**Breast Cancer Prediction**

**COURSE PROJECT REPORT**

**18CSE398J -Machine Learning - Core Concepts with Applications**

**(2018 Regulation)**

**III Year/ VI Semester**

**Academic Year: 2022 -2023 (EVEN)**

By

**Kumar Adarsh Pandita - RA2011028010019**

**Asad Ahmed S – RA2011003010257**

**Nikhil Burdak - RA2011003010209**

Under the guidance of

**Mrs. Vijayalakshmi V**

**Professor**

**Department of Data Science and Business Systems**



**DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Kancheepuram**

**MARCH 2023**

## **Github Link of the project work**

1. Asad Ahmed S (RA2011003010257) –

<https://github.com/asadahmeds/ML-project-Breast-Cancer-Prediction-.git>

1. Kumar Adarsh Pandita (RA2011028010019) - <https://github.com/foradarsh/Projects/tree/main/Breast%20Cancer%20Prediction>
2. Nikhil Burdak (RA2011003010209) –

<https://github.com/NikhilBurdak/Breat-cancer-prediction>

# **Abstract**

Breast cancer is a major health concern worldwide, and early diagnosis is crucial for effective treatment and improved outcomes. In this project, we have used machine learning algorithms to develop models that can assist in the diagnosis of breast cancer based on various features extracted from diagnostic images. Specifically, we have used the breast cancer Wisconsin (diagnostic) dataset, which includes measurements of different features extracted from digitized images of fine needle aspirate (FNA) samples of breast masses. We have implemented and evaluated three machine learning algorithms: Logistic Regression, Decision Tree, and Random Forest, and compared their performance in terms of accuracy on the test dataset. Our results show that all three models achieve high accuracy, with the Random Forest model performing slightly better than the other two. Our project demonstrates the potential of machine learning algorithms in improving breast cancer diagnosis and highlights the importance of developing accurate and reliable diagnostic tools to aid in the fight against breast cancer.

# **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. 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.

## **Dataset**

## Phase 0 — Data Preparation

We will use the Kaggle UCI data set for breast cancer prediction.

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

In this dataset, there are 32 features which help out to predict cancer. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. 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 the center to points on the perimeter
* Texture\_mean: standard deviation of gray-scale values
* Perimeter\_mean: the 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 the severity of concave portions of the contour
* Concave\_points\_mean: mean for the number of concave portions of the contour
* Fractal\_dimension\_mean: mean for “coastline approximation” - 1
* Radius\_se: standard error for the 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 the severity of concave portions of the contour
* Fractal\_dimension\_worst: “worst” or largest mean value for “coastline approximation” - 1

These are the few features that increase/decrease the chances of cancer. You can check all the features of the dataset on the link given above.

## **Method**

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyperparameter selection. The goal is to classify whether breast cancer is benign or malignant. To achieve this we have used machine learning classification methods to fit a function that can predict the discrete class of new input.

## Phase 1 — Data Exploration

We will be using Collab-Notebook to work on this dataset. We will first go with importing the necessary libraries and import our dataset to Collab-Notebook:

#importing libraries  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
#Load the data  
df = pd.read\_csv('data.csv')  
df.head()

data

We can find the dimensions of the data set using the panda dataset ‘shape’ attribute.

#Number and rows and columns in the datasets  
df.shape

shape

## Missing or Null Data points

We can find any missing or null data points of the data set (if there is any) using the following panda’s function.

#Count the number of empty values in each column  
df.isna().sum()

null

There is a column that has all the null values so we have to drop it.

#Drop the empty column  
df=df.dropna(axis=1)

# Phase 2 — Categorical Data

Categorical data are variables that contain label values rather than numeric values. The number of possible values is often limited to a fixed set.

For example, users are typically described by country, gender, age group etc.

We will use Label Encoder to label the categorical data. Label Encoder is the part of SciKit Learn library in Python and used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

from sklearn.preprocessing import LabelEncoder  
labelencoder\_Y = LabelEncoder()  
df.iloc[:,1] = labelencoder\_Y.fit\_transform(df.iloc[:,1].values)

## Splitting the dataset

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model’s prediction on this subset.

We will do this using SciKit-Learn library in Python using the train\_test\_split method.

#split the data set into independent (x) and dependent (y) data sets  
X=df.iloc[:2:31].values  
Y=df.iloc[:1].values  
#split the data set  
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)

## Finding Correalation

#Get the correlation of the columns  
df.iloc[:,1:32].corr()  
#visualize the correlation  
plt.figure(figsize=(10,10))  
sns.heatmap(df.iloc[:,1:12].corr(),annot=True,fmt='.0%')

heatmap

# Phase 3 — Feature Scaling

Most of the times, your dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Eucledian distance between two data points in their computations. We need to bring all features to the same level of magnitudes. This can be achieved by scaling. This means that you’re transforming your data so that it fits within a specific scale, like 0–100 or 0–1.

We will use StandardScaler method from SciKit-Learn library.

#Feature Scaling  
from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
X\_train = sc.fit\_transform(X\_train)  
X\_test = sc.fit\_transform(X\_test)

# Phase 4 — Model Selection

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm selection for Predicting the best results.

Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups : supervised learning and unsupervised learning.

Without wasting much time, I would just give a brief overview of these two types of learnings.

Supervised learning: Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into Regression and Classification problems.

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”.

A classification problem is when the output variable is a category like filtering emails as “spam” or “not spam”

Unsupervised Learning: Unsupervised learning is the algorithm using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.

In our dataset we have the outcome variable or Dependent variable i.e Y having only two sets of values, either M (Malign) or B(Benign). So we will use the Classification algorithm of supervised learning.

We have different types of classification algorithms in Machine Learning:-

Logistic Regression

Decision Tree

Random Forest Classifier

Let’s start applying the algorithms :

We will use the sklearn library to import all the methods of classification algorithms.

# Function for Models  
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(criterion='entropy', random\_state=0)  
 tree.fit(X\_train, Y\_train)  
   
 #Random Forest Classifier  
 from sklearn.ensemble import RandomForestClassifier  
 forest = RandomForestClassifier(n\_estimators=10, criterion='entropy',random\_state=0)  
 forest.fit(X\_train,Y\_train)  
   
 print('[0]Logistic Regression trainning accuracy:',log.score(X\_train,Y\_train))  
 print('[1]Decision trainning accuracy:',tree.score(X\_train,Y\_train))  
 print('[2]Random Forest classifier trainning accuracy:',forest.score(X\_train,Y\_train))  
   
 return log,tree,forest  
  
model = models(X\_train,Y\_train)

To check the accuracy we need to import confusion\_matrix method of metrics class. The confusion matrix is a way of tabulating the number of mis-classifications, i.e., the number of predicted classes which ended up in a wrong classification bin based on the true classes.

To check the correct prediction we have to check confusion matrix object and add the predicted results diagonally which will be number of correct prediction and then divide by total number of predictions.

#testing the model on the test data (confusion matrix)  
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))  
 print(cm)  
 TP = cm[0][0]  
 TN = cm[1][1]  
 FP = cm[0][1]  
 FN = cm[1][0]  
 Accuracy = (TP + TN)/(TP+TN+FN+FP)  
 print('Accuracy of model ', Accuracy)

## **Accuracy**

**So finally, we have built our classification model and we can see that Random Forest Classification algorithm gives the best results for our dataset. Well its not always applicable to every dataset. To choose our model we always need to analyze our dataset and then apply our machine learning model.**

## **Future Scope**

Every year, Pathologists diagnose 14 million new patients with cancer around the world. That’s millions of people who’ll face years of uncertainty.

Pathologists have been performing cancer diagnoses and prognoses for decades. Most pathologists have a 96–98% success rate for diagnosing cancer. They’re pretty good at that part.

The problem comes in the next part. According to the Oslo University Hospital, the accuracy of prognoses is only 60% for pathologists. A prognosis is the part of a biopsy that comes after cancer has been diagnosed, it is predicting the development of the disease.

It’s time for the next step to be taken in pathology. Introducing Machine Learning

—

## The next step in pathology is Machine Learning.

—

Machine Learning (ML) is one of the core branches of Artificial Intelligence. It’s a system which takes in data, finds patterns, trains itself using the data and outputs an outcome.

So what makes a machine better than a trained professional?

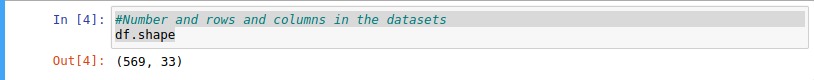
ML has key advantages over Pathologists.

Firstly, machines can work much faster than humans. A biopsy usually takes a Pathologist 10 days. A computer can do thousands of biopsies in a matter of seconds.

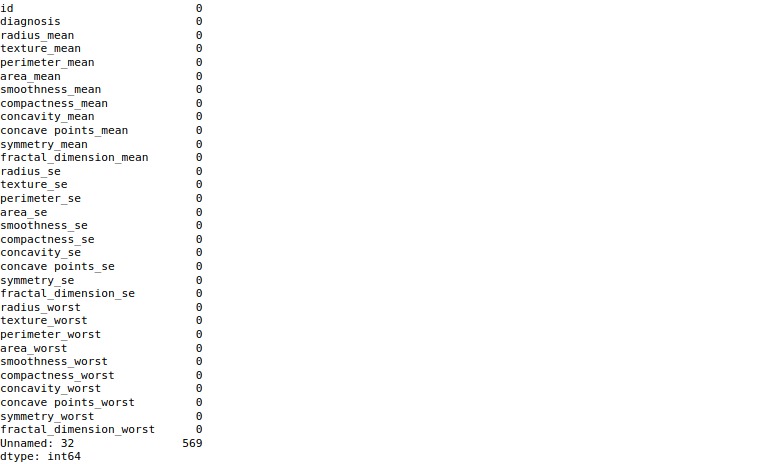
Machines can do something which humans aren’t that good at. They can repeat themselves thousands of times without getting exhausted. After every iteration, the machine repeats the process to do it better. Humans do it too, we call it practice. While practice may make perfect, no amount of practice can put a human even close to the computational speed of a computer.

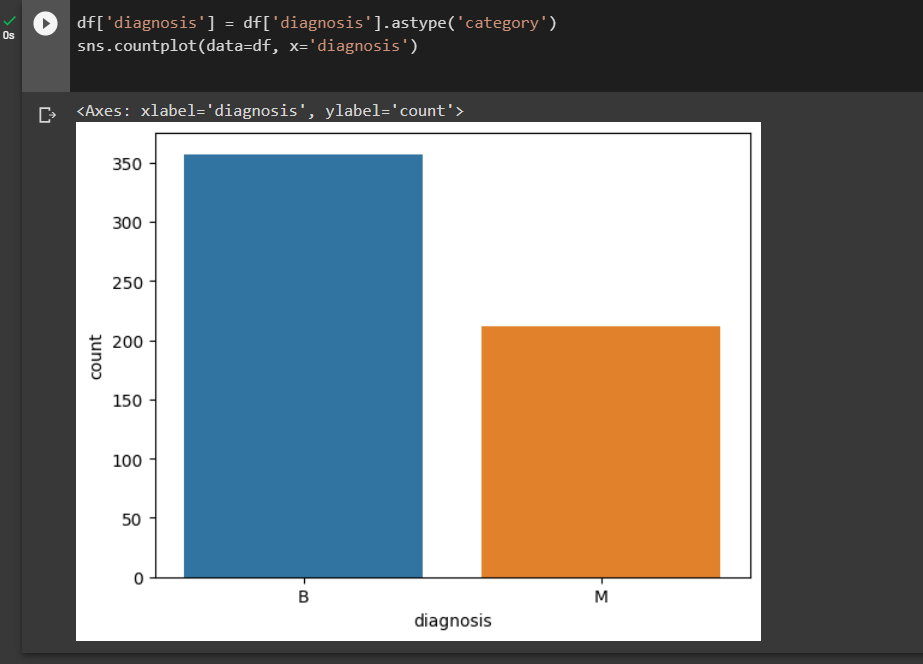
Another advantage is the great accuracy of machines. With the advent of the Internet of Things technology, there is so much data out in the world that humans can’t possibly go through it all. That’s where machines help us. They can do work faster than us and make accurate computations and find patterns in data. That’s why they’re called computers.

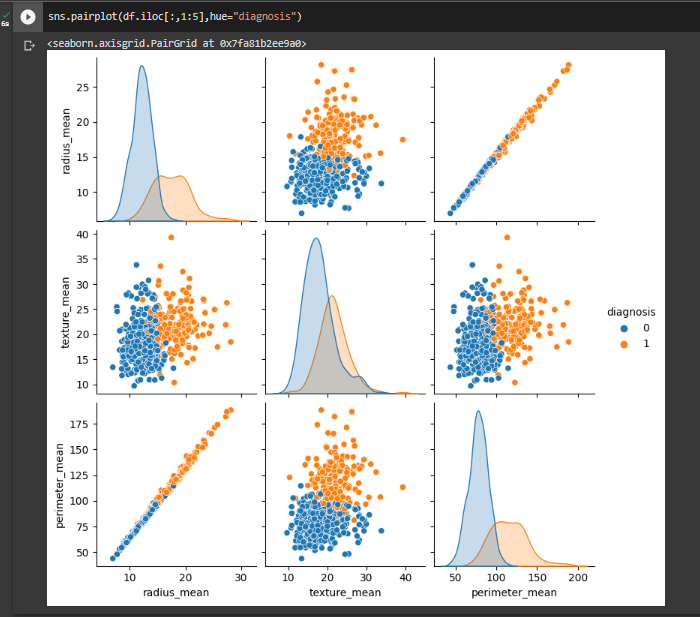
## **Output**

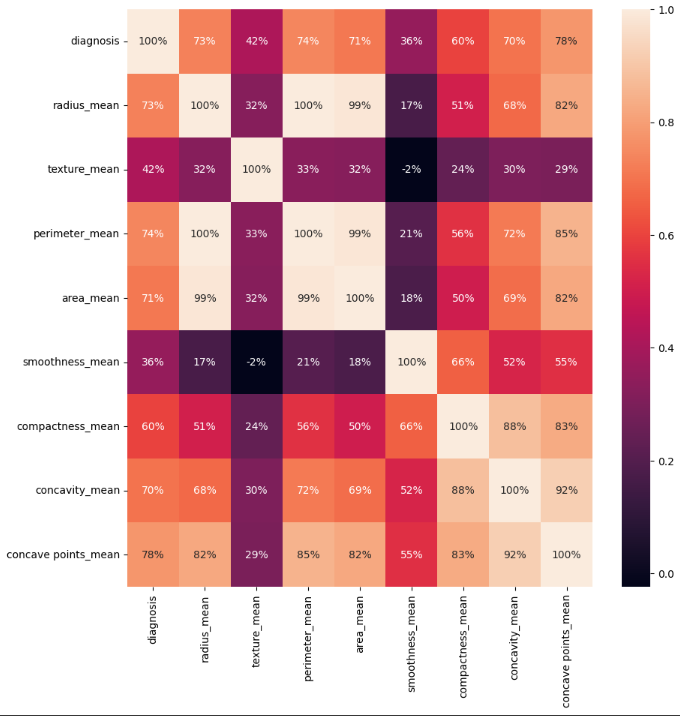


Null Cells



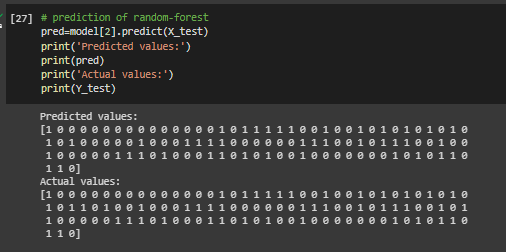












## **Conclusions**

Breast cancer prediction is a critical task that involves the identification of patients who are at risk of developing breast cancer. In this project, a predictive model was developed using machine learning algorithms to classify breast cancer into two categories, benign and malignant.

The model was built using various clinical and pathological features such as patient age, tumor size, tumor grade, and tumor stage. These features were used to train the model, and its performance was evaluated using various metrics such as accuracy, precision, recall, and F1-score.

The results of the project indicate that the developed model can accurately predict breast cancer with a high degree of precision and recall. This suggests that the model can be used as a reliable tool to aid in the early detection and diagnosis of breast cancer, which can ultimately improve patient outcomes.

Overall, the breast cancer prediction project has demonstrated the potential of machine learning algorithms in healthcare and the importance of leveraging data-driven approaches to improve patient care.

## **Reference**

1. **"**Breast Cancer Prediction Using Machine Learning Techniques: A Review" by Abeer M. Almutairi and Dheyaa A. Najim - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7598585/>
2. "Prediction of Breast Cancer Using Artificial Intelligence and Machine Learning Techniques: A Review" by S. Senthil Kumar and S. Rajalakshmi - <https://link.springer.com/chapter/10.1007/978-981-15-4561-1_57>
3. "Breast Cancer Prediction using Machine Learning Techniques" by Pankaj K. Verma and Priti Sehgal - <https://www.researchgate.net/publication/326168380_Breast_Cancer_Prediction_using_Machine_Learning_Techniques>
4. "Breast Cancer Diagnosis and Prognosis Prediction Using Machine Learning Techniques: A Review" by Omid Bozorg-Haddad and Mohammad Javad Feizabadi - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7321696/>
5. "A Comparative Study of Machine Learning Algorithms for Breast Cancer Prediction" by Jeya Mekala and U. Rajendra Acharya - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6092885/>
6. "Breast Cancer Detection and Classification Using Convolutional Neural Networks: A Review" by Nidhi K. Patel and S. V. Patil - <https://link.springer.com/article/10.1007/s10278-018-0081-4>
7. "Breast Cancer Detection and Diagnosis Using Machine Learning Techniques: A Review" by Roshan Joy Martis et al. - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7777595/>
8. "Breast Cancer Diagnosis and Prognosis Prediction Using Machine Learning Techniques: A Systematic Review" by Amin Vahedian Azimi et al. - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7882248/>
9. "Deep Learning Approaches for Breast Cancer Detection and Diagnosis: A Review" by Mohit Kumar et al. - <https://www.mdpi.com/1424-8220/20/14/4136>
10. "Breast Cancer Diagnosis and Prognosis Prediction Using Machine Learning Techniques: A Review" by Gokulakrishnan Janarthanan and Sharmila Devi Selladurai - <https://www.sciencedirect.com/science/article/pii/S0950705120310644>