### **1. OVERVIEW OF BREAST CANCER**

**1.1 Breast Cancer**

Breast cancer is a type of cancer originating from the breast tissue, commonly from the inner lining of the milk ducts or lobules supplying the ducts with milk. Breast cancer occurs in both men and women, although the former type is rare. It remains the number one form of cancer that women are diagnosed with around the world. Even with enhanced treatment, the lack of early detection has put women at even higher risk of dying from this disease. Statistics reveal that there were 40,000 female deaths and 232,670 new cases recorded in the United States in 2014.

**1.2 Types of Breast Cancer**

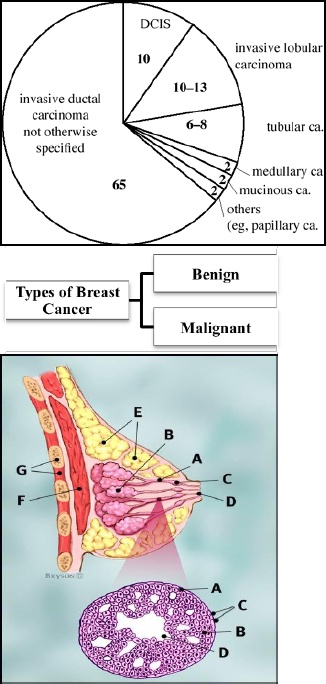
Breast cancer can be of different types depending on the part of the breast it develops on. There have been two broad classification of breast cancer as shown in Figure

**1.3** **Benign Breast Cancer (Non-Invasive):**

It is also known as carcinoma in situ. This type of cancer doesn't spread to neighboring tissue regions and hence is rarely a threat to life. These cells remain entirely in- situ (in their place of origin) because they have not yet developed the ability to spread outside of these ducts, either within the breast or elsewhere in the body. The cancer cells most commonly develop inside the milk ducts and hence it is also known as Ductal carcinoma in situ (DCIS) cancer. Both men and women can develop DCIS.

**1.4 Malignant Breast Cancer (Invasive) :**

Malignant or Invasive is the type in which the cancer has the potential to spread from the breast to other parts of the body and is a threat to life. Often they can removed but sometimes grow back. The most common type of invasive breast cancer is invasive ductal cancer. This accounts for 80 % of all cases of breast cancer.



***Fig. Malignant Breast Cancer***

### **2. Recommended Screening Guidelines**

**Mammography.** The most important screening test for breast cancer is the mammogram. A mammogram is an X-ray of the breast. It can detect breast cancer up to two years before the tumor can be felt by you or your doctor.

**Women age 40–45 or older** who are at average risk of breast cancer should have a mammogram once a year.

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

### 

### **3. Some 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

### **4. Treatment of Breast Cancer**

The prognosis and treatment of breast cancer depend on the stage of the cancer and the type of breast cancer. Breast cancer diagnosed at a later stage requires a different treatment than when diagnosed in its early stages. A patient may have one treatment or a combination.

**4.1 Local Treatment**

**Surgery:**

It either involves removing the cancerous lump (tumor), which is known as breast-conserving surgery or mastectomy that is removal of the whole breast.

**Radiation Therapy:**

It involves controlled doses of radiation are used to kill cancer cells. Usually given to a patient after surgery and chemotherapy to kill any residual cancer cells.

**4.2 Systematic Treatment**

**Chemotherapy:**

It involves using anti-cancer drugs to kill the cancer cells.

**Hormone Therapy:**

Breast cancer may be stimulated to grow by hormones oestrogen or progesterone, which is naturally developed by the body. This treatment involves lowering levels of hormones in the body and reversing the effect.

### **5. Investigation of Breast Cancer**

After being diagnosed with breast cancer, a patient will have various treatment options depending on the stage of breast cancer and the doctor treating. Different factors are worked out for the best treatment, including the type, patients’ age and general health. Figure 8 shows the flow diagram representing a few sets of options available for a patient after being diagnosed with breast cancer.

### **5.1  Data Preparation**

We will use the UCI Machine Learning Repository for breast cancer [dataset](http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29).

[*http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29*](http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29)

The dataset used in this story is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt, which is capable of perform the analysis of cytological features based on a digital scan. The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, than it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valued vector.

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)

The mean, standard error and “worst” or largest (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.

**5.2** **Initial analysis of the dataset**

* The Wisconsin Breast Cancer (original) dataset from the UCI Machine Learning Repository is used in this study. Breast-cancer-Wisconsin has 699 instances (Benign: 458 and Malignant: 241), 2 classes (65.5% malignant and 34.5% benign), and 11 integer-valued attributes.
* The first attribute contains the ID of the patients. Since patient ID would not be useful in the diagnosis of Benign or Malignant tumour, it was removed from analysis.
* There are 16 instances of missing values, represented by ‘?’ in the seventh column, named ‘Bare Nuclei’. Given the severity of accurate analysis in medical field, the column with missing data was removed from analysis.
* The Class labels contain the diagnosis of the tumor as ‘Benign’ and ‘Malignant’, represented by ‘2’ and ‘4’ respectively. For accuracy and ease of analysis when developing a ROC curve, we converted the values ‘2’ and ‘4’ into binary values ‘0’ and ‘1’ (0-Benign, 1-Malignant).

### **Decision Tree**

* A Decision Tree looks to find optimal observations in the data by using subsequent recursive splits [3].
* Each instance in the dataset starts at a root node, then is sorted down the tree into leaf nodes based on the testing condition that is present at each node [4].

#### **Pros**

* Can handle high-dimensional data [3].
* Simple and fast steps to learn about and classify data points [3].

#### **Cons**

* Can be susceptible to noise and overfitting.
* This can be reduced by combining multiple decision trees into a Random Forest and training each tree on different parts of the training set (sampled independently for each tree)

### **6. What is Naive Bayes algorithm?**

Naive Bayes is a classification technique based on Bayes’ Theorem(*Probability theory*) with an assumption that all the features that predicts the target value are independent of each other. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature in determining the target value.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) - *(read as Probability of* ***c\*\* given \*\*x****)*, from P(c), P(x) and P(x|c). Look at the equation below:

P(x∣c) = P(c∣x)P(c)P(x)

where,

* *x is set of features*
* *c is set of classes*
* P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
* P(c) is the prior probability of class **c**.
* P(x|c) is the observation density or likelihood which is the probability of predictor(the query **x**) given class.
* P(x) is the prior probability of predictor **x**, and it is also called as Evidence.

**6.1 Why should we use Naive Bayes ?**

* As stated above, It is ***easy*** to build and is particularly useful for ***very large data sets***.
* It is **extremely fast** for both training and prediction.
* It provide straightforward probabilistic prediction.
* It is often very easily interpretable.
* It has very few (if any) tunable parameters.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

### **6.2 Types of Naive Bayes Classifier:**

#### **Multinomial Naive Bayes:**

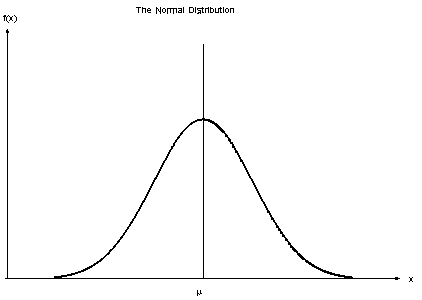
This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

#### **Bernoulli Naive Bayes:**

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

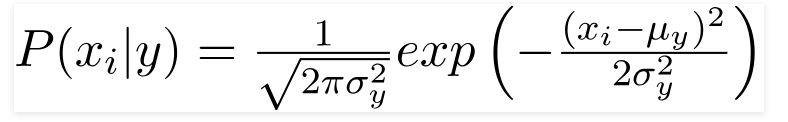
#### **Gaussian Naive Bayes:**

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.



***Fig.2 Normal distribution***

Since the way the values are present in the dataset changes, the formula for conditional probability changes to,



Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

#### **Pros**

* It is a relatively simple model compared to other machine learning models while still having good performance and it’s computationally cheap [8].
* It performs well even with small datasets.
* It is highly used as a text classifier due to its high performance dealing with text.

#### **Cons**

* It assumes that features are independent, which is rarely the case in real world datasets.

## **6.3 Naive Bayes in the Industry**

Now that you have an idea of What exactly is Naïve Bayes, how it works, let’s see where is it used in the Industry?

### 

### **News Categorization:**

Starting with our first industrial use, it is News Categorization, or we can use the term text classification to broaden the spectrum of this algorithm. News on the web is rapidly growing where each news site has its own different layout and categorization for grouping news. Companies use a web crawler to extract useful text from HTML pages of news article contents to construct a Full-Text-RSS. Each news article contents is tokenized(categorized). In order to achieve better classification result, we remove the less significant words i.e. stop – word from the document. We apply the naive Bayes classifier for classification of news contents based on news code.

### **Spam Filtering:**

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They typically use a bag of words features to identify spam email, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with a spam and non-spam emails and then using Bayes’ theorem to calculate a probability that an email is or is not spam.

Particular words have particular probabilities of occurring in spam email and in legitimate email. For instance, most email users will frequently encounter the word “Lottery” and “Lucky Draw” in spam email, but will seldom see it in other emails. Each word in the email contributes to the email’s spam probability or only the most interesting words. This contribution is called the posterior probability and is computed using Bayes’ theorem. Then, the email’s spam probability is computed over all words in the email, and if the total exceeds a certain threshold (say 95%), the filter will mark the email as a spam.

### **Medical Diagnosis:**

Nowadays modern hospitals are well equipped with monitoring and other data collection devices resulting in enormous data which are collected continuously through health examination and medical treatment. One of the main advantages of the Naive Bayes approach which is appealing to physicians is that “all the available information is used to explain the decision”. This explanation seems to be “natural” for medical diagnosis and prognosis i.e. is close to the way how physicians diagnose patients.

When dealing with medical data, Naïve Bayes classifier takes into account evidence from many attributes to make the final prediction and provides transparent explanations of its decisions and therefore it is considered as one of the most useful classifiers to support physicians’ decisions.

### **Weather Prediction:**

Weather is one of the most influential factors in our daily life, to an extent that it may affect the economy of a country that depends on occupation like agriculture. Weather prediction has been a challenging problem in the meteorological department for years. Even after the technological and scientific advancement, the accuracy in prediction of weather has never been sufficient.

A Bayesian approach based model for weather prediction is used, where posterior probabilities are used to calculate the likelihood of each class label for input data instance and the one with maximum likelihood is considered resulting output.

**Hypothesis statement**

* We expect both Naive Bayes and Decision Tree to provide good results, but predict that Naive Bayes will perform slightly better than Decision Tree.
* This is due to the results in Asri et al., in which both models had an accuracy error of under 5%, but Naive Bayes performed better than Decision Tree on the same breast cancer dataset .
* The average accuracy of the Naive Bayes model tested in Asri et al. was 95.99%, and that for Decision Tree was 95.13%

### **7. Training and evaluation methodology**

* The breast cancer dataset, consisting of 699 data points was used.
* The original data set was divided into 70% Training and Validation data and 30% Test data, using Random Sampling.
* The Training and Validation data was further divided into 70% Training and 30% Validation data, using Random Sampling.
* The hyperparameters for both models were selected by using Grid Search and 10-fold cross validation. The parameters that gave the lowest average classification error over 10 runs were chosen.
* The two models were evaluated by comparing the accuracy of the predictions (benign or malignant tumor) by the two models on the Test data set.

### **Decision Tree Parameters**

* Grid Search with 10-fold cross validation was done to optimize hyperparameters.
* The hyperparameters optimised were ‘Number of predictors to select at random for each split’ and ‘Split Criterion’.
* The prior probabilities of the Classes were uniformly distributed to maintain the balance of classes in data set.
* ‘Number of predictors to select at random for each split’ hyperparameter used values from 1 through 8.
* ‘Split Criterion’ hyperparameter used the gini index, the twoing rule, and maximum deviance reduction (cross-entropy).
* The top performing hyperparameters were chosen by averaging the loss function (mean-squared error) over the 10 CV runs for each set of hyperparameters. The results can be seen in Figure

### **Decision Tree Results**

* The top performing hyperparameters were found to be: ‘Number of predictors to select at random for each split’ = ‘2’ and ‘Split Criterion’ = ‘gini index’.
* The performance of all three Split Criterion values generally was worse when measured against a high number of predictors to sample than a lower number of predictors to sample.
* The best performing model had an accuracy of 91.905% on the Test set.

### **7.1 Naive Bayes Parameters**

* Grid Search with 10-fold cross validation was done to optimize hyperparameters.
* The hyperparameter optimised was ‘Distribution Names’.
* The prior probabilities of the Classes were uniformly distributed to maintain the balance of classes in data set.
* The ‘Distribution’ patterns considered were ‘normal’, ‘multinomial’, and ‘kernel’ distribution.
* The top performing hyperparameters were chosen by averaging classifier loss function (mean-squared error) over the 10 CV runs for each set of hyperparameters.

### **Naive Bayes Results**

* The top performing distribution parameter was found to be ‘normal distribution’.
* Both normal and kernel distribution performed relatively well on the validation data, whereas multinomial distribution performed significantly worse than the other two.
* The best performing model had an accuracy of 96.667% on the Test set.

### **7.2 Critical evaluation of results**

* Our hypothesis was correct in predicting that the Naive Bayes model would outperform the Decision Tree, as was the case in Asri et al.
* Naive Bayes was more accurate than the Decision Tree on the test set, with a difference in accuracy of 4.762%.
* Naive Bayes models don’t need much data in order to get good classification accuracy. That makes it less suspect to underfitting or overfitting and can better generalize to unseen data. This is also a contributing factor to why our Naive Bayes model was able to perform even better on the Test set than on the Validation set.
* Decision Tree had almost a 3% worse mean-squared error in the Test set than in the Validation set. This shows that our Decision Tree may not have generalized as well on our Test set. This makes sense since Decision Trees can be prone to noise and overfitting, and therefore perform worse than other models when testing on unseen data compared to other machine learning methods such as Naive Bayes.
* Instead of one Decision Tree, combining many Decision Trees into a Random Forest would reduce the effect of noise and overfitting and may provide a higher accuracy than Naive Bayes. This is based on a comparison by Caruana et al. in which Random Forest outperforms Naive Bayes
* In terms of speed, both models were quick to run, with Naive Bayes taking 0.0264 seconds to run and Decision Tree taking 0.0232 seconds to run. This is consistent with the research of Kaur et al. for Naive Bayes and Patel et al. for Decision Tree. To tune hyperparameters for Decision Tree, the model was run 240 times. Due to the fast speed of decision trees, this process took only 4.460 seconds.
* It is to be considered that a relatively small dataset has been used while analysing the results. The small size of test set allowed for bigger changes in percentage accuracy when small number of points were misclassified. Additionally, with small datasets, Naive Bayes models tend to have an advantage over other models. For larger datasets, this might not be the case.

**Relevance:**

We use python language to makes this project and we also use technology like machine learning.

Naive Bayes is a classifier that uses Bayes theorem to predict the probability that a data point is part of a certain class.

The class with the highest probability is the class that the data point is assigned to.

At last we make a webapp that helps doctor to directly predict the correct tumor just by putting the values of necessary features.

**7.3 Hypothesis statement**

* We expect both Naive Bayes and Decision Tree to provide good results, but predict that Naive Bayes will perform slightly better than Decision Tree.
* This is due to the results in Asri et al., in which both models had an accuracy error of under 5%, but Naive Bayes performed better than Decision Tree on the same breast cancer dataset.
* The average accuracy of the Naive Bayes model tested in Asri et al. was 95.99%, and that for Decision Tree was 95.

**Training and evaluation methodology**

* The breast cancer dataset, consisting of 699 data points was used.
* The original data set was divided into 70% Training and Validation data and 30% Test data, using Random Sampling.
* The Training and Validation data was further divided into 70% Training and 30% Validation data, using Random Sampling.
* The hyperparameters for both models were selected by using Grid Search and 10-fold cross validation. The parameters that gave the lowest average classification error over 10 runs were chosen.
* The two models were evaluated by comparing the accuracy of the predictions (benign or malignant tumor) by the two models on the Test data set.

### **8. Implementation - Code**

**8.1 Naive Bayes Classifier**

# -\*- coding: utf-8 -\*-

"""Naive Bayes Classifier from Scratch 2.0.ipynb

# \*\*Naive Bayes Classifier For Classifying Whether The Tumor Is Benign or Malignant\*\*

\*\*\*

\*\*What is Naive Bayes algorithm?\*\*

Naive Bayes is a classification technique based on Bayes’ Theorem(\*Probability theory\*) with an assumption that all the features that predicts the target value are independent of each other. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature in determining the target value.

> Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) - \*(read as Probability of \*\*c\*\* given \*\*x\*\*)\*, from P(c), P(x) and P(x|c). Look at the equation below:

>

> $$\mathbf{P} \left({x \mid c} \right) = \frac{\mathbf{P} \left ({c \mid x} \right) \mathbf{P} \left({c} \right)}{\mathbf{P} \left( {x} \right)}$$

where,

\* \*x is set of features\*

\* \*c is set of classes\*

\* P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).

\* P(c) is the prior probability of class \*\*c\*\*.

\* P(x|c) is the observation density or likelihood which is the probability of predictor(the query \*\*x\*\*) given class.

\* P(x) is the prior probability of predictor \*\*x\*\*, and it is also called as Evidence.

\*\*Why should we use Naive Bayes ?\*\*

\* As stated above, It is \*\*\_easy\_\*\* to build and is particularly useful for \*\*\_very large data sets\_\*\*.

\* It is \*\*extremely fast\*\* for both training and prediction.

\* It provide straightforward probabilistic prediction.

\* It is often very easily interpretable.

\* It has very few (if any) tunable parameters.

\* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

"""

#Importing the Libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

import random

import scipy.stats as S

# Importing tha Datasets

Data = pd.read\_csv("data.csv")

Data.dropna(axis=1,inplace=True)

Data.head()

"""# Principal Component Analysis (PCA)

it is generally used for dimensionality reduction...

In this data we are using PCA for finding important features which have more effect/weightage on finding posterior probability

"""

def PCA(Data):

Data\_array = np.array(Data)

#\_\_\_\_\_\_ Make Zero Mean Distribution\_\_\_\_\_

Means\_value = np.mean(Data\_array,axis=0) #finding mean of each columns

Means\_value = Means\_value.reshape(1,Data\_array.shape )

Centred\_value = Data\_array - Means\_value #Substracting respective mean with their respective columns values

#\_\_\_finding covariance matrix of zero mean distrubuted value\_\_\_\_

Covariance\_matrix = np.cov(Centred\_value , rowvar=0)

Covariance\_matrix.shape

#\_\_finding eigen values and eigen vectors of covariance matrix

values,vectors = np.linalg.eig(Covariance\_matrix)

values = values.reshape(1,len(values))

values\_index = np.argsort(values) #getting original index on the basis of sorted values

values\_index = values\_index[0]

values\_index = (values\_index[::-1]) # transform sorted\_index to descn. order

values = values[:,values\_index] #getting values which will be in descn. order

#\_\_\_\_\_\_\_finding cummulative sum for calculating weightage change\_\_\_\_

weightage\_of\_features = np.cumsum(values)/np.sum(values)

features\_list\_index=[] #\_\_list of important features index

for i in range(0,len(weightage\_of\_features)):

weightage\_in\_percent = weightage\_of\_features[i]\*100

if weightage\_in\_percent <= 99.9:

features\_list\_index.append(values\_index[i])

return(features\_list\_index)

no\_features = PCA(Data.iloc[:, 2:])

print(no\_features)

"""> By Applying PCA on our dataset, important features are 2+0, 2+1 i.e. 2, 3 index of our data which is radius\_mean and texture\_mean

> \*Now split our Data into training and testing set\*

"""

train, test = train\_test\_split(Data, test\_size=0.3)

"""># Training of Model and Seperating By Class: { Benign, Malignant }"""

# Seperarting data by class

dataB = train[train['diagnosis'] == 'B']

dataM = train[train['diagnosis'] == 'M']

# This function returns the mean and covariance matrix of provided data

def calculate\_mean\_covMat(data):

return data.iloc[:, 2:4].mean(), np.cov(data.iloc[:, 2:4],rowvar=0)

# Calculating mean and covariance matrix of Benign

BT\_mean, BT\_cov = calculate\_mean\_covMat(dataB)

# Calculating mean and covariance matrix of Benign

MT\_mean, MT\_cov = calculate\_mean\_covMat(dataM)

# Calculating the P(B) and P(M) independently

P\_B = dataB.shape[0]/train.shape[0]

P\_M = dataM.shape[0]/train.shape[0]

"""Before we go any further we should know \*\*Posterior Conditional Probability\*\* which is,

$\mathbf{P} \left({x \mid c} \right) = \mathbf{P} \left ({c \mid x} \right) \mathbf{P} \left({c} \right)$

where, $\mathbf{P} \left ({c \mid x} \right)$ is \*\*\*Observation Distribution\*\*\*

And Mathematical Formula of Observation Distribution is

$$ \frac {1}{(\sqrt{2}\pi)^2\sqrt{\textstyle\sum}}e^{-0.5}A^T{\textstyle\sum}^{-1}A $$

where,

\* $ \textstyle\sum $ is a covariance matrix

\* A is a vector which contains

$

A=

\left [

{\begin{array}{c}

R\_i - Mean(radius\\_mean) \\

T\_i - Mean(texture\\_mean) \\

\end{array} }

\right]

$

># Testing of Model

"""

# This function returns the Observation Distribution

def calculateObservationDistribution(test, mean, covMat):

return S.multivariate\_normal.pdf(test, mean, covMat)

# Here we are Calculating the Posterior Conditional Probability of Benign and Malignant Data

PosteriorConditionalProbabilityB = calculateObservationDistribution(test.iloc[:, 2:4], BT\_mean, BT\_cov)\*P\_B

PosteriorConditionalProbabilityM = calculateObservationDistribution(test.iloc[:, 2:4], MT\_mean, MT\_cov)\*P\_M

"""># \*\*\*Prediction\*\*\*"""

# In this section we are labelling whether it is Benign or Malignant

# creating empty list of label prediction

label\_prediction = []

# Comparing PosteriorConditionalProbability of Benign and Malignant

for b, m in zip(range(len(PosteriorConditionalProbabilityB)), range(len(PosteriorConditionalProbabilityM))):

if(PosteriorConditionalProbabilityB[b] > PosteriorConditionalProbabilityM[m]):

label\_prediction.append('B')

else:

label\_prediction.append('M')

# list to array

label\_prediction = np.array(label\_prediction)

"""> # \*\*\*Finding an Accuracy\*\*\*"""

# Comapring all the rows of diagnosis of test data with label prediction

count = 0

total = len(test)

for i in range(total):

if test.iloc[i, 1] == label\_prediction[i]:

count += 1

accuracy = count/total

print('Accuracy = ' + str(accuracy\*100) + '%')

Accuracy = 91.2%

**9. Support Vector Machine (SVM):**

It is used for both classification and regression problems. But mainly it is used for classification problems. The main concept of SVM is to plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Here n would be the features we would have. Following is a simple graphical representation to understand the concept of SVM −

In the above diagram, we have two features hence we first need to plot these two variables in two dimensional space where each point has two co-ordinates, called support vectors. The line splits the data into two different classified groups. This line would be the classifier.

🡺 **Support Vector Machine Implimentation:**

# Part 1 - Data Preprocessing

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('Breast Cancer Data.csv')

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

y = dataset.iloc[:, 1].values

# Encoding categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X\_1 = LabelEncoder()

y = labelencoder\_X\_1.fit\_transform(y)

# Splitting the dataset into the Training set and Test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from time import time

t = time()

clf = SVC()

clf.fit(X\_train, y\_train)

output = clf.predict(X\_test)

acc = accuracy\_score(y\_test, output)

print("The accuracy of testing data: ",acc)

print("The running time: ",time()-t)

**🡺SVM functionality:**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from time import time

def train\_svm():

# Importing the dataset

dataset = pd.read\_csv('Breast Cancer Data.csv')

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

y = dataset.iloc[:, 1].values

# Encoding categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X\_1 = LabelEncoder()

y = labelencoder\_X\_1.fit\_transform(y)

# Splitting the dataset into the Training set and Test set

global X\_test, y\_test

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

global sc

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

clf = SVC(probability=True)

clf.fit(X\_train, y\_train)

return clf

def test\_svm(clf):

t = time()

output = clf.predict(X\_test)

acc = accuracy\_score(y\_test, output)

print("The accuracy of testing data: ",acc)

print("The running time: ",time()-t)

def predict\_svm(clf, inp):

t = time()

inp = sc.transform(inp)

output = clf.predict(inp)

acc = clf.predict\_proba(inp)

print("The running time: ",time()-t)

return output, acc, time()-t;

**🡺 web application:**

from flask import Flask, render\_template, request

from svm\_func import train\_svm, test\_svm, predict\_svm

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from time import time

app = Flask(\_\_name\_\_)

app.url\_map.strict\_slashes = False

@app.route('/')

def hello\_method():

return render\_template('home.html')

@app.route('/predict', methods=['POST'])

def login\_user():

if(request.form['space']=='None'):

data = []

string = 'value'

for i in range(1,31):

data.append(float(request.form['value'+str(i)]))

for i in range(30):

print(data[i])

else:

string = request.form['space']

data = string.split()

print(data)

print("Type:", type(data))

print("Length:", len(data))

for i in range(30):

print(data[i])

data = [float(x.strip()) for x in data]

for i in range(30):

print(data[i])

data\_np = np.asarray(data, dtype = float)

data\_np = data\_np.reshape(1,-1)

out, acc, t = predict\_svm(clf, data\_np)

if(out==1):

output = 'Malignant'

else:

output = 'Benign'

acc\_x = acc[0][0]

acc\_y = acc[0][1]

if(acc\_x>acc\_y):

acc = acc\_x

else:

acc=acc\_y

return render\_template('result.html', output=output, accuracy=round(acc\*100,3), time=t)

@app.route('/profile')

def display():

return render\_template('profile.html')

if \_\_name\_\_=='\_\_main\_\_':

global clf

clf = train\_svm()

test\_svm(clf)

print("Done")

app.run(port=4995)

### **10. CONCLUSION**

The automatic diagnosis of Breast cancer is an important real-world medical problem. Detection of breast cancer in its early stages is the key for treatment. This paper shows how decision trees are used to model actual diagnosis of Breast cancer for local and systematic treatment, along with presenting other techniques that can be applied. Experimental results show the effectiveness of the proposed model. The performance of decision tree technique was investigated for the Breast cancer diagnosis problem.

Naive Bayes is a simple model that doesn’t need much data or hyperparameter tuning, and Decision Trees need more hyperparameter tuning as results while tuning can have a large variance. Balance the Class Labels in data set, without losing observations, by using SMOTE or ADASYN methods. Compare the models using a very large dataset to investigate how size affects the Naive Bayes model, which doesn’t need much data to get accurate results. Evaluate the effectiveness of combining the models into an ensemble model to check the change in accuracy.

### **11. References**

1. Hiba Asri, H. M. (2016). Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis. Procedia Computer Science , 1064–1069 .

2. I.Fotiadis, K. P. ( 2015). Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, Volume 13,8–17.

3. Bhaskar N. Patel, S. G. (2012). Efficient Classification of Data Using Decision. Bonfring International Journal of Data Mining, Vol. 2, №1.

4. Buckles, X. Y. (2003). Gene Expression Classification: Decision Trees vs. SVMs. FLAIRS, 92–96.

5. Zhang, H. (2004). The Optimality of Naive Bayes.New Brunswick: University of New Brunswick.

6. NHS. (n.d.). Retrieved from <https://www.nhs.uk/conditions/breast-cancer.>

7. Caruana, R., & Niculescu-Mizil, A. (n.d.). An Empirical Comparison of Supervised Learning Algorithms. Ithaca: Department of Computer Science, Cornell University.