UNIVERSITY MOHAMED V

Faculty of Sciences Rabat

INFO Department

PROJECT REPORT

1ST MASTER'S DEGREE SECTOR : SID

Subject

Biomedical Field : Breast Cancer Prediction

By

RABANY Brahim

SIDI ELY Mohamed Saleh

Supervisor

Imane LASRI

College year : 2022 – 2023

Contents

List of the Figures

List Of abbreviations

FNA	Fine needle aspiration
DM	Data Mining
SVM	Support Vector Machine

Introduction

Breast cancer is the most commonly diagnosed cancer type, accounting for 1 in 8 cancer diagnoses worldwide. In 2020, there were about 2.3 million new cases of breast cancer globally and about 685 000 deaths from this disease, with large geographical variations observed between countries and world regions. It's necessary to find a solution to predict breast cancer.

Fine needle aspiration(FNA) is a type of biopsy procedure. In fine needle aspiration, a thin needle is inserted into an area of abnormal-appearing tissue or body fluid. As with other types of biopsies, the sample collected during fine needle aspiration can help make a diagnosis or rule out conditions such as cancer.

Based on machine learning and a dataset of FNA, the latter describes characteristics of the cell nuclei present in the image. We will train a model who can predict breast cancer.

This project will make a performance comparison between different data mining algorithms in order to assess the correctness in classifying data with respect to efficiency and effectiveness of each algorithm in terms of accuracy, precision, sensitivity and specificity, in order to find the best diagnosis.

The report is structured as follows : The first part of this report presents the context of our project, the problem and the solution we propose to carry out this work.

The second part focuses on the definition and description of cancer and especially breast cancer.

The third part we will describe the algorithms used to implement this project.

In the last part we will describe the phase of the realization and the coding of the application by indicating the tools used throughout the development.

Chapter 1 : Project Overview

1.1 Aim of the project

The objective of this report is to train machine learning models to predict whether a breast cancer cell is Benign or Malignant. Data will be transformed and its dimension reduced to reveal patterns in the dataset and create a more robust analysis. As previously said, the optimal model will be selected following the resulting accuracy, sensitivity, and f1 score, amongst other factors. We will later define these metrics. We can use machine learning method to extract the features of cancer cell nuclei image and classify them. It would be helpful to determine whether a given sample appears to be Benign ("B") or Malignant ("M").

The machine learning models that we will applicate in this report try to create a classifier that provides a high accuracy level combined with a low rate of false-negatives (high sensitivity).

1.2 Dataset

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.

Fine needle aspiration(FNA) is a type of biopsy procedure. In fine needle aspiration, a thin needle is inserted into an area of abnormal-appearing tissue or body fluid. As with other types of biopsies, the sample collected during fine needle aspiration can help make a diagnosis or rule out conditions such as cancer.

Attribute Information :

- ID number
- Diagnosis (M = malignant, B = benign)

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

- Radius (mean of distances from center to points on the perimeter)
- Texture (standard deviation of gray-scale values)

- Perimeter
- Area
- Smoothness (local variation in radius lengths)
- Compactness (perimeter^2 / area - 1.0)

- Concavity (severity of concave portions of the contour)
- Concave points (number of concave portions of the contour)
- Symmetry
- 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.

1.3 Workflow

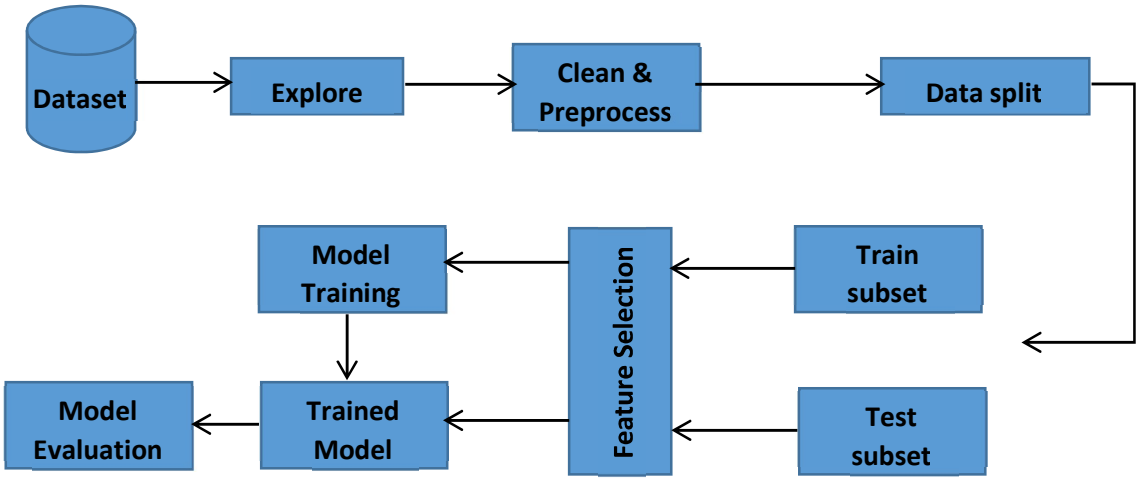


Figure 1 : WorkFlow

Chapter 2 : Medical Data Mining Applications

In today's medical field, data mining is mostly utilized to anticipate various diseases, assist with diagnosis, and advise clinicians on clinical decisions. Data mining, on the other hand, has far greater potential : it may deliver question-based answers, anomaly-based discoveries, more informed decisions, probability measurements, predictive modeling, and decision assistance.

Data Mining Techniques can effectively establish association rules and then identify relevant correlations within a set of medial data.

Some of medical data mining applications :

2.1 Clinical Decision-Making is Improved

The use of CDSS is becoming more frequent in hospitals (clinical decision support systems). These systems either make decisions based on a knowledge base and rules or employ machine learning to generate conclusions based on data analysis.

Data mining is particularly useful in the latter type of solution, such as when comparing a patient's history and symptoms to current clinical studies or similar instances.

2.2 Measuring the Efficacy of Treatment

Data mining may compare and contrast symptoms in a clear and transparent manner, while also explaining the core reason and arranging the most successful treatment approaches. Hospitals can now prepare to give low-cost medical treatments and, as a result, create closer contact with admitted patients in order to collect clinical profiles.

2.3 Data Mining Reduces the Risk of Drug Interactions

Mining tools are used by medical organizations to assist doctors in deciding when to give medication. Because of the potential for lethal interactions, a patient may need to cease taking another prescription in order to take some pharmaceuticals. Analysts can utilize healthcare data to prevent these interactions from happening in the first place.

Because some interactions are uncommon, not all doctors are aware of them. Before generating any hypothesis, scientists can use big data analytics to uncover these fewer common relationships. While data mining aids in the understanding of cardiovascular drug interactions, it can also provide information on other drugs

2.4 Increased Diagnostic Precision

In medical, data mining enables clinicians to make more conclusive, evidence-based diagnoses in less time. While a skilled clinician must still make the final choice, AI-enabled software can evaluate large amounts of data in seconds.

X-ray or MRI pictures, as well as blood tests, can be promptly analyzed and classified to aid in the early diagnosis of cancers and other abnormalities. When it comes to treating complex illnesses with ambiguous symptoms, quickness and accuracy of interpretation can make all the difference.

2.5 Breast Cancer Prediction

2.5.1 What is Cancer

There are many types of cancer. Cancer can develop anywhere in the body and is named for the part of the body where it started. For instance, breast cancer that starts in the breast is still called breast cancer even if it spreads (metastasizes) to other parts of the body.

There are two main categories of cancer:

- **Hematologic (blood) cancers** are cancers of the blood cells, including leukemia, lymphoma, and multiple myeloma.

- **Solid tumor cancers** are cancers of any of the other body organs or tissues. The most common solid tumors are breast, prostate, lung, and colorectal cancers.

These cancers are alike in some ways, but can be different in the ways they grow, spread, and respond to treatment. Some cancers grow and spread fast. Others grow more slowly. Some are more likely to spread to other parts of the body. Others tend to stay where they started.

2.5.2 What is Tumor

A tumor is a lump or growth. Some lumps are cancer, but many are not.

- Lumps that are not cancer are called **benign**
- Lumps that are cancer are called **malignant**

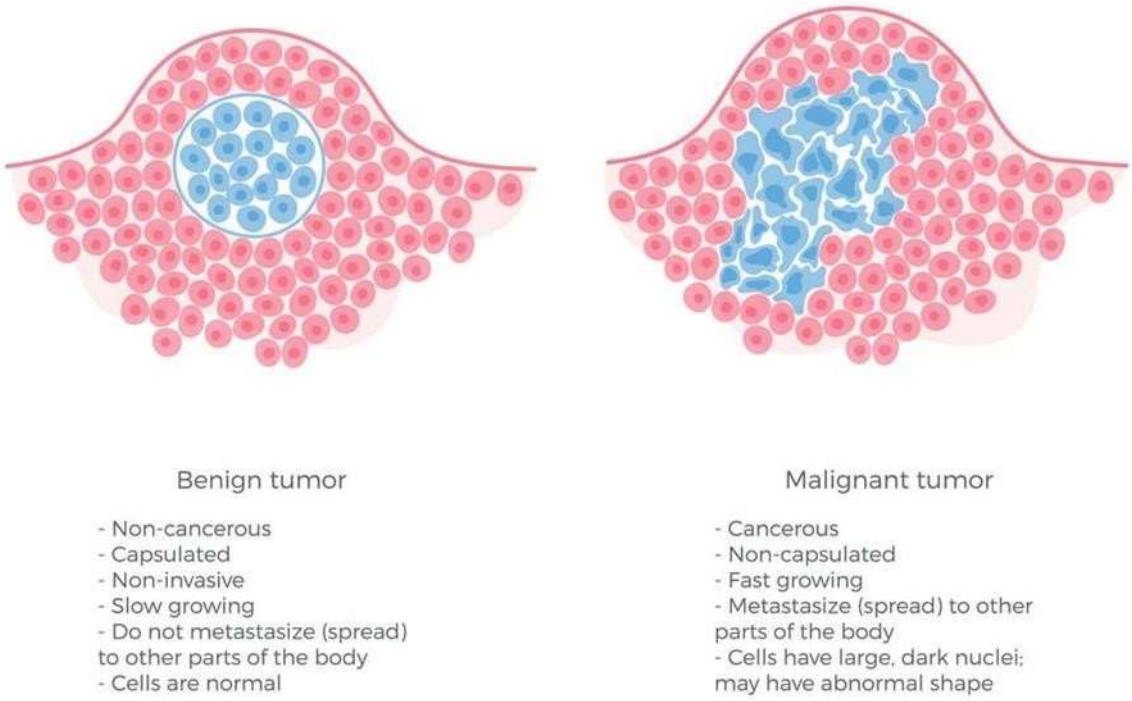


Figure 2 : Bening tumor vs Malignant tumor

What makes cancer different is that it can spread to other parts of the body while benign tumors do not. Cancer cells can break away from the site where the cancer started. These cells can

travel to other parts of the body and end up in the lymph nodes or other body organs causing problems with normal functions.

2.5.3 What is Breast Cancer & When start

Breast cancer is a type of cancer that starts in the breast. It can start in one or both breasts.

When cancer start ?

Cancer starts when cells begin to grow out of control.

Breast cancer occurs almost entirely in women, but men can get breast cancer, too.

It’s important to understand that most breast lumps are benign and not cancer (malignant). Noncancer breast tumors are abnormal growths, but they do not spread outside of the breast. They are not life threatening, but some types of benign breast lumps can increase a woman's risk of getting breast cancer. Any breast lump or change needs to be checked by a health care professional to find out if it is benign or malignant (cancer) and if it might affect your future cancer risk.

Breast cancer is the most commonly diagnosed cancer type, accounting for 1 in 8 cancer diagnoses worldwide. In 2020, there were about 2.3 million new cases of breast cancer globally and about 685 000 deaths from this disease, with large geographical variations observed between countries and world regions.

Chapter 3 : Prediction Algorithms

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 Mining use different kinds of Machine Learning algorithms for large data sets. But, at a 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 about these two types of learning.

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 “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 :

3.1 Random Forest Classification

3.1.1 What is Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Let's dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most of the people.

Figure 3 : Decision tree Example

3.1.2 How does it work

Before understanding the working of the random forest we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

- **Bagging**– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
- **Boosting**– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy.

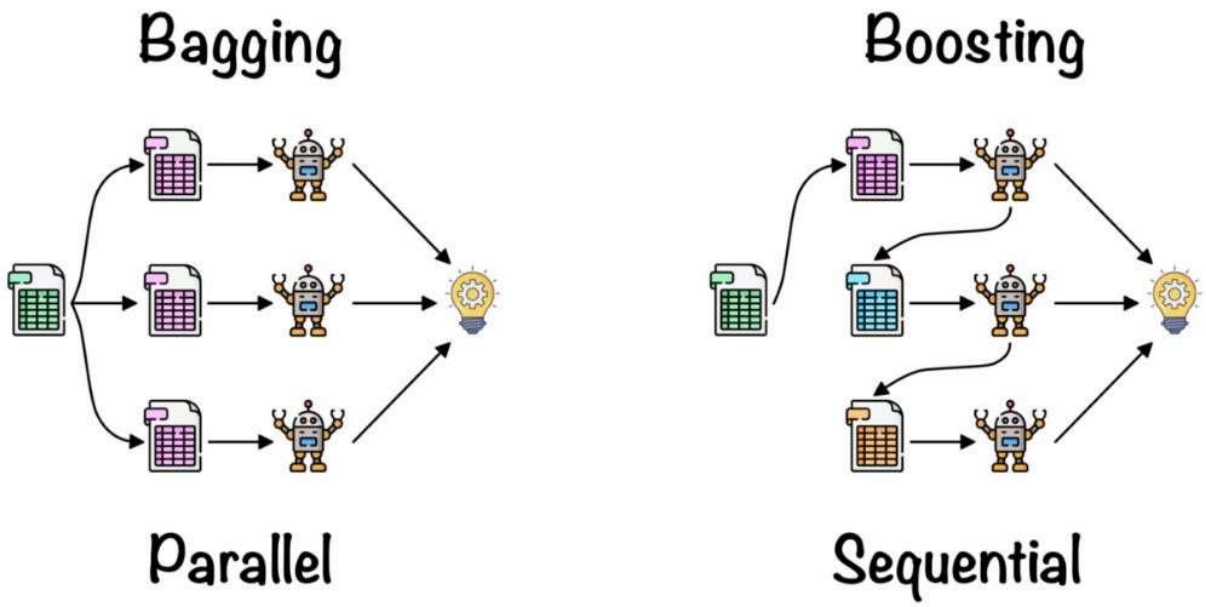


Figure 4 : Bagging & Boosting

As mentioned earlier, Random forest works on the Bagging principle. Now let's dive in and understand bagging in detail.

Bagging, also known as ***Bootstrap Aggregation,*** is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as ***row sampling***. This step of row sampling with replacement is called ***bootstrap***. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as ***aggregation***.

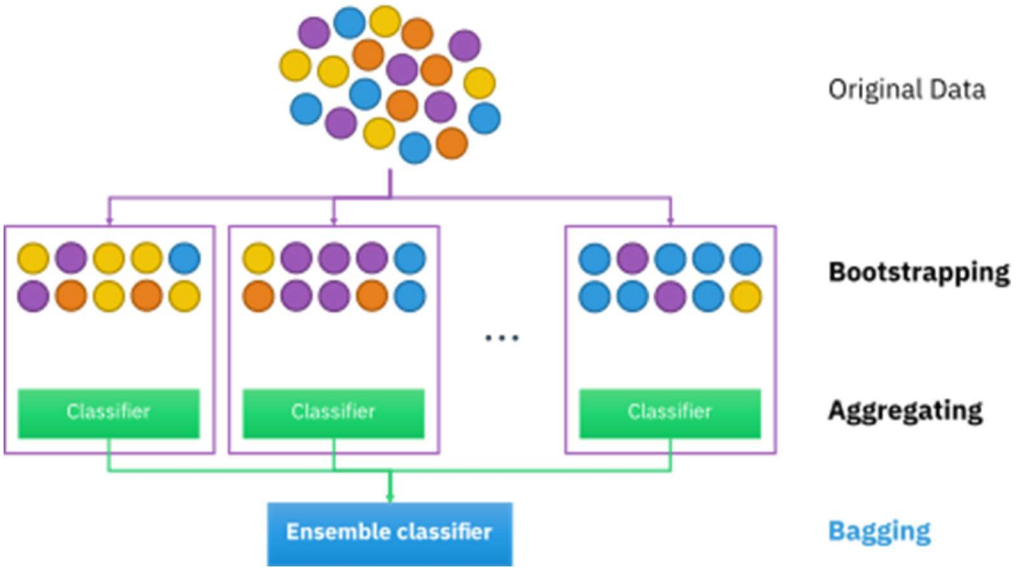


Figure 5 : Random Forest Structure

Now let's look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won't contain unique data. Now the model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown. Now Happy emoji is having a majority when compared to sad emoji. Thus, based on majority voting final output is obtained as Happy emoji.

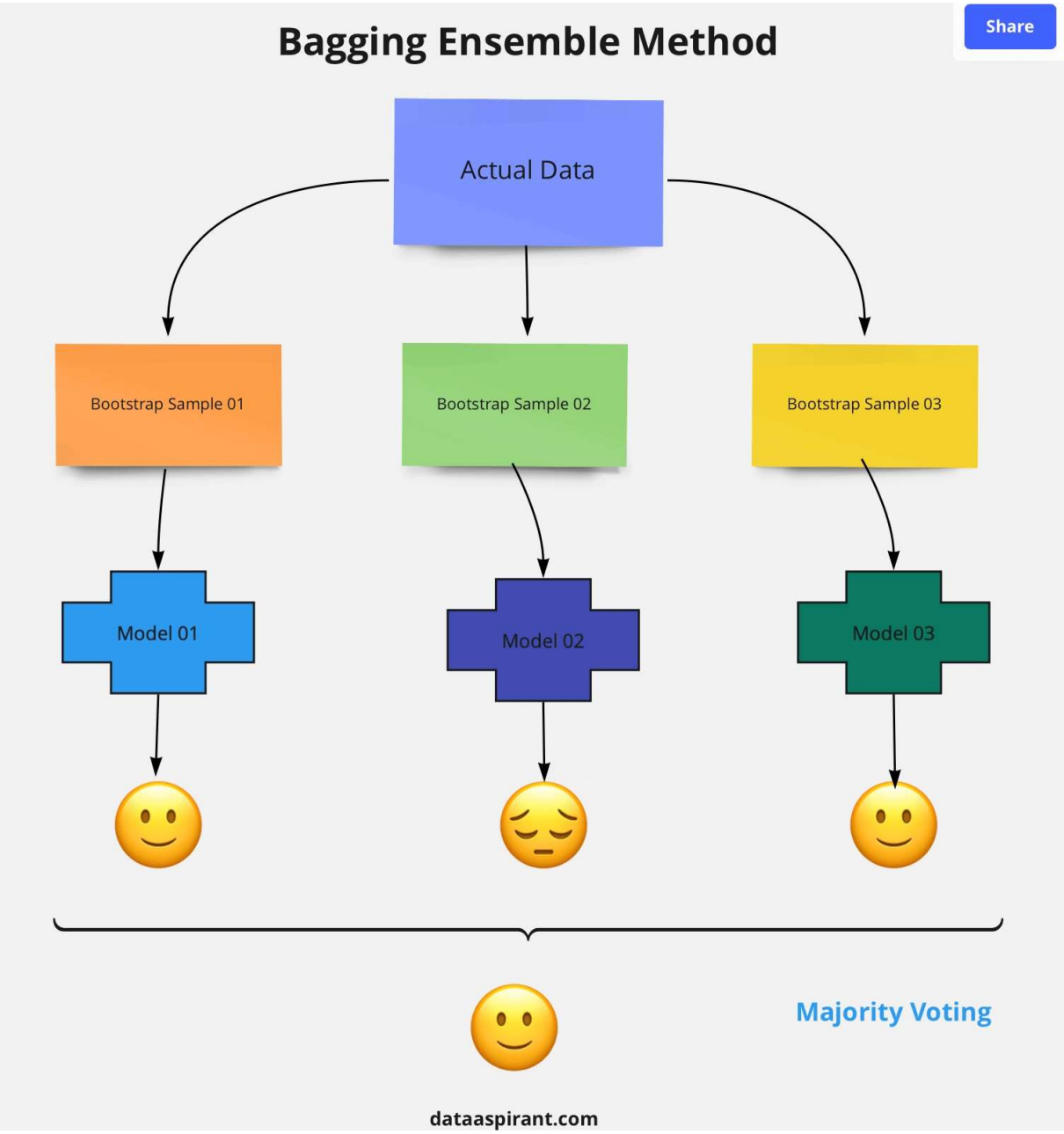


Figure 6 : Bagging Ensemble Method

Steps involved in random forest algorithm:

- In Random forest n number of random records are taken from the data set having k number of records.
- Individual decision trees are constructed for each sample.
- Each decision tree will generate an output.
- Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression respectively.

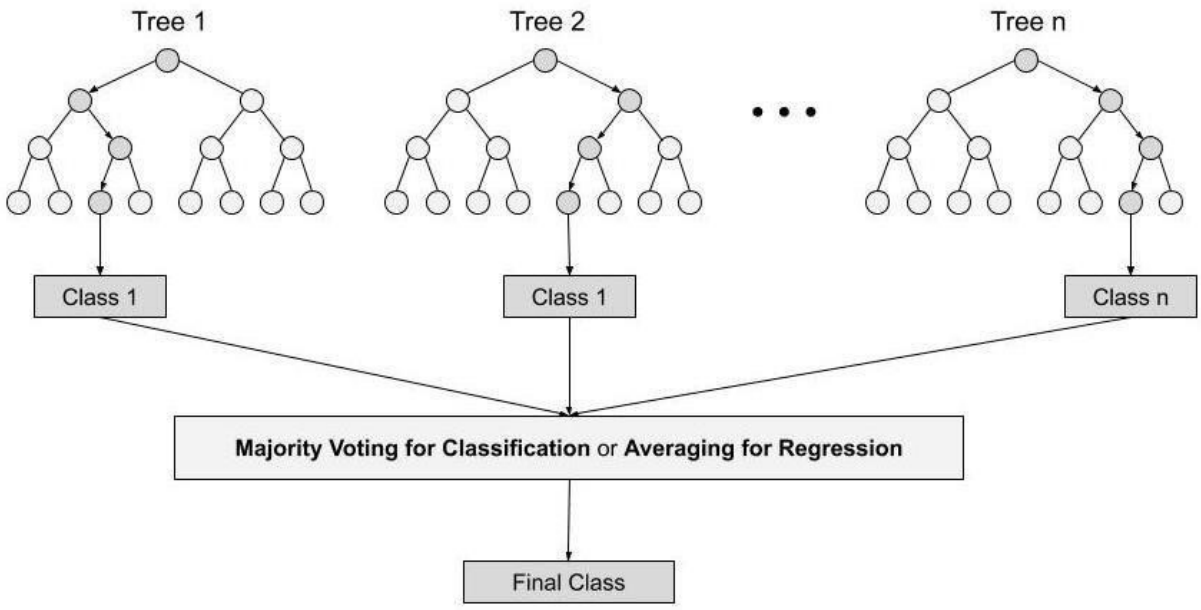


Figure 7 : Majority Voting Based on Trees

For example: consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket and an individual decision tree is constructed for each sample. Each decision tree will generate an output as shown in the figure. The final output is considered based on majority voting. In the below figure you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

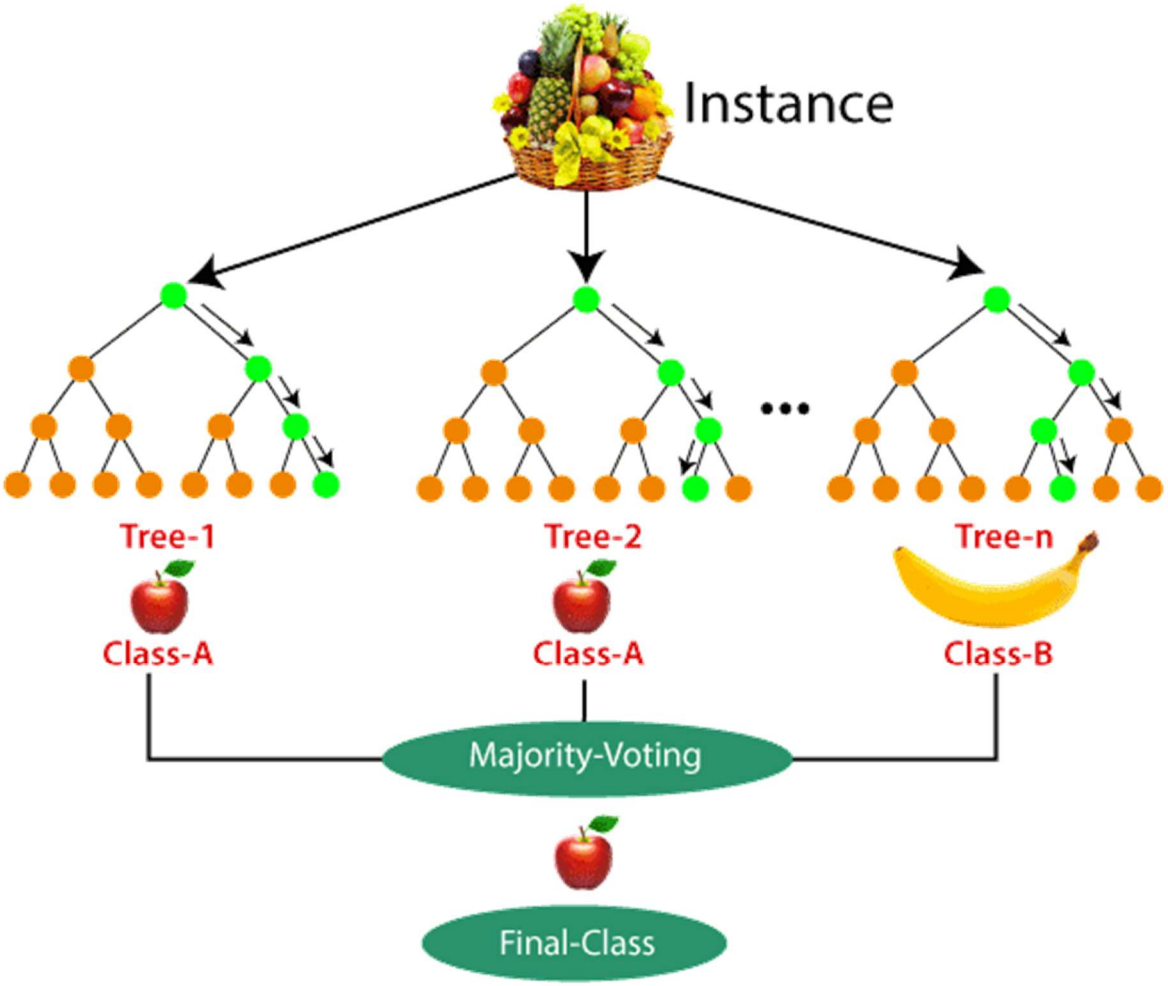


Figure 8 : Random Forest Example

3.1.3 Advantages and Disadvantages of Random Forest Algorithm

3.1.3.1 Advantages

- It can be used in classification and regression problems.
- It solves the problem of overfitting as output is based on majority voting or averaging.
- It performs well even if the data contains null/missing values.
- Each decision tree created is independent of the other thus it shows the property of parallelization.
- It is highly stable as the average answers given by a large number of trees are taken.

- It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.
- It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.
- We don't have to segregate data into train and test as there will always be 30% of the data which is not seen by the decision tree made out of bootstrap.

3.1.3.2 Disadvantages

- Random forest is highly complex when compared to decision trees where decisions can be made by following the path of the tree.
- Training time is more compared to other models due to its complexity. Whenever it has to make a prediction each decision tree has to generate output for the given input data.

3.2 Logistic Regression

3.2.1 What is Logistic Regression

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

3.2.2 How does it work

Before we dive into logistic Regression, let's take a step back and review linear regression.

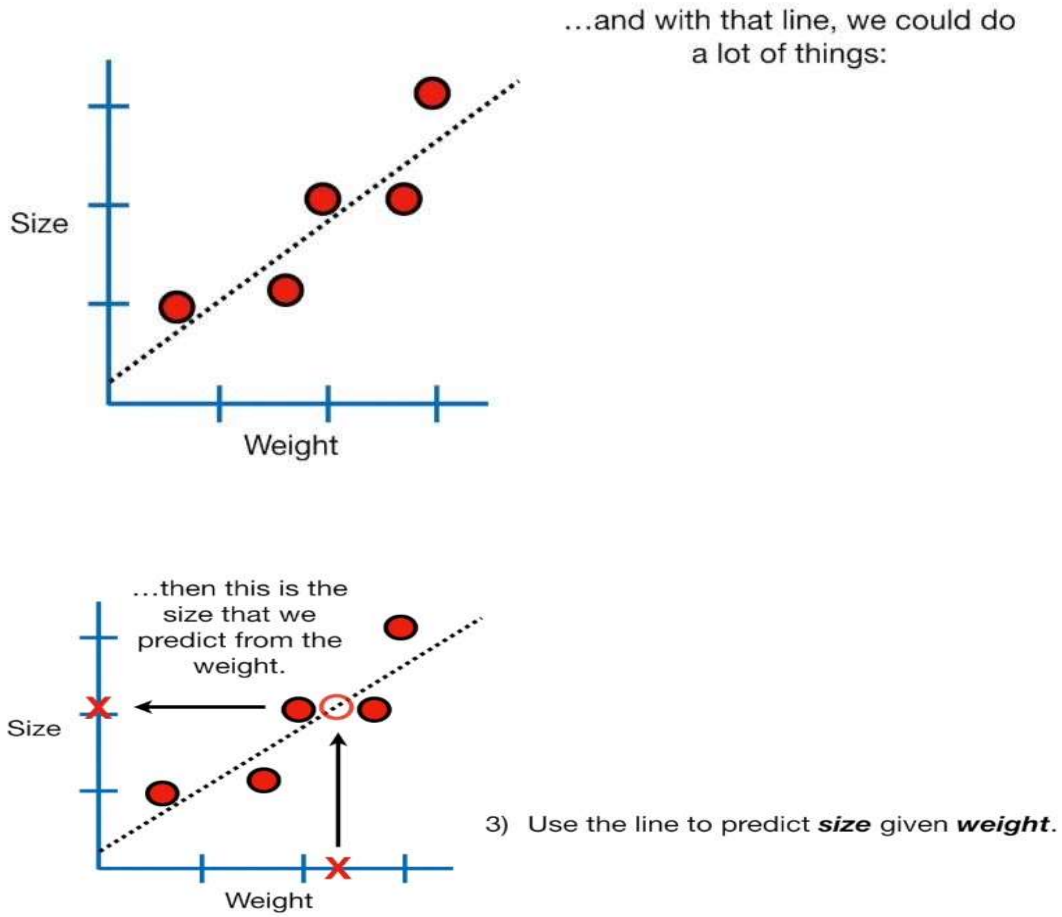
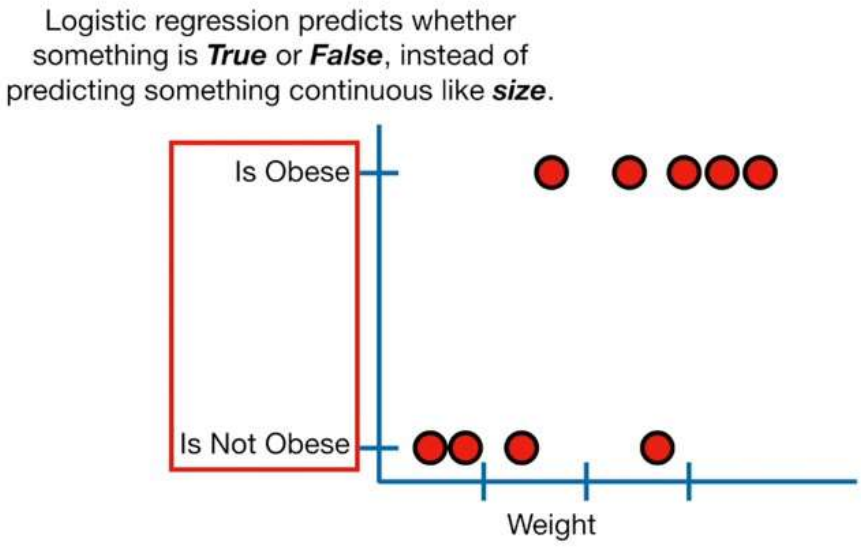


Figure 9 : Linear Regression

Instead of Logistic Regression we can predict binary result.



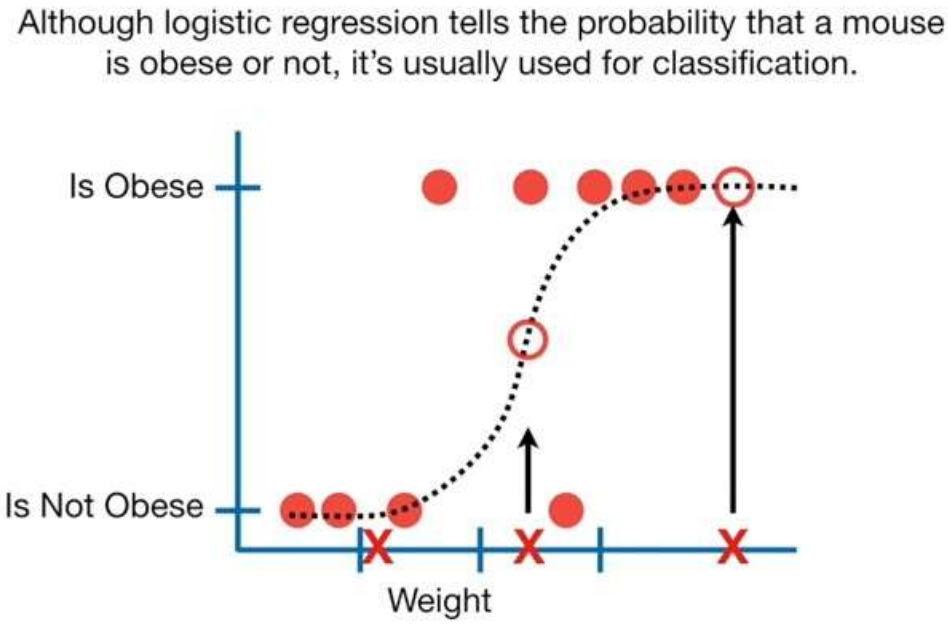


Figure 10 : Logistic Regression

3.2.3 Types of Logistic Regression

There are three main types of logistic regression: binary(wich one we will use in this report), multinomial and ordinal. They differ in execution and theory. Binary regression deals with two possible values, essentially: yes or no. Multinomial logistic regression deals with three or more values. And ordinal logistic regression deals with three or more classes in a predetermined order.

3.2.3.1 Binary Logistic Regression

Binary logistic regression was mentioned earlier in the case of classifying an object as an animal or not an animal—it's an either/or solution. There are just two possible outcome answers. This concept is typically represented as a 0 or a 1 in coding. Examples include:

- Whether or not to lend to a bank customer (outcomes are yes or no).
- Assessing cancer risk (outcomes are high or low).
- Will a team win tomorrow's game (outcomes are yes or no).

3.2.3.2 Multinomial Logistic Regression

Multinomial logistic regression is a model where there are multiple classes that an item can be classified as. There is a set of three or more predefined classes set up prior to running the model. Examples include:

- Classifying texts into what language they come from.
- Predicting whether a student will go to college, trade school or into the workforce.
- Does your cat prefer wet food, dry food or human food?

3.2.3.3 Ordinal Logistic Regression

Ordinal logistic regression is also a model where there are multiple classes that an item can be classified as; however, in this case an ordering of classes is required. Classes do not need to be proportionate. The distance between each class can vary. Examples include:

- Ranking restaurants on a scale of 0 to 5 stars.
- Predicting the podium results of an Olympic event.
- Assessing a choice of candidates, specifically in places that institute ranked-choice voting.

3.3 Support Vector Machine(SVM)

3.3.1 What is SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

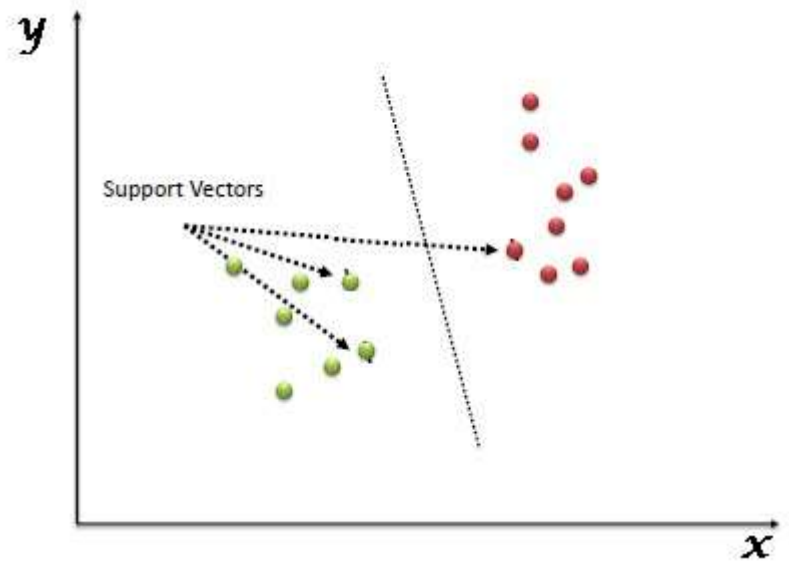


Figure 11 : Hyper-Line

Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line).

3.3.2 How does SVM work

Above, we got accustomed to the process of segregating the two classes with a hyper-plane.

Now the burning question is "How can we identify the right hyper-plane?". Don't worry, it's not as hard as you think!

Let's understand:

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.

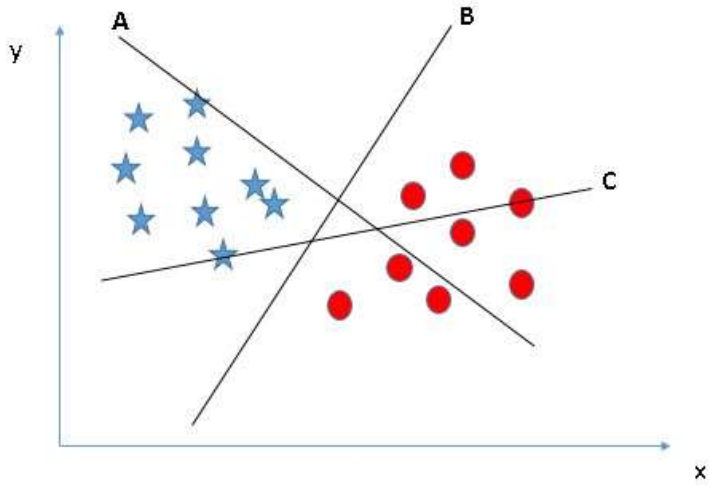


Figure 12 : Hyper-Line Scenario-1

You need to remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.

- **Identify the right hyper-plane (Scenario-2):** Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

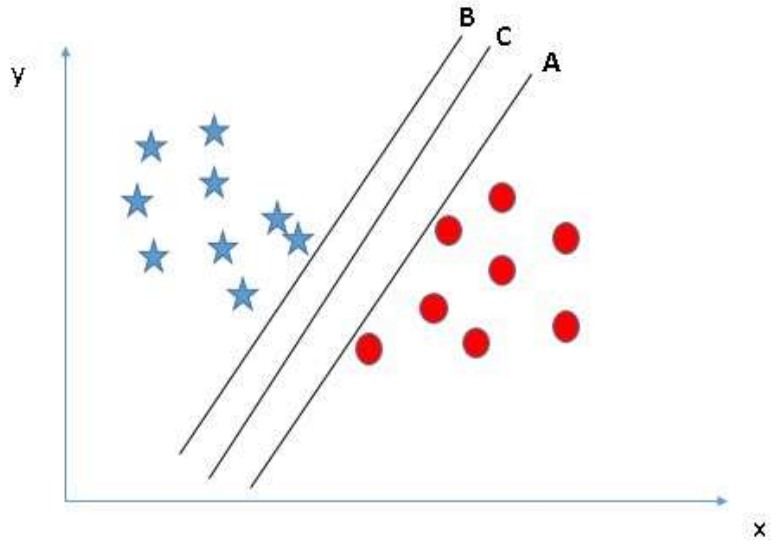


Figure 13 : Hyper-Line Scenario-2

Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let's look at the below snapshot:

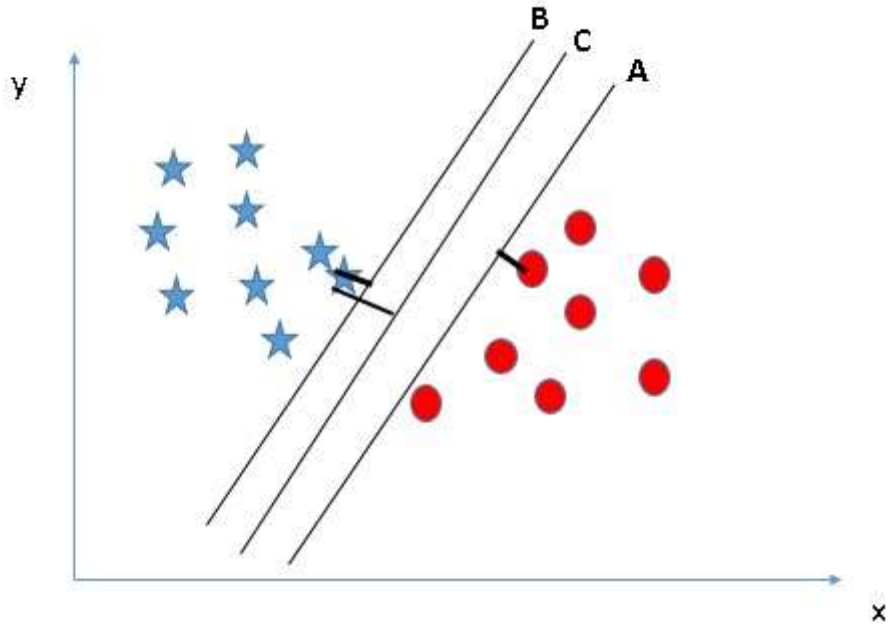


Figure 14 : Hyper-Line Scenario-2(2)

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyperplane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.

- **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane

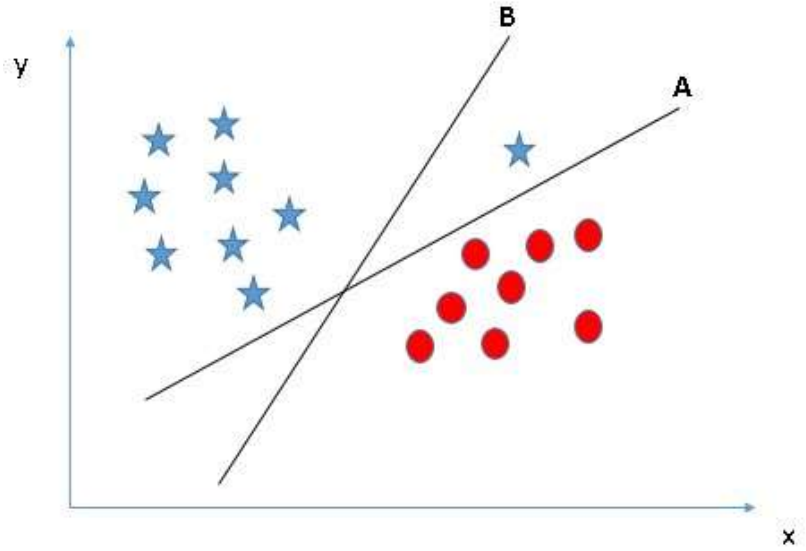


Figure 15 : Hyper-Line Scenario-3

Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

- **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.

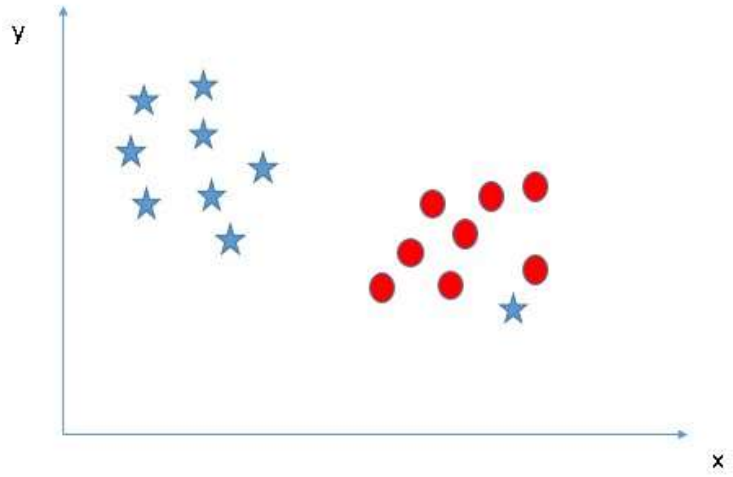


Figure 16 : Hyper-Line Scenario-4

As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.

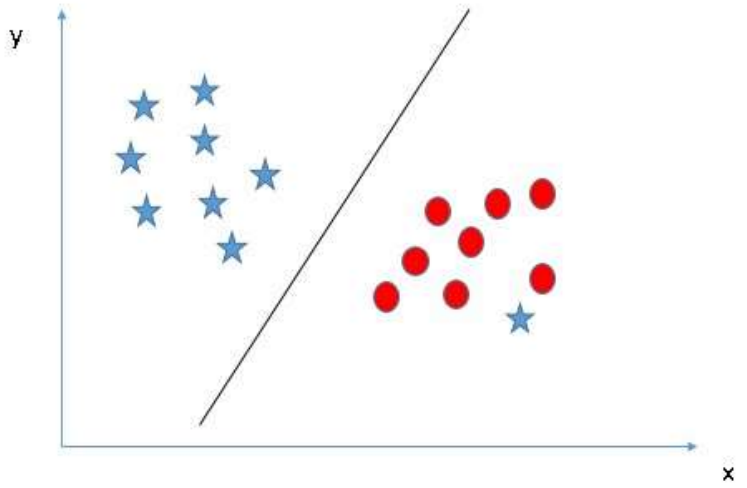


Figure 17 : Hyper-Line Scenario-4(2)

- **Find the hyper-plane to segregate to classes (Scenario-5):** In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

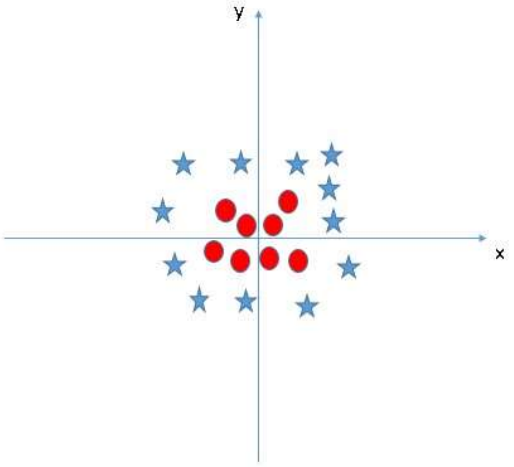


Figure 18 : Hyper-Line Scenario-5

SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:

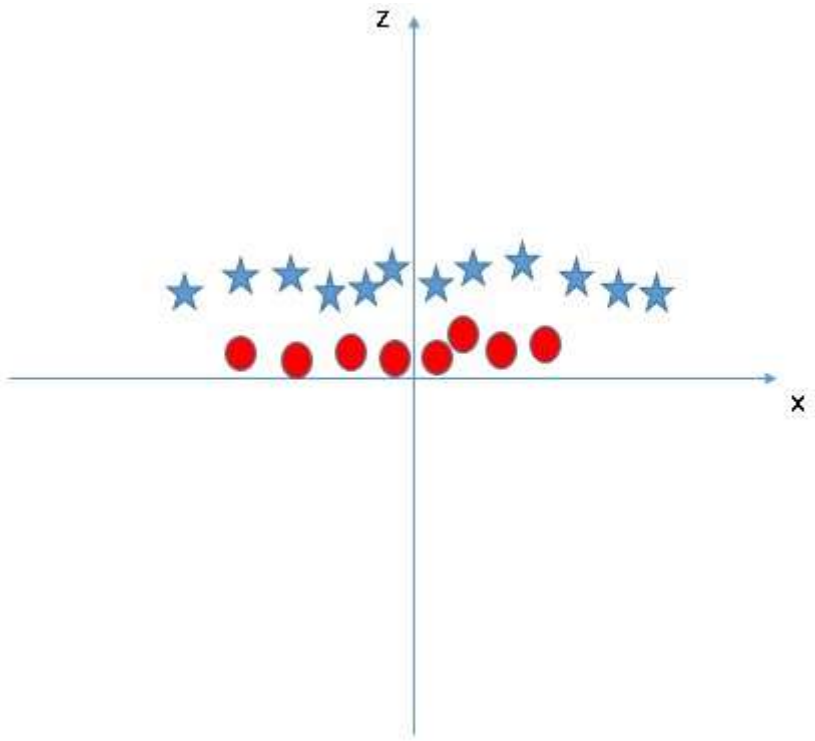


Figure 19 : Hyper-Line Scenario-5(2)

In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the **kernel trick**. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-

linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

When we look at the hyper-plane in original input space it looks like a circle:

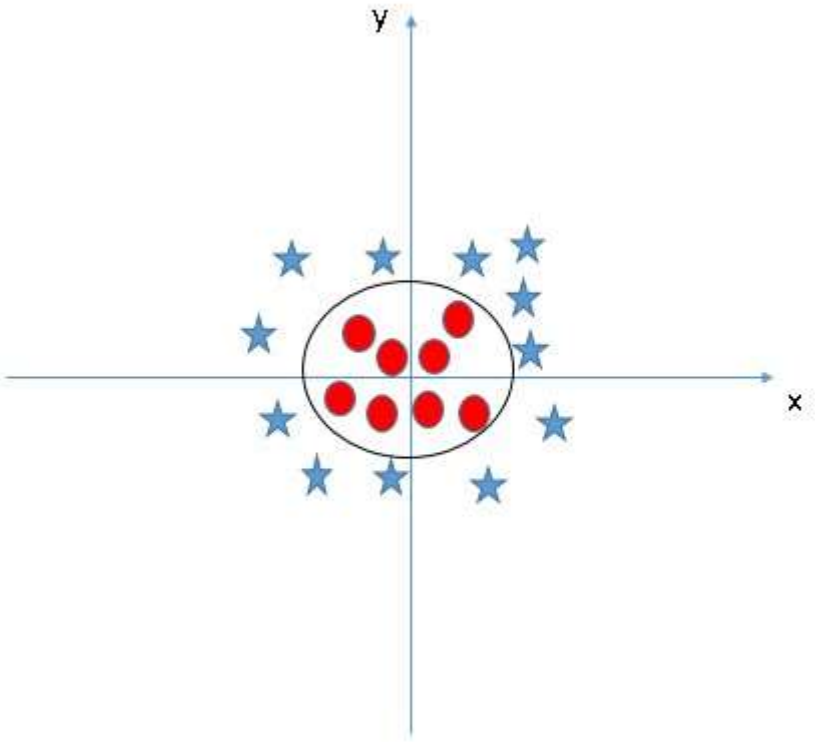


Figure 20 : Hyper-Line Scenario-5(3)

Chapter 4 : Implementation

In this chapter we will explain all implementation steps.

First think it's to import the packages/libraries.

```
In [1]: #import libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Next, I will load the data, and print the first 7 rows of data.

NOTE: Each row of data represents a patient that may or may not have cancer.

```
In [2]: #Load the data
        df = pd.read_csv('data.csv')
        df.head(7)
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089	...	
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.1127	0.07400	...	

7 rows × 33 columns

Explore the data and count the number of rows and columns in the data set. There are 569 rows of data which means there are 569 patients in this data set, and 33 columns which mean there are 33 features or data points for each patient.

```
In [3]: #Count the number of rows and columns in the data set
        df.shape
```

```
Out[3]: (569, 33)
```

Continue exploring the data and get a count of all of the columns that contain empty (NaN, NAN, na) values. Notice none of the columns contain any empty values except the column named 'Unnamed: 32', which contains 569 empty values (the same number of rows in the data set, this tells me this column is completely useless)

```
In [4]: #Count the empty (NaN, NAN, na) values in each column
        df.isna().sum()
```

```
Out[4]: id                         0
        diagnosis                  0
        radius_mean                0
        texture_mean               0
        perimeter_mean             0
        area_mean                  0
        smoothness_mean            0
        compactness_mean           0
        concavity_mean             0
        concave points_mean        0
        symmetry_mean              0
        fractal_dimension_mean     0
        radius_se                  0
        texture_se                 0
        perimeter_se               0
        area_se                    0
        smoothness_se              0
        compactness_se             0
        concavity_se               0
        concave points_se          0
        symmetry_se                0
        fractal_dimension_se       0
        radius_worst               0
        texture_worst              0
        perimeter_worst            0
        area_worst                 0
        smoothness_worst           0
        compactness_worst          0
        concavity_worst            0
        concave points_worst       0
        symmetry_worst             0
        fractal_dimension_worst    0
        Unnamed: 32              569
        dtype: int64
```

Remove the column '*Unnamed: 32*' from the original data set since it adds no value. And **Get the new count** of the number of rows and columns.

```
In [5]: #Drop the column with all missing values (na, NAN, NaN)
        #NOTE: This drops the column Unnamed
        df = df.dropna(axis=1)

In [6]: #Get the new count of the number of rows and cols
        df.shape

Out[6]: (569, 32)
```

Get a count of the number of patients with Malignant (M) cancerous and Benign (B) noncancerous cells.

```
In [7]: #Get a count of the number of 'M' & 'B' cells
        df['diagnosis'].value_counts()

Out[7]: B    357
        M    212
        Name: diagnosis, dtype: int64
```

Encode the categorical data. Change the values in the column 'diagnosis' from M and B to 1 and 0 respectively, then print the results.

1 ----> M

0 ----> B

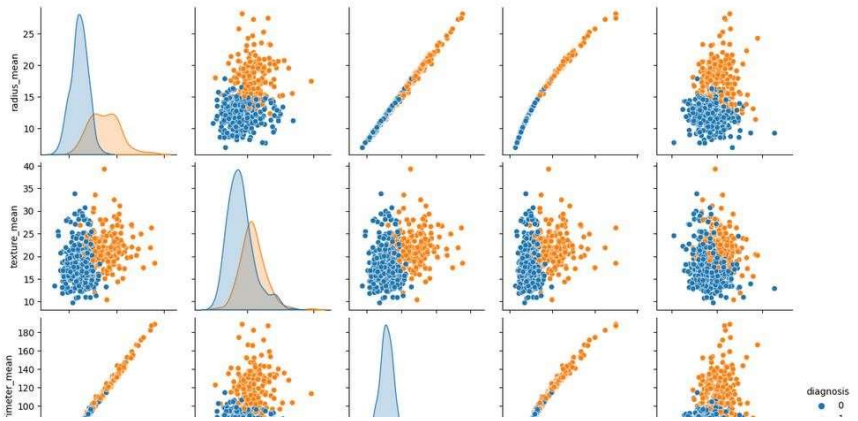
```
In [9]: #Encoding categorical data values
        from sklearn.preprocessing import LabelEncoder
        labelencoder_Y = LabelEncoder()
        df.iloc[:,1]= labelencoder_Y.fit_transform(df.iloc[:,1].values)
        print(labelencoder_Y.fit_transform(df.iloc[:,1].values))
```

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

Create a pair plot. A "pairs plot" is also known as a scatter plot, in which one variable in the same data row is matched with another variable's value.

```
In [14]: sns.pairplot(df.iloc[:,1:7], hue='diagnosis')

Out[14]: <seaborn.axisgrid.PairGrid at 0x7f8001f4ee60>
```



Get the correlation of the columns.

```
In [12]: #Get the correlation of the columns
         df.corr()
```

Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
id	1.000000	0.039769	0.074626	0.099770	0.073159	0.096893	-0.012968	0.000096	0.050080
diagnosis	0.039769	1.000000	0.730029	0.415185	0.742636	0.708984	0.358560	0.596534	0.696360
radius_mean	0.074626	0.730029	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764
texture_mean	0.099770	0.415185	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418
perimeter_mean	0.073159	0.742636	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136
area_mean	0.096893	0.708984	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983
smoothness_mean	-0.012968	0.358560	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984
compactness_mean	0.000096	0.596534	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121
concavity_mean	0.050080	0.696360	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000
concave points_mean	0.044158	0.776614	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391
symmetry_mean	-0.022114	0.330499	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667
fractal_dimension_mean	-0.052511	-0.012838	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783
radius_se	0.143048	0.567134	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925
texture_se	-0.007526	-0.008303	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218
perimeter_se	0.137331	0.556141	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391
area_se	0.177742	0.548236	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427
smoothness_se	0.096781	-0.067016	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564

Visualize the correlation by creating a heat map.

```
In [13]: plt.figure(figsize=(20,20))
         sns.heatmap(df.corr(), annot=True, fmt='.0%')
```

Out[13]: <AxesSubplot: >

Now I am done exploring and cleaning the data. I will set up my data for the model by first splitting the data set into a feature data set also known as the independent data set (X), and a target data set also known as the dependent data set (Y)

```
In [15]: X = df.iloc[:, 2:31].values
         Y = df.iloc[:, 1].values
```

Split the data again, but this time into 75% training and 25% testing data sets.

```
In [16]: 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)
```

Create a function to hold many different models (e.g. Logistic Regression, Support Vector Machines, Random Forest) to make the classification. These are the models that will detect if a

patient has cancer or not. Within this function I will also print the accuracy of each model on the training data.

```
In [19]: def models(X_train,Y_train):

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

           #Using SVM
           from sklearn.svm import SVC
           svc_rbf = SVC(kernel = 'rbf', random_state = 0)
           svc_rbf.fit(X_train, Y_train)

           #Using RandomForestClassifier method of ensemble class to use Random Forest Classification algorithm
           from sklearn.ensemble import RandomForestClassifier
           forest = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
           forest.fit(X_train, Y_train)

           #print model accuracy on the training data.
           print('[0]Logistic Regression Training Accuracy:', log.score(X_train, Y_train))
           print('[1]Support Vector Machine (RBF Classifier) Training Accuracy:', svc_rbf.score(X_train, Y_train))
           print('[2]Random Forest Classifier Training Accuracy:', forest.score(X_train, Y_train))

           return log, svc_rbf, forest
```

Create the model that contains all of the models, and look at the accuracy score on the training data for each model to classify if a patient has cancer or not.

```
In [20]: model = models(X_train,Y_train)

         [0]Logistic Regression Training Accuracy: 0.9483568075117371
         [1]Support Vector Machine (RBF Classifier) Training Accuracy: 0.903755868544601
         [2]Random Forest Classifier Training Accuracy: 0.9953051643192489
```

Show the confusion matrix and the accuracy of the models on the test data. The confusion matrix tells us how many patients each model misdiagnosed (number of patients with cancer that were misdiagnosed as not having cancer also known as false negative, and the number of patients who did not have cancer that were misdiagnosed with having cancer also known as false positive) and the number of correct diagnosis, the true positives and true negatives.

In [22]:
```
#Show other ways to get the classification accuracy & other metrics

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

for i in range(len(model)):
  print('Model ',i)
  #Check precision, recall, f1-score
  print( classification_report(Y_test, model[i].predict(X_test)) )
  #Another way to get the models accuracy on the test data
  print( accuracy_score(Y_test, model[i].predict(X_test)))
  print()#Print a new line
```

```
Model  0
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        90
           1       0.89      0.96      0.93        53

    accuracy                           0.94       143
   macro avg       0.94      0.95      0.94       143
weighted avg       0.95      0.94      0.94       143

0.9440559440559441

Model  1
              precision    recall  f1-score   support


           0       0.92      0.99      0.95        90
           1       0.98      0.85      0.91        53

    accuracy                           0.94       143
   macro avg       0.95      0.92      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
```

From the accuracy and metrics above, the model that performed the best on the test data was the Random Forest Classifier with an accuracy score of about 96.5%. So I will choose that model to detect cancer cells in patients. Make the prediction/classification on the test data and show both the Random Forest Classifier model classification/prediction and the actual values of the patient that shows rather or not they have cancer.

I notice the model, misdiagnosed a few patients as having cancer when they didn't and it misdiagnosed patients that did have cancer as not having cancer. Although this model is good, when dealing with the lives of others I want this model to be better and get it's accuracy as close to 100% as possible or at least as good as if not better than doctors. So a little more tuning of each of the models is necessary.

Conclusion

Breast Cancer represents one of the diseases that makes highest number of deaths every year. At present, only few accurate prognostic and predictive factors are used clinically for managing the patients with breast cancer.

In this report we have worked to collect the suitable dataset needed to help in this predictive analysis. This dataset is then processed to remove all the junk data. The predictive analysis method is being used in many different fields and is slowly picking up pace. It is helping us by using smarter ways to solve or predict a problem's outcome.

Our scheme was developed to reduce the time and cost factors of the patients as well as to minimize the work of a doctor. We have tried to use a very simple and understandable model to do this job. Next, machine learning algorithms should be used on the training data and the testing data should be used to check if the outcomes are accurate enough.

In the future, we can also use a dataset to predict breast cancer based on Mammography or MRI pictures. Artificial Neural Networks in this case can be applied to make the prediction better and smarter. Accuracy can be increased by selecting better features.

References

Reports:

1. https://github.com/gmineo/Breast-Cancer-PredictionProject/blob/master/Report.pdf
2. https://scholarworks.calstate.edu/downloads/tx31qq02n

Web Site:

1. https://fr.wikipedia.org/wiki/Wikipédia
2. https://www.iarc.who.int/news-events/current-and-future-burden-ofbreast-cancer-global-statistics-for-2020-and-2040/
3. https://www.cancer.org/cancer/breast-cancer
4. https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data/