

VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
Jnana Sangama, Belagavi - 590 018



PROJECT REPORT ON  
**BREAST CANCER DETECTION USING  
SUPPORT VECTOR MACHINE**

*Thesis submitted in partial fulfillment for the Award of Degree of*  
**Bachelor of Engineering**  
in  
**Electronics and Communication Engineering**

Submitted by

JAYPREET SINGH  
KAUSHIK G

1RN16EC045  
1RN15EC055

*Under the Guidance of*  
**Prakash Tunga P**  
*Assistant Professor*



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING  
(Accredited by NBA for the Academic years 2018-19, 2019-20 and 2020-21)

**RNS INSTITUTE OF TECHNOLOGY**  
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)  
(UG Programs - CSE, ECE, ISE, EIE and EEE have been Accredited by NBA  
for the Academic years 2018-19, 2019-20 and 2020-21)  
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

2020 - 21

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**Jnana Sangama, Belagavi - 590 018**



**PROJECT REPORT ON**  
**BREAST CANCER DETECTION USING SUPPORT**  
**VECTOR MACHINE**

*Thesis submitted in partial fulfillment for the Award of Degree of*  
**Bachelor of Engineering**

in  
**Electronics and Communication Engineering**

**Submitted by**

JAYPREET SINGH  
KAUSHIK G

1RN16EC045  
1RN15EC055

*Under the Guidance of*  
**Prakash Tunga P**  
*Assistant Professor*



ESTD 2001

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**  
**(Accredited by NBA for the Academic years 2018-19, 2019-20 and 2020-21)**

**RNS INSTITUTE OF TECHNOLOGY**  
**(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)**  
**(UG Programs - CSE, ECE, ISE, EIE and EEE have been Accredited by NBA**  
**for the Academic years 2018-19, 2019-20 and 2020-2021)**  
**Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098**

**2020 - 21**

# RNS INSTITUTE OF TECHNOLOGY

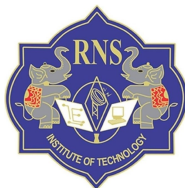
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

(UG Programs - CSE, ECE, ISE, EIE and EEE have been Accredited by NBA  
for the Academic years 2018-19, 2019-20 and 2020-21)

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Accredited by NBA for the Academic years 2018-19, 2019-20 and 2020-21)



ESTD 2001

## CERTIFICATE

This is to Certify that the project entitled “**BREAST CANCER DETECTION USING SUPPORT VECTOR MACHINE**” is carried out by **JAYPREET SINGH (1RN16EC045)**, **KAUSHIK G (1RN15EC055)**, in partial fulfillment for the award of degree of Bachelor of Engineering in **Electronics and Communication Engineering** of Visvesvaraya Technological University, Belagavi, during the year 2020-2021. It is certified that all corrections / suggestions indicated during internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in aspect of the project work prescribed for the award of degree of **Bachelor of Engineering**.

Prakash Tunga P

Assistant Professor

Dr. Vipula Singh

Head of the Department

Dr. M K Venkatesha

Principal

**External Viva**

Name of the examiners

Signature with date

1 .....

.....

2 .....

.....

# RNS INSTITUTE OF TECHNOLOGY

(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

(UG Programs - CSE, ECE, ISE, EIE and EEE have been Accredited by NBA  
for the Academic years 2018-19, 2019-20 and 2020-2021)

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

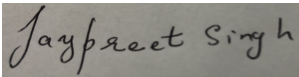
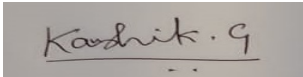
(Accredited by NBA for the Academic years 2018-19, 2019-20 and 2020-21)



ESTD 2001

## DECLARATION

We hereby declare that the entire work embodied in this project report titled, **“BREAST CANCER DETECTION USING SUPPORT VECTOR MACHINE ”** submitted to **Visvesvaraya Technological University**, Belagavi, is carried out at the department of **Electronics and Communication Engineering, RNS Institue of Technology, Bengaluru** under the guidance of **Prakash Tunga P**, Assistant Professor. This report has not been submitted for the award of any Diploma or Degree of this or any other University.

Name	USN	Signature
1. JAYPREET SINGH	1RN16EC045	
2. KAUSHIK G	1RN15EC055	

# Acknowledgement

The joy and satisfaction that accompany the successful completion of any task would be incomplete without thanking those who made it possible. We consider ourselves proud to be a part of RNS Institute of Technology, the institution which moulded us in all our endeavors.

We express our gratitude to **Late Dr.R N Shetty**, our beloved chairman for providing state of the art facilities.

We would like to express our sincere thanks to **Dr. M K Venkatesha**, Principal and **Dr. Vipula Singh**, Professor and HOD, Department of ECE, for their valuable guidance and encouragement throughout our program.

We express our profound gratitude to the coordinators who have given valuable suggestions and guidance throughout the project. We would like to express our sincere gratitude to our guide **Prakash Tunga P**, Assistant Professor, for her/his guidance, continuous support and motivation in completing the project successfully.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, teaching and non-teaching staff of the department, the library staff and all our friends who have directly or indirectly supported us.

**JAYPREET SINGH**

**KAUSHIK G**

# Abstract

According to World Health Organization (WHO), Breast cancer is one of the most common cancer among the women and it is the second dangerous cancer after lung cancer. From the research, it is estimated that almost 15 percent of all cancer death among women is due to breast cancer. In case of any system people visit to oncologist to gather the details of cancer, doctor use various scanning techniques like breast ultrasound, diagnostic mammogram, Magnetic resonance imaging (MRI), biopsy etc. Based on these test results, doctor recommend further tests or therapies. If the cancer cells are predicted at early stage then survivability chances of the patient increases. An alternate way to identify breast cancer is by using machine learning algorithms. These algorithms are fast in producing results and can be a part of test routine. The research is carried out for the proper diagnosis and categorization of patients into malignant and benign groups.

# Table of Contents

<b>Abstract</b>	ii
<b>Table of Contents</b>	iii
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Objectives . . . . .	2
1.3 Methodology . . . . .	2
1.4 Applications . . . . .	3
1.5 Advantages and Disadvantages . . . . .	3
1.6 Organisation of the Report . . . . .	4
<b>2 LITERATURE SURVEY</b>	<b>5</b>
<b>3 SYSTEM REQUIREMENTS</b>	<b>6</b>
3.1 Software Requirement Specification . . . . .	6
3.2 Functionality . . . . .	6
3.3 Requirements . . . . .	6
3.4 Hardware Requirements . . . . .	6
3.5 Software Requirements . . . . .	7
3.6 High Level design . . . . .	7
3.7 Architecture of IPython Notebook . . . . .	7
3.8 Algorithms . . . . .	7
3.9 Classification in SVM . . . . .	8
3.10 Unbalanced problems . . . . .	9
3.11 Regression . . . . .	10
3.12 Complexity . . . . .	11
3.13 Types of SVM . . . . .	11
3.14 How an SVM works . . . . .	11
3.15 Some applications of SVM . . . . .	14
3.16 SVM Advantages and Disadvantages . . . . .	14
3.17 Python Libraries . . . . .	15
3.18 Numpy . . . . .	15
3.19 Sklearn . . . . .	16
3.20 Pandas . . . . .	17

3.21 Matplotlib . . . . .	19
3.22 Seaborn . . . . .	19
3.23 seaborn countplot . . . . .	21
3.24 Scatter plot . . . . .	22
<b>4 SYSTEM IMPLEMENTATION</b>	<b>23</b>
4.1 Python . . . . .	23
4.2 Integrated Development Environment . . . . .	25
4.3 Programming Coding Guidelines . . . . .	25
4.4 Methodology of the project . . . . .	26
4.5 Data Collection and Exploration . . . . .	26
4.6 Data Cleaning . . . . .	26
4.7 Data Pre-processing . . . . .	27
4.8 Model Evaluation . . . . .	28
4.9 Flow Chart . . . . .	28
4.10 Summary of flowchart . . . . .	29
<b>5 RESULTS AND DISCUSSION</b>	<b>30</b>
5.1 Dataset . . . . .	30
5.2 Accuracy of Model . . . . .	31
5.3 Input patient new data in the program . . . . .	31
5.4 Results . . . . .	31
<b>6 CONCLUSION AND FUTURE WORK</b>	<b>33</b>
6.1 Conclusion . . . . .	33
6.2 Future Work . . . . .	33
<b>References</b>	<b>34</b>



# Chapter 1

## INTRODUCTION

Breast cancer 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 modeling

### 1.1 Motivation

In 2020, there were 2.3 million women diagnosed with breast cancer and 685000 deaths globally. As of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years, making it the world's most prevalent cancer. It was the most common cancer in women worldwide, contributing 25.4 percent of the total number of new cases diagnosed in 2018.

In India, breast cancer has surpassed cancers of the cervix and the oral cavity to be the most common cancer and the leading cause of cancer deaths. In 2018, 162,468 new cases of breast cancer were diagnosed, representing 27.7 percent of all new cancers among Indian women and 11.1 percent of all cancer deaths.

This machine learning model helps to detect cancer in its early stages. Even the most experienced practitioner can predict this cancer with an accuracy of not more than 74 percent. The accuracy of this model is more than 95 percent, which makes it an excellent tool for the detection of the cancerous cells.

This model also helps in early detection of the cancerous cells which will help to reduce the fatalities caused by this cancer.

## 1.2 Objectives

1. 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 hyper parameter selection.
2. To classify whether the breast cancer is benign or malignant.
3. To achieve this we have used machine learning classification methods to fit a function that can predict the discrete class of new input.

## 1.3 Methodology

Methods are divided into four steps:

1. Linear Regression: This is a machine learning algorithm which is based on supervised learning. This regression performs the task to predict a dependent variable value based on given independent variable. So this regression technique finds out a linear relationship between input and output. The following steps have been used for generating linear regression model.
  - a) Import package and class: The first step is to import the package numpy and the class Pipeline from sklearn.pipeline, class, Linear regression from sklearn.linear\_model and the class Standard Scaler from sklearn.preprocessing. Now we have all the functionalities that we need to implement linear regression model.
  - b) Provide data and create a model: The data is then provided and eventually the appropriate transformations are done. Then the regression model is created by using existing data.
  - c) Checking the results and predicting the response: The model is then fitted. After this we'll get the results to check whether the model works satisfactorily. Once there is a satisfactory model, it will be then used for predictions with either existing or new data.
2. Polynomial Regression: This is a form of regression analysis in which relationship between the independent variable  $x$  and the dependent variable  $y$  is modeled as an  $n$ th degree polynomial. Some polynomial terms will be added to the Multiple Linear Regression equation to convert it into Polynomial regression. This is a linear model with some modification in order to increase the accuracy. The dataset used in Polynomial regression for training is of non-linear in nature. It makes use of Linear Regression model to fit the complicated and non-linear functions and dataset. The original features are converted into polynomial features

of degree 2 and is then modeled using a linear model. The main steps involved in polynomial regression are given below.

3. Data pre-processing and building a Polynomial regression model: In data pre-processing the data gets encoded so that it can be brought to such a state that now the machine can easily parse it. The features of data can now be easily interpreted by the algorithm. Then the linear regression model is built and fitted to the dataset. In building polynomial regression model, linear regression model is taken as reference. Once the polynomial regression model is built, it will be different from the simple linear model. Here we are using Polynomial Features class of pre-processing libraries.
4. Visualizing the result and predicting the output: The result of polynomial regression model is then visualized and then the final result with the polynomial regression model is then predicted and compared with Linear Model.

## 1.4 Applications

1. Can be used in medical automation.
2. Detection of cancerous cells.
3. Early diagnosis of cancer.

## 1.5 Advantages and Disadvantages

### Advantages

1. Flexible service architecture for future extension
2. Very fast
3. Available 24x7
4. Higher Accuracy

### Disadvantages

1. It is restricted to one output.
2. It doesn't provide the stage of cancer.
3. The parameter generation which are used as an input is a complicated task.
4. It does not provide information for any other type of cancer.

## 1.6 Organisation of the Report

1. Introduction: This chapter just provides the introduction to our project and discusses the motivation and the different objectives of our project and just gives a glimpse of the methodology we used and what were the challenges that we faced and the solution to those challenges.
2. Literature Survey: This chapter gives a brief summary of the paper we used for reference and explains the methodology behind it and how it contributed to making of our project and the ideas we took from the different paper.
3. System Analysis: This chapter gives us the insight of the technical details of our project such as the software requirement specification, high level design.
4. System Implementation: This chapter elaborates the whole process and the methodology behind the implementation of the project in a step by step process and provides with the insight of programming coding guidelines used during the making of our project.
5. Results And Discussion: In this chapter we we discuss the outputs rather the results obtained after completion of our project with the efficiency of the model or the accuracy of making correct predictions by our model.
6. Conclusion and Future Work: In this chapter we discuss about how the project has been represented and to show the work carried out in building a system model. It also talks about the other enhancements that can be made into the project so that it is user friendly and make more accurate predictions.

# Chapter 2

## LITERATURE SURVEY

The methodology used for breast cancer detection has been discussed. The method we used is the support vector machine classifier (SVM). Also had many merits and demerits of SVM classifier. We also discussed the merits and demerits of svm classifier. The demerits being that it is not suitable for large data set. It requires more memory. The merits being that it is more efficient. It can be used for classification and regression problems and solutions. Here we also come across that svm classifier has more advantages than other classifiers as compared to other classifier such as random forest, decision tree etc. These were the main points we discussed.

More on the support vector machine classifier has been discussed. It is told that more demerits were in svm choosing an appropriate kernel function is difficult. It requires long training time so it is very difficult to train the data set. Few more merits are svm is used to handle high dimensional data. It has better accuracy compared to other algorithms such as random forest, linear regression, decision tree etc. Hence this classifier is used more nowadays, it gives more accuracy and good result hence we use this classifier in all fields such as medical, engineering, banking etc. These were the main points we discussed.

We see how to diagnose breast cancer using different classifiers. Here, the dataset is split into test dataset and train dataset. After this we consider various algorithm such as decision tree, naïve bayes, support vector machine, random forest, k nearest neighbor etc for classification. Then the algorithm which has better accuracy is found. It is concluded that SVM classifier performs better as compared to other classifiers and significantly helps in finding out the patient has cancer or not. The classifier has a good functioning with respect to other classifier. It also gives better performance and diagnosis and comparatively better result compared to other classifier. Hence we can choose this classifier to solve our machine learning problems compared to other classifiers.

# Chapter 3

## SYSTEM REQUIREMENTS

### 3.1 Software Requirement Specification

A Software Requirements Specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

### 3.2 Functionality

Various steps are as follows:

1. Data Collection
2. Data Cleaning
3. Data Pre-processing
4. Model Development
5. Model Evaluation

### 3.3 Requirements

The PCs used must be at least be INTEL CORE i3 machines so that they can give optimum performance of the product. In addition to these requirements, the system should also embrace the following requirements.

### 3.4 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

1. Processor : Pentium IV processor
2. Processor Speed: 2.4 GHz
3. RAM: 1 GB
4. Storage Space: 3.8 GB

## 3.5 Software Requirements

Operating System:

- Windows 10
- Web browser
- Anaconda Navigator
- Jupyter Notebook
- Python3
- Libraries: numpy, pandas, sklearn, Matplot, seaborn, Scatterplot

## 3.6 High Level design

High level design explains the architecture that would be used for developing software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

## 3.7 Architecture of IPython Notebook

The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process developing, documenting, and executing code, as well as communicating the results. The Jupiter notebook combines two components.

- web application: A browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.
- Notebook documents: A representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

## 3.8 Algorithms

SVM

Support vector machines are a set of supervised learning methods used for classification, regression, and outliers detection. All of these are common tasks in machine

learning. We can use them to detect cancerous cells based on millions of images or you can use them to predict future driving routes with a well-fitted regression model.

There are specific types of SVMs you can use for particular machine learning problems, like support vector regression (SVR) which is an extension of support vector classification (SVC).

The main thing to keep in mind here is that these are just math equations tuned to give you the most accurate answer possible as quickly as possible.

SVMs are different from other classification algorithms because of the way they choose the decision boundary that maximizes the distance from the nearest data points of all the classes. The decision boundary created by SVMs is called the maximum margin classifier or the maximum margin hyper plane.

## 3.9 Classification in SVM

SVC, NuSVC and LinearSVC are classes capable of performing binary and multi-class classification on a dataset.

SVC and NuSVC are similar methods, but accept slightly different sets of parameters and have different mathematical formulations. On the other hand, LinearSVC is another (faster) implementation of Support Vector Classification for the case of a linear kernel. LinearSVC does not accept parameter kernel, as this is assumed to be linear. It also lacks some of the attributes of SVC and NuSVC, like support.

As other classifiers, SVC, NuSVC and LinearSVC take as input two arrays: an array X of shape (n samples, n features) holding the training samples, and an array y of class labels (strings or integers), of shape (n samples).

### 3.9.1 Multi-class-classification

SVC and NuSVC implement the “one-versus-one” approach for multi-class classification. In total,  $n \text{ classes} * (n \text{ classes} - 1) / 2$  classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision function shape option allows to monotonically transform the results of the “one-versus-one” classifiers to a “one-vs-rest” decision function of shape (n samples, n classes).

### 3.9.2 Score and probabilities

The decision function method of SVC and NuSVC gives per-class scores for each sample (or a single score per sample in the binary case). When the constructor option



probability is set to True, class membership probability are enabled. In the binary case, the probabilities are calibrated using Platt scaling logistic regression on the SVM's scores, fit by an additional cross-validation on the training data.

The cross-validation involved in Platt scaling is an expensive operation for large datasets. In addition, the probability estimates may be inconsistent with the scores:

1. The “argmax” of the scores may not be the argmax of the probabilities.
2. In binary classification, a sample may be labeled by predict as belonging to the positive class even if the output of predict\_proba is less than 0.5; and similarly, it could be labeled as negative even if the output of predict\_proba is more than 0.5.

Platt's method is also known to have theoretical issues. If confidence scores are required, but these do not have to be probabilities, then it is advisable to set probability=False and use decision function instead of predict\_proba.

### 3.10 Unbalanced problems

In problems where it is desired to give more importance to certain classes or certain individual samples, the parameters class weight and sample weight can be used.

SVC (but not NuSVC) implements the parameter class weight in the fit method. It's a dictionary of the form class label value, where value is a floating point number greater than 0 that sets the parameter C of class class label to C value. The figure below illustrates the decision boundary of an unbalanced problem, with and without weight correction. The below figure 3.1 shows unbalanced problems.

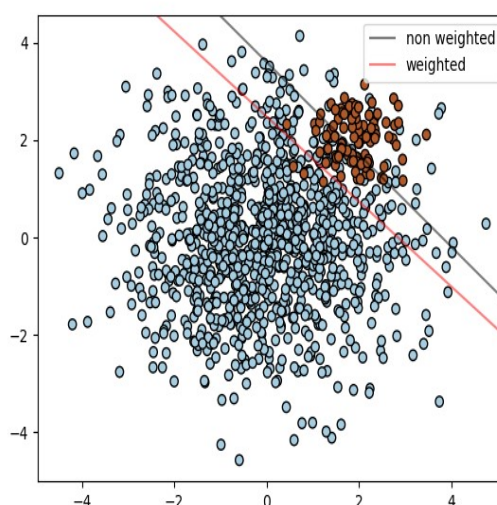


Figure 3.1: Unbalanced problems

SVC, NuSVC, SVR, NuSVR, LinearSVC, LinearSVR and OneClassSVM implement also weights for individual samples in the fit method through the sample weight parameter. Similar to class weight, this sets the parameter  $C$  for the  $i$ -th example to  $C \text{ sample\_weight}[i]$ , which will encourage the classifier to get these samples right. The figure below illustrates the effect of sample weighting on the decision boundary. The figure 3.2 shows effect of sample weight.

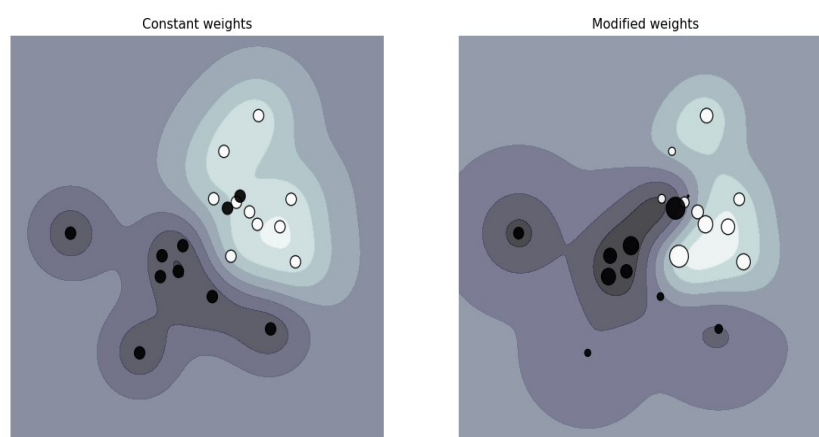


Figure 3.2: Effect of sample weighting on the decision boundary

### 3.11 Regression

The method of Support Vector Classification can be extended to solve regression problems. This method is called Support Vector Regression.

The model produced by support vector classification (as described above) depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by Support Vector Regression depends only on a subset of the training data, because the cost function ignores samples whose prediction is close to their target.

There are three different implementations of Support Vector Regression: SVR, NuSVR and LinearSVR. LinearSVR provides a faster implementation than SVR but only considers the linear kernel, while NuSVR implements a slightly different formulation than SVR and LinearSVR.

### 3.12 Complexity

Support Vector Machines are powerful tools, but their compute and storage requirements increase rapidly with the number of training vectors. The core of an SVM is a quadratic programming problem (QP), separating support vectors from the rest of the training data. The QP solver used by the libsvm-based implementation scales between  $O(n \text{ features} \times n \text{ samples}^2)$  and  $O(n \text{ features} \times n \text{ samples}^3)$  depending on how efficiently the libsvm cache is used in practice dataset dependent. If the data is very sparse  $n \text{ features}$  should be replaced by the average number of non-zero features in a sample vector.

For the linear case, the algorithm used in LinearSVC by the liblinear implementation is much more efficient than its libsvm-based SVC counterpart and can scale almost linearly to millions of samples and/or features.

### 3.13 Types of SVM

**Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

**Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

### 3.14 How an SVM works

A simple linear SVM classifier works by making a straight line between two classes. That means all of the data points on one side of the line will represent a category and the data points on the other side of the line will be put into a different category. This means there can be an infinite number of lines to choose from.

A 2-D example helps to make sense of all the machine learning. The below figures 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 show different types of separation in data.

#### Linearly Separable Data

Let us understand the working of SVM by taking an example where we have two classes that are shown in the below image which are a class A: Circle class B: Triangle. Now, we want to apply the SVM algorithm and find out the best hyperplane that divides the both classes.

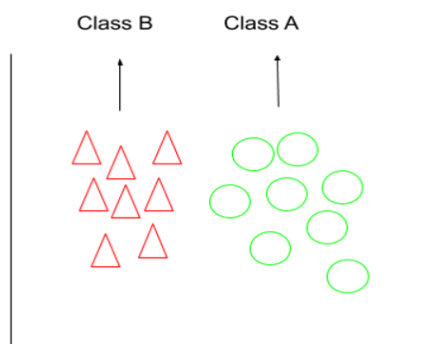


Figure 3.3: Class A and B

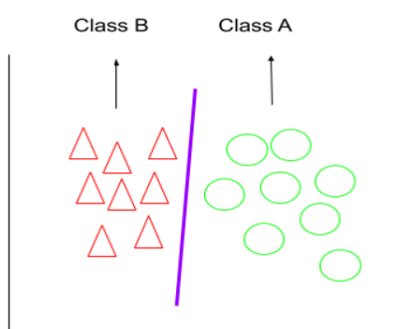


Figure 3.4: Labelled Data

SVM takes all the data points in consideration and gives out a line that is called ‘Hyper plane’ which divides both the classes. This line is termed as ‘Decision boundary’. Anything that falls in circle class will belong to the class A and vice-versa.

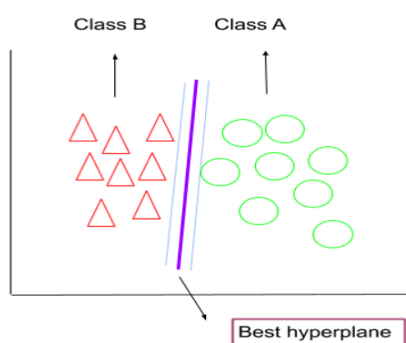


Figure 3.5: Hyperplane

All hyperplanes are not good at classification

There can be many hyperplanes that you can see but the best hyper plane that divides the two classes would be the hyperplane having a large distance from the hyperplane from both the classes. That is the main motive of SVM to find such best hyperplanes.

There can be different dimensions which solely depends upon the features we have. It is tough to visualize when the features are more than 3.

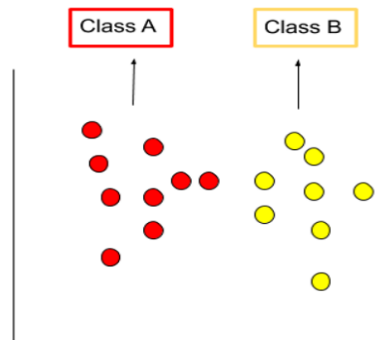


Figure 3.6: Class A- Red Class- B Yellow

Consider we have two classes that are red and yellow class A and B respectively. We need to find the best hyper plane between them that divides the two classes.

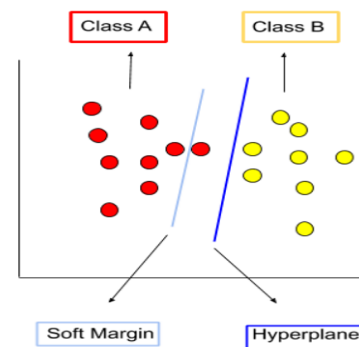


Figure 3.7: Soft margin and hyper plane

Soft margin permits few of the above data points to get misclassified. Also, it tries to make the balance back and forth between finding a hyper plane that attempts to make less misclassifications and maximize the margin.

### Linearly Non-separable Data

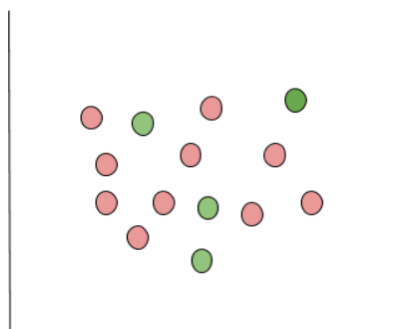


Figure 3.8: Linearly non-separable data set

If the data is non linearly separable as shown in the above figure then SVM makes use of kernel tricks to make it linearly separable. The concept of transformation of non-linearly separable data into linearly separable is called Cover's theorem - "given a set of training data that is not linearly separable, with high probability it can be transformed into a linearly separable training set by projecting it into a higher-dimensional space via some non-linear transformation". Kernel tricks help in projecting data points to the higher dimensional space by which they became relatively more easily separable in higher-dimensional space.

### 3.15 Some applications of SVM

1. Cancer Diagnosis and Prognosis - Cancer detection is one of the top research fields in the world right now. SVM helps in diagnosis and prognosis by running numerous models. There is a lot of data in the form of an image that is analyzed. We have thousands of datasets. We train hundreds of models using SVM to classify cancer as malign or benign.
2. Face detection–SVM classify parts of the image as a face and non-face and create a square boundary around the face, which we can see in the latest mobile phones.
3. Classification of images–Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
4. Handwriting recognition–We use SVMs to recognize handwritten characters used widely.

### 3.16 SVM Advantages and Disadvantages

The advantages of support vector machines are:

1. Effective in high dimensional spaces.
2. Still effective in cases where number of dimensions is greater than the number of samples.
3. Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
4. Versatile different Kernel functions can be specified for the decision function.
5. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines are:

1. If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
2. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).
3. The support vector machines in scikit-learn support both dense and sparse sample vectors as input.

## 3.17 Python Libraries

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry due to its libraries. Python libraries that used in Machine Learning are: Numpy, Sklearn, Pandas, Matplotlib.

## 3.18 Numpy

NumPy (Numerical Python) is an open-source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures. It provides ndarray, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

### 3.18.1 What's the difference between a Python list and a NumPy array?

NumPy gives you an enormous range of fast and efficient ways of creating arrays and manipulating numerical data inside them. While a Python list can contain different data types within a single list, all of the elements in a NumPy array should be homogeneous. The mathematical operations that are meant to be performed on arrays would be extremely inefficient if the arrays weren't homogeneous.

### 3.18.2 Why use NumPy?

NumPy arrays are faster and more compact than Python lists. An array consumes less memory and is convenient to use. NumPy uses much less memory to store data and it provides a mechanism of specifying the data types. This allows the code to be optimized even further.

## 3.19 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistency interface in Python.

It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release on 1st Feb. 2010.

### 3.19.1 Installation process

If you already installed NumPy and Scipy, following are the two easiest ways to install scikit-learn

#### Using pip:

Following command can be used to install scikit-learn via pip

```
pip install -U scikit-learn
```

#### Using conda:

Following command can be used to install scikit-learn via conda

```
conda install scikit-learn
```



On the other hand, if NumPy and Scipy is not yet installed on your Python workstation then, you can install them by using either pip or conda.

Another option to use scikit-learn is to use Python distributions like Canopy and Anaconda because they both ship the latest version of scikit-learn.

### 3.19.2 Features of SKlearn

Rather than focusing on loading, manipulating and summarizing data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows:

**Supervised Learning algorithms** Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc, are the part of scikit-learn.

**Unsupervised Learning algorithms** On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

**Clustering** This model is used for grouping unlabeled data.

**Cross Validation** It is used to check the accuracy of supervised models on unseen data.

**Dimensionality Reduction** It is used for reducing the number of attributes in data which can be further used for summarizing, visualization and feature selection.

**Ensemble methods** As name suggest, it is used for combining the predictions of multiple supervised models.

**Feature extraction** It is used to extract the features from data to define the attributes in image and text data.

**Feature selection** It is used to identify useful attributes to create supervised models.

**Open Source** It is open-source library and also commercially usable under BSD license.

## 3.20 Pandas

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

### 3.20.1 Features of Pandas

1. Fast and efficient DataFrame object with default and customized indexing.
2. Tools for loading data into in-memory data objects from different file formats.
3. Data alignment and integrated handling of missing data.
4. Reshaping and pivoting of date sets.
5. Label-based slicing, indexing and sub-setting of large data sets.
6. Columns from a data structure can be deleted or inserted.
7. Group by data for aggregation and transformations.
8. High performance merging and joining of data.
9. Time Series functionality.

### 3.20.2 Installing pandas

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip.

#### **pip install pandas**

If you install Anaconda Python package, Pandas will be installed by default with the following:

**Anaconda**-(from <https://www.continuum.io>) is a free Python distribution for SciPy stack. It is also available for Linux and Mac.

**Canopy**-(<https://www.enthought.com/products/canopy/>) is available as free as well as commercial distribution with full SciPy stack for Windows, Linux and Mac.

**Python-** (x,y) is a free Python distribution with SciPy stack and Spyder IDE for Windows OS. (Downloadable from <http://python-xy.github.io/>)

## 3.21 Matplotlib

It also provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPython or Tkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open source equivalent of MATLAB.

Matplotlib was originally written by John D. Hunter in 2003. The current stable version is 2.2.0 released in January 2018.

### 3.21.1 Installation of Matplotlib

Matplotlib and its dependency packages are available in the form of wheel packages on the standard Python package repositories and can be installed on Windows, Linux as well as MacOS systems using the pip package manager.

#### **pip3 install matplotlib**

In case Python 2.7 or 3.4 versions are not installed for all users, the Microsoft Visual C++ 2008 (64 bit or 32 bit for Python 2.7) or Microsoft Visual C++ 2010 (64 bit or 32 bit for Python 3.4) redistributed packages need to be installed.

If you are using Python 2.7 on a Mac, execute the following command.

#### **xcode-select --install**

Upon execution of the above command, the sub-process32 - a dependency, may be compiled.

On extremely old versions of Linux and Python 2.7, you may need to install the master version of sub process 32.

## 3.22 Seaborn

In the world of Analytics, the best way to get insights is by visualizing the data. Data can be visualized by representing it as plots which is easy to understand, explore and grasp. Such data helps in drawing the attention of key elements.

To analyse a set of data using Python, we make use of Matplotlib, a widely implemented 2D plotting library. Likewise, Seaborn is a visualization library in Python. It is built on top of Matplotlib.

### 3.22.1 Seaborn Vs Matplotlib

It is summarized that if Matplotlib “tries to make easy things easy and hard things possible”, Seaborn tries to make a well-defined set of hard things easy too”.

Seaborn helps resolve the two major problems faced by Matplotlib; the problems are

1. Default Matplotlib parameters.
2. Working with data frames.

### 3.22.2 Important Features of Seaborn

Seaborn is built on top of Python’s core visualization library Matplotlib. It is meant to serve as a complement, and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in

1. Built in themes for styling matplotlib graphics.
2. Visualizing uni-variate and bi-variate data.
3. Fitting in and visualizing linear regression models.
4. Plotting statistical time series data.
5. Seaborn works well with NumPy and Pandas data structures.
6. It comes with built in themes for styling Matplotlib graphics.

#### **seaborn.pairplot() :**

To plot multiple pairwise bivariate distributions in a dataset, we can use the pairplot() function. This shows the relationship for (n, 2) combination of variable in a Data Frame as a matrix of plots and the diagonal plots are the univariate plots.

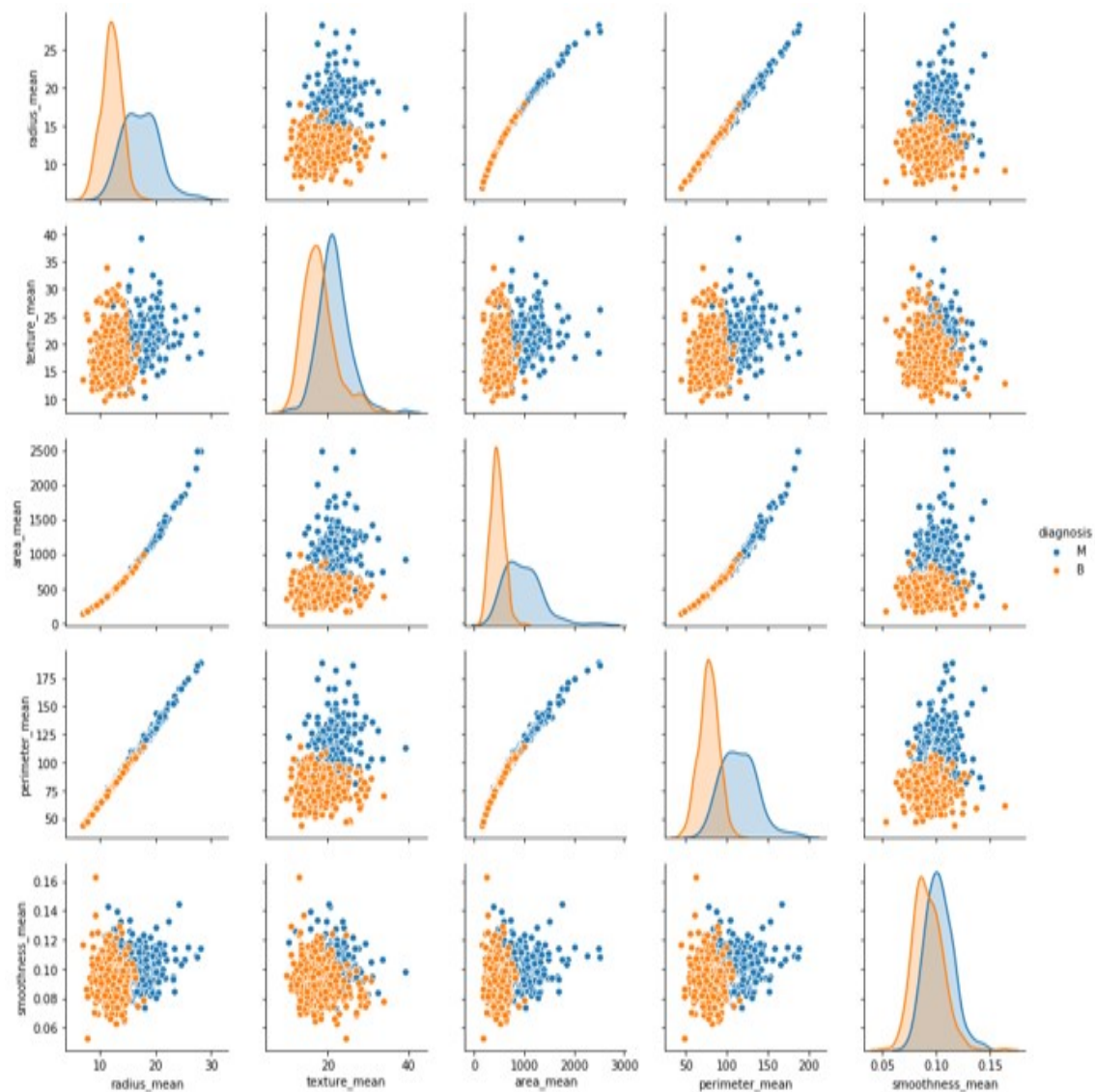


Figure 3.9: Seaborn graph

### 3.23 seaborn countplot

Seaborn count plot method is used to Show the counts of observations in each categorical bin using bars.

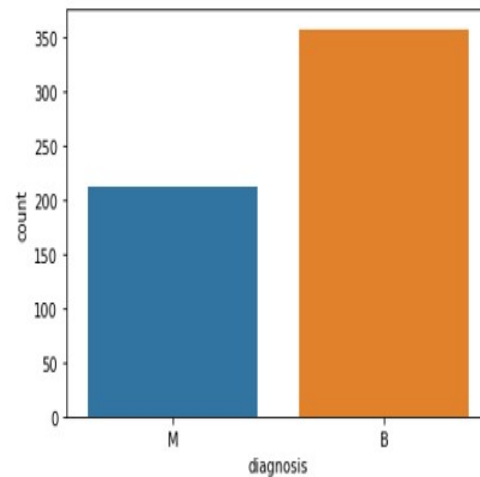


Figure 3.10: Seaborn countplot

### 3.24 Scatter plot

Scatter plot can be used with several semantic groupings which can help to understand well in a graph. They can plot two-dimensional graphics that can be enhanced by mapping up to three additional variables while using the semantics of hue, size, and style parameters. All the parameter control visual semantic which are used to identify the different subsets. Using redundant semantics can be helpful for making graphics more accessible. The below figure 3.11 shows the scatter plot graph.

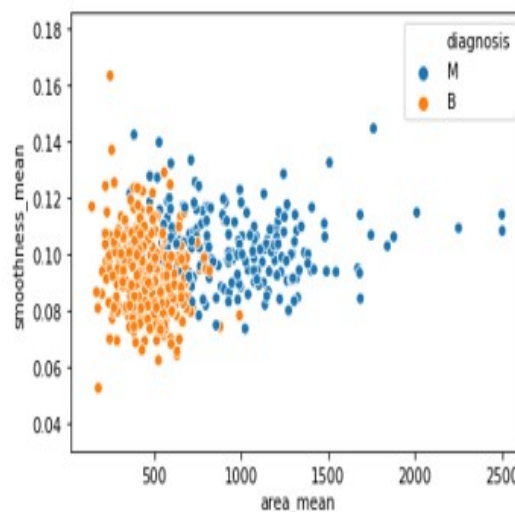


Figure 3.11: Scatter plot

# Chapter 4

## SYSTEM IMPLEMENTATION

Implementation is the process of defining how the project should be built, ensuring that it is operational and meets quality standards and qualifies the objective of the project. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

### 4.1 Python

Python is an interpreted high-level programming language for general-purpose programming and has a design philosophy that emphasizes code readability, notably using significant white space. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. Python is the most popular language for RL and ML, A great choice of libraries is one of the main reasons Python is the most popular programming language used for RL. A library is a module of code that allows use to reach some functionality or perform different actions.

Working in the ML and RL industry means dealing with a bunch of data that you need to process in the most convenient and effective way. The low entry barrier allows more data scientists to quickly pick up Python and start using it for RL development without wasting too much effort into learning the language. Python programming has prominent features like easy to use, flexible, contains powerful tools, and that makes the process of learning easier. Its simple syntax allows you to comfortably work with complex systems, ensuring clear relations between the system Libraries like Matplotlib

allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.

### 4.1.1 Applications of Python

As mentioned before, Python is one of the most widely used language over the web.

1. **Easy-to-learn** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2. **Easy-to-read** Python code is more clearly defined and visible to the eyes.
3. **Easy-to-maintain** Python's source code is fairly easy-to-maintain.
4. **A broad standard library** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
5. **Interactive Mode** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6. **Portable** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7. **Extendable** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
8. **Databases** Python provides interfaces to all major commercial databases.
9. **GUI Programming** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
10. **Scalable** Python provides a better structure and support for large programs than shell scripting.

### 4.1.2 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org/>

You can download Python documentation from <https://www.python.org/doc/>. The documentation is available in HTML, PDF, and PostScript formats.



### 4.1.3 Installing Python

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

### 4.1.4 Windows Installation

Here are the steps to install Python on Windows machine.

1. Open a Web browser and go to <https://www.python.org/downloads/>.
2. Follow the link for the Windows installer python-XYZ.msi file where XYZ is the version you need to install.
3. To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
4. Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

## 4.2 Integrated Development Environment

You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

1. **Unix** IDLE is the very first Unix IDE for Python.
2. **Windows** Python Win is the first Windows interface for Python and is an IDE with a GUI.
3. **Macintosh** The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.

## 4.3 Programming Coding Guidelines

Following Code guidelines are important to programmers for a number of reasons:

1. 80 percent of the lifetime cost of a piece of software goes to maintenance.
2. Hardly any software is maintained for its whole life by the original author.
3. Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly.
4. If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create.

## 4.4 Methodology of the project

This Project ensure a systematic and structured way of designing the project in a very transparent manner for achieving the objective goals, consider it as a standard procedure for building the machine learning projects, this is explained in brief in future topics.

## 4.5 Data Collection and Exploration

First import the data set (csv file) in our working python area (jupyter notebook) for performing various techniques, the code snippet for the importing dataset into the notebook is given in the code below, data exploring refers to view the dataset in an organized way.

## 4.6 Data Cleaning

Removing the missing data.

Always check to see if there is missing data.As this dataset is huge, removing data points with missing data probably has no effect on the models being trained.

In [4]: `df_cancer.tail()`

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	rs
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	NaN	NaN	...	
569	925601	M	21.01	23.45	102.10	NaN	0.11100	0.10340	0.24390	0.13890	...	

5 rows x 32 columns

In [5]: `df_cancer.shape`

Out[5]: (570, 32)

Figure 4.1: Data cleaning

## 4.7 Data Pre-processing

This is the most important part in the machine learning workflow. Since the algorithm is totally dependent on how data feed into it, feature engineering which is an integrated step in data pre-processing should be given top most priority for every machine learning project.

Some of the advantages of the pre-processing of the data is that it Reduces over fitting less redundant data will be possible which means less opportunity to make decisions based on noise and it Improves Accuracy which means fewer misleading data means modelling accuracy improves. Reduces Training time fewer data points reduce algorithm complexity and algorithms train faster.

## 4.8 Model Evaluation

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models.

## 4.9 Flow Chart

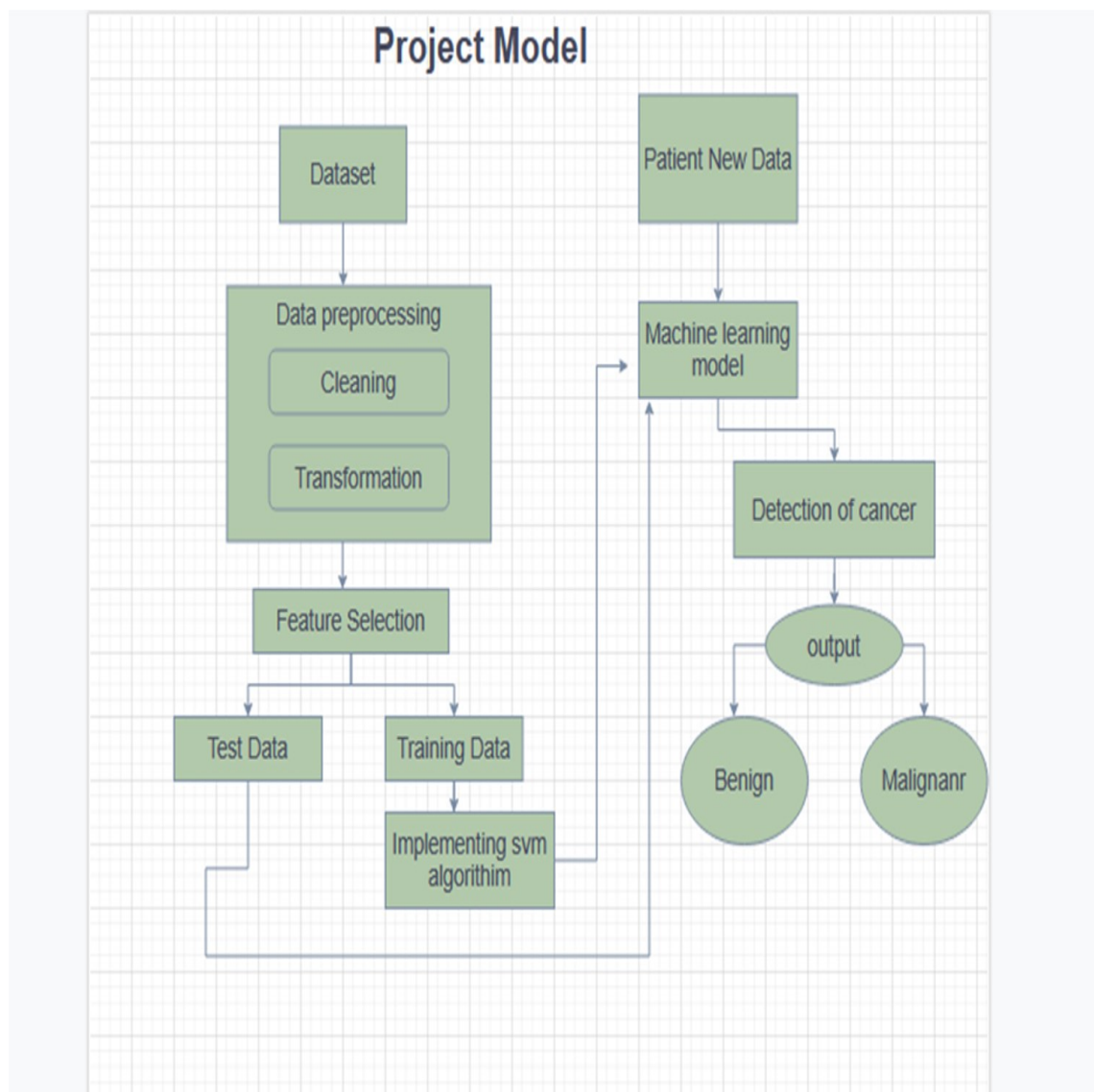


Figure 4.2: Flow chart

The above figure shows the proper working of the project model with steps.

---

## 4.10 Summary of flowchart

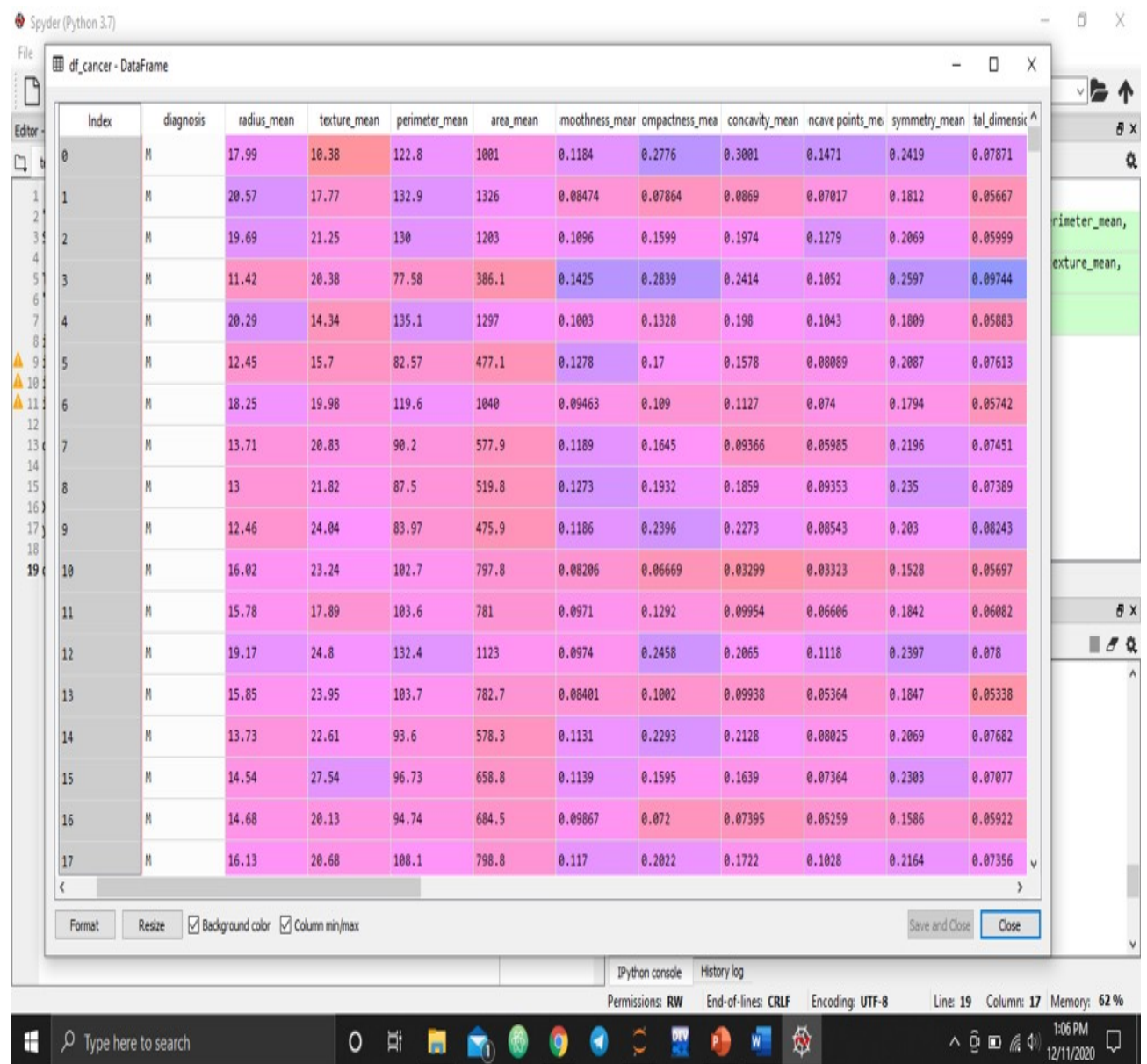
1. Importing the dataset.
2. Checking the Null values and initializing them with zeros.
3. Selecting the features required for the model training.
4. Dividing the dataset into training and testing data.
5. Model training using machine learning algorithm and checking accuracy on testing data.
6. Providing data and getting output.

# Chapter 5

## RESULTS AND DISCUSSION

### 5.1 Dataset

The datasets are used for machine learning research and have been cited in peer reviewed academic journals. Datasets are an integral part of the field of machine learning.



Index	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	npoints_mean	symmetry_mean	tal_dimensic
0	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871
1	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
2	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999
3	M	11.42	20.30	77.50	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744
4	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883
5	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613
6	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742
7	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451
8	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389
9	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243
10	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697
11	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082
12	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078
13	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338
14	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682
15	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077
16	M	14.68	20.13	94.74	604.5	0.09067	0.072	0.07395	0.05259	0.1586	0.05922
17	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356

Figure 5.1: Dataset

## 5.2 Accuracy of Model

Accuracy of the dataset is 98 percent when the data set is pre-processed second and final time. The accuracy we have got is very good for the accurate results.

```
In [41]: accuracy = (90+51)/(143)
          accuracy*100
```

```
Out[41]: 98.6013986013986
```

```
In [42]: print(classification_report(y_test,y_predict))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	91
1	0.98	0.98	0.98	52
micro avg	0.99	0.99	0.99	143
macro avg	0.98	0.98	0.98	143
weighted avg	0.99	0.99	0.99	143

Figure 5.2: Accuracy of model

## 5.3 Input patient new data in the program

This is the final step, after finding the accuracy of the dataset and training the model we can give multiple input data of the new patient. There are 30 attributes in the dataset we have to enter all the data values of the particular patient. After giving necessary input data to the program we can get the final result either cancerous cells or non-cancerous cells. The below figure shows the various input of the data from the user.

## 5.4 Results

Detection of breast cancerous cell by identifying the type of cell. The parameters are identified by the help of microscopic image of cells.





# Chapter 6

## CONCLUSION AND FUTURE WORK

As the conclusion of the project has been represented briefly to show the work carried out in building a system model. It also talks about the other enhancements that can be made into the project so that it is user friendly and make more accurate predictions.

### 6.1 Conclusion

The system will be developed with care and it will be free of errors and at the same time efficient and less time consuming. System will be robust also, the provision will be provided for future developments in the system.

In this project, different types of models will be reviewed and their accuracies computed and compared with each other, so that the best prediction model can be used by doctors in real life to identify breast cancer relatively faster than previous methods.

### 6.2 Future Work

In this project we learned to build a breast cancer tumour predictor on the dataset and created graphs and results for the same. It has been observed that a good dataset provides better accuracy. Selection of appropriate algorithms with good home dataset will lead to the development of prediction systems.

These systems can assist in proper treatment methods for a patient diagnosed with breast cancer. There are many treatments for a patient based on breast cancer stage data mining and machine learning can be a very good help in deciding the line of treatment to be followed by extracting knowledge from such suitable databases.

# References

- [1] <https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/>
- [2] <https://towardsdatascience.com/building-a-simple-machine-learning-model-on-breast-cancer-data-eca4b3b99fa3> data Set: <http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+>
- [3] <https://tatwan.github.io/How-To-Plot-A-Confusion-Matrix-In-Python/> [http://en.wikipedia.org/wiki/Virtual\\_Network\\_Computing](http://en.wikipedia.org/wiki/Virtual_Network_Computing)
- [4] LibVNC server [https://seaborn.pydata.org/tutorial/axis\\_grids.html](https://seaborn.pydata.org/tutorial/axis_grids.html)
- [5] DirectFB documentation <https://seaborn.pydata.org/generated/seaborn.pairplot.html>
- [6] <http://www.kaggle.com/dataset>
- [7] <http://en.wikipedia.org/wiki>

## PROJECT CODE

**# importing the libraries**

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

**#importing our cancer dataset**

```
df_cancer = pd.read_csv('C: //users //jprt //Desktop //Desktop
Folders //project file //data.csv')
X = df_cancer.iloc[:, 2:32]
y = df_cancer.iloc[:, 1]
df_cancer.head()
X.head()
X.tail()
```

**# Checking count of Null values in respective columns**

```
nulls = X.isnull().sum()
nulls>nulls > 0]
```

**# Filling Null values with 0**

```
X.fillna(0, inplace=True)
X.tail()
y.head()
y.tail()
```

## **# Data Visualization**

```
sns.pairplot(df_cancer, hue = 'diagnosis', vars = ['radius_mean',  
'texture_mean', 'area_mean', 'perimeter_mean', 'smoothness_mean']  
) plt.show()
```

```
sns.countplot(df_cancer['diagnosis'], label = "Count") plt.show()
```

```
sns.scatterplot(x = 'area_mean', y = 'smoothness_mean', hue =  
'diagnosis', data = df_cancer) plt.show()
```

## **# Encoding of y**

```
from sklearn.preprocessing import LabelEncoder  
label_y = LabelEncoder()  
y = label_y.fit_transform(y)  
for i in range(5):  
    print(y[i])
```

## **# splitting dataset**

```
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=5)
```

```
X_train.shape
```

```
X_test.shape
```

```
y_train.shape
```

```
y_test.shape
```

## **# Model training**

```
from sklearn.svm import SVC
classifier = SVC()
classifier.fit(X_train, y_train)
```

### **# Model Evaluation**

```
y_predict = classifier.predict(X_test)
from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test,y_predict)
sns.heatmap(cm, annot=True)
plt.show()
```

### **# improving Model**

```
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
X_train_scaled = sc_x.fit_transform(X_train)
X_test_scaled = sc_x.transform(X_test)
from sklearn.svm import SVC
classifier = SVC()
classifier.fit(X_train_scaled, y_train)
y_predict = classifier.predict(X_test_scaled)
```

### **# Model Evaluation**

```
from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test,y_predict)
sns.heatmap(cm, annot=True)
plt.show()
```

```
# Runtime input and prediction of new value
```

```
test = []  
for cols in X.columns:  
    print("input value of "+cols)  
    p = float(input())  
    test.append(p)
```

```
# after input of values
```

```
scaled_data = sc_x.transform([test])  
prediction = classifier.predict(scaled_data)  
# Printing output  
# If prediction is 1 it is malignant [Cancerous]  
# If prediction is 0 it is Benign [Non-Cancerous]  
if prediction[0]:  
    print("cancerous cells detected with an accuracy  
of",accuracy*100,"percent")  
else:  
    print("Cancerous cells not detected with an accuracy  
of",accuracy*100,"percent")
```