An Investigative Approach To Employ Different Machine Learning Algorithms in Detecting Brain Cancer

by

Md. Akib Mohtasim (170021016)

Zahidul Islam (170021049)

Ashraful Alam (170021096)

A Dissertation Submitted to the Academic Faculty for Partial Completion of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

May 2022

CERTIFICATE OF APPROVAL

The thesis titled “Performance Investigation of Different Machine Learning Algorithms in Predicting Brain Cancer” submitted by Md. Akib Mohtasim (170021016), Zahidul Islam (170021049), and Ashraful Alam (170021096) has been determined to be satisfactory and approved as partial fulfillment of the Bachelor of Science in Electrical and Electronic Engineering degree requirement on May, 2022

Approved by:

[illegible]

(Signature of the Supervisor)
Mirza Muntasir Nishat
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology

Signature of the Authors:

[illegible]	[illegible]	[illegible]
Md. Akib Mohtasim	**Zahidul Islam**	**Ashraful Alam**
ID: 170021016	**ID: 1700211049**	**ID: 170021096**

Table of Contents

Dedication

This project is dedicated to our friends, families, well-wishers, and, most importantly, our outstanding professors. They serve as a source of motivation and inspiration which invigorate us in every step of our lives. We wouldn't be where we are without them and they are the reason we start every day with profound energy and drive.

Acknowledgements

First and foremost, we express our heartfelt gratitude to The Almighty Allah for providing us with the patience and strength to complete this thesis work smoothly. We are extremely grateful to our honorable supervisor, ***Mr. Mirza Muntasir Nishat***, Assistant Professor, Department of EEE, IUT for giving us the opportunity to work under him. He has shown us the appropriate way to do a research project and recommended us to gather a solid foundation for our work. He has always motivated us to create new and inventive works. We would have lost track and became bewildered without his help and advice from the start.

We would also like to thank our friend and express our gratefulness to ***Tasnimul Hasan,*** our batchmate at the Department of EEE, IUT, for sincerely giving us his time and effort whenever it was necessary. He helped us with our simulations and produced the results. Without him, we could not have conducted our work and we are forever indebted to him for his contribution to this project.

Abstract

This thesis takes an exploratory method to investigate the performance of several machine learning algorithms in more accurately detecting brain tumors. It primarily accomplishes this by detecting both malignant and benign lesions in the brain. In recent years, brain cancer has been linked to the highest newborn cancer fatality rates worldwide. Various machine learning techniques have been shown to be an excellent tool for detecting brain while it is still in its early stages. To train and test the model classifier, a dataset containing 700 instances and 1500 features from the Kaggle Data repository was used. Thirteen Eleven machine learning algorithms were studied and their performance parameters like confusion matrix and accuracy were analyzed. Furthermore, a thorough comparison was carried out through the compuration of precision, sensitivity, F1 score, recall, specificity, cross validation score and error rate of each algorithm.

CHAPTER 1

INTRODUCTION

1.1 Brain Cancer: A Brief Study

A brain tumor is a brain development of aberrant cells [1]. The human body is composed of numerous types of cells. Each cell serves a specific function. These cells in the body divide and expand in a regular pattern, resulting in the formation of new cells. These cells contribute to the health and efficient operation of the human body. When precise cells lose their ability to regulate their own growth, they expand in an unorganized fashion. These cells formed form a mass of tissue which is called tumor. A brain tumor is a collection of abnormal cells in the brain. Tumors can be benign or malignant. Malignant tumors cause cancer, whereas benign tumors do not. A brain tumor can develop in the brain or elsewhere in the central nervous system (CNS), such as the spine or the cranial nerves. The brain controls the majority of physical processes, including conscience, movement, sensory experiences, thinking, discourse, and memory. A tumor can impair the brain's capacity to operate correctly.

There are roughly 120 different types of brain tumors, cysts and pustules, that are differentiate by where they emerge as well as the cells they contain. A few of the tumors listed below, known as "Skullbase tumors," can grow from bone or even other tissues outside of the brain. However, due to their proximity to the brain, they seem to be more likely to have an impact on brain, which is why they are included in this list.

i) Typically Benign Tumors: Meningioma, Pituitary Adenoma, Craniopharyngioma, Schwannoma, Nasopharyngeal Angiofibroma, Choroid Plexus Tumor, Dysembryoplastic Neuroepithelial Tumor, Neurofibroma, Hemangioblastoma, Chondroma, Giant Cell Tumor, Osteoma.

ii) Other Benign Brain Lesions and Cysts: Arachnoid Cyst, Colloid Cyst, Dermoid or Epidermoid Cyst, Encephalocele, Fibrous Dysplasia, Rathke's Cleft Cyst, Petrous Apex Lesion,

iii) Brain Tumors with Variable Grades (From More Benign to Malignant): Glioma, Ependymal Tumors, Hemangiopericytoma, Germ Cell Tumors, Pineal Tumors,

iv) Brain Cancer Types: Typically Malignant Brain Tumors-Chordoma, Chondrosarcoma, Medulloblastoma, Olfactory Neuroblastoma, Lymphoma, Gliosarcoma, Rhabdomyosarcoma, Paranasal Sinus Cancer, Atypical Teratoid/Rhabdoid Tumor (AT/RT) [2].

Brain tumor symptoms might be generic or specialized. Brain cancer, like other diseases, does not exhibit many symptoms in the early stages. It takes a long time for its symptoms and aftereffects to manifest. However, by the time symptoms appear, it is too late. The main reason is that it does not have many serious symptoms at first. A common symptom is tumor pressure on the brain or spinal cord. Various symptoms appear when a specific area of the brain is not functioning properly due to the tumor Many persons with brain tumors were detected after visiting their doctor because of an ailment, such as a headache or other abnormalities. General symptoms include: Headaches, Myoclonic, muscle spasms, Tonic-Clonic, Sensory (Change in sensation), Complex partial, Personality or memory changes, Nausea Fatigue, tiredness, difficulty sleeping, impaired memory, changes in walking ability or perform daily activities.

The following symptoms may be specific to the location of the tumor :

- Near the tumor, patients may experience pressure or headaches.
- A cerebellar tumor is linked to dizziness and difficulty with manual dexterity.
- A tumor in the cerebrum's frontal lobe has been associated with alterations in judgment, including loss of initiative, lethargy, and muscle numbness or tingling.
- A tumor in the occipital lobe or temporal lobe of the cerebrum causes partial or full vision loss.
- A tumor in the frontal and temporal lobes of the cerebrum can cause changes in speech, hearing, memory, or emotional state, including aggression and trouble interpreting or recovering words.
- A tumor in the lateral or periosteum lobe of the cerebral hemispheres is linked to altered touch or tension perception, arm or leg vulnerability solely on a single side of the body, or uncertainty between both the left and right sides of the body.
- A pineal gland tumor might make it difficult to look upward.
- A pituitary brain tumor is associated to lactation, which is the release of breast milk, as well as irregular menstruation and adult expansion in the feet and hands.

.

- A tumor in the brain stem may make it difficult swallowing, trigeminal neuralgia or nerve damage, or double sight.
- A tumor in the temporal lobe, occipital lobe, or brain stem can cause vision alterations such as loss of vision or double vision [3].

At present 19.3 million people are affected by brain cancer worldwide and men are more susceptible to it compared to women [4].

Human inspection is the conventional method for the detection of tumors in magnetic resonance images of the brain. An MRI uses magnetic fields instead of x-rays to create detailed images. An MRI can also be used to ascertain the size of the tumor. Prior to the scan, a special pigment referred as a contrast medium is performed to produce a clearer image. The above pigment can be implanted into a patient's artery or ingested orally in the form of pills or liquid. The "diffusion weighted imaging" MRI procedure aids in disclosing the cell membrane brain structure.

Another technique known as "perfusion imaging" determines how much blood reaches the tumor and these techniques may aid doctors in predicting how well a treatment will work.[5] As The human observation in predicting the tumor may be erroneous due to noise and distortions in the images this method is inefficient for large amounts of data and takes a long time [5].

As a result, in order to preserve radiologist time, computerized tumor detection techniques are developed. MRI encephalon tumor detection is especially intricate and variety of tumors. To identify tumors in encephalon MRIs, machine learning algorithms are utilized.

1.2 What is Machine Learning?

The term "machine learning" was coined by Arthur Samuel, an IBM computer scientist and pioneer in AI and computer gaming additionally Samuel engendered a computer program that plays checkers [44]. As a discipline, machine learning aims to investigate the design and evaluation of algorithms that can learn from and prognosticates the outcome variable. ML has proved invaluable because it can solve problems at a rate and scale that the human mind cannot match also machines can be trained to recognize patterns in and out of data by directing massive amounts of computational power toward a single task or a set of tasks [6].

Machine learning has become one of the most effective and important tools in the engineering field as artificial intelligence has grown and Machine learning is a type of artificial intelligence that has been applied also it enables computerized systems to learn and grow from prior experience without being explicitly programmed [6].

Machine learning enables computers to learn on their own without the need for human intervention. It also gives computers the ability to take actions based on previous experience. To provide the previous experience, a model comprised of massive datasets is required. These datasets contain information about a specific event that the model can use to train itself. These datasets are a machine learning model's most valuable resource. Based on the data, the model performs all of the steps necessary to provide the best possible output to the user. Depending on the type of the model there are mainly three types of machine learning approaches:

i) Supervised Machine Learning.

ii) Unsupervised Machine Learning.

iii) Reinforcement Machine Learning.

1.3 Machine Learning in Healthcare

Machine Learning is now used in almost every aspect of our lives. It is used in everything from checking daily email to launching a rocket into space. The healthcare sector is not the only one where machine learning is being used. It will eventually become one of the most widely used areas of Machine Learning. Pre-Screening through different algorithms will aid the healthcare professionals for fast, prompt and accurate analysis.

Machine Learning for Health Technology uses self-learning neural networks in algorithms to improve treatment quality by analyzing data sources on a patient's condition, X-rays, CT scans, and various tests and preventive care [7]. It's also worth noting that deep learning is increasingly being used to detect cancer cells. This same model is shown a plethora of cancer cell images in order to "memorize" their appearance, and Machine Learning are among the most common sub - sets of Artificial Intelligence. Its goal is to use records to "train" models.. According to a Deloitte survey of 1,100 US companies using Artificial Intelligence, 63 percent were concentrating on Machine Learning.[7]

Aside from the widely used Chatbots, we should pay special attention to the application of Machine Learning in Medical metaheuristics in:

i) Oncology
ii) Pathology
iii) Rare diseases

Machine Learning in Healthcare technologies in oncology search for cancer-affected cells with the same accuracy as an experienced physician. When a pathologist examines organic fluids from patients, Machine Learning's ability to perform better and quicker analysis can help. Human vision under a microscope is not even 1/2 as fast as a computer-controlled model in terms of analysis, so hospitals and research facilities can benefit from deploying CNN-based hospital uses applications. [45]

CHAPTER 2

Literature Review

Nowadays machine learning (ML) is one of the most noteworthy and effective technologies in the medical industry to diagnose and predict different types of diseases and their stages [8-9]. The huge amount of data set of diagnoses of different diseases can be fed into different machine learning algorithms to explore their patterns and features. This implementation of algorithms in medical databases can help medical professionals significantly to take constructive decisions on diseases, aid them to lessen human-made errors, and eventually ensure a healthy life for mass people.

2.1 Related Works

In recent times as machine learning techniques are being popular in medical sectors for diagnosing; chronic kidney disease is also in the queue to be predicted by dint of machine learning algorithms.

1. Baoshi Chen worked with the common machine learning algorithms such as BP NN, KNN and got the best result using SVM. Their overall classification accuracy rate is 96.05%, the classification accuracy rate for brain tumors is 98.79%. The classification accuracy of normal is 84.21%. [41]
2. Heba Mohsen et al. worked on the DNN learning architecture for classification where the classifier is identifying the brain tumors in brain MRIs. They yielded the best result of 97% precision using DNN. [42]
3. Masoumeh Siar et al. used CNN with Radial Basis Function and Decision Tree. Their method proposed accuracy 99.12% on the test data.[43]

2.2 Research Objective and Outline

In this research, we investigated thirteen Machine Learning algorithms which displayed adequate results in predicting brain tumor. The primary goal of our research work is easing the process to an extent. Considering the present constraints in this field and the importance of detection of Brian Cancer, we had some objectives before we kicked off our undergraduate research work. The objectives are:

1. Studying about Brain Cancer: The goal was to understand the disease, how it functions in our brain, what are the symptoms, how it spreads, what are the stages that are crucial for this disease and the importance of early detection of this disease.

2. Applying Machine Learning Algorithms: Our main objective is to apply several machine learning algorithms into a dataset that contains different attributes about brain cancer. We have applied 13 machine learning algorithms and compared them on the basis of the outcome that was provided by the supervised machine learning algorithms.

3. Getting the proper algorithm: After applying 13 machine learning algorithms, we considered all the possible factors in a real-life scenario and selected the proper algorithm for the highest precision and accuracy.

Chapter 1 provides introduction to our dissertation. Chapter 2 is comprised of a Literature review that includes the current information, containing significant discoveries, and input on a particular subject theoretical and methodological. Another part of chapter 2 contains details about the dataset we have dragged from Kaggle. Chapter 3 talks about the details on all the 13 machine learning methods we used to achieve our goal. The methodology of the work such as the Data Description, Data Pre-processing, Feature Scaling, Data Frames and correlation heatmaps, Hyper Parameter Tuning are
all described in chapter 4. In chapter 6 we have discussed the results and analysis of our model. Here we have discussed the confusion matrices, Comparison of accuracies between different algorithms, Precision, sensitivity, F1 score, and graphical representation for both tuned and without tuned states.

Chapter 6 contains the conclusions of the current work as well as the information about a future query to improve the performance of the methods proposed.

Chapter-3

Study of Machine Learning Algorithms

The Boosting algorithm is used to deal with training data after scaling, and cross validation and processing. It's an ensemble learning strategy that employs a set of Machine Learning algorithms to transform weak learners into strong ones, improving the model's accuracy. Ensemble learning is a way of integrating numerous learners to make a Machine Learning model perform better. When compared to a single model, this sort of learning produces more efficient and accurate models. Unlike many ML models, boosting algorithms try to enhance prediction ability by training a sequence of weak models, each one compensating for the flaws of its predecessors.

However, there are times when the data is so distributed that it is impossible to categorize it using a boundary, which may result in overfitting error. Data is compared and computed in a higher-dimensional connection to overcome this problem without changing the data itself. The procedure is referred to as the kernel trick, and the higher dimensions employed in the computation are referred to as kernels. Random Forest, K-Nearest Neighbor, Extra Trees, Decision Trees, Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Support Vector Classifiers (SVC), NuSVC, LinearSVC, XGBoost, Ada Boost, Gradient Boosting, and Light GBM(LGBM) are among the thirteen algorithms studied and implemented in this study. Ten-fold cross validation is used in all of these methods, and hyperparameter adjustment is used to improve accuracy. Below is a basic summary of the different types of boosting algorithms:

3.1. Linear Discriminant Analysis (LDA):

Linear Discriminant Analysis is, as the name implies, a linear model for relegation and dimensions of space truncation. The most common application is in pattern classification problems for feature extraction also this has been around for a long time while Fisher first developed a linear discriminant for two classes in 1936, and C.R Rao later generalized it for multiple classes in 1948.[8]

Logistic Regression is a popular linear relegation model that works well for binary relegation but falls short in multiple relegation quandaries with well-disunited classes. While LDA handles these quite well. LDA, like PCA, can be utilized in data preprocessing to minimize the number of features lowering computing costs significantly. Face detection algorithms also make use of LDA. LDA is used in Fisherfaces to retrieve useful information from various faces. It yields efficient outcomes when combined with eigenfaces.

FLD's fundamental conception is to project data points onto a line in order to optimize between-class scatter while minimizing within-class scatter so this may appear cryptic, but it is actually quite straightforward [46].

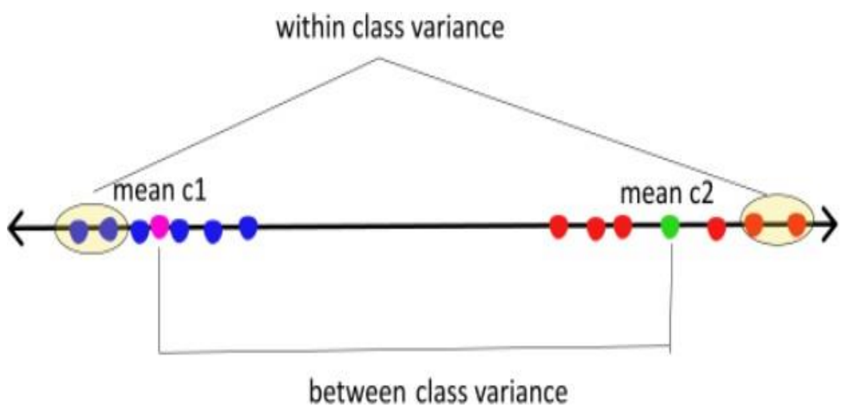


Fig: Mean between class variance in LDA [48]

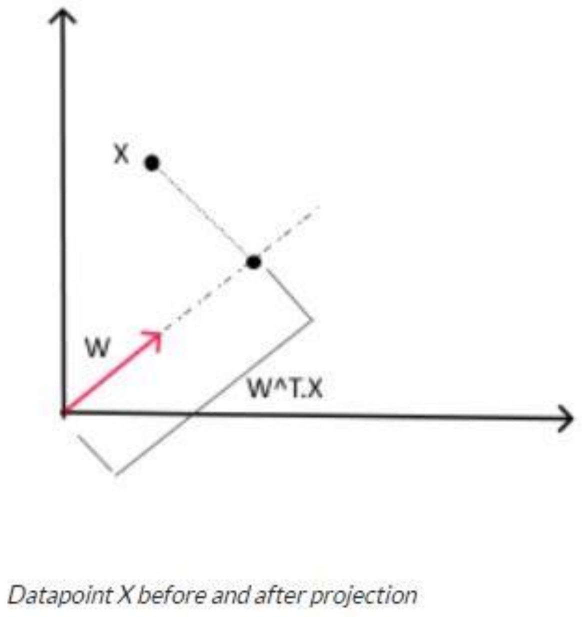


Fig: Datapoint X before and after projection [48]

As per Fisher's LDA:

Arg max J(WO= (M1-M2)2/S12+S22… ....(1)

Numerator:

$$(M_1 - M_2)^2$$

$$= (W^T m_1 - W^T m_2)(W^T m_1 - W^T m_2)^T \quad = W^T(m_1 - m_2)(m_1 - m_2)^T W$$

$$= W^T S_b W \qquad \text{..........} \quad (2)$$

S_b = between class scatter

Denominator: For denominator we have S12+S22.

Class Scatter before projection

$$S_i = \sum_{x(n)\in C_i} (x(n) - m_i)(x(n) - m_i)^T$$

Scatter for projected samples:

$$S_i^2 = \sum_{x(n)\in C_i} (W^T x(n) - M_i)(W^T x(n) - M_i)^T$$

$$M_i = W^T m_i$$

With little re-arrangement we will get:

$$S_i^2 = W^T\Big(\sum_{x(n)\in C_i} (x(n) - m_i)(x(n) - m_i)^T\Big)W$$

$$S_i^2 = W^T S_i W$$

So,

$$S_1^2 + S_2^2 = W^T(S_1 + S_2)\,W = W^T S_W W \quad \text{..........} \ (3)$$

S_W = within class scatter

LDA For Multiple Classes:

LDA can be applied to a variety of classes. The generalized forms of between-class and within class matrices are shown below-

$$S_k = \sum_{x(n) \in C_i} (x(n) - m_i)\,(x(n) - m_i)^T \quad \ldots\ldots \qquad (4)$$

S_w = within class scatter

C =number of distinct classes

$$S_w = \sum_{k=1}^{C} S_k \quad \ldots\ldots\ldots \qquad (5)$$

$$m_i = \frac{1}{N_i} \sum_{x(n) \in C_i} x(n)$$

$$S_b = \sum_{i=1}^{C} N_i (m_i - m)(m_i - m)^T \quad \ldots\ldots\ldots\ldots \qquad (6)$$

S_b = between class scatter

m = mean of all data points

$$m = \frac{1}{N} \sum_{i=1}^{n} x_i$$

$$W = eig(S_w^{-1} S_b) \quad \ldots\ldots\ldots\ldots \qquad (7)$$

3.2. Quadratic Discriminant Analysis

A statistical classifier that uses a quadratic decision surface to separate two or more classes of data is known as quadratic discriminant analysis and this classifier is used when there is a difference between the covariance matrices [9,15].

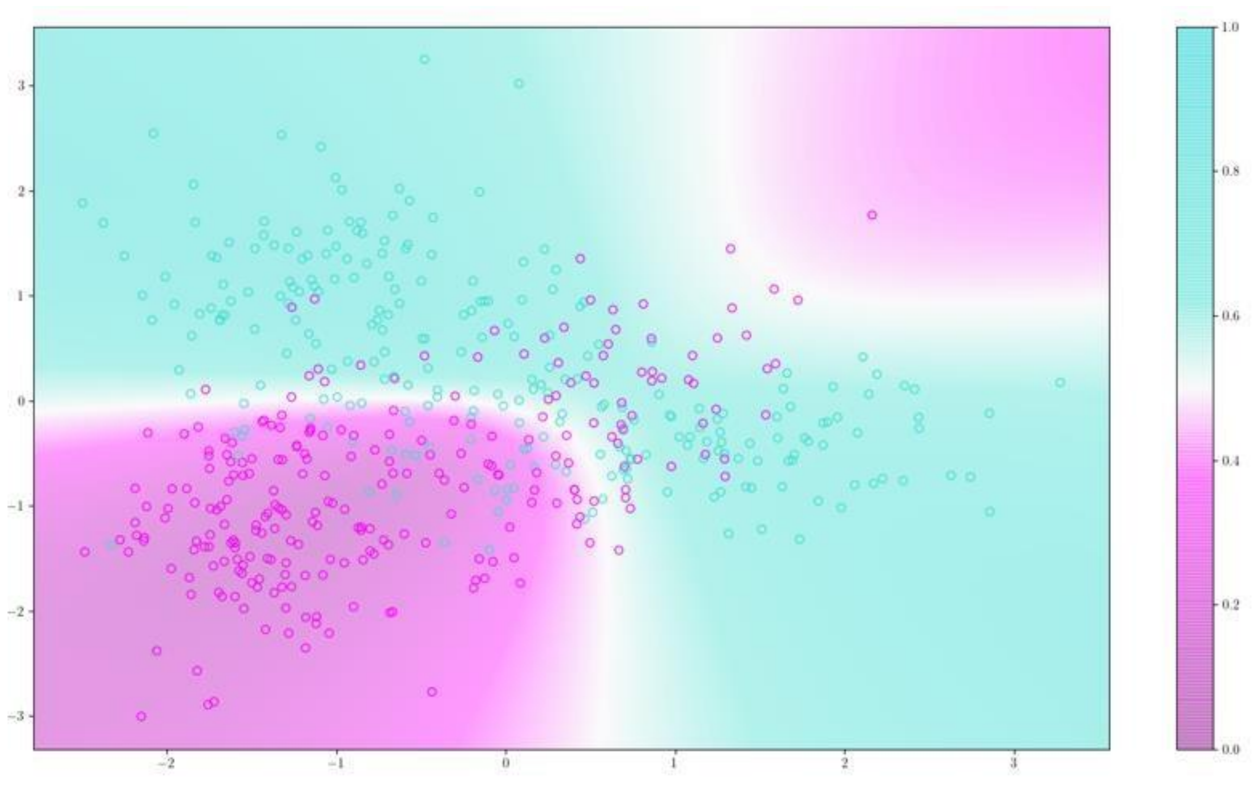


Illustration of the decision boundary generated by a QDA

Fig: Illustration of the decision boundary generated by a QDA[49]

For each of the class y the covariance matrix is given by:

$$\Sigma_y = \frac{1}{N_y - 1} \sum_{y_i = y} (x_i - \mu_y)(x_i - \mu_y)^T$$

By adding the following term and solving (taking log both side). The quadratic Discriminant function is given by:

$$\delta_k(x) = \log \pi_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}_k^{-1} \mu_k + x^T \mathbf{\Sigma}_k^{-1} \mu_k - \frac{1}{2} x^T \Sigma_k^{-1} x - \frac{1}{2} \log |\Sigma_k|$$

QDA is basically a variant of LDA where an individual covariance matrix is estimated for every class of observations. If we have prior knowledge that individual classes exhibit distinct covariances, it exhibits distinct covariances. One limitation is that it cannot be used as a dimensionality reduction technique [10].

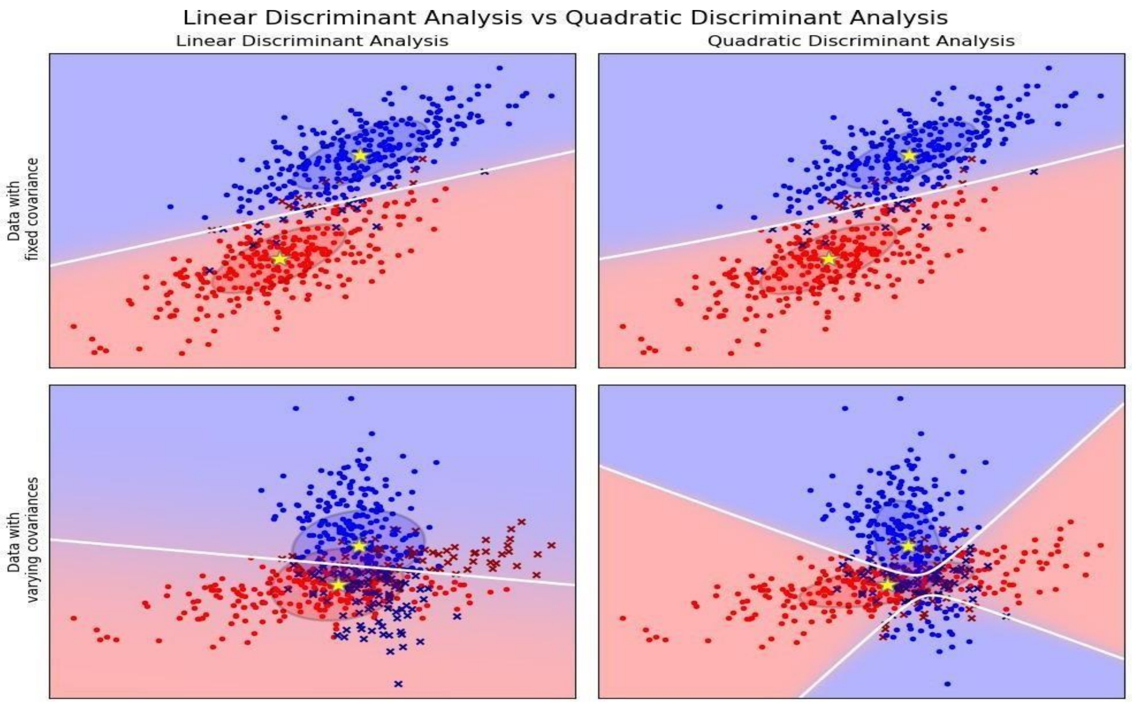


Fig: LDA vs QDA [49]

3.3. XGBoost

Extreme Gradient Boosting XGBoost is a type of decision tree that is geared for speed and performance. In this method, decision trees are built successively. Weights are highly important in XGBoost. Weights are assigned to all of the independent factors, which are then input into the decision tree, which predicts outcomes. The weight of variables predicted wrongly by the tree is escalated, and these variables are passed into the second decision tree. Those individual classifiers/predictors then ensemble to give a strong and more accurate model. [11]

On small to medium structured or tabular datasets, this technique is one of the most effective for regression, ranking, classification and predictions. It does a wide range of tasks by boosting the weak features in datasets using gradient descent architecture. [12].

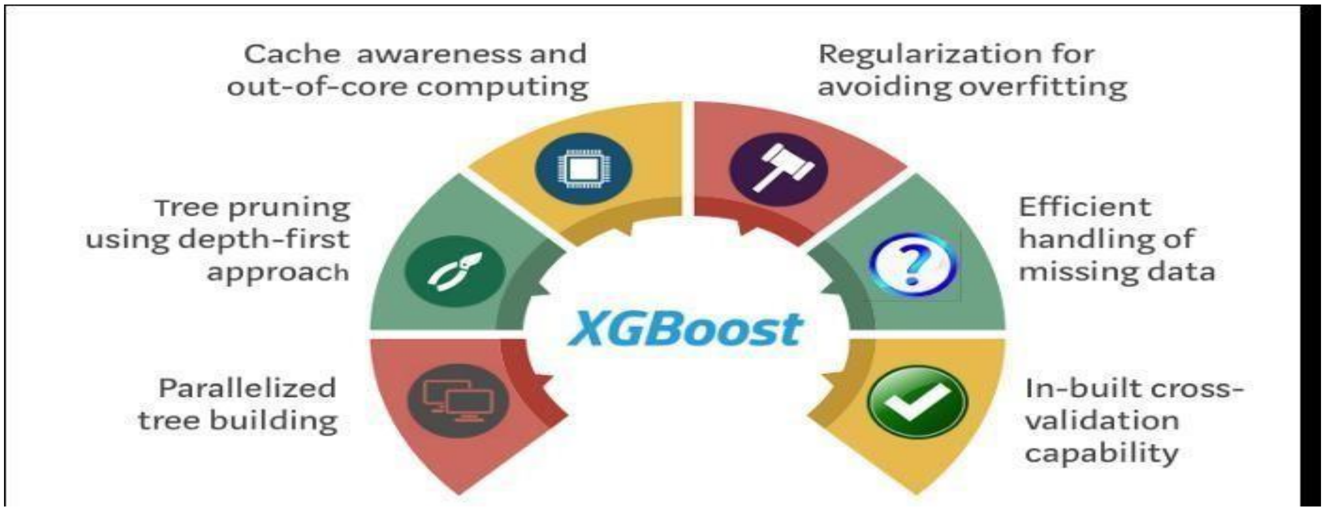

Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Fig: XGBoost Properties and Characteristics [47]

3.4. AdaBoost

AdaBoost is short for Adaptive Boosting. AdaBoost develops a composite powerful learner composite through several rounds. It gradually adds weak learners until a strong learner emerges. A new weak learner is added to the ensemble throughout each step of training, and a weighting vector is changed to focus on instances that were previously misclassified. [13] As the outcome, the classifier is more accurate than weak learner classifiers. The error rate of the weak estimator identifies the flaw:

For a weak classifier C:

$X : n \times d, Y : n \times k$ *with sample weights* $W : n \times 1$

$$Error = \frac{\sum_{j=1}^{n} W_j I(C(X_j) \neq Y_j)}{\sum_{j=1}^{n} w_j}, I(x) = \begin{cases} 1, \text{ if } x \text{ is True} \\ 0, \text{ if } x \text{ is False} \end{cases} \quad [14]$$

AdaBoost recognizes incorrectly categorized data points in each iteration and increases their weights (while decreasing the weights of correct ones) so that the next classifier will pay additional attention to them.

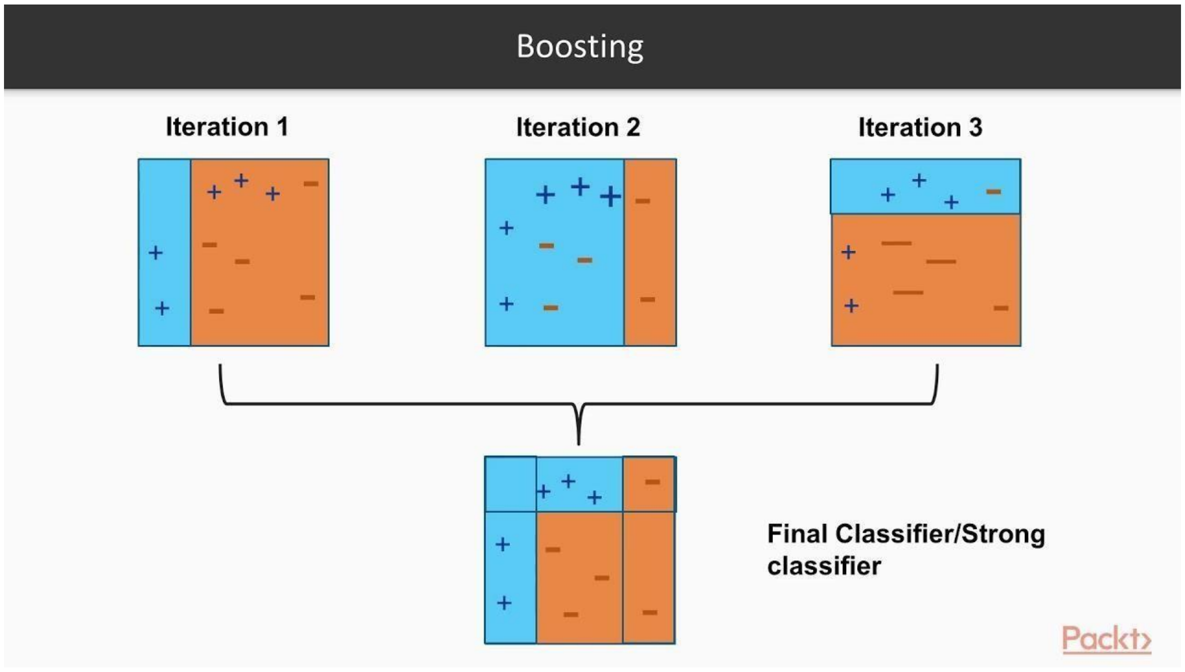


Fig: Iteration Diagram of AdaBoost [47]

3.5 Decision Tree

Decision Tree is another supervised learning method, which uses basic chained decision rules learnt from earlier input variables to train a model to categorize a target variable. The variables are separated iteratively until they reach a stopping point based on a set of impurity criteria. The model of decision tree looks much like an inverted tree where the first rule of decision resides at highest-level and following decision-making rules spreads below like branches of trees. To forecast a class label for a record instance, we begin from the root of the tree [24]. The root value features are compared to the record's feature. We jump to the next node by following the branch that corresponds to that value based on the comparison. The next node is selected depending on which node can provide the maximum gain values. Gini impurity is chosen as the utilized model among numerous impurity measuring techniques. [27-29].

$$Gt = 1 - \sum_{i=1}^{c} p_i^2$$

Here,

G(t) = Gini impurity at node t

p_i = Proportion of observation at class c of node t

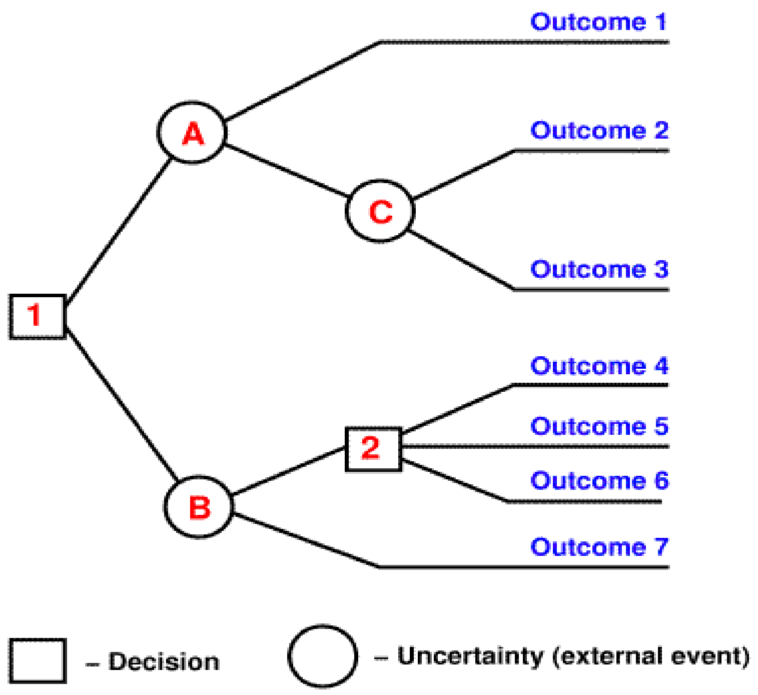


Fig: Tree Diagram of Decision Tree Probable Outcomes [47]

3.6 Random Forest

Random Forest is a regression and classification learning technique that works by constructing many decision trees during training and provides output classes for individual trees. Random forest is based on the assumption that a group of generally uncorrelated models working together will outperform any of the individual constituent models. It generates decent prediction results even without hyper parameter tuning. Bagging is a minor adjustment that uses the de-correlated treeby generating a slew of decision trees using bootstrapped samples from training dta [30]. During bootstrapping, it filters a limited number of feature columns out of all feature columns. Bootstrap modeling reduces variation while increasing the bias [29]. It has an effective method for estimating missing data. After training, the following is a formula for predicting unknown inputs:

$$f = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Where, B= Optimal number of trees

Also, uncertainty of the prediction can be written as:

$$\sigma = \sqrt{\frac{\sum_{b-1}^{B} (f_b x' - f)^2}{B - 1}}$$

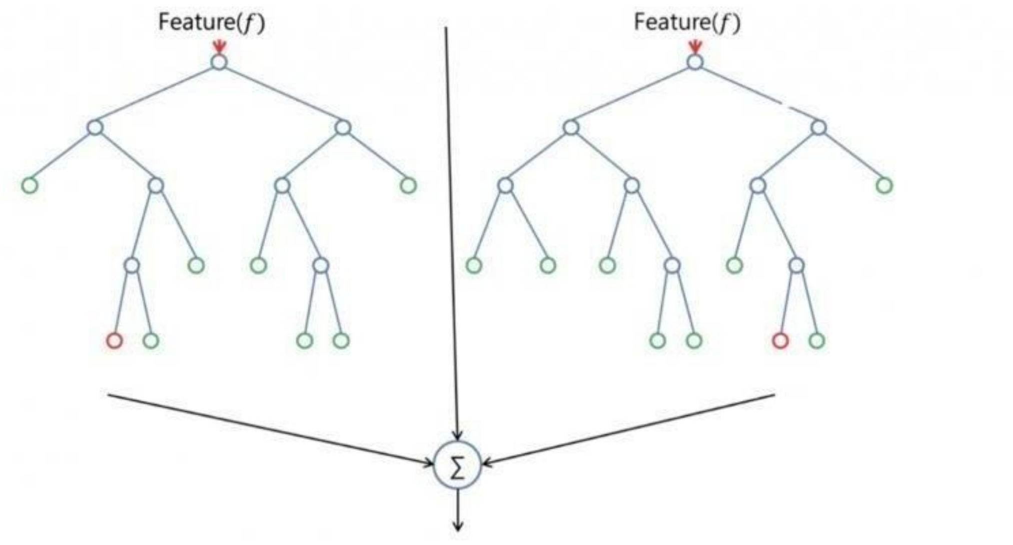


Fig: Random Forest Feature Tree Diagram [47]

3.7 Gradient Boosting

Gradient boosting takes a fresh take on the problem. Gradient boosting, rather than altering data point weights, focuses on the distinction between the forecast as well as the reality.

Gradient boosting has three components:

1. An optimized loss function
2. A slow learner who can't make accurate forecasts
3. A model in which weak learners are included to lower the loss function.

Here is the whole algorithm in math formulas:

1. Set a constant value as the model's initial value:

$$F_0 = \underset{y}{\text{argmin}} \sum_{i=1}^{n} L(y_i, \gamma)$$

2. Where $m = 1$ to M:

   2-1. Compute residuals $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}}$

   2-2. Create terminal node explanations by training a regression tree with features x against r R_{jm} for $j = 1, \dots, J_m$

   2-3. Compute $\gamma_{jm} = \underset{\gamma}{\text{argmin}} \sum_{x_i \epsilon R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) for\ j = 1, \dots, J_m$

   2-4. Update the model:

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1(x \epsilon R_{jm})$$

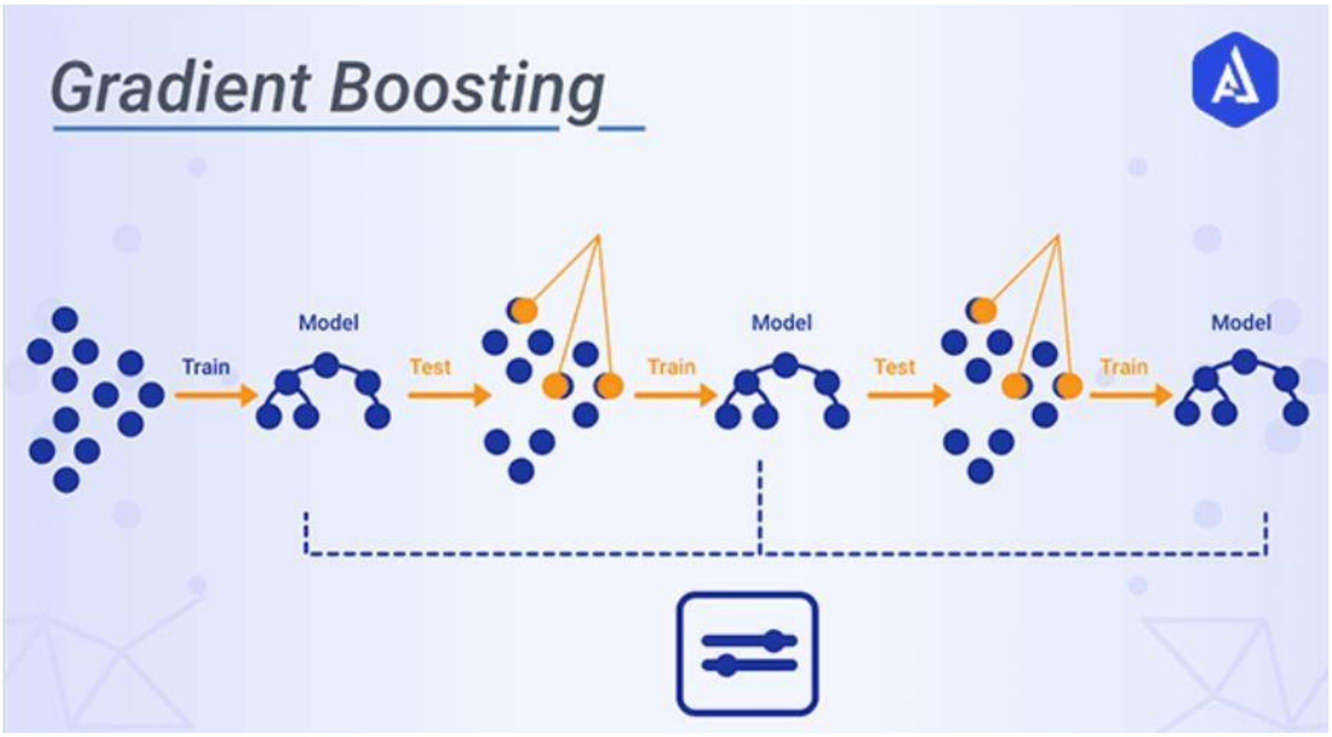


Fig: Implementation Roadmap of Gradient Boosting [50]

3.8 LightGBM

LightGBM is a tree-based gradient boosting system that is distributed and efficient. It uses a histogram to make discrete bins out of continuous variables, resulting in improved memory and quicker training utilization. The framework employs a leaf-wise tree growth approach, unlike many other tree-based systems that use depth-wise growth. Leaf-wise tree growth techniques converge quicker than depth-wise tree growth algorithms. [19-21]

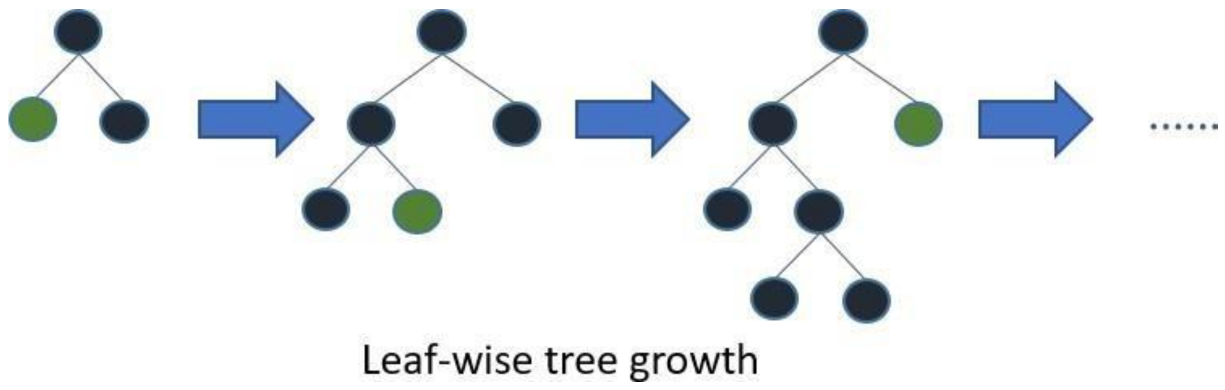


Fig: Leaf-wise tree growth of LightGBM Algorithm [47]

3.9 K-Nearest Neighbor

K-Nearest Neighbors is among the most basic and extensively used supervised machine learning techniques.

The K-NN method anticipates that new data and existing cases are linked and allocates the new case to the category that is the most similar to the current ones. It doesn't really train any dataset; rather, it anticipates that an observation belongs to the class with the most k-nearest neighbors.

The value of K is selected such that the number of mistakes encountered when producing accurate predictions are kept to a minimal. The datapoint nearest to the observation point might be considered t he most comparable to the observation point, hence distance is used as a measure to assess similarity. [24-25]. There are other additional distance measurements, including as:

Euclidean,

$$d_{euclidean} = \sqrt{\sum_{i=1}^{n}(x_i^2 - y_i^2)}$$

Manhattan distance,

$$d_{man\ hattan} = \sum_{i=1}^{n} |x_i - y_i|$$

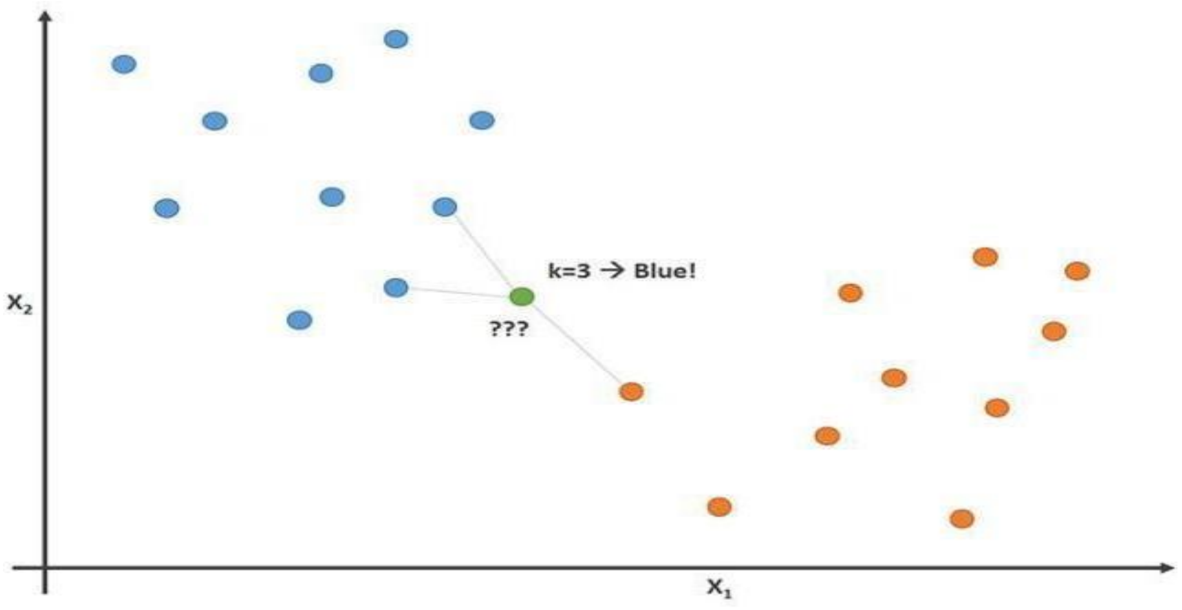


Fig: Visual representation of the working process of KNN Algorithm [47]

3.10 Extra Trees:

Extremely Randomized Trees Classifier (Extra Trees Classifier) is a form of ensemble learning approach that outputs a classification result by aggregating the outcomes of several de-correlated decision trees gathered in a "forest." It is conceptually identical to a Random Forest Classifier, with the exception of how the decision trees in the forest are constructed [28]. The number of decision trees in the ensemble, the number of input features to randomly choose and examine for each split point, and the minimum number of samples necessary in a node to establish a new split point are the three major hyperparameters to adjust in the method [29-30].

The algorithm's variance is increased by the random selection of split points, which makes the decision trees in the ensemble less correlated. Rise the number of trees in the ensemble to counteract the increase in variance.[30]

3.11 Support Vector Classifier (SVC):

The Support Vector Method is such a classifier that attempts to minimize the distance between two sets of closely related data divided by a line [31]. The distance between the categorization border and the nearest data is the margin in this case [31]. Hyperplane, on the other hand, is the boundary plane for an n-dimensional model. The classifier that is based on the Support Vector Machine (SVM) is known as Support Vector Classifier (SVC) [51].

The primary feature of this classifier is that the parameter C, which is a regularization parameter, is applied to it. The *libsvm* library was leveraged in this approach where the classifier's multiclass support is based on a one vs. one (OVO) method. As a result, the OVO decision function of *libsvm* is employed for multiclass classification, which has the shape (n samples, n classes * (n classes - 1) / 2) [31]. In case of binary classification, this parameter is ignored. However, different types of kernels can be applied such as ‘linear’, ‘poly’, ‘RBF’, ‘sigmoid’, ‘precomputed’. If no kernel is specified, RBF is applied by default [31].

For setting the value of gamma, ‘scale’ is set as default which is:

$$gamma = \frac{1}{(n\_features * X.\mathrm{var}())}$$

3.12 Linear SVC:

The technique in Linear SVC can scale a vast quantity of data more correctly or quickly. When it comes to penalties and loss functions, however, it is more flexible. Linear SVC utilizes the liblinear function instead of the libsvm function [31]. Furthermore, the kernel used for this technique is 'linear' and the regularization parameter, C, is set to 1 by default for this algorithm and adjusted later during tuning. L2 and 'squared hinge' must be used in linear SVC. The multiclass support is based on a one-vs-all strategy. The multiclass argument, which trains the classes, is set to 'OVR' by default [31].

3.13 NuSVC:

NuSVC (Nu-Support Vector Classifier) is the same as SVC with the exception of the use of nu parameter instead of C [51]. C has the drawback of being positive and unbounded which makes it difficult to work with during cross validation. Hence it is easier to interpret the regularization more in terms of Nu(**ν**) rather than C [51]. **ν** is lower bounded with the fraction of support vectors and the upper boundary consists of the fraction of outliers [51]. Apart from these slight differences, it's mostly similar to regular SVC being based on the similar library *liblinear* [52].

The relationship between ν and C is ν=(A+B)/C, where A and B are constants [51].

Chapter-4

Methodology

4.1 Data Description

In this study, the algorithms were applied on the Brain Tumor dataset of Kaggle machine learning repository accomodating 700 instances and 1500 features. Kaggle dataset repository is a popular and dependable source of data for application of ML algorithms. We have made use of this repository to collect this dataset. There are 4 types of brain cancers namely- Pituitary Adenoma, Meningiomas, Glioma and Germinoma in this dataset. Each of them has 122, 247, 155 and 176 samples respectively.

Sl. No.	**Types of Brain Cancer**	**Samples**
1.	Pituitary Adenoma	122
2.	Germinoma	247
3.	Meningioma's	155
4.	Glioma	176

4.2 Data Preprocessing

Data preprocessing is an instrumental part of data mining. In the data preprocessing step of machine learning, the data of the large datasets get transformed and encoded so that the machine can easily compile it i.e., convert the data, images into zeros (0) and ones (1). There are primarily two kinds of data in a dataset. Which are:

- Numerical data: Data that can be expressed using numbers is known as numerical data. For example: age, birth year, any kind of quantity etc.
- Categorical data: Data whose values are extracted from a defined set of values. For example: Monday, Thursday, Boolean expressions (true, false)

Both kind of data are pushed through some steps of data preprocessing. The steps are:

A. Data Quality Assessment: Machine learning model run poorly on poor quality data, they do not provide accurate predictions, classification, regression results on poor data. So, the quality of the data will possess significant impact on the results that different machine learning models will provide. In this step, the following type of data will be filtered and furnished:

    a. Missing values
    b. Duplicate values
    c. Inconsistent values

B. Feature Assemble: In this step, the random, unorganized data will be put into perspective to build some patterns and categorize the data. This will minimize the memory storage requirement and fasten up the machine learning models.

C. Feature Sampling: In machines learning, we often need to deal with a very large dataset consisting hundreds or thousands of instances. The information a dataset carries is directly proportional to its size. But working with such huge datasets are money and time consuming. Instead, we can get satisfactory results by taking a portion of the dataset into the analysis which can save both time and money. This is called feature sampling. This enables the machine learning model to learn more quickly and accurately.

D. Reduction of dimensionality: The operation of reducing the number of input variables in a dataset in referred to as dimensionality. Reduction of dimensionality is used in areas that deal with large numbers of observation, such as digital signal processing, speech recognition and bioinformatics.

E. Feature Encoding: Sometimes there are some categorical entries in the dataset like true/false, yes/no etc. But machine learning models can only work with numerical entries. For this reason, it is imperative that the categorical values are transformed into numerical values. This process is called feature encoding.

To get the best out of the model, data has to be well organized and clean. It was done so here by reorganizing the data in steps. We got the data screened in for determining features that had 80% dependencies and similarities. We stored the linear relationship of the features in the dataset in a matrix form upon calculation. The application of Pearson Correlation Coefficient followed. The equation is as follows:

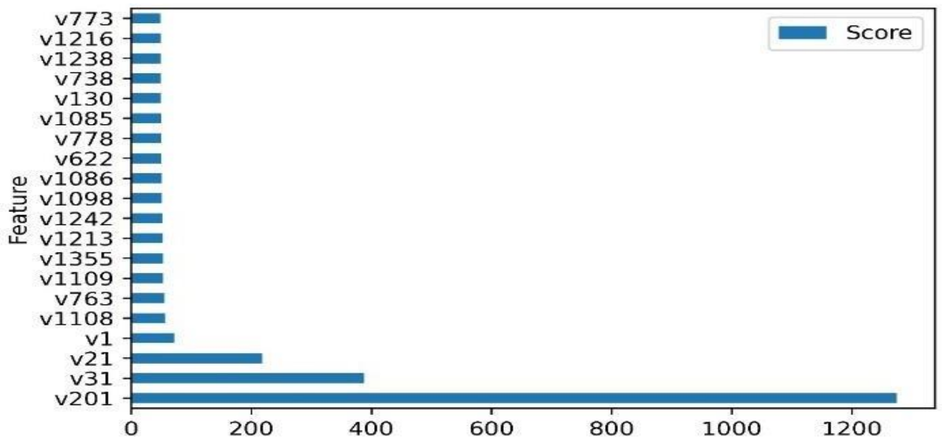
$$R_{a,b} = \frac{n\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{n\sum a_i^2 - (\sum a_i)^2}\sqrt{n\sum b_i^2 - (\sum b_i)^2}}$$

Here, n = number of data for the features, a_i = ith data of feature a, b_i = ith data of feature b, $R_{a,b}$ = correlation coefficient.

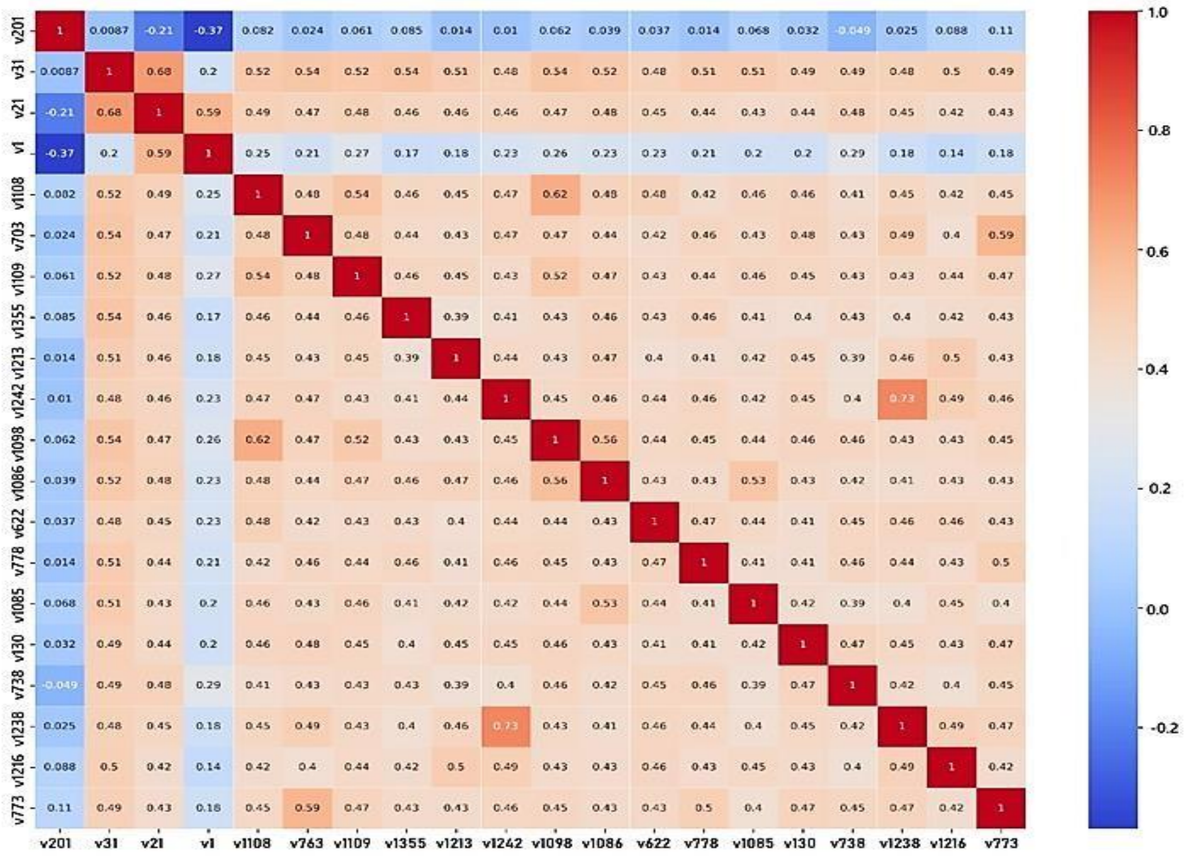
Following this, 1285 of the 1500 features were discarded for the aforementioned amount of dependencies and similarities. With the features reduced to 215 from 1500, top 20 were calculated using k-best algorithm. Correlation heatmap was produced following this. All of their values were scaled between 0 and 1 using min-max scaler. Then we split the data into training and testing sets following stratified train test split at 70:30 ratio

4.3 Data Frames & Correlation Heatmaps:

K-best Algorithm Score for determining the top 20 features:



Produced Correlation Heatmap:



4.4 Hyper-Parameter Tuning:

Eleven machine learning algorithms were implemented to predict patients with chronic kidney diseases or without chronic kidney diseases. First of all, algorithms were implemented with default hyper parameter setting. Accuracy along with other performance metrics were calculated. For acquiring better prediction hyper parameter tuning of the machine learning algorithms is essential. There are two popular ways of hyper parameter tuning:

- RandomizedsearchCV
- GridsearchCV

RandomizedsearchCV: RandomizedsearchCV is a strategy of tuning hyperparameters where random combinations of the hyperparameters are implemented for finding the best fit for the model. The parameters are chosen completely at random. RandomizedsearchCV is very effective in case of many parameters to try and the training time is a concern.[32]

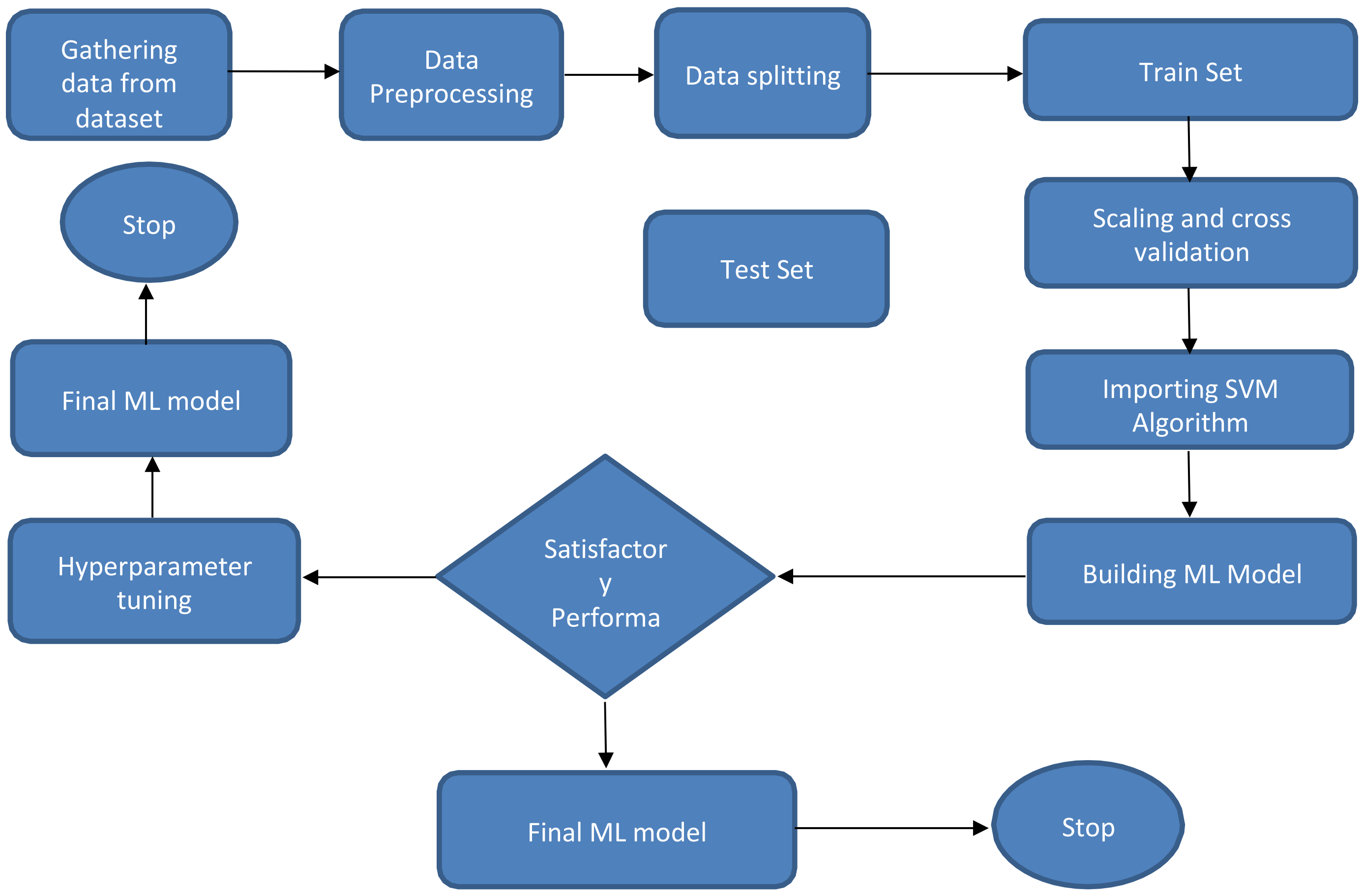
GridsearchCV: GridsearchCV is an effective technique for finding the parameters in supervised learning algorithms and making the model more generalized. GridsearchCV runs every possible combination of parameters of interest. Then, the best set of parameters are chosen. [32]

GridsearchCV performs well in case of a small number of hyperparameters. On the other hand, if the number of parameters is high and calculation time is a concern, it is better choice to use the RandomzedsearchCV. [32]

In this research hyperparameters were tuned with GridsearchCV technique. Then performance metrics were calculated accordingly to compare with the default hyperparameters.

4.5 Methodology Flow Chart:

The entire methodology can be illustrated by the following flow chart:



Chapter-5

Results Analysis

5.1 Confusion Matrices

Confusion Matrix is a performance indicator for ML models with two or more classes as output. It is made up of a table with four combinations of actual and expected values. For machine learning classification models where output can be two or more classes, it is useful for calculating other performance metrics like Recall, Precision, Specificity and Accuracy. [33]

	Predicted	
Actual	TF	FP
	FN	TP

True Positive: Expected positive incident comes out as positive.

True Negative: Expected negative incident gives a negative output.

False Positive (Type 1 Error): The Predicted positive incident is actually negative.

False Negative (Type 2 Error): The expected negative incident is actually positive.

Confusion Matrices of eleven algorithms in detecting CKD from our research is given below:

5.2 Comparison of Accuracies among Different Algorithms

One of the most frequently used and simplest performance metrics for ML algorithms is Accuracy. The ration of the data points that were successfully predicted to the total available data points gives us the accuracy of a certain model. It can be calculated by dividing the summation of True Positive (TP) and True Negative (TN) by the summation of True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). TN and TP are data points that the algorithm has rightly defined as true or false. Whereas the False Positive and False Negatives are incorrectly classified to be True or False.[1] The equation of accuracy for classification problem is given below:

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \quad [33]$$

This along with other metrices like precision and recall that use various other ratios of these parameters provide us with a detailed insight on the classification of data points.

The accuracies of all the algorithms with and without tuning is tabulated below:

TABLE

COMPARISON OF ACCURACIES AMONG DIFFERENT ALGORITHMS

Algorithm	**Accuracy (Without Tuning)**	**Accuracy (With Tuning)**
Random Forest	0.9667	0.9762
Extra Trees	0.9333	0.9524
KNN	0.8	0.8476
Decision Tree	0.9	0.9238
QDA	0.9333	0.9619
LDA	0.7095	0.7095
XGB	0.9524	0.9619
Ada Boost	0.7048	0.8762
Gradient Boosting	0.9571	0.9714
LGBM	0.9667	0.9762
SVC	0.9286	0.9476
NuSVC	0.8571	0.9571
Linear SVC	0.7286	0.7429

The Comparison between accuracies of different algorithms are expressed graphically in the diagram below:

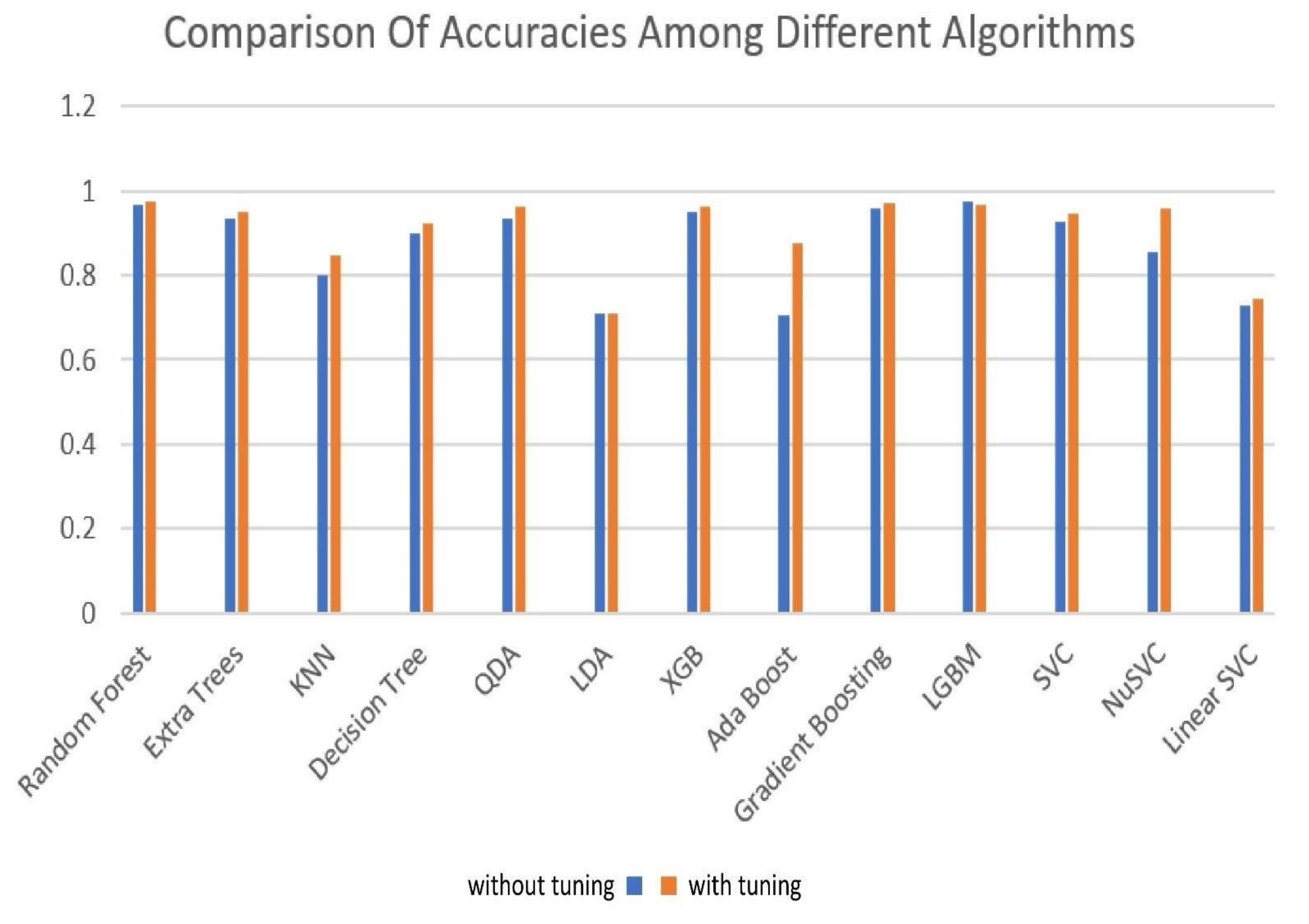


Fig: Comparison of Accuracies Among Different Algorithms

5.3 Comparison of Precisions among Different Algorithms

A model's accuracy does not always imply that it is the best. Only if the dataset is symmetric, meaning that false negative and false positive incidences have the same value, can accuracy be a useful statistic. This is not the case with illness detection models. As a result, other metrics such as accuracy were used to assess the model's success. The ratio of correctly predicted positive occurrences to total predicted positive incidents is known as precision. Low false positive rate suggests high precision.[33] It gives us a count of how many positive class predictions are actually in the positive class. Precision and recall are commonly used together in the F1 Score to create a single device calculation. [35]

$$\text{Precision}=\frac{TP}{TP+FP}$$ [33]

Comparison of precisions among different algorithms for both default hyperparameters and tuned hyper parameters is given below:

TABLE
COMPARISON OF PRECISIONS AMONG DIFFERENT ALGORITHMS

Algorithm	**Precision (Without Tuning)**	**Precision (With Tuning)**
Random Forest	0.9667	0.9762
Extra Trees	0.9333	0.9524
KNN	0.8	0.8476
Decision Tree	0.9	0.9238
QDA	0.9333	0.9619
LDA	0.7095	0.7095
XGB	0.9524	0.9619
Ada Boost	0.7048	0.8762
Gradient Boosting	0.9571	0.9714
LGBM	0.9667	0.9762
SVC	0.9286	0.9476
NuSVC	0.8571	0.9571
Linear SVC	0.7286	0.7429

The Comparison between Precisions of different algorithms are expressed graphically in the diagram below:

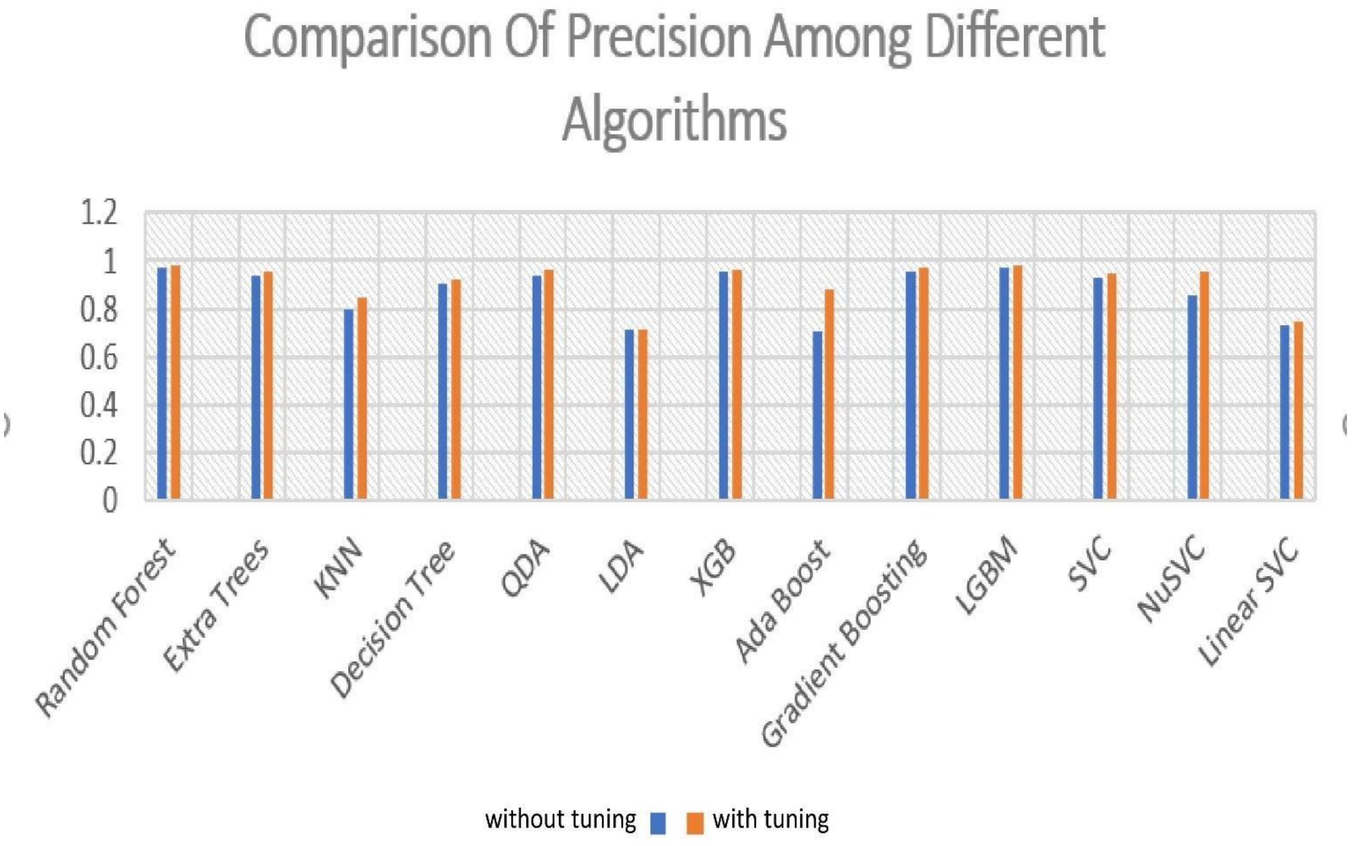
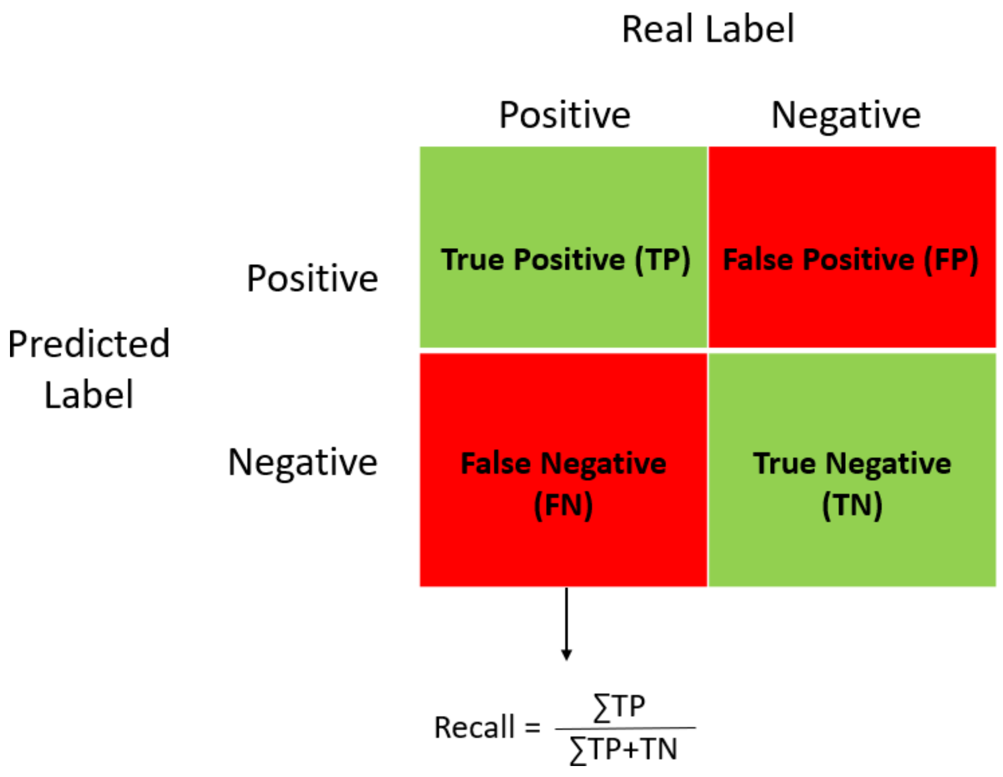


Fig: Comparison of Precisions Among Different Algorithms

5.4 Comparison of Recalls among Different Algorithms

The ratio of correctly anticipated positive occurrences to all positive episodes is known as recall. The issue of how many patients with true CKD did the model detect is answered via recall. Recall doesn't only apply to a binary classifier only. It can be used in higher classes as well [36]. It is usually the ration between the True Positive and the summation of True Positive and False Negative. [33]

$$\text{Recall}=\frac{TP}{TP+FN}\ [33]$$



Comparison of recalls among different algorithms for both default hyperparameters and tuned hyperparameters are given below:

TABLE
COMPARISON OF RECALLS AMONG DIFFERENT ALGORITHMS

Algorithm	**Recall (Without Tuning)**	**Recall (With Tuning)**
Random Forest	0.9667	0.9762
Extra Trees	0.9333	0.9524
KNN	0.8	0.8476
Decision Tree	0.9	0.9238
QDA	0.9333	0.9619
LDA	0.7095	0.7095
XGB	0.9524	0.9619
Ada Boost	0.7048	0.8762
Gradient Boosting	0.9571	0.9714
LGBM	0.9667	0.9762
SVC	0.9286	0.9476
NuSVC	0.8571	0.9571
Linear SVC	0.7286	0.7429

Comparison of recalls among all the implemented algorithms are represented graphically as follows:

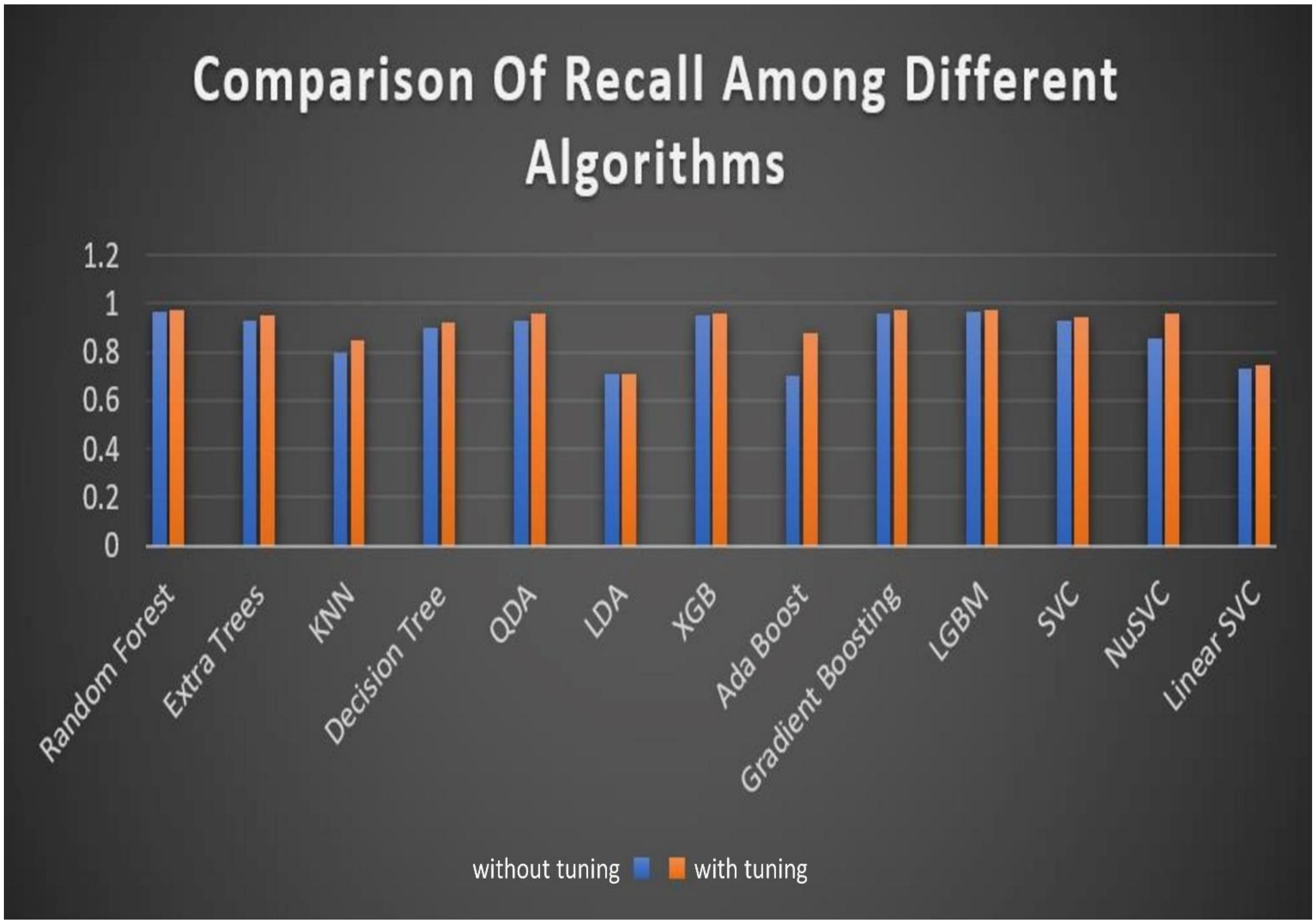


Fig: Comparison of Precisions Among Different Algorithms

5.5 Comparison of F1-scores among Different Algorithms

F1_score as we know it, is one of the most crucial evaluation metrics in Machine Learning. F1 Score refers to the weighted average of Precision and Recall. It sums up precision and recall in its calculation to diligently portray a convalescent predictive performance. [37] This performance metric calculates with both false negative and false positives being accounted for. F1-score gives more insights than accuracy if the model deals with an uneven class distribution [33]. Accuracy is the best performance parameter if both the false negatives and false positives have similar weight. But this is not true in diseases detection models. So, we took recall and precision into account for out model. F1_Sore is calculated with the formula below:

Formula

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

The F1_Scores of all algorithms in tabulated form is shown below:

TABLE
COMPARISON OF F1-SCORES AMONG DIFFERENT ALGORITHMS

Algorithm	**F1_Score (Without Tuning)**	**F1_Score (With Tuning)**
Random Forest	0.9667	0.9762
Extra Trees	0.9333	0.9524
KNN	0.8	0.8476
Decision Tree	0.9	0.9238
QDA	0.9333	0.9619
LDA	0.7095	0.7095
XGB	0.9524	0.9619
Ada Boost	0.7048	0.8762
Gradient Boosting	0.9571	0.9714
LGBM	0.9667	0.9762
SVC	0.9286	0.9476
NuSVC	0.8571	0.9571
Linear SVC	0.7286	0.7429

Comparison of recalls among all the implemented algorithms are represented graphically as follows:

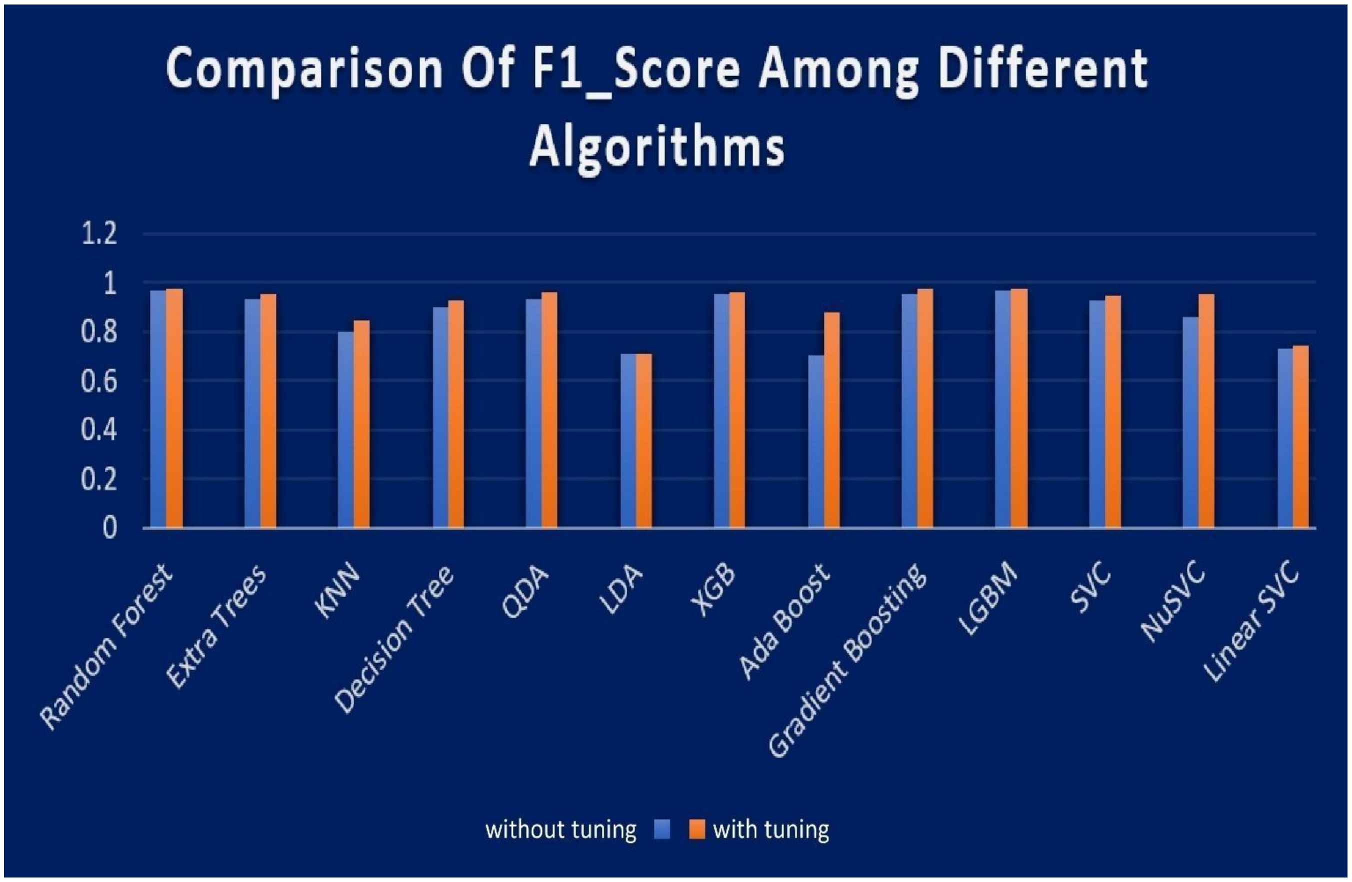


Fig: Comparison OfF1_ScoreAmong Different Algorithms

5.6 Comparison of Specificity among Different Algorithms

When evaluating model performance, sensitivity is frequently compared to specificity. The proportion of real negatives that the model managed to detect correctly is measured by specificity [38] . It implies that some true negatives will be predicted as positives, which might be delineated as false positives. It is also called True Negative Rate (TNR). Specificity (actual negative rate) and false positive rate would always add up to 1.[38] The majority of negative findings are accurately identified by a high specificity model, but many negative outcomes are wrongly labeled as positive by a low specificity model.

If we try to understand it with a model, it'd be the measure of people not suffering from the disease to the ones who were predicted to not be suffering from the disease correctly. In other words, it's the ratio of people who is actually healthy to the ones diagnosed to be healthy correctly.[33]

$$\text{Specificity}= TN/(TN+FP) \text{ [38]}$$

Here, TN= True Negative

FP= False Positive

Comparison of Specificity among different algorithms for both default hyperparameters and tuned hyper parameters is given below:

TABLE

Comparison Of Specificity Among Different Algorithms

Algorithm	**Specificity (Without Tuning)**	**Specificity (With Tuning)**
Random Forest	0.8958	0.9167
Extra Trees	0.7447	0.7959
KNN	0.9048	0.9706
Decision Tree	0.7609	0.9333
QDA	0.7143	0.8776
LDA	0.0612	0.0612
XGB	0.8571	0.8776
Ada Boost	0	0.7143
Gradient Boosting	0.8776	0.898
LGBM	0.9184	0.9184
SVC	0.7347	0.8936
NuSVC	0.6939	0.9184
Linear SVC	0.0204	0

Comparison of recalls among all the implemented algorithms are represented graphically as follows:

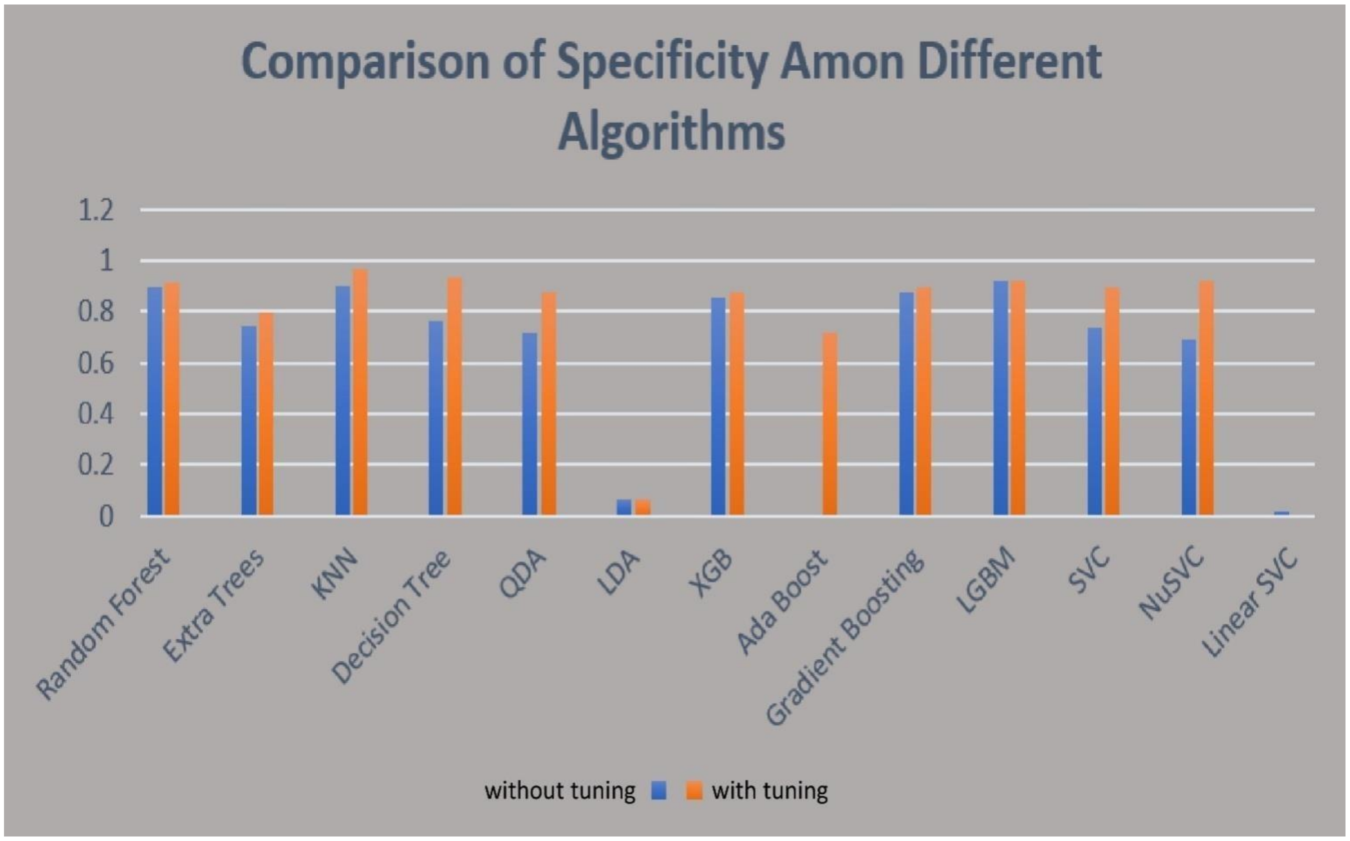


Fig: Comparison of Specificity Among Different Algorithms

5.7 Comparison Of Cross Validation Score Among Different Algorithms:

When a machine learning model or algorithm captures the noise of the data, overfitting occurs. It implies that the model fits the data too well. It yields better results in training data sets but poor results on new ones. Underfitting on the other hand cannot capture the underlying data trends. It in turn occurs when the model or algorithm doesn't fit the data well.[39] When the missing data is mishandled or no outliner treatment is available, features are removed for dependency or similarity, underfitting is observed. It makes the dataset too simple which causes the inconsistency in fitting of the model or algorithm. Cross validation is used to overcome this ambiguity. [33,39]

Cross validation basically refers to the division of data sets into a number of subsets. Then they are divided as testing sets and training sets. The model trains one set at a given time and the rest of the time is used on the other set. Types of cross validation are - K-fold Cross Validation, Stratified K-Fold Cross Validation, Leve One Out Cross Validation.[39] The basic idea remains the same, but the application is different depending on the process of selection of the value of K. K is chosen in such a manner that each set can represent part of a broader dataset.

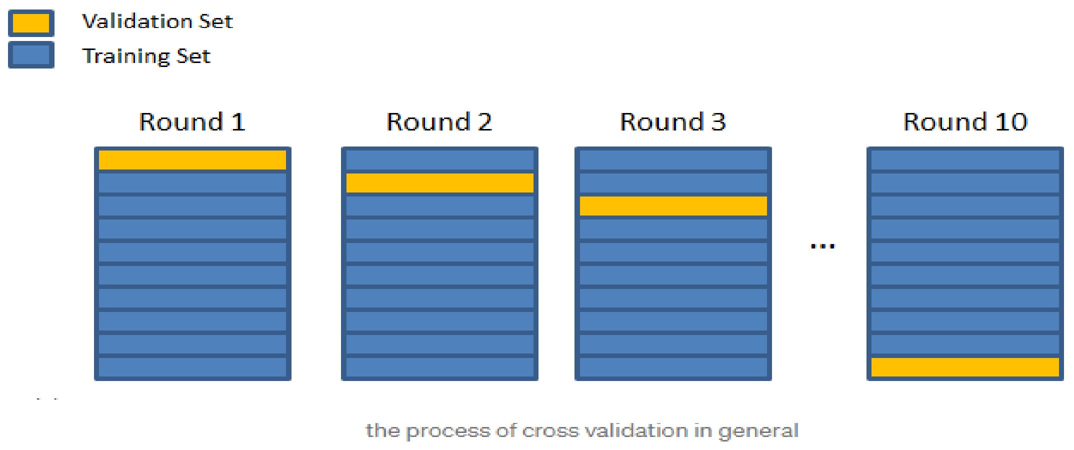


Fig: The iteration of cross validation for each of the subsets of a dataset[39]

Comparison of Cross Validation Score among different algorithms for both with and without hyperparameter tuning is given below:

TABLE

Comparison OfCross Validation Score Among Different Algorithms

Algorithm	**Cross Validation Score (Without Tuning)**	**Cross Validation Score (With Tuning)**
Random Forest	0.9557	0.97
Extra Trees	0.9629	0.9629
KNN	0.8514	0.8629
Decision Tree	0.9171	0.9243
QDA	0.9743	0.9571
LDA	0.6957	0.6957
XGB	0.9529	0.9571
Ada Boost	0.6829	0.8243
Gradient Boosting	0.9314	0.9557
LGBM	0.955	0.9629
SVC	0.9471	0.9657
NuSVC	0.8986	0.9571
Linear SVC	0.7286	0.7329

Comparison of Cross Validation Score among all the implemented algorithms are represented graphically as follows:

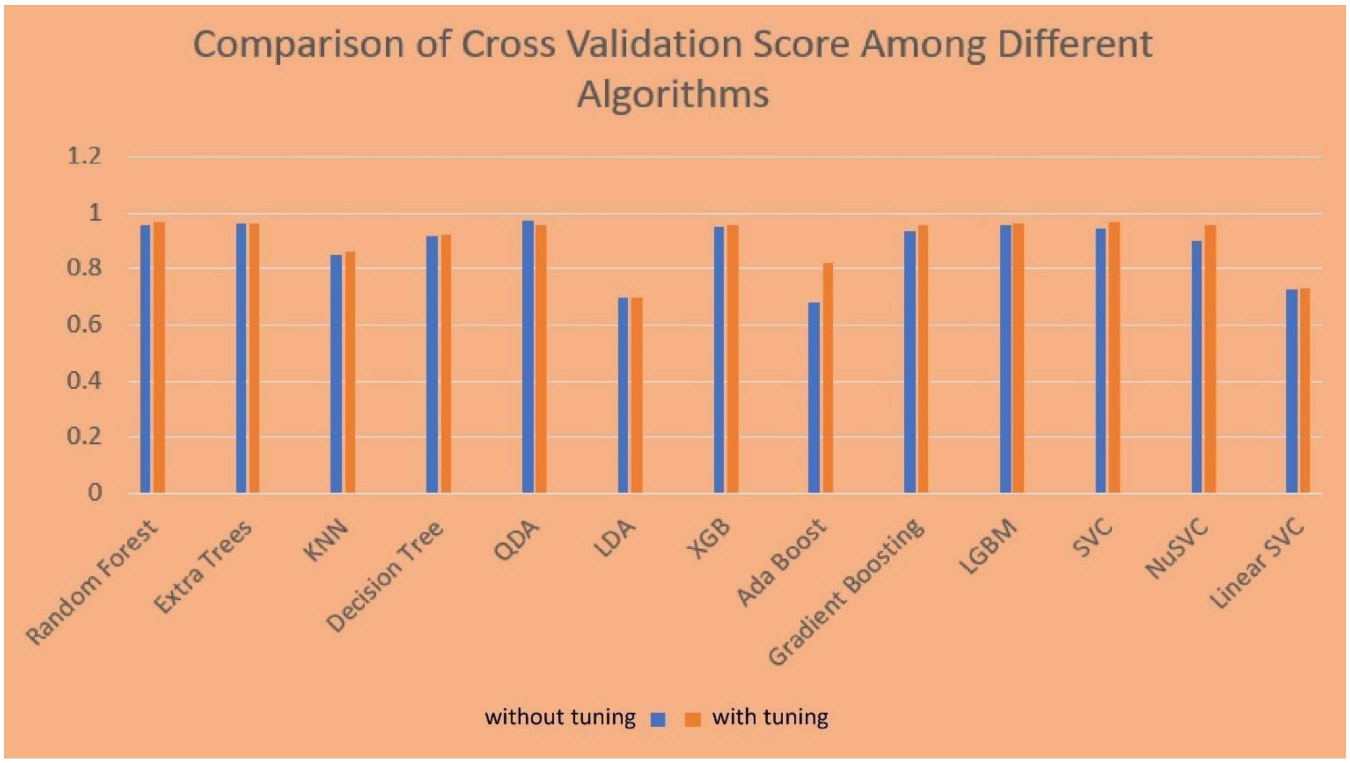


Fig: Comparison of Cross Validation Score Among Different Algorithms

5.8 Comparison of Error Rate Among Different Algorithms:

Error rate is basically the measure of the proportion of prediction that went wrong. Error is expressed as error rate if the expected outputs are categorical. [40] The total number of incorrect predictions to the total number of instances in the dataset gives us the measure of Error Rate (ERR).

Here, FP = False Positive

FN = False Negative

TP = True Positive

TN = True Negative

$$\text{ERR} = \frac{FP+FN}{TP+TN+FP+FN} = \frac{FP+FN}{P+N}$$

The value of error rate always lies from 0 to 1. So, the best error rate can be 0.0 whereas the worst is 1.0[40]

Error rates of different algorithms both with and without hyperparameter tuning is tabulated below:

TABLE

Comparison of Error Rate Among Different Algorithms

Algorithm	**Error Rate (Without Tuning)**	**Error Rate (With Tuning)**
Random Forest	0.0556	0.037
Extra Trees	0.1121	0.0917
KNN	0.1184	0.172
Decision Tree	0.1442	0.1132
QDA	0.9743	0.9571
LDA	0.6957	0.6957
XGB	0.0833	0.0556
Ada Boost	0.4587	0.1495
Gradient Boosting	0.0741	0.0463
LGBM	0.0556	0.037
SVC	0.1376	0.0841
NuSVC	0.1468	0.0826
Linear SVC	0.4667	0.4587

Comparison of Cross Validation Score among all the implemented algorithms are represented graphically as follows:

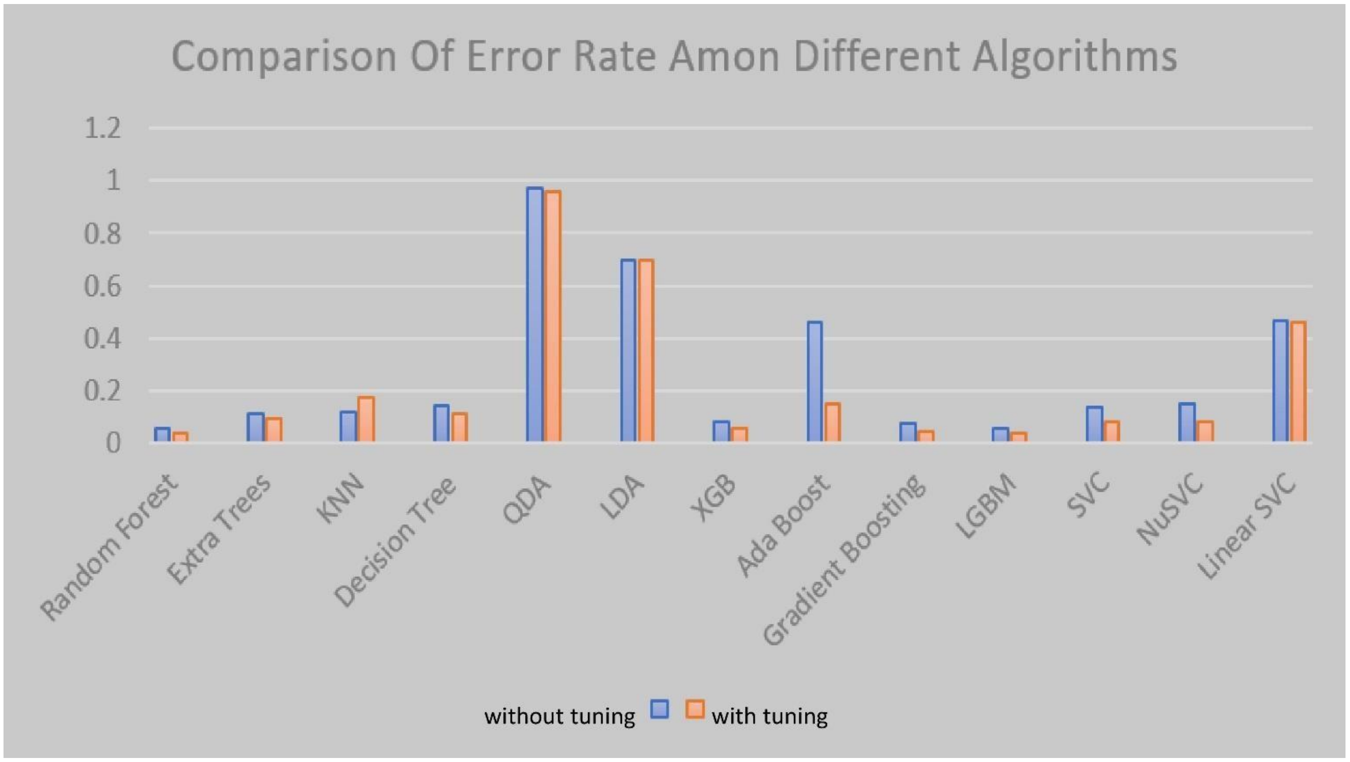


Fig: Comparison of Error Rate among all the implemented algorithms

5.9 Performance Metrices Comparison

After investigating all the performance metrics for all the algorithms, best performance metrics among all the algorithms are given below:

TABLE

BEST PERFORMANCE METRICS

Performance Metric	Algorithm (Without Tuning)	Algorithm (With Tuning)
Accuracy	LGBM, Random Forest- 0.9667	LGBM, Random Forest- 0.9762
Precision	LGBM, Random Forest- 0.9667	LGBM, Random Forest- 0.9762
Recall	LGBM, Random Forest- 0.9667	LGBM, Random Forest- 0.9762

Cross Validation Score	QDA- 0.9743	Random Forest- 0.97
F1-Score	LGBM, Random Forest- 0.9667	LGBM, Random Forest- 0.9762
Specificity	LGBM-0.9184	KNN-0.9706
Error Rate	Random Forest, LGBM- 0.0556	Random Forest, LGBM- 0.037

From the table it is seen that Random Forest and LightGBM has the best overall performance. They have the best results in all performance evaluations both before and after hyperparameter tuning bar one, where KNN has the best specificity upon hyperparameter tuning.

Chapter-6

Conclusion and Future Works

Brain is an indispensable and irreplaceable organ of human body. Malignant brain tumors resulting in cancer has been fatal and leading cause of children death worldwide for quite some time [31]. Through early detection of brain cancer, we can save hundreds of thousands of lives and avert catastrophic complications. Machine learning techniques used to computer-aided diagnosis systems will be extremely useful in predicting chronic renal illnesses. We evaluated different brain tumor features and constructed 13 machine learning algorithms to detect brain cancer. This study has a significant advantage in terms of producing consistent accurate results while working with a large number of features.

The models, however, must be tested on a much bigger scale before being employed as a clinical assistant to medical practitioners. This work can be expanded by employing larger and more practical data sets that can aid in the early and accurate detection of brain tumors. In the current industrial generation, digitalization is blooming more than ever and everything is being computerized. Hence, in future, we aim to make our model more interactive which will take live input from a device or sensor instead of a dataset and give the corresponding prediction based on the trained model and make the lives of healthcare professionals easier through proper pre-screening.

References

[1] https://www.abta.org/about-brain-tumors/brain-tumor-education/?gclid=EAIaIQobChMI_r66sYvP9wIVj5pmAh2FKQ7WEAAYASAAEgJs9fD_BwE

[2] https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor/brain-tumor-types

[3] https://www.cancer.net/cancer-types/brain-tumor/symptoms-and-signs

[4] Sung, H, Ferlay, J, Siegel, RL, Laversanne, M, Soerjomataram, I, Jemal, A, Bray, F. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin. 2021: 71: 209- 249. https://doi.org/10.3322/caac.21660

[5] https://www.cancer.net/cancer-types/brain-tumor/diagnosis#:~:text=In%20general%2C%20diagnosing%20a%20brain,after%20a%20biopsy%20or%20surgery.

[6] https://www.expert.ai/blog/machine-learning-definition/

[7] https://spd.group/machine-learning/machine-learning-in-healthcare/#:~:text=Machine%20Learning%20for%20healthcare%20technologies,%2C%20various%20tests%2C%20and%20screenings.

[8] https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/

[9] https://en.wikipedia.org/wiki/Quadratic_classifier#Quadratic_discriminant_analysis

[10] https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/#:~:text=A%20disadvantage%20of%20QDA%20is,k)%2Blog%CF%80k.

[11] https://www.mygreatlearning.com/blog/xgboost-algorithm/

[12] https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost- algorithm-long-she-may-rein-edd9f99be63d#:~:text=XGBoost%20is%20a%20decision%2Dtree,all%20other%20algorithms%20or%20frameworks.

[13] https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

[14] https://www.mathworks.com/discovery/adaboost.html

[15] https://en.wikipedia.org/wiki/Quadratic_classifier#Quadratic_discriminant_analys is

[16] https://www.statology.org/quadratic-discriminant-analysis/

[17] https://en.wikipedia.org/wiki/AdaBoost

[18] https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

[19] https://www.kdnuggets.com/2020/06/lightgbm-gradient-boosting-decision-tree.html

[20] https://lightgbm.readthedocs.io/en/latest/Features.html

[21] https://www.quora.com/How-does-the-LightGBM-algorithm-work-conceptually

[22] Dangeti, Pratap. Statistics for machine learning. Packt Publishing Ltd,2017.

[23] https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c

[24] Albon, Chris. Machine learning with python cookbook: Practical solutions from preprocessing to deep learning. " O'Reilly Media, Inc.",2018.

[25] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[26] https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4#:~:text=Decision%20Tree%20algorithm%20belongs%20to%20the%20family%20of%20supervised%20learning%20algorithms.&text=The%20goal%20of%20using%20a,prior%20data(training%20data).

[27] Dangeti, Pratap. Statistics for machine learning. Packt Publishing Ltd,2017.

[28] https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/

[29] https://towardsdatascience.com/extra-trees-please-cec916e24827

[30] https://jagan-singhh.medium.com/extra-extremely-randomized-trees-5ce9026bd07f

[31] M. M. Nishat, F. Faisal, T. Hasan, M. F. B. Karim, Z. Islam and M. R. K. Shagor, "An Investigative Approach to Employ Support Vector Classifier as a Potential Detector of Brain Cancer from MRI Dataset," *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021, pp. 1-4, doi: 10.1109/ICECIT54077.2021.9641168.

[32] https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10

[33] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[34] https://deepai.org/machine-learning-glossary-and-terms/accuracy-error-rate

[35] https://deepchecks.com/glossary/precision-in-machine-learning/

[36] https://deepchecks.com/glossary/recall-in-machine-learning/

[37] https://towardsdatascience.com/essential-things-you-need-to-know-about-f1-score-dbd973bf1a3

[38] https://classeval.wordpress.com/introduction/basic-evaluation-measures/

[39] https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

[40] https://users.sussex.ac.uk/~christ/crs/ml/lec03a.html

[41] BaoshiChen ,Lingling Zhang , Hongyan Chen , Kewei Liang ,
XuzhuChen , A Novel Extended Kalman Filter with Support Vector Machine Based Method for the Automatic Diagnosis and Segmentation of Brain Tumors, Computer Methods and Programs in Biomedicine (2020),

[42] Heba Mohsen, El-Sayed A. El-Dahshan, El-Sayed M. El-Horbaty, Abdel-Badeeh M. Salem, Classification using deep learning neural networks for brain tumors, Future Computing and Informatics Journal,Volume 3, Issue 1,2018,Pages 68-71,ISSN 2314-7288

[43] M. Siar and M. Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm," 2019 9th International Conference on Computer and Knowledge Engineering

(ICCKE), 2019, pp. 363-368, doi: 10.1109/ICCKE48569.2019.8964846

[44] https://history.computer.org/pioneers/samuel.html

[45] Hongjun Zhang, Jin Yao, "Automatic Focusing Method of Microscopes Based on Image Processing", Mathematical Problems in Engineering, vol. 2021, Article ID 8243072, 9 pages, 2021. https://doi.org/10.1155/2021/8243072

[46] http://www.facweb.iitkgp.ac.in/~sudeshna/courses/ml08/lda.pdf

[47] http://103.82.172.44:8080/xmlui/handle/123456789/1462

[48] https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/

[49] https://scikit-learn.org/stable/modules/lda_qda.html

[50] http://www.akira.ai/glossary/gradient-boosting

[51] https://www.datatechnotes.com/2020/06/nu-support-vector-classification.html

[52] https://datascience.stackexchange.com/questions/51813/what-are-the-differences-between-svc-nusvc-and-linearsvc