**CHAPTER 3 :**

**MINI PROJECT TITLE : DISEASE PREDICTION**

**INTRODUCTION**

Given how important the health sector is in curing prescribers\' problems, Disease Prediction based on Symptoms with Machine Learning is a system that predicts diseases based on the user\'s knowledge of clinical manifestations, assuring solid findings based on such facts. If the user simply has to know a little bit about the sickness and the patient isn\'t in any danger, this technique can be used to learn a little bit about minor ailments. It\'s a system that provides medical advice and tactics to clients, as well as a tool to help them figure out what ailment they have utilizing this forecast. It\'s also a big benefit for the healthcare sector as well as individuals who don\'t want to travel to a hospital or clinic for their initial diagnosis. The user can learn a lot about the condition that has been revealed to him or her by simply inputting the side effects and other critical information, and the health sector can benefit from this method by simply asking the patient for symptoms and giving them a diagnosis. To achieve Disease Prediction based on Symptoms, we used Machine Learning techniques, Python Programming with Tkinter Interface, and a dataset acquired from hospitals. The phrases Disease Predictor, Machine Learning, and Tkinter Interface are used in this research.

A well-functioning healthcare system is critical to the economy and the well-being of humanity. Between the world, we live in now and the world we lived in a few decades ago, there has been a substantial amount of change. Everything has gotten more disorderly and unattractive. In this situation, doctors and nurses are doing everything they can to save people's lives, even if it means putting their own lives in danger. Virtual doctors are board-certified doctors who choose to practice online using video and phone consultations rather than in-person consultations, albeit this is not always practicable in an emergency. In the absence of human error, machines are thought to be superior to humans because they can do jobs faster while maintaining a consistent level of precision. A disease predictor, often known as a virtual doctor, may accurately predict a patient's sickness without the need for human involvement. A disease predictor can save a person's life in extreme instances, such as COVID-19 and EBOLA, by recognizing their health without requiring physical touch. There are virtual doctors on the market now, but they lack the capacity to provide the kind of precision that is required. This Condition's Prognosis To forecast sickness, we'll use hospital data and Machine Learning methods based on the Python programming language and the Tkinter interface. Doctors may make errors when diagnosing a patient's disease, however, disease prediction systems with machine learning algorithms can help produce accurate results in these situations. For this project, we employed a mix of approaches, algorithms, and technologies to develop a system that can forecast a patient's status based on their symptoms. The symptoms are compared to the information previously saved in the system. We can accurately forecast the percentage of disease in a patient by combining those datasets with the patient's symptoms. The dataset and symptoms are uploaded to the system's prediction model, where the data is pre-processed for future references before the user picks the features and enters the symptoms. The data is then classified using a variety of algorithms and approaches, such as Decision Tree, KNN, and Naive Bayes, to mention a few.

Machine learning is programming icomputers to optimize a performance using example data or past data. Machine learning is study of computer i systems that learn from data and experience .Machine learning i algorithm has two passes: Training, Testing .Prediction of a disease by using patient’s symptoms and history machine learning technology is struggling from past decades. Machine Learning technology gives a good platform in medical field, so that a healthcare issues can be solved efficiently. in medical i field. We are applying machine learning to maintained complete hospitall data Machinei learningg technology which allows building modelss to get quickly analyzee dataa and deliver resultss faster, with the use of machine learning technology doctors can make good i decision for patient diagnosess and treatment options, which leadss to improvementt of patient healthcare services. Healthcare is the most prime examplee of how machine learning is use There is a need to study and make a system which will make it easy for an end users to predict the chronic diseases without visiting physician or doctor for diagnosis. To detect the Various Diseases through the examining Symptoms of patient’s using different techniques of Machine Learning Models. To Handle Text data and Structured data is no Proper method. The Proposed system will consider both structure and unstructured data. The Predictions Accuracy will Increase using Machine Learning.

The system predicts the chronic diseases which is for particular region and for the particular community. The Prediction of Diseases is done only for particular diseases. In this System Big Data & CNN Algorithm is used for Diseases risk prediction. For S type data, system is using Machine Learning algorithm i.e K-nearest Neighbors, Decision Tree, Naïve Bayesian. The accuracy of the System is upto 94.8%. Existing paper, we streamline machine learning algorithms for effective prediction of chronic disease outbreak in disease-frequent communities. We experiment the modified prediction models over real life hospital data collected from central China.

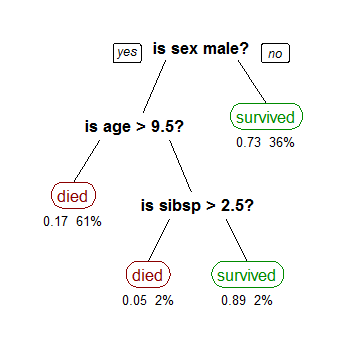
**METHODOLOGY**

# decision trees in machine learning

a tree has many analogies in real life, and turns out that it has influenced a wide area of **machine learning**, covering both **classification and regression**. in decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. as the name goes, it uses a tree-like model of decisions. though a commonly used tool in data mining for deriving a strategy to reach a particular goal, its also widely used in machine learning, which will be the main focus of this article.

## how can an algorithm be represented as a tree?

for this let’s consider a very basic example that uses titanic data set for predicting whether a passenger will survive or not. below model uses 3 features/attributes/columns from the data set, namely sex, age and sibsp (number of spouses or children along).



a decision tree is drawn upside down with its root at the top. in the image on the left, the bold text in black represents a condition/**internal node**, based on which the tree splits into branches/ **edges**. the end of the branch that doesn’t split anymore is the decision/**leaf**, in this case, whether the passenger died or survived, represented as red and green text respectively.

although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can’t ignore the simplicity of this algorithm. the **feature importance is clear** and relations can be viewed easily. this methodology is more commonly known as **learning decision tree from data** and above tree is called **classification tree** as the target is to classify passenger as survived or died. **regression trees** are represented in the same manner, just they predict continuous values like price of a house. in general, decision tree algorithms are referred to as cart or classification and regression trees.

**so, what is actually going on in the background?** growing a tree involves deciding on **which features to choose** and **what conditions to use** for splitting, along with knowing when to stop. as a tree generally grows arbitrarily, **you will need to trim it down** for it to look beautiful. lets start with a common technique used for splitting.

## **recursive binary splitting**



in this procedure all the features are considered and different split points are tried and tested using a cost function. the split with the best cost (or lowest cost) is selected.

consider the earlier example of tree learned from titanic dataset. in the first split or the root, all attributes/features are considered and the training data is divided into groups based on this split. we have 3 features, so will have 3 candidate splits. now we will calculate how much [accuracy](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db)each split will cost us, using a function. the split that costs least is chosen, which in our example is sex of the passenger. this algorithm is recursive in nature as the groups formed can be sub-divided using same strategy. due to this procedure, this algorithm is also known as the **greedy algorithm**, as we have an excessive desire of lowering the cost. **this makes the root node as best predictor/classifier.**

## cost of a split

lets take a closer look at **cost functions used for classification and regression**. in both cases the cost functions try to **find most homogeneous branches, or branches having groups with similar responses**. this makes sense we can be more sure that a test data input will follow a certain path.

regression : sum(y — prediction)²

lets say, we are predicting the price of houses. now the decision tree will start splitting by considering each feature in training data. the mean of responses of the training data inputs of particular group is considered as prediction for that group. the above function is applied to all data points and cost is calculated for all candidate splits. again the split with lowest cost is chosen. another cost function involves reduction of standard deviation, more about it can be found [here](http://www.saedsayad.com/decision_tree_reg.htm).

classification : g = sum(pk \* (1 — pk))

a gini score gives an idea of how good a split is by how mixed the response classes are in the groups created by the split. here, pk is proportion of same class inputs present in a particular group. a perfect class purity occurs when a group contains all inputs from the same class, in which case pk is either 1 or 0 and g = 0, where as a node having a 50–50 split of classes in a group has the worst purity, so for a binary classification it will have pk = 0.5 and g = 0.5.

## when to stop splitting?

you might ask when to stop growing a tree? as a problem usually has a large set of features, it results in large number of split, which in turn gives a huge tree. such trees are complex and can lead to overfitting. so, we need to know when to stop? one way of doing this is to **set a minimum number of training inputs to use on each leaf.** for example we can use a minimum of 10 passengers to reach a decision(died or survived), and ignore any leaf that takes less than 10 passengers. another way is to set **maximum depth** of your model. **maximum depth refers to the the length of the longest path from a root to a leaf.**

## pruning

the performance of a tree can be further increased by pruning. it involves removing the branches that make use of features having low importance. this way, we reduce the complexity of tree, and thus increasing its predictive power by reducing overfitting.

pruning can start at either root or the leaves. the simplest method of pruning starts at leaves and removes each node with most popular class in that leaf, this change is kept if it doesn't deteriorate [accuracy](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db). its also called **reduced error pruning**. more sophisticated pruning methods can be used such as **cost complexity pruning** where a learning parameter (alpha) is used to weigh whether nodes can be removed based on the size of the sub-tree. this is also known as **weakest link pruning.**

## advantages of cart

* simple to understand, interpret, visualize.
* decision trees implicitly perform variable screening or feature selection.
* can handle both numerical and categorical data. can also handle multi-output problems.
* decision trees require relatively little effort from users for data preparation.
* nonlinear relationships between parameters do not affect tree performance.

## **disadvantages of cart**

* decision-tree learners can create over-complex trees that do not generalize the data well. this is called overfitting.
* decision trees can be unstable because small variations in the data might result in a completely different tree being generated. this is called [variance](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db), which needs to be lowered by methods like bagging and [boosting](https://towardsdatascience.com/boosting-the-accuracy-of-your-machine-learning-models-f878d6a2d185).
* greedy algorithms cannot guarantee to return the globally optimal decision tree. this can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
* decision tree learners create [biased](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db) trees if some classes dominate. it is therefore recommended to balance the data set prior to fitting with the decision tree.

**CODE:**

**CHAPTER:4**

**RESULTS:**

**CONCLUSION:**

The Prediction Engine that allows the user to examine whether or not he/she has any unwellness or disorder supported the given symptoms. The user interacts with the Prediction Engine by filling a collection of symptoms that holds the parameter set provided as associate input to the trained models. The Prediction Engine makes use of 3 algorithms to predict the presence of a unwellness namely: call Tree, Random Forest and Naive Bayes. The reason to settle on these 3 algorithms are: 1) They effective, if the coaching information is massive. 2) A single dataset is provided as associate input to any or all these three algorithms with bottom or no modification. 3) A common scalar is accustomed normalize the input provided to those three algorithms.

**REFERENCES**

[1] Kaveeshwar, S.A., and Cornwall, J., 2014, “The current state of unwellness mellitus in India”. AMJ, 7(1), pp. 45-48 [2] Dean, L., McEntyre, J., 2004, “The Genetic Landscape of unwellness [Internet]. Bethesda (MD): National Center for Biotechnology info (US); Chapter one, Introduction to unwellness. 2004 Jul 7. [3] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan and A. Alamri, \"HealthCPS: aid cyberphysical system power-assisted by cloud and massive data\", IEEE Syst. J, vol. 11, no. 1, pp. 88-95, Mar. 2017. [4] Allen Daniel Sunny, Sajal Kulshreshtha, Satyam Singh, Srinabh, Mohan Ba and H Sarojadevi, \"Disease identification System By Exploring Machine Learning Algorithms\", International Journal of Innovations in Engineering and Technology (IJIET), vol. 10, no. 2, May