



# Welcome

## Cancer Prediction

## **Abstract of the project:**

Abstract—

**Cancer is the second cause of death in the world. 8.8 million patients died due to cancer in 2015. Breast cancer is the leading cause of death among women. Several types of research have been done on early detection of breast cancer to start treatment and increase the chance of survival. Most of the studies concentrated on mammogram images. However, mammogram images sometimes have a risk of false detection that may endanger the patient's health. It is vital to find alternative methods which are easier to implement and work with different data sets, cheaper and safer, that can produce a more reliable prediction. This paper proposes a hybrid model combined of several Machine Learning (ML) algorithms including Support Vector Machine (SVM), Artificial Neural Network (ANN), K-Nearest Neighbor (KNN), Decision Tree (DT) for effective breast cancer detection. This study also discusses the datasets used for breast cancer detection and diagnosis. The proposed model can be used with different data types such as image, blood, etc. Index Terms—Breast Cancer; Breast Cancer Detection; Medical Images; Machine Learning.**

**INTRODUCTION :**

**World Health Organization (WHO) reported the breast cancer is the most common cancer amongst women globally. It is also the highest ranked type of cancer cause the death among women in the world. In Malaysia, Breast cancer has the highest rate of cancer deaths, around 25%, and it is the commonest cancer among women . Around 5% of Malaysian women are at risk of breast cancer while Europe and the**

United States, it is around 12.5% . It confirms that women with breast cancer in Malaysia present at a later stage of the disease compared to women from other countries . Usually, breast cancer can be easily detected if specific symptoms appear. However, many women who are suffering from breast cancer have no symptoms. Hence, regular breast cancer screening is very important for early detection. Early detection of breast cancer aids for early diagnosis and treatment, because the prognosis is very important for longterm survival. Since early detection, diagnosis, and treatment of cancer can reduce the risk of death, it plays a significant role in saving the life of the patient. Any delay in detection of cancer in early stages leads to disease progression and complication of treatment, therefore long waiting time prior to diagnosis of breast cancer and starting the treatment process is of prognostic concern. Previous studies on the investigation of the consequences of a late diagnosis of cancer confirm that it is strongly associated with progression of the disease to more advanced stages, consequently less chance to save the patient's life. In a systematic review conducted by Prof MA Richards et al. an analysis of 87 studies strongly concluded that female patients with breast cancer who start their therapy less than 3 months after the appearance of symptoms significantly have a higher chance of survival compare to those who wait for more than 3 months. Many previous studies confirm that detection of breast cancer in early stages significantly increase the chance of survival because it prevents the spreading of malignant cells throughout the entire body. The main contribution of this paper is to review the role of machine learning techniques in early detection of the breast cancer. Artificial Intelligence (AI) can be applied to improve breast cancer detection and diagnosis, as well as

prevent overtreatment. Nevertheless, combining AI and Machine Learning (ML) methods enables the prediction and empower accurate decision making. For example, deciding on the biopsy results for detecting breast cancer if the patient needs surgery or not. Currently, Mammograms are the most used test available, however, still, they have false positive (high-risk) results which shows abnormal cells that can lead to unnecessary biopsies and surgeries. Sometimes surgery is done to remove lesions reveals that it is benign which is not harmful. This means that the patient will go through unnecessary painful and expensive surgery. ML Algorithms were introduced with many features such as effective performance on healthcare related dataset which involve images, x-rays, blood samples, etc. Some methods are appropriate for the small dataset whereby others are suitable for huge datasets. However, noise can be a problematic concern in some methods. This paper is organized as follows, Section II introduces the breast cancer briefly, Section III explains the ML algorithms used for detecting breast cancer. A summary of previous related works is given in section IV. Finally, Section V concludes the paper.

## II. BREAST CANCER

Breast cancer is the most found disease in the women, worldwide, where abnormal growth of a mass of tissue, cause the expansion of malignant cells leads to acute breast cancer. These malignant cells are originally created from milk glands of the breast. These malignant cells which are the main reason for breast cancer can be classified into different groups according to their unusual progress and capability affecting other normal cells. The capability of affecting means whether these malignant cells affect only the local cells or can spread throughout the full body. The effect of spreading these

Journal of Telecommunication, Electronic and Computer Engineering 22 e-

ISSN: 2289-8131 Vol. 10 No. 3-2 malignant cells throughout the whole body of the patient is called as metastasis. It is very important to prevent this spreading effect by a diagnosis of cancer in the early stages using advanced techniques and equipment. In recent decades, there are many efforts to employ artificial intelligence and other related methods to assist in the detection of cancer in earlier stages. Early detection of cancer boosts the increase of survival chance to 98% .Different types of cancers whereby breast cancer is leading with 24% as follows. Figure 1: Types of cancer III. MACHINE LEARNING METHODS Machine Learning is a process that machines (computers) are trained with data to make the decision for similar cases. ML is employed in various applications, such as object recognition, network, security, and healthcare. There are two ML types i.e. single and hybrid methods like ANN, SVM, Gaussian Mixture Model (GMM), K-Nearest Neighbor (KNN), Linear Regressive Classification (LRC), Weighted Hierarchical Adaptive Voting Ensemble (WHAVE), etc. Following are the used ML algorithms: A. Artificial Neural Network (ANN) ANN is a model like human brains nerve system that has a large number of nodes connected to each other. Each node has two states: 0 means inactive and 1 means active. Also, each node has a positive or negative weight that adjusts the strength of the node and can activate or deactivate it. ANN provides samples of data to train the machine. The trained machine is used to detect the pattern of hidden data. It can search for patterns among patients' healthcare and personal records to identify high-risk lesions. B. Support Vector Machine (SVM) SVM is a supervised pattern classification model which is used as a training algorithm for learning classification and regression rule from gathered data. The purpose of this method is to separate data until a

hyperplane with high minimum distance is found. SVM is used to classify two or more data types. SVM include single or hybrid models such as Standard SVM (St-SVM), Proximal Support Vector Machine (PSVM), Newton Support Vector Machine (NSVM), Lagrangian Support Vector Machines (LSVM), Linear Programming Support Vector Machines (LPSVM), and Smooth Support Vector Machine (SSVM).

C. K-Nearest Neighbors (KNN) KNN is a supervised learning method which is used for diagnosing and classifying cancer. In this method, the computer is trained in a specific field and new data is given to it. Additionally, similar data is used by the machine for detecting (K) hence, the machine starts finding KNN for the unknown data. It is recommended to choose a large dataset for training also K value must be an odd number.

D. Decision Tree (DT) DT is a data mining technique used for early detection of breast cancer. It is a model that presents classifications or regressions as a tree. In this model, the data set is broken to small sub-data, then to smaller ones. As a result, the tree is developed and at the last level, the result is revealed. In a tree structure, the leaves characterize the class labels whereby the branches characterize conjunctions of feature leading to the class labels Hence, DT is not sensitive to noise.

E. Random Forest (RF) Algorithm RF algorithm is used at the regularization point where the model quality is highest, variance and bias problems are compromised. RF builds numerous numbers of DTs using random samples with a replacement to overcome the problem of DTs. Each tree classifies its observations, and majority votes decision is chosen. RF is used in the unsupervised mode for assessing proximities among data points.

F. AdaBoost Classifier This algorithm is used for classification and regression to predict breast cancer existence. It

converts weak learners to strong ones by combining all weak learners to form a single strong rule. It gets the weight of the node and changes it continuously until an accurate result is found. However, it is sensitive to noise and quality of features.

**G. Naïve Bayes (NB)**

**Classifier Naïve Bayes** refers to a probabilistic classifier that applies Bayes' theorem with robust independence assumptions. In this model, all properties are considered separately to detect any existing relationship between them. It assumes that predictive attributes are conditionally independent given a class. Moreover, the values of the numeric attributes are distributed within each class. NB is fast and performs well even with a small dataset. However, it is difficult to find independent properties in real life. have deployed NB classifier for breast cancer detection and it gave the maximum accuracy with only five dominant.

**IV. PREVIOUS RELATED WORKS**

Several studies have been conducted on the implementation of ML on Breast Cancer detection and diagnosis using different methods or combination of several algorithms to increase the accuracy.

S. Gc et al. worked on extracting features including variance, range, and compactness. They used SVM classification to evaluate the performance. Their findings showed the highest variance of 95%, range 94%, compactness 86%. According to their results, SVM can be considered as an appropriate method for Breast Cancer Detection.

	24%	13%	10%	6%	6%	41%
Breast						
Trachea						
Bronchus						
Lung						
Colorectum						
Ovary						
Cervix						
Uteri						
Other						
Early						

**Detection of Breast Cancer Using Machine Learning Techniques**

e-ISSN: 2289-8131 Vol. 10 No. 3-2 23

Chunqiu Wang et al. chose Microwave Tomography Imaging (MTI) to extract features and classify the images using ANN. Two different techniques were compared in this study, GMM and KNN. Their results showed that the sensitivity

obtained by KNN is 87%, while for GMM is 67%. The accuracy was 85% for KNN and 75% for GMM. The result for Matthews Correlation Coefficient (MCC) was 67% and 48% for KNN and GMM, respectively. Finally, the specificity was 84% for KNN and 86% for GMM. According to their findings, Sensitivity, Accuracy, and MCC for KNN were better than GMM, but GMM was better in Specificity and Precision.

Chowdhary and Acharjya focused on mammogram images as they are cheaper and more efficient in detection. However, since selecting and extracting features are important for improving performance, Fuzzy Histogram Hyperbolization (FHH) was chosen to increase the quality of images, Fuzzy C-mean for segmenting, and Gray level dependence model for extracting the features. Their method showed 94% accuracy for detecting malignant breast lesions. In a study conducted by Aminikhanghahi et al, wireless cyber mammography images were explored. After selecting features and extracting them, the researcher has chosen two different ML techniques, SVM and GMM to check their accuracy. Their findings showed that SVM is more accurate if there is no noise or error, else GMM is better and safer. Durai et al. Have selected Data Mining technique for detecting diseases including breast cancer. They used LRC and compared it with four other techniques including BFI, ID3, J48, and SVM. The result shows that LRC is the most accurate one with 99.25% accuracy. Wang and Yoon chose four methods of Data Mining to measure their effectiveness in detection. These models were: SVM, ANN, Naïve Bayes Classification and Adaboost tree. In addition, PCs and PCi were used for making hybrid models. After checking the accuracy, they have found out that Principal Component Analysis (PCA) can be a critical factor to improve performance. Hafizah et al.compared SVM and ANN using four



different datasets of breast and liver cancer including WBCD, BUPA JNC, Data, Ovarian. The researchers have demonstrated that both methods are having high performance but still, SVM was better than ANN. Azar and El-Said worked on six different methods of SVM. They have compared ST-SVM with LPSVM, LSVM, SSVM, PSVM, and NSVM to find out which method performs the best in accuracy, sensitivity, specificity, and ROC. LPSVM proved to be the best with accuracy 97.1429%, sensitivity 98.2456%, specificity 95.082%, and ROC 99.38%. Therefore, LPSVM has the highest performance. Deng and Perkowski used a new method called Weighted Hierarchical Adaptive Voting Ensemble (WHAVE). They compared the accuracy of WHAVE with seven other methods that had the highest accuracies in previous researchers. WHAVE proved to achieve the highest performance value of 99.8%. Rehman et al. extracted different features including Phylogenetic trees, Statistical Features and Local Binary Patterns from mammography images. They used a hybrid model combined with SVM and RBF for classification. They checked the accuracy of each feature separately. In this step the best accuracy value was 76% for 90 features that were chosen based on Taxonomic Indices based Feature (TIF) Vector, 68% for Statistical and LBP based Feature Vector, then the features were combined (Taxonomic Indices, Statistical and LBP based Feature Vector) and again checked for accuracy. The evaluation results were the best after 4 times testing. The researchers claimed that to increase performance and efficiency of detecting breast cancer is performed by using different features. Mejia et al. have chosen Thermogram images for detecting breast cancer as it is cheaper and safer than other methods. It can detect cancer in the earlier stage compared to other images or tests, and it doesn't have any limitation

such as pregnancy, size or density of breast. Also, it doesn't need any complex features for extracting. They selected 18 cases with 9 abnormal and 9 normal cases. KNN classifier was used to improve the accuracy. The results were 88.88% for abnormal and 94, 44% for normal cases. Ayeldeen et al. used AI and its techniques for breast cancer detection. They used 5 different methods for performance comparison. RF algorithm showed the highest result with 99% performance. Avramov and Si worked on feature extraction and the impact of the selection on performance. They applied 4 ways of correlation selection (PCA, T-Test Significance and Random feature selection) and 5 models of classification (LR, DT, KNN, LSVM, and CSVM). Best result was achieved by stacking the logistic, SVM and CSVM improve accuracy to 98.56%. Ngadi et al. used NSVC algorithm to test different classification methods including RBF, Poly, and Linear. Then they compared the results with other classification methods such as Naïve Bayes, DT, K-NN, SVM, RF, and Adaboost. RF has the best performance result with 93% accuracy. This proves that NSVC was better than the other methods. Jiang and Xu used Diffusion-Weighted Magnetic Resonance Image (DWI) for breast cancer detection. They used two types of features; one based on ROI and another one based on ADC- on 61 patient's data. Moreover, they implemented RF-RFE and RF algorithm was used. The study findings show that the accuracy of RF-RFE and RF and Histogram + GLCM is 77.05% which indicates that featurebased texture has a critical role in improving performance and detection. Salma selected two different data sets from WBCD and KDD also they used FM-ANN for both of them. They compared the results with other techniques (RBF, FNN, and MNN). After training and testing KDD achieved better accuracy of

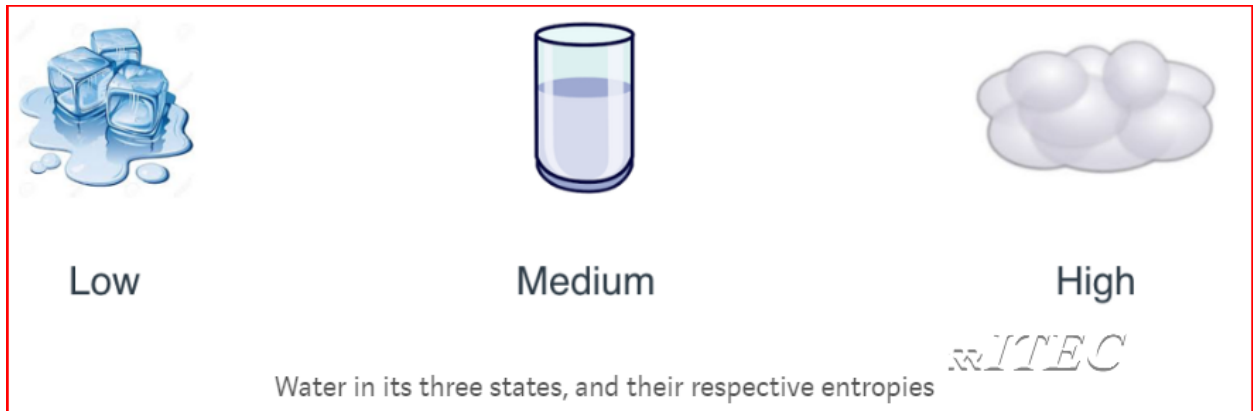
99.96% due to the number of features were more. Comparing the results FM- ANN proved to be more accurate. Bevilacqua et al. selected MR images for training and testing. After extracting data and processing, they used ANN for classification and detecting breast cancer. However, when Genetic Algorithm was used to optimize ANN, the observed specificity was 90.46%, sensitivity was 89.08% and the average accuracy was improved to 89.77% and high accuracy changed to 100%. Table 1 represents all the related work ML method used in this study. It contains the references, type of extracted features, data sets and measured performances. Performance is the most significant feature

## Entropy in Physics

Entropy, so far, had been a concept in physics. Namely, it is the (log of the) number of microstates or microscopic configurations.

In simple terms, if the particles inside a system have many possible positions to move around, then the system has **high entropy**, and if they have to stay rigid, then the system has **low entropy**

For example, water in its three states, solid, liquid, and gas, has different entropies. The molecules in ice have to stay in a lattice, as it is a rigid system, so **ice has low entropy**. The molecules in water have more positions to move around, so **water in liquid state has medium entropy**. The molecules inside water **vapor** can pretty much go anywhere they want, so water vapor has **high entropy**.



## Entropy and Knowledge:

To introduce the notion of entropy in probability, we'll use an example throughout this whole article. Let's say we have 3 buckets with 4 balls each. The balls have the following colors:

- Bucket 1: 4 red balls
- Bucket 2: 3 red balls, and 1 blue ball
- Bucket 3: 2 red balls, and 2 blue balls



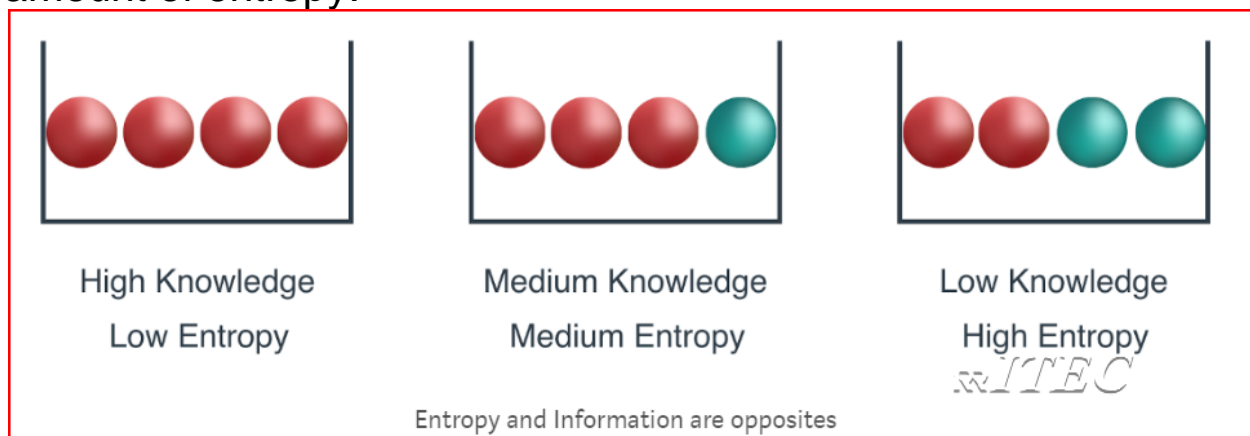
**If we drawn a ball at random.**

- In the first bucket, we'll know for sure that the ball coming out is red.
- In the second bucket, we know with 75% certainty that the ball is red, and with 25% certainty that it's blue.
- In the third bucket, we know with 50% certainty that the ball is red, and with the same certainty that it's blue.

So it makes sense to say that Bucket 1 gives us the most amount of “knowledge” about what ball we’ll draw (because we know for sure it’s red), that Bucket 2 gives us some knowledge, and that Bucket 3 will give us the least amount of knowledge.

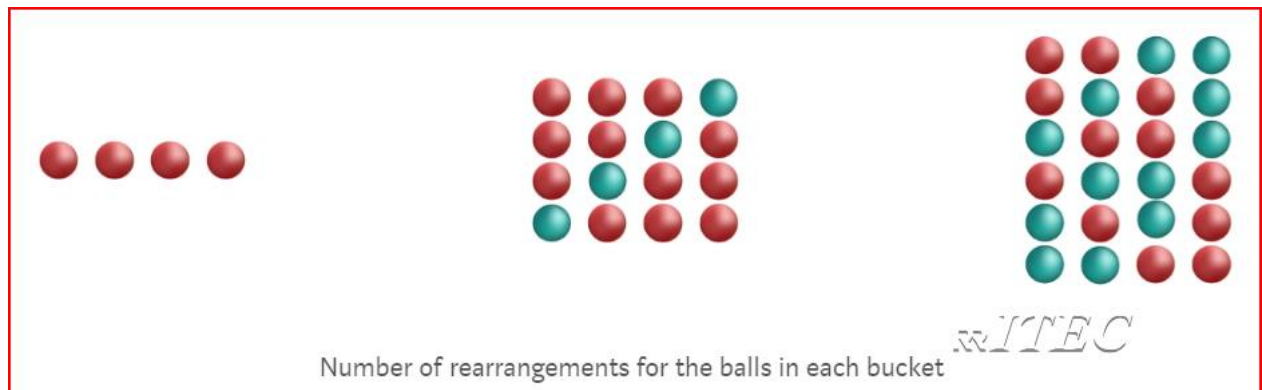
Well, **Entropy** is in some way, the **opposite of knowledge**.

So we’ll say that Bucket 1 has the least amount of entropy, Bucket 2 has medium entropy, and Bucket 3 has the greatest amount of entropy.



## Entropy and Probability:

So now the question is, how do we cook up a formula which gives us a low number for a bucket with 4 red balls, a high number for a bucket with 2 red and 2 blue balls, and a medium number for a bucket with 3 red and 1 blue balls? Well, as a first attempt, let’s remember the definition of entropy: If molecules have many possible rearrangements, then the system has high entropy, and if they have very few rearrangements, then the system has low entropy. So a first attempt would be to count the number of rearrangements of these balls. In this case, we have 1 possible rearrangement for Bucket 1, 4 for Bucket 2, and 6 for Bucket 3, this number given by the binomial coefficient.



This number of arrangements won't be part of the formula for entropy, but it gives us an idea, that if there are many arrangements, then entropy is large, and if there are very few arrangements, then entropy is low. In the next section, we'll cook up a formula for entropy. The idea is, to consider the probability of drawing the balls in a certain way, from each bucket.

### **Entropy and an Interesting Game Show**

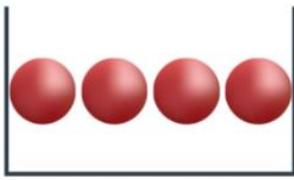
So, in order to cook up a formula, we'll consider the following game. The spoiler is the following: The probability of winning this game, will help us get the formula for entropy.

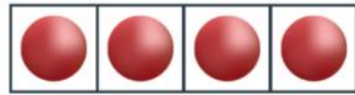
In this game, we're given, again, the three buckets to choose. The rules go as follows:

We choose one of the three buckets. We are shown the balls in the bucket, in some order. Then, the balls go back in the bucket. We then pick one ball out of the bucket, at a time, record the color, and return the ball back to the bucket. If the colors recorded make the same sequence than the sequence of balls that we were shown at the beginning, then we win 1,000,000 dollars. If not, then we lose. This may sound complicated, but it's actually very simple. Let's say for example that we've picked Bucket 2, which has 3 red balls, and 1 blue ball. We're shown the balls in the bucket in some order, so let's say, they're shown to us in that precise order, red, red, red, blue. Now, let's try to draw the balls to get that sequence, red, red, red, blue. What's the probability of this happening? Well...

In order for the first ball to be red, the probability is  $3/4$ , or 0.75. For the second ball to be red, the probability is again  $3/4$ . (Remember that we put the first ball back in the bucket after looking at its color.) For the third ball to be red, the probability is again  $3/4$ . For the fourth ball to be blue, the probability is now  $1/4$ , or 0.25. As these are independent events, then the probability of the 4 of them to happen, is  $(3/4)(3/4)(3/4)(1/4) = 27/256$ , or 0.105. This is not very likely. In the figures below, we can see the probabilities of winning if we pick each of the three buckets.

For exposition, the following three figures show the probabilities of winning with each of the buckets. For Bucket 1, the probability is 1, for Bucket 2, the probability is 0.105, and for Bucket 3, the probability is 0.0625.

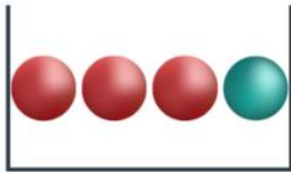


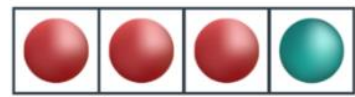

$$1 \times 1 \times 1 \times 1 = 1$$

Win lots of money!



Probability of winning with Bucket 1 is 1





$$0.75 \times 0.75 \times 0.75 \times 0.25 = 0.105$$

Win lots of money!



Probability of winning with Bucket 2 is 0.105




$$0.5 \times 0.5 \times 0.5 \times 0.5 = 0.0625$$

Win lots of money!



Probability of winning with Bucket 3 is 0.0625

Probability of winning with Bucket 1 is 1

Probability of winning with Bucket 2 is 0.105

Probability of winning with Bucket 3 is 0.0625

Ok, now we have some measure that gives us different values for the three Buckets. The probability of winning at this game, gives us:

- 1.0 for Bucket 1
- 0.105 for Bucket 2
- 0.0625 for Bucket 3



In order to build the entropy formula, we want the opposite, some measure that gives us a low number for Bucket 1, a medium number for Bucket 2, and a high number for Bucket 3. No problem, this is where logarithms will come to save our life.

### **Turning Products into Sums**

The following is a very simple trick, yet used very widely, particularly in Machine Learning. See, products are never very good. Here we have a product of 4 numbers, which is not bad, but imagine if we had a million data points. How would the product of a million small probabilities (between 0 and 1) would look? It would be a ridiculously tiny number. In general we want to avoid products as much as we can. What's better than a product? Well, a sum! And how do we turn products into sums? Exactly, using the logarithm function, since the following identity will be very helpful:

$$\log(a*b) = \log a + \log b$$

Logarithm identity So, what do we do? Well, we have a product of four things, we take the logarithm, and that becomes the sum of four things. In the case of Bucket 2 (3 red balls, 1 blue ball), we have the following:

$$0.75 * 0.75 * 0.75 * 0.25 = 0.10546875$$

And taking the logarithm (in this case, we'll take the logarithm, and multiply by -1, to make things positive), we get:

- $-\log(0.75 * 0.75 * 0.75 * 0.25) = 3.245$
- $-\log(0.75) - \log(0.75) - \log(0.75) - \log(0.25) = 3.245$

For purposes of this post, we'll take logarithm base 2

Now, as a final step, we take the average, in order to normalize. And that's it, that's the entropy! For Bucket 2, it's 0.811:

$$1/4 * (-\log(0.75) - \log(0.75) - \log(0.75) - \log(0.25)) = 0.81125$$

If we calculate the entropy **for Bucket 1** (4 red balls), we get:


$$1/4 * (-\log(1) - \log(1) - \log(1) - \log(1)) = 0$$

And for Bucket 3 (2 red balls, 2 blue balls), we get:

$$1/4 * (-\log(0.50) - \log(0.5) - \log(0.5) - \log(0.5)) = 1$$

So we have our formula for entropy, **the negative logarithm of the probability** of winning at our game. Notice that this is low for Bucket 1, high for Bucket 3, and medium for Bucket 2.

For the formula lovers out there, the general formula is as follows. If our bucket has  $m$  red balls, and  $n$  blue balls, the formula is as follows:



$$\text{Entropy} = \frac{-m}{m+n} \log_2 \left( \frac{m}{m+n} \right) + \frac{-n}{m+n} \log_2 \left( \frac{n}{m+n} \right)$$

General formula for Entropy

## Multi-class Entropy:

So far we've been dealing with two classes, red and blue. In order to relate Entropy with Information Theory, we need to look at entropy with several classes. Let's switch to letters, to make this more clear. We have the following three buckets, with 8 letters each. Bucket 1 has the letters AAAAAAAAAA, Bucket 2 has the letters AAAABBCD, and Bucket 3 has the letters AABBCDD. While it's straightforward to see that Bucket 1 has the least amount of entropy, the difference between Bucket 2 and Bucket 3 is not obvious. We'll see below that Bucket 3 has the highest

entropy of the three, while Bucket 2 has medium



The formula for entropy generalizes very easily to more classes. This is the general formula

$$Entropy = -\sum_{i=1}^n p_i \log_2 p_i$$

Where there are  $n$  classes, and  $p_i$  is the probability an object from the  $i$ -th class appearing. For our three buckets, we have the following:

In this case, since Bucket 1 has only one class (the letter A), and the probability of it appearing is 1, then the entropy is:

### For Bucket 1

$$Entropy = -1 * \log(1) = 0$$

### For Bucket 2:

since we have 4 classes (the letters A, B, C, and D), and the probability of A appearing is 4/8, for B it's 2/8, for C it's 1/8, and for D it's 1/8, then the entropy is:

$$Entropy = -4/8 * \log_2(4/8) - 2/8 * \log_2(2/8) - 1/8 * \log_2(1/8) - 1/8 * \log_2(1/8) = 1.75$$

**For Bucket 3**, since we have 4 classes (the letters A, B, C, and D), and the probability of each appearing is 1/4, then the entropy is:

$$Entropy = -2/4 * \log_2(2/4) - 2/4 * \log_2(2/4) - 2/4 * \log_2(2/4) - 2/4 * \log_2(2/4) = 2$$

Ok, so we've calculated the entropy for our three buckets

<b>AAAAAAAA</b>	<b>AAAABBCD</b>	<b>AABBCDD</b>
Bucket 1	Bucket 2	Bucket 3
Entropy = 0	Entropy = 1.75	Entropy = 2
Entropy for the three buckets		
<i>WITTEC</i>		

## Information Theory

Here's another way to see entropy. Let's say we want to draw a random letter from one of the buckets. On average, how many questions do we need to ask to find out what letter it is?

First, let's get the easy case out of the way. If the bucket is Bucket 1, we know for sure that the letter is an A. So right there, we know that for Bucket 1, we need to ask 0 questions on average, to guess what letter we got. For the sake of redundancy, let's put it in a formula:

### Average Number of questions = 0

Now, for buckets 2 and 3, naively, one would think that 4 questions is enough to find out any letter. Namely, the following four questions would be enough:

- Is the letter an A?
- Is the letter a B?
- Is the letter a C?
- Is the letter a D?

So, first off, the fourth question is redundant, since if the answer to all the previous ones is "no", then we know for sure that the letter is a D. So three questions is enough. Now, can we do better than that? Well, our questions don't need to be independent. We can tailor our question 2 based on the answer to question 1, as follows:

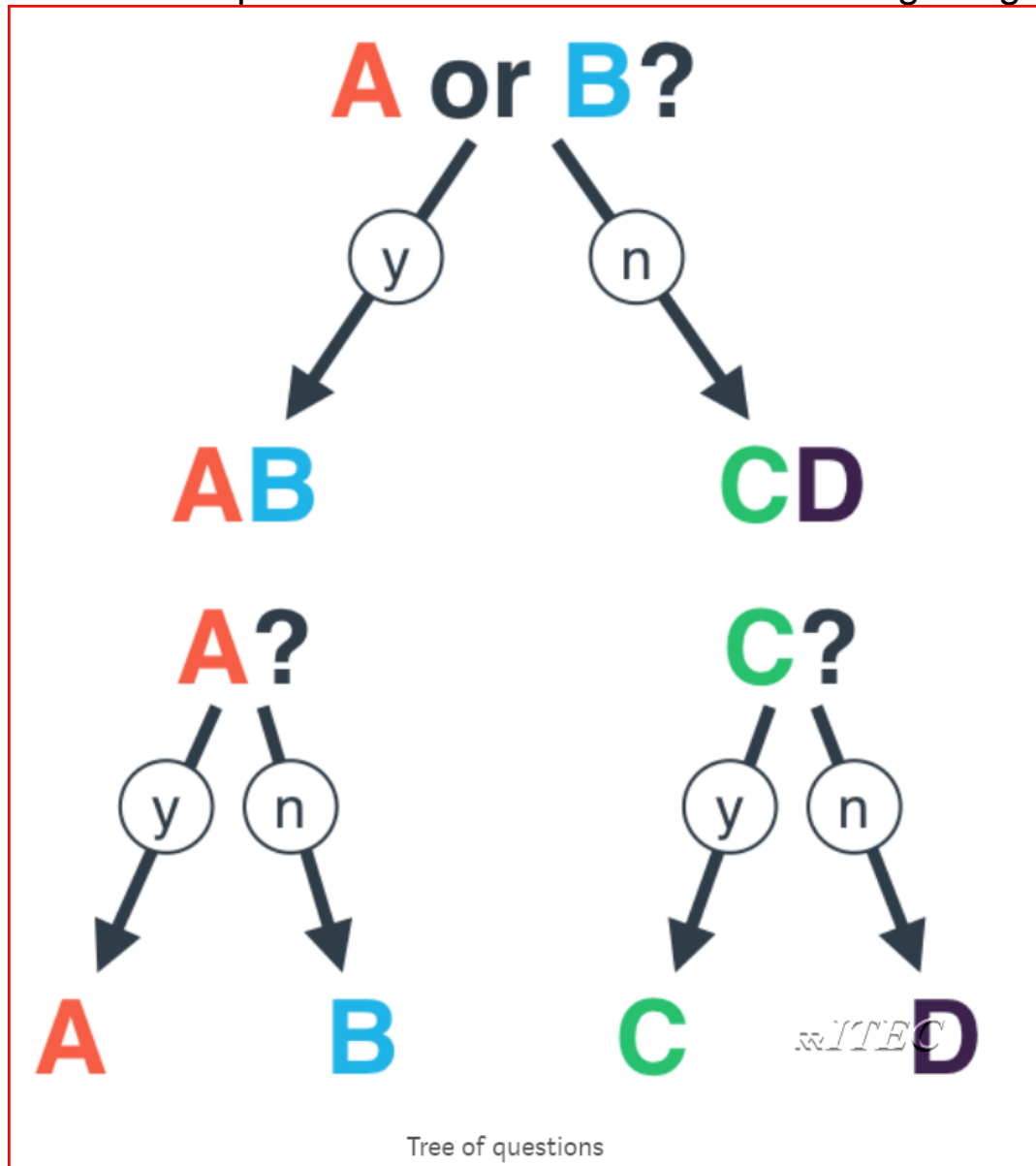
- Is the letter A or B?

a) If the answer to question 1 is “yes”: Is the letter A? If the answer to question 1 is “no”: Is the letter C?

And that will actually do it, because based on the two answers, we get the following:

- “Yes” and “Yes”: Letter is A
- “Yes” and “No”: Letter is B
- “No” and “Yes”: Letter is C
- “No” and “No”: Letter is D

This tree of questions can be seen in the following image:

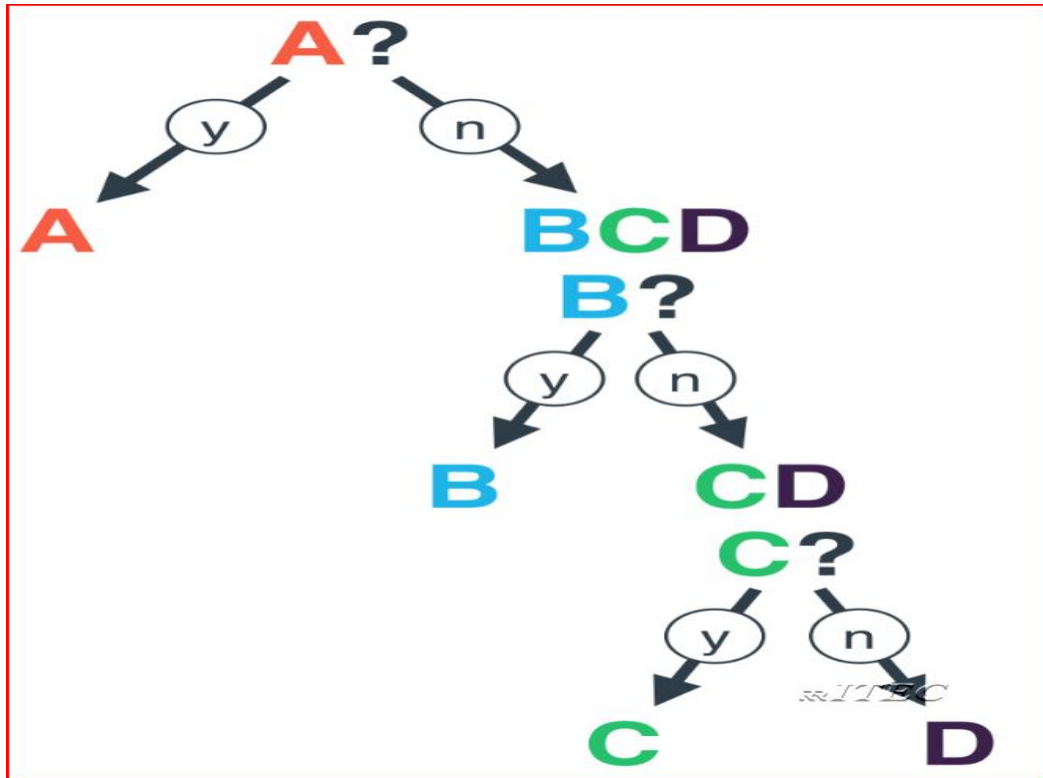


Now, for Bucket 3, each letter appears with probability  $1/4$ , since there are 8 letters, and 2 of each. Thus, the average number of questions to find out the letter drawn out of Bucket 2 is precisely 2, as the next formula states:

$$\text{Average Number of questions} = \frac{1}{4} * 2 + \frac{1}{4} * 2 + \frac{1}{4} * 2 + \frac{1}{4} * 2 = 2$$

Now, \*\* let's look at Bucket 1 \*\*. Of course, if we use the same question tree as we used for Bucket 2, we can see that the

average number of questions is 2. But we can do a bit better. Actually, let's use the first attempt. First asking if the letter is A, then B, then C. That's the following tree:



In this case, we have the following:

- If the letter is A, we found out in 1 question.
- If the letter is B, we found out in 2 questions.
- If the letter is C or D, we found out in 3 questions.

Now the trick is the following. A appears much more often than C and D, so on average, we may be doing much better. How much better? Well, recall that Bucket 2 has the letters AAAABBCD, so A appears 1/2 the time, B appears 1/4 of the time, and C and D appear each 1/8 of the time. So the average number of questions is:

$$\text{Average Number of questions} = 1/2 * 1 + 1/4 * 2 + 1/8 * 3 + 1/8 * 3 = 1.75$$

Well, that's exactly the entropy! Here's the connection between Entropy and Information Theory. If we want to find out a letter drawn out of a bucket, the average number of questions we must ask to find out (if we ask our questions in the smartest possible way), is at least the entropy of the set. This means, the entropy of the set is a lower bound on the number of questions we must ask in average to find out. In the cases we saw above, the number of questions is exactly the entropy. In general, this won't happen, we may need to ask more questions than the entropy. But we will never be able to do it with less questions than the entropy of the set.

## **Tree Based Models:**

**Understand entropy and gini calculations**

**From Google Drive:Google Drive\01 DS ML DL NLP and AI With Python Lab Copy\02 Lab Data\Python\tree\_weather.xlsx**

**From Github: [Refer excel worksheet](#)**

- 1. What is decision tree?**
  - a. Sequence of if-else questions about individual features**
  - b. Decision-Tree has a hierarchy of nodes.**
- 2. Node: question or prediction.**
- 3. Building Blocks of a Decision-Tree are three nodes**
  - a. Root: no parent node, question giving rise to two children nodes.**
  - b. Internal node: one parent node, question giving rise to two children nodes.**



- c. Leaf: one parent node, no children nodes ==> \*\*  
prediction \*\*

## 1. Decision Tree for Classification

*1.1 Business Problem Statement: predict whether a tumor is malignant(cancer) or benign(non Cancer) based on two features the mean radius of the tumor (radius\_mean) and its mean number of concave points (concave points\_mean)*

### Step 1: Import required Modules

```
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
DecisionTreeClassifier?
```

### Step 2: Load data

```
#os.chdir('C:\\Users\\Hi\\Google Drive\\01 Data Science Lab Copy\\02
Lab Data\\Python')
```

```
df = pd.read_csv('Wisconsin_Breast_Cancer_Dataset.csv')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 33 columns):
```

id	569	non-null	int64
diagnosis	569	non-null	object
radius_mean	569	non-null	float64
texture_mean	569	non-null	float64
perimeter_mean	569	non-null	float64
area_mean	569	non-null	float64
smoothness_mean	569	non-null	float64
compactness_mean	569	non-null	float64
concavity_mean	569	non-null	float64

```

concave points_mean      569 non-null float64
symmetry_mean            569 non-null float64
fractal_dimension_mean   569 non-null float64
radius_se                569 non-null float64
texture_se               569 non-null float64
perimeter_se             569 non-null float64
area_se                  569 non-null float64
smoothness_se            569 non-null float64
compactness_se           569 non-null float64
concavity_se             569 non-null float64
concave points_se        569 non-null float64
symmetry_se              569 non-null float64
fractal_dimension_se     569 non-null float64
radius_worst             569 non-null float64
texture_worst            569 non-null float64
perimeter_worst          569 non-null float64
area_worst               569 non-null float64
smoothness_worst         569 non-null float64
compactness_worst        569 non-null float64
concavity_worst          569 non-null float64
concave points_worst     569 non-null float64
symmetry_worst           569 non-null float64
fractal_dimension_worst  569 non-null float64
Unnamed: 32              0 non-null float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
X = df[["radius_mean", "concave points_mean"]]
y = df["diagnosis"]
X[15:25]

```

Out[5]:

```
radius_mean  concave points_mean
```

15	14.540	0.07364
----	--------	---------

	radius_mean	concave points_mean
16	14.680	0.05259
17	16.130	0.10280
18	19.810	0.09498
19	13.540	0.04781
20	13.080	0.03110
21	9.504	0.02076
22	15.340	0.09756
23	21.160	0.08632
24	16.650	0.09170

y[15:25]

Out[6]:

15	M
16	M
17	M
18	M
19	B
20	B
21	B

```
22      M
23      M
24      M
Name: diagnosis, dtype: object
```

```
y = y.replace('M',1)
y = y.replace('B',0)
SEED = 1 # for reproducing
```

**Step 3: Create training and test sets**

```
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y, test_size = 0.2, random_state=SEED, stratify=y)
```

```
print(X_train.shape) # (455, 2)
print(y_train.shape) # (455,)
print(X_test.shape) # (114, 2)
print(y_test.shape) # (114,)
(455, 2)
(455,)
(114, 2)
(114,)
```

**Step 4: Create DecisionTreeClassifier Model with a maximum depth of 6**

```
dt = DecisionTreeClassifier(max_depth=6,
                           random_state=SEED,
                           criterion='gini')
```

*# Fit dt to the training set*

```
dt.fit(X_train, y_train)
```

**Out[10]:**

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=6,
                       max_features=None, max_leaf_nodes=None,
```

```

        min_impurity_decrease=0.0, min_impur
ity_split=None,
        min_samples_leaf=1, min_samples_spli
t=2,
        min_weight_fraction_leaf=0.0, presor
t=False, random_state=1,
        splitter='best')

```

**Step 5: Predict test set labels using model**

```
y_pred = dt.predict(X_test)
```

**Step 6: Test the Performance**

*# Compute test set accuracy*

```
acc = accuracy_score(y_test, y_pred)
```

```
print("Test set accuracy: {:.2f}".format(acc))
```

```
Test set accuracy: 0.89
```

```
print(confusion_matrix(y_test, y_pred))
```

```
[[65  7]
 [ 6 36]]
(65+36)/114
```

**Out[13]:**

```
0.8859649122807017
```

*Note: Not bad! Using only two features, your tree was able to achieve an accuracy of 89% :)*

**LogisticRegression Vs Decision Tree Classification**

*# Import LogisticRegression from sklearn.linear\_model*

```
from sklearn.linear_model import LogisticRegression
```

*# Instantiate logreg*

```
logreg = LogisticRegression(random_state=1)
```

*# Fit logreg to the training set*

```
logreg.fit(X_train, y_train)
```

**Out[22]:**

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                    penalty='l2', random_state=1, solver='liblinear', tol=0.0001,
                    verbose=0, warm_start=False)
```

*# predict*

```
y_pred1 = logreg.predict(X_test)
acc1 = accuracy_score(y_test, y_pred1)
acc1
```

**Out[25] :**

0.9122807017543859

By observing accuracies, On this data set which algorithm is good ?

○ DT Classifier ○ Logistic Regression

Home Work : Similarly apply SVC and KNN on this data set

## Coding

**Step 1: Import Required Modules**

```
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

## Step 2: Load Data

```
#os.chdir("C:\\Users\\Hi\\Google Drive\\01 Data Science Lab Copy\\02  
Lab Data\\Python")
```

```
df = pd.read_csv("Wisconsin_Breast_Cancer_Dataset.csv")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 33 columns):
```

id	569 non-null	int64
diagnosis	569 non-null	object
radius_mean	569 non-null	float64
texture_mean	569 non-null	float64
perimeter_mean	569 non-null	float64
area_mean	569 non-null	float64
smoothness_mean	569 non-null	float64
compactness_mean	569 non-null	float64
concavity_mean	569 non-null	float64
concave points_mean	569 non-null	float64
symmetry_mean	569 non-null	float64
fractal_dimension_mean	569 non-null	float64
radius_se	569 non-null	float64
texture_se	569 non-null	float64
perimeter_se	569 non-null	float64
area_se	569 non-null	float64
smoothness_se	569 non-null	float64
compactness_se	569 non-null	float64
concavity_se	569 non-null	float64
concave points_se	569 non-null	float64
symmetry_se	569 non-null	float64
fractal_dimension_se	569 non-null	float64
radius_worst	569 non-null	float64
texture_worst	569 non-null	float64
perimeter_worst	569 non-null	float64
area_worst	569 non-null	float64
smoothness_worst	569 non-null	float64

```
compactness_worst      569 non-null float64
concavity_worst        569 non-null float64
concave points_worst   569 non-null float64
symmetry_worst         569 non-null float64
fractal_dimension_worst 569 non-null float64
Unnamed: 32            0 non-null float64
```

```
dtypes: float64(31), int64(1), object(1)
```

```
memory usage: 146.8+ KB
```

```
X = df.iloc[:,2:32]
```

```
type(X)
```

```
Out[3]:
```

```
pandas.core.frame.DataFrame
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 30 columns):
```

```
radius_mean      569 non-null float64
texture_mean     569 non-null float64
perimeter_mean   569 non-null float64
area_mean        569 non-null float64
smoothness_mean  569 non-null float64
compactness_mean 569 non-null float64
concavity_mean   569 non-null float64
concave points_mean 569 non-null float64
symmetry_mean    569 non-null float64
fractal_dimension_mean 569 non-null float64
radius_se        569 non-null float64
texture_se       569 non-null float64
perimeter_se     569 non-null float64
area_se          569 non-null float64
smoothness_se    569 non-null float64
compactness_se   569 non-null float64
concavity_se     569 non-null float64
concave points_se 569 non-null float64
symmetry_se      569 non-null float64
```



```
fractal_dimension_se      569 non-null float64
radius_worst              569 non-null float64
texture_worst             569 non-null float64
perimeter_worst          569 non-null float64
area_worst                569 non-null float64
smoothness_worst          569 non-null float64
compactness_worst         569 non-null float64
concavity_worst           569 non-null float64
concave points_worst      569 non-null float64
symmetry_worst            569 non-null float64
fractal_dimension_worst   569 non-null float64
dtypes: float64(30)
```

```
memory usage: 133.4 KB
```

```
y=df["diagnosis"]
```

```
y[15:25]
```

```
Out[8]:
```

```
15      M
16      M
17      M
18      M
19      B
20      B
21      B
22      M
23      M
24      M
```

```
Name: diagnosis, dtype: object
```

```
y = y.replace('M',1)
```

```
y = y.replace('B',0)
```

```
SEED = 1 #for reproducing
```

Step 3: Create Training and Test sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.2, random_state=SEED, stratify=y)
```

```
X_train.shape # (455, 2)
```

```
Out[12]:
```

```
(455, 30)
```

```
y_train.shape # (455,)
```

```
Out[13]:
```

```
(455,)
```

```
X_test.shape # (114, 2)
```

```
Out[14]:
```

```
(114, 30)
```

```
y_test.shape # (114,)
```

```
Out[15]:
```

```
(114,)
```

**Step 4: Create Model using criterion as *entropy***

*# Create dt\_entropy model, set 'entropy' as the information criterion*

```
dt_entropy = DecisionTreeClassifier(max_depth=8,  
                                     criterion='entropy',  
                                     random_state=SEED)
```

*# Fit dt\_entropy to the training set*

```
dt_entropy.fit(X_train, y_train)
```

```
Out[17]:
```

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=8,  
                        max_features=None, max_leaf_nodes=None,  
                        min_impurity_decrease=0.0, min_impurity_split=None,  
                        min_samples_leaf=1, min_samples_split=2,  
                        min_weight_fraction_leaf=0.0, presort=False, random_state=1,  
                        splitter='best')
```

*# Use dt\_entropy to predict test set labels*

```
y_pred = dt_entropy.predict(X_test)
# Evaluate accuracy_entropy
accuracy_entropy = accuracy_score(y_test, y_pred)
accuracy_entropy
```

Out[20]:

0.9298245614035088

Step 5: Create Model using criterion as gini

*# Instantiate dt\_gini, set 'gini' as the information criterion*

```
dt_gini = DecisionTreeClassifier(max_depth=8,
                                criterion='gini',
                                random_state=SEED)
```

*# Fit dt\_entropy to the training set*

```
dt_gini.fit(X_train, y_train)
```

Out[22]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=8,
                        max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort=False, random_state=1,
                        splitter='best')
```

*# Use dt\_entropy to predict test set labels*

```
y_pred_gini = dt_gini.predict(X_test)
```

*# Evaluate accuracy\_gini*

```
accuracy_gini = accuracy_score(y_test, y_pred_gini)
accuracy_gini
```

Out[25]:

0.9298245614035088

**Step 6: compare entropy and gini accuracy**

*# Print accuracy\_entropy*

```
print('Accuracy achieved by using entropy: ', accuracy_entropy)
```

*# Print accuracy\_gini*

```
print('Accuracy achieved by using the gini index: ', accuracy_gini)
```

```
Accuracy achieved by using entropy:  0.9298245614035088
```

```
Accuracy achieved by using the gini index:  0.9298245614035088
```

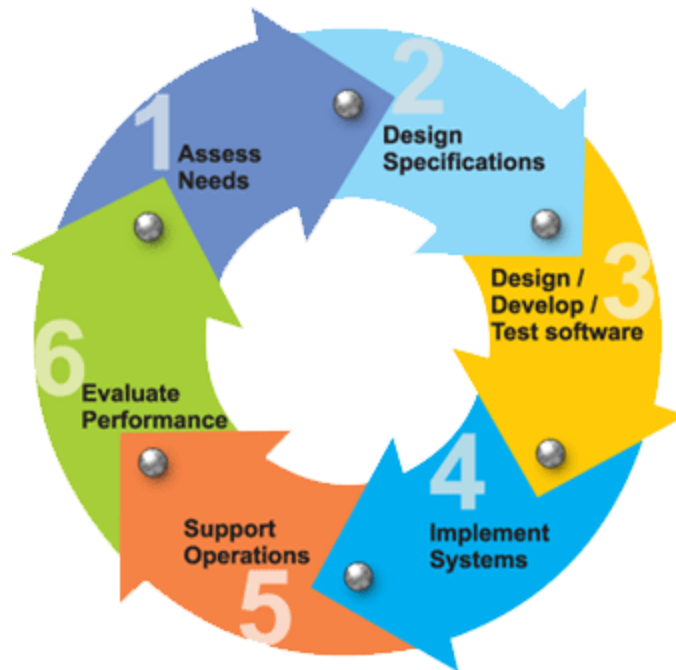
**Note:** Notice how the two models achieve exactly the same accuracy. Most of the time, the gini index and entropy lead to the same results. The gini index is slightly faster to compute and is the default criterion used in the DecisionTreeClassifier model of scikit-learn.

## **SYSTEM ANALYSIS**

### **1) INTRODUCTION**

#### **Software Development Life Cycle:-**

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development.



### **Requirements:-**

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.

### **Design**

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this

phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

### **Implementation**

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation may overlap with both the design and testing phases. Many tools exist (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

### **Testing**

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model. Now let's delve into some of the traditional and widely used variations.

### **SDLC METHDOLOGIES**

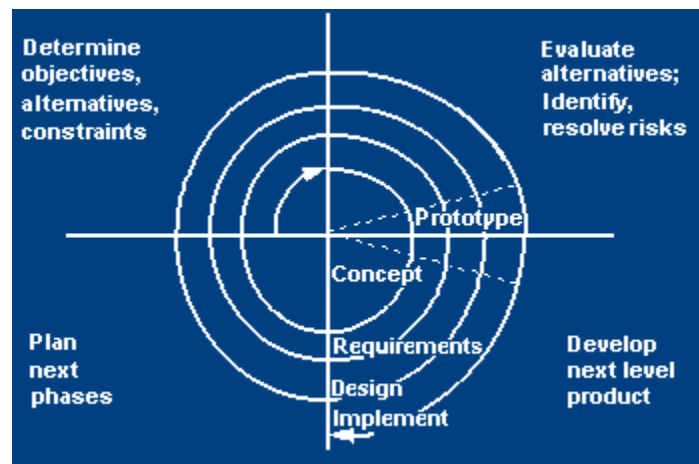
This document plays a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basis during testing phase. Any changes

made to the requirements in the future will have to go through formal change approval process.

**SPIRAL MODEL** was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

**The following diagram shows how a spiral model acts like:**



The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
  1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
  2. Defining the requirements of the second prototype.
  3. Planning an designing the second prototype.
  4. Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.



- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

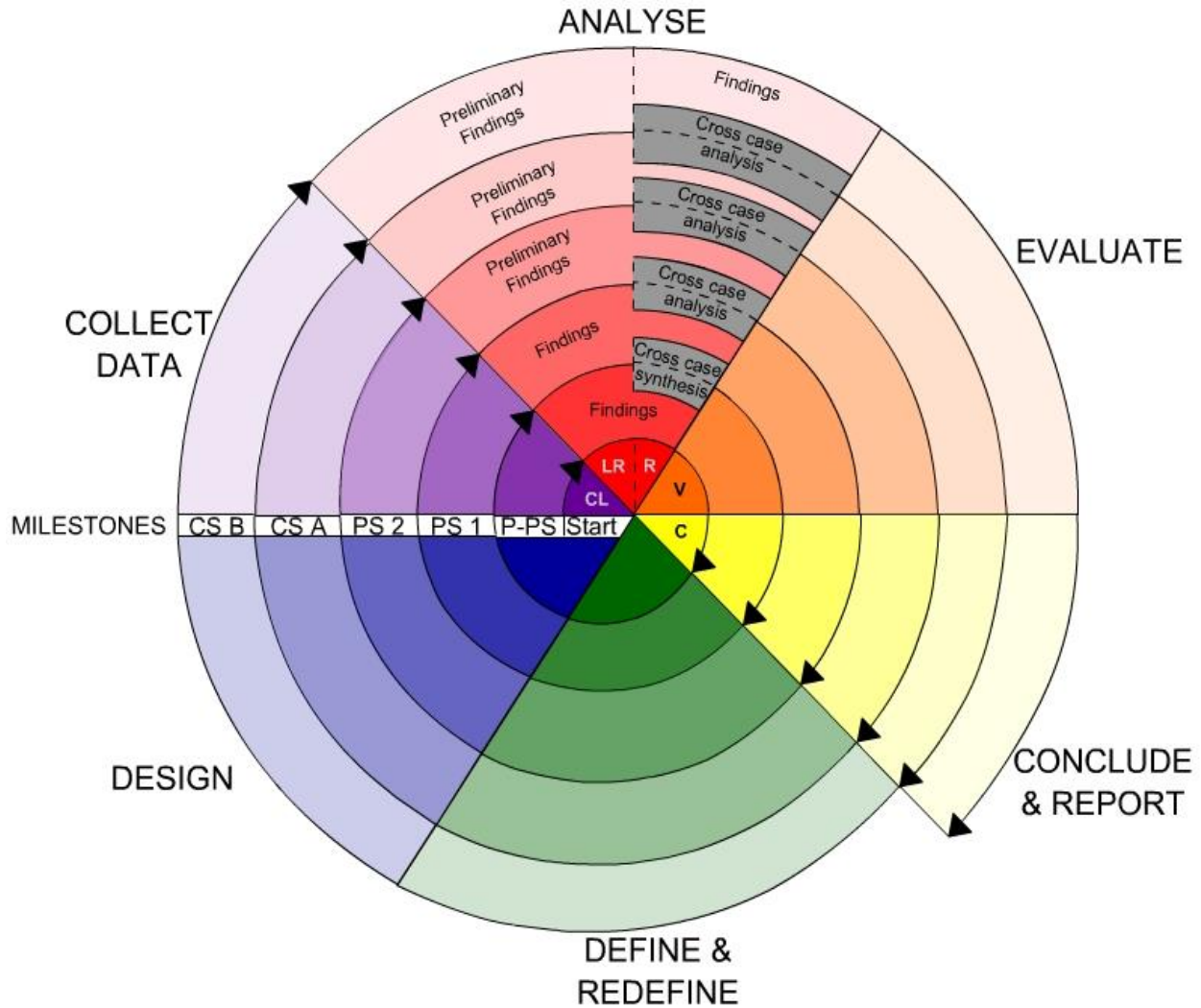
## **2) STUDY OF THE SYSTEM**

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows

1. Administrative User Interface Design
2. The Operational and Generic User Interface Design

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.



### **3) Hardware and Software requirements**

What is Cloudera:-

Cloudera is revolutionizing enterprise data management by offering the first unified Platform for Big Data: The Enterprise Data Hub. Cloudera offers enterprises one place to store, process, and analyze all their data, empowering them to extend the value of existing investments while

enabling fundamental new ways to derive value from their data.

Why do customers choose Cloudera:-

Cloud era was the first commercial provider of python-related software and services and has the most customers with enterprise requirements, and the most experience supporting them, in the industry. Cloud era's combined offering of differentiated software (open and closed source), support, training, professional services, and indemnity brings customers the greatest business value, in the shortest amount of time, at the lowest TCO.

### **Data Mining**

There is a huge amount of data available in the Information Industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it. Extraction of information is not the only process we need to perform; data mining also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation. Once all these processes are over, we would be able to use this information in many applications such as Fraud Detection, Market Analysis, Production Control, Science Exploration, etc.

What is Data Mining?

Data Mining is defined as extracting information from huge sets of data. In other words, we can say that data mining is the procedure of mining knowledge from data. The information or knowledge extracted so can be used for any of the following applications:

Market Analysis

Fraud Detection

Customer Retention

Production Control

Science Exploration

Data Mining Applications

Data mining is highly useful in the following domains: Market Analysis and Management

Corporate Analysis

& Risk Management Fraud Detection

Apart from these, data mining can also be used in the areas of production control, customer retention, science exploration, sports, astrology, and Internet Web Surf-Aid

Market Analysis and Management

Listed below are the various fields of market where data mining is used:  
Customer Profiling - Data mining helps determine what kind of people buy what kind of products.

Identifying Customer Requirements - Data mining helps in identifying the best products for different customers. It uses prediction to find the factors that may attract new customers.

Cross Market Analysis - Data mining performs Association/correlations between product sales.

Target Marketing - Data mining helps to find clusters of model customers who share the same characteristics such as interests, spending habits, income, etc.

Determining Customer purchasing pattern - Data mining helps in determining customer purchasing pattern.

Providing Summary Information - Data mining provides us various multidimensional summary reports.

### Corporate Analysis and Risk Management

Data mining is used in the following fields of the Corporate Sector:

Finance Planning and Asset Evaluation - It involves cash flow analysis and prediction, contingent claim analysis to evaluate assets.

Resource Planning - It involves summarizing and comparing the resources and spending. Competition - It involves monitoring competitors and market directions.

## Fraud Detection

Data mining is also used in the fields of credit card services and telecommunication to detect frauds. In fraud telephone calls, it helps to find the destination of the call, duration of the call, time of the day or week, etc. It also analyzes the patterns that deviate from expected norms.

Data mining essential step in the process of knowledge discovery

- 1.Data cleaning (to remove noise and inconsistent data)
2. Data integration (where multiple data sources may be combined)
3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)

7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

Based on this view, the architecture of a typical data mining system may have the following major components

Database, data warehouse, World Wide Web, or other information repository: This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories.

Data cleaning and data integration techniques may be performed on the data. Database or data warehouse server: The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional

interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).

**Data mining engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.

**Pattern evaluation module:** This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used.

For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

**User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based



on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

## **DATA MINING ON WHAT KIND OF DATA**

### **RELATIONAL DATABASES**

A database system, also called a database management system (DBMS), consists of a collection of interrelated data, known as a database, and a set of software programs to manage and access the data. A relational database is a collection of tables, each of which is assigned a unique name. Each table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows).

Each tuple in a relational table represents an object identified by a unique key and described by a set of attribute values. A semantic data model, such as an entity-relationship (ER) data model, is often constructed for relational databases. An ER data model represents the database as a set of entities and their relationships. Relational data can be accessed by database queries written in a relational query language, such as SQL.

## **DATA WAREHOUSES**

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and that usually resides at a single site Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing

## **OBJECT-RELATIONAL DATABASES**

Based on an object-relational data model Extends the relational model by providing a rich data type for handling complex objects and object orientation Objects that share a common set of properties can be grouped into an object class. Each object is an instance of its class. Object classes can be organized into class/subclass hierarchies

## **ADVANCED DATA AND INFORMATION SYSTEMS**

With the progress of database technology, various kinds of advanced data and information systems have emerged and are undergoing development to address the requirements of new applications handling spatial/temporal data (such as maps) engineering design data (such as the design of buildings, system components, or integrated circuits) hypertext and multimedia data (including text, image, video, and audio data) time-related data (such as historical records or stock exchange data) stream data (such as video surveillance and sensor data, where data flow in and out

like streams) the World Wide Web (a huge, widely distributed information repository made available by the Internet)

## **THE WORLD WIDE WEB**

The World Wide Web and its associated distributed information services, such as Yahoo! and Google provide rich, worldwide, on-line information services, where data objects are linked together to facilitate interactive access Capturing user access patterns in such distributed information environments is called Web usage mining (or Weblog mining)

Database or data warehouse server responsible for fetching the relevant data, based on the user's data mining request can be decouples/loose coupled/tightly coupled with the database layer

Knowledge base the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns interestingness constraints or thresholds, metadata, concept hierarchies, etc.

Data mining engine this is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis query languages (DMQL) based on mining primitives to access the data

Pattern evaluation module interacts with the data mining modules so as to focus the search toward interesting patterns may use interestingness

thresholds to filter out discovered patterns may be integrated with the mining module

User interface communicates between users and the data mining system allows the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms

### **System Analysis**

The **Systems Development Life Cycle (SDLC)**, or *Software Development Life Cycle* in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems.

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

### **SOFTWARE MODEL OR ARCHITECTURE ANALYSIS:**

Structured project management techniques (such as an SDLC) enhance management's control over projects by dividing complex tasks into manageable sections. A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. But none of the SDLC models discuss the key issues like Change management, Incident management and Release management processes within the SDLC process, but, it is addressed in the overall project management. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. The —one size fits all approach to applying SDLC methodologies is no longer appropriate. We have made an attempt to address the above mentioned defects by using a new hypothetical model for SDLC described elsewhere. The drawback of addressing these management processes under the overall project management is missing of key technical issues pertaining to software development process that is, these issues are talked in the project management at the surface level but not at the ground level.

#### **4) Proposed System:**

Our goal is to implement machine learning model in order to classify, to the highest possible degree of accuracy, credit card fraud from a dataset gathered from Kaggle. After initial data exploration, we knew we would implement a logistic regression model for best accuracy reports.

Logistic regression, as it was a good candidate for binary classification. Python sklearn library was used to implement the project, We used Kaggle datasets for Credit card fraud detection, using pandas to data frame for class ==0 for no fraud and class==1 for fraud, matplotlib for plotting the fraud and non fraud data, train\_test\_split for data extraction (Split arrays or matrices into random train and test subsets) and used Logistic Regression machine learning algorithm for fraud detection and print predicting score according to the algorithm. Finally Confusion matrix was plotted on true and predicted.

### **5) Functional requirements**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,.
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.

- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.
- Understanding user's preferences, expertise level and his business requirements through a friendly questionnaire.
- Input data can be in four different forms - Relational DB, text files, .xls and xml files. For testing and demo you can choose data from any domain. User-B can provide business data as input.

### **Non-Functional Requirements**

1. Secure access of confidential data (user's details). SSL can be used.
2. 24 X 7 availability.
3. Better component design to get better performance at peak time
4. Flexible service based architecture will be highly desirable for future extension

## **FEASIBILITY REPORT**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical

feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

### **1) TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is



technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

## **2) OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?

- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

### **3) ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed

using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

### **Determining Economic Feasibility:**

Assessing the economic feasibility of an implementation by performing a cost/benefit analysis, which as its name suggests compares the full/real costs of the application to its full/real financial benefits. The alternatives should be evaluated on the basis of their contribution to net cash flow, the amount by which the benefits exceed the costs, because the primary objective of all investments is to improve overall organizational performance.

Type	Potential Costs	Potential Benefits
Quantitative	<ul style="list-style-type: none"><li>• Hardware/software upgrades</li><li>• Fully-burdened cost of labor (salary + benefits)</li><li>• Support costs for the application</li><li>• Expected operational costs</li><li>• Training costs for users to learn the application</li><li>• Training costs to train developers in new/updated technologies</li></ul>	<ul style="list-style-type: none"><li>• Reduced operating costs</li><li>• Reduced personnel costs from a reduction in staff</li><li>• Increased revenue from additional sales of your organizations products/services</li></ul>

Qualitative	<ul style="list-style-type: none"> <li>Increased employee dissatisfaction from fear of change</li> </ul>	<ul style="list-style-type: none"> <li>Improved decisions as the result of access to accurate and timely information</li> <li>Raising of existing, or introduction of a new, barrier to entry within your industry to keep competition out of your market</li> <li>Positive public perception that your organization is an innovator</li> </ul>
-------------	--	---

The table includes both qualitative factors, costs or benefits that are subjective in nature, and quantitative factors, costs or benefits for which monetary values can easily be identified. I will discuss the need to take both kinds of factors into account when performing a cost/benefit analysis.

Very often you will need to improve the existing operations, maintenance, and support infrastructure to support the operation of the new application that you intend to develop. To determine what the impact will be you will need to understand both the current operations and support infrastructure of your organization and the operations and support characteristics of your new application. To operate this application END-TO-END VMS. The user no need to require any technical knowledge that we are used to develop this project is Asp.net C# .net. That the application

providing rich user interface by user can do the operation in flexible manner.

## **Software Requirement Specifications**

### **Functional requirement:**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

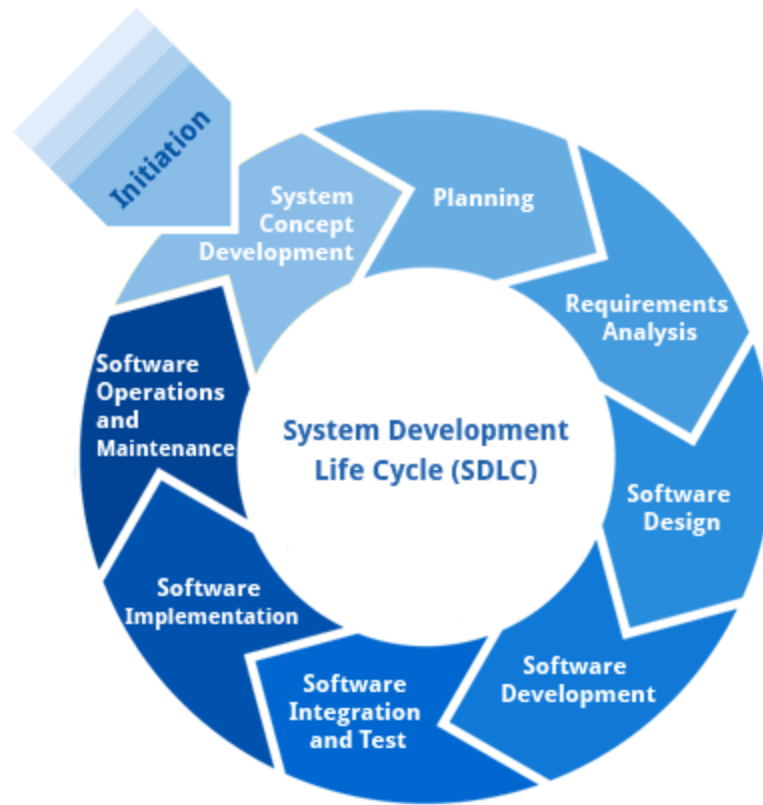
- External Outputs, whose destination is outside the organization,.
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.
- Understanding user's preferences, expertise level and his business requirements through a friendly questionnaire.
- Input data can be in four different forms - Relational DB, text files, .xls and xml files. For testing and demo you can choose data from any domain. User-B can provide business data as input.

### **Non functional requirements:**

- Secure access of confidential data (user's details). SSL can be used.
- 24 X 7 availability.
- Better component design to get better performance at peak time
- Flexible service based architecture will be highly desirable for future extension

### **Software Development Life Cycle**

The **Systems Development Life Cycle (SDLC)**, or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies use to develop these systems.



## **Requirement Analysis and Design**

Analysis gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization. Options for changing the business process may be considered. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software

development. Much care is taken during this phase. The logical system of the product is developed in this phase.

### **Implementation**

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, Java, .Net are used for coding. With respect to the type of application, the right programming language is chosen.

### **Testing**

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).

### **Maintenance**



Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

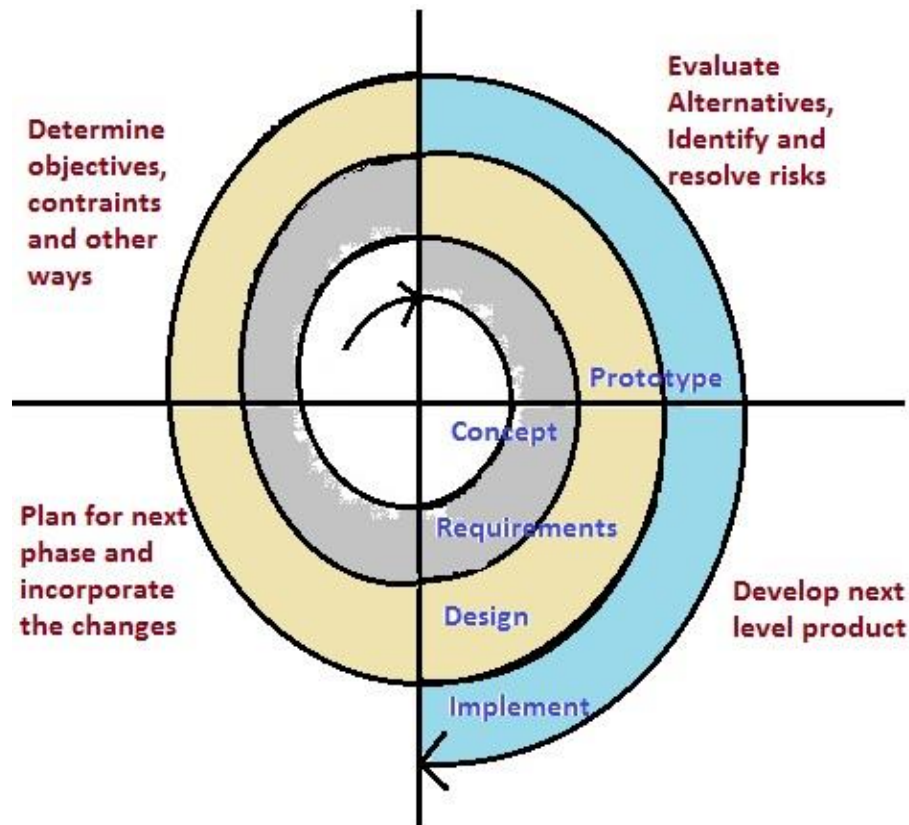
### **SDLC METHDOLOGIES**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

**SPIRAL MODEL** was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

**The following diagram shows how a spiral model acts like:**



The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and

represents an approximation of the characteristics of the final product.

- A second prototype is evolved by a fourfold procedure:
  5. Evaluating the first prototype in terms of its strengths, weakness, and risks.
  6. Defining the requirements of the second prototype.
  7. Planning and designing the second prototype.
  8. Constructing and testing the second prototype.
- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.

- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

## **FUNCTIONAL REQUIREMENTS**

### **OUTPUT DESIGN:**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

### **OUTPUT DEFINITION**

**The outputs should be defined in terms of the following points:**

- Type of the output

- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

## **INPUT DESIGN**

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

## **INPUT STAGES:**

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion

- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

### **INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

### **INPUT MEDIA:**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy

- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

### **ERROR AVOIDANCE**

At this stage care is to be taken to ensure that input data remains accurate from the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

### **ERROR DETECTION**

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

## **DATA VALIDATION**

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

## **USER INTERFACE DESIGN**

It is essential to consult the system users and discuss their needs while designing the user interface:

## **USER INTERFACE SYSTEMS CAN BE BROADLY CLASSIFIED**

**AS:**



1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.

2. Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

## **USER INITIATED INTERFACES**

User initiated interfaces fall into two approximate classes:

1. Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
2. Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms oriented interface is chosen because it is the best choice.

## **COMPUTER-INITIATED INTERFACES**

The following computer – initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives

another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

### **ERROR MESSAGE DESIGN:**

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

### **2) PERFORMANCE REQUIREMENTS**

Performance is measured in terms of the output provided by the application.

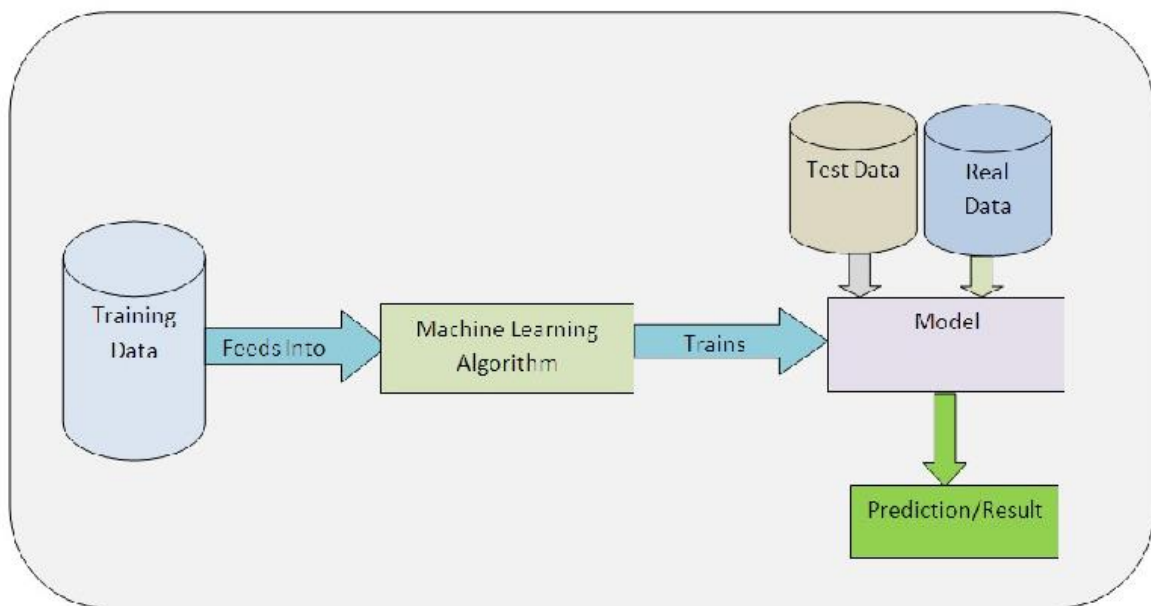
Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been

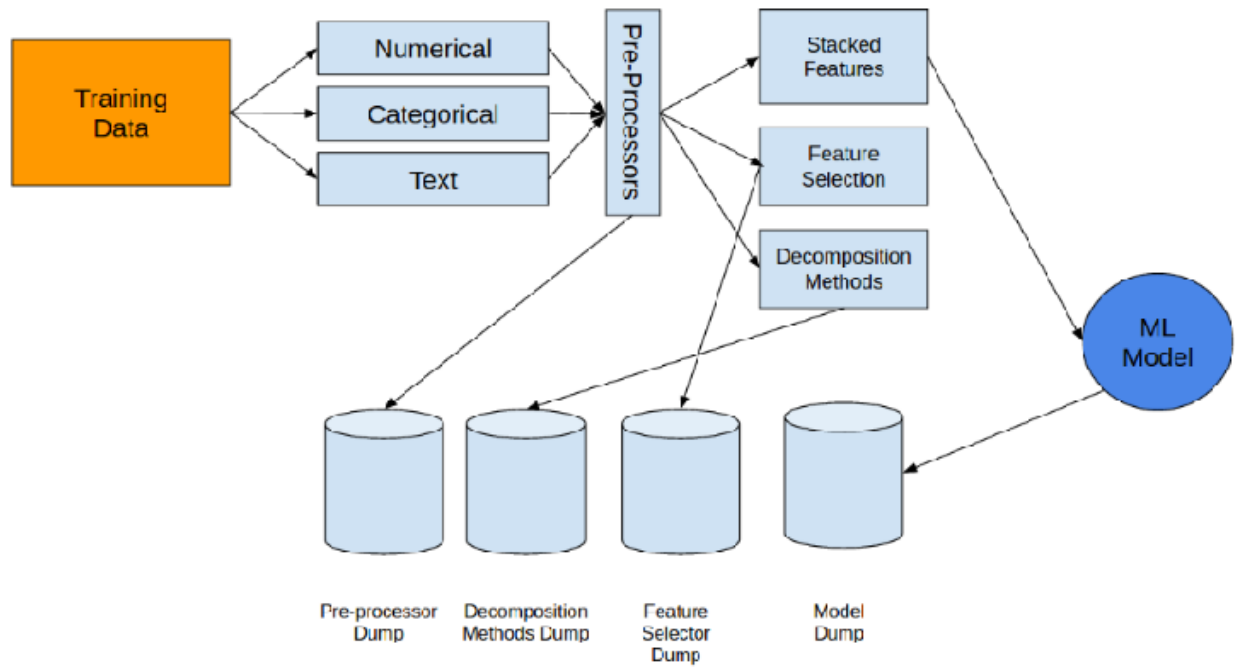
designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

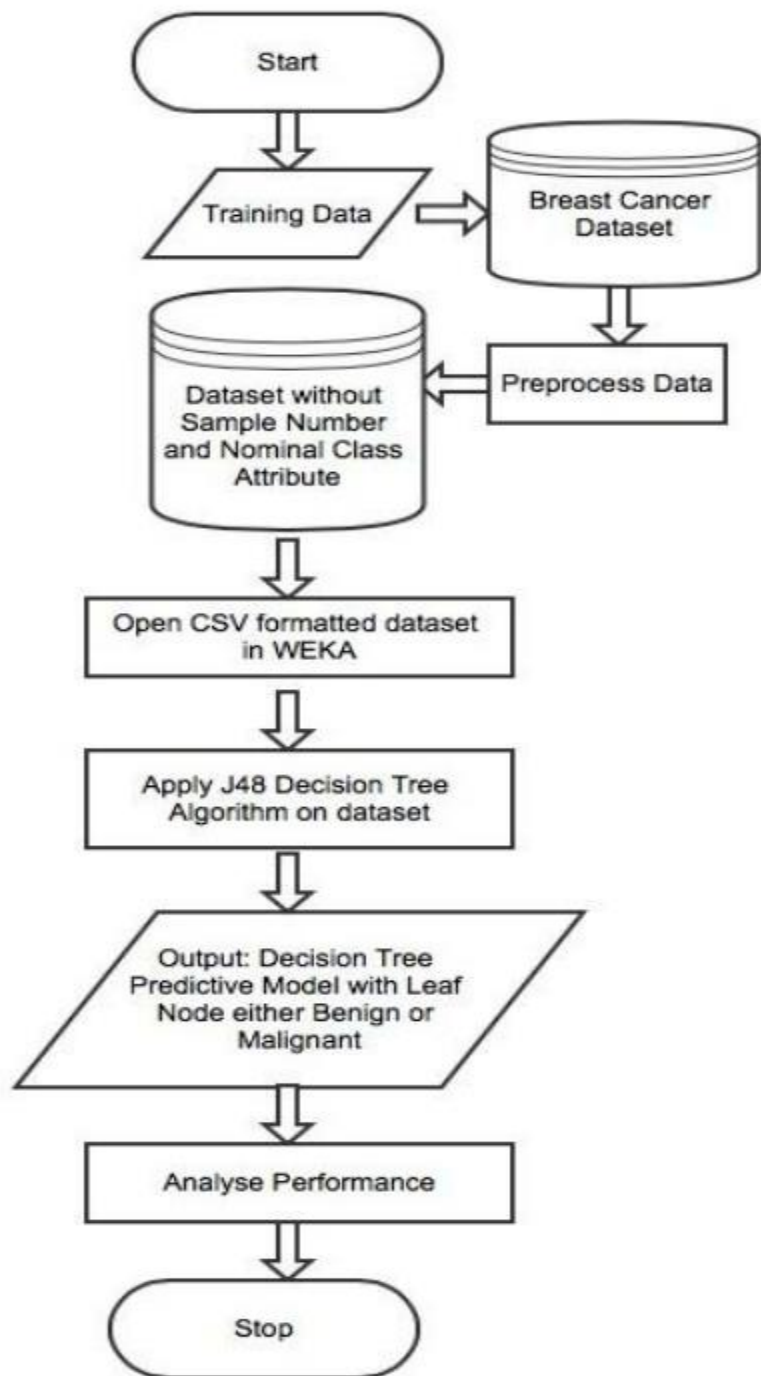
The requirement specification for any system can be broadly stated as given below:

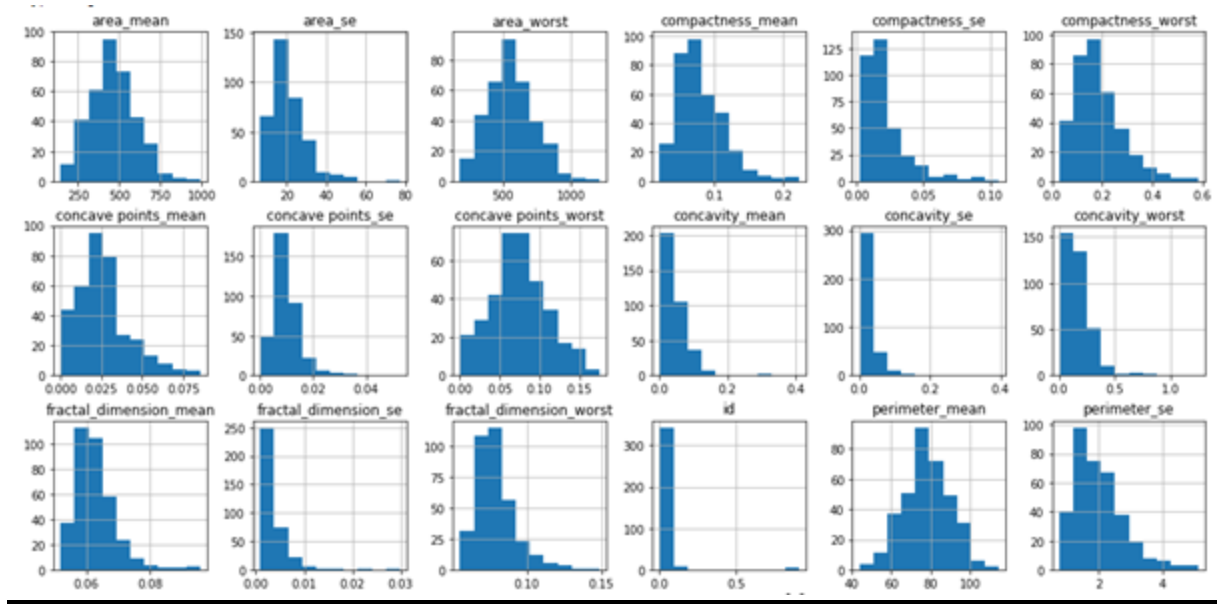
- The system should be able to interface with the existing system

## **UML Diagrams:**

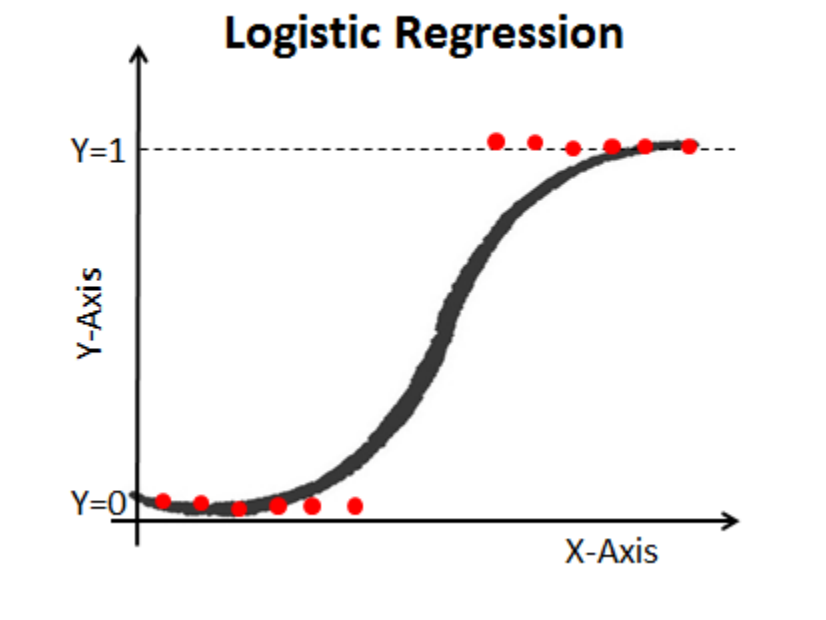




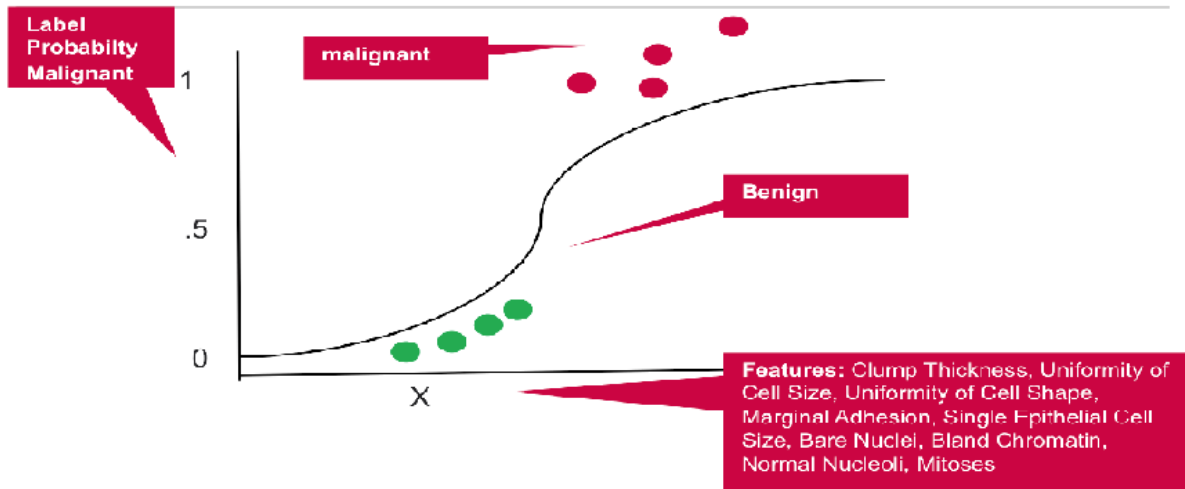




## Class Diagrams:

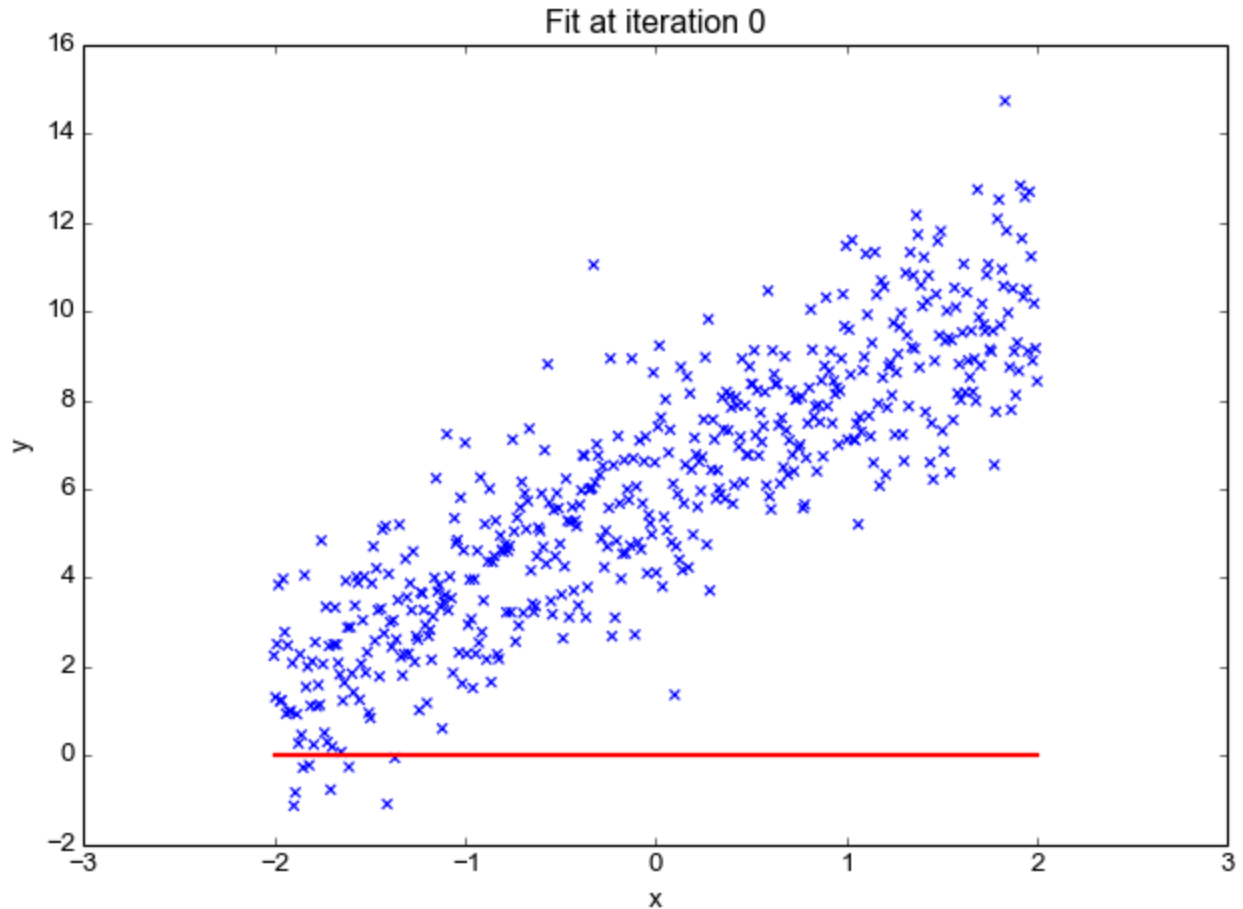


### Breast Cancer Logistic Regression Example



## Test Cases:

The whole point of regression is to find a hyperplane (fancy word for multi-dimensional line) that minimizes the cost function to create the best possible relationship between data points.

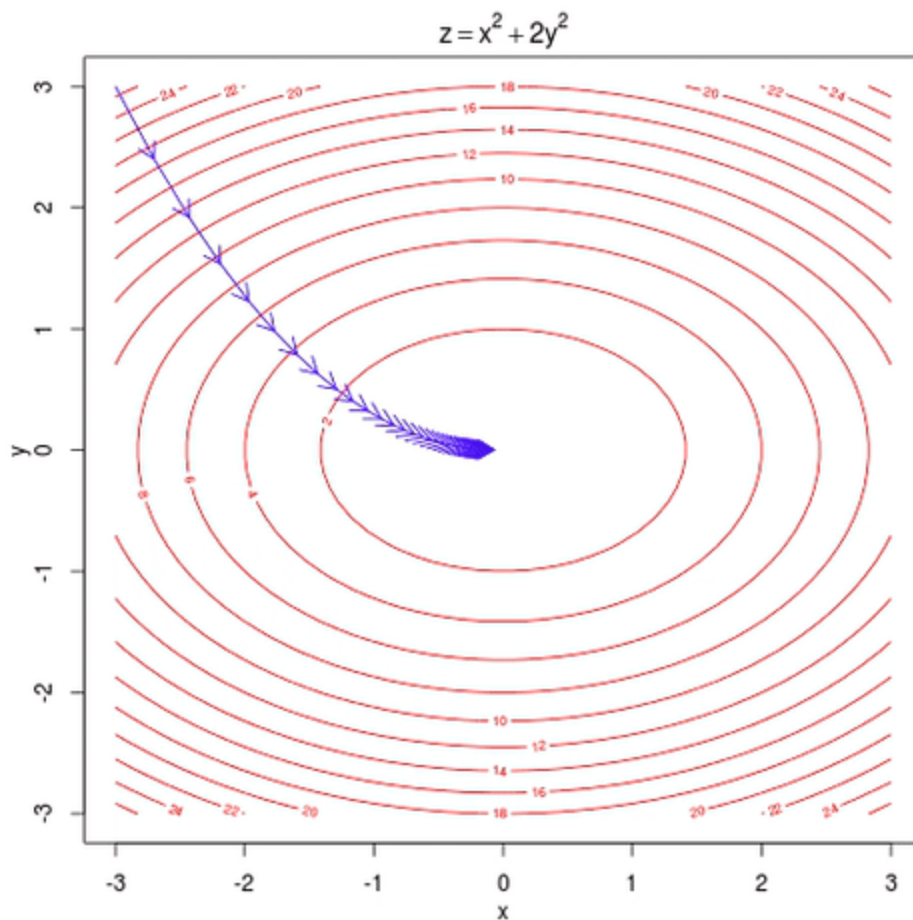


Linear regression making the relationship more accurate

It starts with a random line with no correlation that reiterates using gradient descent to become the optimum relation.

Regression is done using an algorithm called **Gradient Descent**. In this algorithm, the cost function is reduced by the model adjusting its parameters.





Think of descent as you running down a hill, trying to get to the lowest point.

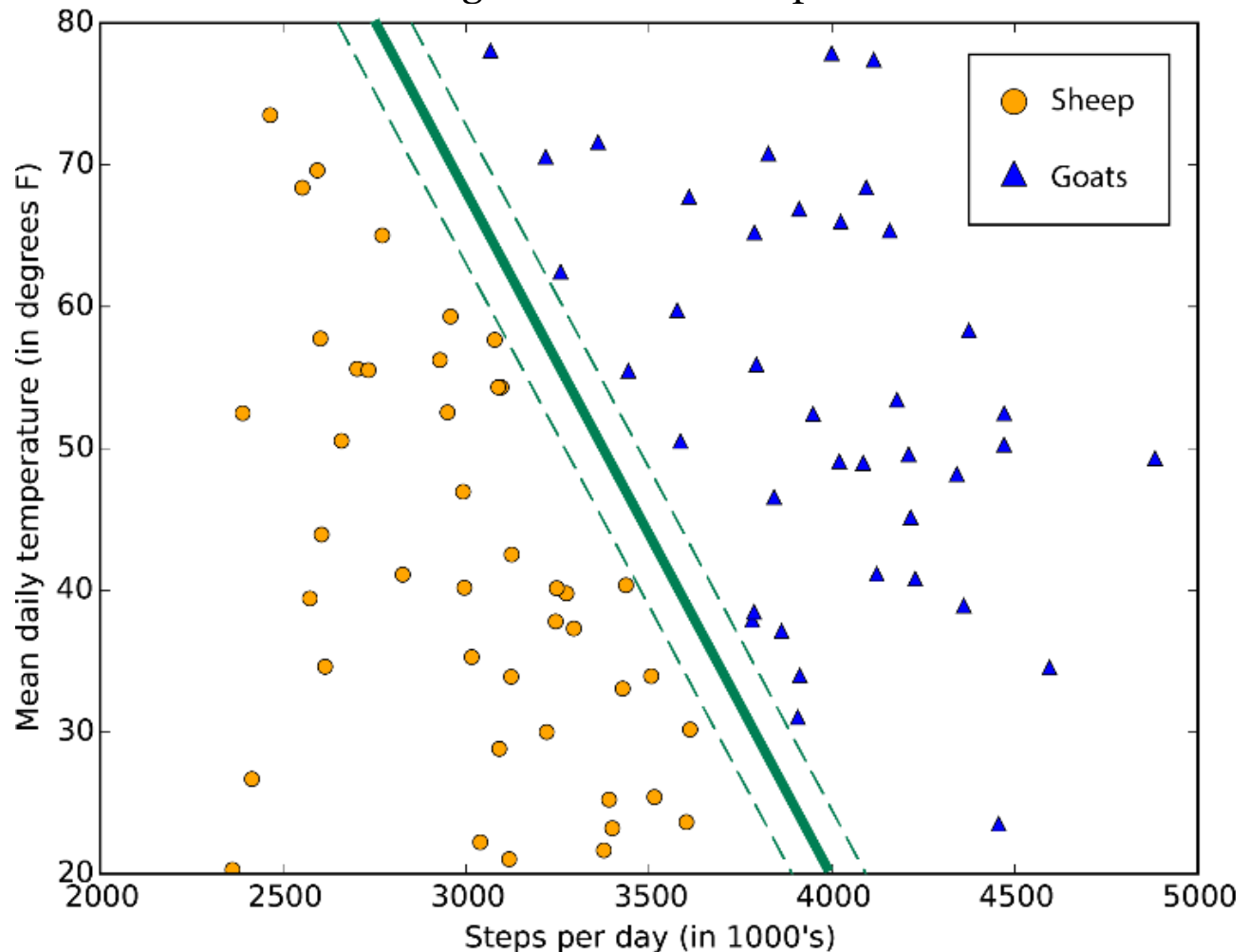
Meanwhile, as gradient descent reduces the cost function lower and lower, the outcome becomes more accurate too.

That's how your model gets more accurate, by using regression to better fit the given data.

# Classification Categorizes Data Points Into Groups

Supervised learning models can do more than just regression. One of ML's most useful tasks is classification.

Classification algorithms **make boundaries between data points** classifying them as a certain group, depending on their characteristics matched against the model's parameters.

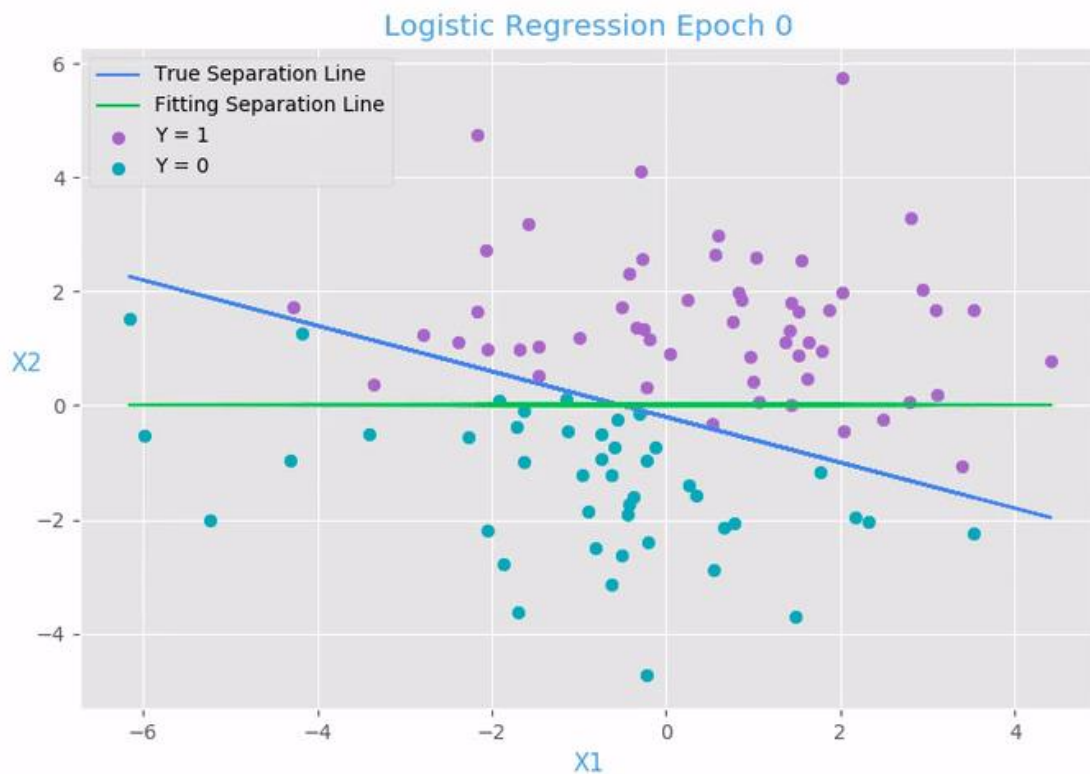


In this model, data points are classified as either being sheep or goat. This is conditional on their steps per day depending on average daily temperature.

The boundary between the classes is created using a process called logistic regression.

An important fact to remember is that **the boundary does not depend on the data.**

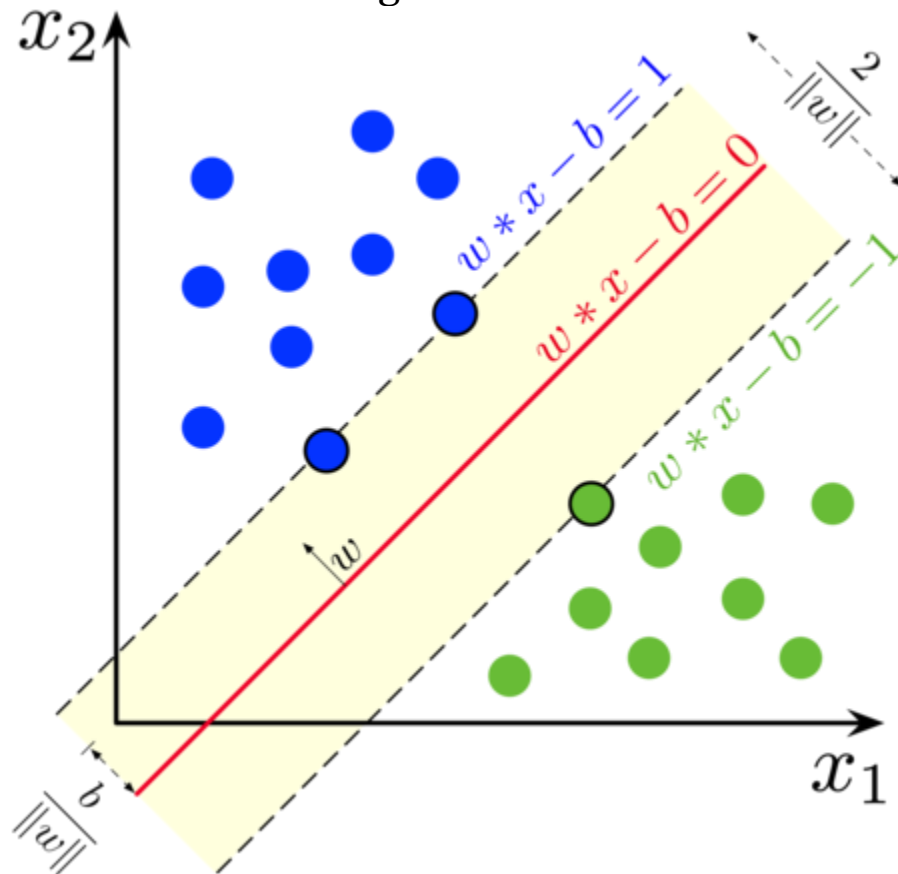
Remember the cost function? Surprise! it's also used in classification.



In classification, it is used similarly to regression to find the best possible fit to the data.

# Supported Vector Machines

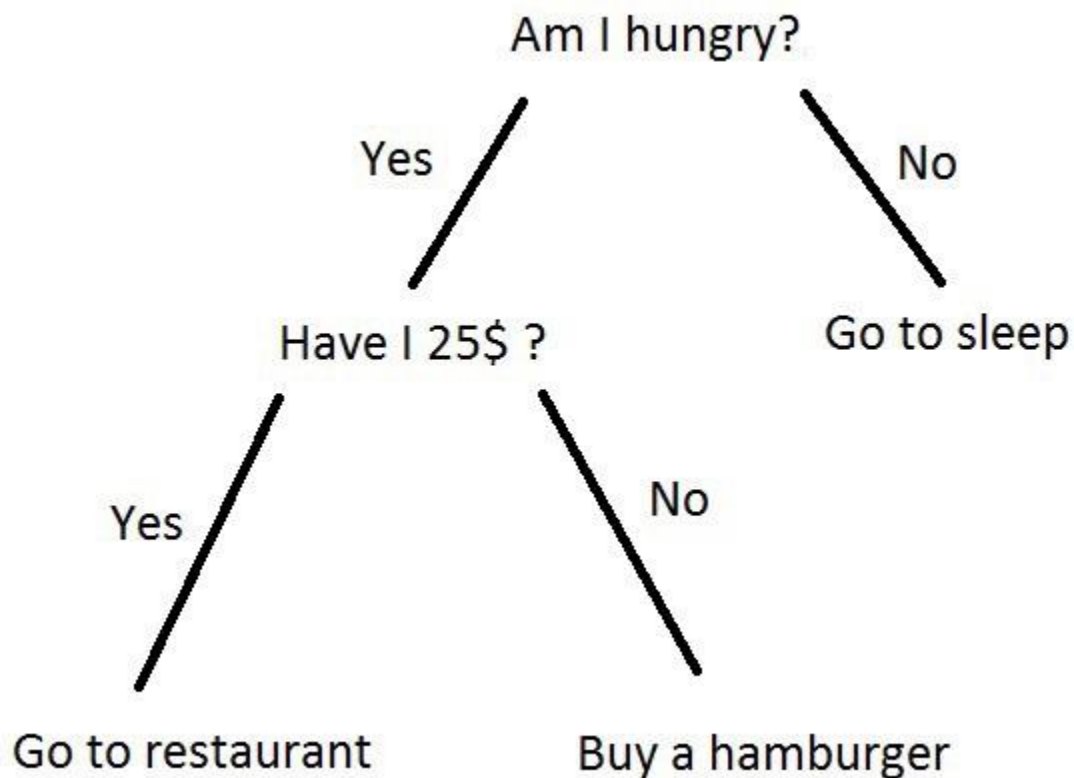
SVM's are supervised learning algorithms used in both classification and regression.



The goal of an SVM algorithm is to classify data by creating a **boundary with the widest possible margin** between itself and the data.

# Decision Trees Narrow Down to an Outcome

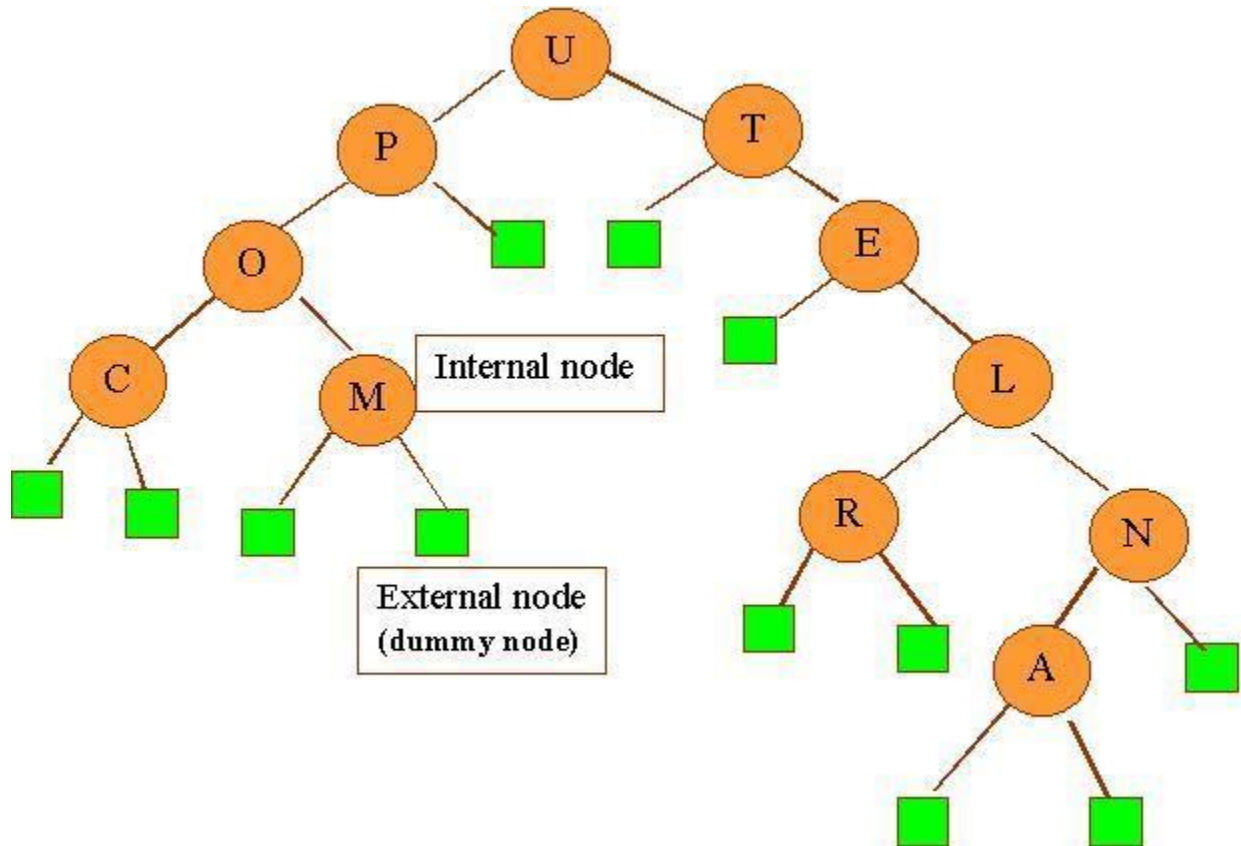
A Decision Tree is a tree-like model (*if trees grew upside down*) representation of probability and decision making in ML.



The process of deciding what you'll be eating

As seen in the figure above, DT's use conditional statements to narrow down on the probability of a certain value taking place for an instance. It uses the DT model to predict the **probability of an instance having a certain outcome.**

DT's keep splitting into further nodes until every input has an outcome.



Basically, **internal nodes split further** while **external nodes are like a stop sign**.

**Screens Shorts:-**

New Tab x Desktop/MLPROJECTS/CANCER DS\_07\_of\_02\_Tree\_based\_model: x WhatsApp x +

localhost:8891/notebooks/Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_02\_Tree\_based\_models\_with\_two\_features%20(1).ipynb

jupyter DS\_07\_of\_02\_Tree\_based\_models\_with\_two\_features (1) Last Checkpoint: 06/01/2019 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

**Step 4: Create DecisionTreeClassifier Model with a maximum depth of 6**

```
In [9]: dt = DecisionTreeClassifier(max_depth=6,
    random_state=SEED,
    criterion='gini')
```

```
In [10]: # Fit dt to the training set
dt.fit(X_train, y_train)
```

```
Out[10]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=6,
    max_features=None, max_leaf_nodes=None,
    min_impurity_decrease=0.0, min_impurity_split=None,
    min_samples_leaf=1, min_samples_split=2,
    min_weight_fraction_leaf=0.0, presort=False, random_state=1,
    splitter='best')
```

**Step 5: Predict test set labels using model**

```
In [11]: y_pred = dt.predict(X_test)
```

**Step 6: Test the Performance**

```
In [12]: # Compute test set accuracy
```

Activate Windows  
Go to Settings to activate Windows.

Type here to search

New Tab x Desktop/MLPROJECTS/CANCER DS\_07\_of\_02\_Tree\_based\_model: x WhatsApp x +

localhost:8891/notebooks/Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_02\_Tree\_based\_models\_with\_two\_features%20(1).ipynb

jupyter DS\_07\_of\_02\_Tree\_based\_models\_with\_two\_features (1) Last Checkpoint: 06/01/2019 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

**LogisticRegression Vs Decision Tree Classification**

```
In [0]: # Import LogisticRegression from sklearn.linear_model
from sklearn.linear_model import LogisticRegression
```

```
In [0]: # Instantiate Logreg
logreg = LogisticRegression(random_state=1)
```

```
In [0]: # Fit logreg to the training set
logreg.fit(X_train, y_train)
```

```
Out[22]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
    intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
    penalty='l2', random_state=1, solver='liblinear', tol=0.0001,
    verbose=0, warm_start=False)
```

```
In [0]: # predict
y_pred1 = logreg.predict(X_test)
```

```
In [0]: acc1 = accuracy_score(y_test, y_pred1)
acc1
```

```
Out[25]: 0.9122807017543859
```

Activate Windows  
Go to Settings to activate Windows.

Type here to search



New Tab x Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_03\_Tree\_based\_modi x DS\_07\_of\_02\_Tree\_based\_modi x WhatsApp x + - x

localhost:8891/notebooks/Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_03\_Tree\_based\_models\_Gini\_vs\_entropy%20(1).ipynb

jupyter DS\_07\_of\_03\_Tree\_based\_models\_Gini\_vs\_entropy (1) Last Checkpoint: 06/01/2019 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

### Step 3: Create Training and Test sets

```
In [0]: X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size = 0.2,
                                                    random_state=SEED,
                                                    stratify=y)

In [0]: X_train.shape # (455, 2)
Out[12]: (455, 30)

In [0]: y_train.shape # (455,)
Out[13]: (455,)

In [0]: X_test.shape # (114, 2)
Out[14]: (114, 30)

In [0]: y_test.shape # (114,)
Out[15]: (114,)
```

### Step 4: Create Model using criterion as entropy

Activate Windows  
Go to Settings to activate Windows.

Type here to search

New Tab x Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_03\_Tree\_based\_modi x DS\_07\_of\_02\_Tree\_based\_modi x WhatsApp x + - x

localhost:8891/notebooks/Desktop/MLPROJECTS/CANCER%20PREDICTION/DS\_07\_of\_03\_Tree\_based\_models\_Gini\_vs\_entropy%20(1).ipynb

jupyter DS\_07\_of\_03\_Tree\_based\_models\_Gini\_vs\_entropy (1) Last Checkpoint: 06/01/2019 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

### Step 4: Create Model using criterion as entropy

```
In [0]: # Create dt_entropy model, set 'entropy' as the information criterion
dt_entropy = DecisionTreeClassifier(max_depth=8,
                                   criterion='entropy',
                                   random_state=SEED)

In [0]: # Fit dt_entropy to the training set
dt_entropy.fit(X_train, y_train)

Out[17]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=8,
                               max_features=None, max_leaf_nodes=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, presort=False, random_state=1,
                               splitter='best')

In [0]: # Use dt_entropy to predict test set labels
y_pred = dt_entropy.predict(X_test)

In [0]: # Evaluate accuracy_entropy
accuracy_entropy = accuracy_score(y_test, y_pred)

In [0]: accuracy_entropy
Out[20]: 0.9298245614035088
```

Activate Windows  
Go to Settings to activate Windows.

Type here to search



Step 5: Create Model using criterion as gini

```
In [0]: # Instantiate dt_gini, set 'gini' as the information criterion
dt_gini = DecisionTreeClassifier(max_depth=8,
                                criterion='gini',
                                random_state=SEED)

In [0]: # Fit dt_entropy to the training set
dt_gini.fit(X_train, y_train)

Out[22]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=8,
                                max_features=None, max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, presort=False, random_state=1,
                                splitter='best')

In [0]: # Use dt_entropy to predict test set labels
y_pred_gini = dt_gini.predict(X_test)

In [0]: # Evaluate accuracy_gini
accuracy_gini = accuracy_score(y_test, y_pred_gini)

In [0]: accuracy_gini

Out[25]: 0.9298245614035088
```

Activate Windows  
Go to Settings to activate Windows.

Step 6: compare entropy and gini accuracy

```
In [0]: # Print accuracy_entropy
print('Accuracy achieved by using entropy: ', accuracy_entropy)
# Print accuracy_gini
print('Accuracy achieved by using the gini index: ', accuracy_gini)

Accuracy achieved by using entropy: 0.9298245614035088
Accuracy achieved by using the gini index: 0.9298245614035088
```

**Note:** Notice how the two models achieve exactly the same accuracy. Most of the time, the gini index and entropy lead to the same results. The gini index is **slightly** faster to compute and is the default criterion used in the DecisionTreeClassifier model of scikit-learn.

```
In [0]:
```

Activate Windows  
Go to Settings to activate Windows.

The screenshot shows a Jupyter Notebook running in a web browser. The notebook is titled "DS\_07\_of\_02\_Tree\_based\_models\_with\_two\_features (1)". The output of a cell shows a table with two columns: "radius\_mean" and "concave points\_mean". The input for the next cell is "y[15:25]". The output of the next cell shows a list of predictions: 15 M, 16 M, 17 M, 18 M, 19 B, 20 B, 21 R.

	radius_mean	concave points_mean
15	14.540	0.07364
16	14.680	0.05259
17	16.130	0.10280
18	19.810	0.09498
19	13.540	0.04781
20	13.080	0.03110
21	9.504	0.02076
22	15.340	0.09756
23	21.160	0.08632
24	16.650	0.09170

```
In [6]: y[15:25]
```

```
Out[6]: 15 M  
16 M  
17 M  
18 M  
19 B  
20 B  
21 R
```

## Conclusion:

Firstly, machines can work much faster than humans. A biopsy usually takes a Pathologist 10 days. A computer can do thousands of biopsies in a matter of seconds.

Machines can do something which humans aren't that good at. They can **repeat** themselves thousands of times without getting exhausted. **After every iteration, the machine repeats the process to do it better.** Humans do it too, we call it practice. While practice may make perfect, no amount of practice can put a human even close to the computational speed of a computer.

Another advantage is the great accuracy of machines. With the advent of the Internet of Things technology, there is so much data

out in the world that humans can't possibly go through it all. That's where machines help us. They can do work faster than us and make accurate computations and find patterns in data. That's why they're called computers.

Every year, Pathologists diagnose 14 million new patients with cancer around the world. That's millions of people who'll face years of uncertainty.

Pathologists have been performing cancer diagnoses and prognoses for decades. Most pathologists have a 96–98% success rate for diagnosing cancer. They're pretty good at that part.

The problem comes in the next part. According to the Oslo University Hospital, the accuracy of prognoses is only 60% for pathologists. A prognosis is the part of a biopsy that comes after cancer has been diagnosed, it is predicting the development of the disease.

It's time for the next step to be taken in pathology.