test our model on the testing data(@sunam100)
         # Test model accuracy on confusion matrix(@sunam100)
         from sklearn.metrics import confusion_matrix

         for i in range( len(model)):
           print('Model', i)
           cm = confusion_matrix(Y_test, model[i].predict(X_test))

           TP = cm[0][0]
           TN = cm[1][1]
           FN = cm[1][0]
           FP = cm[0][1]

           print(cm)
           print('Testing Accuracy = ',(TP + TN)/(TP + TN + FN + FP))
           print()
```

Accuracy:

```
Model 0
[[86  4]
 [ 3 50]]
Testing Accuracy =  0.951048951048951

Model 1
[[83  7]
 [ 2 51]]
Testing Accuracy =  0.9370629370629371

Model 2
[[87  3]
 [ 2 51]]
Testing Accuracy =  0.965034965034965
```

Another Way to get metrics of the model:

```
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score

for i in range( len(model)):
  print('Model', i)
  print( classification_report(Y_test, model[i].predict(X_test)))
  print( accuracy_score(Y_test, model[i].predict(X_test)))
  print()
```

The performance of the models:

```
Model 0
              precision    recall  f1-score   support

           0       0.97      0.96      0.96        90
           1       0.93      0.94      0.93        53

    accuracy                           0.95       143
   macro avg       0.95      0.95      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951

Model 1
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        90
           1       0.88      0.96      0.92        53

    accuracy                           0.94       143
   macro avg       0.93      0.94      0.93       143
weighted avg       0.94      0.94      0.94       143

0.9370629370629371

Model 2
              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
```

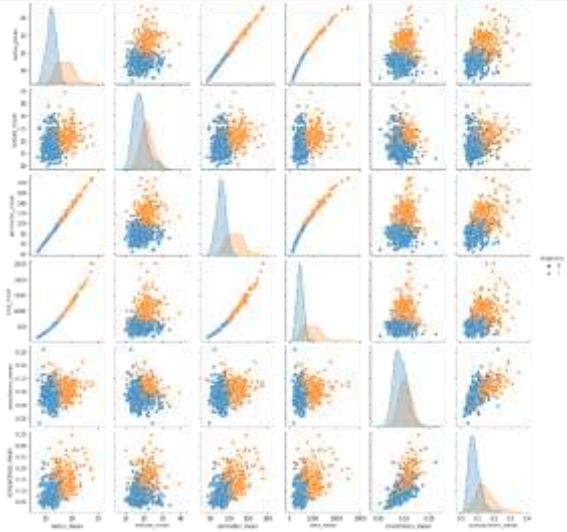
VI. SOURCE CODE

```
#Import Libraries
importnumpyasnp
importpandasaspd
importmatplotlib.pyplotasplt
importseabornassns
#Load the data
fromgoogle.colabimportfiles
uploaded=files.upload()
df=pd.read_csv('data.csv')
df.head(30)
#Count the no. of rows and column
df.shape
df.isna().sum()
df=df.dropna(axis=1)
df.shape
df['diagnosis'].value_counts()
#Visualise the count
```

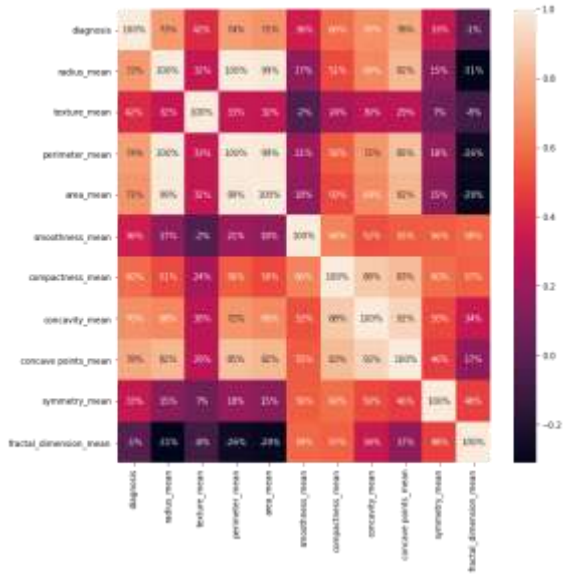
International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
sns.countplot(df['diagnosis'],label='count')
df.dtypes
#Encode the catagorical data
fromsklearn.preprocessingimportLabelEncoder
labelencoder_Y=LabelEncoder()
df.iloc[:,1]=labelencoder_Y.fit_transform(df.iloc[:,1].values)

df.iloc[:,1]
#Create a pair plot
sns.pairplot(df.iloc[:,1:8],hue='diagnosis')
```



```
#Visualize the Correlations
plt.figure(figsize=(10,10))
sns.heatmap(df.iloc[:,1:12].corr(),annot=True,fmt='.0%')
<matplotlib.axes._subplots.AxesSubplot at 0x7f755a4a8ad0>
```



```
#Split the Data set into independent (X) and dependent (Y) data sets
X=df.iloc[:,2:31].values
Y=df.iloc[:,1].values
fromsklearn.model_selectionimporttrain_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25,random_state=0)
print(X,Y)
```

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
[[ 17.99    10.38   122.8    ...   0.7119   0.2654   0.4601]
 [ 20.57    17.77   132.9    ...   0.2416   0.186    0.275 ]
 [ 19.69    21.25   130.     ...   0.4504   0.243    0.3613]
 ...
 [ 16.6     28.08   108.3    ...   0.3403   0.1418   0.2218]
 [ 20.6     29.33   140.1    ...   0.9387   0.265    0.4087]
[  7.76    24.54    47.92   ...   0.       0.       0.2871]]
#Scale The data (Feature Scaling)
fromsklearn.preprocessingimportStandardScaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.fit_transform(X_test)
print(X_train)
[[-0.65079907 -0.43057322 -0.68024847 ... -0.69592933 -0.36433881
   0.32349851]
 [-0.82835341  0.15226547 -0.82773762 ... -1.29277423 -1.45036679
   0.62563098]
 [ 1.68277234  2.18977235  1.60009756 ...  0.26255563  0.72504581
  -0.51329768]
 ...
 [-1.33114223 -0.22172269 -1.3242844  ... -0.78274313 -0.98806491
  -0.69995543]
 [-1.25110186 -0.24600763 -1.28700242 ... -1.36015587 -1.75887319
  -1.56206114]
 [-0.74662205  1.14066273 -0.72203706 ...  0.47201917 -0.2860679
  -1.24094654]]
# Create a function for the model
defmodels(X_train,Y_train):

#Logistic Regression
fromsklearn.linear_modelimportLogisticRegression
log=LogisticRegression(random_state=0)
log.fit(X_train,Y_train)

#Decision Tree
fromsklearn.treeimportDecisionTreeClassifier
tree=DecisionTreeClassifier(criterion='entropy',random_state=0)
tree.fit(X_train,Y_train)

#Random Forest Classifier
fromsklearn.ensembleimportRandomForestClassifier
forest=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
forest.fit(X_train,Y_train)

#Print the models Accuray on the training data
print('[0]Logistic Regression Training Accuracy:',log.score(X_train,Y_train))
print('[1]Decision Tree Classifier Training Accuracy:',tree.score(X_train,Y_train))
print('[2]Random Forest Classifier Training Accuracy:',forest.score(X_train,Y_train))

returnlog,tree,forest
model=models(X_train,Y_train)
[0]Logistic Regression Training Accuracy: 0.9906103286384976
[1]Decision Tree Classifier Training Accuracy: 1.0
[2]Random Forest Classifier Training Accuracy: 0.9953051643192489
fromsklearn.metricsimportconfusion_matrix
```

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 4 Apr. 2021, pp: 338-347 www.ijaem.net ISSN: 2395-5252

```
foriinrange(len(model)):
print('Model',i)
cm=confusion_matrix(Y_test,model[i].predict(X_test))

TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]

print(cm)
print('Testing Accuracy = ',(TP+TN)/(TP+TN+FN+FP))
print()
```

```
Model 0
[[86  4]
 [ 3 50]]
Testing Accuracy =  0.951048951048951

Model 1
[[83  7]
 [ 2 51]]
Testing Accuracy =  0.9370629370629371

Model 2
[[87  3]
 [ 2 51]]
Testing Accuracy =  0.965034965034965
```

```
fromsklearn.metricsimportclassification_report
fromsklearn.metricsimportaccuracy_score

foriinrange(len(model)):
print('Model',i)
print(classification_report(Y_test,model[i].predict(X_test)))
print(accuracy_score(Y_test,model[i].predict(X_test)))
print()
```

```
Model 0
              precision    recall  f1-score   support

           0       0.97      0.96      0.96        90
           1       0.93      0.94      0.93        53

    accuracy                           0.95       143
   macro avg       0.95      0.95      0.95       143
weighted avg       0.95      0.95      0.95       143

0.951048951048951

Model 1
              precision    recall  f1-score   support

           0       0.98      0.92      0.95        90
           1       0.88      0.96      0.92        53

    accuracy                           0.94       143
   macro avg       0.93      0.94      0.93       143
weighted avg       0.94      0.94      0.94       143
```

0.9370629370629371

Model 2

	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

VII. CONCLUSION

Looking back on this project, the overall outcome of results to be observed. This can be evaluated by looking at how well our objectives were met. We have successfully made models that can predict the cancer cells by physical examination data. We also noticed that during the Training period the model 1 that is Decision tree classifier had a accuracy of 100% that is it predicted every outcome correctly but when we gave it with real data during the testing period it's accuracy reduced to 93%. We observed that during the testing period the model random forest classifier reacted or predicted the best with a 96% accuracy. This observation helped us to conclude to this statement – The performance at the training period do not give us the whole picture we have to test it on real data. This passion project helped us to get the in-depth knowledge and experience on the machine learning models. True Artificial Intelligence is decades away but we can achieve it by using machine and deep learning models.

REFERENCES

[1]. SAS Institute Inc. editor.SAS/STAT User's Guide, Version 6. 4th ed. Cary, NC: SAS Institute Inc. 1989.

[2]. Black MM, Speer FD. Nuclear structure in cancer tissues. Surg Gynecol Obstet.105:97-102, 1957.

[3]. Al-Hashimi MMY, Wang XJ. Breast cancer in Iraq, incidence trends from 2000-2009. Asian Pac J Cancer Prev 2014; 15(1): 281–6.

[4]. Cortes, C.; Vapnik, V. Support-vector networks. Mach. Learn. 1995, 20, 273–297.

[5]. Moreno-Seco, F.; Micó, L.; Oncina, J. A modification of the LAESA algorithm for approximated k-NN classification. Pattern Recognit. Lett. 2003, 24, 47–53.

[6]. De Mántaras, R.L. A distance-based attribute selection measure for decision tree induction. Mach. Learn. 1991, 6, 81–92.

[7].] Pavlopoulos, S.A.; Delopoulos, A.N. Designing and implementing the transition to a fully digital hospital. IEEE Trans. Inf. Technol. Biomed. 1999, 3, 6–19.

[8]. Wenbin Yue, Zidong Wang, Hongwei Chen, Annette Payne, Xiaohui Liu. "Machine Learning with Applications in Breast Cancer Diagnosis and Prognosis". 9 May 2018.

[9]. De Mántaras, R.L. A distance-based attribute selection measure for decision tree induction. Mach. Learn. 1991, 6, 81–92.