Breast Cancer Classfication

A PROJECT REPORT

Submitted by

GVS Madhan Chaithanya
Reg. No. 17MCA1032

in partial fulfillment for the award of the degree of

Master of Computer Applications

School of Computing Science and Engineering
Vellore Institute of Technology
Vandalur - Kelambakkam Road, Chennai - 600 127

April - 2019

School of Computing Science and Engineering

DECLARATION

I hereby declare that the project entitled **Breast Cancer Classfication** submitted by me to the School of Computing Science and Engineering, VIT Chennai, 600 127 in partial fulfillment of the requirements of the award of the degree of **Master of Computer Applications** is a bona-fide record of the work carried out by me under the supervision of **Dr.Vergin Raja Sarobin M**. I further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place: Chennai
Date:

Signature of Candidate
(GVS Madhan Chaithanya)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled **Breast Cancer Classfication** is prepared and submitted by **GVS Madhan Chaithanya (Reg. No. 17MCA1032)** to VIT Chennai, in partial fulfullment of the requirement for the award of the degree of **Master of Computer Applications** is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Guide/Supervisor	**Program Chair**
Name: Dr.Vergin Raja Sarobin M	Name: Dr.Asha
Date:	Date:

Examiner	**Examiner**
Name:	Name:
Date:	Date:

(Seal of SCSE)

Acknowledgement

I wish to express my sincere gratitude to **Dr.Asha S**. Program chair of **MCA** and to Dr.P.Nithyanandam. Project Coordinator for providing me an opportunity to do my project work. I would like to express my gratitude and sincere thanks to my internal guide **Dr. Vergin Raja Sarobin M**. who inspite of her busy schedule guided me in the correct path. I wish to express my sincere thanks to our management for providing me an excellent academic environment and facilities for pursuing MCA program. I am grateful to **Dr. Vaidehi Vijayakumar**, Dean of School of Computing Science and Engineering, Vellore Institute Technology, Chennai. I thank my family and friends who motivated me during the course of the project work.

GVS Madhan Chaithanya
Reg. No. 17MCA1032

Abstract

Breast cancer the most common cancer among women worldwide accounting for 25% of all cancer cases and affected 2.1 million people in 2015 early diagnosis significantly increases the chance of survival. The key challenge in cancer detection is how to classify tumors into malignant or benign. Machine learning techniques can dramatically improve the accuracy of diagnosis. Research indicates that most experienced physicians can diagnosis is achieved using machine learning techniques. In this project, our task is to classify tumors into malignant or benign tumors using features of pain from several images. From them, we are going to classify using some classification techniques such as Support Vector Machine (SVM), K-Nearest Neighbors(KNN), Logistic Regression, Decision Tree and Random Forest. By using them we can conclude the best accuracy.

So, by using the python and anaconda distribution packages we are classifying them and predicting the accuracy and finally by predicting it's good accuracy.

Contents

List of Figures

Chapter 1

Introduction

Early detection of cancer is essential for a rapid response and better chances of cure. Unfortunately, early detection of cancer is often difficult because the symptoms of the disease at the beginning are absent. Thus, cancer remains one of the topics of health research, where many researchers have invested with the aim of creating evidence that can improve treatment, preventions, and diagnostics[5].

Cancer was defined as the disease caused by an uncontrolled division of abnormal cells in the part of the body. The two keywords are uncontrolled and abnormal. All division of cells going on in our body all the time, it's how we grow, it's how we differentiate and it's important for our livelihood. But what if those cells run or behave become abnormal? And their division is completely uncontrolled? Kind of like an accelerator with no brake pedal and no traffic lights. This is red wagon down the hill at 90 miles an hour with no brakes in sight, what happens? Disaster, that's cancer. When a cell becomes abnormal divide and proliferate without any control and without any regulation[5].

Breast Cancer the most common cancer among the women worldwide accounting for 25 percent of all cancer cases and affected 2.1 million people in 2015 early diagnosis significantly increases the chances of survival. The key challenge in cancer detection is how to classify tumors into malignant or benign machine learning techniques can improve the accuracy of diagnosis. Research indicates that most experienced physicians can diagnose cancer with 79 percent accuracy while 91percent correct diagnosis is achieved using machine learning techniques. Breast cancer, in particular, is one of that's of key relevance. It tends to be most cancer affecting women in the united states and second-leading cause-related deaths, it's not just the united states. And also it is the second biggest killer in India. Nearly 8.17 percent of cancer deaths in the world in the year 2017-18[6].

School of Computing Science and Engineering, Vellore Institute of Technology, Chennai

Information and Communication Technologies (ICT) can play potential roles in cancer care. In fact, Big data has advanced not only the size of data but also creating value from it; Big data, that becomes synonymous of data mining, business analytics, and business intelligence(BI), has made a big change in BI from reporting and decision to prediction results. Data mining approaches, for instance, applied to medical science topics rise rapidly due to their high performance in predicting outcomes, reducing costs of medicine, promoting patients' health, improving healthcare value and quality and in making a real-time decision to save people's lives.

Breast Cancer's causes are multifactorial and involve family history, obesity, hormones, radiation therapy, and even reproductive factors. Every year, one million women are newly diagnosed with breast cancer, according to the report of the world health organization half of them would die, because it's usually late when doctors detect cancer. Breast Cancer is caused by a typo or mutation in a single cell, which can be shut down by the system or causes a reckless cell division. If the problem is not fixed after a few months, masses are formed from cells containing wrong instructions. The second major cause of women's death is breast cancer (after lung cancer). 246,660 of women's new cases of invasive breast cancer are expected to be diagnosed in the US during 2016 and 40,450 of women's death is estimated. Breast cancer represents about 12 percent of all new cancer cases and 25 percent of all cancers in women.

Recently studies of International Agency for Research by WHO of Globocan 2018 report, Malignant tumors expand to the neighboring cells, which can lead to metastasize or reach other parts, whereas benign masses can't expand to other tissues, the expansion is then only limited to the benign mass. Detection of BC may be hard at the beginning of the disease, due to the absence of symptoms, after some clinical tests, the accurate diagnosis should have the ability to differentiate the benign and malignant tumors. A good detection provides low false positive (FP) rate and false negative (FN) rate.

Recently studies of International Agency for Research by WHO of Globocan 2018 report, Malignant tumors expand to the neighboring cells, which can lead to metastasize or reach other parts, whereas benign masses can't expand to other tissues, the expansion is then only limited to the benign mass[1]. Detection of BC may be hard at the beginning of the disease, due to the absence of symptoms, after some clinical tests, the accurate diagnosis should have the ability to differentiate the benign and malignant tumors. A good detection provides low false positive (FP) rate and false negative (FN) rate.

Research in this area is a quest of knowledge through surveys, studies, and

School of Computing Science and Engineering, Vellore Institute of Technology, Chennai

experiments conducted with applications in order to discover and interpret new knowledge to prevent and minimize the risk-averse consequences. To understand this problem more precisely, tools are still needed to help oncologists to choose the treatment required for healing or prevention of recurrence by reducing the harmful effects of certain treatments and their costs. To develop tools for cancer management, machine learning methods and clinical factors, such as patient age and histopathological variables form the basis for daily decision making are used. Several studies have been developed in this topic by using gene expressions or using image processing.

Machine learning is a set of tools utilized for the creation and evaluation of algorithms that facilitate prediction, pattern recognition, and classification. ML is based on four steps: Collecting data, picking the model, training the model, testing the model. The relation between breast cancer and machine learning is not recent, it had been used for decades to classify tumors and other malignancies, predict sequences of genes responsible for cancer and determine the prognostic.

The classification's aim is to put each observation in a category that it belongs to. In this study, we used some machine learning classifiers which are support vector machines, knearest neighbor, logistic regression, decision tree, random forest, artificial neuron network. The purpose is to determine whether a patient has a benign or malignant tumor. In this study, we customize some techniques of machine learning for classification of breast cancer. We use the Wisconsin breast cancer database which is the publicly available breast cancer database and is widely studied. The purpose of this report is developing effective machine learning approaches for cancer classification using some classifiers in a data set. The performance of each classifier will be evaluated in terms of accuracy, training process and testing process[3].

1.1 Risk Factors:

What are the risk factors for developing breast cancer? There are two main factors to group this. They are that they can do something about the modifiable ones and those that really out of their control, what we call non-modifiable. Truthfully, when we loop at all of the risk factors for developing breast cancer, the two main risk factors are being a woman and getting older. What are we going to do about that? There are other non-modifiable risk factors as well. Genetic makeup, family history, some of them may carry mutations, mistakes in genes that were designed to protect them from getting cancer. So having a mutation or a mistake in those

genes, that's kind of like a chink in their armor. It increases the risk of developing breast cancer. Some of them have intrinsically benign breast disorder, things like atypia. Atypia is funny looking cells. There are not cancer or precancer in and of themselves, but they are an increase in their risk of developing cancer and there are hormones that they make. Their ovaries produce hormones when they're premenopausal and their adrenal glands which sit on top of their kidney also make hormones. These hormones, which are converted to estrogen in peripheral fat also increase their risk of developing breast cancer[1].

In general, there's little that we can do about non-modifiable risk factors. What are the things the women can do something about that? Well, they don't have to take hormones. Many women take hormones. Many women take hormones to reduce menopausal side effects. Hot flashes, vaginal dryness. Taking these hormones does increase their risk of developing breast cancer and so not taking them or taking them in the lowest dose for the shortest amount of time possible decreases their risk of developing breast cancer. We know that breastfeeding also decreases their risk. So reducing their alcohol intake to make it mild to moderate say, one to two glasses a week will reduce their risk of developing breast cancer. Of course, a healthy lifestyle, eating the right food, exercise, maintaining normal body weight, reduced obesity also reduces their risk of developing breast cancer.

1.2 Facts and Proofs

1.2.1 World-wide Cancer facts

Figure 1.1: World Geography

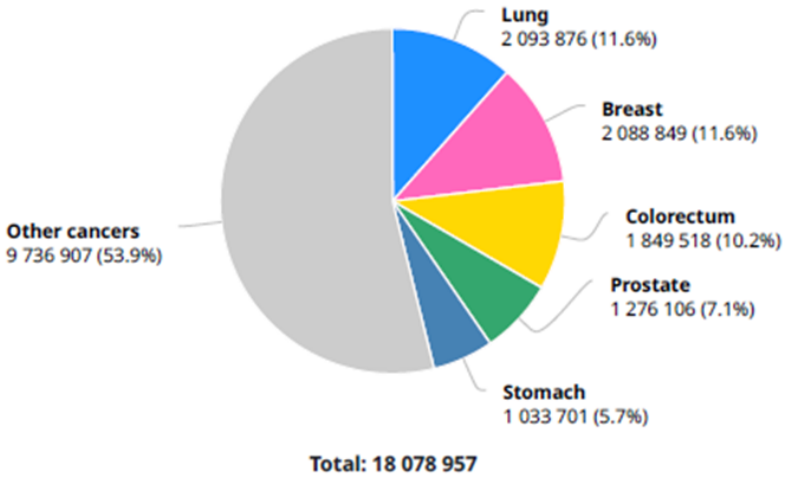

Figure 1.2: World Geography both sexes

According to the world population, the count is 7 632 819 272. Among that number of new cases of cancer is 18 078 957. The number of death cases is 9 555 027. And Number of prevalent cases (5-year) 43 841 302[1].

A number of new cases in 2018 by both males and females of all ages, the total cases are 18,078,957. The following figure will give you a glance at the statistics of cancer data.

A number of new cases in 2018 by females of all ages, the total cases are 8,622,539. Among that female, those are suffering from cancers like breast, colorectum, lung, cervix uteri, thyroid, other cancers.

Summary statistics for 2018 of all cancers, it includes the steps and comparison of both males and females. And in that, we will see the new cases and death cases and age-standard incidence and mortality.

1.2.2 Breast Cancer facts

According to the study of GLOBALCAN-2018 report, some facts of breast cancer data was shared by them. Not only the breast cancer the report belongs and made with the organization help of WHO(World Health Organization)[1]. This was created. See the figures from [fig-1.5 to fig-1.8]

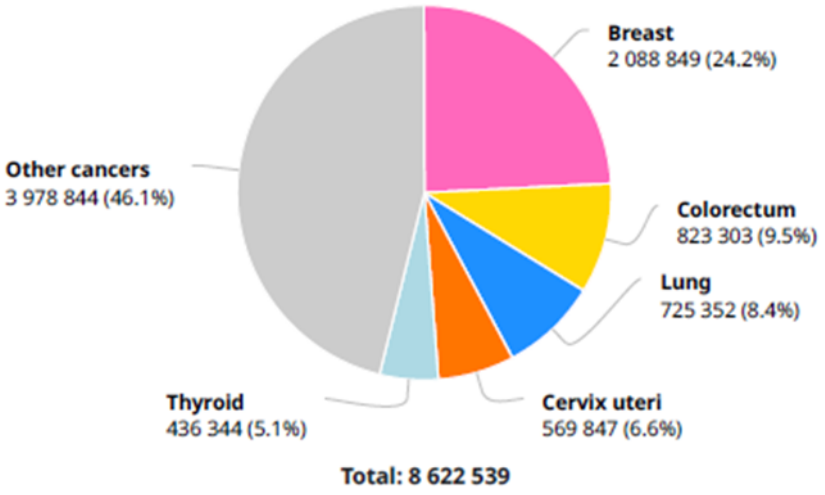

Figure 1.3: World Geography female cancers

	Males	Females	Both sexes
Population	3 850 719 284	3 782 099 828	7 632 819 272
Number of new cancer cases	9 456 418	8 622 539	18 078 957
Age-standardized incidence rate (World)	218.6	182.6	197.9
Risk of developing cancer before the age of 75 years (%)	22.4	18.3	20.2
Number of cancer deaths	5 385 640	4 169 387	9 555 027
Age-standardized mortality rate (World)	122.7	83.1	101.1
Risk of dying from cancer before the age of 75 years (%)	12.7	8.7	10.6
5-year prevalent cases	21 014 830	22 826 472	43 841 302
Top 5 most frequent cancers excluding non-melanoma skin cancer (ranked by cases)	Lung Prostate Colorectum Stomach Liver	Breast Colorectum Lung Cervix uteri Thyroid	Lung Breast Colorectum Prostate Stomach

Figure 1.4: Summary statistics

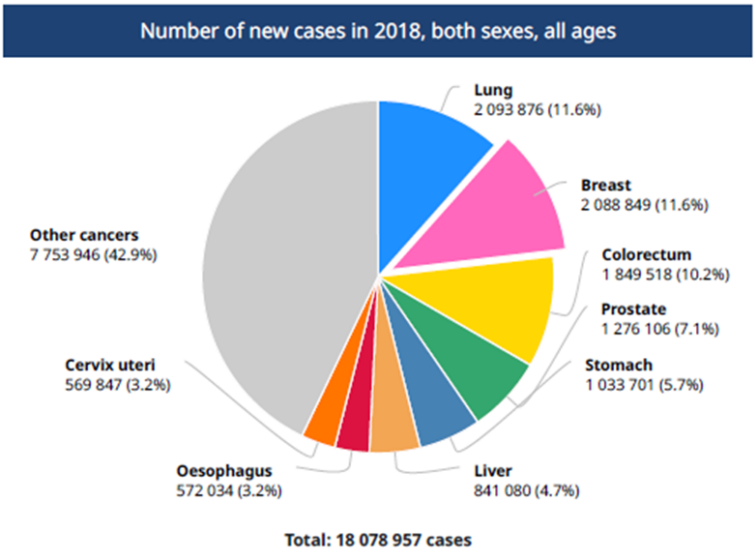

Figure 1.5: Breast cancer incidence

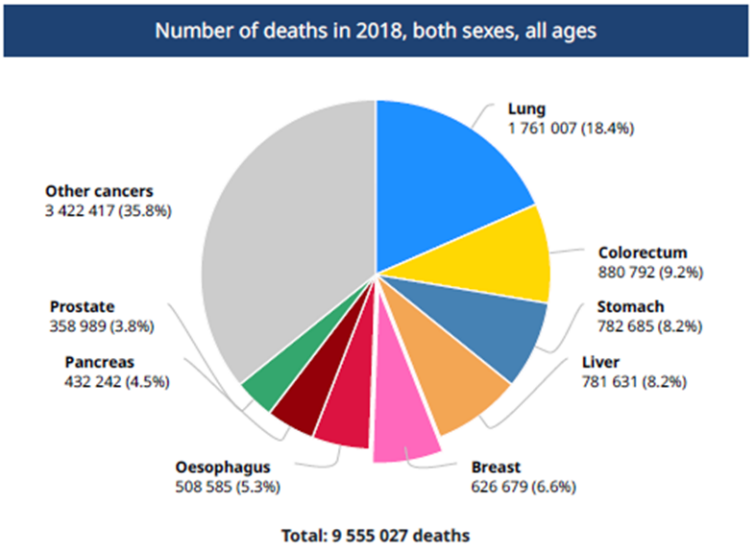

Figure 1.6: Breast cancer mortality

Cancer incidence and mortality statistics worldwide and by region												
	Incidence						Mortality					
	Both sexes		Males		Females		Both sexes		Males		Females	
	New cases	Cum. risk 0-74 (%)	New cases	Cum. risk 0-74 (%)	New cases	Cum. risk 0-74 (%)	Deaths	Cum. risk 0-74 (%)	Deaths	Cum. risk 0-74 (%)	Deaths	Cum. risk 0-74 (%)
Eastern Africa	40 310	3.15	-	-	40 310	3.15	20 165	1.62	-	-	20 165	1.62
Middle Africa	14 486	2.89	-	-	14 486	2.89	7 864	1.64	-	-	7 864	1.64
Northern Africa	53 917	5.06	-	-	53 917	5.06	20 058	1.96	-	-	20 058	1.96
Southern Africa	14 820	4.93	-	-	14 820	4.93	5 002	1.60	-	-	5 002	1.60
Western Africa	45 157	3.92	-	-	45 157	3.92	20 983	1.92	-	-	20 983	1.92
Caribbean	14 097	5.50	-	-	14 097	5.50	5 496	1.95	-	-	5 496	1.95
Central America	35 349	4.17	-	-	35 349	4.17	9 341	1.14	-	-	9 341	1.14
South America	150 288	6.19	-	-	150 288	6.19	37 721	1.45	-	-	37 721	1.45
North America	262 347	9.32	-	-	262 347	9.32	46 963	1.38	-	-	46 963	1.38
Eastern Asia	476 509	4.15	-	-	476 509	4.15	119 678	0.93	-	-	119 678	0.93
South-Eastern Asia	137 514	4.17	-	-	137 514	4.17	50 935	1.61	-	-	50 935	1.61
South-Central Asia	241 077	2.81	-	-	241 077	2.81	123 060	1.53	-	-	123 060	1.53
Western Asia	55 914	4.81	-	-	55 914	4.81	16 904	1.45	-	-	16 904	1.45
Central and Eastern Europe	149 024	6.10	-	-	149 024	6.10	49 951	1.80	-	-	49 951	1.80
Western Europe	169 640	9.90	-	-	169 640	9.90	41 629	1.65	-	-	41 629	1.65
Southern Europe	119 577	8.51	-	-	119 577	8.51	28 064	1.41	-	-	28 064	1.41
Northern Europe	84 272	9.63	-	-	84 272	9.63	18 063	1.46	-	-	18 063	1.46
Australia and New Zealand	22 062	10.16	-	-	22 062	10.16	3 631	1.37	-	-	3 631	1.37
Melanesia	2 116	5.30	-	-	2 116	5.30	1 046	2.73	-	-	1 046	2.73
Polynesia	252	7.46	-	-	252	7.46	78	2.46	-	-	78	2.46
Micronesia	121	4.44	-	-	121	4.44	47	1.71	-	-	47	1.71
Low HDI	105 620	3.40	-	-	105 620	3.40	52 846	1.78	-	-	52 846	1.78
Medium HDI	402 800	3.34	-	-	402 800	3.34	183 827	1.61	-	-	183 827	1.61
High HDI	666 731	4.29	-	-	666 731	4.29	184 014	1.12	-	-	184 014	1.12
Very high HDI	912 469	8.16	-	-	912 469	8.16	205 616	1.44	-	-	205 616	1.44
World	2 088 849	5.03	-	-	2 088 849	5.03	626 679	1.41	-	-	626 679	1.41

Figure 1.7: Breast cancer incidence and motality among world wide

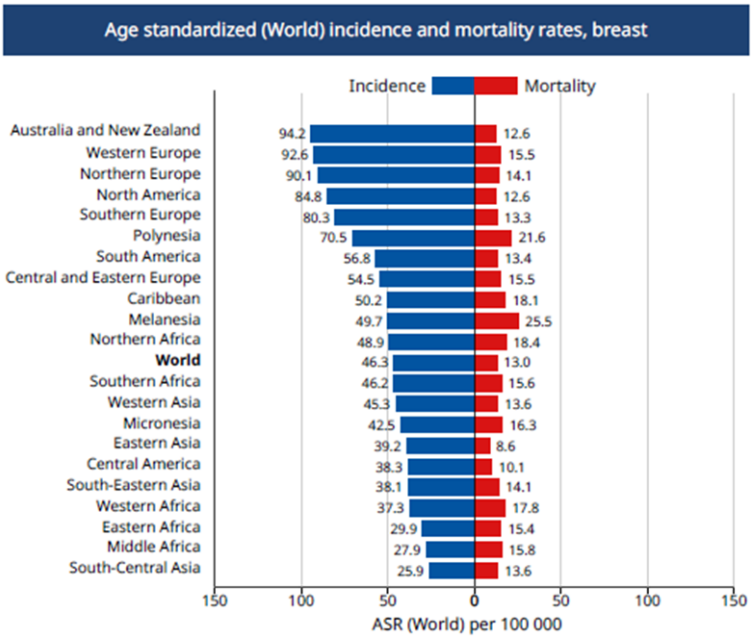

Figure 1.8: Compartive graph of incidence and mortality

Chapter 2

Overview/Literature Study

Breast cancer is the second leading cause of death for women all over the world. But early detection and prevention can significantly reduce the chances of death. This paper deals with different statistical and deep learning analysis of Wisconsin Breast Cancer Database for improving the accuracy in detection and classification of breast cancer based on different attributes. Efficient Approaches for Accuracy Improvement of Breast Cancer Classification Using Wisconsin Database[5]. In this paper, they have analyzed the data on the basis of Wisconsin Breast cancer database and we experimented with Naive Bayes, SVM, Logistic Regression, KNN, Random Forest Neural Network and CNN classifiers and obtained highest accuracy. They did a deep investigation in the performance of different deep networks on this dataset. For deep networks, they have found that the convergence time significantly increases and it gets harder to optimize the network. In case of convolutional neural network, they have found the best result with three hundred feature maps. The same result might be obtained with different configuration of the network. Their results of CNN classifier (98.06 percent accuracy) show comparatively better performance in comparison the work of Karabatak and Cevdet-Ince (2009) [8] where the accuracy was 97.4 percent using Association Rules(AR) and Neural Network(NN). Such comparative analysis on breast cancer classification would provide further encouragement and insights on the efficient approaches for detection of cancer problems.

Comparative Study of Machine Learning Algorithmsfor Breast Cancer Detection and Diagnosis-2016 IEEE. In this paper ML techniques have been widely used in the medical field and have served as a useful diagnostic tool that helps physicians in analyzing the available data as well as designing medical expert sys-

tems. This paper presented three of the most popular ML techniques commonly used for breast cancer detection and diagnosis, namely Support Vector Machine (SVM), Random Forest (RF) and Bayesian Networks (BN). The main features and methadolgy of each of the three ML techniques was described. Performance comparison of the investigated techniques has been carried out using the Original Wisconsin Breast Cancer Data set[6].

Simulation results obtained has proved that classification performance varies based on the method that is selected. Results have showed that SVMs have the highest performance in terms of accuracy, specificity and precision. However, RFs have the highest probability of correctly classifying tumor[7].

Malignant tumors expand to the neighboring cells, which can lead to metastasize or reach other parts, whereas benign masses can't expand to other tissues, the expansion is then only limited to the benign mass. Detection of BC may be hard at the beginning of the disease, due to the absence of symptoms, after some clinical tests, the accurate diagnosis should have the ability to differentiate the benign and malignant tumors. A good detection provides low false positive (FP) rate and false negative (FN) rate[8].

In the Wisconsin Breast Cancer datasets, they used most two main algorithms, which are: NB and KNN, since our target and challenge from breast cancer classification is to build classifiers that are precise and reliable. After an accurate comparison between our algorithms, we noticed that KNN achieved a higher efficiency of 97.51 percent, however, even NB has a good accuracy at 96.19 percent if the dataset is larger, the KNN's time for running will increase[9].

Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis. In this paper, n important challenge in data mining and machine learning areas is to build accurate and computationally efficient classifiers for Medical applications. In this study, they employed four main algorithms: SVM, NB, k-NN and C4.5 on the Wisconsin Breast Cancer (original) datasets. We tried to compare efficiency and effectiveness of those algorithms in terms of accuracy, precision, sensitivity and specificity to find the best classification accuracy. of SVM reaches and accuracy of 97.13 percent and outperforms, therefore, all other algorithms. In conclusion, SVM has proven its efficiency in Breast Cancer prediction and diagnosis and achieves the best performance in terms of precision and low error rate.

All experiments are executed within a simulation environment and conducted in WEKA data mining tool[10].

Support vector machines combined with feature selection for breast cancer diagnosis. In this paper, breast cancer diagnosis based on a SVM-based method combined with feature selection has been proposed. Experiments have been conducted on different training-test partitions of the Wisconsin breast cancer dataset (WBCD), which is commonly used among researchers who use machine learning methods for breast cancer diagnosis. The performance of the method is evaluated using classification accuracy, sensitivity, specificity, positive and negative predictive values, receiver operating characteristic (ROC) curves and confusion matrix. The results show that the highest classification accuracy (99.51 percent) is obtained for the SVM model that contains five features, and this is very promising compared to the previously reported results[8].

Breast Cancer Diagnosis by using k-Nearest Neighbor with Different Distances and Classification Rules. In this paper of study,we have highlighted the algorithm K-nearest neighbors for classification. We used this algorithm with several different types of distances and classification rules (majority, consensus and random) in function of the parameter k that we varied in the interval [1; 50]. This algorithm was used in the medical diagnosis that is in the diagnosis and classification of cancer. This experiments were conducted on the database WBCD (Wisconsin Breast Cancer Database) obtained by the University Hospital of Wisconsin[9]. The results advocate the use of the k-nn algorithm with both types of Euclidean distance and Manhattan. These distances are effective in terms of classification and performance but are consuming much time. Nevertheless, they remain two types of distance that give the best results (98; 70% for Euclidean distance and 98; 48% for Manhattan whith k = 1), these values are not significantly affected even when k = 1 is increased to 50[10].

In this study, the effect of dimensionality reduction using independent component analysis (ICA) on breast cancer decision support systems with several classifiers such as artificial neural network (ANN), k-nearest neighbor (k-NN), radial basis function neural network (RBFNN), and support vector machine (SVM) is investigated. The results of the applied original thirty features ofWisconsin diagnostic breast cancer (WDBC) are compared with the reduced one dimension by ICA. The accuracy rates of the classifications with thirty original features except

RBFNN have slightly decreased from 97.53%, 91.03%, and 95.25% to 90.5%, 91.03%, and 90.86%, respectively. However, the one-dimensional feature vector causes RBFNN classifier to be more distinguishing with the increased accuracy from 87.17% to 90.49%. Furthermore, the sensitivity rates which define the successfully recognized malignant samples are increased from 93.5% to 96.63% for RBFNN and from 96.07% to 97.47% for SVM, while the others have slight decrease at the rate between 0.96% and 3.09%. If the objective is to increase the rate of the successfully identifiedmalignant breast cancer using RBFNN or decrease computational complexity without loss of the high accuracy rate, feature reduction applying ICA can be a high performance solution[11].

Chapter 3

System Design

3.1 Tools and Technology

3.1.1 Python

Python is an interpreted, high-level, general-purpose programming language.Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.

3.1.2 Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS.

Jupyter

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The ”notebook” term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format

depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

3.1.3 Libraries

Scikit learn Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

3.2 Dataset Description

The dataset which we used for this study is Breast Cancer Wisconsin (Diagnostic) Database. Predicting if the cancer diagnosis is benign or malignant based on several observations/features. 30 features are used, for example:

- radius (mean of distances from the 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)

Datasets are linearly separable using all 30 input features. Number of Instances: 569. Class Distribution: 212 Malignant, 357 Benign. Target class:

- Malignant
- Benign

The data source we collected from the resource of the UCI Machine Learning Repository. To conducting from the comparative study to predict which classification is better to predict. 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. The classes of this data set are:

- WDBC-Malignant.
- WDBC-Benign.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image. Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

3.3 Flow Chart

While doing the classfication process steps, the following are the steps will going on this paper. This steps are applicable for any type of classifcation alogrithms. So in this paper we are seeing the alogorithms like Support vector machines(svm), K-nn, loggistic regression, random forest and decision tree. By, all follow these flow chart throughout the paper.(See fig:3.1,3.2)

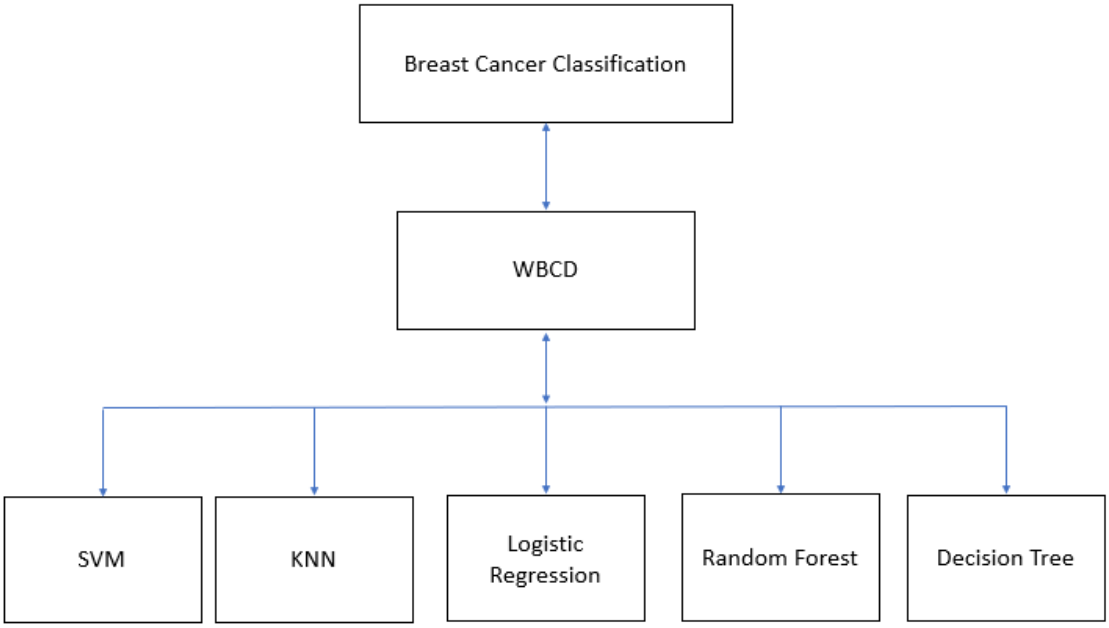

Figure 3.1: Machine Learing Algorithms

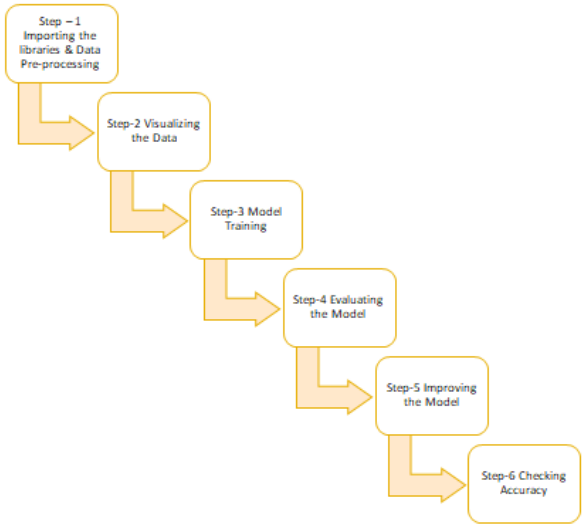

Figure 3.2: Steps while doing the Classification

Chapter 4

Methodology

The learning process of machine learning techniques can be divided into two main categories, supervised learning and unsupervised learning. In supervised learning, a set of data instances are used to train the machine and are labeled to give the correct result. However, in unsupervised learning, there are no predetermined data sets and no notion of the expected outcome, which means that the goal is harder to achieve.

Classification is among the most common methods that go under supervised learning. It uses historical labeled data to develop a model that is then used for future predictions. In the medical field, clinics and hospitals maintain large databases that contain records of patients with their symptoms and diagnosis. Therefore, researchers make use of this knowledge to develop classification models that can make an inference based on historical cases. The medical inference has, therefore, become a much simpler task with machine-based support using the sheer amount of medical data that is available today. Depending on the user data and their availability. In this report, we will see some classification and neural network algorithms to predict breast cancer.

4.1 SVM(Support Vector Machines)

Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. It is an effective statistical learning method for classification. A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by

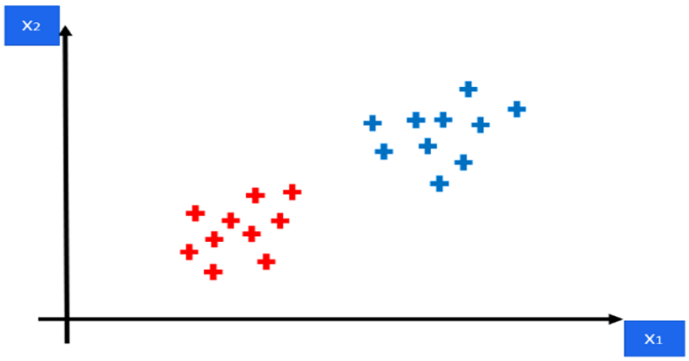

Figure 4.1: Malignant and Benign Case Points

their class, either class 0 or class 1. In two-dimensions, you can visualize this as a line and let's assume that all of our input points can be completely separated by this line.

The SVM learning algorithm finds the coefficients that result in the best separation of the classes by the hyperplane. The distance between the hyperplane and the closest data points is referred to as the margin. The best or optimal hyperplane that can separate the two classes is the line that has the largest margin. Only these points are relevant in defining the hyperplane and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane. In practice, an optimization algorithm is used to find the values for the coefficients that maximize the margin. SVM might be one of the most powerful out-of-the-box classifiers and worth trying on our dataset.

SVM has the advantage of fast training technique, even with a large number of input data. Therefore it has been used for many recognition problems such as object recognition and face detection.

SVM was initially developed in the 1960s then they have refined again in the 1990s and only now they're becoming very popular in machine learning because they are demonstrating that they can be very very popular because they are somewhat different to other machine learning algorithms.

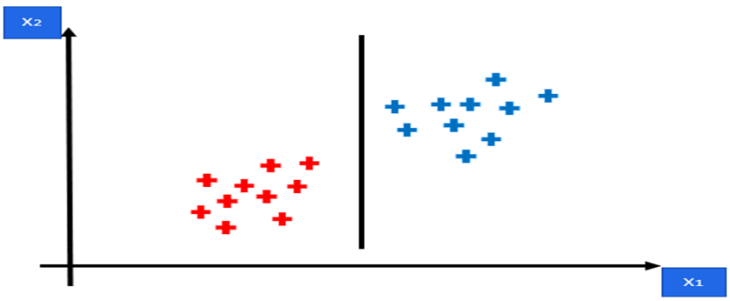

Figure 4.2: Adding Seperation

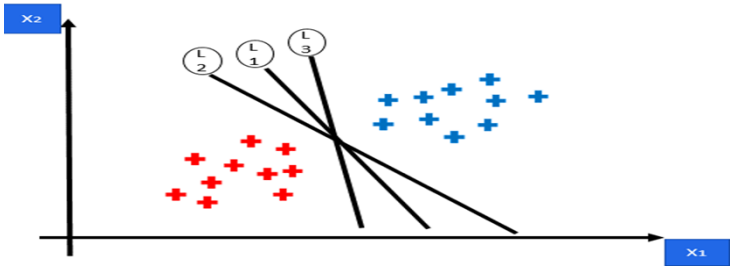

Figure 4.3: Adding more seperations

Here we've got as usual points on two-dimensional space for simplicity sake. We've got just two columns x1 and x2. And we've got some observations some already are red and some are green. So we've already classified them but now how do we derive a line that's going to separate them. So how do we actually separate those points? Because that's a separation or in other words that decision boundary or going to be very important for us. When we start adding new points. So that's the point of our classification. That's the purpose of our classification we want to create a boundary between these two so that when we in the future add new points that we want to classify haven't been classified yet. We will know where they will fall either in the green area or in the red area. So how can we separate these points when we see here.(see fig:4.1)

Well one way is to draw a line like L1 in our two-dimensional space and then say anything to the right will be green anything to the left will be red and if a new point falls somewhere on this space we will know where it falls.(see fig:4.2)

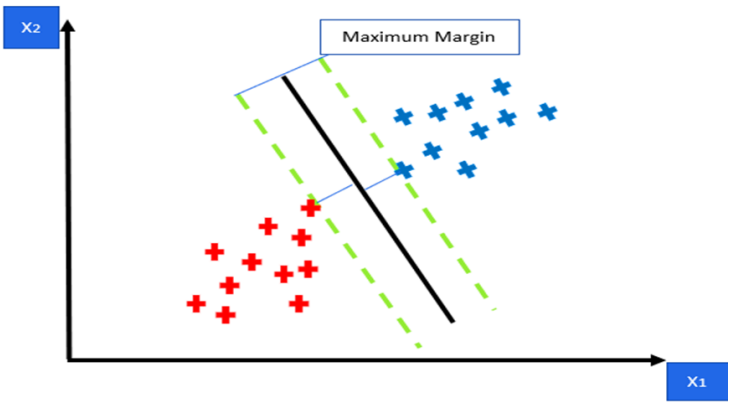

Figure 4.4: Maximum Margin

However, there's another way we can draw a horizontal line L2 like above or we can draw a diagonal line like L3, or we can actually draw another diagonal line L4 or we can draw a diagonal line like L5. So there's a lot of different lines that we can create that will achieve the same result they'll separate our points to two classes.

But at the same time, they all in the future will have different consequences so when we add new points depending on where that point will fall it'll either be classed as a part of the f=green zone and red zone or we want to find the optimal line and that's what SVMs are all about, finding the best line or the best decision boundary which will help us separate our space into classes. Let's see how the SVM actually searches for this line. (see fig:4.3)

Maximum Margin Line:

Well, the line is searched through the maximum margin. So here you can see a line and this is the line and would draw.(see fig:4.4)

So basically it's the line that separates those two classes of points. And at the same time, it has the maximum margin which means his distance. So this line is drawn equidistant from this red point and this green point and we'll find out exactly why these points in second. And the distance between the line and each one of these points that's equidistant. And that's margins. So the sum of these two distances has to be maximized in order for this line to be the result of the

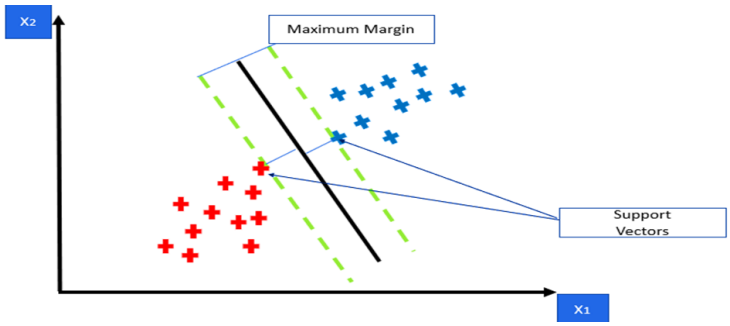

Figure 4.5: Identifying Support Vectors

SVM. And the two points which are actually dotted lines are called vectors. So basically that these dotted two lines are supporting this whole algorithm. So even if you change the algorithm will be exactly the same. So that the other points they don't contribute to the results of the algorithm only these dotted two points are contributing and therefore they are called support vectors.(see fig:4.5)

Why those are calling supporting vectors? Because in a multidimensional space when we have more than just two variables or we can have three, five, ten or hundred variables. Each point is actually no longer point because you can't visualize it on a two-dimensional plane or even a three-dimensional space and therefore each of those points that we see here is considered is actually a vector in a multi-dimensional space so the more general term for points that we see here are vectors and this is something that is studied in mathematics in university or high school mathematics basically.

So generally speaking they are all vectors just in this particular example and we have two dimensions then we call them points but in reality, there are pictures and that's why they're called support vectors. So hence those specific vectors are the one's supporting kind of this decision boundary or this way we're building this algorithm that's why they're important and that's the whole algorithm is called support vector machines.

Hyperplanes:

Well, we've to go the line in the middle which is called the maximum margin hyperplane or the maximum margin classifier. So in a two-dimensional space, it's

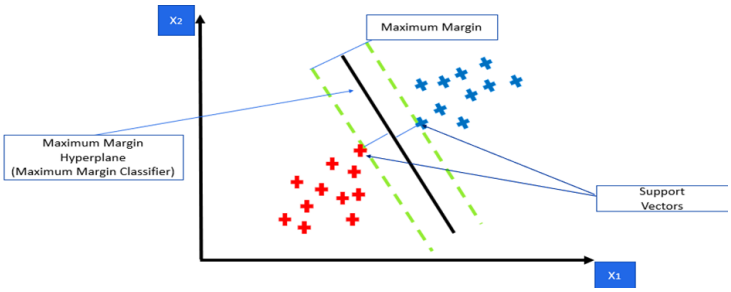

Figure 4.6: Hyperplanes

just lien a classifier is just the line. But actually, in a multidimensional space, it's a hyperplane. It's also called as maximum margin hyperbola. So those all of the ones that we saw were also hyperplane but there weren't the maximum margin hyperplanes and we check that yourself. So we can draw a different hyperplane there and just check out the marginal. It'll always be less because this is the one with the maximum margin. And then you've got the green and the red dotted lines. So the green one is called positive hyperplane and the red ones were called the negative hyperplane. It doesn't really matter in which order you name them just the point is he one of them is positive and negative or basically anything to the right of the positive is classified as the green category or the positive category anything to the left to classify as a negative category or the red category in our case. So that's how the supervised machine learning algorithm world but there's some complicated mathematics behind it but the essence of the intuitive part of it is exactly this. That was working with a linearly separable data set where we can actually it's given to us by default that we can put a line through a chart which will separate the two categories and then we're just searching for the one with the maximum margin. So conceptually when we think about it it's actually a pretty simple algorithm when we think about it this way.(see fig:4.6)

What's so special about SVMs? And why are they different from other machine learning algorithms? So imagine we're trying to teach a machine. How to distinguish between apples and oranges how to classify a fruit into either an apple an orange. So we're telling a machine that, all right we're going to give some test data. So have look at all these apples. These are apple oranges. Analyze them, look at them see what parameters they have and then next time they're going to give you. I'm going to give you a fruit which will be better an apple an orange and we're going to need to classify it and tell whether it's an apple or an

orange. So that's kind of a machine learning problem. Now in our case here we can see let's say on the right we have oranges on the left we have apples. So what predominately ml would do is they would look at the most apples the apple and the most oranges the orange. So they would look the most stock standard common type of apples and the most stock standard common type of oranges and our case would be apple some more there in that in the very heart of the apple class far away from the oranges. And for the oranges would be somewhere over there. So also in the very heart of the orange class far away from the apple so they were tried. A machine would try to learn from the apples that are very like apples so it would know what an apple is and it also tried to learn from oranges so it would know what an orange actually is and that's how most of the ml algorithms work and then based on that it would be able to come up with some predictions and classifying for new data elements and variables that we would get it in the case of support vector machine. It's a bit different. Instead of looking at the most stocks standard apples and stocks and oranges what these support vector machines do are they actually look at the apples that are very much like an orange. So here we can see an apple which is not your standard apple is orange and color right. So it's very easy to infuse the apple of orange and they would look at the oranges which are not stock standard oranges which are more like apples than anything else so we can order the lemon here. So those of us in the image just out of the oranges the SVM would pick the one that is that looks the most like an apple in this case. We have a green orange. It's not normal to have a green orange when we think of orange. So what that is those are the support vectors we can see that they're actually very close to the boundary. So they're very close to the apple or the red one would be very close to the green ones and the orange or the green mark there would be very close to the red ones and therefore the support vector machine in that sense we can think of it is like a more extreme type of algorithm a very rebellious type of algorithm a very risky type of algorithm because it looks at very extreme case which is very close to the boundary and it uses that to construct its analysis. That itself makes the SVM algorithm very special, very different to most of the other ml algorithms. That's why at times they perform much better than non-supported of vector machine algorithms.

4.2 KNN(K-Nearest Neighbors)

The model representation for KNN is the entire training dataset.Predictions are made for a new data point by searching through the entire training set for the K

most similar instances (the neighbors) and summarizing the output variable for those K instances. For regression problems, this might be the mean output variable, for classification problems this might be the mode (or most common) class value. The trick is in how to determine the similarity between the data instances. The simplest technique if your attributes are all of the same scale (all in inches for example) is to use the Euclidean distance, a number you can calculate directly based on the differences between each input variable. KNN can require a lot of memory or space to store all of the data, but only performs a calculation (or learn) when a prediction is needed, just in time. You can also update and curate your training instances over time to keep predictions accurate.

The idea of distance or closeness can break down in very high dimensions (lots of input variables) which can negatively affect the performance of the algorithm on your problem. This is called the curse of dimensionality. It suggests you only use those input variables that are most relevant to predicting the output variable.

So let's imagine that we have a scenario where we have two categories already present in our data set. So we've identified two categories and one is category one on the left which is red color is a benign case, category two is green is the right. And for simplicity's sake, we're just going to take into consideration two variables or two columns in our data set. So all of his groupings is happening based on these columns x1 and x2. Now let's say we add a new data point out data set. The question is should it fall into the red category or should fall to the green category. How do we decide that? So how do we classify this new data point to a cluster of first as a red data point or a green data point and that's where the nearest neighbors algorithm will come to assist us.

At the end of performing this algorithm, we 'll be able to identify whether it's a red or green point and in this case, the point turned out to be red. So how does the K nearest neighbor algorithm work? How did it do that? Well, we're going to build a step by step rule guides to the k and then and after we've built it we're going to actually perform it manually to see how it works. We'll see is a very very simple algorithm.

Step-1 Choose the number k of neighbors.

step-2 Take the K nearest neighbors of the new data point, according to the Euclidean distance. Other distance such as a Manhattan distance or any other

distances that we might be considering. But in most cases, Euclidean distance is so we're to stick to those.

Step-3 Among these K neighbors. Count the no.of data points in each category. So how many data points fell into one category to the other category and so on if we might even have more than two categories in our data set. So we just need to calculate how many fall into each category.

Step-4 Assign the new data point to the category where you counted the most neighbors. As simple as that's why it's called K-nearest Neighbors. And then your model is ready as it is it's a very simple algorithm and moral which is going to do a manual understanding to really solidify this knowledge. description

Here we've got the new data point has been added tour scatterplot as we saw previously. How do we find the nearest neighbors of this new data point? Well have a look at the Euclidean distance that we're going to use so quickly and distance is a very basic type of distance that we define in geometry it'ss the one we use in geometry.

Basically, if we have two points over here one and two then the distance between the two points is measured according to this formula.Euclidean Distance between P1 and P2 =

$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Here on our data point how are going to identify which other closest five neighbors. We just look at them and seeing the distances, identify the closest ones, second closest, and up to fifth closest.

Step 3 among these K neighbors count the no.of data points in each category is in category 1 in the red when we have 3 neighbors on category 2 we have 2 neighbors.

Step 4 is assigned the new data point to the category where you counted the most neighbors. We need to assign it to the red category as simple as that. Now we have classified this new point and our model is ready. It's a very straight forward algorithm. One of the simplest we can thinks about it.

4.3 Logistic Regression

Logistic regression is another technique borrowed by machine learning from the field of statistics. It is the go-to method for binary classification problems (problems with two class values).

Logistic regression is like linear regression in that the goal is to find the values for the coefficients that weight each input variable. Unlike linear regression, the prediction for the output is transformed using a non-linear function called the logistic function.

The logistic function looks like a big S and will transform any value into the range 0 to 1. This is useful because we can apply a rule to the output of the logistic function to snap values to 0 and 1 (e.g. IF less than 0.5 then output 1) and predict a class value.

Because of the way that the model is learned, the predictions made by logistic regression can also be used as the probability of a given data instance belonging to class 0 or class 1. This can be useful for problems where you need to give more rationale for a prediction.

Like linear regression, logistic regression does work better when you remove attributes that are unrelated to the output variable as well as attributes that are very similar (correlated) to each other.

It's a fast model to learn and effective on binary classification problems.

4.4 Random Forest

Similar to how a jury of people is used to make a court decision, Random Forest brings together many decision trees to ensemble a forest of trees. The argument used is that having a single decision tree can either provide a simple model or a very specific one. Using Random Forest results in increased stability as compared to using single decision trees. This indicates that Random Forest is insensitive to the noise of the input data set. One of the primary reasons behind using Random Forest in cancer detection is its ability to handle data minorities. For example,

a tumor can be classified as either benign or malignant, despite the later class is only 25 percent of the input data set.

The Random Forest method is based on a recursive approach in which every iteration involves picking one random sample of size N from the data set with replacement, and another random sample from the predictors without replacement. Then the data obtained is partitioned. The out-of-bag data is then dropped and the above steps repeated many times depending on how many trees are needed. Finally, a count is made over the trees that classify the observation in one category and in the other. Cases are then classified based on a majority vote over the decision trees.

4.5 Decision Tree

Decision trees are supervised algorithms which recursively partition the data based on its attributes until some stopping condition is reached Decision Tree Classifier (DTC) is one of the possible approaches to multistage decision-making. The most important feature of DTCs is their capability to break down a complex decision-making process into a collection of simpler decisions, thus providing a solution, which is often easier to interpret.

The classification and regression trees (CART) methodology proposed is perhaps best known and most widely used. CART uses cross-validation or a large independent test sample of data to select the best tree from the sequence of trees considered in the pruning process. The basic CART building algorithm is a greedy algorithm in that it chooses the locally best discriminatory feature at each stage in the process. This is suboptimal but a full search for a fully optimized set of question would be computationally very expensive. The CART approach is an alternative to the traditional methods for prediction. In the implementation of CART, the dataset is split into the two subgroups that are the most different with respect to the outcome. This procedure is continued on each subgroup until some minimum subgroup size is reached.

4.6 Improving the models

The model development cycle goes through various stages, starting from data collection to model building. we believe this is the most under – rated step of predictive modeling.

It is important that you spend time thinking on the given problem and gaining the domain knowledge. So, how does it help?This practice usually helps in building better features later on, which are not biased by the data available in the data-set. This is a crucial step which usually improves a model's accuracy.

At this stage, you are expected to apply structured thinking to the problem i.e. a thinking process which takes into consideration all the possible aspects of a particular problem.Let's dig deeper now. Now we'll check out the proven way to improve the accuracy of a model:

4.6.1 Algorithm Tuning

We know that machine learning algorithms are driven by parameters. These parameters majorly influence the outcome of learning process.

The objective of parameter tuning is to find the optimum value for each parameter to improve the accuracy of the model. To tune these parameters, you must have a good understanding of these meaning and their individual impact on model. You can repeat this process with a number of well performing models.

Data Normalization

Why Data Normalization is necessary for Machine Learning models Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. It is required only when features have different ranges.

Grid Search

Grid-searching is the process of scanning the data to configure optimal parameters for a given model. Depending on the type of model utilized, certain parameters are necessary. Grid-searching does NOT only apply to one model type. Grid-searching can be applied across machine learning to calculate the best parameters to use for any given model. It is important to note that Grid-searching can be extremely computationally expensive and may take your machine quite a long time to run. Grid-Search will build a model on each parameter combination possible. It iterates through every parameter combination and stores a model for each combination.

4.7 Sample Code

```
#### Step-1 Improting the libraries and Data
    Preprocessing ####
# importing the libraries
# Import Pandas for data manipulation using dataframes
import pandas as pd
# Import Numpy for data statistical analysis
import numpy as np
# Import matplotlib for data visualisation
import matplotlib.pyplot as plt
# Statistical data visualization
import seaborn as sns

# Import Cancer data drom the Sklearn library
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()

cancer

cancer.keys()

print(cancer['DESCR'])

print(cancer['target_names'])
```

```
print(cancer['target'])

print(cancer['feature_names'])

print(cancer['data'])

cancer['data'].shape

df_cancer = pd.DataFrame(np.c_[cancer['data'], cancer[
    'target']], columns = np.append(cancer['
    feature_names'], ['target']))
df_cancer.head()
df_cancer.tail()

#### Step-2 Visualzing the Data ####
# Generating the pairplot for the mean 1 to 5 features
sns.pairplot(df_cancer, hue='target', vars=['mean
    radius', 'mean texture', 'mean area', 'mean
    perimeter', 'mean smoothness'])
# Generating the pairplot for the mean 6 to 10
    features
sns.pairplot(df_cancer, hue='target', vars=['mean
    compactness', 'mean concavity','mean concave points
    ','mean symmetry','mean fractal dimension',])
# Generating the pairplot for the error 1 to 5
    features
sns.pairplot(df_cancer, hue='target', vars=['radius
    error', 'texture error', 'perimeter error', 'area
    error','smoothness error'])
# Generating the pairplot for the error 6 to 10
    features
sns.pairplot(df_cancer, hue='target', vars=['
    compactness error', 'concavity error', 'concave
    points error', 'symmetry error','fractal dimension
    error'])
# Generating the pairplot for the worst 1 to 5
    features
```

```
sns.pairplot(df_cancer, hue='target', vars=['worst_
    radius', 'worst_texture', 'worst_perimeter', 'worst
    _area', 'worst_smoothness'])
# Generating the pairplot for the worst 6 to 10
    features
sns.pairplot(df_cancer, hue='target', vars=['worst_
    compactness', 'worst_concavity', 'worst_concave_
    points', 'worst_symmetry', 'worst_fractal_dimension
    '])

# Genarating the count plot for the target attribute
    to
# identify the tumors
sns.countplot(df_cancer.target, label = 'count')

# Generating the correlation for the mean features
features_mean= list(df_cancer.columns[0:10])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_mean].corr(), annot=
    True, square=True)

# Generating the correlation for the error features
features_error= list(df_cancer.columns[10:20])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_error].corr(), annot=
    True, square=True)

# Generating the correlation for the worst features
features_worst= list(df_cancer.columns[20:30])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_worst].corr(), annot=
    True, square=True)

#### Step-3 Model Trianing ####
# Let's drop the target label coloumns
# see the target column was not there
X = df_cancer.drop(['target'], axis = 1)
X.head(5)
```

```
# passing the target attribute to the new variable
y = df_cancer['target']
y.head(5)

# Spliting the dataset into training set and testing
    set
# for that we required model selection library
from sklearn.model_selection import *
X_train, X_test, y_train, y_test = train_test_split(X,
    y, test_size = 0.25, random_state = 5)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

###### Let's implement using this into Support Vector
    Classifier using Support Vector Machines
# implement using this into Support Vector Classifier
    using Support Vector Machines
from sklearn.svm import SVC
from sklearn.metrics import *
svc_model = SVC()
svc_model.fit(X_train, y_train)

#### Step-4 Evaluating the model ####
# For evaluating the model we need to declare the new
    variable
# y_predict
y_predict = svc_model.predict(X_test)
# Creating the confusion matrix
Labels= ['Benign', 'Malignant']
cm = confusion_matrix(y_test, y_predict)
sns.heatmap(cm, xticklabels=Labels, yticklabels=Labels
    ,annot= True, square=True)
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
```

```
plt.show()

# Checking the Accuracy Score
print('accuracy is',accuracy_score(y_predict,y_test))
# Cheking the report
print(classification_report(y_test, y_predict))

#### Step-5 Improving the model ####
min_train = X_train.min()
min_train

range_train = (X_train - min_train).max()
range_train

X_train_scaled = (X_train - min_train) / range_train
X_train_scaled.head()

sns.scatterplot(x = X_train['mean area'], y = X_train[
    'mean smoothness'], hue = y_train)

sns.scatterplot(x = X_train_scaled['mean area'], y =
    X_train_scaled['mean smoothness'], hue = y_train)

min_test = X_test.min()
min_test

range_test = (X_test - min_test).max()
range_test

X_test_scaled = (X_test - min_test) / range_test
X_test_scaled.head()

from sklearn.svm import SVC
from sklearn.metrics import *
svc_model = SVC()
svc_model.fit(X_train_scaled, y_train)

y_predict = svc_model.predict(X_test_scaled)
```

```
Labels = ['Benign', 'Malignant']
cm = confusion_matrix(y_test,y_predict)
sns.heatmap(cm, xticklabels=Labels, yticklabels=Labels
    ,annot= True, square=True)
plt.title("Confusion Matrix")
plt.ylabel("True Class")
plt.xlabel('Predicted Class')
plt.show()

print('accuracy is',accuracy_score(y_predict,y_test))

print(classification_report(y_test, y_predict))

#### Improving the model-part 2###

param_grid = {'C':[0.1,1,10,100], 'gamma'
    :[1,0.1,0.01,0.001], 'kernel':['rbf']}
from sklearn.model_selection import GridSearchCV
grid = GridSearchCV(SVC(), param_grid, refit = True,
    verbose = 4)
grid.fit(X_train_scaled, y_train)

grid.best_params_
grid_predict = grid.predict(X_test_scaled)

y_predict = svc_model.predict(X_test_scaled)
Labels = ['Benign', 'Malignant']
cm = confusion_matrix(y_test,grid_predict)
sns.heatmap(cm, xticklabels=Labels, yticklabels=Labels
    ,annot= True, square=True)
plt.title("Confusion Matrix")
plt.ylabel("True Class")
plt.xlabel('Predicted Class')
plt.show()

print(classification_report(y_test, grid_predict))
```

```
print('accuracy is',accuracy_score(grid_predict,y_test
    ))

#### Step-1 Improting the libraries and Data Preprocessing ####
# importing the libraries
# Import Pandas for data manipulation using dataframes
import pandas as pd
# Import Numpy for data statistical analysis
import numpy as np
# Import matplotlib for data visualisation
import matplotlib.pyplot as plt
# Statistical data visualization
import seaborn as sns

# Import Cancer data drom the Sklearn library
from sklearn.datasets import load_breast_cancer
cancer = load_breast_cancer()

cancer

cancer.keys()

print(cancer['DESCR'])

print(cancer['target_names'])

print(cancer['target'])

print(cancer['feature_names'])

print(cancer['data'])

cancer['data'].shape

df_cancer = pd.DataFrame(np.c_[cancer['data'], cancer['target']], co
df_cancer.head()
df_cancer.tail()
```

```
#### Step-2 Visualzing the Data ####
# Generating the pairplot for the mean 1 to 5 features
sns.pairplot(df_cancer, hue='target', vars=['mean radius', 'mean tex
# Generating the pairplot for the mean 6 to 10 features
sns.pairplot(df_cancer, hue='target', vars=['mean compactness', 'mea
# Generating the pairplot for the error 1 to 5 features
sns.pairplot(df_cancer, hue='target', vars=['radius error', 'texture
# Generating the pairplot for the error 6 to 10 features
sns.pairplot(df_cancer, hue='target', vars=['compactness error', 'co
# Generating the pairplot for the worst 1 to 5 features
sns.pairplot(df_cancer, hue='target', vars=['worst radius', 'worst
# Generating the pairplot for the worst 6 to 10 features
sns.pairplot(df_cancer, hue='target', vars=['worst compactness', 'w

# Genarating the count plot for the target attribute to
# identify the tumors
sns.countplot(df_cancer.target, label = 'count')

# Generating the correlation for the mean features
features_mean= list(df_cancer.columns[0:10])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_mean].corr(), annot=True, square=True

# Generating the correlation for the error features
features_error= list(df_cancer.columns[10:20])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_error].corr(), annot=True, square=Tru

# Generating the correlation for the worst features
features_worst= list(df_cancer.columns[20:30])
plt.figure(figsize=(10,10))
sns.heatmap(df_cancer[features_worst].corr(), annot=True, square=Tru

#### Step-3 Model Trianing ####
# Let's drop the target label coloumns
# see the target column was not there
X = df_cancer.drop(['target'], axis = 1)
X.head(5)
```

```
# passing the target attribute to the new variable
y = df_cancer['target']
y.head(5)

# Spliting the dataset into training set and testing set
# for that we required model selection library
from sklearn.model_selection import *
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

###### Let's implement using k-Nearest Neighbors#####

from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import normalize
neighbors = []
cv_scores = []
from sklearn.model_selection import cross_val_score
#perform 10 fold cross validation
for k in range(1,10):
    neighbors.append(k)
    knn = KNeighborsClassifier(n_neighbors = k)
    scores = cross_val_score(knn,X_train,y_train,cv=10, scoring = '
    cv_scores.append(scores.mean())
print(np.round(cv_scores,3))
plt.plot(neighbors,cv_scores,color='blue')
plt.xlabel('k_values')
plt.ylabel('accuracy')
plt.show()

#Misclassification error versus k
MSE = [1-x for x in cv_scores]

#determining the best k
optimal_k = neighbors[MSE.index(min(MSE))]
```

```
print('The optimal number of neighbors is %d ' %optimal_k)

#plot misclassification error versus k

plt.figure(figsize = (10,6))
plt.plot(neighbors, MSE)
plt.xlabel('Number of neighbors')
plt.ylabel('Misclassification Error')
plt.show()

knn=KNeighborsClassifier(n_neighbors=optimal_k)
knn.fit(X_train,y_train)
#### Step-4 Evaluating the model ####
from sklearn.metrics import classification_report
y_pred = knn.predict(X_test)

print('accuracy is',accuracy_score(y_pred,y_test))

print(classification_report(y_test, y_pred))
# Generating the Confusion Matrix
Labels = ['Benign','Malignant']
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, xticklabels=Labels, yticklabels=Labels, an
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```

Chapter 5

Results

In the first step of importing the data. The process starts with importing the libraries like pandas, numpy, matplotlib, seaborn. The next process is importing the cancer data which already avilable in sklearn library loading the step as (from sklearn.datasets import load breast Cancer) load breast cancer. The next process is check the keys, description, targetnames, feature names, shape...etc. The next step load the data see the first some rows. If we can see the dataset description understanding the data by loading and processing the steps we can perform our result. The loading of the head function we will get the some rows is as follows:(see fig:5.1,5.2)

Same as the end of data also we need to check what's the data looks like, we need to load tail function as follows:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	peri
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	1
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	1
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	1
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	1

5 rows × 31 columns

Figure 5.1: Dataset Begining Values

```
df_cancer.tail()
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	pe
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	...	26.40	
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	...	38.25	
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648	...	34.12	
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...	39.42	
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05884	...	30.37	

5 rows × 31 columns

Figure 5.2: Dataset Ending Values

5.1 Visualizing the Data

Visualizing the Mean Data Firstly, in the data set we had 30 features which are mean, error and wosrt. Among them, mean has the attributes like mean radius, mean texture, mean area, mean perimeter, mean smoothness, mean compactness, mean concavity, mean concave points, mean symmetry, mean fractal dimension. This all representing as pair plot as follows:(see fig:5.3,5.4)

Visualizing the Error Data The error attibutes are radius error, texture error, perimeter error, area error, smoothness error, compactness error, concavity error, concave points error, symmetry error, fractal dimension error. This all representing as pair plot as follows:(see fig:5.5,5.6)

Visualzing the Worst Data The error attibutes are worst radius, worst texture, worst perimeter, worst area, worst smoothness, worst compactness, worst concavity, worst concave points, worst symmetry, worst fractal dimension. This all representing as pair plot as follows:(see fig:5.7,5.8)

Data The following figure will show how many Benign cases and Malignant case. This will be in countplot as follows: (See fig:5.9)

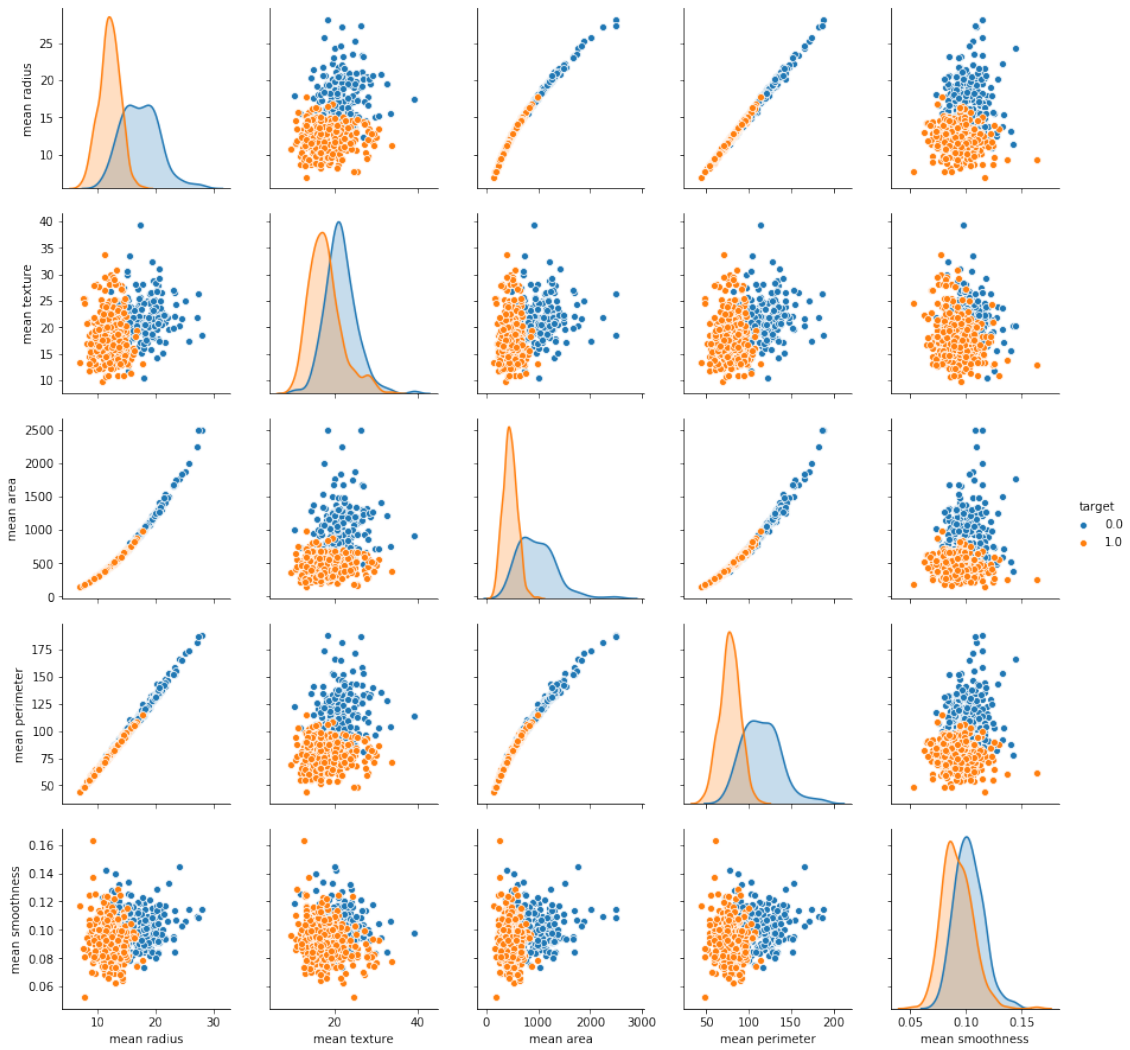

Figure 5.3: Mean Values of Pairplot-1

Figure 5.4: Mean Values of Pairplot-2

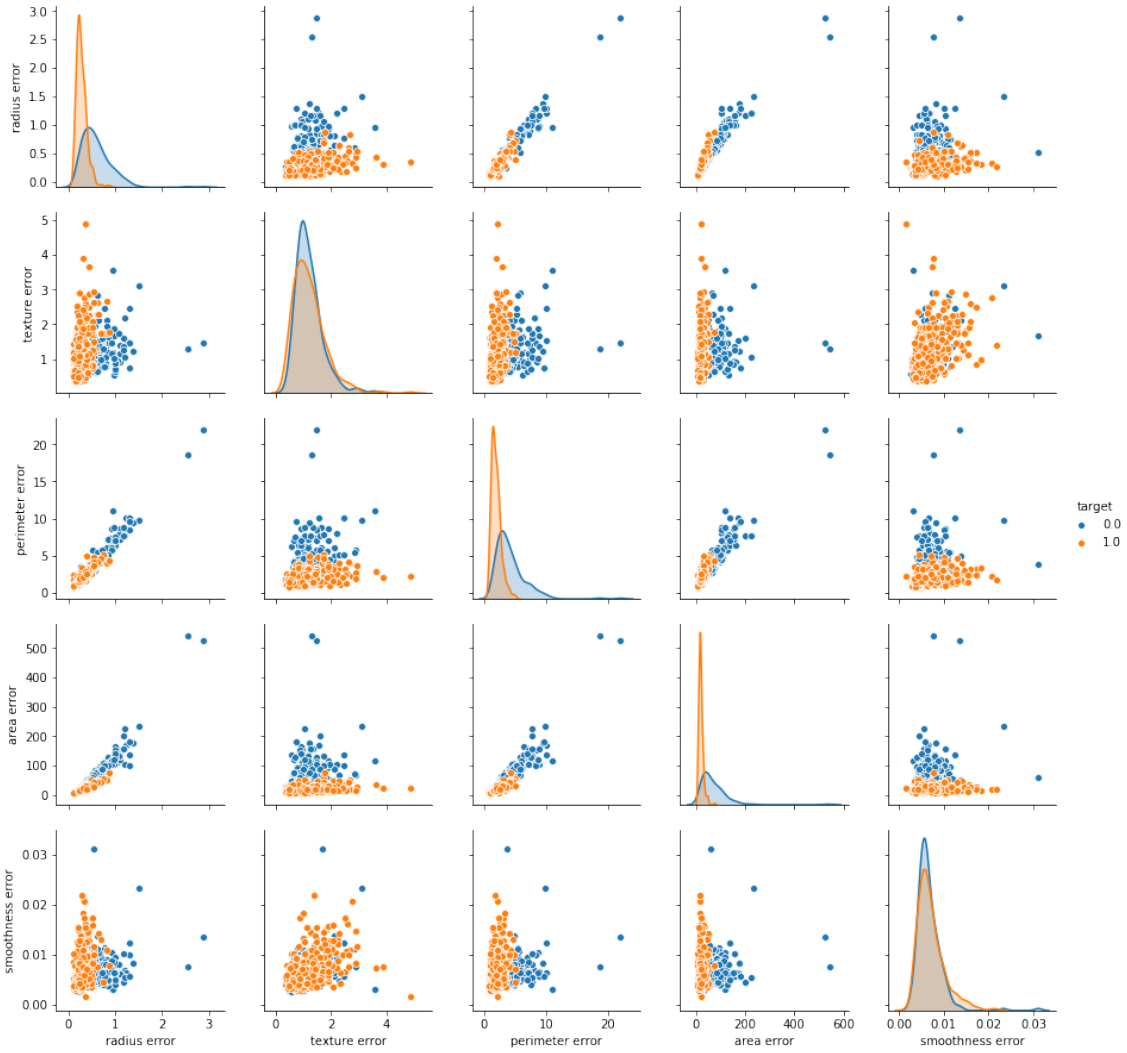

Figure 5.5: Error Values of Pairplot-1

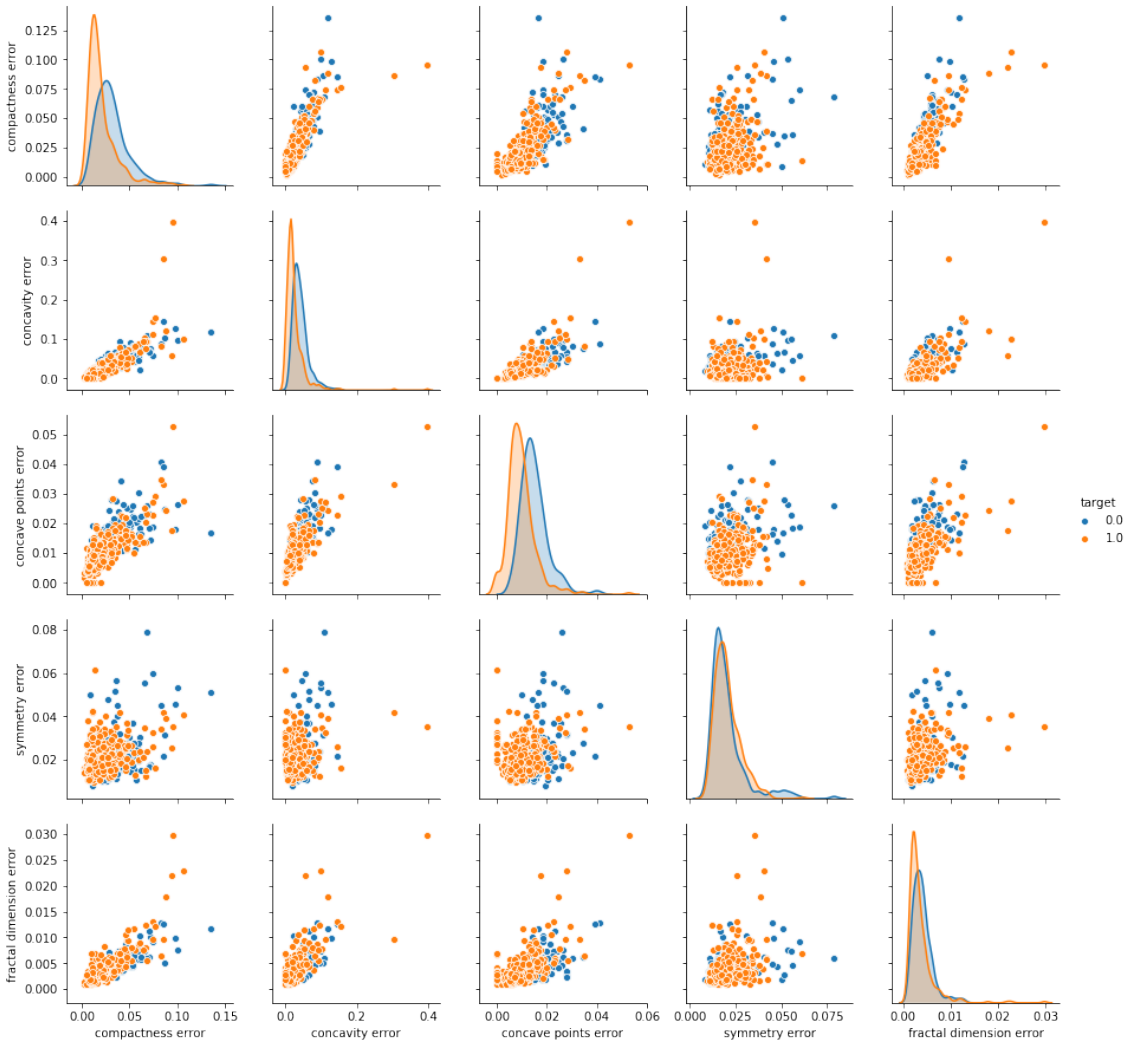

Figure 5.6: Error Values of Pairplot-2

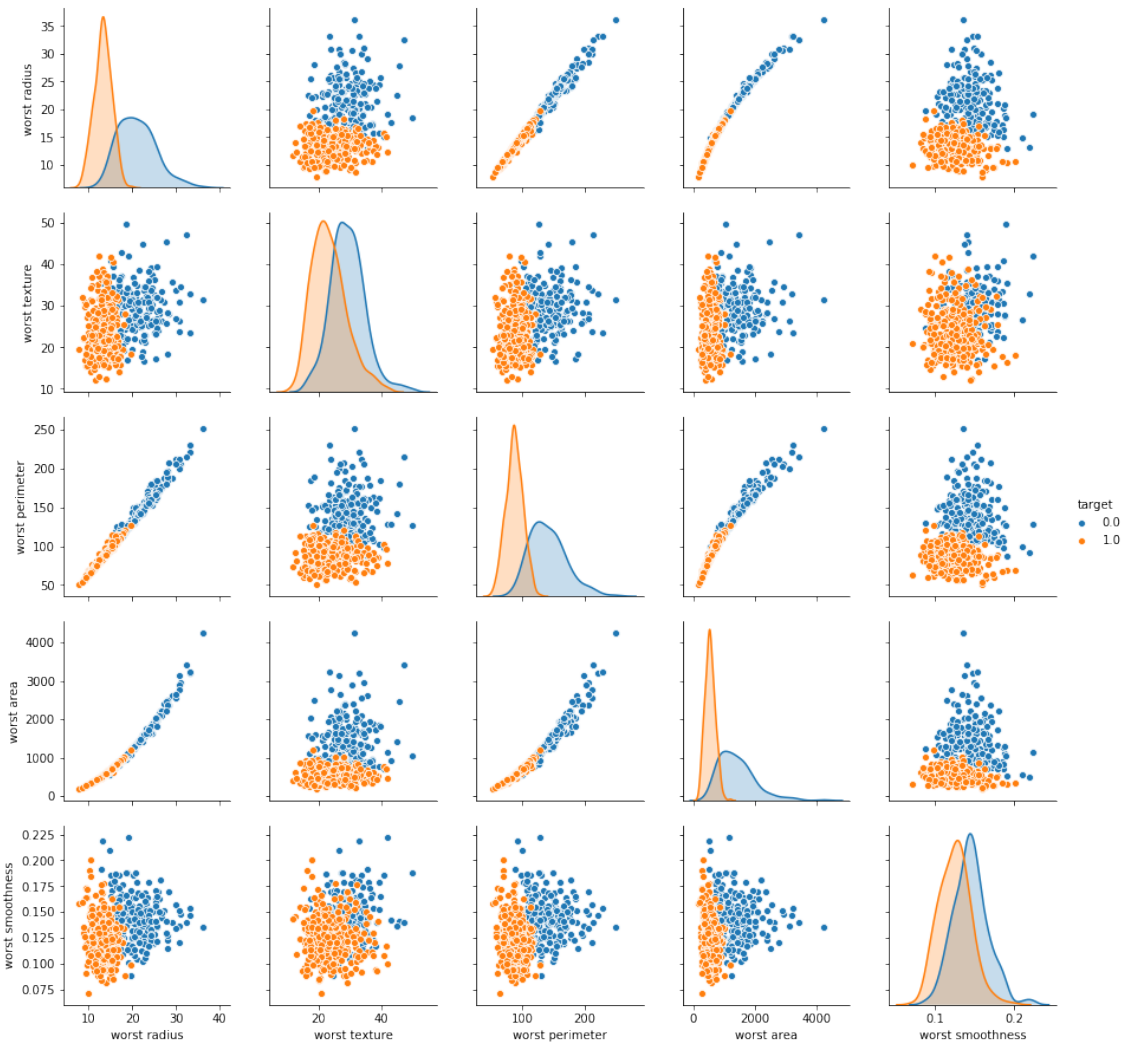

Figure 5.7: Worst Values of Pairplot-1

Figure 5.8: Worst Values of Pairplot-2

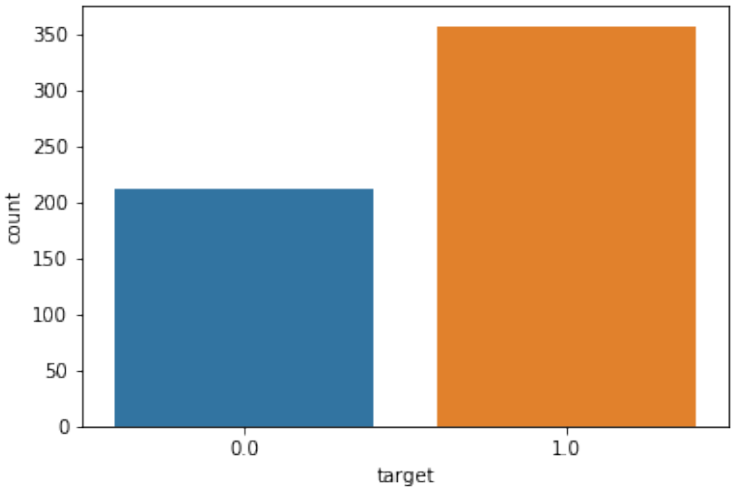

Figure 5.9: Countplot of the whole data

5.2 Correlation of Data

Correlation of Data It's objective is to be find the correlation between the variables. Here also we dividing the mean features, error features, worst feature. From this we will genetrate heatmap and we will find how many variaables are stongly correlated variables. This will be as follows: (see fig:5.10,5.11.5.12)

5.3 Model Training

Firstly in the dataset we added an extra feature called as target feature. Which it will deals with how many benign and malignant cases are their to differentiate. And it will also be called as preprocessor step. So, for model training we need to send it to another variable y. So the steps how involved is can be shown as follows: (see fig:5.13, 5.14, 5.15)

Spliting the data Second step of model training is dividing the data into training and testing sets. From this how many splits has been done. This will shown as follows:(see fig:5.16)

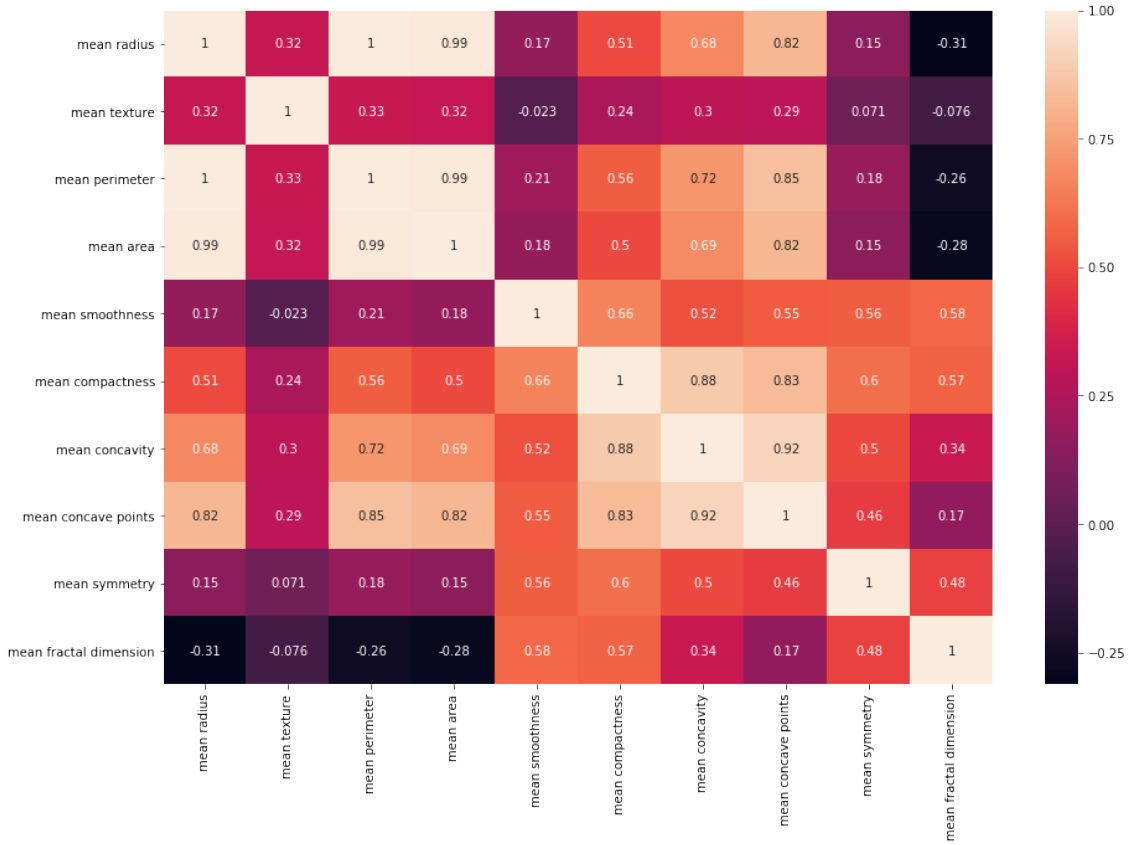

Figure 5.10: Heatmap of Mean Features

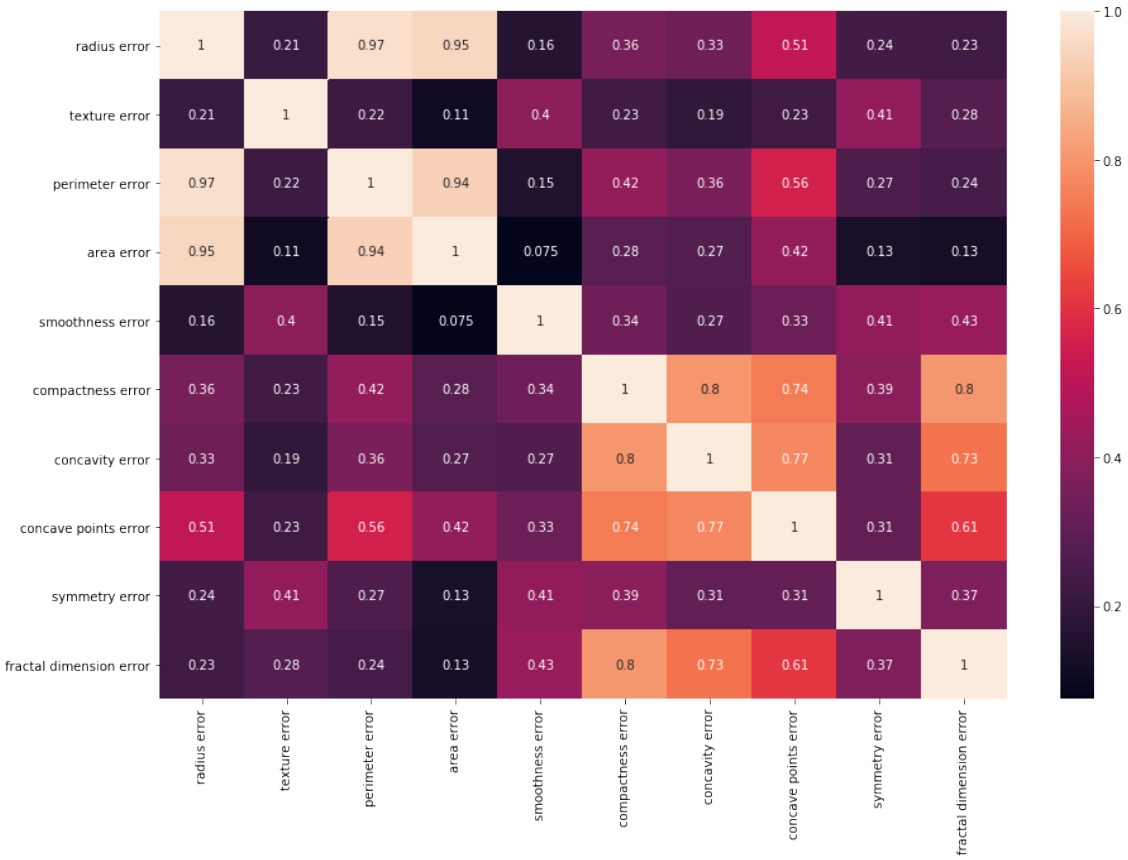

Figure 5.11: Heatmap of Error Features

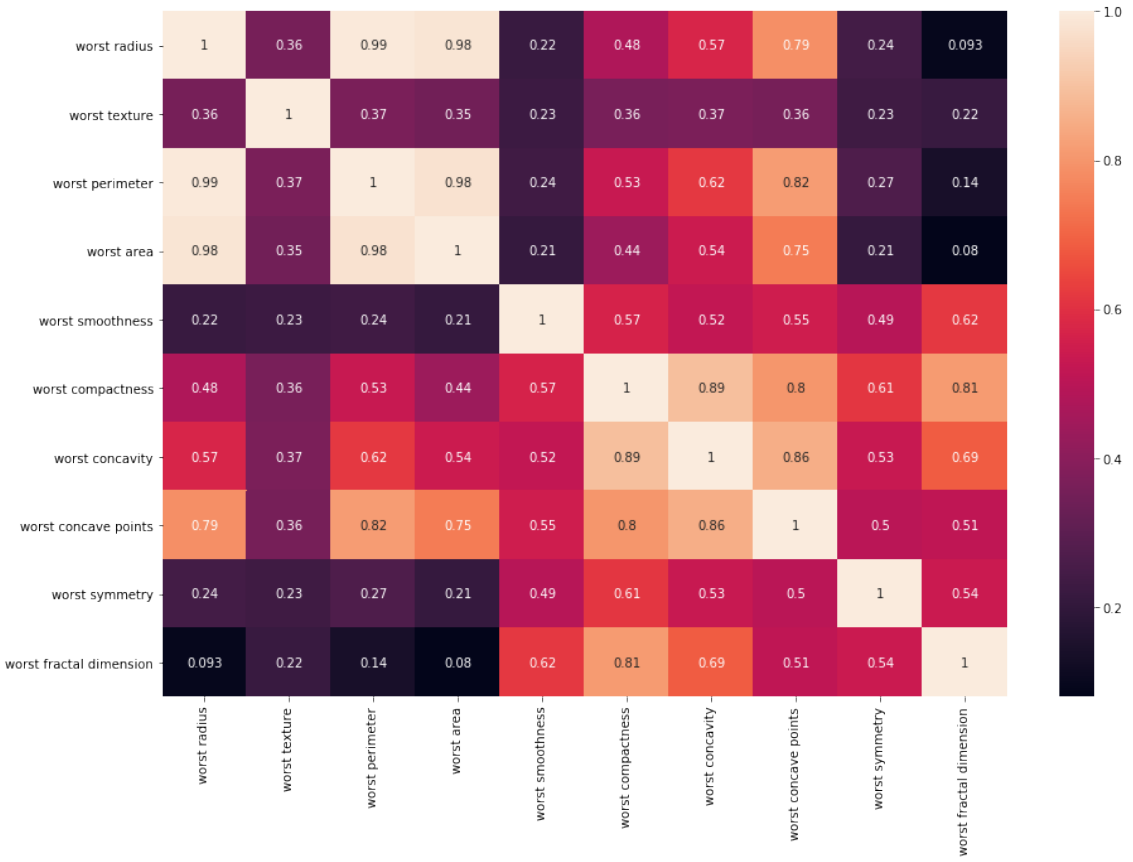

Figure 5.12: Heatmap of Worst Features

```
# Let's check the data from the first rows
# See the target attribute
df_cancer.head(5)
```

mean metry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	target
).2419	0.07871	...	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890	0.0
).1812	0.05667	...	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902	0.0
).2069	0.05999	...	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758	0.0
).2597	0.09744	...	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300	0.0
).1809	0.05883	...	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678	0.0

Figure 5.13: Data with Target Attribute

```
# Let's drop the target label coloumns
# see the target column was not there
X = df_cancer.drop(['target'], axis = 1)
X.head(5)
```

mean metry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
.2419	0.07871	...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
.1812	0.05667	...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
.1809	0.05883	...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

Figure 5.14: Data with no Target Attribute

```
# passing the target attribute to the new variable
y = df_cancer['target']
y.head(5)
```

```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
Name: target, dtype: float64
```

Figure 5.15: Target Attribute with new Variable

```
# Spliting the dataset into training set and testing set
# for that we required model selection library
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)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(426, 30)
(143, 30)
(426,)
(143,)
```

Figure 5.16: Spliting the data into Training & Testing data

```
# implement using this into Support Vector Classifier using Support Vector Machines
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
svc_model = SVC()
svc_model.fit(X_train, y_train)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Figure 5.17: Implementing the SVC trough SVM

```
# For evaluating the model we need to declare the new variable
# y_predict
y_predict = svc_model.predict(X_test)
# Creating the confusion matrix
Labels= ['Benign', 'Malignant']
cm = confusion_matrix(y_test, y_predict)
sns.heatmap(cm, xticklabels=Labels, yticklabels=Labels,annot= True, square=True)
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```

Figure 5.18: Evaluating the SVM model code

5.4 Support Vector Machine(SVM)

Implementation Using support vector classifier from support vector machines. (see Fig: 5.17)

5.4.1 Evaluating the model

After implementing the svm, we need to evaluate the model how many are classified into spelled and mispelled things in this SVM. (see fig:5.18, 5.19)

Accuracy of the model which is accurated as follows:

So, by seeing the model accuracy SVM parameters is not correctly classified. From that we need to improve the model. (see fig:5.20)

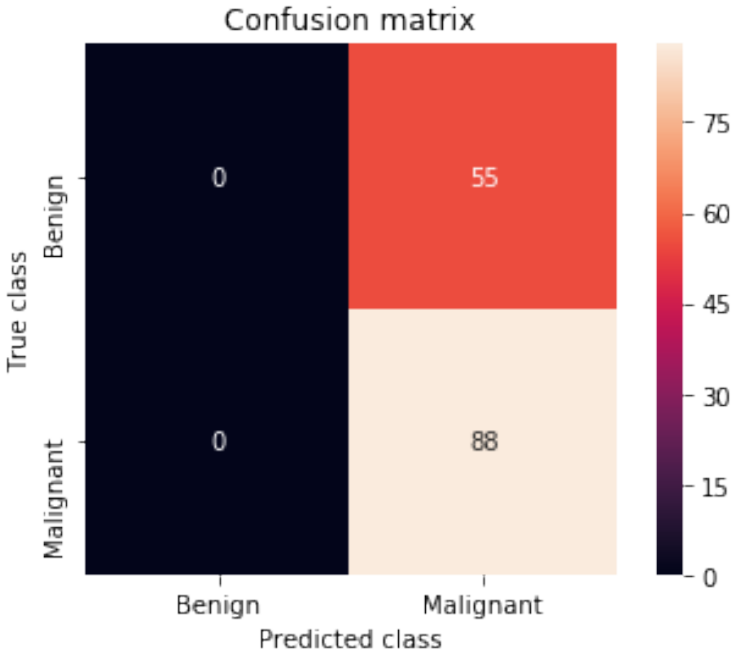

Figure 5.19: Evaluating the SVM model output

```
print('accuracy is',accuracy_score(y_predict,y_test))
```

```
accuracy is 0.6153846153846154
```

Figure 5.20: Accuracy of SVM model

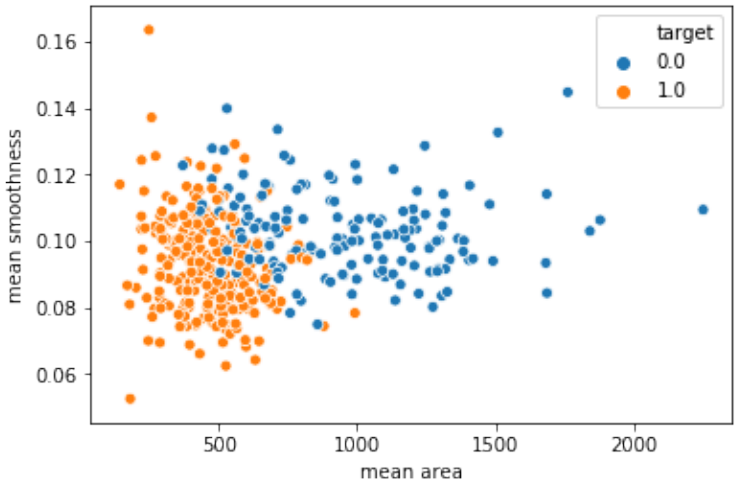

Figure 5.21: Before Applying Normalization

5.4.2 Improving the Model-1

In this we applying some normalization mechanism, because the data set values are not equal range. So that we are applying the normalization. For example, mean area and mean smmothness. In the mean area the values are in the range of 500,1000,1500,2000 and mean smoothness is less range of computation values like 0.06, 0.08, 0.10, 0.12, 0.14, 0.16. The difference of those by applying the normalization. We will see in the scatter plot as follows:(see fig:5.21, 5.22)

This is the first part of improving the model. After performing normarlization and evaluating the model, the result is as follows: (see fig:5.23,5.24)

After the first part of improving the model. Their is another chance of improving the model is also applicable for the above result. Because, when you see the accuracy score is not better accuracy. So for that we need improving the part model-2.

5.4.3 Improving the Model-2

Their is a another new strategy of improving the accuracy score is Grid Search CV. The results of this model and accuracy as follows: (see fig:5.25, 5.26)

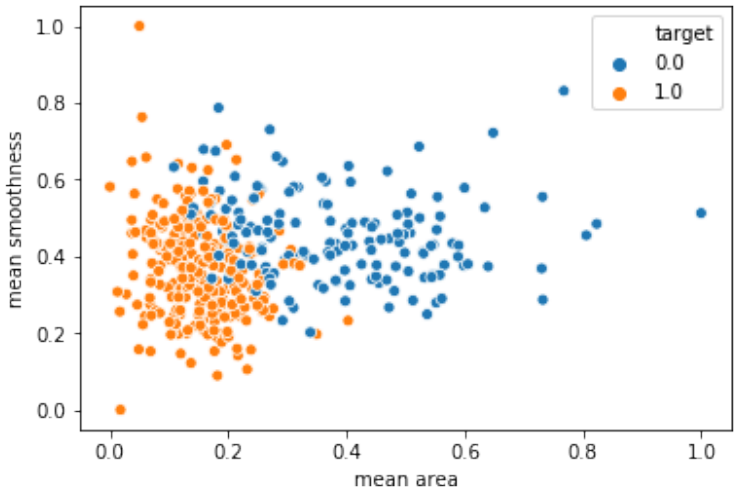

Figure 5.22: After Applying Normalization

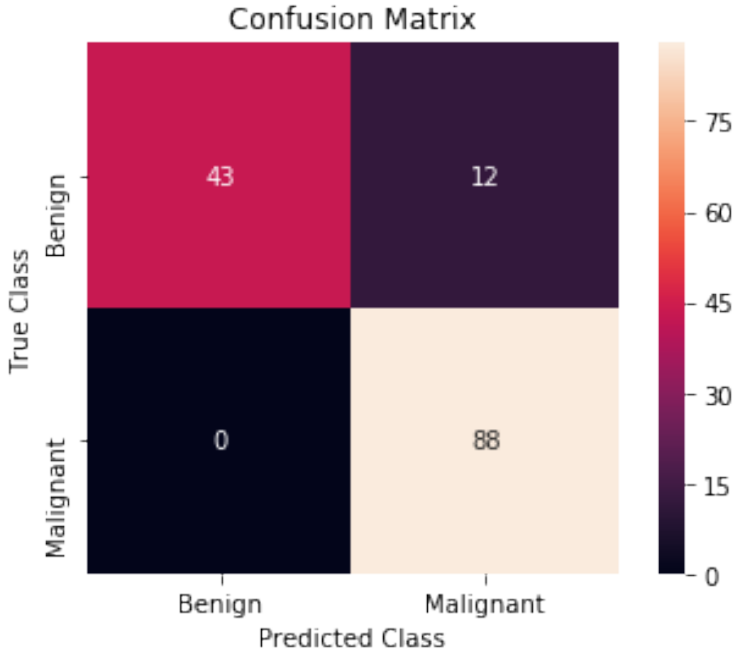

Figure 5.23: Applying Normalization Final Values

```
print('accuracy is',accuracy_score(y_predict,y_test))
```

```
accuracy is 0.916083916083916
```

Figure 5.24: Improving the part model-1 Accuaracy Score

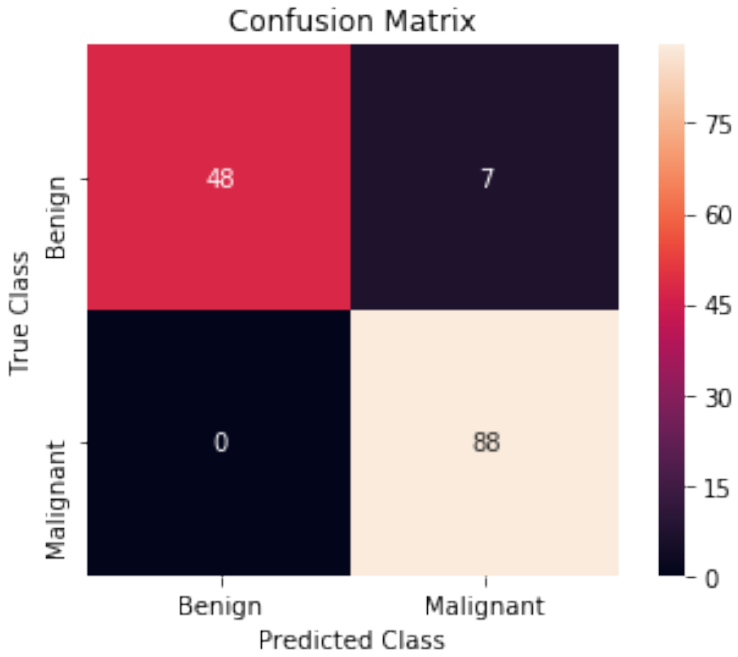

Figure 5.25: Applying GridsearchCV Final Values

```
print('accuracy is',accuracy_score(grid_predict,y_test))
```

```
accuracy is 0.951048951048951
```

Figure 5.26: GridSearchCV Accuaracy Score

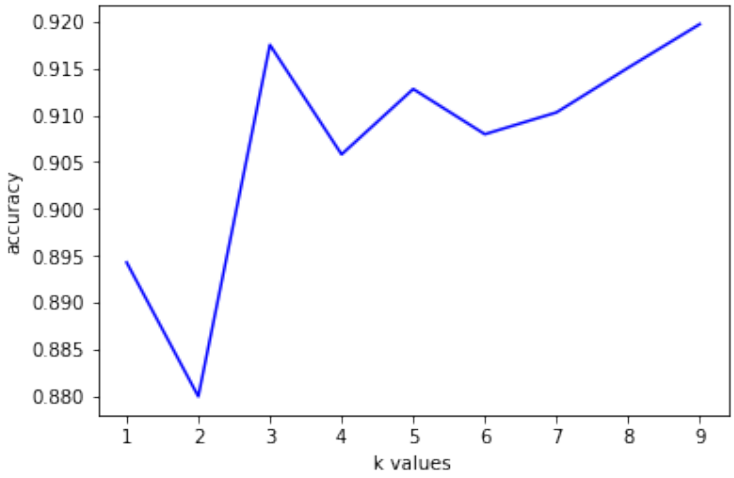

Figure 5.27: KNN performing 10 fold cross validation

5.5 K Nearest Neigbors(K-NN)

Cross-validation is when the dataset is randomly split up into 'k' groups. One of the groups is used as the test set and the rest are used as the training set. The model is trained on the training set and scored on the test set. Then the process is repeated until each unique group as been used as the test set.

Split the dataset into K equal partitions (or "folds") So if k = 10 and dataset has suppose some observations Each of the 10 folds would have some observations. (See fig:5.27, 5.28, 5.29)

Hypertuning model parameters using GridSearchCV When built our initial k-NN model, we set the parameter 'n_neighbors' to 3 as a starting point with no real logic behind that choice.

Hypertuning parameters is when you go through a process to find the optimal parameters for your model to improve accuracy. In our case, we will use GridSearchCV to find the optimal value for 'n_neighbors'.

GridSearchCV works by training our model multiple times on a range of parameters that we specify. That way, we can test our model with each parameter and figure out the optimal values to get the best accuracy results. (see fig:5.30)

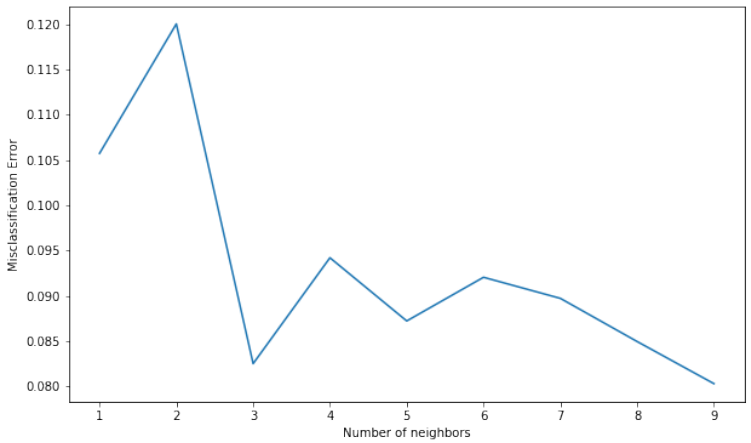

Figure 5.28: KNN plot misclassification error versus k

```
print('accuracy is',accuracy_score(y_pred,y_test))
```

```
accuracy is 0.951048951048951
```

Figure 5.29: KNN Accuracy Score

```
# Tuning hyper-parameters for accuracy
{'n_neighbors': 11}
0.923
# Tuning hyper-parameters for recall
{'n_neighbors': 35}
0.97
```

Figure 5.30: KNN GridSearchCV Score

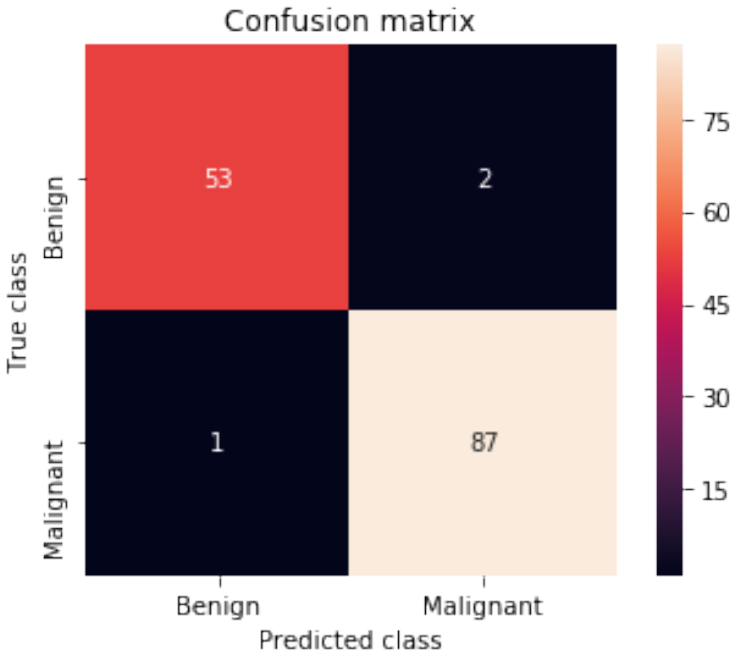

Figure 5.31: Logistic Regression Evaluation

5.6 Logistic Regression

In this we need to do the evaluating the model. By that we can conclude the results of logistic regression.

This model is been correclty given the predicted values. Only less number of misplled classes are there.(seefig:5.31)

```
print('Accuracy is',accuracy_score(y_predict,y_test))
```

```
Accuracy is 0.9790209790209791
```

Figure 5.32: Logistic Regression Accuracy Score

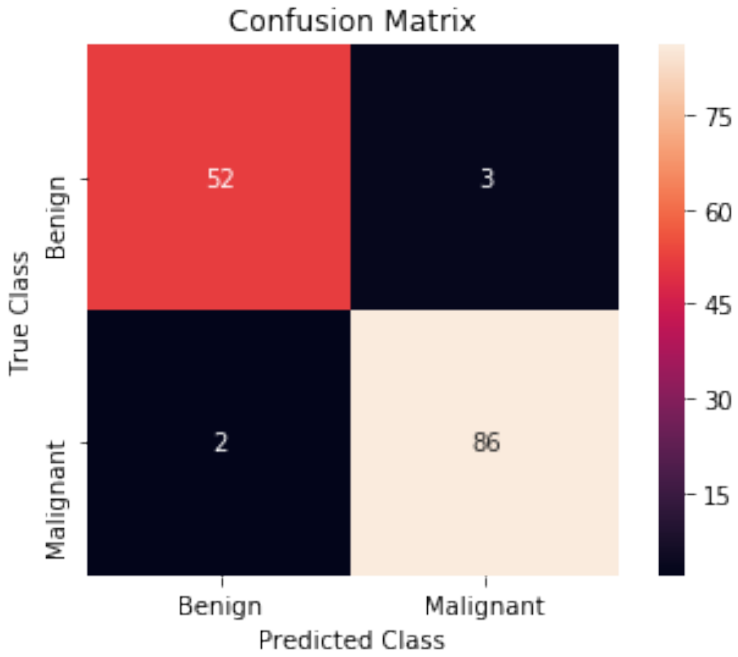

Figure 5.33: Random Forest Evaluation Score

The accuracy score is above the 95%. So, no need to perform the algorithm tuning. The problem case for breast cancer this logistic regression is also usable.[fig:5.32]

5.7 Random Forest

By using random forest ensmeble classsifer and evaluating the model into spelled and mispllled category. [See fig:5.33, 5.34]

The final accuracy of decision tree is

5.8 Decision Tree

By using decison tree classsifer and evaluating the model into spelled and mispllled category.[see fig:35]

```
print('Accuracy is',accuracy_score(y_pred,y_test))
```

```
Accuracy is 0.965034965034965
```

Figure 5.34: Random Forest Accuracy Score

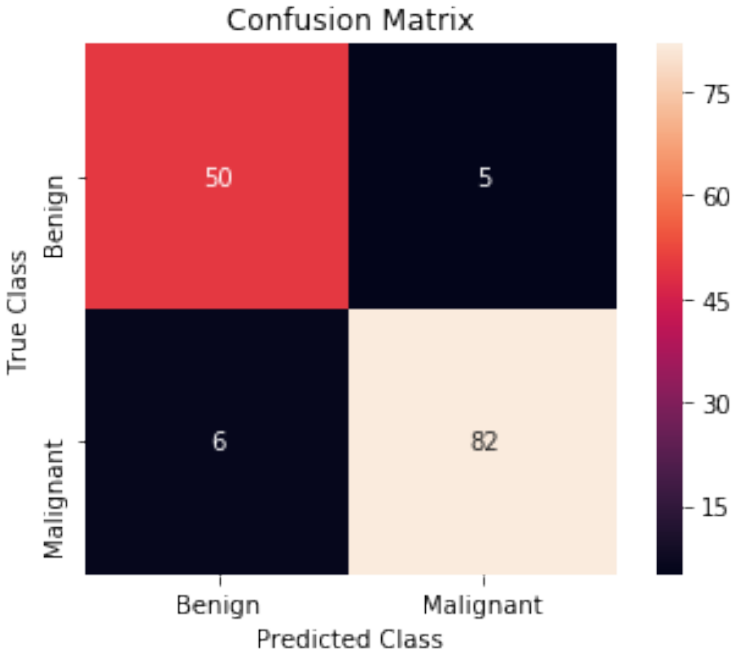

Figure 5.35: Decision Tree Evaluation Score

```
print('Accuracy is',accuracy_score(y_pred,y_test))
```

```
Accuracy is 0.9230769230769231
```

Figure 5.36: Decision Tree Accuracy Score

The final accuracy of decision tree is [see fig:5.36]

Chapter 6

Conclusion

ML techniques have been widely used in the medical field and have served as a useful diagnostic tool that helps physicians in analyzing the available data as well as designing medical expert systems. This paper presented three of the most popular ML techniques commonly used for breast cancer detection and diagnosis, namely Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Logistic Regression, Decision Tree, Random Forest. The main features and methodology of each of the these ML techniques were described. Performance comparison of the investigated techniques has been carried out using the Wisconsin Diagnostic Breast Cancer Data set.

Simulation results obtained has proved that classification performance varies based on the method that is selected. Results have shown that SVMs have the lowest performance in terms of accuracy, specificity, and precision. But we applied by improving the model for getting better accuracy level. However, Random Forests have the highest probability of correctly classifying tumor.

Appendices

Appendices are provided to give supplementary information, which is not included in the main text may serve as a separate part contributing to main theme.

Bibliography

[1] http://gco.iarc.fr/

[2] https://scikit-learn.org/stable/index.html

[3] https://www.udemy.com/machinelearning/

[4] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[5] Shajib Ghosh; Jubaer Hossain; Dr. Shaikh Anowarul Fattah; Dr. Celia Shahnaz; Asir Intisar Khan. *Efficient Approaches for Accuracy Improvement of Breast Cancer Classification Using Wisconsin Database˙*. 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) 21 - 23 Dec 2017, Dhaka, Bangladesh. 978-1-5386-2175-2/17/$31.00 ©2017 IEEE.

[6] Dana Bazazeh and Raed Shubair. *Comparative Study of Machine Learning Algorithms for Breast Cancer Detection and Diagnosis.*. 978-1-5090-5306-3/16/$31.00 c2016 IEEE.

[7] Meriem AMRANE; Saliha OUKID; Ikram GAGAOUA; Tolga ENSAR. *Breast Cancer Classification Using Machine Learning.* DOI: 78-1-5386-5135-3/18/$31.00 ©2018 IEEE.

[8] Mehmet Fatih Akay. *Support vector machines combined with feature selection for breast cancer diagnosis.* doi:10.1016/j.eswa.2008.01.009.

[9] Hiba Asri ; Hajar Mousannif; Hassan Al Moatassime; Thomas Noel. *Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis*. The 6th International Symposium on Frontiers in Ambient and Mobile Systems(FAMS 2016). Procedia Computer Science 83 (2016) 1064 – 1069.

[10] Seyyid Ahmed Medjahed; Tamazouzt Ait Saadi; Abdelkader Benyettou. *Breast Cancer Diagnosis by using k-Nearest Neighbor with Different Distances and Classification Rules.* International Journal of Computer Applications (0975 - 8887) Volume 62 - No. 1, January 2013.

[11] AhmetMert; Niyazi KJlJ; Erdem Bilgili; and Aydin Akan. *Breast Cancer Detection with Reduced Feature Set.* Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine Volume 2015, Article ID 265138, 11 pages. *http://dx.doi.org/10.1155/2015/265138.*.