PIE INFOCOMM PRIVATE LIMITED

PIE INFOCOMM

RESEARCH, TRAINING AND DEVELOPMENT

A SYNOPSIS ON

BREAST CANCER DETECTION USING PYTHON MACHINE LEARNING

Under the Guidance of

AISHWARYA SAXENA

BY-

KRISHNA CHAVAN

SAMIRA ARTE

INDEX:-

INTRODUCTION:-

Breast cancer is the second most exposed cancer in the world. When the growth of breast tissues are out of control is called breast cancer. Breast cancer prediction and prognosis are major challenge to medical community. Breast cancer are prominent cause of death for women. Recurrence of cancer is the biggest fears for cancer patient and this can affect their quality of life. The aim of our project is to predict breast cancer from cancer features with high accuracy. The present paper gives a comparison between the performance of: Logistic Regression , Random Forest and kNN which are among the most influential data mining algorithms.

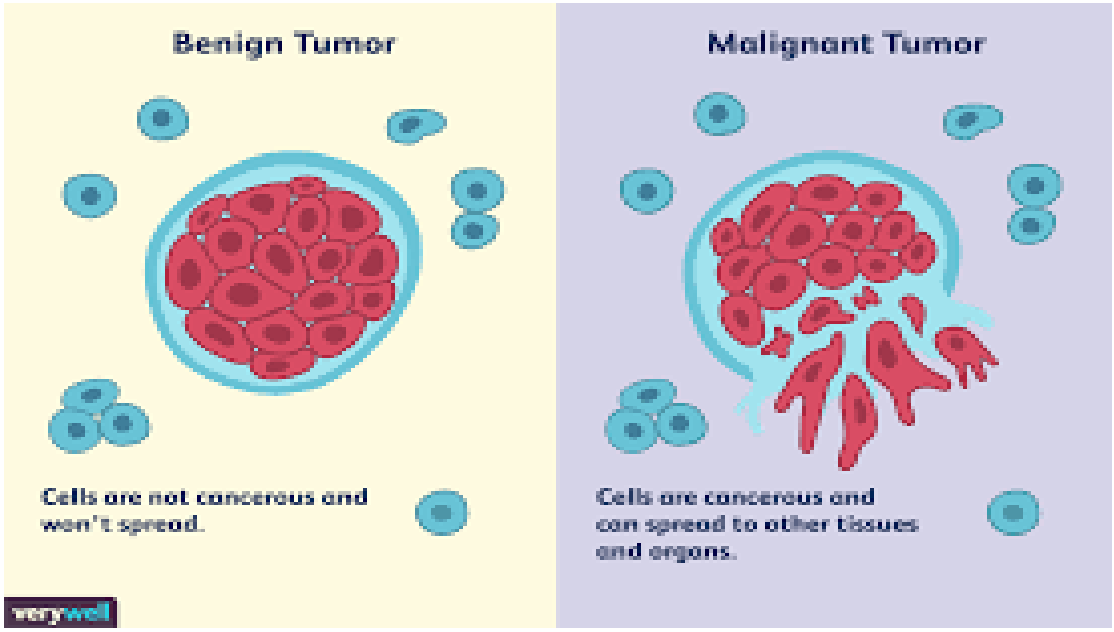
OBJECTIVE :-

The purpose of the project "Breast Cancer Prediction and Detection Using Machine Learning”is to classify the cancer cell as MALIGNANT OR BENIGN .Our aim is to predict and detect breast cancer early even if the tumour size is petite with non-invasive and painless, early detection and diagnosis of such type of disease is a challenging task in order to reduce the number of deaths. The objective of this study is to assess the prediction accuracy of the classification algorithms in terms of efficiency and effectiveness.

BACKGROUND :-

Based on the cancer cell we have classified them in two-

- MALIGNANT
- BENIGN



MALIGNANT- Malignant means that the tumor is made of cancer cells, and it can invade nearby tissues. Some cancer cells can move into the bloodstream or lymph nodes, where they can spread to other tissues within the body—this is called metastasis. Cancer can occur anywhere in the body including the breast, intestines, lungs, reproductive organs, blood, and skin.

BENIGN- A benign breast condition refers to a lump, cyst, or nipple discharge (fluid) of the female or male breast that is not cancerous. For women, the most common ones are: Fibrocystic breast changes. Fibrosis feels like scar tissue and can be rubbery and firm

DATA SET-We have taken the the Kaggle UCI data set for breast cancer prediction

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

In this datasets, There are 32 features which help out to predict cancer.They describe characteristics of the cell nuclei present in the image.

Attribute Information:

1. ID number
2. Diagnosis (M = malignant, B = benign) 3-32) real-valued features are computed for each cell nucleus:

- Radius_mean : Mean of distances from center to points on the perimeter
- Texture_mean : standard deviation of gray-scale values
- Perimeter_mean : mean size of the core tumor
- Smoothness_mean : mean of local variation in radius lengths
- Compactness_mean : mean of perimeter^2 / area - 1.0
- Concavity_mean : mean of severity of concave portions of the contour
- Concave points_mean : mean for number of concave portions of the contour
- Fractal_dimension_mean : mean for "coastline approximation" - 1
- Radius_se : standard error for standard deviation of gray-scale values
- Smoothness_se : standard error for local variation in radius lengths
- Compactness_se : standard error for perimeter^2 / area - 1.0
- Concavity_se : standard error for severity of concave portions of the contour and which are available on data set
- Fractal_dimension_worst : "worst" or largest mean value for "coastline approximation" - 1

These are the few features that increase/decrease the chances of cancer

ABOUT PYTHON :-

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.
Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects

MACHINE LEARNING :-

Machine learning is **a** method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

LIBRARIES USED :-

NUMPY- NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python

PANDAS- pandas is a fast, powerful, flexible and easy to use **open source data analysis and** manipulation tool, built on top of the Python programming language.

SEABORN- Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data

MATPLOTLIB- Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a **multi-platform data visualization library built on NumPy** arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

ALGORITHM USED :-

We have use three algorithms in this project-

- LOGISTIC REGRESSION
- DECISION TRAINING
- RANDOM FOREST CLASSIFIER TRAINING

LOGISTIC REGRESSION :-

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts

P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

DECISION TRAINING :-

Decision trees are supervised learning algorithms used for both, classification and regression tasks where we will concentrate on classification in this first part of our decision tree tutorial. Decision trees are assigned to the information based learning algorithms which use different measures of information gain for learning. We can use decision trees for issues where we have continuous but also categorical input and target features. The main idea of decision trees is to find those descriptive features which contain the most "information" regarding the target feature and then split the dataset along the values of these features such that the target feature values for the resulting sub_datasets are as pure as possible --> The descriptive feature which leaves the target feature most purely is said to be the most informative one. This process of finding the "most informative" feature is done until we accomplish a stopping criteria where we then finally end up in so called **leaf nodes**.

RANDOM FOREST CLASSIFIER TRAINING:-

The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically *a set of decision trees (DT) from a randomly selected subset of the training set and then* It collects the votes from different decision trees to decide the final prediction. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

HARDWARE REQUIREMENT:-

Hardware tools	Minimum Requirements
Processors	I3 or above
Hard disk	10 GB
RAM	4GB
Monitor	17” coloured
Mouse	Optical/touchpad
Keyboard	122 keys/laptop keyboard

SOFTWARE REQUIREMENTS:-

Software tools	Minimum Requirements
Operating System	Windows, linux,MacOs
Technology	Python, Machine Learning
Version	3.6 or above
Scripting Language	Python
IDE	Visual Studio, Jupyter Notebook

CODING :-

```
import numpy
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
df=pd.read_csv( "data.csv" )
df.head()
df.info()
df.isna().sum()
df.shape
df= df.dropna(axis=1)
df.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)
df.head()
sns.pairplot(df.iloc[:,1:5],hue="diagnosis")
df.iloc[:,1:32].corr()

plt.figure(figsize=(10,10))

sns.heatmap(df.iloc[:,1:10].corr(),annot=True,fmt=".0%" )

X=df.iloc[:,2:32].values

Y=df.iloc[:,1].values
print(X)
print(Y)

from sklearn.model_selection import train_test_split

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.20,random_state=0)

from sklearn.preprocessing import StandardScaler

X_train=StandardScaler().fit_transform(X_train)

X_test=StandardScaler().fit_transform(X_test)

def models(X_train,Y_train):

    # Logistic regression

    from sklearn.linear_model import LogisticRegression

    log=LogisticRegression(random_state=0)

    log.fit(X_train,Y_train)

    #decision tree

    from sklearn.tree import DecisionTreeClassifier
```

```
    tree=DecisionTreeClassifier(random_state=0,criterion="entropy" )
    tree.fit(X_train,Y_train)
    #Random forest
    from sklearn.ensemble import RandomForestClassifier

forest=RandomForestClassifier(random_state=0,criterion="entropy",n_estimators=
10)
    forest.fit(X_train,Y_train)
   print('[0]Logistic regression accuracy:',log.score(X_train,Y_train))
    print('[1]Decision Tree accuracy:',tree.score(X_train,Y_train))
    print('[2]Random Forest accuracy:',forest.score(X_train,Y_train))



    return log,tree,forest


model=models(X_train,Y_train)
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report



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

```
pred=model[2].predict(X_test)

print('Predicted values:')

print(pred)

print('actual values:')

print(Y_test)

from joblib import dump

dump(model[2],"Cancer_prediction_model.joblib" )
```

OUTPUT:-

FUTURE SCOPE :-

The analysis of the results signifies that the integration of multidimensional data along with different classification, feature selection and dimensionality reduction techniques can provide auspicious tools for inference in this domain. Further research in this field should be carried out for the better performance of the classification techniques so that it can predict on more variables. We are intending how to parametrize our classification techniques hence to achieve high accuracy. We are looking into many datasets and how further Machine Learning algorithms can be used to characterize Breast Cancer. We want to reduce the error rates with maximum accuracy.

CONCLUSION:-

In this Project we have taken three models in which we have found out that Decision tree accuracy is least that is 94.00. Logistic regression accuracy is better than decision tree that is 96.00 whereas the Random Forest Classifier has the highest accuracy among all of about 98.00

In summary, Random Forest Classifier was able to show its power in terms of effectiveness and efficiency based on accuracy and recall.

REFRENCES & BIBLIOGRAPHY:-

- **INFORMATION:** https://www.wikipedia.org/

  https://www.google.com/

- **DATASET(breast cancer)**: https://www.kaggle.com/datasets

- **MECHANISM**: https://www.researchgate.net/publication/341508593