**1. INTRODUCTION**

**1.1 Background**

Breast cancer is a type of cancer which occurs mostly among women throughout worldwide, the majority of breast cancer cases representation and cancer-related deaths increasing drastically which makes it the most significant health issue in women.

The ratheripe diagnosis of Breast Cancer can improve the pessimistic prediction and chance of survival significantly, as it can increase timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. Thus, the correct diagnosis of Breast Cancer and classification of patients into malignant or benign groups is the subject which needs more research. Machine learning is widely recognized as the methodology of choice in Breast Cancer pattern classification and classification modeling.

Data mining and classification methods both are effective ways for the classification problems. In a medical field, there always binary classification (either yes or no) which helps to analyze the problem in better way.

When healthy cells of female breast change and start growing excessively from the original growth of cells which is the evidence of the formation of breast cancer, tumour is a form a excessive layer of cells or large mass . A tumour could be benign or malignant. A benign tumour is not so dangerous because the tumour can grow but will not spread. Malignant tumour is cancerous , meaning it grows and spread to other body organs which are harmful. Breast cancer spreads when the cancer starts to grow in other parts of the body or when breast cancer cells comes in contact with other part of body through the lymph or blood vessels. This is known as metastasis. Our study covers locally advanced breast cancer and early-stages of cancer, which includes stages I, II, and III. The stage of breast cancer helps us to describes in which part of body the cancer is located, how much the cancer has grown, and if or where it has spread.

Although mostly the breast cancer grows and spread to neighbor of lymph nodes, cancer can also starts to spreads further through the other parts of body such as the nearby portions of breast. This is the stage IV breast cancer.

If breast cancer tumor develop after the prior treatment, it can be cure again , meaning in the breast or territorial lymph nodes. The territorial lymph nodes are those close to the breast, such as the lymph nodes below the arm. It can also reoccur anywhere in any body organ , called a distant reduplication or metastatic recurrence.

Cancer or tumor is a group of disease that involve abnormal cell growth with the potential to spread to other parts of the body but not all tumors are cancerous. There are 100 types of cancer, including breast cancer, skin cancer, lung cancer, colon cancer and lymphoma. Breast cancer is the one of the popular and second leading cause of cancer death in women. The chance that a woman will die from breast cancer is about 1 in 37 which is 2.7%.

Mostly the patient did not notice that they have breast cancer in the early stage because breast cancer starts when cells in the breast begin to grow out of control. This cell that from tumor usually can be seen on an x-ray or felt as a lump. Breast cancer occurs almost in women, but men also can get breast cancer.

Each year, an estimated 1.6 million new cases are diagnosed worldwide and in 2015, 560 thousand women die because of breast cancer (World Health Organization and National Cancer Registry of Malaysia 2005-2007). However, death rates from breath cancer was dropped from 1989 to 2007. Since 2007, breast cancer death rates have been steady in women younger than 50, but have 2 continued to decreased in older women (American Cancer Society’s Cancer Statistic Centre,2016). The decreases of the death rates in older women is believed that they noticed and get result of finding breast cancer earlier through screening and increased awareness which is get the treatment as soon as possible.

Breast cancer is hard to diagnose but when finding the breast cancer as early as possible, it will gives a better chance of successful treatment. To found out the breast cancer, it need to do the screening test. Screening test can help find breast cancer in its early stages, even before any symptoms appear. There are some common symptom of breast cancer. The most common symptom is a new lump or mass. A painless, hard mass that has irregular edges is more likely to be cancer, but breast cancers can be tender, soft, or rounded. So that, if there have any new breast mass or lump or breast change, it is important to checked by a health care provider experienced in diagnosing breast disease.

So that, it is important to all people especially women to be aware of changes in the breasts and to know the signs and symptoms of breast cancer. In this project, we propose a rule-based algorithm that functions as a reliable decision support system for breast cancer prediction.

**Objective :**

The main objectives of our project is to develop a flexible diagnostic model which can assist in :

* a better understanding of breast cancer survivability in the absence of data,
* providing better insights into factors associated with patient survivability, and
* Establishing cohorts of patients that share similar properties.

**1.2 Purpose of the Project**

The “Breast Cancer Data Analysis” is a predicting model which uses machine learning techniques and a diagnostic model which focuses on the prediction of cancer .

The Breast Cancer Data analysis model works on the basis of symptoms of various attributes present in patient report. It can take the attributes like tumor length, height, perimeter and radius etc. we can implement the various algorithm in our data frame in which knn is the best classification algorithm among other algorithm because it gives good recall score in comparison to others. After classification algorithm we implement feature extraction technique(Principle Component Analysis) to remove redundancy (attributes having common values). After applying feature extraction technique we conclude that some attributes can also be applicable to predict cancer . But, by removing redundancy recall value decreases . So, we don’t remove redundancy from data frame. That’s why we use ensemble technique .

Main purpose of the model is to solve few problems which helps in daily life :

* To raise the awareness of breast cancer among women .
* To predict whether the patient have enough time for treatment .
* Prediction of the stage of breast cancer.
* Improve the accuracy of model whether it predicts the cancer is benign or malign using different techniques.

**1.3 Problem Statement :**

In today’s world health is the biggest issue as people are more effecting due to the food they eat and the environment in which they live.

Doctors are not able to judge the cancer before the time period as doctors are not aware of the symptoms the patient carry which leads them to the cancer. Doctors are able to predict cancer of mostly last stage because of less information availability. So, machine learning model helps the doctor to predict at the early stage of the cancer by taking all the necessary points into consideration. Machine Learning models are fast and flexible to predict the cancer in the patient. As model are not replacing the doctors but give the more flexibility to analyze the data given of the patient.

As predicting cancer is classification problem so we applied various classification models to predict. As all the models are not predicting well on the dataset where KNN comes out of picture and perform well on the dataset as KNN is the distance measure model so features of benign and malign cancer are less distance. Hence KNN performing best in all of other models. Using PCA which help to remove the redundancy given to the standalone model and it will improve the accuracy by removing the redundancy.

In the end use the k fold technique to work on the full data and provide the measure of Spread or variance of the data to the model and then it will perform best on the test data or unseen data.

**1.4 Limitation of Work**

This system will focuses on prediction of breast cancer based on the details that have be given. This system does not give the decision on what the stages of cancer ( I,II,III or IV), it just only to predict whether the person have breast cancer or not. The result may not be 100% accurate but the result from this system will help the user to quickly take action and alert for the breast cancer by seek consultant from doctor or diagnose the breast cancer using screening test.

**2. LITERATURE REVIEW**

**2.1 Introduction**

Basically, in this chapter, the study of previous research is done. The related journal and articles was analyzed to find out what are the weakness of the previous research that we can overcome. Research paper related to rule-based has proved to be a powerful tool for decision making system, such as expert systems and pattern classification systems. Rule-based has already been used in some medical expert systems. The related system of cancer prediction is also been reviewed to help in understanding and gain knowledge about how to implement the system in the real applications. Here, some of the previous paper that has some weakness that can be solved through this project.

**2.2 Research towards Existing System**

Breast cancer is one of cancer killer in the world and can also kill men as well, not only women. Early detection of cancer is essential for a fast response and better chances of cure. But it is difficult to detect when it is in beginning because some of the symptoms of the disease are absent at the beginning. Machine learning methods and clinical factors was use to develop tools of cancer management. Rule based is one of the machine learning methods. It has proved to be a powerful tool of decision making systems. Rule based set theory has already been used in some medical expert system.

In traditional rule-based approach, knowledge is encoded in the form of antecedent consequent structure. When new data is encountered, it is matched to the antecedents clauses of each rule, and those rules where antecedent match a data exactly are fired, establishing the consequent clauses. This process continues until desire conclusion is reached. In the past decade, fuzzy logic has proved to be wonderful tool for intelligent systems in medicine. Some examples of using fuzzy logic to develop fuzzy intelligence systems are fuzzy systems in their micro processors, fuzzy cameras and camcorders that map image data to lens settings, and fuzzy voice commands “up”, “land”, “hover” to control unmanned helicopters. (Bart Kosko, Fuzzy Engineering, Prentice Hakk, 1997).

Konstantina Kourou, Themis P. Exarchos (2015) write a research paper “Machine learning applications in cancer prognosis and prediction” variety of these techniques, including Support Vector Machines (SVMs) and Decision Trees (DTs) have been widely applied in cancer research for the development of predictive models, resulting in effective and accurate decision making. Even though it is evident that the use of ML methods can improve our understanding of cancer progression, an appropriate level of validation is needed in order for these methods to be considered in the everyday clinical practice.

Hiba Asri, Hajar Mousannif (2016) “Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis” In his research paper, a compliance comparison between different machine learning algorithms: Support Vector Machine (SVM), Decision Tree (C4.5), Naive Bayes (NB) and k Nearest Neighbors (k-NN) on the Wisconsin Breast Cancer (original) datasets is conducted.

The main objective is to assess the correctness in classifying data with respect to efficiency and effectiveness of each algorithm in terms of accuracy, precision, sensitivity and specificity. Experimental results show that SVM gives the highest accuracy

(97.13%) with lowest error rate.

Ahmed Abd El-Hafeez Ibrahim published a paper “Performance analysis of various Open Source Tools on four Breast Cancer Dataset using Ensemble Classifiers Technique”.Analysis the performance of different classification algorithms shows that using Ensemble Classifiers Techniques improved the accuracy .

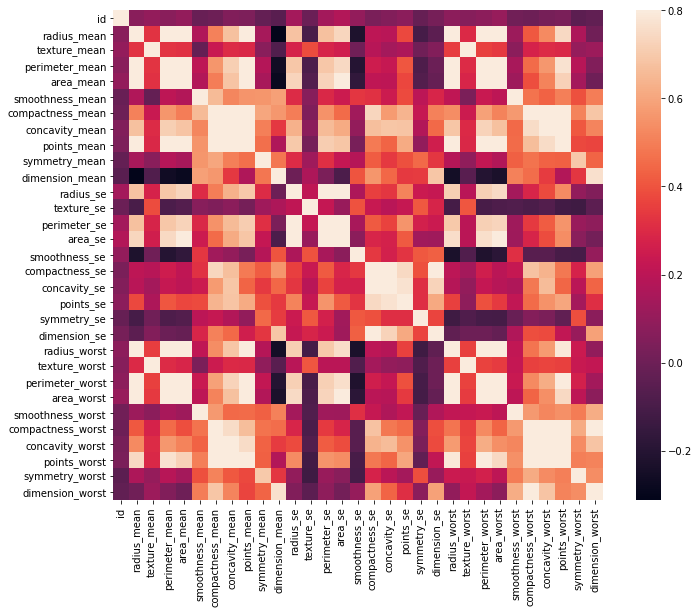
**2.3 Research Related with Others Method**

Breast Cancer Diagnosis by using k-Nearest Neigbor with Different Distance and Classification Rules by Seyyid Ahmad Medjahed, Tamazouzt Ait Saadi, Abdelkader Benyettou, 2013. This paper is to analyze the distance by using different values of the parameters “k” and by using several rules of classificarion and to evaluate the performance that can be used in the K-NN algorithms. In this paper, they study and analyze several distance and different values of the nearest neighbors parameter k, by using different classification rules in the k-nearest neighbor algorithm. K-nearest neigbors algorithms is one of the most used algorithms in machine learning and its a learning method based on instances that does required a learning phase. Before classifiying anew element, they compare it to other elements using a similarity measures. Its k-nearest neigbors are then considered, the class that appears most among the neighbors is assigned to the element to be classified. The neighbors are weighted by the distance that separate it to the new elements to classify.

As different models use different approaches for the prediction of breast cancer . Each approach give its own accuracy rate for the prediction of breast cancer. Mostly models use SVM (Support Vector Machine) for the prediction as a classification algorithm to find accuracy . In our model we use a different approach to predict the cancer .Using KNN algorithm on the data collected of breast cancer patient reports we apply ensemble technique . As a result our model can predict the cancer with a accuracy rate of 98 % which is more than any of the approach used be another researchers.

**3. METHODOLOGY AND REQUIRMENTS**

This section gives insights the dependency of target variables on independent variables using machine learning techniques to predict the type of cancer .The dependent variable is “diagnosis” whereas independent variables i.e. area\_mean, perimeter\_mean, radius\_mean, texture\_mean etc. are assumed to be predictors . For the analysis on these variables are done in two different ways.



**Fig 3.1 Heat Map of Data Attributes**

Firstly, classification algorithms are used to check the accuracy score of every classification algorithm and KNN is used to evaluate the value of target variable that is diagnosis 

**Fig 3.2 Value Counts of Malign and benign data**

While evaluating the data model various types of errors are occurred like over fitting, introduced from having too large of a training set and bias occur due to too small of a test set. So we applied PCA on the data to remove the redundancy of the model and to decrease the complexity of the model by removing the extra features from the model and inputting the spread of the data at each dimension.

While evaluating the data is divided into two parts, first is the training set and second is the test set. e

Firstly, k-fold cross validation or k-fold CV randomly divides the set of data into the number of K splits as value of K will be decided on the basis of the shape of the data , then using k-1 blocks datasets are trained and to test the performance of different algorithm, the remaining blocks are. Repeat the whole process k times and average value is calculated.

Second method is percentage split method in which we use train test split to split the data into percentage. Mainly 70% of the dataset is used for training and 30% of the dataset is used for testing.

We can evaluate the effectiveness of model by classification matrix, used to calculate the accuracy,precision,recall,f1-score.

Recall = TP/TP+FN

Precision=TP/TP+FP

Accuracy=TN+TP/TN+TP+FN+FP

F1-score=2\*precision\*recall/ precision+ recall

**\**

|  |  |  |
| --- | --- | --- |
| **n= 569** | **Predicted :**  **B** | **Predicted :**  **M** |
| **Actual : B** | **107** | **2** |
| **Actual : M** | **7** | **55** |

**Table 3.1 Confusion Matrix**

**3.1 Methodology Phases**

**3.1.1 Planning Phase**

In this phase, planning phase discussed a reason for selected goals include the detail overview of the goals. In this phase, first objective of the project are describe to identify a risk for fit people who can possibly get cancer and to make early detection. The title of this project has selected, Breast Cancer Prediction System. The abstract was done with all information gathered. Then, the entire requirement that involved in the system will be identified.

**3.1.2 Risk Analysis Phase**

In risk analysis phase, the requirements are studied and brain storming sessions were done to identify the potential risks. The risk that may exist is when it is difficult to differentiate the symptoms or it may be risk when the information of the symptoms is false. Once the risk was identified, risk mitigation strategy is planned and finalized.

**3.1.3 Engineering Phase**

This phase involve the actual development and testing. Breast Cancer Prediction System will be developed and will be tested. It will combine all the modules to become a complete system and will do integrating testing to make sure this system will function nicely. The development involves code, test cases, results, test summary, and report. 18

**3.1.4 Evaluation Phase**

In evaluation phase, admin will use and evaluate of the system. Then, user will provide their feedback and approval for the system. The features implemented document will be an output from this phase.

**4. ALGORITHMS**

This section gives insights the dependency of target variables on independent variables using machine learning techniques to predict the type of cancer .The dependent variable is “

diagnosis” whereas independent variables i.e. area\_mean, perimeter\_mean, radius\_mean, texture\_mean etc. are assumed to be predictors . For the analysis on these variables are done in two different ways.

Firstly, classification algorithms are used to check the accuracy score of every classification algorithm and KNN is used to evaluate the value of target variable that is diagnosis

While evaluating the data model various types of errors are occurred like over fitting, introduced from having too large of a training set and bias occur due to too small of a test set.

Analysis of cancer can be done in two different ways:

1. By doing the laboratory test
2. Test done by the doctors which is time consuming and predict after the particular stage has been crossed and patients are more prone to death.

**4.1 KNN**

KNN is not a parametric model , this means that it do not make any assumptions on any dataset while other algorithms make an assumption on the dataset.

We can also say that,when there is no knowledge about the dataset is given KNN is the first choice for the classification models because it measure the distance between the data point.

* KNN is the slowest algorithm in the machine learning since it do not use any training data points to do any modeling.
* Lack of modeling means that KNN keeps all the dataset of training. To be more exact, all (or most) the training data is needed during the testing phase
* KNN does not try to learn a function from the training data, so it is called Nonparametric model, It memorize the pattern from the dataset.
* In KNN it does not require any training on the dataset it simply calculates the distance matrix of the data points present in the data and using the data points whenever the new data points comes it makes the prediction using the distance matrix.

The working behind the classification KNN algorithm is one of easiest of all the supervised. algorithms of machine learning :

1. Its working is very simple as it first calculates distance of a data points which is new to the all other data points of training. The distance can be of any type e.g., Manhattan or Euclidean etc.

2. It takes the K-nearest data points, where K is preferred as taken as odd, if k taken as even then algorithm false prediction

3. Finally the data points is added to the cluster where the distance from the cluster is minimum.

4. This algorithm segregates unlabeled data points into well-defined groups.

KNN uses two distance matrix named as euclidian and manhattan. We use Euclidian distance instead of manhattan distance because it gives better result then manhattan distance

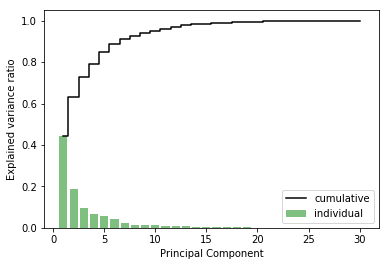
KNN uses two voting methods one is weighed and unweighed.

**4.2 PCA**

In model we pass the center value to the data .Model does not know covariance present in the data and PCA is the way to tell the covariance or the spread of the data.PCA is done on the independent variable then only it will tab the information otherwise model is not able to tab the information related to the covariance of the data.

X1,X2 are the independent be passed in the model .Instead of passing X1 and X2 we passed the relation between X1 and X2 to Y. So multicolinearity is taken care by PCA process which helps to reduce redundancy effect in the model. Redundancy is the process present in the data in the form of noise in the dataset.

As noise decreases information tends to increase.



**Fig4.2 Cumulative graphs of Eigen values**

Methods to imply PCA on dataset.

1. Firstly,we standardize the data (using z-score)
2. Secondly, convert the data into covariance matrix.
3. Using linear algebra we convert the covariance matrix into eigen vectors and eigen values.
4. Then, we sort the eigen vectors according to eigen values.
5. Then, creating the cumulative plot of the eigen values.
6. Then, we reduce the number of features and apply dot product on standardize data.

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.

* **Dimensionality:** It is the number of random variables in a dataset or simply the number of features, or rather more simply, the number of columns present in your dataset.
* **Correlation:** It shows how strongly two variable are related to each other. The value of the same ranges for -1 to +1. Positive indicates that when one variable increases, the other increases as well, while negative indicates the other decreases on increasing the former. And the modulus value of indicates the strength of relation.
* **Orthogonal:** Uncorrelated to each other, i.e., correlation between any pair of variables is 0.
* **Eigenvectors:** Eigenvectors and Eigenvalues are in itself a big domain, let’s restrict ourselves to the knowledge of the same which we would require here. So, consider a non-zero vector **v**. It is an eigenvector of a square matrix **A**, if **Av** is a scalar multiple of **v**. Or simply:

**Av = ƛv**

Here, **v** is the eigenvector and **ƛ** is the eigenvalue associated with it.

* **Covariance Matrix:** This matrix consists of the covariances between the pairs of variables. The (i,j)th element is the covariance between i-th and j-th variable.

Technically, a principal component can be defined as a linear combination of optimally-weighted observed variables. The output of PCA are these principal components, the number of which is less than or equal to the number of original variables. Less, in case when we wish to discard or reduce the dimensions in our dataset.

The PCs possess some useful properties which are listed below:

1. The PCs are essentially the linear combinations of the original variables, the weights vector in this combination is actually the eigenvector found which in turn satisfies the principle of least squares.
2. The PCs are orthogonal, as already discussed.
3. The variation present in the PCs decrease as we move from the 1st PC to the last one, hence the importance.

The least important PCs are also sometimes useful in regression, outlier detection, etc.

### ****Implementing PCA on a 2-D Dataset****

**Step 1: Normalize the data**

First step is to normalize the data that we have so that PCA works properly. This is done by subtracting the respective means from the numbers in the respective column. So if we have two dimensions X and Y, all X become 𝔁- and all Y become 𝒚-. This produces a dataset whose mean is zero.

**Step 2: Calculate the covariance matrix**

Since the dataset we took is 2-dimensional, this will result in a 2x2 Covariance matrix.

https://s3.amazonaws.com/files.dezyre.com/images/Tutorials/Covariance+Matrix.JPG

Please note that Var[X1] = Cov[X1,X1] and Var[X2] = Cov[X2,X2].

**Step 3: Calculate the eigenvalues and eigenvectors**

Next step is to calculate the eigenvalues and eigenvectors for the covariance matrix. The same is possible because it is a square matrix. **ƛ** is an eigenvalue for a matrix **A** if it is a solution of the characteristic equation:

**det( ƛI - A ) = 0**

Where, **I** is the identity matrix of the same dimension as **A** which is a required condition for the matrix subtraction as well in this case and ‘**det’**is the determinant of the matrix. For each eigenvalue **ƛ**, a corresponding eigen-vector **v**, can be found by solving:

**( ƛI - A )v = 0**

**Step 4: Choosing components and forming a feature vector:**

We order the eigenvalues from largest to smallest so that it gives us the components in order or significance. Here comes the dimensionality reduction part. If we have a dataset with n variables, then we have the corresponding neigenvalues and eigenvectors. It turns out that the eigenvector corresponding to the highest eigenvalue is the principal component of the dataset and it is our call as to how many eigenvalues we choose to proceed our analysis with. To reduce the dimensions, we choose the first p eigenvalues and ignore the rest. We do lose out some information in the process, but if the eigenvalues are small, we do not lose much.

Next we form a feature vector which is a matrix of vectors, in our case, the eigenvectors. In fact, only those eigenvectors which we want to proceed with. Since we just have 2 dimensions in the running example, we can either choose the one corresponding to the greater eigenvalue or simply take both.

Feature Vector = (eig1, eig2)

**Step 5: Forming Principal Components:**

This is the final step where we actually form the principal components using all the math we did till here. For the same, we take the transpose of the feature vector and left-multiply it with the transpose of scaled version of original dataset.

**NewData = FeatureVectorT x ScaledDataT**

New Data is the Matrix consisting of the principal components,

Feature Vector is the matrix we formed using the eigenvectors we chose to keep, and Scaled Data is the scaled version of original dataset

(‘T’ in the superscript denotes transpose of a matrix which is formed by interchanging the rows to columns and vice versa. In particular, a 2x3 matrix has a transpose of size 3x2)

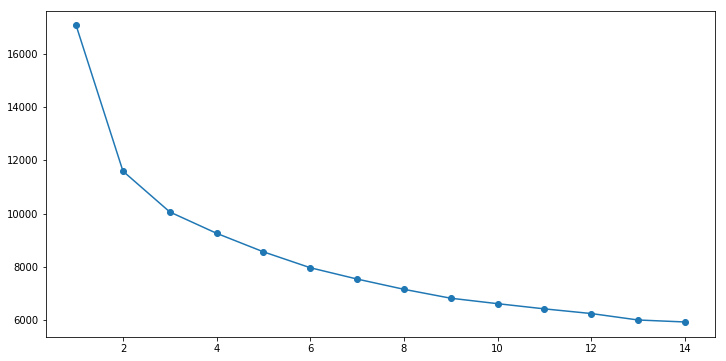
If we go back to the theory of eigenvalues and eigenvectors, we see that, essentially, eigenvectors provide us with information about the patterns in the data. In particular, in the running example of 2-D set, if we plot the eigenvectors on the scatterplot of data, we find that the principal eigenvector (corresponding to the largest eigenvalue) actually fits well with the data. The other one, being perpendicular to it, does not carry much information and hence, we are at not much loss when deprecating it, hence reducing the dimension.

All the eigenvectors of a matrix are perpendicular to each other. So, in PCA, what we do is represent or transform the original dataset using these orthogonal (perpendicular) eigenvectors instead of representing on normal **x** and **y** axes. We have now classified our data points as a combination of contributions from both **x**and **y.**The difference lies when we actually disregard one or many eigenvectors, hence, reducing the dimension of the dataset. Otherwise, in case, we take all the eigenvectors in account, we are just transforming the co-ordinates and hence, not serving the purpose.

**4.3 K Means**

K means is an unsupervised technique which helps to group the data on the basis distance apart from one another.

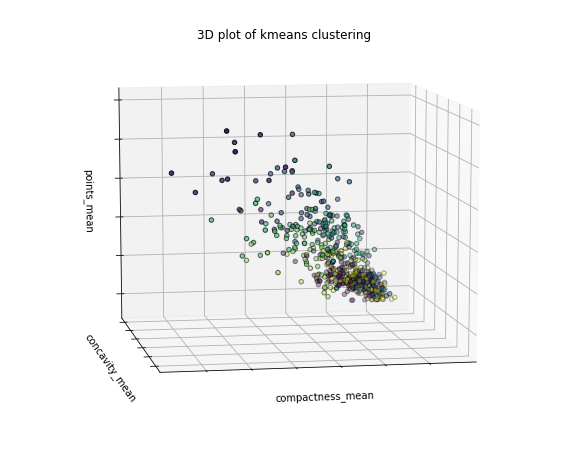
K means cluster our data as much as possible to the equal density(equal variance group)



**Fig 4.3.1 Selecting K value Plot**

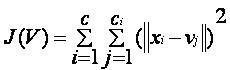
The working of algorithm :

1. First we input the k value according to the elbow plot.
2. We take mean of every data point and the mean which is closest to the cluster mean then the data point will attach to that cluster .
3. We repeat the process for a given k value we have our clusters.

****

**Fig 4.3.2 3D plot of K means Clustering**

k-means is  one of  the simplest unsupervised  learning  algorithms  that  solve  the well  known clustering problem. The procedure follows a simple and  easy  way  to classify a given data set  through a certain number of  clusters (assume k clusters) fixed apriori. The  main  idea  is to define k centers, one for each cluster. These centers  should  be placed in a cunning  way  because of  different  location  causes different  result. So, the better  choice  is  to place them  as  much as possible  far away from each other. The  next  step is to take each point belonging  to a  given data set and associate it to the nearest center. When no point  is  pending,  the first step is completed and an early group age  is done. At this point we need to re-calculate k new centroids as barycenter of  the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done  between  the same data set points  and  the nearest new center. A loop has been generated. As a result of  this loop we  may  notice that the k centers change their location step by step until no more changes  are done or  in  other words centers do not move any more. Finally, this  algorithm  aims at  minimizing  an objective function know as squared error function given by:

[](https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm/kmeans.JPG?attredirects=0)

where,  
                           *‘||xi- vj||’* is the Euclidean distance between *xi* and *vj.*

*‘ci’* is the number of data points in *ith* cluster.

*‘c’* is the number of cluster centers.

**Algorithmic steps for k-means clustering**

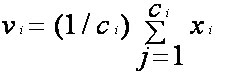
Let  X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.

1) Randomly select *‘c’* cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:



where,*‘ci’* represents the number of data points in *ith* cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

**Advantages**

1) Fast, robust and easier to understand.

2) Relatively efficient: O(tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t  is # iterations. Normally, k, t, d << n.

3) Gives best result when data set are distinct or well separated from each other.

**Disadvantages**

1) The learning algorithm requires apriori specification of the number of  cluster centers.

2) The use of Exclusive Assignment - If  there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.

3) The learning algorithm is not invariant to non-linear transformationsi.e.with different representation of data we get different results (data represented in form of cartesian co-ordinates and polar co-ordinates will give different results).

4) Euclidean distance measures can unequally weight underlying factors.

5) The learning algorithm provides the local optima of the squared error function.

6) Randomly choosing of the cluster center cannot lead us to the fruitful result. Pl. refer [Fig](https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm/k-means_initial_cluster_selection).

7) Applicable only when mean is defined i.e. fails for categorical data.

8) Unable to handle noisy data and outliers*.*

9) Algorithm fails for non-linear data set.

**4.4 Ensemble techniques**

By combining several models we can improve the machine learning results and this technique is known as ensemble technique .This approach gives better predictive performance compared to single model.

Ensemble techniques are used to decrease the variance as well as bias error and ensemble learning helps to stack more then one machine learning model.

*Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. That is why ensemble methods placed first in many prestigious machine learning competitions, The*[*Statsbot*](http://statsbot.co/?utm_source=blog&utm_medium=article&utm_campaign=ensemble)*team wanted to give you the advantage of this approach and asked a data scientist, Vadim Smolyakov, to dive into three basic ensemble learning techniques.*Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to **decrease** **variance**(bagging), **bias** (boosting), or **improve predictions** (stacking).

Ensemble methods can be divided into two groups:

* *sequential* ensemble methods where the base learners are generated sequentially (e.g. AdaBoost).  
  The basic motivation of sequential methods is to**exploit the dependence between the base learners.** The overall performance can be boosted by weighing previously mislabeled examples with higher weight.
* *parallel* ensemble methods where the base learners are generated in parallel (e.g. RandomForest).   
  The basic motivation of parallel methods is to **exploit independence between the base learners** since the error can be reduced dramatically by averaging.

Most ensemble methods use a single base learning algorithm to produce homogeneous base learners, i.e. learners of the same type, leading to *homogeneous ensembles*.There are also some methods that use heterogeneous learners, i.e. learners of different types, leading to *heterogeneous ensembles*. In order for ensemble methods to be more accurate than any of its individual members, the base learners have to be as accurate as possible and as diverse as possible.

There are three types of ensemble learning

1 Bagging

2 Boosting

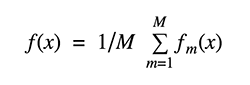
3 Stacking

**4.4.1 BAGGING :**

Bagging is a method in which resampling of data with replacement .It is a technique which works only on decreasing the variance error .It works on the principle of Bootstrap sampling.

Bootstrap sampling is a technique in which chances of same data point occurs in a bag comes fourth times if we take five bags .So there are so many duplicates or triplicate in every bag .Because records are picked with replacement .So there are certain records which are unpicked known as out of bag records(OOB) and typically we use OOB records as testing .By doing this we are providing generalization to the model.

Bagging stands for bootstrap aggregation. One way to reduce the variance of an estimate is to average together multiple estimates. For example, we can train M different trees on different subsets of the data (chosen randomly with replacement) and compute the ensemble:

\

Bagging uses bootstrap sampling to obtain the data subsets for training the base learners. For aggregating the outputs of base learners, bagging uses *voting for classification* and *averaging for regression*.

We can study bagging in the context of classification on the Iris dataset. We can choose two base estimators: a decision tree and a k-NN classifier. Figure 1 shows the learned decision boundary of the base estimators as well as their bagging ensembles applied to the Iris dataset.

Accuracy: 0.63 (+/- 0.02) [Decision Tree]  
Accuracy: 0.70 (+/- 0.02) [K-NN]  
Accuracy: 0.64 (+/- 0.01) [Bagging Tree]  
Accuracy: 0.59 (+/- 0.07) [Bagging K-NN]

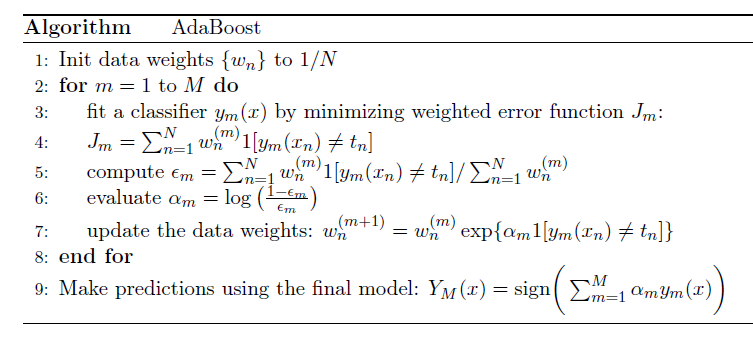
**4.4.2 BOOSTING :**

Boosting is same like bagging but in bagging all bags works parallely. But in boosting one bag has to complete the process first then next bag starts doing this process .The first bag has to complete this task which picks all the records with replacement .In first bag all records have equal probability but for second bag the probability is not equal as we changed the probability of some data points .The probability of certains data points gets changed because bag one misclassify the certain data points so it increases its probability due to which bag two will solve the data point with higher probability and last bag is the highest complex bag because highly misclassified data points which contain the high probability value makes the model bag more complex so it is a sequential approach.

Boosting refers to a family of algorithms that are able to convert weak learners to strong learners. The main principle of boosting is to fit a sequence of weak learners− models that are only slightly better than random guessing, such as small decision trees− to weighted versions of the data. More weight is given to examples that were misclassified by earlier rounds.

The predictions are then combined through a weighted majority vote (classification) or a weighted sum (regression) to produce the final prediction. The principal difference between boosting and the committee methods, such as bagging, is that base learners are trained in sequence on a weighted version of the data.

The algorithm below describes the most widely used form of boosting algorithm called **AdaBoost**,which stands for adaptive boosting.



We see that the first base classifier y1(x) is trained using weighting coefficients that are all equal. In subsequent boosting rounds, the weighting coefficients are increased for data points that are misclassified and decreased for data points that are correctly classified.

The quantity epsilon represents a weighted error rate of each of the base classifiers. Therefore, the weighting coefficients alpha give greater weight to the more accurate classifiers.

**4.4.3 STACKING :**

Stacking is the simplest approach in which there are two or three models which performs better on the dataset .will be group together and weighed voting is used to evaluate the model.

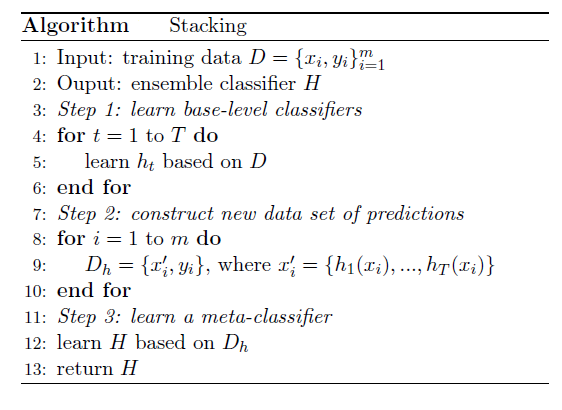
So, assume that we want to fit a stacking ensemble composed of L weak learners. Then we have to follow the steps thereafter:

* split the training data in two folds
* choose L weak learners and fit them to data of the first fold
* for each of the L weak learners, make predictions for observations in the second fold
* fit the meta-model on the second fold, using predictions made by the weak learners as inputs

In the previous steps, we split the dataset in two folds because predictions on data that have been used for the training of the weak learners are **not relevant for the training of the meta-model**. Thus, an obvious drawback of this split of our dataset in two parts is that we only have half of the data to train the base models and half of the data to train the meta-model. In order to overcome this limitation, we can however follow some kind of “k-fold cross-training” approach (similar to what is done in k-fold cross-validation) such that all the observations can be used to train the meta-model: for any observation, the prediction of the weak learners are done with instances of these weak learners trained on the k-1 folds that do not contain the considered observation. In other words, it consists in training on k-1 fold in order to make predictions on the remaining fold and that iteratively so that to obtain predictions for observations in any folds. Doing so, we can produce relevant predictions for each observation of our dataset and then train our meta-model on all these predictions.

Stacking is an ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The base level models are trained based on a complete training set, then the meta-model is trained on the outputs of the base level model as features.

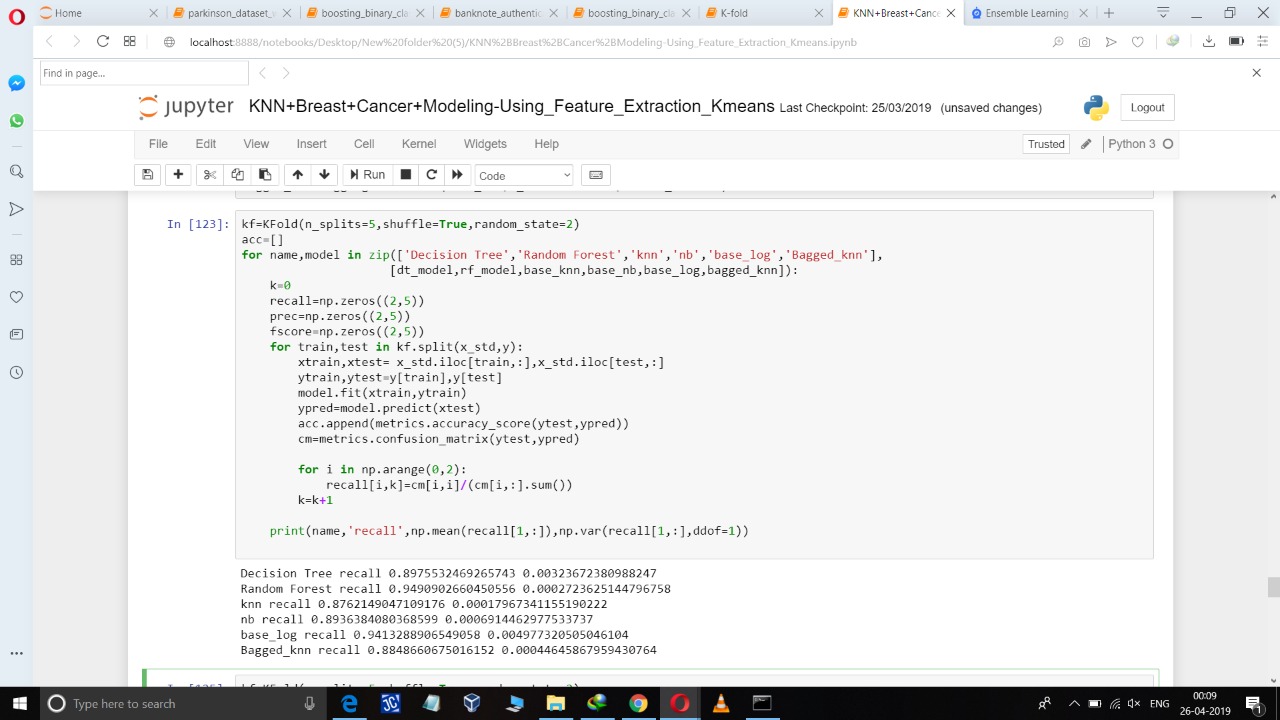
The base level often consists of different learning algorithms and therefore stacking ensembles are often heterogeneous. The algorithm below summarizes stacking.



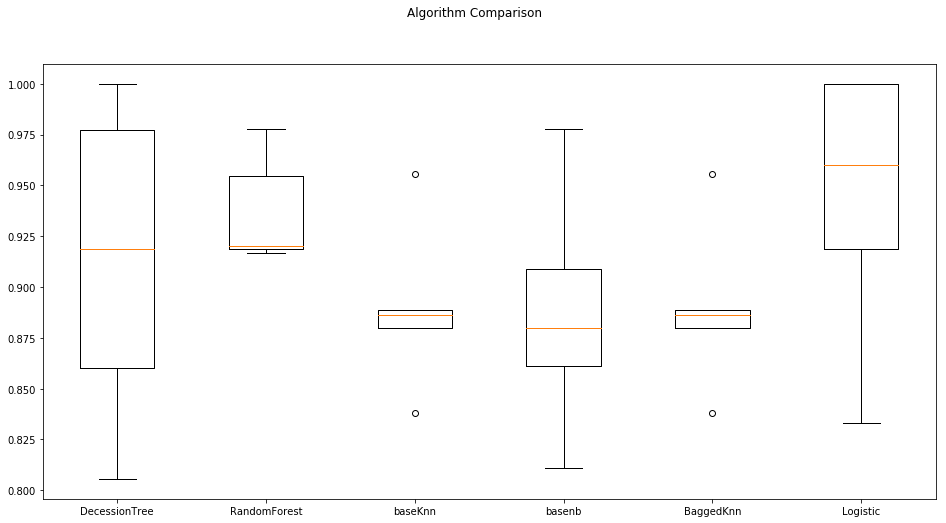
The following accuracy is visualized in the top right plot of the figure above:

Accuracy: 0.91 (+/- 0.01) [KNN]  
Accuracy: 0.91 (+/- 0.06) [Random Forest]  
Accuracy: 0.92 (+/- 0.03) [Naive Bayes]  
Accuracy: 0.95 (+/- 0.03) [Stacking Classifier]

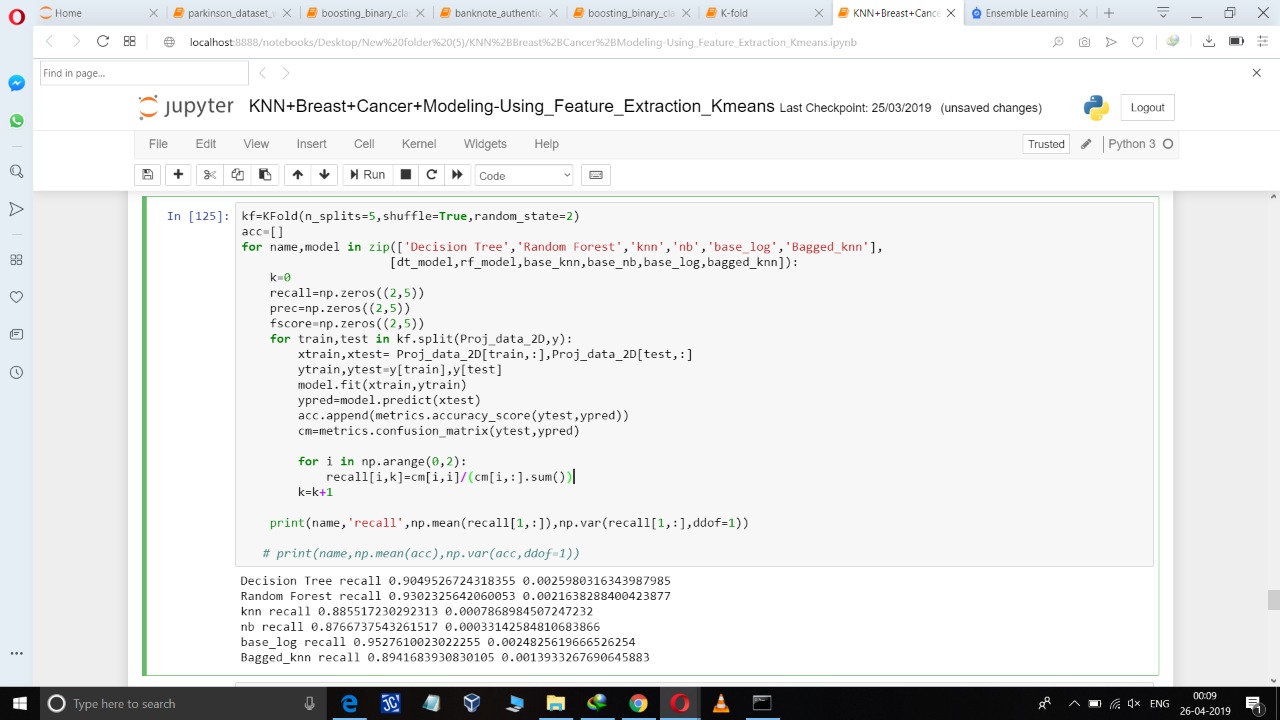
The stacking ensemble is illustrated in the figure above. It consists of k-NN, Random Forest, and Naive Bayes base classifiers whose predictions are combined by Logistic Regression as a meta-classifier. We can see the blending of decision boundaries achieved by the stacking classifier.



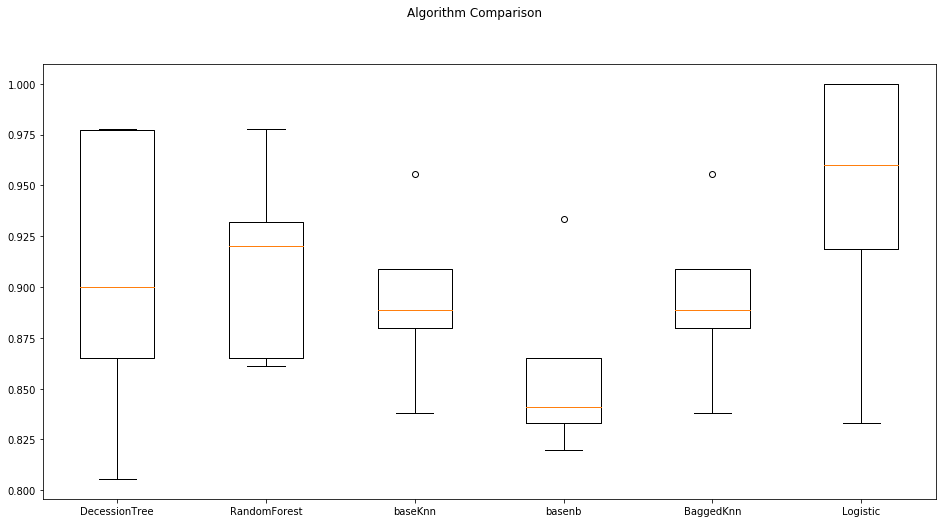
**Fig 4.4.1 : Bagged KNN model With variance and bias error using standardize data**

****

**Fig 4.4.2 : Graphical Representation Std Data**



**Fig 4.4.3 : Bagged KNN model With variance and bias error using PCA data**



**Fig 4.4.4 : Graphical Representation local Data**

**5. Hardware and Software Requirements**

**5.1 Hardware Requirements**

* Personal operating system /laptop
* Internet connection 4G LTE
* 20 GB minimum storage capacity
* Processor: I5 processor octacore
* Graphic 2.00 GHz
* RAM: 4.00 GB
* OS: Windows 8.1

**5.2 Software Requirements**

* Python 3.0
* Jupyter Notebook
* Anaconda
* Web Browser (Mozilla Firefox, Google Chrome)

**5.3 Library Used**

1. **import numpy as np**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

1. **import pandas as pd**

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the [Python](https://www.python.org/) programming language.

1. **from sklearn.neighbors import KNeighborsClassifier**

[sklearn.neighbors](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.neighbors)  provides functionality for unsupervised and supervised neighbors-based learning methods. Unsupervised nearest neighbors is the foundation of many other learning methods, notably manifold learning and spectral clustering. Supervised neighbors-based learning comes in two flavors: [classification](https://scikit-learn.org/stable/modules/neighbors.html#classification) for data with discrete labels, and [regression](https://scikit-learn.org/stable/modules/neighbors.html#regression) fordata with continuous labels.

1. **from scipy.stats import zscore**

**scipy.stats.zscore(arr, axis=0, ddof=0)**function computes the relative **Z-score** of the input data, relative to the sample mean and standard deviation.

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

Split arrays or matrices into random train and test subsets Quick utility that wraps input validation and next(ShuffleSplit().split(X, y)) and application to input data into a single call for splitting (and optionally subsampling) data in a oneliner.

1. **from sklearn import metrics**

The module [**sklearn.metrics**](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics) also exposes a set of simple functions measuring a prediction

error given ground truth and prediction:

* functions ending with \_score return a value to maximize, the higher the better.
* functions ending with \_error or \_loss return a value to minimize, the lower the better. When converting into a scorer object using [**make\_scorer**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.make_scorer.html#sklearn.metrics.make_scorer), set the greater\_is\_better parameter to False (True by default; see the parameter description below).

**7.import seaborn as sns**

Seaborn is a Python data visualization library based on [matplotlib](https://matplotlib.org/). It provides a high-level interface for drawing attractive and informative statistical graphics.

1. **import matplotlib.pyplot as plt**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) notebook, web application servers, and four graphical user interface toolkits.

**9. %matplotlib inline**

%matplotlib is a magic function in IPython. ... %matplotlib inline sets the backend of matplotlib to the 'inline' backend: With this backend, the output of plotting commands is displayed **inline** within frontends like the Jupyter notebook, directly below the code cell that

produced it.

**10. from sklearn.cluster import KMeans**

The plots display firstly what a K-means algorithm would yield using three clusters. It is then

shown what the effect of a bad initialization is on the classification process: By setting n\_init

to only 1 (default is 10), the amount of times that the algorithm will be run with different

centroid seeds is reduced. The next plot displays what using eight clusters would deliver and

finally the ground truth.

**11.from sklearn.ensemble import RandomForestClassifier**

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

**12.from sklearn.tree import DecisionTreeClassifier**

**Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

**13. from sklearn.preprocessing import StandardScaler**

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

Standardize features by removing the mean and scaling to unit variance

The standard score of a sample x is calculated as:

z = (x - u) / s

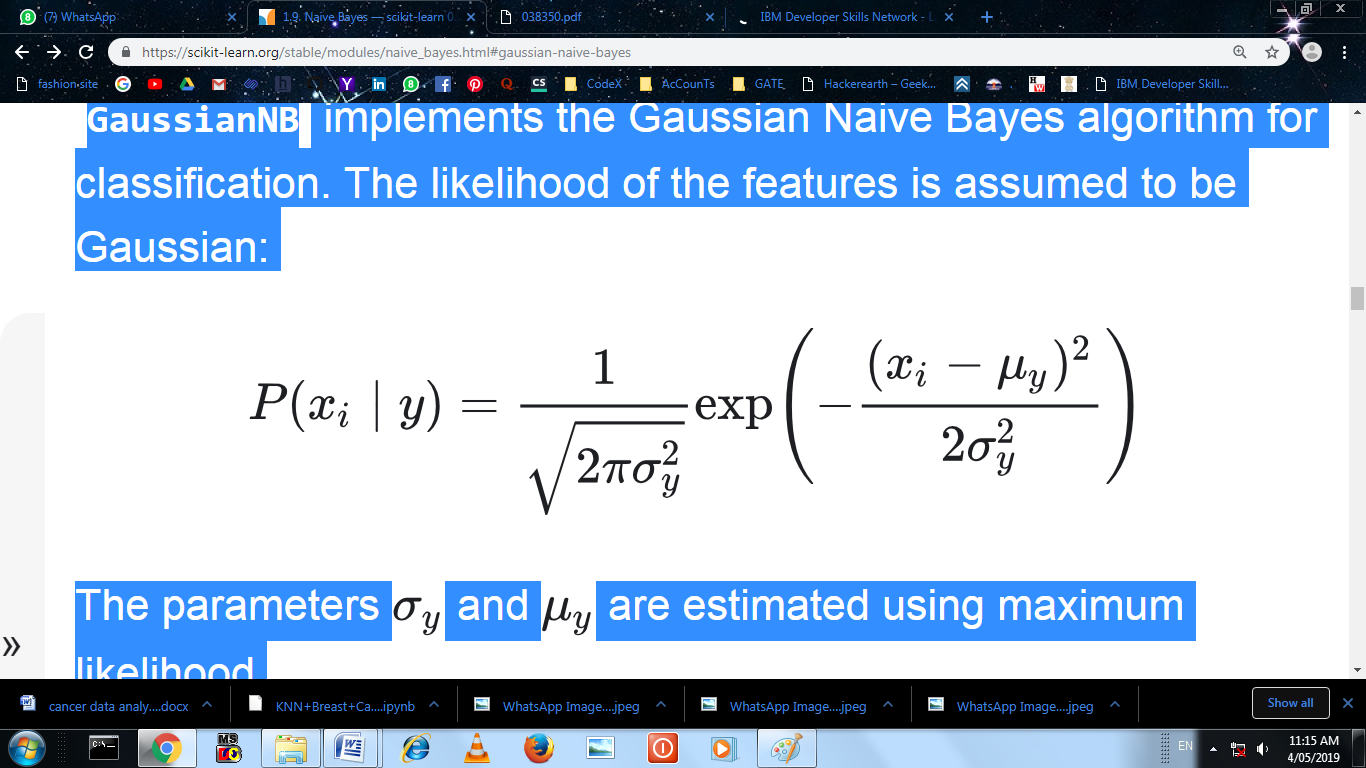
**14. from sklearn.model\_selection import cross\_val\_score,KFold**

The simplest way to use cross-validation is to call the [**cross\_val\_score**](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html#sklearn.model_selection.cross_val_score) helper function on the estimator and the dataset.

The following example demonstrates how to estimate the accuracy of a linear kernel support vector machine on the iris dataset by splitting the data, fitting a model and computing the score 5 consecutive times (with different splits each time).

**15. from sklearn.naive\_bayes import GaussianNB**

[GaussianNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB) implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:



The parameters **σy** and **μy** are estimated using maximum likelihood.

**16. from sklearn.linear\_model import LogisticRegression**

Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a [logistic function](https://en.wikipedia.org/wiki/Logistic_function).

The implementation of logistic regression in scikit-learn can be accessed from class [Logistic Regression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression). This implementation can fit binary, One-vs- Rest, or multinomial logistic regression with optional L2 or L1 regularization.

**17. from sklearn.metrics import roc\_curve,auc**

The function [**roc\_curve**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html#sklearn.metrics.roc_curve) computes the [receiver operating characteristic curve, or ROC curve](https://en.wikipedia.org/wiki/Receiver_operating_characteristic).

“A receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives (TPR = true positive rate) vs. the fraction of false positives out of the negatives (FPR = false positive rate), at various threshold settings. TPR is also known as sensitivity, and FPR is one minus the specificity or true negative rate.”

**18. from sklearn.ensemble import AdaBoostClassifier**

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights w1, w2, …, wN to each of the training samples. Initially, those weights are all set to wi=1/N, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence.

**19.from sklearn.ensemble import BaggingClassifier**

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

This algorithm encompasses several works from the literature. When random subsets of the dataset are drawn as random subsets of the samples, then this algorithm is known as Pasting . If samples are drawn with replacement, then the method is known as Bagging . When random subsets of the dataset are drawn as random subsets of the features, then the method is known as Random Subspaces . Finally, when base estimators are built on subsets of both samples and features, then the method is known as Random Patches

**# To enable plotting graphs in Jupyter notebook**

**6.FlOWCHART**

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes. If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields. Flowcharts are sometimes called by more specialized names such as Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN),  or Process Flow Diagram (PFD). They are related to other popular diagrams, such as Data Flow Diagrams (DFDs) and Unified Modeling Language (UML) Activity Diagrams.

## Flowchart symbols

Here are some of the common flowchart symbols. For a more comprehensive list, see our full [flowchart symbols page](https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained).

|  |  |
| --- | --- |
| Terminal/Terminator  **Process**  **Terminator** |  |
| Process |  |
| Decision  **Desicison**  Docoument |  |
| Document  **Data** |  |
| Data, or Input/Output |  |
| Stored Data |  |
| Flow Arrow |  |
|  |  |
| Predefined process |  |
| On-page connector/reference |  |
| Off-page connector/reference |  |

## Flowcharts for computer programming/algorithms :

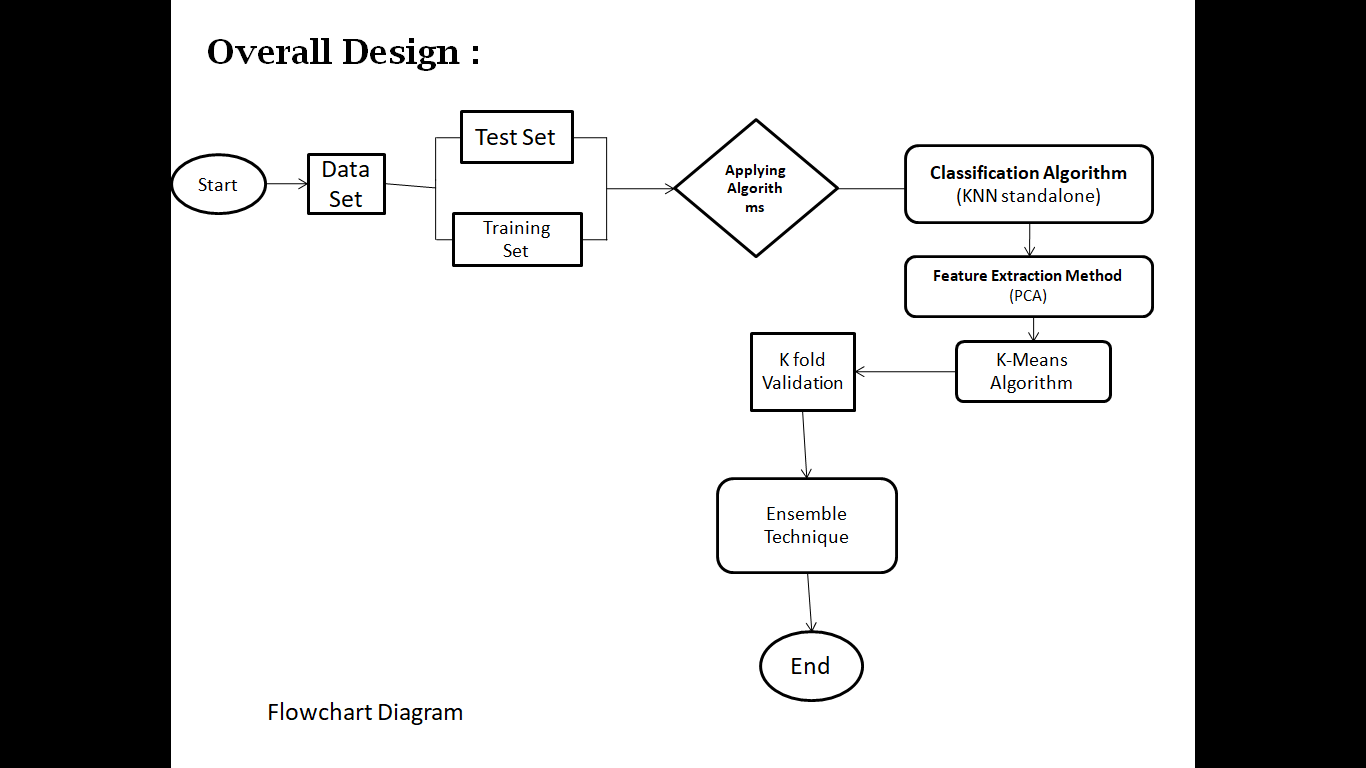
As a visual representation of data flow, flowcharts are useful in writing a program or algorithm and explaining it to others or collaborating with them on it. You can use a flowchart to spell out the logic behind a program before ever starting to code the automated process. It can help to organize big-picture thinking and provide a guide when it comes time to code. More specifically, flowcharts can:

* Demonstrate the way code is organized.
* Visualize the execution of code within a program.
* Show the structure of a website or application.
* Understand how users navigate a website or program.

Often, programmers may write pseudocode, a combination of natural language and computer language able to be read by people. This may allow greater detail than the flowchart and serve either as a replacement for the flowchart or as a next step to actual code.

Related diagrams used in computer software include:

* Unified Modeling Language (UML): This is a general-purpose language used in software engineering for modeling.
* Nassi-Shneiderman Diagrams: Used for structured computer programming. Named after Isaac Nassi and Ben Shneiderman, who developed it in 1972 at SUNY-Stony Brook. Also called Structograms.
* DRAKON charts: DRAKON is an algorithmic visual programming language used to produce flowcharts.

****

**Fig 6.1: Flowchart of Overall Design**

1. **CONCLUSION**

This paper deals with KNN algorithm to classify cancer tumors as either benign or malignant. We applied feature selection on the dataset to remove duplicate and irrelevant features. we applied symmetrical uncertainty attribute evaluation in WEKA for feature selection. Our proposed approach is evaluated and compared using Wisconsin breast cancer dataset. The experimental result showed that accuracy, precision, recall, and F-measure are increased by our proposed method when compared with different models. In future, we will work on feature selection techniques to improve the accuracy of the model.

1. **REFRENCES**

* Konstantina Kourou a, Themis P. Exarchos a,b, Konstantinos P. Exarchos a,

Michalis V. Karamouzis c, Dimitrios I. Fotiadis a,b, “Machine learning applications in cancer prognosis and prediction” ,Computational and Structural Biotechnology Journal 13 (2015) 8–17

* Hiba Asria\*,Hajar Mousannifb,Hassan Al Moatassimec,Thomas Noeld “Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis” Procedia Computer Science 83 ( 2016 ) 1064 – 1069
* Ahmed Abd El-Hafeez Ibrahim1, Atallah Ibrahin Hashad2 and Negm Eldin Mohamed Shawky3

Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt “Performance Analysis of Various Open Source Tools on Four Breast Cancer Datasets using Ensemble Classifiers Techniques”

* Coleman MP, Rachet B, Woods LM, Mitry E, Riga M, et al. Trends and socioeconomic inequalities in cancer survival in England and Wales up to 2001. British Journal of Cancer. 2004;90(7):1367–1373. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2409687/)] [[PubMed](https://www.ncbi.nlm.nih.gov/pubmed/15054456)]
* Sai Prasad Potharaju et al., “A Novel M-Cluster of Feature Selec-tion Approach Based on Symmetrical Uncertainty for Increasing Classification Accuracy of Medical Datasets”, Journal of Engineer-ing Science and Technology Review, volume 6, pp.154-162, 2017.
* [8] Bangsuk Jantawan et al., “A Comparison of Filter and Wrapper Approaches with Data Mining Techniques for Categorical Varia-bles Selection”, International Journal of Innovative Research in Computer and Communication Engineering, Volume 2, 2014.