DETECTION OF AUTISM SPECTRUM DISORDER BY MACHINE LEARNING ALGORITHM

by

Afsana Hossain Bristy (170021002)
Tasnimul Hasan (170021056)
Md Minhajul Islam Shawon (170021084)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment 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

A dissertation on

DETECTION OF AUTISM SPECTRUM DISORDER BY MACHINE LEARNING ALGORITHM

Approved by:

(Signature of the Supervisor)

Dr. Md. Ashraful Hoque
Professor
Dean Faculty of Engineering
Department of Electrical and Electronic Engineering
Islamic University of Technology

(Signature of the Co-Supervisor)

Mirza Muntasir Nishat
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology

(Signature of the Co-Supervisor)

Fahim Faisal
Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology

Table of Contents

List of Tables

List of Figures

List of Acronyms

ASD	Autism Spectrum Disorder
UCI	University of California Irvine Repository
KNN	K-Nearest Neighbor
GNB	Gaussian Naïve Bayes
MNB	Multinomial Naïve Bayes
BNB	Binomial Naïve Bayes
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
MCAR	Missing Completely At Random

Dedication

With this work, we dedicate it to our families, friends, and everyone else who has stood by our sides and been a key source of comfort and inspiration through the high points and low points, the joys and sorrows of life. We wish them the best of luck, happiness, and fulfillment in their respective endeavors.

Acknowledgements

First, we express our heartfelt gratitude to Allah, the Almighty, for providing us with the patience and strength to complete this thesis work successfully. For providing us with the opportunity to work under the supervision of our respected supervisor, **Prof. Dr. Md. Ashraful Hoque**, dean of the Faculty of Engineering at IUT, we are really grateful. He has pointed us in the right path in terms of how to undertake a research project and has encouraged us to gather a solid foundation for our work. He has always served as an inspiration for us to produce more and more inventive work. Without his guidance and encouragement, we would have lost our way and become disoriented right from the start.

We would like to convey our heartfelt gratitude to our co-supervisor, **Mr. Mirza Muntasir Nishat**, Lecturer in the Department of Electrical and Electronic Engineering at IUT, for generously donating his time and effort whenever it was needed. At a time when we were completely unfamiliar with the topic of research, he exposed us to it. It has been tremendously beneficial to have his constant guidance and perspective. Whenever we were stuck with a difficulty, he would always find a way to motivate us and find a solution for us, no matter how complicated the situation was. Our ability to conduct this investigation would be severely limited if we didn't have his clear instructions.

We are also grateful to our co-supervisor, **Mr. Fahim Faisal**, Assistant Professor, Department of Electrical and Electronic Engineering, IUT, for providing us with clear instructions, valuable time, and valuable input that greatly aided us in the completion of our thesis work. Throughout the entire process, he provided us with invaluable insights gleaned from his years of expertise in the field of research, which enabled us to gain a clear understanding of how to begin and proceed through a thesis work.

Abstract

This work proposes an investigative strategy to examining the efficacy of alternative boosting algorithms in terms of improving the accuracy of diagnosing Autism Spectrum Disorder (ASD). When it comes to autism spectrum disorder (ASD), the average number of cases per 10,000 people has increased from 1.9 in 1980 to 14.8 in 2010, according to data from Asia. Early detection and identification are crucial for improved treatment outcomes in ASD. Different boosting machine learning algorithms have been shown to be an effective technique for detecting autism spectrum disorder (ASD) when it is still in its early stages. This research utilized a dataset containing a total of 1100 instances that was amalgamated from three datasets, with 104 instances being teenagers and 704 instances being adults, with the remainder of the instances being kid instances, from the University of California Irvine (UCI) repository. This dataset was used to train and test the model classification classifier. To begin the functioning of the classifiers, several methodologies were used to build distinct data frames and correlation heatmaps, which were then compared. Eight machine learning methods were investigated, and their performance parameters such as the confusion matrix and accuracy were measured and compared to one another. Furthermore, a comprehensive comparative research was carried out by simulating the precision, sensitivity, F1 score, and ROC-AUC of each algorithm and comparing the results.

Chapter 1

Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder in which a person has a lasting influence on interaction and communication in everyday life. Symptoms of autism spectrum disorder (ASD) are typically seen in children. According to the ASD theory, the problem begins in childhood and persists into adulthood, affecting both children and adults. Autism can be diagnosed at any point of a person's life and is referred to as a "behavioral disorder" since it manifests itself in the first two years of adolescence and adulthood.

When it comes to neurodevelopmental disorders, Autism Spectrum Disorder (ASD) is a group of conditions that are not curable but can be mitigated by early intervention[1].

1.1 Autism Spectrum Disorder: A Brief Study

Autism Screening Disorder (ASD) is a neural developmental disorder or a neurological condition that results in significant challenges such as social, behavioral, and communicational challenges and can have a negative effect on the affected individual's entire family. People with ASD face difficulties communicating normally with other people, have some repetitive or restricted behaviors, sensitive to certain things such as touch, light, or sound, and show reciprocal social interaction. The causes of Autism are mainly linked with genetic as well as neurological factors [2]. The symptoms of ASD are likely to appear for a child from an early age to 3 years and continue for a lifetime. Children and adults with ASD face insomnia and their brain functions differently [3]. Early detection and treatment of ASD can help to minimize the effects of the patient as the disease is not completely curable at all. As there are no specific medical tests for ASD detection, the symptoms of ASD can be observed to determine if someone has ASD or not [4].

Autism is expected to be an underappreciated but important health issue in a low-income country like Bangladesh. Autism was found to be 0.2 and 0.84 per 1000 children in group research conducted by Mullick and Rabbani in 2005 and 2009. According to a systematic analysis, the prevalence of ASD in Bangladesh ranges from 0.15 to 0.8 percent 6. The patient registry data at Cambridge Medical University revealed a rise in the number of autistic patients receiving care that increased from 12 to 105 between 2001 to 2009. In 2013 a national study

was conducted in Bangladesh and the study showed that the prevalence of autism was found to be 0.15 percent among 7200 people in 7 Upazilas which are Pekua, Wazhirpur, Pirgonj, Debhhata, Godagari, Kushtia, and Modhupur [5]. Many of the ASD patients remain untreated because of the less knowledge or the undetected symptoms of the disease. For this, the affected individual as well as, their parents suffer more.

It is essential to identify the ASD symptoms and detect the patient as soon as possible and to do this we need Autism screening, also known as a preliminary diagnosis that is the first step in the assessment process for determining if an individual has autism[6]. To provide automatic efficient and effective classification models for the ASD problem, machine learning methods combine mathematical and different search methods from computer science. A variety of machine learning techniques, including decision trees, support vector machines, rule classifiers, and neural networks, have recently been applied to the ASD problem by researchers. ASD diagnosis is a common machine learning classification problem that develops a model from observing previous cases and controls which are classified. The new case diagnosis form can then be used to predict whether or not a person has ASD using this model [7].

1.2 Providing Assistance to Children with Autism

Autism is believed to be a lifelong illness, and there is currently no known cure for the condition. Moreover, early, well-planned, and on an individual basis specialized academic efforts in specifically adapted environments will have a significant impact on the development of numerous children with autism. The primary goals are to support the kid in developing intentional communication skills. Information about the unique learning styles of children with autism should be included in educational techniques that are tailored to their needs. Individuals with autism can benefit from academic strategies that are specifically tailored to their needs. Several ABA methods, as well as the systematic teaching method within the TEACCH-model (Treatment and Education of Autistic and Related Communication Handicapped Children), are examples of such strategies. The first stage is early detection, diagnosis, and assessment, and the second step is providing parents with accurate information. A specifically designed nursery school and later schooling are essential requirements for the kid; in addition, a specially adapted home environment, as well as everyday activities, are essential requirements for the adult. Adults and adolescents may be able to continue to participate in

academic activities in order to further acquire skills that will allow them to be more independent and participate more actively[8].

1.3 Machine Learning

Machine learning has risen to become one of the most efficient and vital techniques in the engineering sector as a result of the development of artificial intelligence. Machine learning is a type of artificial intelligence that is used in the real world. It gives a programmed system the potential to learn and improve from past knowledge without having to be expressly programmed in this manner[7].

Machine learning assists computers in learning for autonomously without the need for human intervention or assistance. As a bonus, it provides the computers with the ability to conduct actions based on their previous experiences. It is necessary to have a model that is composed of large datasets in order to provide the prior experience. These datasets provide information about a specific event that is used to assist the model in learning how to train itself. These datasets are the most significant resource for a machine learning model since they contain the most information. Based on the data, the model performs all of the processes necessary to assist the user in obtaining the desired result.

There are three basic types of machine learning methodologies, each of which is based on the type of model being used.

a) Supervised Machine Learning is a type of Machine Learning under Supervision.
b) Unsupervised Machine Learning is a type of machine learning that does not require supervision.
c) Reinforcement Machine Learning is another type of machine learning.

A model that already incorporates data from previous experience is used in supervised learning, whereas an unsupervised model is used with data that is completely new. Or, to put it another way, a supervised model directs the input to the output, but an unsupervised system simply travels to the output without following any rules. This method employs the model that interacts with the dynamic model, in which a certain task is fixed and must be accomplished by the computer, as opposed to the traditional method of teaching.

1.4 Machine Learning in Healthcare

Machine learning is now being employed in almost every aspect of our lives, including healthcare and education. Machine learning is used in a variety of applications ranging from everyday checking emails to launching a rocket into deep space. The healthcare industry is not the only field in which machine learning is being used, and this is not an exclusive list. Eventually, it will become one of the most significant applied fields of machine learning.

Due to technological advancements, machine learning is becoming increasingly popular in the healthcare industry and is assisting patients and physicians in a variety of ways. In healthcare, the most common applications of machine learning are in the areas of billing automation, clinical decision assistance, and the establishment of clinical care recommendations. On the basis of a proper diagnosis, clinical assistance and guidelines assist us in accurately diagnosing diseases and taking precautionary measures as needed. Machine learning has the potential to be a revolutionary and effective technique for testing medical problems. It's mostly about exploring and extracting information from massive datasets.

Autism Spectrum Disorder (ASD) requires to be identified at an early stage. The use of machine learning can significantly assist health practitioners in diagnosing Autism Spectrum Disorder with the help of a symptom-based approach. Researchers all over the world have already begun to use data-science and machine learning algorithms to determine autism spectrum disorder (ASD) in its early and mid-stages, in order to assist medical professionals in providing better treatment to the general public before ASD reaches a critical stage and increases the risk of other fatal diseases.

One of the most notable and effective technologies in the medical business today is Machine Learning , and it may be used to diagnose and forecast many diseases and illnesses and their stages, as well as their progression. As machine learning is concerned with the investigation of large datasets and the patterns, features, modes, and so on that they include. It is possible to feed different machine learning techniques with a massive amount of data set containing diagnosis of various diseases. This application of algorithms in medical databases can immensely assist medical professionals in making constructive decisions about diseases, preventing errors, and ensuring that the general public lives a healthy and productive life. In this study, we analyzed eight Machine Learning algorithms that demonstrated proficiency in identifying Autism Spectrum Disorder.

1.5 Thesis Outline

In chapter – 1, The Brief Study of Autism Spectrum Disorder, Machine learning is explained. The necessity of ML algorithms for increasing the accuracy of detecting the disorder & the contribution of Machine Learning in Healthcare is also described.

In chapter – 2, The relevant research works published recently on Autism Spectrum Disorder data classification using Machine Learning algorithms are analyzed comparatively that in-line with our research topic of interest.

In chapter – 3, the central architecture & methodology of our entire work, performance matrices that we calculated are narrated. The Data preprocessing & Feature engineering process that is being performed in this thesis model are visually depicted, which is an integral part of any Machine learning research.

In chapter – 4, the 8-Machine learning algorithms & their working procedures are briefly described that we've implemented & optimized to compare the performance matrices with other algorithms.

In chapter – 5, describes the confusion & performance matrices of all the 8 algorithms & compares the overall result & performance with QDA (Quadratic Discriminant Analysis) algorithm, which portrays 99.77% accuracy.

Chapter – 6, Finally, the discussion of our research work is brought to a close in by identifying future-scopes for improving and better implementing our study of particular interest.

Chapter 2

Literature Review

Today, machine learning (ML) is one of the most significant and effective methods in the medical industry for diagnosing and predicting various types and stages of diseases. Different machine learning algorithms can be fed a massive data collection of disease diagnoses to examine their patterns and characteristics. This incorporation of algorithms in medical databases will assist medical practitioners make more informed decisions about diseases, reduce human error, and ultimately assure a healthy life for the general population.

2.1 Relevant Research

Research has gained know how and when to grasp and analyze different algorithm and predictive models in recent years for the diagnosis and treatment of neuro-developmental disorders such as ASD. Machine study is useful for recognizing challenging methodologies. Machine-learning algorithms are therefore used to carry out the binomial classification task to identify disorder detection characteristics. Some researchers have concentrated on research into autism. In particular, the ASD diagnosis showed that machine learning techniques can improve diagnostic effectiveness.

Allison et al. [9] suggested therefore a short quantitative checklist, which can be used at different stages of the patient's life, including kids, children, teenagers and young adults. Later on, Thabtah et al. [10] suggested a new model of mobile applications designed to solve the problem by delivering a friendly, efficient and usable mobile autism spectrum screening tool called ASD Tests to customers and the health community. The application is an important tool for data collection in children, young people and adults with respect to ASD. There were more than 1400 cases. Two algorithms were used by Thabtah, namely Naïve Bayes and Logistic Regression. Results obtained using the Naïve Bayes algorithm in children amounted to 92.80%,

adolescents to 91.34 % and adults to 95.73%. The logistic regression results for children were 97.94%, 94.23% for youth, and 99.85% for adults.

A set of 5 behavioral attributes which show decent sensitivity and specificities almost matched the level of performance of the established ADOS algorithms, were reduced to the original autism diagnostic monitoring schedule (ADOS) by Charlotte Küpper et al [11] . They used vector support machines to increase ASD detection by acknowledging some behavioral attributes in module 4 of ADOS. They used clinical routine data for 385 adults with ASD, 673 adolescents with high-level functions and 283 suspects of ASD. In order to compare model performances, the areas of the achieved minimized subset ROC(AUCs) predictions, the existing 11 sub-set items presented by the ADOS algorithm and the ADOS as a whole were evaluated. They have increased their accuracy in their models when dividing their data into subgroups.

In order to develop a new mobile app that was used for ASD prediction in all aged people, Omar et al. [12] proposed an effective predictive model based on machine learning. The model has been developed by combining Random Forest-CART and Random Forest-ID3 to use a new algorithm. The developed model was assessed by the AQ-10 dataset, and 250 actual datasets from people with or without autistic features were collected. The results of the evaluation showed that in children the proposed model has an accuracy of 92.26%, in adolescents 93.78% and in adults with AQ-10 datasets 97.10%, which has an accuracy far higher than in actual data sets.

A procedure to identify autism with the optimal behavior sets has been proposed by Vaishali R, Sasikala R. [13] and others. In this work, a dataset of 21 ASD diagnostics collected from the UCI machine learning repository experimenting with the binary firefly feature selection wrapper based on swarm intelligence. The other experimental hypothesis claims that a machine learning model can achieve a better grading precision with minimum subsets of features. It is found that 10 of the 21 features of an ASD dataset are sufficient for distinguishing ASD from non-ASD patients through a swarm-based, single-objective binary firefly selection wrapper.

Rahman (2010) advocated for the use of an interactive computer game to increase the intelligence of autistic children's speech in their communication. Autism is a condition for which there is no definitive treatment.

Autistic children being served by giving them with games and educational opportunities to help them enhance their abilities. By utilizing the classification approach supervised learning SVM, Santos investigates the first identification of Autism in the year 2013. This means that the symptoms of the patient throughout childhood, backed by preverbal vocalization, are taken into

consideration (support vector machine). Chaminade began a pilot research in 2012 to employ magnetic resonance imaging (MRI) to analyze young individuals with autism who were engaging with a humanoid robot. Prud'hommeaux[14] and colleagues investigate the issues associated with machine learning algorithms for classification of non-standardized text. It was argued by Kathleen T Quach[15] that the categorization challenge stems from the fact that autism spectrum disorder (ASD) may be a tremendously diverse condition that would include subgroups with completely different genetic expression fingerprints. It should be beneficial to categorize the ASD class into subgroups and supplement the input set with clinical parameters in order to improve classification accuracy. In this paper, Alexander Genkin[16] and colleagues present a straightforward Bayesian logistic regression strategy that makes use of a Laplace prior to avoid overfitting and produces dense predictive models for text data. Their research involved applying this approach to a variety of document classification techniques and demonstrating that it generates compact predictive models that are at the very least as effective as those produced by SVM (support vector machine) classifiers or ridge (logistic regression) in combination with feature selection.

2.2 Comparative Analysis of Relevant Research

Research Paper	Algorithm	Best Accuracy
[13]	Swarm intelligence based single-objective binary firefly feature selection wrapper	92.12%-97.95%
[17]	LibSVM IBk Naïve Bayes	93.26% 92.3% 91.34%
[18]	Deep Embedded Presentation Learning	99% sensitivity 99% specificity.
[19]	Logistic Regression and Fuzzy Rule	97.33% accuracy 97.06% sensitivity
[20]	Artificial Neural Network	100%
[21]	Linear Discriminant Analysis (LDA) K-Nearest Neighbor (KNN)	90.8% 88.5%
[22]	FNN MLP J48	98.49 (1.03) 99.95(0.26) 100(0.00)
[23]	Modified Grasshopper Optimization Algorithm (MGOA) with Random forest Classifier	100%

[6]	kNN model	94.97%
[4]	Convolutional Neural Network	99.53%, 98.30%, 96.88% for Autistic Spectrum Disorder Screening in Data for Adult, Children, and Adolescents respectively.
[3]	Recursive Feature Elimination and Stability Selection methods	100%
[24]	Random Forest classifier	96%
[25]	Levenberg-Marquardt Algorithm	98.38%

Table 2.1: *Comparative Analysis*

2.3 Research Objective and Outline

The primary goal of our research is to make the process as painless as possible for participants. We established some goals before beginning our undergraduate research work, taking into consideration the current limits in this field and the need of early identification of Autism Spectrum Disorder.

The objectives are given in the following:

i) Studying about Autism Spectrum Disorder (ASD): The goal was to gain a better understanding of the disease, including how it acts in the patient's body, what symptoms to look for, how it affects the patient's body, what phases of the disease are critical, and the importance of early diagnosis of the condition in order to save lives.

ii) Our primary goal is to apply many machine learning algorithms to a dataset that comprises various parameters related to autism spectrum disorder (ASD) and see which ones perform the best. To compare the results of the supervised machine learning algorithms, we deployed eight machine learning algorithms and evaluated them on the basis of the outcomes that they provided.

iii) Building a computer-aided diagnosis system for the efficient detection of Autism Spectrum Disorders: In addition to maintaining satisfactory performance parameters, we worked very hard on developing a computer-aided diagnosis system that can accurately detect Autism Spectrum Disorders from any input data. The method is supervised by a huge dataset that already contains previous attributes on ASD as well as the consequence of those qualities.

The first chapter of our dissertation serves as an introduction. The second chapter is a survey of the literature that covers current knowledge, including substantive findings, as well as feedback on a specific subject from a theoretical and methodological standpoint. Another section of chapter 2 offers information on the dataset we obtained from the University of California Irvine (UCI). All of the 8 machine learning algorithms we used to achieve our goal are discussed in detail throughout Chapter 3, which is the last chapter. Various aspects of the work's approach, such as data description, data pre-processing, Feature Scaling, Data Frames and correlation heatmap, and Hyper Parameter Tuning, are covered in detail in Chapter 4. Our model's results and analysis were discussed in detail in Chapter 6. Among the topics covered were confusion matrices, comparison of accuracies across different algorithms, precision, sensitivity and F1 score, as well as AUC-ROC curves for both hyperparameter tuned and untuned states, among other things.

2.4 Data Set and Attributes

The data set is the necessary raw material for the use of machine-learning algorithms. In this study, we used the Autism Spectrum Disorder dataset from the University of California, Irvine machine learning repository to test our methods. When implementing machine learning techniques, the UCI dataset collection is one of the most widely utilized and dependable resources available. The dataset we used contained a total of 1100 instances that were amalgamated from three datasets, with 104 instances being teenagers, 704 instances being adults, and the remaining instances being kid instances. The attributes consists of age, gender, ethnicity, born with jaundice, family member with PDD, who is completing the test, country of residence, used the screening app before, screening method type, question 1 answer, question 2 answer, question 3 answer, question 4 answer, question 5 answer, question 6 answer, question 7 answer, question 8 answer, question 9 answer, question 10 answer, screening score.

Chapter 3

Methodology

3.1 Basic Methodology

The basic working flow of our research is depicted by a flowchart below:

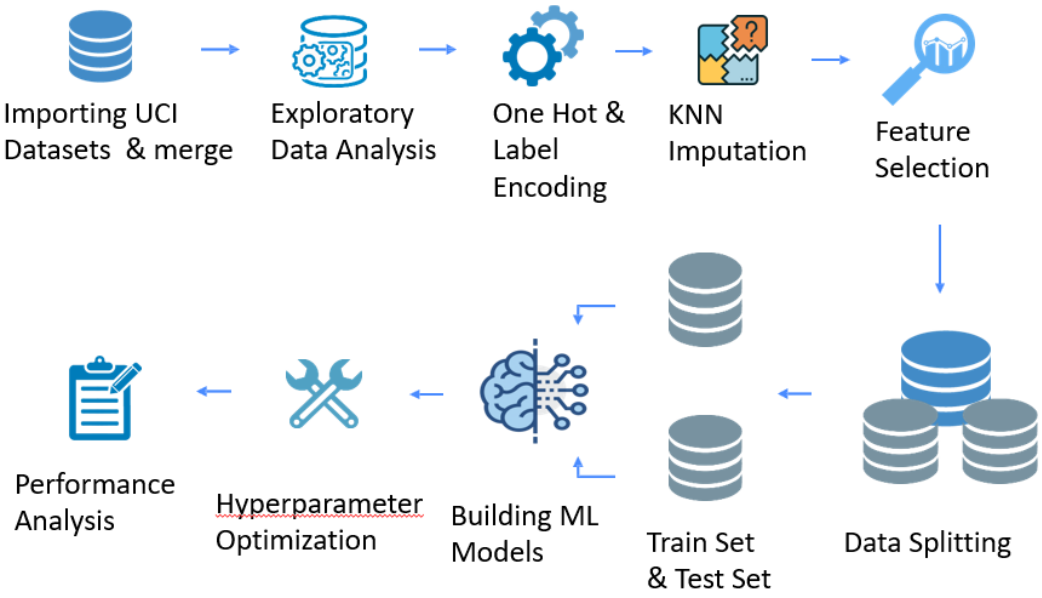


Figure 3.1: Basic Methodology Flowchart

3.2 Description of Basic Methodology

The first step was to obtain dataset available in the UCI Machine Learning repository. Then we preprocessed the datasets and sequentially merge them. The preprocessed dataset contains 1100 instances merged from three datasets, with 104 adolescents, 704 adults, and the remainder being child instances. The following part will include the details of the dataset and its description. A Python Notebook of Anaconda Navigator was used to process the dataset after which it was uploaded to a Machine Learning and Deep Learning platform known as "Jupyter Notebook." The dataset was subjected to Exploratory Data Analysis in order to create a visual representation of it. It was necessary to perform data preprocessing and feature extraction in order to make the dataset suitable for the Machine Learning models. One hot encoding was implied to age description, which is a categorical feature because most machine learning

algorithms cannot deal with categorical information directly and must be converted to a numerical format for easy access. Some instances with missing values are filled using the median method, where label encoding was used for categorical information, and the null values were filled using KNN imputer. After that, the data was divided into two groups: train sets and test sets followed by applying cross validation and selection of hyper-parameters.. Following that, a specific machine learning algorithm was chosen to be used in the model's implementation. The optimization of hyper-parameters was carried out in order to make the models more efficient and compact. Afterwards, the model was trained using the training dataset, and the results were later tested using the test dataset. Confusion matrices for each situation were depicted. The next step was to evaluate the performance of each model using several performance matrices such as accuracy, precision, specificity, sensitivity, cross validation score, recall, error-rate and F-1 score, among others. Finally, different performance parameters were observed and best performing model was proposed.

3.3 Data Description

This study involved applying our algorithms to the Autism Spectrum Disorder dataset from the University of California Irvine (UCI) machine learning repository, which contained a total of 1100 instances that had been amalgamated from three datasets, with 104 instances being teenagers, 704 instances being adults, and the remaining instances being children. The University of California, Irvine dataset repository is one of the most widely used and dependable datasets for applying machine learning algorithms in the world. Table I contains a list of all 22 qualities and the information about each of them.

Attribute	**Type**	**Description**
Age	Number	years
Gender	String	Male or Female
Ethnicity	String	List of common ethnicities in text format
Born with jaundice	Boolean (yes or no)	Whether the case was born with jaundice
Family member with PDD	Boolean (yes or no)	Whether any immediate family member has a PDD
Who is completing the test	String	Parent, self, caregiver, medical staff, clinician ,etc.
Country of residence	String	List of countries in text format
Used the screening app before	Boolean (yes or no)	Whether the user has used a screening app

Screening Method Type	Integer (0,1,2,3)	The type of screening methods chosen based on age category (0=toddler, 1=child, 2= adolescent, 3= adult)
Question 1 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 2 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 3 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 4 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 5 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 6 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 7 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 8 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 9 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Question 10 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Screening Score	Integer	The final score obtained based on the scoring algorithm of the screening method used. This was computed in an automated manner

Table 3.2: Attribute Information

3.4 Data Preprocessing

Data preprocessing is a critical component of data mining operations. Noise, missing values, and possibly an unusable format are all common features of real-world data, which makes it unsuitable for use in machine learning models when they are directly applied. Data preprocessing is a necessary task for cleaning data and preparing it for use by a machine learning model. It also helps to improve the accuracy and efficiency of a machine learning model by increasing the amount of data available. As part of machine learning, the data from big datasets is changed and encoded so that the machine can more easily compile it. For example, the data and images are converted into zeros (0) and ones (1) so that the computer can more easily compile it. In a dataset, there are primarily two sorts of information. The main two types of datasets are

i. Numerical data: These data are the kind of information that can be expressed in numerical form. Consider the following examples: age, birth year, any type of amount, and so on.

ii. Categorical Data: Data whose values are derived from a specific set of values are referred to as extracted values. For instance, Monday, Thursday, and Boolean expressions are examples (true, false).

Both types of data are subjected to a number of stages in data preprocessing before being used. The steps are in the following:

1. Data Quality Assessment: Machine learning models perform badly when dealing with poor quality data; they do not produce accurate predictions, classification, or regression results when dealing with poor quality data. As a result, the data quality will have a considerable impact on the outcome that various machine learning algorithms will produce. Following are the types of information that will be evaluated and provided in this step.
    - Missing values
    - Duplicate values
    - Inconsistent values

2. Feature Assemble: It is in this step that the seemingly random and unstructured data will be put into perspective in order to identify patterns and categorize the information. This will reduce the amount of memory storage required while also speeding up the machine learning algorithm.
3. Feature Sampling: In the field of machine learning, we frequently have to deal with a very large dataset that contains hundreds or thousands of instances of the same thing. The amount of information included inside a dataset is directly proportional to its size. However, working with such large datasets is both expensive and time-consuming. Instead, we can obtain sufficient results by incorporating only a fraction of the dataset into the study, which can save both time and cost while still producing satisfactory results. Feature sampling is the term used to describe this process. This helps the machine learning model to adapt more rapidly and accurately as a result of the data collected.
4. Reduction of dimensionality: It is the decrease of the number of input variables in a dataset that is referred to as "dimensionality reduction." It is used in domains that

deal with huge numbers of observations, such as digital signal processing, speech recognition, and bioinformatics, to reduce the amount of observations.

5. Feature Encoding: Some categorical entries in the dataset are present at times, such as true/false, yes/no, and so on. Machine learning models, on the other hand, can only function with numerical data. In order to achieve this, it is important to convert the category values into numerical values. Feature encoding is the term used to describe this process.

Data must be accurate and well-organized in order for machine learning algorithms to be successfully implemented. Initial analysis revealed that this data collection contained a number of missing values, which made it difficult to come up with a reasonable conclusion. As a result, various data frames were created in order to implement the algorithms and examine the outcomes. Initially, the median approach is used to fill in some cases with missing values. This method makes use of label encoding to fill in categorical information, and KNN imputer is utilized to fill in null values. Due to the fact that most machine learning algorithms cannot deal with categorical information directly and must be translated to a numerical format for easier access, one hot encoding implied a categorical component to age description.

3.5 Correlation Heatmap

Typically, correlation heatmaps depict a 2D correlation matrix between two discrete dimensions, with colored cells used to represent data from a monochromatic scale, as opposed to a 2D correlation matrix between two continuous dimensions. The values of the first dimension appear as rows in a table, whereas the values of the second dimension appear as columns in a table. The hue of the cell is proportional to the number of measurements that have the same dimensional value as the dimensional value in question. It is for this reason that correlation heatmaps are useful for data analysis since they make patterns easily readable while also highlighting the contrasts and variance within the same data. A correlation heatmap, like a standard heatmap, is aided by a colorbar, which makes the data more accessible and understandable to the viewer. orrelation heat-map is displayed in the attached figure. Coding framework of **Jupyter notebook** 6.0.3 from Anaconda navigator. Highly **correlated features** have been found from correlation matrix.

Finally, **top fifteen** features were calculated utilizing **k-best algorithm.**

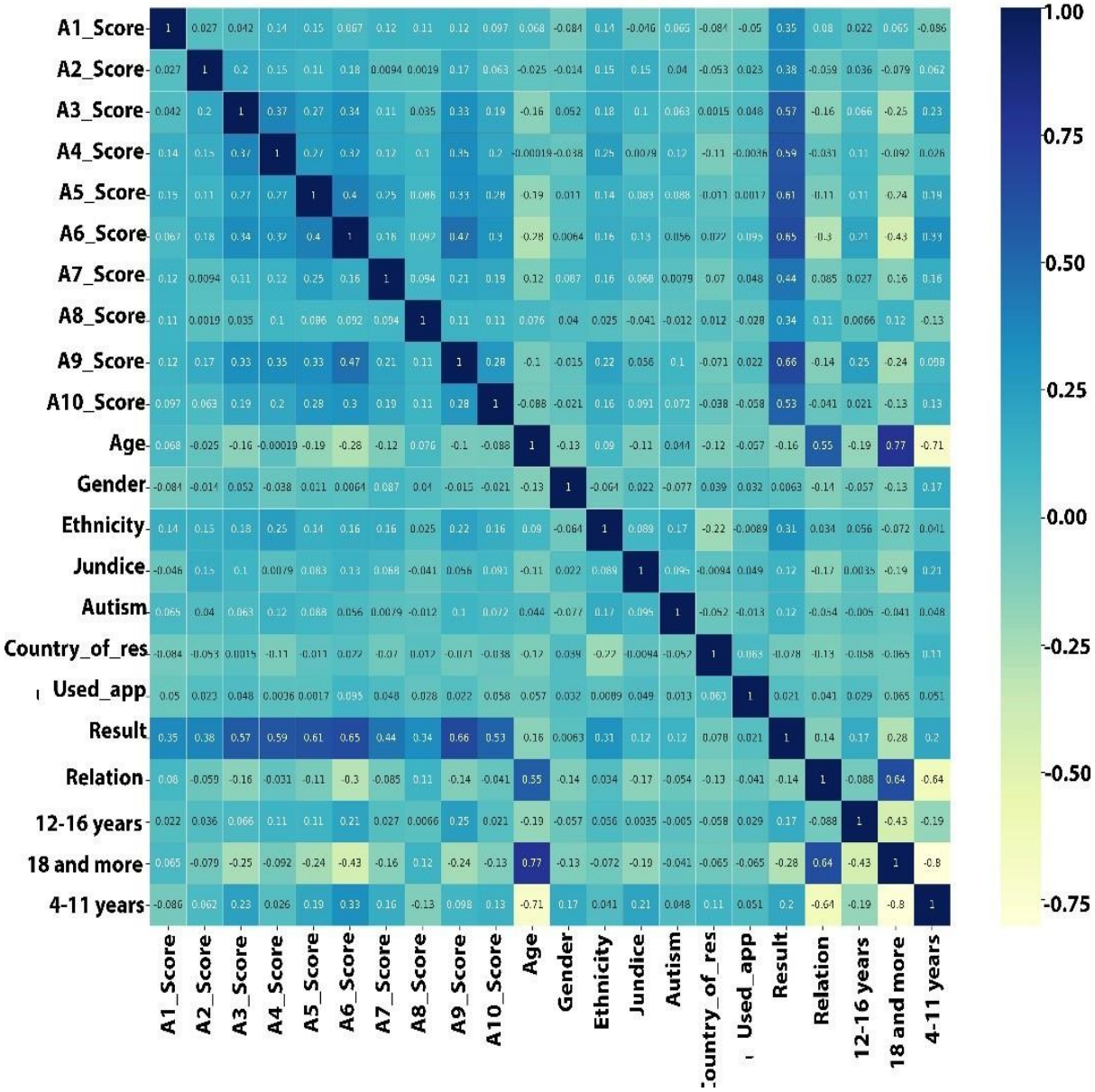


Figure 3.3: Correlation Heatmap

3.6 Hyper Parameter Tuning

There were eight machine learning algorithms built in order to predict whether a patient will have Autism Spectrum Disorder or would not have Autism Spectrum Disorder. First and foremost, algorithms were implemented with the default hyper parameter settings in place. It was determined how accurate the system was, as well as other performance measures. Optimizing the hyper parameter tweaking of machine learning algorithms is critical for

obtaining improved prediction results. Hyper parameter adjustment can be accomplished in one of two ways. They are described in the following

- GridsearchCV
- RandomizedsearchCV

GridsearchCV: Create a grid consisting of hyper - parameters and the various values assigned to them. A model is trained on the validation set for each conceivable combination, and a score is calculated for each combination. Through the use of this strategy, every single combination of the supplied potential hyper parameter values is tried at least once. While the approach performs a thorough sweep of all possible combinations, it can be wasteful in terms of training time and expense due to the vast sweep required. GridsearchCV is a technique that is effective for identifying the parameters in supervised learning algorithms and for making the model more generic in general. At the time of running GridsearchCV, each and every conceivable combination of the parameters of the model is tested. After that, the optimum set of parameters is selected.

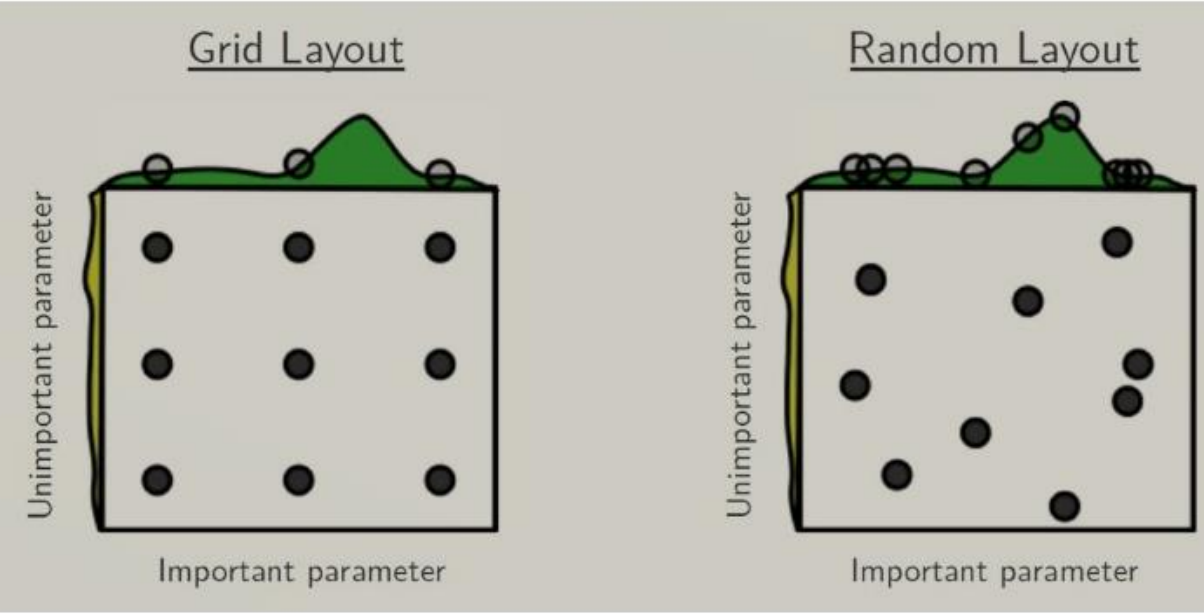


Figure 3.4: Grid Layout & Random Layout

RandomizedsearchCV: It is similar to grid search, except that instead of training and scoring on every potential hyperparameter combination, random hyperparameter combinations are chosen. You have the option of adjusting the number of search iterations based on your time and resource limitations. RandomizedsearchCV is a tuning strategy for hyperparameters in which random combinations of the hyper - parameters are used to find the best fit for the model.

RandomizedsearchCV is a technique for tuning hyperparameters that is used to find the best fit for the model. The settings are chosen at random from a large pool of options. In situations where there are many parameters to test and training time is a problem, RandomizedsearchCV is extremely effective.

GridsearchCV performs admirably when only a modest number of hyperparameters are present. Alternatively, if the number of parameters is large and calculation time is an issue, it is preferable to use the RandomzedsearchCV algorithm.

The GridsearchCV technique was used to fine-tune the hyperparameters in this study. Then, in order to compare the results with the default hyperparameters, performance metrics were computed in the appropriate manner.

3.7 All about Missing Values

Despite the abundance of MOOCs and other online tools, there are still skill shortages in addressing specific data problems. Managing missing data in real-world datasets is one example. Beginner students may take this issue lightly, but they are not at fault. Due to the availability of small, manageable toy datasets, the complexity of data missingness concerns has been grossly underestimated, despite the urgency of the issue. Many novice data scientists stop with simple mean, median, and mode imputation. Although these strategies may adequate for simple datasets, they are insufficient for addressing missing data in large datasets.

Despite the fact that each situation is unique, missing-ness can be broadly classified into three categories.

MCAR (Missing Completely At Random): This is an actual instance of data being lost at random. For example, abrupt mistakes in data input, temporary sensor failures, or generally missing data that is not related with any external factor are all examples of what I mean. The percentage of people that are absent is minimal.

MAR (**Missing At Random):** This is an expanded instance of MCAR. Even though missing data may appear random at first glance, it will have a regular relationship with other observable patterns, such as data missing from observational equipment during scheduled maintenance breaks. The quantity of null values can vary.

MNAR (Missing Not At Random): There may be a huge number of missing values, and the reasons for the missing-ness are often related with things that are beyond our control or awareness.

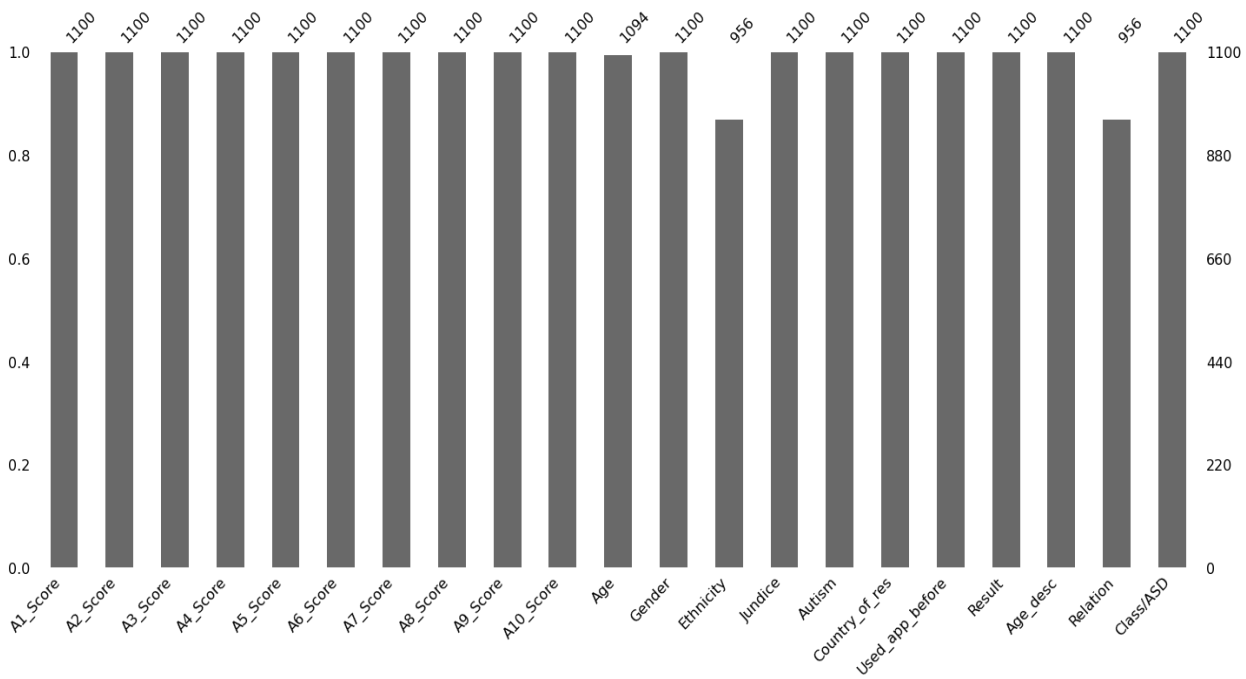


Figure 3.5: Feature Counts

There are three features with varying shares of missing values. The first step in determining the missingness type is to create a missingness matrix.

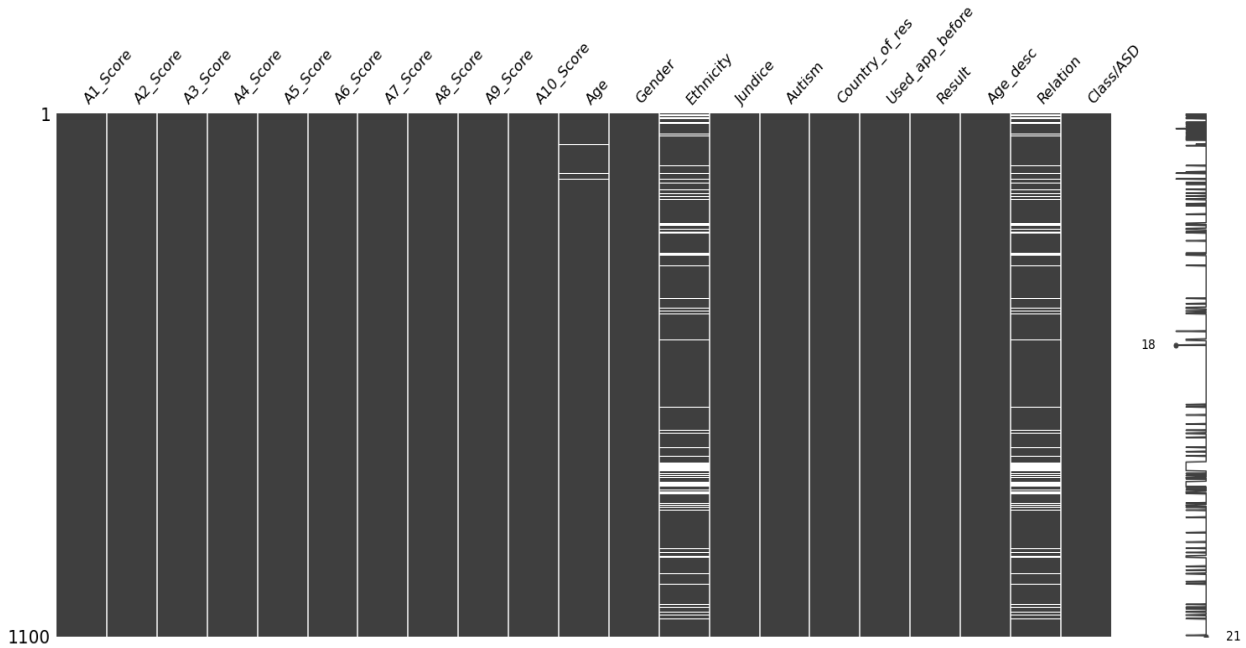


Figure 3.6: missingness matrix

This matrix illustrates the distribution of nulls throughout the sample. Where missing values exist is represented by white segments or lines. Age column can be termed MCAR for two reasons:

- The fraction of values that are missing is insignificant.
- The dataset has a large number of missing values that are distributed totally at random.

Ethnicity and Relationship, on the other hand, are both lacking a significant number of data points. So, is there a connection between their absence and something else? MSNO provides a missingness heatmap that illustrates the relationship between missingness and correlation.

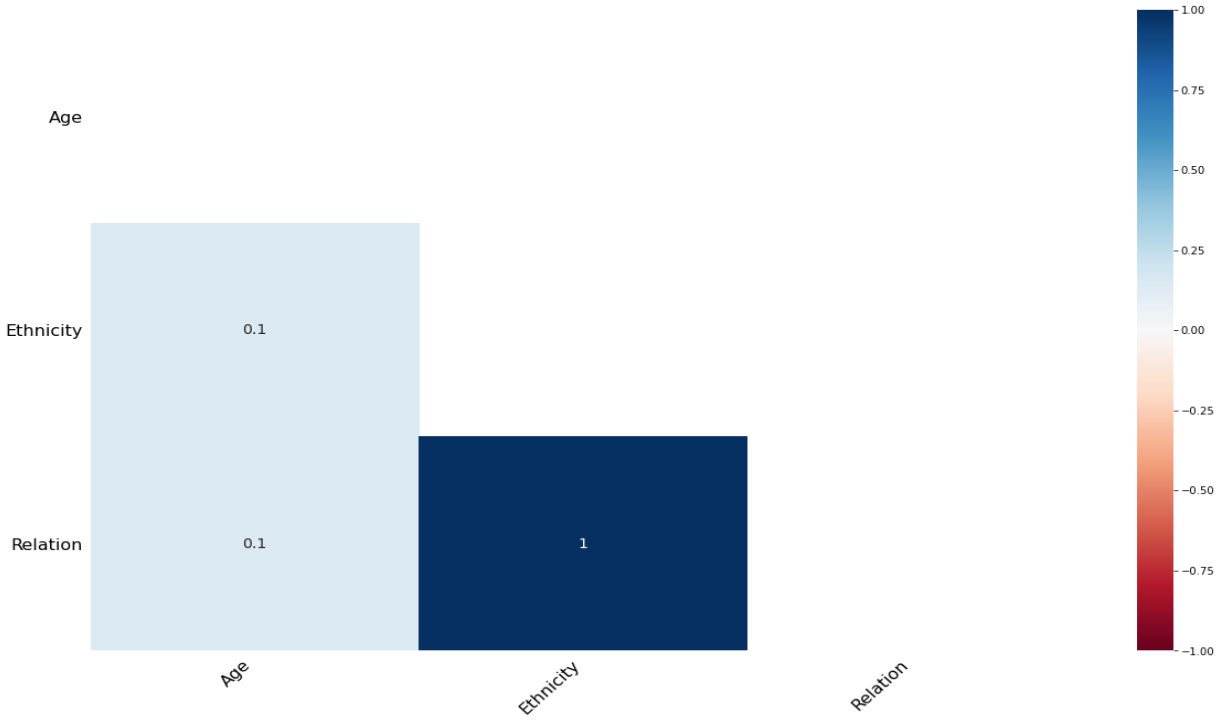


Figure 3.7: Missingness heatmap

It is also possible to see the relationship between missing characteristics and the dendogram by charting it.

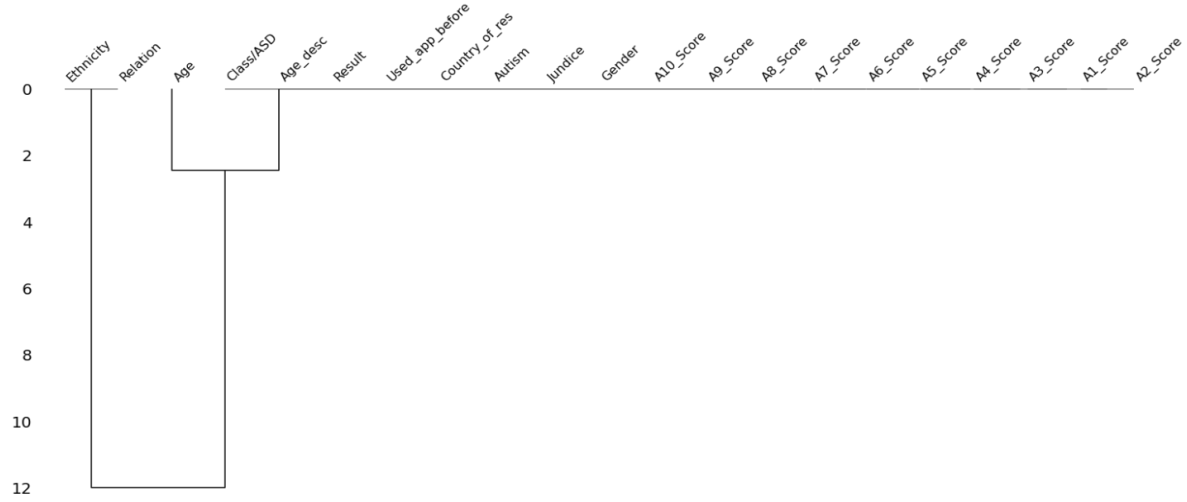


Figure 3.8: Dendogram

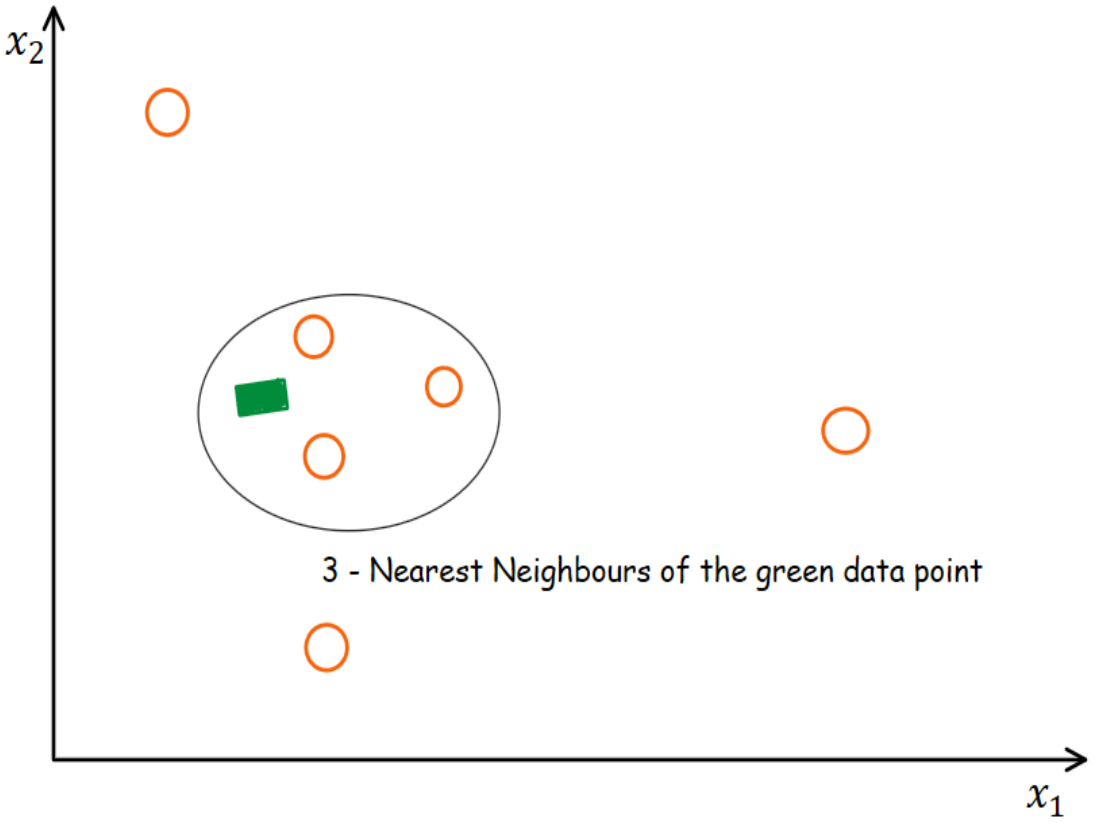
3.7.1 KNN Imputer

In today's environment, data is collected from a variety of sources for purposes such as analysis, insight development, theory validation, and other purposes. When data is obtained from many sources, it is common for some information to be lost. Human error or a flaw with the data gathering or extraction procedure could be to blame for this. In data preparation, dealing with these missing values becomes a critical stage in the process. The selection of an interpolation method is critical since it has a substantial impact on the final result of the task. The knn imputer from the scikit learn company is a missing value interpolation method that is extensively used. Compared to standard interpolation approaches, it is often regarded as a superior alternative. In this project, we present a method for filling missing values in a dataset with observations from adjacent data points that is based on prior research. This is accomplished through the use of the knn imputer implementation.

When it comes to imputation of missing data, the univariate technique is a straightforward method of guessing values that may or may not yield accurate results in all cases. When we have variables such as vehicle density on the road and pollutant levels in the air, and there are only a few observations on pollutant levels, it may not be a good strategy to estimate pollutant levels using average or median pollutant levels. k-nearest neighbor (KNN) algorithms, for example, can assist in interpolating the values of missing data in this situation. Sociologists and community researchers believe that people reside in communities due to a sense of security, attachment to the community, and interpersonal relationships that establish community identity through involvement in various activities.

Calculation of distance in the case of missing data

k-nearest neighbor (KNN) is a comparable interpolation approach for data that finds nearby points by measuring distance and predicts missing values using the entire values of neighboring observations. Typically, Euclidean distance is used to identify neighboring points in a dataset.



Consider an example to illustrate this point. Consider the observations (2, 0), (2, 2) in two-dimensional space (3, 3). The following diagram illustrates these points:

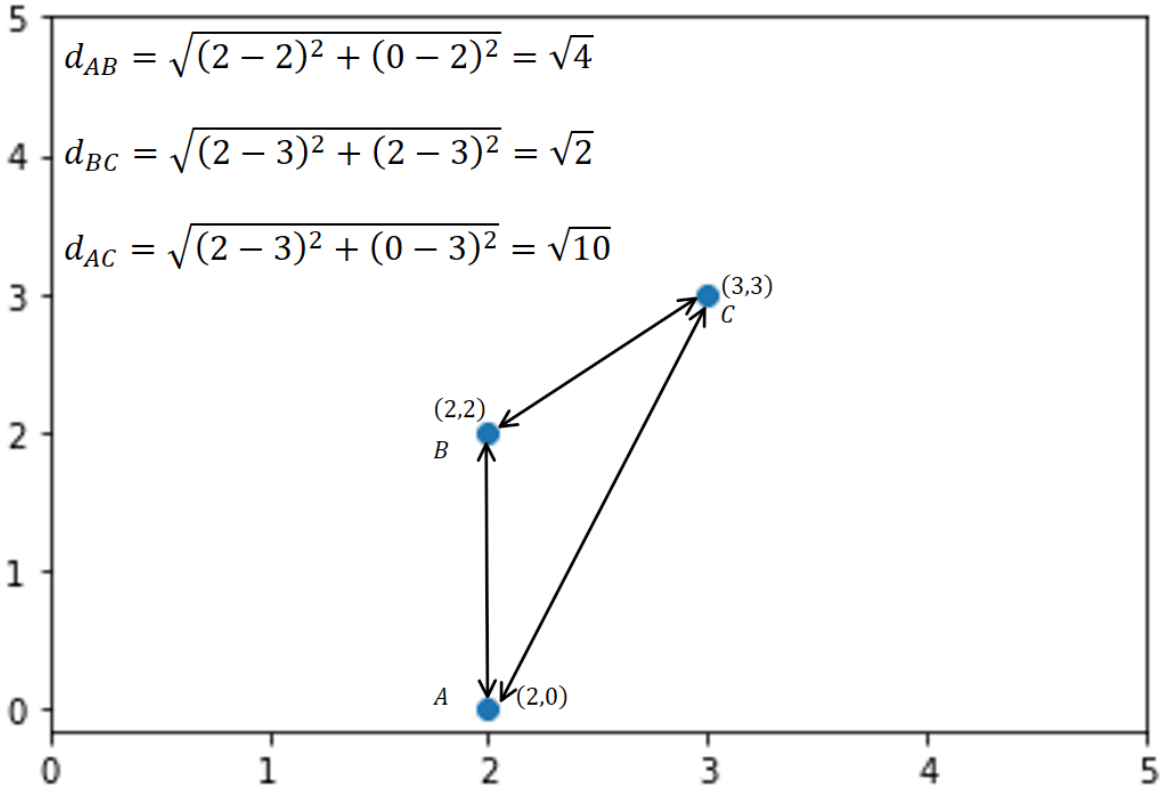


Figure 3.9: Distance Calculation

The point with the least Euclidean distance is considered the closest neighbor. For instance, the nearest neighbor of point a to point 1 is point B. The nearest neighbor of 1 for point B is point C. In the case of missing coordinates, the Euclidean distance is computed by discarding the missing values and increasing the weight of the remaining data where:

$$d_{xy} = \sqrt{weight * squared\ distance\ from\ present\ coordinates}$$

and

$$weight = \frac{Total\ number\ of\ coordinates}{Number\ of\ present\ coordinates}$$

The research also demonstrates that, after varying the K values for interpolation, it is vital to validate the model using cross validation to ensure that it is correct. Despite the fact that missing value interpolation is a relatively new research subject, KNN is a straightforward and successful method.

3.7.2 Synthetic Minority Over-Sampling Technique

A technique to the creation of classifiers using unbalanced datasets is presented in detail. A dataset is unbalanced if the categorization categories are not nearly evenly represented across all of the data points. Real-world data sets are frequently constituted primarily of "regular" cases, with just a tiny fraction of "strange" or "interesting" occurrences. It is also true that the cost of misclassifying an atypical (interesting) sample as a normal example is frequently far larger than the cost of the reverse mistake. Under-sampling of the majority (normal) class has been presented as a useful method of boosting the sensitivity of a classifier to the minority class. This research demonstrates that a combination of our strategy of over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can result in superior classifier performance (in ROC space) than simply under-sampling the majority class. This research also demonstrates that a combination of our strategy of over-sampling the minority class and under-sampling the majority class may yield superior classifier performance (in ROC space) than altering the loss ratios in Ripper or class priors in Naive Bayes. Our strategy of over-sampling the minority class entails the creation of synthetic minority class cases. Experiments are carried out with the help of C4.5, Ripper, and a Naive Bayes classifier. The area under the receiver operating characteristic curve (AUC) and the ROC convex hull algorithm are used to evaluate the procedure.

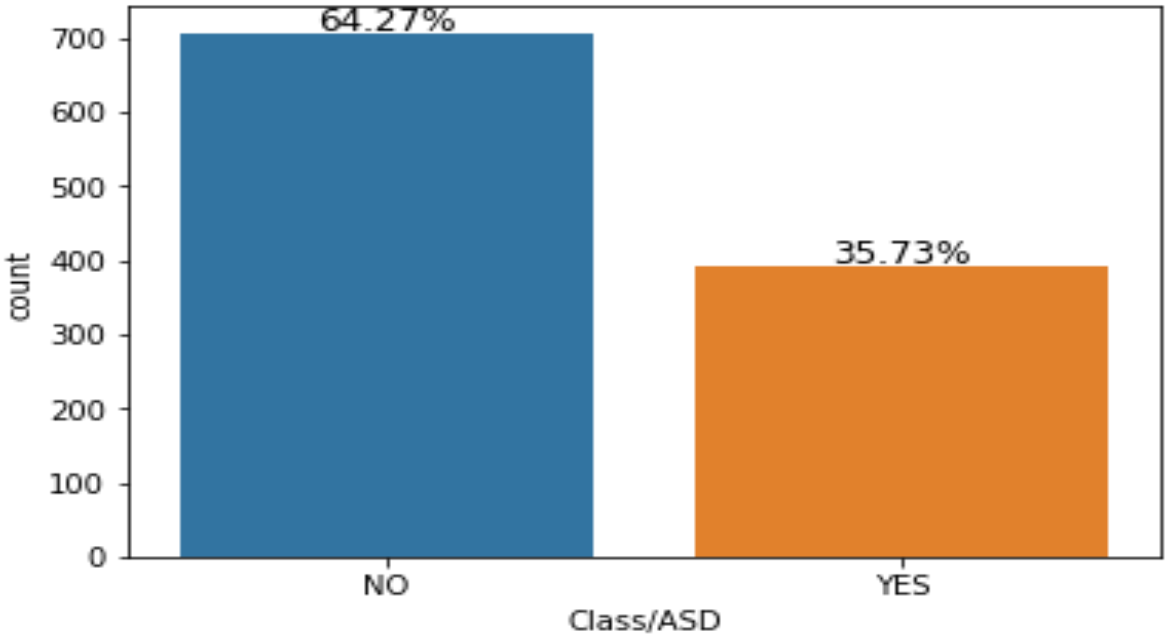


Figure 3.10: Imbalance Dataset

Because our dataset was initially unbalanced, we used the smote oversampling strategy to improve the effectiveness of the model. The results were promising.

Chapter 4

Introduction to Algorithms

In Artificial Intelligence, machine learning is a subset of the study of algorithms that work and adapt to diverse settings as a result of their experience. A collection of data, referred to as training data, is used to gain experience during the training process. Machine learning algorithms detect or classify data without being specifically taught to do so after they have gone through the training process. Eight supervised classification learning algorithms for the identification of Autism Spectrum Disorder are examined in this research, and the results of these algorithms are briefly compared under various criteria.

4.1 K-Nearest Neighbors

The K-Nearest Neighbors technique is one of the most straightforward and widely used supervised ML algorithms. The K-NN algorithm assumes that the new data and existing cases are comparable, and it places the new instance in the category that is the most similar to the existing categories, rather than the most similar to the new data. Technically, it does not train any dataset; rather, it predicts that an observation will fall into the class that has the greatest proportion of k-nearest neighbors surrounding it.

K-nearest neighbors (KNN) is a supervised technique that may be used for both regression and classification problems. It is characterized as being typically basic, nonparametric, and sluggish. Because it works on the basis of distance measurement, it has the simplest functioning technique available.

Step 1: First, the training and test datasets are put into the system.

Step-2: Then the data points with the closest 'k' (any integer) number of data points are selected.

Step-3: Following that, the following operations are carried out on each and every set of data in the test dataset:

i) It is necessary to measure the distance between each row of training data points and each row of test data points.

ii) The test data points are sorted in ascending order based on the distance between the two points observed.

iii) In the next step, upper K rows of the sorted array are selected in order to assign a class to the test data point based on the most bright class of the sorted rows. This process is repeated until the test data point is assigned a class that corresponds to the most vibrant class of the sorted rows.

Step-4: The algorithm has reached its conclusion.

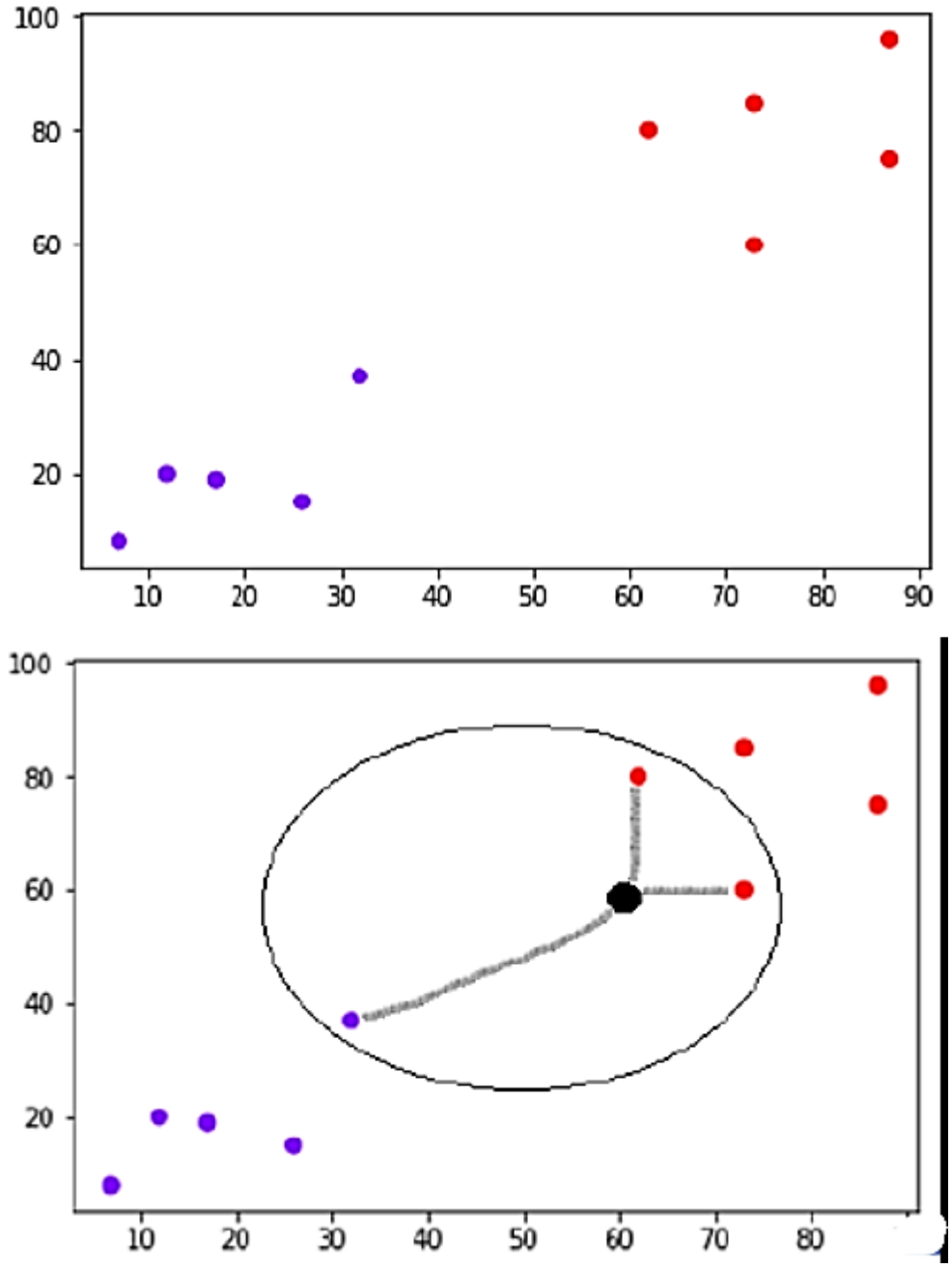


Figure 4.1: KNN Architecture

It is important to choose the value of K in such a method that the smallest amount of errors is encountered while creating correct predictions. For the purpose of determining similarity, distance is assumed to be a metric; that is, the data point that is the closest to the point under

inspection can be deemed as being the most similar to the point under observation. There are many different types of distance measures to choose from.

Distance Functions:

i) Euclidean distance:

$$\sqrt{\sum_{j=1}^{k}(p_j - q_j)^2}$$

ii) Manhattan distance:

$$\sum_{j=1}^{k}|p_j - q_j|$$

iii) Minkowski distance:

$$\left(\sum_{j=1}^{k}(|p_j - q_j|)^n\right)^{\frac{1}{n}}$$

iv) Hamming distance:

$$D_H = \sum_{j=1}^{k}|p_j - q_j|$$

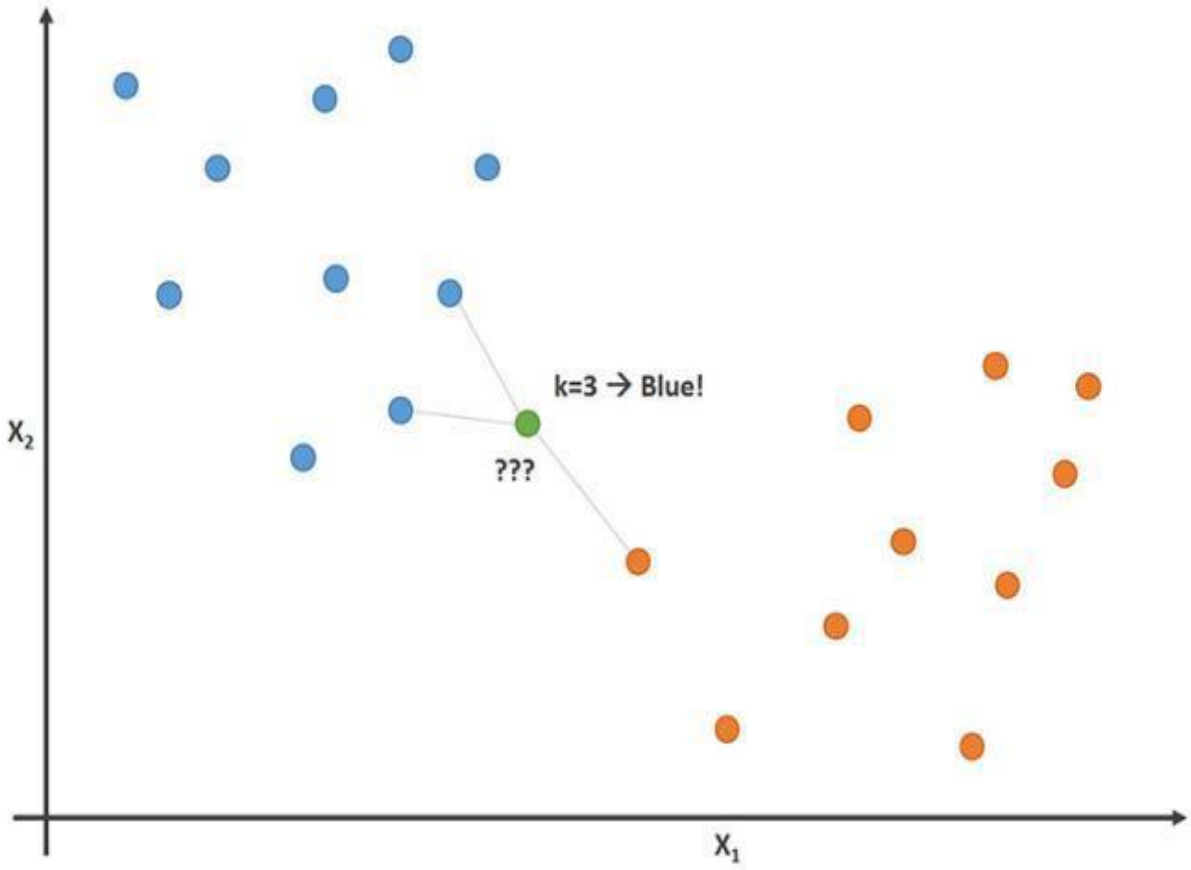
$$p = q \Rightarrow D = 0$$

$$p \neq q \Rightarrow D = 1$$

e.g.:

p	q	Distance
Apple	Apple	0
Apple	Orange	1

The KNN algorithm is depicted in the following illustration. This is how KNN algorithm mainly works.



4.2 Extra Tree Classifier

It is a form of ensemble learning technique that aggregates the outcomes of several de-correlated decision trees accumulated in a "forest" to get a classification result. Extremely Randomized Trees Classifier (Extra Trees Classifier) is one example of this type of ensemble learning technique. In terms of idea, it is nearly identical to a Random Forest Classifier, and the only difference between them is in the method by which the decision tree algorithm in the forest are constructed.

Thus every Decision Tree in the Extra Trees Forest is derived from the training sample that was used to construct it. Then, at each test node, each tree is presented with a random sample of k features from the feature-set, from which each decision tree must determine the best feature to divide the data based on some mathematical criteria, and the process is repeated until the data is split (typically the Gini Index). Due to the random sampling of features, numerous de-correlated decision trees are generated as a result of this process.

In order to perform feature selection by using this forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in

the decision of feature of split (or the Gini Index if the Gini Index is used in the construction of the forest) is computed (or the Gini Index if the Gini Index is used in the construction of the forest) is calculated. The Gini Importance of the feature is the value assigned to this attribute. Each feature is sorted in decreasing order according to its Gini Importance, and the user picks the top k features based on his or her preference from among those sorted in descending order.

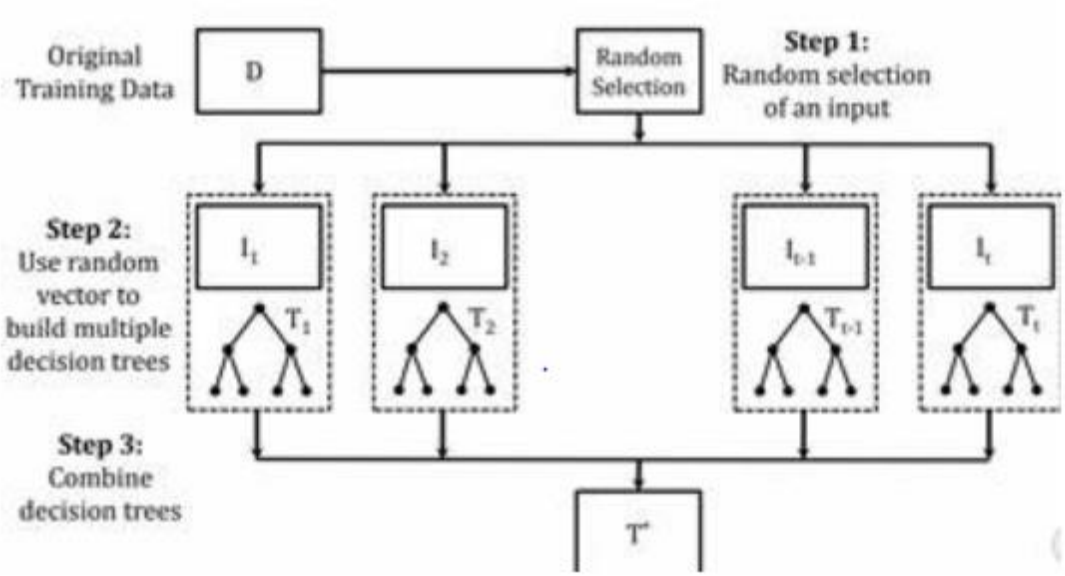


Figure 4.2: Visual Representation of Extra Trees Classifier[26]

4.3 Gaussian Naïve Bayes Classifier (GNB)

With high independence assumptions, Gaussian Naive Bayes is a probabilistic classification algorithm that applies Bayes' theorem to a large number of observations. Independence refers to the idea that the existence of one value of a characteristic does not influence the presence of another value of that feature in the classification process (unlike independence in probability theory). The term "naive" relates to the usage of the premise that the characteristics of an object are independent of each other. While Naive Bayes classifiers are considered to be highly expressive, flexible, and reasonably accurate in the domain of machine learning, their performance degrades fast as the size of the training set increases. The effectiveness of Naive Bayes classifiers is attributed to a variety of different characteristics. Most importantly, they do not necessitate any fine-tuning of the classification model's parameters, they scale very well

with size of the training data set, and they are capable of handling continuous features with ease.

The formula for Bayes Theorem can be written as

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A).P(B|A)}{P(B)}$$

Where, The probability of A occurring = P(A)

The probability of B occurring = P(B)

The probability of A given B = P(A|B)

The probability of B given A = P(B|A)

The probability of both A & B occurring = P(A∩B)

A common assumption when working with continuous data is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution, which is not always the case. It is presumed that the features have a high probability of occurring.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}\right)$$

Sometimes assume variance— is independent of Y (i.e.σ_i), or independent of Xi (i.e.σ_k) or both (i.e.σ).

Gaussian Naive Bayes accommodates continuous valued features & models each as if it were a Gaussian (normal) distribution. A simple model can be constructed by assuming that the data is characterized by a Gaussian distribution with no covariance (i.e., independent dimensions) between the various parameters. To fit this model, all that is required is calculating the mean and standard deviation of the points within each label, this is all that is required to establish a normal distribution of this type.

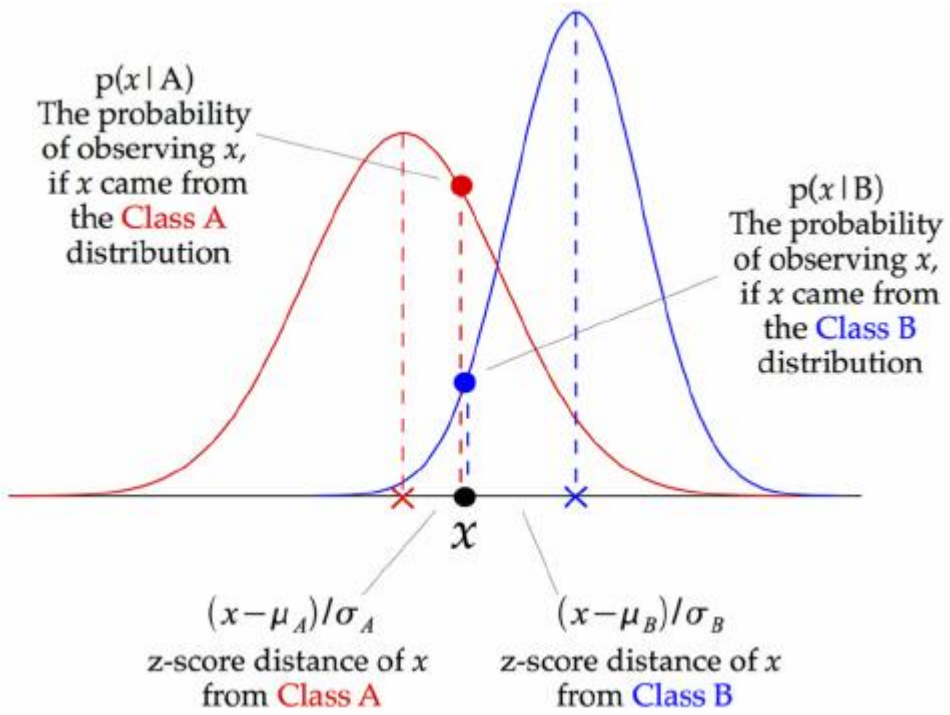


Figure 4.3: Illustration of how a Gaussian Naive Bayes (GNB) classifier works.

The given figure shows how a Gaussian Naive Bayes (GNB) classifier operates in practice. It is determined at each data point how far it is from each class mean, which is defined as the distance from the class mean multiplied by the standard deviation of each class.

4.4 Bernoulli Naïve Bayes Classifier (BNB)

Bernoulli Naive Bayes is a version of the Naive Bayes algorithm. If the data consists solely of binary characteristics and is distributed according to the Bernoulli distribution, this is the algorithm to employ. In particular, the primary characteristic of Bernoulli Naive Bayes is that it only accepts binary values for characteristics such as true or false, yes or no, success or failure, 0 or 1, and so on. As a result, when the feature values are binary, we know that we must apply the Bernoulli Naive Bayes classifier to classify the data.

Because we are dealing with binary data, let us assume 'p' to represent the likelihood of success and 'q' to represent the probability of failure, with q=1-p.

For a random variable 'X' with a Bernoulli distribution, consider the following:

The Bernoulli Distribution

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

where 'x' can only have one of two possible values: 0 or 1.

The Bernoulli Naive Bayes Classifier is based on the rule that goes as follows:

$$P(x_i|y) = P(i|y)x_i + \big(1 - P(i|y)\big)(1 - x_i)$$

4.5 Multinomial Naïve Bayes Classifier (MNB)

Multinomial Naive Bayes is amongst the most popular supervised learning classifiers that is used for the analysis of categorical text data. It is one of the most popular supervised learning classifications. Known as the Multinomial Naive Bayes method, it is a Bayesian learning approach that is commonly used in Natural Language Processing (NLP). A text, such as an email or a newspaper item, is tagged by the algorithm, which uses the Bayes theorem to guess what it is. For a given sample, it calculates the likelihood of each tag appearing and outputs the tag with the highest likelihood. It is made up of a collection of algorithms that all have one thing in common: each item being classified is completely independent to any other feature in the dataset. The presence or absence of a feature has no effect on whether or not another characteristic is included or excluded.

It is straightforward to put into practice because all we have to do is calculate probability. There are no restrictions on the type of data that can be used using this strategy. It's simple to use and may be applied to real-time apps to forecast their needs. It is extremely scalable and is capable of dealing with massive datasets with ease.

The accuracy of this algorithm's prediction is lower than those of other probability algorithms in general. It is not acceptable for use in regressive situations. Unlike other classification techniques, the Naive Bayes technique cannot be used to predict numerical values and can only be used to classify textual input.

An event model that uses a multinomial distribution represents the frequencies with which specific events have been generated by a multinomial(p1,....., pn), where pi is the chance that event I occurs and pn is the probability that event I does not occur (or K such multinomials in the multiclass case). A feature vector X=(x1,...,xn) is then represented as a histogram, with xi representing the number of times event I was seen in a given instance. In document categorization, this is the event model that is most commonly used, with events indicating the

occurrence of a word in a single document, for example. The probability of viewing a histogram x can be calculated as

$$P(x|C_k)= \frac{(\sum_i^n x_i)!}{\prod_i^n x_i!} \prod_i^n p_{ki}^{x_i}$$

4.6 Discriminant Analysis Algorithm

Discriminant analysis methods, particularly linear and quadratic, are the two basic classifiers with a linear and nonlinear supervisory aspect, respectively, that are based on discriminant analysis techniques. These algorithms are significant because they are innately multiclass, and they have been demonstrated to work effectively in reality by providing easily calculated solutions to the problems[27].

4.6.1 Linear Discriminant Analysis (LDA):

It is also known as Linear Discriminant Analysis, Normal Discriminant Analysis, or Discriminant Function Analysis, and it is a dimensionality reduction technique that is often used for supervised classification tasks, among other things. It is used to model distinctions between groups, i.e. to distinguish between two or more classes. It is also used to project the features of a higher-dimensional region into a lower-dimensional space in order to make them more visible.

For example, we have two classes that need to be separated as efficiently as possible. Classes can have a variety of characteristics. When only one trait is used to categorize them, there may be some overlap, as illustrated in the following image. As a result, we will continue to increase the amount of features required for proper classification.

Increasing the number of component axes for the purpose of class separation.

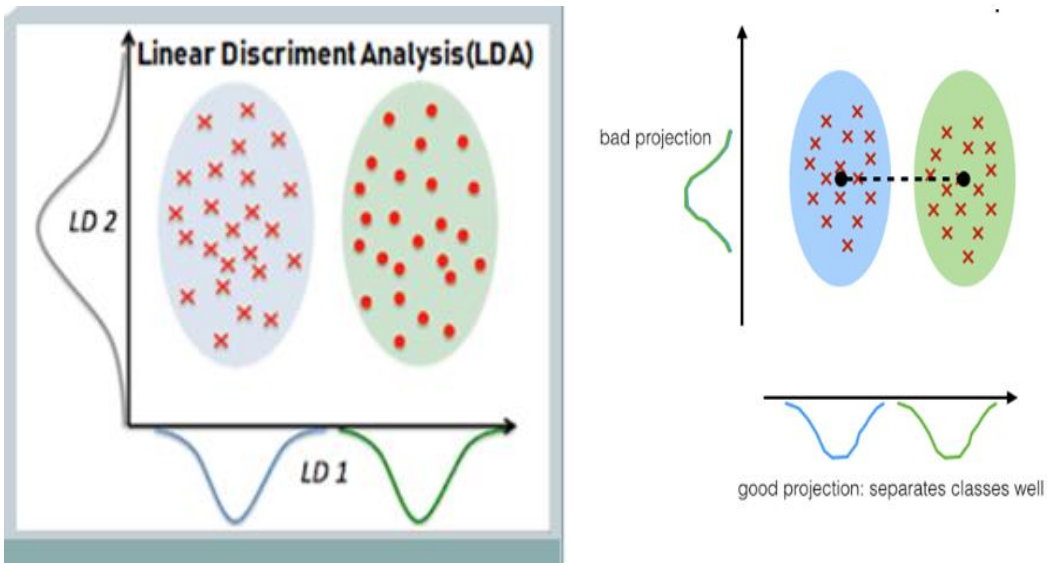


Figure 4.4: Illustration of how a Linear Discriminant Analysis (LDA) works.

It has a tendency to measure and optimize the associativity of the distinct classes, which can also be referred to as the interclass variance or the interclass matrix, among other things[27]. LDA discriminant function is denoted by the notation:

$$\partial_k(x) = 2\mu_k^T \sum_k^{-1} X - \mu_k^T \sum_k^{-1} \mu_k - 2log(\pi_k)$$

Where, X = set of measurements

μ_k = mean vector

π_k = prior probability

$\sum_k =$ covariance matrix

k = class

If the requirement or dependent variable is categorical with an interval in presence of the indicator or independent variable, the purpose of LDA is to evaluate outcomes. Thus, it is possible to construct discriminating functions that are nothing more than the linear combination of distinct variables that perfectly distinguish between the categories of the dependent variable, which are called discriminating functions. In addition, the consistency of the categorization is evaluated[28].

4.6.2 Quadratic Discriminant Analysis (QDA):

Quantitative discriminant analysis (QDA) is closely connected to linear discriminant analysis (LDA), which makes the assumption that the measurements are regularly distributed. In contrast to LDA, QDA does not make the assumption that the covariance of each of the classes is the same as the others. Quadratic discrimination necessitates more calculation and data collection than linear discrimination since the parameters necessary are more complicated to determine. If there isn't a significant difference between the group covariance matrices, the latter will perform as well as quadratic discrimination in most situations. Discrimination based on quadratic functions is the most general type of Bayesian discrimination.

LDA can only learn linear limits, whereas QDA can learn quadratic limits, and is hence more mobile, whereas LDA cannot. The QDA algorithm is a simplified variant of the LDA technique, because the measurements are typically spread among only two sets of points. QDA, on the other hand, does not take into consideration the postulation that the covariance of each of the classes is equal, as does LDA in this case. Furthermore, the surface separating the sub-spaces would be referred to as a conic area (like a hyperbola, or parabola). Now, the covariance matrix for each of the classes y can be calculated as follows:

$$\sum\nolimits_{y} = \frac{1}{N_y - 1} \sum\nolimits_{y_i = y} \left(x_i - \mu_y\right) \left(x_i - \mu_y\right)^T$$

By including the following term and completing the equation (taking log both side). The quadratic Discriminant function can be written as follows:

$$\delta_k(x) = log\pi_k - \frac{1}{2}\mu_k^T \sum\nolimits_{k}^{-1} \mu_k + x^T \sum\nolimits_{k}^{-1} \mu_k - \frac{1}{2} x^T \sum\nolimits_{k}^{-1} x - \frac{1}{2} log \left| \sum\nolimits_{k} \right|$$

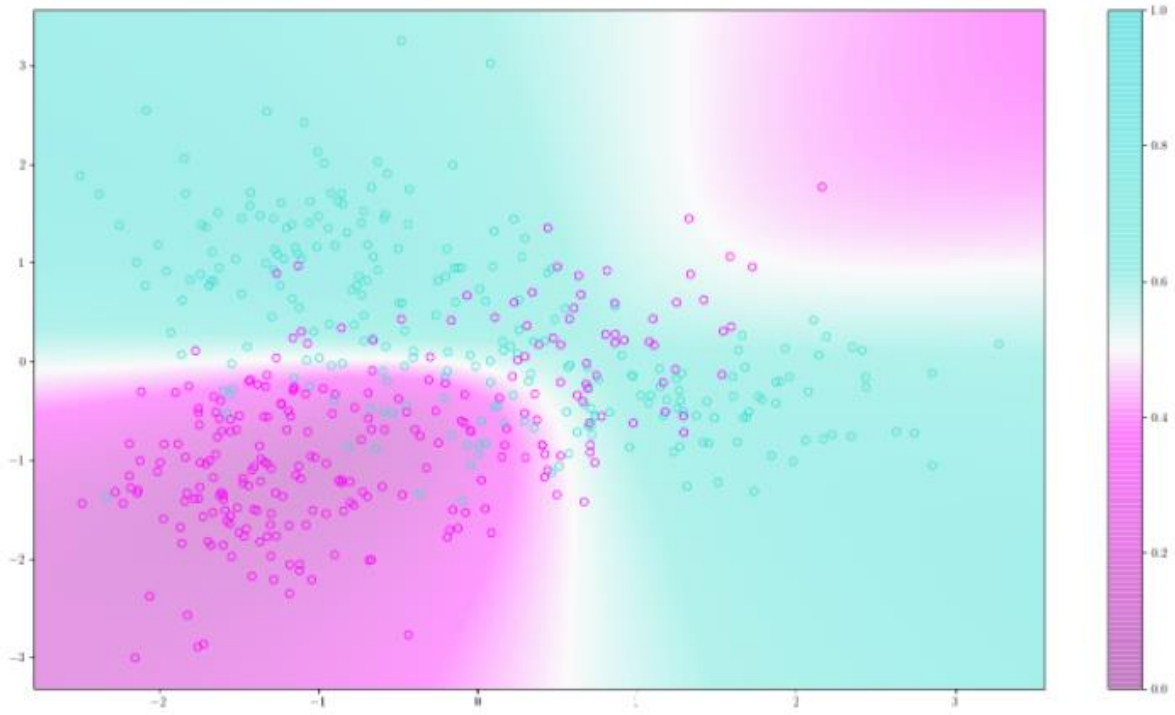


Figure 4.5: Illustration of decision boundary generated by QDA.

4.7 Ridge Classifier

According to Ridge regression method, the Ridge Classifier turns the label data into [-1, 1] and fixes the issue with the regression approach, which is based on Ridge regression method. The target class with the greatest value in the prediction is chosen, and for multiclass data, multi-output regression is used to choose the target class.

Roughly speaking, Ridge regression is a technique for estimating the coefficients of multiple regression models in situations where linearly independent variables are substantially correlated with one another. Among the various domains where it has been applied are econometrics, chemistry, and engineering, among others.

Ridge regression was developed as a potential solution to the inaccuracies of least square estimators when linear regression models contain some multi collinear (highly correlated) independent variables. This was accomplished by developing a ridge regression estimator to account for the high correlation between independent variables (RR). Due to the fact that its variance and mean square estimator are frequently smaller than the least square estimators previously computed, it provides a more precise estimate of the ridge parameters. Specifically, there are three phases in the scikit-learn implementation of the Ridge Classifier[29]:

i. Initialization Phase: It is necessary to enter the many settings that affect how the classifier operates during the Initialization phase. These settings are set when the Ridge Classifier is first created and are not changed after that. In this case, the parameters are as follows:
   - alpha: It is the regularization constant that is used to limit the variation of estimations and increase classification accuracy in the classification algorithm.
   - max iteration: This parameter specifies the number of iterations to be used by the solvers.
   - solver: The ridge classifier features a large number of built-in solutions that are employed in the data training process. The auto option selects the most appropriate solver for a given set of circumstances. A few of the solvers available are the cholskey, cholskey kernel, sparse cg, and lsqr algorithms.

ii. Fit Phase: During the fit phase, we provide the classifier with two inputs: a matrix X and a vector Y. For each row x of the matrix X, the feature vector which maps to the class y that is already in the corresponding element in the vector Y is represented by the feature vector in the vector Y. In order to learn a model from this data, the Classifier develops a coefficient vector that is the best fit for all of the information. It operates on a linear model, in which the coefficient vector is also used to depict the coefficients of a linear equation, in which the elements of the feature vector x are the variables, and the Ridge Classifier operates on this model. When it comes to classification, the scikit-learn implementation takes a one-versus-all approach. In order to accomplish this, the classifier generates separate vectors Y for each class in the original result vector, where it selects one of the classes in the original result vector as the base class and maps it to the value '1', while mapping all other classes to the value '-1', and so on. Multinomial classification is reduced to a series of independent binomial classifications in this manner.

iii. Predict Phase: During this phase, the matrix is sent to the classifier, and it is the classifier's responsibility to generate the classes that correspond to each row of the matrix during the previous phase. It accomplishes this by making use of the coefficient matrix that was formed during the Fit step of the process. The vectorizer matrix pertaining to each test case is multiplied by a scalar matrix consisting of weights associated with the various attributes in each test case. The length of the

feature vector determines the size of both the matrix and the feature vector. The scalar product yields a value for each class based on the weight vectors used in the calculation. The highest possible value among them is assigned to the appropriate class.

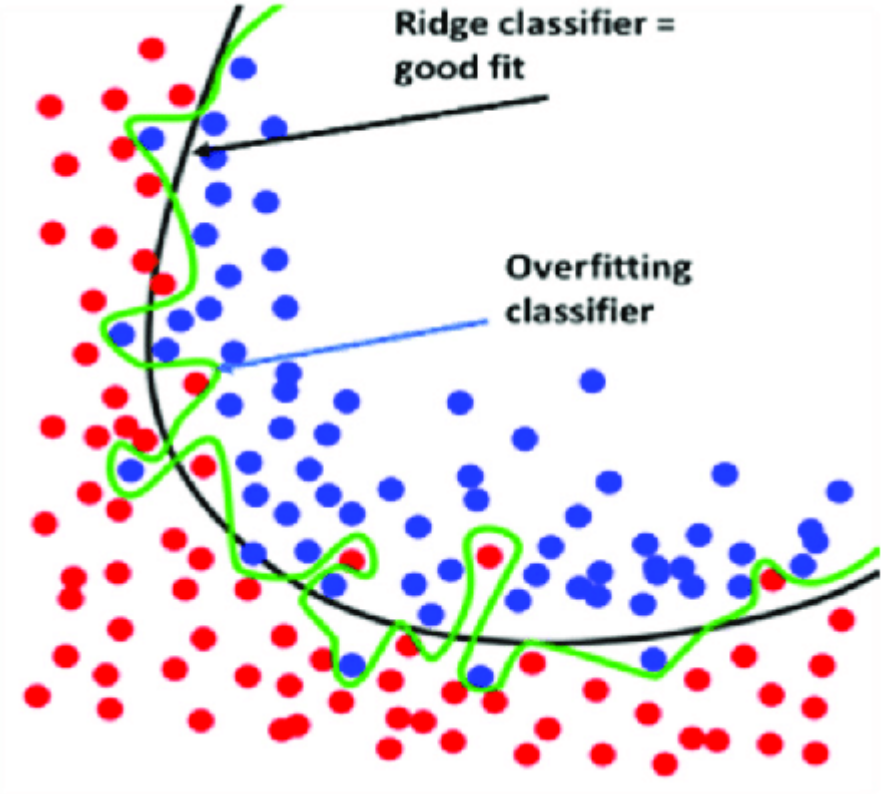


Figure 4.6: Heuristic view of ridge classifier vs. an over-fitting classifier

Chapter 5

Results Analysis

5.1 Confusion Matrices

The confusion matrix is a matrix that is used to evaluate the performance of classification models when applied to a given collection of test information. A determination may only be made if the true values for the test data are known in advance. The matrix itself is straightforward to comprehend, but the associated terminologies may be difficult to grasp. A model performance error matrix is also referred as an error matrix since it displays the errors in a matrix representation of the model's performance. It is comprised of a table with four different combinations of projected and actual results. It is useful for determining additional performance indicators such as recall, precision, specificity, accuracy, and the AUC-ROC Curve.

	Predicted	
Actual	TF	FP
	FN	TP

True Positive: The positive incidence that was predicted occurs in reality. The model predicted that a patient would have ASD, which he or she actually have.

True Negative: The situation that was predicted to be unfavorable actually happens to be negative. The model predicted that a patient did not have an autism spectrum disorder, which he or she does not have.

False Positive: (Type 1 Error): The situation that was predicted to be positive turned out to be negative. The model predicted that a patient has autism spectrum disorder, which he or she does not have.

False Negative: (Type 2 Error): The predicted negative event turns out to be a positive one. The model predicted that a patient did not have ASD, despite the fact that he or she does.

The following are the Confusion Matrices of eight algorithms in detecting Autism Spectrum Disorder (ASD) based on our research:

TABLE 5.1

CONFUSION MATRICES FOR DIFFERENT ALGORITHMS

KNN (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	276	27
	True	31	106

KNN (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	273	30
	True	23	114

Gaussian NB (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	291	12
	True	4	133

Gaussian NB (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	287	16
	True	0	137

Bernoulli NB (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	283	20
	True	5	132

Bernoulli NB (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	281	22
	True	4	133

Multinomial NB (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	236	67
	True	30	107

Multinomial NB (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	236	67
	True	28	109

Extra Tree (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	302	1
	True	3	134

Extra Tree (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	302	1
	True	3	134

LDA (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	287	16
	True	0	137

LDA (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	287	16
	True	0	137

QDA (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	239	64
	True	60	77

QDA (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	302	1
	True	0	137

Ridge Classifier (Before Oversampling & Tuning)		Predicted	
		False	True
Actual	False	288	15
	True	0	137

Ridge Classifier (After Oversampling & Tuning)		Predicted	
		False	True
Actual	False	287	16
	True	0	137

5.2 Formula of Performance Matrices

The following are the formulae that were utilized in our research to measure the performance matrices of ML models:

- $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 - Score = \frac{2 \times precision \times Recall}{Precision+Recall}$

5.3 Comparison of Accuracy among Different Algorithms

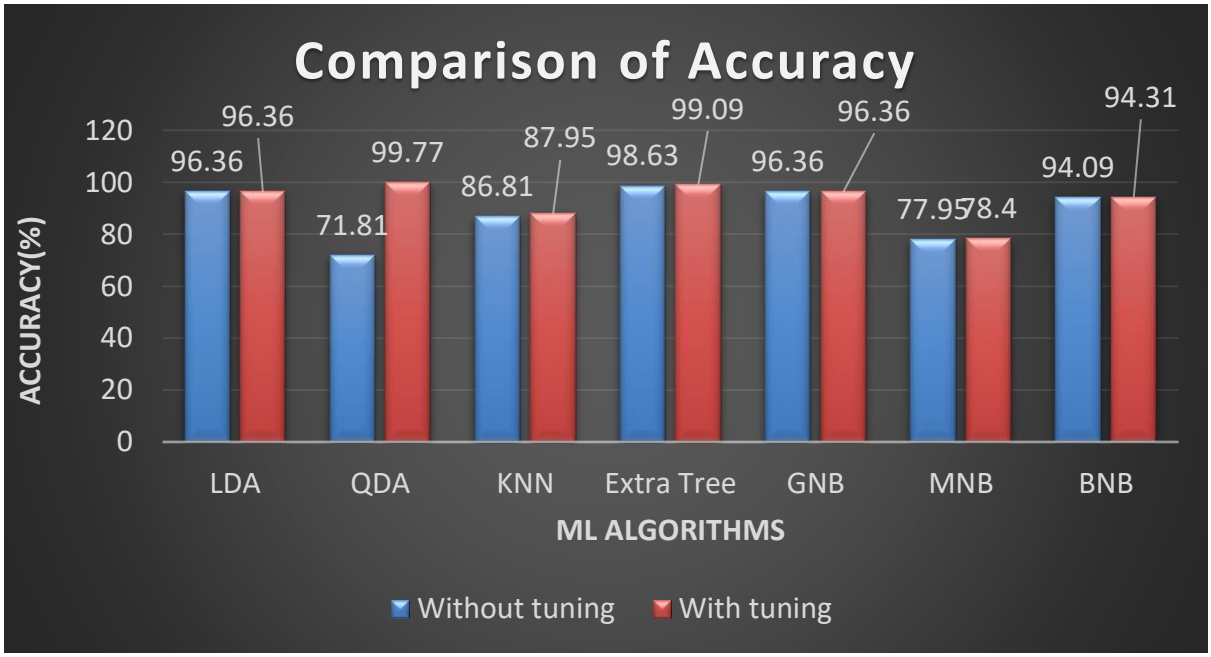
In terms of performance metrics, accuracy is the most often used and intuitive one. It is defined as the ratio of correctly anticipated incidents to the total number of incidents. The following is the equation for accuracy in the classification problem:

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}}$$

TABLE 5.2

COMPARISON OF ACCURACIES AMONG DIFFERENT ALGORITHMS

Algorithm	Accuracy Without tuning(%)	Accuracy With tuning(%)
LDA	96.36	96.36
QDA	71.81	99.77
KNN	86.81	87.95
Extra Tree	98.63	99.09
GNB	96.36	96.36
MNB	77.95	78.4
BNB	94.09	94.31



5.4 Comparison of Precision among Different Algorithms

Higher accuracy does not always imply that the model is the most accurate one. Inaccuracy can be an excellent measure of performance only in cases when the dataset is symmetric, meaning that both false positive and false negative incidences have the same value. However, this is not true in the case of illness detection models. As a result, other metrics such as precision were analyzed in order to assess the overall performance of the model. The precision of a prediction is defined as the ratio of correctly predicted positive incidents to the total number of correctly predicted positive incidents. Low false positive rate is associated with high precision.

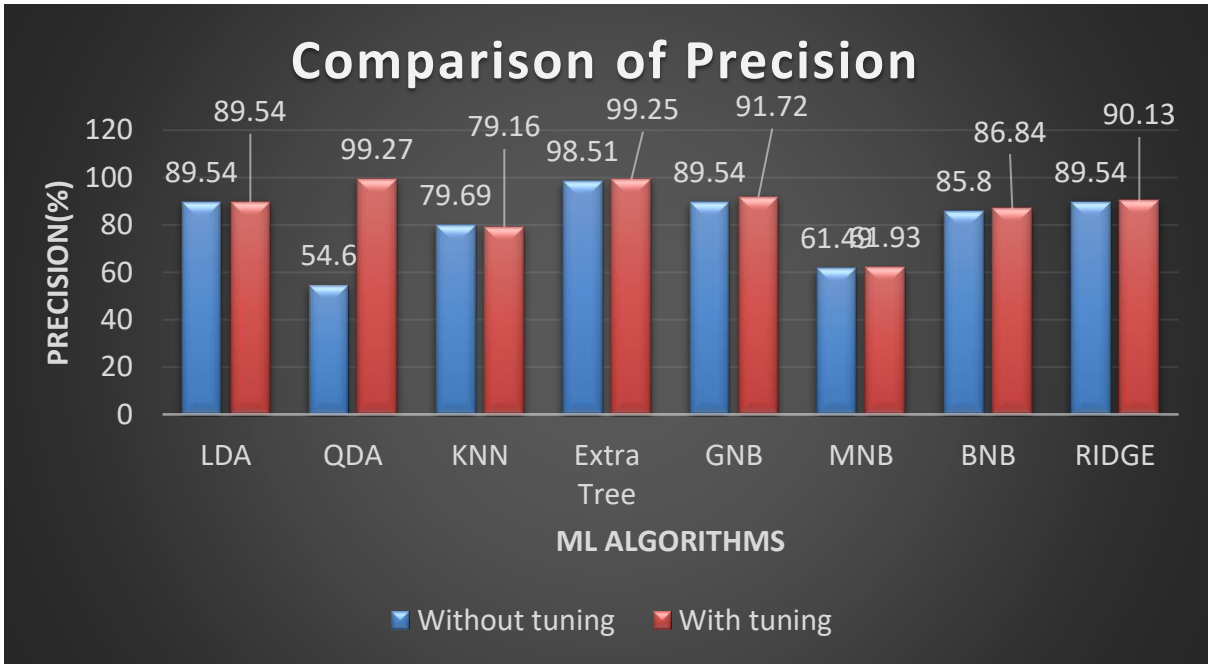
$$Precision = \frac{TP}{TP + FP}$$

This section contains a comparison of precisions among different algorithms for both default hyperparameters and modified hyperparameters. The results are as follows:

TABLE 5.3

COMPARISON OF PRECISIONS AMONG DIFFERENT ALGORITHMS

Algorithm	Precision Without tuning	Precision With tuning
LDA	89.54	89.54
QDA	54.6	99.27
KNN	79.69	79.16
Extra Tree	98.51	99.25
GNB	89.54	91.72
MNB	61.49	61.93
BNB	85.8	86.84



5.5 Comparison of F1-scores among Different Algorithms

The F1 Score is the weighted average of the Precision and Recall measurements. True positives and false negatives are both used in the computation for this performance statistic. F1-score provides more information than accuracy when dealing with a model that has an uneven class distribution. If the weights of false positives and false negatives are similar, accuracy is the most important performance measure. In disease detection models, on the other hand, this is not the case. As a result, Precision and Recall are taken into consideration in our approach. As a result, F1-scores for eight different algorithms are computed.

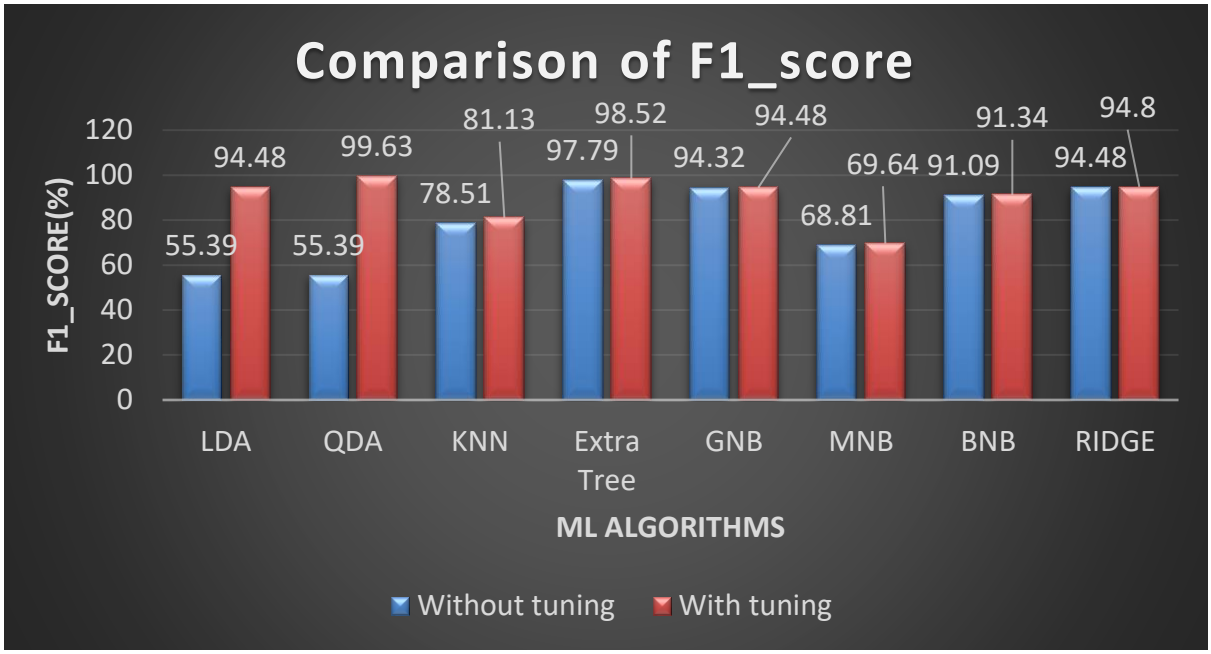
The following table shows a comparison of F1-scores between different algorithms for both default hyperparameters and adjusted hyperparameters:

TABLE 5.4

COMPARISON OF F1-SCORES AMONG DIFFERENT ALGORITHMS

Algorithm	F1-Score Without tuning	F1-Score With tuning
LDA	55.39	94.48
QDA	55.39	99.63

KNN	78.51	81.13
Extra Tree	97.79	98.52
GNB	94.32	94.48
MNB	68.81	69.64
BNB	91.09	91.34



5.6 Comparison of AUC-ROC among Different Algorithms

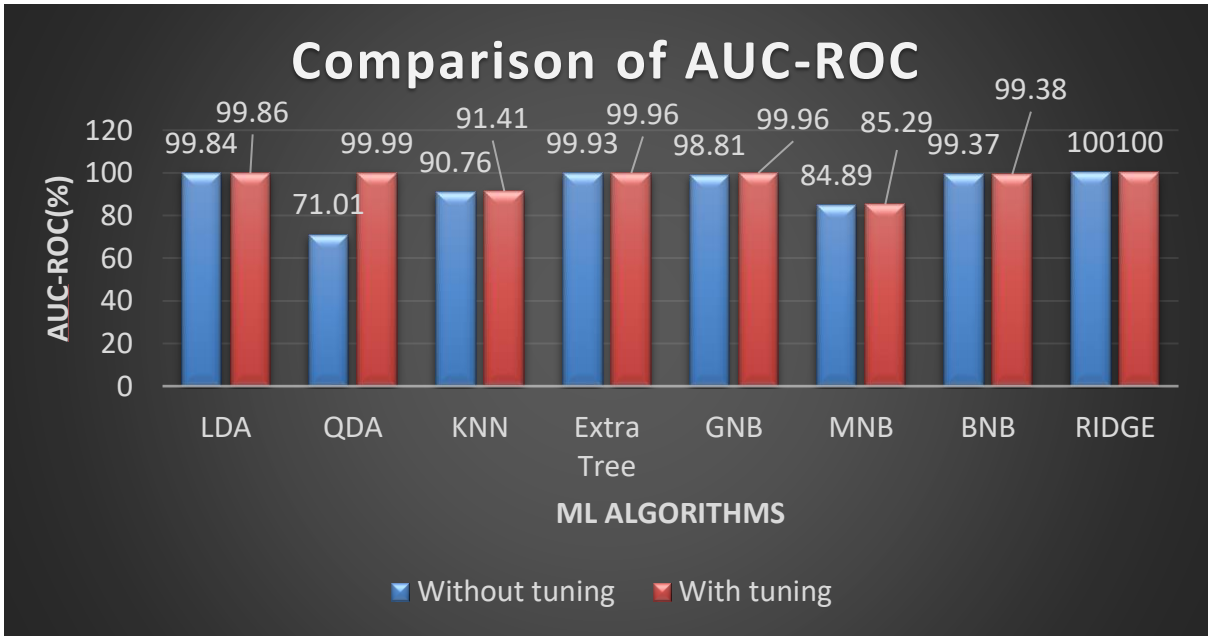
Area Under the receiver's curve Operating Characteristics is one of the most essential performance measures for evaluating the performance of classification models. ROC is a probability curve, whereas AUC indicates the degree of separability. It represents the model's capacity to differentiate between classes. A higher AUC value indicates that the model is more adept at prediction. On the other side, a model with an AUC close to 0 will have poor predictive ability.

The following table compares the AUC-ROC of various algorithms with both default and adjusted hyperparameters:

TABLE 5.5

COMPARISON OF AUC-ROC AMONG DIFFERENT ALGORITHMS

Algorithm	Roc-Auc Without tuning	Roc-Auc With tuning
LDA	99.84	99.86
QDA	71.01	99.99
KNN	90.76	91.41
Extra Tree	99.93	99.96
GNB	98.81	99.96
MNB	84.89	85.29
BNB	99.37	99.38



5.7 Comparison Specificity among Different Algorithms

When evaluating the performance of a model, sensitivity is frequently contrasted to specificity. Specificity is the proportion of genuine negatives that the model accurately identifies. This suggests that there will be an additional proportion of actual negatives that were projected as positives and could be referred to as false positives. This fraction is also known as the True

Negative Rate (TNR). Specificity (actual negative rate) plus false positive rate would always add up to 1. A high specificity indicates that the model properly identifies the majority of negative findings, whereas a low specificity indicates that the model incorrectly labels a significant number of negative results as positive.

Specificity can be calculated mathematically as follows:

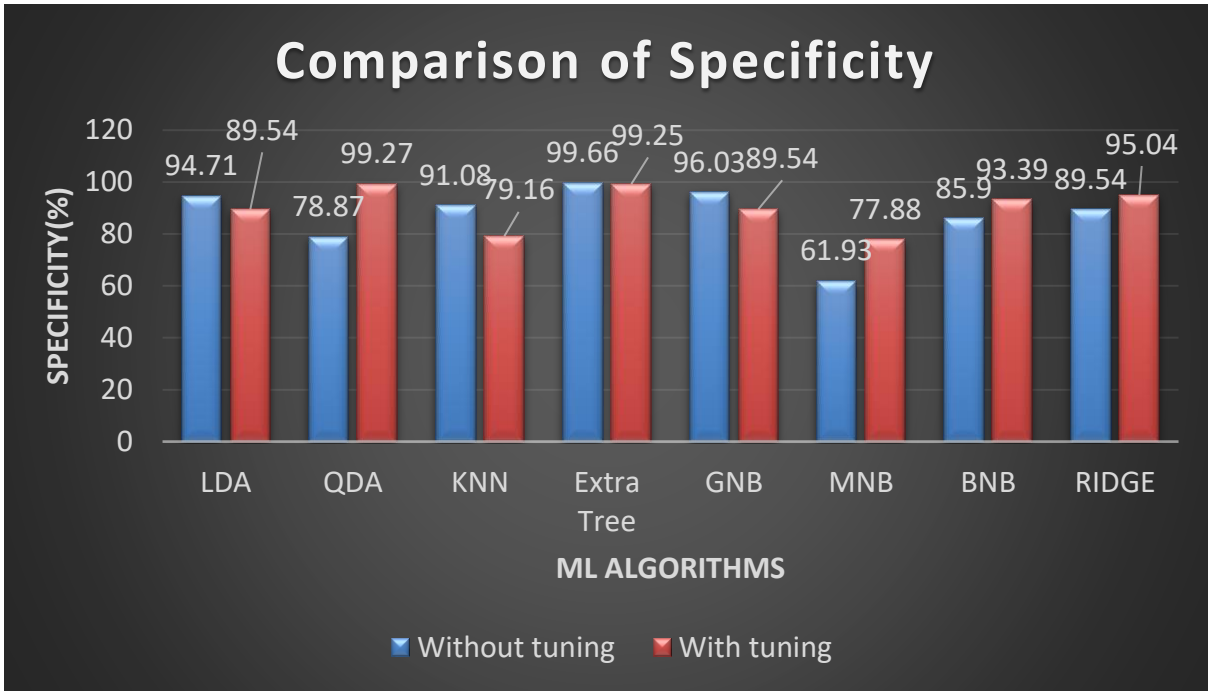
$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$$

Below is a comparison of the specificity of various algorithms.

TABLE 5.6

COMPARISON OF SPECIFICTY AMONG DIFFERENT ALGORITHMS

Algorithm	Specificity Without tuning	Specificity With tuning
LDA	94.71	89.54
QDA	78.87	99.27
KNN	91.08	79.16
Extra Tree	99.66	99.25
GNB	96.03	89.54
MNB	61.93	77.88
BNB	85.9	93.39



5.8 Comparison Cross Validation Score among Different Algorithms

Cross-validation is largely utilized in applied machine learning to assess the performance of a machine learning model on unobserved data. That is, using a small sample to estimate how well the model is predicted to perform when used to generate predictions on data that was not included during training.

It is a common technique because it is easy to comprehend and because it typically yields a less biased or less optimistic estimate of the model's skill than other techniques, such as a simple train/test split.

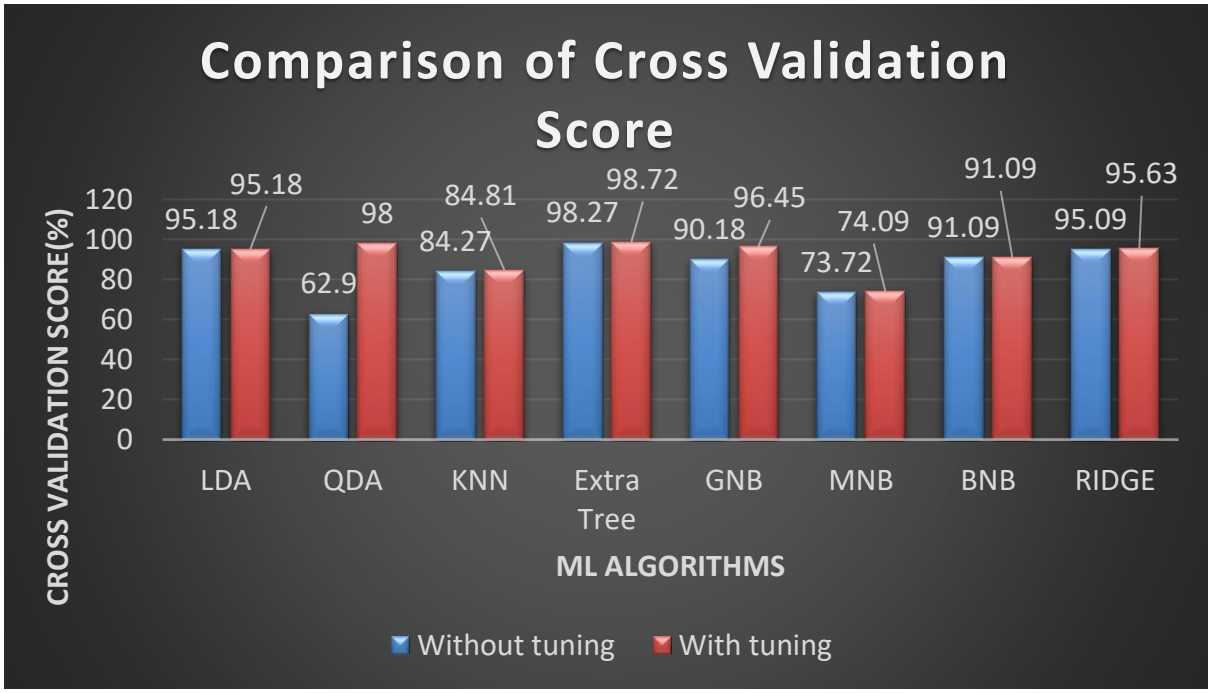
The comparison is given in below:

TABLE 5.7

COMPARISON OF CROSS VALIDATION SCORE AMONG DIFFERENT ALGORITHMS

Algorithm	Cross Validation Score Without tuning (%)	Cross Validation Score With tuning (%)

LDA	95.18	95.18
QDA	62.9	98
KNN	84.27	84.81
Extra Tree	98.27	98.72
GNB	90.18	96.45
MNB	73.72	74.09
BNB	91.09	91.09



5.9 Comparison Sensitivity among Different Algorithms

Sensitivity is a measure of a machine learning model's ability to identify positive events. The true positive rate (TPR) is also known as recall. Sensitivity is used to evaluate model performance since it reveals the number of positive events that the model accurately identified. A model with a high sensitivity will have few false negatives, meaning it will miss some positive examples. In other words, sensitivity quantifies a model's capacity to correctly recognize positive instances. This is essential because we need our models to be able to identify all positive events for accurate prediction. The total of sensitivity (rate of true positives) and

false negatives would equal 1. The higher the real positive rate, the better the model is in correctly identifying positive cases.

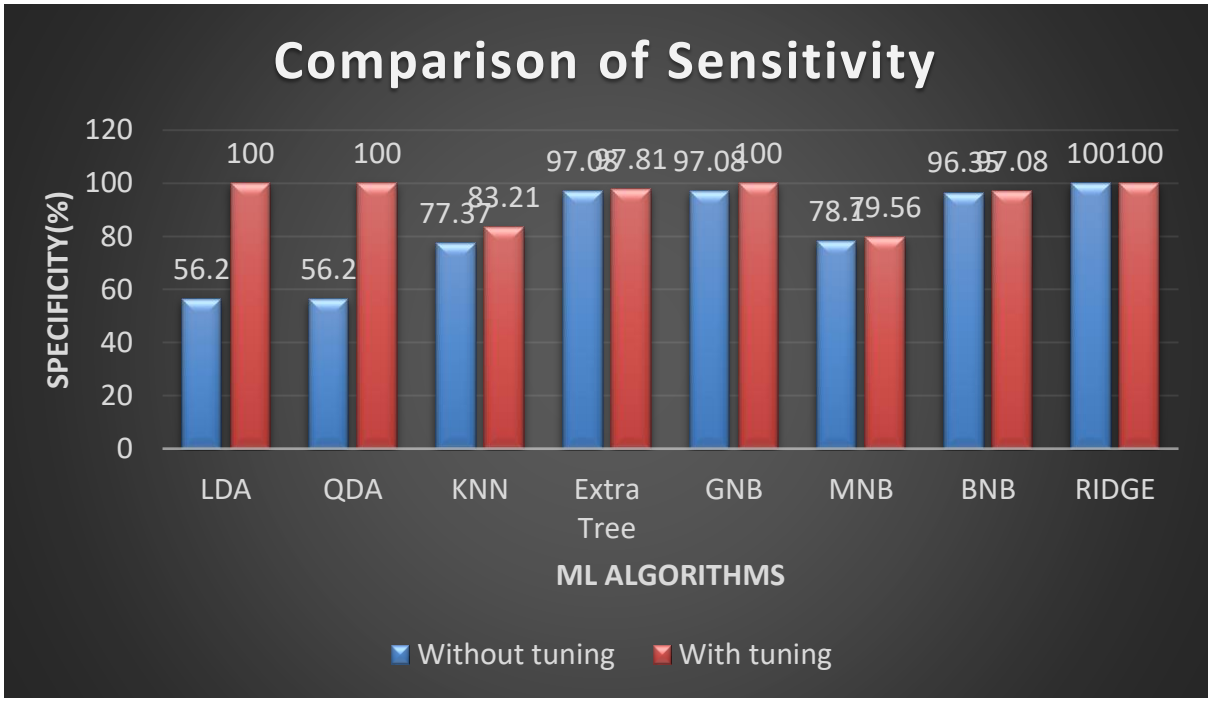
$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

The sensitivity or true positive rate can be estimated mathematically as follows:

TABLE 5.8

COMPARISON OF SENSITIVITY AMONG DIFFERENT ALGORITHMS

Algorithm	Sensitivity Without tuning (%)	Sensitivity With tuning (%)
LDA	56.2	100
QDA	56.2	100
KNN	77.37	83.21
Extra Tree	97.08	97.81
GNB	97.08	100
MNB	78.1	79.56
BNB	96.35	97.08



5.10 Comparison Error-Rate among Different Algorithms

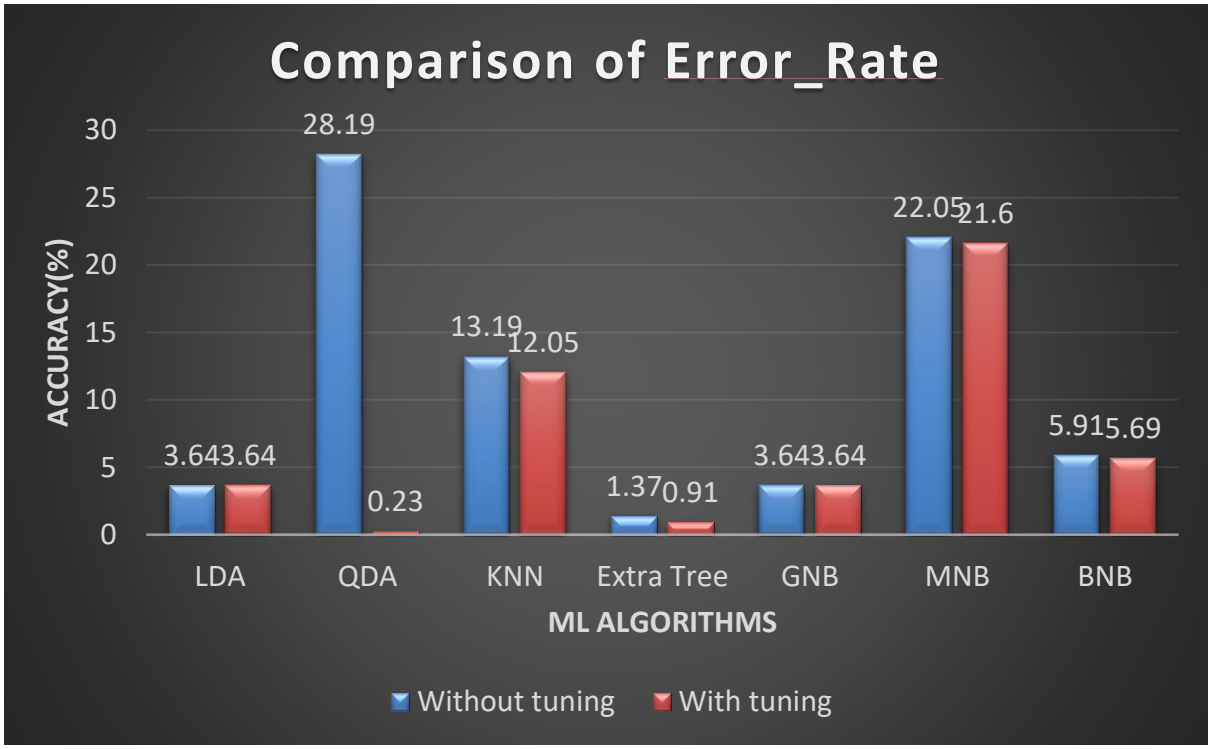
When categorical goal values are present, the error is reported as an error rate.

This is the rate at which the prediction is incorrect.

TABLE 5.9

COMPARISON OF ERROR-RATE AMONG DIFFERENT ALGORITHMS

Algorithm	Error Rate Without tuning(%)	Error Rate With tuning(%)
LDA	3.64	3.64
QDA	28.19	0.23
KNN	13.19	12.05
Extra Tree	1.37	0.91
GNB	3.64	3.64
MNB	22.05	21.6
BNB	5.91	5.69



Chapter 6

Conclusion & Future Scope

6.1 Conclusion

Autism spectrum disorder is an extremely difficult disease to diagnose. There is no single answer to this problem, despite the fact that many scholars and scientists have proposed numerous approaches. Our objective was to facilitate computer-assisted diagnostics.

ASD can be mild or severe if genes involved with cognitive and neuropsychiatric development are altered in DNA. This illness can have an influence on individuals at any stage of life. This being the case, early detection is essential for limiting the severity of the condition and developing effective treatments. Subsequently, intellectuals have begun implementing Machine Learning (ML) techniques due to their efficiency.

There were three types of datasets, and we consolidated them all into a single dataset. We experimented with methods for discriminant analysis, which are unique in terms of

implementations of algorithms utilized by scholars to date. In total, eight algorithms were applied twice, both with and without hyperparameter adjustment. We observed how GridSearchCV improved performance, and we adorned all the data with graphical representations.

In this study, a total of eight classifier algorithms are studied, and well-structured analytical models with hyperparameter tweaking are developed for improved results. In practically every performance metric, the performance of QDA with hyperparameter adjustment has been superior to that of all other algorithms. With hyperparameter optimization, QDA achieves an accuracy of 99.77 percent, a precision of 99.27 percent, an F1-score of 99.63 percent, a Cross Validation Score of 98 percent, and an error rate of 0.23 percent.

6.2 Future Scope

The complexity of an e-healthcare system coordinated by ML must play a significant part in the clinical judgment of physicians, guide patients to prospective ailments, and aid hospitals in mapping and treating them meticulously.

We do not intend to stop here, however. We wish to implement our knowledge of the applicability of diagnosis using real data collected from hospitals in Bangladesh that treat autism. We are also attempting to create hybrid models that are more efficient. There is a dearth of user-friendly medical diagnosis and solution-providing applications, thus we hope that our machine learning model can be used to develop a workable application for people all over the world. The complexity of an e-healthcare system coordinated by ML must necessarily play a significant part in physicians' clinical judgment, guide patients to prospective diseases, and aid hospitals in mapping and treating them precisely.

References

[1] T. Akter *et al.*, "Machine learning-based models for early stage detection of autism spectrum disorders," *IEEE Access*, vol. 7, pp. 166509–166527, 2019.

[2] F. Thabtah, "Machine learning in autistic spectrum disorder behavioral research: A review and ways forward," *Informatics for Health and Social Care*, vol. 44, no. 3. Taylor and Francis Ltd, pp. 278–297, Jul. 03, 2019, doi: 10.1080/17538157.2017.1399132.

[3] K. Akyol, "Assessing the importance of autistic attributes for autism screening," *Expert Syst.*, vol. 37, no. 5, pp. 1–10, 2020, doi: 10.1111/exsy.12562.

[4] S. Raj and S. Masood, "Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 994–1004, 2020, doi: 10.1016/j.procs.2020.03.399.

[5] S. Akhter, A. H. M. E. Hussain, J. Shefa, G. K. Kundu, F. Rahman, and A. Biswas, "Prevalence of Autism Spectrum Disorder (ASD) among the children aged 18-36 months in a rural community of Bangladesh: A cross sectional study [version 1; referees: 1 approved, 2 approved with reservations]," *F1000Research*, vol. 7, no. May, pp. 1–15, 2018, doi: 10.12688/f1000research.13563.1.

[6] K. D. Rajab, A. Padmavathy, and F. Thabtah, "Machine Learning Application for Predicting Autistic Traits in Toddlers," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3793–3805, 2021, doi: 10.1007/s13369-020-05165-3.

[7] F. Thabtah, "Autism spectrum disorder screening: Machine learning adaptation and DSM-5 fulfillment," in *ACM International Conference Proceeding Series*, May 2017, vol. Part F1293, pp. 1–6, doi: 10.1145/3107514.3107515.

[8] M. S. Mythili and A. R. M. Shanavas, "A Study on Autism Spectrum Disorders using Classification Techniques," *Int. J. Soft Comput. Eng.*, no. 5, pp. 2231–2307, 2014, [Online]. Available: http://www.ijsce.org/wp-content/uploads/papers/v4i5/E2433114514.pdf.

[9] C. Allison, B. Auyeung, and S. Baron-Cohen, "Toward brief 'red flags' for autism screening: The short Autism Spectrum Quotient and the short Quantitative Checklist in 1,000 cases and 3,000 controls," *J. Am. Acad. Child Adolesc. Psychiatry*, vol. 51, no. 2, pp. 202-212.e7, 2012, doi: 10.1016/j.jaac.2011.11.003.

[10] F. Thabtah, "An accessible and efficient autism screening method for behavioural data and predictive analyses," *Health Informatics J.*, vol. 25, no. 4, pp. 1739–1755, 2019,

doi: 10.1177/1460458218796636.

[11] C. Küpper *et al.*, "Identifying predictive features of autism spectrum disorders in a clinical sample of adolescents and adults using machine learning," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, 2020, doi: 10.1038/s41598-020-61607-w.

[12] K. S. Oma, P. Mondal, N. S. Khan, M. R. K. Rizvi, and M. N. Islam, "A Machine Learning Approach to Predict Autism Spectrum Disorder," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, no. April, 2019, doi: 10.1109/ECACE.2019.8679454.

[13] R. Vaishali and R. Sasikala, "A machine learning based approach to classify Autism with optimum behaviour sets," *Int. J. Eng. Technol.*, vol. 5, no. x, pp. 1–6, 2017, doi: 10.14419/ijet.v7i3.18.14907Published.

[14] E. T. Prud'hommeaux, B. Roark, L. M. Black, and J. van Santen, "Classification of atypical language in autism," *Acl Hlt 2011*, no. June, p. 88, 2011.

[15] K. T. Quach, "Application of Artificial Neural Networks in Classification of Autism Diagnosis Based on Gene Expression Signatures," 2012.

[16] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007, doi: 10.1198/004017007000000245.

[17] P. Mohan and I. Paramasivam, "Feature reduction using SVM-RFE technique to detect autism spectrum disorder," *Evol. Intell.*, no. 0123456789, 2020, doi: 10.1007/s12065-020-00498-2.

[18] H. Wang, L. Li, L. Chi, and Z. Zhao, "Autism Screening Using Deep Embedding Representation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11537 LNCS, pp. 160–173, 2019, doi: 10.1007/978-3-030-22741-8_12.

[19] K. Akyol, Y. Gultepe, and A. Karaci, "A Study on Autistic Spectrum Disorder for Children Based on Feature Selection and Fuzzy," *Int. Congr. Engieneering Life Sci.*, no. October, pp. 804–807, 2018.

[20] M. Thomas and A. Chandran, "Artificial Neural Network for Diagnosing Autism Spectrum Disorder," *Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018*, vol. 3, no. 2, pp. 930–933, 2018, doi: 10.1109/ICOEI.2018.8553781.

[21] O. Altay and M. Ulas, "Prediction of the autism spectrum disorder diagnosis with linear discriminant analysis classifier and K-nearest neighbor in children," *6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding*, vol. 2018-Janua, pp. 1–4,

2018, doi: 10.1109/ISDFS.2018.8355354.

[22] G. Lucarelli and M. Borrotti, *Artificial Intelligence Applications and Innovations*, vol. 559, no. 691154. Cham: Springer International Publishing, 2019.

[23] N. Goel, B. Grover, Anuj, D. Gupta, A. Khanna, and M. Sharma, "Modified Grasshopper Optimization Algorithm for detection of Autism Spectrum Disorder," *Phys. Commun.*, vol. 41, p. 101115, 2020, doi: 10.1016/j.phycom.2020.101115.

[24] S. B. Shuvo, J. Ghosh, and A. S. Oyshi, "A Data Mining Based Approach to Predict Autism Spectrum Disorder Considering Behavioral Attributes," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 1–5, 2019, doi: 10.1109/ICCCNT45670.2019.8944905.

[25] A. Choudhury and C. M. Greene, "Prognosticating Autism Spectrum Disorder Using Artificial Neural Network: Levenberg-Marquardt Algorithm," *arXiv*, vol. 2, no. 6, 2018, doi: 10.26502/acbr.50170058.

[26] N. Chakrabarty and S. Biswas, "Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets," *J. Electron. Informatics*, vol. 2, no. 2, pp. 96–136, 2020, doi: 10.36548/jei.2020.2.004.

[27] M. M. Nishat *et al.*, "Detection of Autism Spectrum Disorder by Discriminant Analysis Algorithm," *Lect. Notes Data Eng. Commun. Technol.*, vol. 95, pp. 473–482, 2022, doi: 10.1007/978-981-16-6636-0_36.

[28] M. J. Zaki and W. Meira, Jr, "Linear Discriminant Analysis," *Data Min. Mach. Learn.*, pp. 501–516, 2020, doi: 10.1017/9781108564175.025.

[29] A. Singh, B. S. Prakash, and K. Chandrasekaran, "A comparison of linear discriminant analysis and ridge classifier on Twitter data," *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2016*, pp. 133–138, 2017, doi: 10.1109/CCAA.2016.7813704.

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

[31] https://www.javatpoint.com/data-preprocessing-machine-learning

[32] https://medium.com/analytics-vidhya/why-hyper-parameter-tuning-is-important-for-your-model-1ff4c8f145d3

[33] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[34] https://www.upgrad.com/blog/gaussian-naivebayes/#:~:text=What%20is%20Gaussian%20Na%C3%AFve%20Bayes,theorem%20with%20strong%20independence%20assumptions.

[35] https://iq.opengenus.org/bernoulli-naive-bayes/

[36] https://www.upgrad.com/blog/multinomial-naive-bayes-explained/#:~:text=The%20Multinomial%20Naive%20Bayes%20algorithm%20is%20a%20Bayesian%20learning%20approach,tag%20with%20the%20greatest%20chance.

[37] https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/#:~:text=Linear%20discriminant%20analysis%20(LDA)%20is,values%2C%20which%20form%20a%20template.

[38] https://medium.com/machine-learning-researcher/dimensionality-reduction-pca-and-lda-6be91734f567

[39] https://www.geeksforgeeks.org/quadratic-discriminant-analysis/

[40] https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/

[41] https://datacadamia.com/data_mining/discriminant_analysis_quadratic

[42] https://towardsdatascience.com/going-beyond-the-simpleimputer-for-missing-data-imputation-dd8ba168d505

[43] https://towardsdev.com/how-to-identify-missingness-types-with-missingno-61cfe0449ad9

[44] https://www.analyticsvidhya.com/blog/2020/07/knnimputer-a-robust-way-to-impute-missing-values-using-scikit-learn/

[45] https://www.jair.org/index.php/jair/article/view/10302

[46] https://arxiv.org/abs/1106.1813

[47] https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/