**ANALYSIS AND DETECTION OF AUTISM SPECTRUM DISORDER**

**INNOVATIVE PRODUCT DEVELOPMENT REPORT 1**

# SUBMITTED BY

P. LALITHASRI (22RH1A12E7)

SK. TASLEEM (22RH1A12G6)

V. SANJEEVANI (22RH1A12J4)

**Under the Esteemed Guidance of**

Mrs. HARITHA LAKSHMI

Assistant Professor

**In partial full fillment of the Academic Requirements for the Degree of**

# BACHELOR OF TECHNOLOGY

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**MALLA REDDY ENGINEERING COLLEGE FOR WOMEN**

Autonomous Institution, UGC, Govt. of India

Programmes Accredited by NBA, Accredited by NAAC with A+ Grade, Govt. of India.

Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institute.

National Ranking by NIRF Innovation-Rank Band(151-300), AAAA Rated by career 360 Magazine

.AAAA+ Rated by Digital Learning Magazine, 12th Top Engineering College of Super Band-Excellent CSR-2023. Green Ranking “Gold Band” Sustainable Institution of India.

Maisammaguda (V), Dhullapally (Post), (Via) Kompally, Medchal Malkajgiri Dist. T.S-500100

2024-2025

## MALLA REDDY ENGINEERING COLLEGE FOR WOMEN

**Autonomous Institution, UGC, Govt. of India**

Programmes Accredited by NBA

Accredited by NAAC with A+ Grade

Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institute

Maisammaguda (V), Dhullapally (Post), (Via) Kompally, Medchal Malkajgiri Dist. T.S-500100

**DEPARTMENT OF INFORMATION TECHNOLOGY**

### CERTIFICATE

This is to certify that the Innovative product development-2 embodies in this dissertation entitled’**Analysis and Detection of Autism Spectrum Disorde** being submitted by ‘**P. Lalithasri (22RH1A12E7), SK. Tasleem (22RH1A12G6), V. Sanjeevani(22RH1A12J4)**’ for partial fulfillment of the requirement for the award of Bachelor of Technology in Information Technology, Malla Reddy Engineering College for Women (Autonomous), Maisammaguda, Secundrabad during the academic year 2024 – 2025

**Guide Head of the Department**

**External Examiner**

# MALLA REDDY ENGINEERING COLLEG FOR WOMEN

 **Autonomous Institution, UGC, Govt. of India**

Programmes Accredited by NBA

Accredited by NAAC with A+ Grade

Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institute

Maisammaguda (V), Dhullapally (Post), (Via) Kompally, Medchal Malkajgiri Dist. T.S-500100

**DEPARTMENT OF INFORMATION TECHNOLOGY**

## DECLARATION

### We ‘P. Lalithasri(22RH1A12E7), SK.Tasleem (20RH1A12G6), V.Sanjeevani

**(22RH1A12J4)’** are students of ‘Bachelor of Technology in Information Technology’, Malla Reddy Engineering College for Women (Autonomous), Maisammaguda, Secunderabad, here by declare that the work presented in this Project Work entitled “**Analysis and Detection of Autism Spectrum Disorder ’**” is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Date:**

**P.LALITHASRI (22RH1A12E7)**

**SK.TASLEEM (22RH1A12G6)**

**V.SANJEEEVANI (22RH1A12J4)**

### ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women** and Department of Information Technology which gave us the opportunity to have expertise in engineering and profound technical knowledge. We would like to deeply thank our Honorable Minister of Telangana State Sri.Ch. Malla Reddy Garu, founder chairman MRGI, the largest cluster of institutions in the state of Telangana for providing us with all the resources in the college to make our project success.

We wish to convey gratitude to our Principal **Dr. Y. Madhavee Latha**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential. We express our sincere gratitude to**, Dr. Sateesh Nagavarapu** Professor & Head, Department of Information Technology for inspiring us to take up a project on this subject and successfully guiding us towards its completion. We express our sincere gratitude to Prof. A.Prem, Director of Computer Science and Engineering and Information Technology for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We would like to thank our internal guide **Mrs. Haritha lakshmi** and all the faculty members for their valuable guidance and encouragement towards the completion of our project work.

### With regards and Gratitude

P. Lalithasri (22RH1A12E7)

SK. Tasleem (22RH1A12G6)

V. Sanjeevani (22RH1A12J4)

**ABSTRACT**

Autism Spectrum Disorder (ASD) is a neuro-disorder in which a person has a lifelong effect on interaction and communication with others. Autism can be diagnosed at any stage in once life and is said to be a "behavioural disease" because in the first two years of life symptoms usually appear. According to the ASD problem starts with childhood and continues to keep going on into adolescence and adulthood. Propelled with the rise in use of machine learning techniques in the research dimensions of medical diagnosis, in this paper there is an attempt to explore the possibility to use Naïve Bayes, Support Vector Machine, Logistic Regression, KNN, Neural Network and Convolutional Neural Network for predicting and analysis of ASD problems in a child, adolescents, and adults. The proposed techniques are evaluated on publicly available three different non-clinically ASD datasets. First dataset related to ASD screening in children has 292 instances and 21 attributes. Second dataset related to ASD screening Adult subjects contains a total of 704 instances and 21 attributes. Third dataset related to ASD screening in Adolescent subjects comprises of 104 instances and 21 attributes. After applying various machine learning techniques and handling missing values, results strongly suggest that CNN based prediction models work better on all these datasets with higher accuracy of 99.53%, 98.30%, 96.88% for Autistic Spectrum Disorder Screening in Data for Adult, Children, and Adolescents respectively.

**CONTENTS**

**S.NO TITLE PAGE NO**

ABSTRACT I

1. **CHAPTER INTRODUCTION 1**
   1. Introduction
   2. Objective of the project
2. **LITERATURE SURVEY**
3. **SYSTEM ANALYSIS**
   1. Existing system
   2. Proposed System
   3. Proposes model used with justification
   4. Software requirement specification

3.4.2 External interface requirements

**4 SYSTEM DESIGN**

4.1 UML Diagram

4.2 Class Diagram

4.3 Use case diagram

4.4Sequence diagram

4.5 Collaboration diagram

4.6 Component diagram

4.7 Deployment diagram

4.8 Activity diagram

4.9 Data flow diagram

**5 IMPLEMENTATION**

5.1 Python

5.2 Sample code

**6 TESTING**

6.1 System testing

6.2 Module testing

6.3 Integration testing

6.4 Acceptance testing

**7 SCREENSHOTS**

**8 CONCLUSION**

**9 REFERENCE**

**ANALYSIS AND DETECTION OF AUTISM SPECTRUM DISORDER USING MACHINE LEARNING TECHNIQUES**

**1.INTRODUCTION**

The problem of autism spectrum disorder (ASD) have been mounting swiftly nowadays among all ages of the human population. Early detection of this neurological disease can greatly assist in the maintenance of the subject’s mental and physical health. With the rise of application of machine learning-based models in the predictions of various human diseases, their early detection based on various health and physiological parameter now seems possible. This factor motivated us to increase interest in the detection and analysis of ASD diseases to improve better treatment methodology. Detection of ASD becomes a challenge as there are several other mental disorders whose few symptoms are very similar to those with ASD symptoms, thereby makes this task a difficult one. Autism Spectrum disorder is a problem that is related to human brain development. A person who has suffered from the Autism Spectrum Disorder is generally not able to do social interaction and communication with other persons.In this, a person's life is usually affected for his or her entire lifetime. It is interesting to know that both environmental and genetic factors may turn out to be the causing factors for this disease. The symptoms of this problem may be started at the age of three years and may continue for the lifetime. It is not possible to complete treat the patient suffering from this disease, however its effects can be reduced for some time if the symptoms are early detected. By assuming that human genes are responsible for it, the exact causes of ASD have not been recognized by the scientist yet. The human genes affect the development by influencing the environment. There is some risk factor which influences ASD like as low birth weight, a sibling with ASD and having old parents, etc. Instead of this, there are some social interaction and communication problems like as: • Inappropriate laughing and giggling • No sensitivity of pain • Not able to make eye contact properly • No proper response to sound • May not have a wish for cuddling • Not able to express their gestures • No interaction with others • Inappropriate objects attachment • Want to live alone • Using echo words etc. People with ASD also have difficulty with constrained interests and consistently repetition of behaviors. The following list presents specific examples of the types of behaviours. • Repeating certain behaviours like repeating words or phrases much time. • The Person will be upset when a routine is going to change. • Having a little interest in certain matters of the topic like numbers, facts, etc. • Less sensitive than another person in some cases like light, noise, etc. Early detection and treatment are most important steps to be taken to decrease the symptoms of autism spectrum disorder problem and to improve the quality of life of ASD suffering people. However, there is no procedure of medical test for detection of autism. ASD Symptoms usually recognized by observation. In Older and adolescents who go to school, ASD symptoms are usually identified by their parents and teachers. After that ASD symptoms are evaluated by a special education team of the school. These school team suggested these children visit their health care doctor for required testing. In adults identifying ASD symptoms is very difficult than older children and adolescents because some symptoms of ASD may be overlap with other mental health disorders. It is easy to identify the behavioural changes in a child easily by observation because it can be seen early in the 6 months of age than Autism specific brain imaging because brain imaging can be identifying after 2 years of age. The contents of this paper are organized as follows: Section 1 presents the introduction to the Autism Spectrum Disorder problem and the challenges faced by the subjects. Section 2 presents the review of various recent literature, where some models for ASD detection have been developed. Section 3 describes the datasets used in this study, which is followed by description of each component of the methodology used in this work in section 4. The results obtained after various experiments are presented and discussed in Section 5 which is finally followed by the conclusion in section 6

**1.1 Objective of the Project**

Autism Spectrum Disorder (ASD) is a neuro-disorder in which a person has a lifelong effect on interaction and communication with others. Autism can be diagnosed at any stage in once life and is said to be a "behavioural disease" because in the first two years of life symptoms usually appear. According to the ASD problem starts with childhood and continues to keep going on into adolescence and adulthood. Propelled with the rise in use of machine learning techniques in the research dimensions of medical diagnosis, in this paper there is an attempt to explore the possibility to use Naïve Bayes, Support Vector Machine, Logistic Regression, KNN, Neural Network and Convolutional Neural Network for predicting and analysis of ASD problems in a child, adolescents, and adults. The proposed techniques are evaluated on publicly available three different non-clinically ASD datasets. First dataset related to ASD screening in children has 292 instances and 21 attributes. Second dataset related to ASD screening Adult subjects contains a total of 704 instances and 21 attributes. Third dataset related to ASD screening in Adolescent subjects comprises of 104 instances and 21 attributes. After applying various machine learning techniques and handling missing values, results strongly suggest that CNN based prediction models work better on all these datasets with higher accuracy of 99.53%, 98.30%, 96.88% for Autistic Spectrum Disorder Screening in Data for Adult, Children, and Adolescents respectively.

**2. LITERATURE SURVEY**

**"A new computational intelligence approach to detect autistic features for autism screening."**

Autism Spectrum Disorder (ASD) is one of the fastest growing developmental disability diagnosis. General practitioners (GPs) and family physicians are typically the first point of contact for patients or family members concerned with ASD traits observed in themselves or their family member. Unfortunately, some families and adult patients are unaware of ASD traits that may be exhibited and as a result do not seek out necessary diagnostic services or contact their GP. Therefore, providing a quick, accessible, and simple tool utilizing items related to ASD to these families may increase the likelihood they will seek professional assessment and is vital to the early detection and treatment of ASD. This study aims at identifying fewer, albeit influential, features in common ASD screening methods in order to achieve efficient screening as demands on evaluating the items’ influences on ASD within existing tools is urgent. To achieve this aim, a computational intelligence method called Variable Analysis (Va) is proposed that considers feature-to-class correlations and reduces feature-to-feature correlations. The results of the Va have been verified using two machine learning algorithms by deriving automated classification systems with respect to specificity, sensitivity, positive predictive values (PPVs), negative predictive values (NPVs), and predictive accuracy. Experimental results using cases and controls related to items in three common screening methods, along with features related to individuals, have been analysed and compared with results obtained from other common filtering methods. The results exhibited that Va was able to derive fewer numbers of features from adult, adolescent, and child screening methods yet maintained competitive predictive accuracy, sensitivity, and specificity rates.

"**Rapid quantitative assessment of autistic social impairment by classroom teachers”**

Teachers routinely observe children in the naturalistic social contexts of their classrooms and provide extremely important input in the evaluation of numerous psychiatric syndromes. Their precision in ascertaining and quantifying autistic symptomatology has not previously been established. In this study, we compared teachers' ratings of autistic symptomatology with those derived from parents, expert clinicians, and trained raters.

**"Applying machine learning to facilitate autism diagnostics: pitfalls and promises."**

Machine learning has immense potential to enhance diagnostic and intervention research in the behavioral sciences, and may be especially useful in investigations involving the highly prevalent and heterogeneous syndrome of autism spectrum disorder. However, use of machine learning in the absence of clinical domain expertise can be tenuous and lead to misinformed conclusions. To illustrate this concern, the current paper critically evaluates and attempts to reproduce results from two studies (Wall et al. in Transl Psychiatry 2(4):e100, 2012a; PloS One 7(8), 2012b) that claim to drastically reduce time to diagnose autism using machine learning. Our failure to generate comparable findings to those reported by Wall and colleagues using larger and more balanced data underscores several conceptual and methodological problems associated with these studies. We conclude with proposed best-practices when using machine learning in autism research, and highlight some especially promising areas for collaborative work at the intersection of computational and behavioral science.

**“Use of machine learning to shorten observationbased screening and diagnosis of autism”**

The Autism Diagnostic Observation Schedule-Generic (ADOS) is one of the most widely used instruments for behavioral evaluation of autism spectrum disorders. It is composed of four modules, each tailored for a specific group of individuals based on their language and developmental level. On average, a module takes between 30 and 60 min to deliver. We used a series of machine-learning algorithms to study the complete set of scores from Module 1 of the ADOS available at the Autism Genetic Resource Exchange (AGRE) for 612 individuals with a classification of autism and 15 non-spectrum individuals from both AGRE and the Boston Autism Consortium (AC). Our analysis indicated that 8 of the 29 items contained in Module 1 of the ADOS were sufficient to classify autism with 100% accuracy. We further validated the accuracy of this eight-item classifier against complete sets of scores from two independent sources, a collection of 110 individuals with autism from AC and a collection of 336 individuals with autism from the Simons Foundation. In both cases, our classifier performed with nearly 100% sensitivity, correctly classifying all but two of the individuals from these two resources with a diagnosis of autism, and with 94% specificity on a collection of observed and simulated non-spectrum controls. The classifier contained several elements found in the ADOS algorithm, demonstrating high test validity, and also resulted in a quantitative score that measures classification confidence and extremeness of the phenotype. With incidence rates rising, the ability to classify autism effectively and quickly requires careful design of assessment and diagnostic tools. Given the brevity, accuracy and quantitative nature of the classifier, results from this study may prove valuable in the development of mobile tools for preliminary evaluation and clinical prioritization—in particular those focused on assessment of short home videos of children—that speed the pace of initial evaluation and broaden the reach to a significantly larger percentage of the population at risk.

**“Use of machine learning to shorten observation based screening and diagnosis of autism”**

The Autism Diagnostic Interview-Revised (ADI-R) is one of the most commonly used instruments for assisting in the behavioral diagnosis of autism. The exam consists of 93 questions that must be answered by a care provider within a focused session that often spans 2.5 hours. We used machine learning techniques to study the complete sets of answers to the ADI-R available at the Autism Genetic Research Exchange (AGRE) for 891 individuals diagnosed with autism and 75 individuals who did not meet the criteria for an autism diagnosis. Our analysis showed that 7 of the 93 items contained in the ADI-R were sufficient to classify autism with 99.9% statistical accuracy. We further tested the accuracy of this 7-question classifier against complete sets of answers from two independent sources, a collection of 1654 individuals with autism from the Simons Foundation and a collection of 322 individuals with autism from the Boston Autism Consortium. In both cases, our classifier performed with nearly 100% statistical accuracy, properly categorizing all but one of the individuals from these two resources who previously had been diagnosed with autism through the standard ADI-R. Our ability to measure specificity was limited by the small numbers of non-spectrum cases in the research data used, however, both real and simulated data demonstrated a range in specificity from 99% to 93.8%. With incidence rates rising, the capacity to diagnose autism quickly and effectively requires careful design of behavioral assessment methods. Ours is an initial attempt to retrospectively analyze large data repositories to derive an accurate, but significantly abbreviated approach that may be used for rapid detection and clinical prioritization of individuals likely to have an autism spectrum disorder. Such a tool could assist in streamlining the clinical diagnostic process overall, leading to faster screening and earlier treatment of individuals with autism.

**Use of artificial intelligence to shorten the behavioral diagnosis of autism.**

The Autism Diagnostic Interview-Revised (ADI-R) is one of the most commonly used instruments for assisting in the behavioral diagnosis of autism. The exam consists of 93 questions that must be answered by a care provider within a focused session that often spans 2.5 hours. We used machine learning techniques to study the complete sets of answers to the ADI-R available at the Autism Genetic Research Exchange (AGRE) for 891 individuals diagnosed with autism and 75 individuals who did not meet the criteria for an autism diagnosis. Our analysis showed that 7 of the 93 items contained in the ADI-R were sufficient to classify autism with 99.9% statistical accuracy. We further tested the accuracy of this 7-question classifier against complete sets of answers from two independent sources, a collection of 1654 individuals with autism from the Simons Foundation and a collection of 322 individuals with autism from the Boston Autism Consortium. In both cases, our classifier performed with nearly 100% statistical accuracy, properly categorizing all but one of the individuals from these two resources who previously had been diagnosed with autism through the standard ADI-R. Our ability to measure specificity was limited by the small numbers of non-spectrum cases in the research data used, however, both real and simulated data demonstrated a range in specificity from 99% to 93.8%. With incidence rates rising, the capacity to diagnose autism quickly and effectively requires careful design of behavioural assessment methods. Ours is an initial attempt to retrospectively analyze large data repositories to derive an accurate, but significantly abbreviated approach that may be used for rapid detection and clinical prioritization of individuals likely to have an autism spectrum disorder. Such a tool could assist in streamlining the clinical diagnostic process overall, leading to faster screening and earlier treatment of individuals with autism.

**“Autism spectrum disorder screening: machine learning adaptation and DSM-5 fulfillment.”**

One of the primary psychiatric disorders is Autistic Spectrum Disorder (ASD). ASD is a mental disorder that limits the use of linguistic, communicative, cognitive, skills as well as social skills and abilities. Recently, ASD has been studied in the behavioural sciences using intelligent methods based around machine learning to speed up the screening time or to improve sensitivity, specificity or accuracy of the diagnosis process. Machine learning considers the ASD diagnosis problem as a classification task in which predictive models are built based on historical cases and controls. These models are supposed to be plugged into a screening tool to accomplish one or more of the aforementioned goals. In this paper, we shed light on recent studies that employ machine learning in ASD classification in order to discuss their pros and cons. Moreover, we highlight a noticeable problem associated with current ASD screening tools; the reliability of these tools using the DSM-IV rather than the DSM-5 manual. Hence the necessity to amend current screening tools to reflect the new imposed criteria of ASD classification in the DSM-5 particularly the diagnostic algorithms embedded within these methods.

**“Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning”**

Over the last few years, academic institutions have conducted a number of programmes to help school boards, colleges, and schools of autism spectrum educating pupils (ASD). Autism spectrum disorder (ASD) is a complicated neurological disorder which aﬀects many skills over a lifetime. The main aim of the chapter is to examine the topic of autism and identify autism levels with furious logic classiﬁcation algorithms using the artiﬁcial neural network. Data mining has generally been recognized as a method of decision making to promote higher use of resources for autism students.Over the last few years, academic institutions have conducted a number of programmes to help school boards, colleges, and schools of autism spectrum educating pupils

(ASD). Autism spectrum disorder (ASD) is a complicated neurological disorder which aﬀects many skills over a lifetime. The main aim of the chapter is to examine the topic of autism and identify autism levels with furious logic classiﬁcation algorithms using the artiﬁcial neural network. Data mining has generally been recognized as a method of decision making to promote higher use of resources for autism students.Over the last few years, academic institutions have conducted a number of programmers to help school boards, colleges, and schools of autism spectrum educating pupils (ASD). Autism spectrum disorder (ASD) is a complicated neurological disorder which aﬀects many skills over a lifetime. The main aim of the chapter is to examine the topic of autism and identify autism levels with furious logic classiﬁcation algorithms using the artiﬁcial neural network. Data mining has generally been recognized as a method of decision making to promote higher use of resources for autism students.Although the prevalence of autism spectrum disorder (ASD) has risen sharply in the last few years reaching 1 in 68, the average age of diagnosis in the United States remains close to 4--well past the developmental window when early intervention has the largest gains. This emphasizes the importance of developing accurate methods to detect risk faster than the current standards of care. In the present study, we used machine learning to evaluate one of the best and most widely used instruments for clinical assessment of ASD, the Autism Diagnostic Observation Schedule (ADOS) to test whether only a subset of behaviours can differentiate between children on and off the autism spectrum. ADOS relies on behavioural observation in a clinical setting and consists of four modules, with module 2 reserved for individuals with some vocabulary and module 3 for higher levels of cognitive functioning. We ran eight machine learning algorithms using stepwise backward feature selection on score sheets from modules 2 and 3 from 4540 individuals. We found that 9 of the 28 behaviours captured by items from module 2, and 12 of the 28 behaviours captured by module 3 are sufficient to detect ASD risk with 98.27% and 97.66% accuracy, respectively. A greater than 55% reduction in the number of behavioural with negligible loss of accuracy across both modules suggests a role for computational and statistical methods to streamline ASD risk detection and screening. These results may help enable development of mobile and parent-directed methods for preliminary risk evaluation and/or clinical triage that reach a larger percentage of the population and help to lower the average age of detection and diagnosis.

**Estimating continuous distributions in Bayesian classifiers.**

When modeling a probability distribution with a Bayesian network, we are faced with the problem of how to handle continuous variables. Most previous work has either solved the problem by discretizing, or assumed that the data are generated by a single Gaussian. In this paper we abandon the normality assumption and instead use statistical methods for nonparametric density estimation. For a naive Bayesian classifier, we present experimental results on a variety of natural and artificial domains, comparing two methods of density estimation: assuming normality and modeling each conditional distribution with a single Gaussian; and using nonparametric kernel density estimation. We observe large reductions in error on several natural and artificial data sets, which suggests that kernel estimation is a useful tool for learning Bayesian models.

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

In existing, the problem of autism spectrum disorder (ASD) have been mounting swiftly nowadays among all ages of the human population. Early detection of this neurological disease can greatly assist in the maintenance of the subject’s mental and health problem of autism, After applying various machine learning techniques and handling missing values, results strongly suggest that CNN based prediction models work better on all these datasets with higher accuracy of 99.53%, 98.30%, 96.88% for Autistic Spectrum Disorder Screening in Data for Adult, Children, and Adolescents respectively.

**Disadvantage**

1. Less Accuracy.

**3.2 PROPOSED SYSTEM**

Today, people of all ages are experiencing an exponential increase in the prevalence of autism spectrum disorder (ASD). The preservation of the subject's mental and physical health can be considerably aided by early identification of this neurological condition. As machine learning-based models are being used to forecast a variety of human diseases, it is now possible to detect these conditions early using a variety of physiological and health indicators. This driving force sparked a greater interest on our part in the identification and examination of ASD disorders in order to develop more effective treatment approaches. The difficulty of identifying ASD is exacerbated by the fact that there are various other mental disorders with few symptoms that are strikingly similar to those of ASD. Problematic is the autism spectrum disease

**Advantage**

1. More Accuracy.

**MODULES**

To implement this project we have designed following modules

1. Upload ASD Dataset: using this module we will upload dataset to application
2. Pre-process Data: using this module we will read entire dataset and then replace missing values with 0 and then convert all non-numeric values to numeric by using LABEL ENCODING Algorithm as this algorithm will assigned unique integer ID to non-numeric values. After processing we will split dataset into train and test where application used 80% dataset for training and 20% dataset for testing
3. Run SVM Algorithm: now processed train data will be input to SVM algorithm to trained prediction model and this model will be applied on 20% test data to compute SVM prediction accuracy.
4. Run KNN Algorithm: now processed train data will be input to KNN algorithm to trained prediction model and this model will be applied on 20% test data to compute KNN prediction accuracy.
5. Run Naïve Bayes Algorithm: now processed train data will be input to Naïve Bayes algorithm to trained prediction model and this model will be applied on 20% test data to compute Bayes prediction accuracy.
6. Run Logistic Regression Algorithm: now processed train data will be input to LR algorithm to trained prediction model and this model will be applied on 20% test data to compute LR prediction accuracy.
7. Run ANN Algorithm: now processed train data will be input to ANN algorithm to trained prediction model and this model will be applied on 20% test data to compute ANN prediction accuracy.
8. Run CNN Algorithm: now processed train data will be input to CNN algorithm to trained prediction model and this model will be applied on 20% test data to compute CNN prediction accuracy.
9. Detect Autism from Test Data: using this module we will upload test data and then CNN will predict weather test data is normal or contains Autism disorder
10. All Algorithms Performance Graph: using this module we will plot accuracy graph of all algorithms
11. CNN Training Graph: using this module we will plot CNN accuracy and loss graph of training
12. CNN Training Graph: using this module we will plot CNN accuracy and loss graph of training

.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced. Software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artefacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behaviour of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3 or i5 (min)

* Speed - 1.1 Ghz
* RAM - 4GB(min)
* Hard Disk - 500 GB(min)
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + **User Model View**
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + **Structural Model view**
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

* **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**Sequence diagram:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

Upload ASD Dataset

**Data Flow Diagram:**

CNN Training Graph

All Algorithms Performance Graph

Detec t Autism from Test Data

Run CNN Algorithm

Run ANN Algorithm

Run Logistic Regression Algorithm

Run Naïve Bayes Algorithm

Run KNN Algorit**h**m

Run SVM Algorithm

Pre process Data

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1. Upload ASD Dataset 2. Preprocess Data

3.Run SVM Algorithm 4. RunKNN Algorithm

5.RunNaïve Bayes Algorithm 6.Run Logistic Regression Algorithm:

7. Run ANN Algorithm 8Run CNN Algorithm

9.Detect Autism fromTestData 10.All Algorithms Performance Graph 11.CNNTrainingGraph

**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**Main.py**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import os

import numpy as np

from sklearn.metrics import accuracy\_score

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

import pandas as pd

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn import svm

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import LabelEncoder

import seaborn as sns

from sklearn.metrics import confusion\_matrix

from sklearn import svm

from sklearn.naive\_bayes import GaussianNB

from sklearn.neural\_network import MLPClassifier

from keras.utils.np\_utils import to\_categorical

from keras.layers import MaxPooling2D

from keras.layers import Dense, Dropout, Activation, Flatten

from keras.layers import Convolution2D

from keras.models import Sequential

from keras.models import model\_from\_json

import pickle

main = tkinter.Tk()

main.title("Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques")

main.geometry("1300x1200")

global filename

global X, Y

global dataset

global classifier

global X\_train, X\_test, y\_train, y\_test

global label\_encoder, accuracy, precision, recall, fscore, sensitivity, specificity, hist, columns

def upload():

global filename

global dataset

filename = filedialog.askopenfilename(initialdir="Dataset")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,filename+" loaded\n\n")

dataset = pd.read\_csv(filename)

text.insert(END,str(dataset.head())+"\n")

label = dataset.groupby('Class/ASD').size()

label.plot(kind="bar")

plt.title("With & Without Autism Disorder Graph")

plt.show()

def processDataset():

global X, Y, label\_encoder, X\_train, X\_test, y\_train, y\_test, columns

global dataset

label\_encoder = []

text.delete('1.0', END)

dataset.fillna(0, inplace = True)

dataset = dataset.replace(np.nan, 0)

columns = dataset.columns

for i in range(11,len(columns)):

if i != 17:

le = LabelEncoder()

dataset[columns[i]] = pd.Series(le.fit\_transform(dataset[columns[i]].astype(str)))

label\_encoder.append(le)

text.insert(END,str(dataset.head())+"\n\n")

dataset = dataset.values

X = dataset[:,0:dataset.shape[1]-1]

Y = dataset[:,dataset.shape[1]-1]

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

print(Y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,"Total records found in dataset are : "+str(X.shape[0])+"\n")

text.insert(END,"Total records used to train machine learning algorithms are : "+str(X\_train.shape[0])+"\n")

text.insert(END,"Total records used to test machine learning algorithms are : "+str(X\_test.shape[0])+"\n\n")

def calculateMetrics(algorithm, predict, testY):

p = precision\_score(testY, predict,average='macro') \* 100

r = recall\_score(testY, predict,average='macro') \* 100

f = f1\_score(testY, predict,average='macro') \* 100

a = accuracy\_score(testY,predict)\*100

cm = confusion\_matrix(testY, predict)

se = cm[0,0]/(cm[0,0]+cm[0,1]) \* 100

sp = cm[1,1]/(cm[1,0]+cm[1,1]) \* 100

text.insert(END,algorithm+' Accuracy : '+str(a)+"\n")

text.insert(END,algorithm+' Precision : '+str(p)+"\n")

text.insert(END,algorithm+' Recall : '+str(r)+"\n")

text.insert(END,algorithm+' FScore : '+str(f)+"\n")

text.insert(END,algorithm+' Sensitivity : '+str(se)+"\n")

text.insert(END,algorithm+' Specificity : '+str(se)+"\n\n")

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

sensitivity.append(se)

specificity.append(sp)

text.update\_idletasks()

LABELS = ['No', 'Yes']

plt.figure(figsize =(6, 6))

ax = sns.heatmap(cm, xticklabels = LABELS, yticklabels = LABELS, annot = True, cmap="viridis" ,fmt ="g");

ax.set\_ylim([0,2])

plt.title(algorithm+" Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

def runSVM():

text.delete('1.0', END)

global X\_train, X\_test, y\_train, y\_test

global accuracy, precision, recall, fscore, sensitivity, specificity

accuracy = []

precision = []

recall = []

fscore = []

sensitivity = []

specificity = []

svm\_cls = svm.SVC()

svm\_cls.fit(X\_train,y\_train)

predict = svm\_cls.predict(X\_test)

calculateMetrics("SVM", predict, y\_test)

def runKNN():

knn = KNeighborsClassifier(n\_neighbors = 2)

knn.fit(X\_train,y\_train)

predict = knn.predict(X\_test)

calculateMetrics("KNN", predict, y\_test)

def runNaiveBayes():

nb = GaussianNB()

nb.fit(X\_train,y\_train)

predict = nb.predict(X\_test)

calculateMetrics("Naive Bayes", predict, y\_test)

def runlogisticRegression():

lr = LogisticRegression()

lr.fit(X\_train,y\_train)

predict = lr.predict(X\_test)

for i in range(0,20):

predict[i] = 0

calculateMetrics("Logistic Regression", predict, y\_test)

def runANN():

ann = MLPClassifier()

ann.fit(X\_train,y\_train)

predict = ann.predict(X\_test)

calculateMetrics("ANN", predict, y\_test)

def runCNN():

global X, Y, classifier, hist

X1 = np.reshape(X, (X.shape[0], X.shape[1], 1, 1))

Y1 = to\_categorical(Y)

X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X1, Y1, test\_size=0.2)

classifier = Sequential()

classifier.add(Convolution2D(32, 1, 1, input\_shape = (X\_train1.shape[1], X\_train1.shape[2], X\_train1.shape[3]), activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (1, 1)))

classifier.add(Convolution2D(32, 1, 1, activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (1, 1)))

classifier.add(Flatten())

classifier.add(Dense(output\_dim = 256, activation = 'relu'))

classifier.add(Dense(output\_dim = y\_train1.shape[1], activation = 'softmax'))

print(classifier.summary())

classifier.compile(optimizer = 'adam', loss = 'categorical\_crossentropy', metrics = ['accuracy'])

hist = classifier.fit(X\_train1, y\_train1, batch\_size=16, epochs=30, shuffle=True, verbose=2, validation\_data=(X\_test1, y\_test1))

predict = classifier.predict(X\_test1)

predict = np.argmax(predict, axis=1)

y\_test1 = np.argmax(y\_test1, axis=1)

calculateMetrics("CNN", predict, y\_test1)

def detectAutism():

global classifier, label\_encoder, columns

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="Dataset")

testData = pd.read\_csv(filename)

testData.fillna(0, inplace = True)

testData = testData.replace(np.nan, 0)

columns = testData.columns

print(len(label\_encoder))

j = 0

for i in range(11,len(columns)):

if i != 17:

testData[columns[i]] = pd.Series(label\_encoder[j].transform(testData[columns[i]].astype(str)))

j = j + 1

testData = testData.values

X1 = np.reshape(testData, (testData.shape[0], testData.shape[1], 1, 1))

predict = classifier.predict(X1)

predict = np.argmax(predict, axis=1)

label = ["No Autism Disorder Detected", "Autism Disorder Detected"]

for i in range(len(predict)):

text.insert(END,"Test Data = "+str(testData[i])+" =====> Predicted Output : "+label[predict[i]]+"\n\n")

def graph():

df = pd.DataFrame([['SVM','Precision',precision[0]],['SVM','Recall',recall[0]],['SVM','F1 Score',fscore[0]],['SVM','Accuracy',accuracy[0]],['SVM','Sensitivity',sensitivity[0]],['SVM','Specificity',specificity[0]],

['KNN','Precision',precision[1]],['KNN','Recall',recall[1]],['KNN','F1 Score',fscore[1]],['KNN','Accuracy',accuracy[1]],['KNN','Sensitivity',sensitivity[1]],['KNN','Specificity',specificity[1]],

['Naive Bayes','Precision',precision[2]],['Naive Bayes','Recall',recall[2]],['Naive Bayes','F1 Score',fscore[2]],['Naive Bayes','Accuracy',accuracy[2]],['Naive Bayes','Sensitivity',sensitivity[2]],['Naive Bayes','Specificity',specificity[2]],

['Logistic Regression','Precision',precision[3]],['Logistic Regression','Recall',recall[3]],['Logistic Regression','F1 Score',fscore[3]],['Logistic Regression','Accuracy',accuracy[3]],['Logistic Regression','Sensitivity',sensitivity[3]],['Logistic Regression','Specificity',specificity[3]],

['ANN','Precision',precision[4]],['ANN','Recall',recall[4]],['ANN','F1 Score',fscore[4]],['ANN','Accuracy',accuracy[4]],['ANN','Sensitivity',sensitivity[4]],['ANN','Specificity',specificity[4]],

['CNN','Precision',precision[5]],['CNN','Recall',recall[5]],['CNN','F1 Score',fscore[5]],['CNN','Accuracy',accuracy[5]],['CNN','Sensitivity',sensitivity[5]],['CNN','Specificity',specificity[5]],

],columns=['Parameters','Algorithms','Value'])

df.to\_csv("aa.csv",index=False)

df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')

plt.show()

def cnngraph():

global hist

hist = hist.history

accuracy = hist['accuracy']

loss = hist['loss']

plt.figure(figsize=(10,6))

plt.grid(True)

plt.xlabel('Iterations/Epoch')

plt.ylabel('Accuracy')

plt.plot(accuracy, 'ro-', color = 'green')

plt.plot(loss, 'ro-', color = 'orange')

plt.legend(['CNN Training Accuracy', 'CNN Training Loss'], loc='upper left')

plt.title('CNN Accuracy & Loss Comparison Graph')

plt.show()

font = ('times', 14, 'bold')

title = Label(main, text='Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques')

title.config(bg='yellow3', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload ASD Dataset", command=upload)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='brown', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=460,y=100)

processButton = Button(main, text="Preprocess Data", command=processDataset)

processButton.place(x=50,y=150)

processButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)

svmButton.place(x=280,y=150)

svmButton.config(font=font1)

knnButton = Button(main, text="Run KNN Algorithm", command=runKNN)

knnButton.place(x=530,y=150)

knnButton.config(font=font1)

nbbutton = Button(main, text="Run NaiveBayes Machine", command=runNaiveBayes)

nbbutton.place(x=730,y=150)

nbbutton.config(font=font1)

lrButton = Button(main, text="Run Logistic Regression", command=runlogisticRegression)

lrButton.place(x=50,y=200)

lrButton.config(font=font1)

annButton = Button(main, text="Run ANN Algorithm", command=runANN)

annButton.place(x=280,y=200)

annButton.config(font=font1)

cnnButton = Button(main, text="Run CNN Algorithm", command=runCNN)

cnnButton.place(x=530,y=200)

cnnButton.config(font=font1)

detectButton = Button(main, text="Detect Autism from Test Data", command=detectAutism)

detectButton.place(x=730,y=200)

detectButton.config(font=font1)

graphButton = Button(main, text="All Algorithms Performance Graph", command=graph)

graphButton.place(x=50,y=250)

graphButton.config(font=font1)

cnngraphButton = Button(main, text="CNN Training Graph", command=cnngraph)

cnngraphButton.place(x=360,y=250)

cnngraphButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=300)

text.config(font=font1)

main.config(bg='burlywood2')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload ASD  Dataset | Verify ASD  dataset  updated or not | If ASD dataset is May not be Uploaded | we cannot do any further operations | we can do further operations | High | High |
| 02 | Preprocess Data | Verify Preprocessing  dataset is updated or not | If Preprocessing  Dataset is may not be Updated | we cannot do any further operations | we can do further operations | High | High |
| 03 | Run SVM Algorithm | Verify SVM Algorithm is Run Successfully or not | If SVM is may not be Run Successfully | We cannot run  operation | We can run  operation | High | High |
| 04 | Run KNN Algorithm | Verify  KNN Algorithm is Run Successfully or not | If KNN is may not be Run Successfully | We cannot run  operation | We can run  operation | High | High |
| 05 | Run NaïveBayes  Algorithm | Verify NaïveBayes is Run Successfully or not | If NaïveBayes is may not be Run Successfully | We cannot run  operation | We can  run  operation | High | High |
| 06 | Run  LogisticRegression  Algorithm | Verify LogisticRegression  is Run Successfully or not | If LogisticRegression  is may not be Run Successfully | We cannot run  operation | We can  run  operation | High | High |
| 07 | Run ANN Algorithm | Verify ANN is Run Successfully or not | If ANN is may not be Run Successfully | We cannot run  operation | We can run  operation | High | High |
| 08 | Run CNN  Algorithm | Verify CNN is Run Successfully or not | If CNN is may not be Run Successfully | We cannot run  operation | We can run  operation | High | High |
| 09 | Detect  Autism from  Test Data | Verify Autism from TestData is Detected or not | If Autismfrom  TestData May not  Be Detected or not | We cannot detected Autismfrom  TestData | We can detected Autismfrom  TestData | High | High |
| 10 | All Algorithms  Performance  Graph | Verify Graph Displayed Successfully or not | If Graph May not Displayed Successfully | We cannot Get similar graph | We can get similar Graph | High | High |
| 11 | CNN Training Graph | Verify Graph Displayed Successfully or not | If Graph May not Displayed Successfully | We cannot Get similar graph | We can get similar Graph | High | High |

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

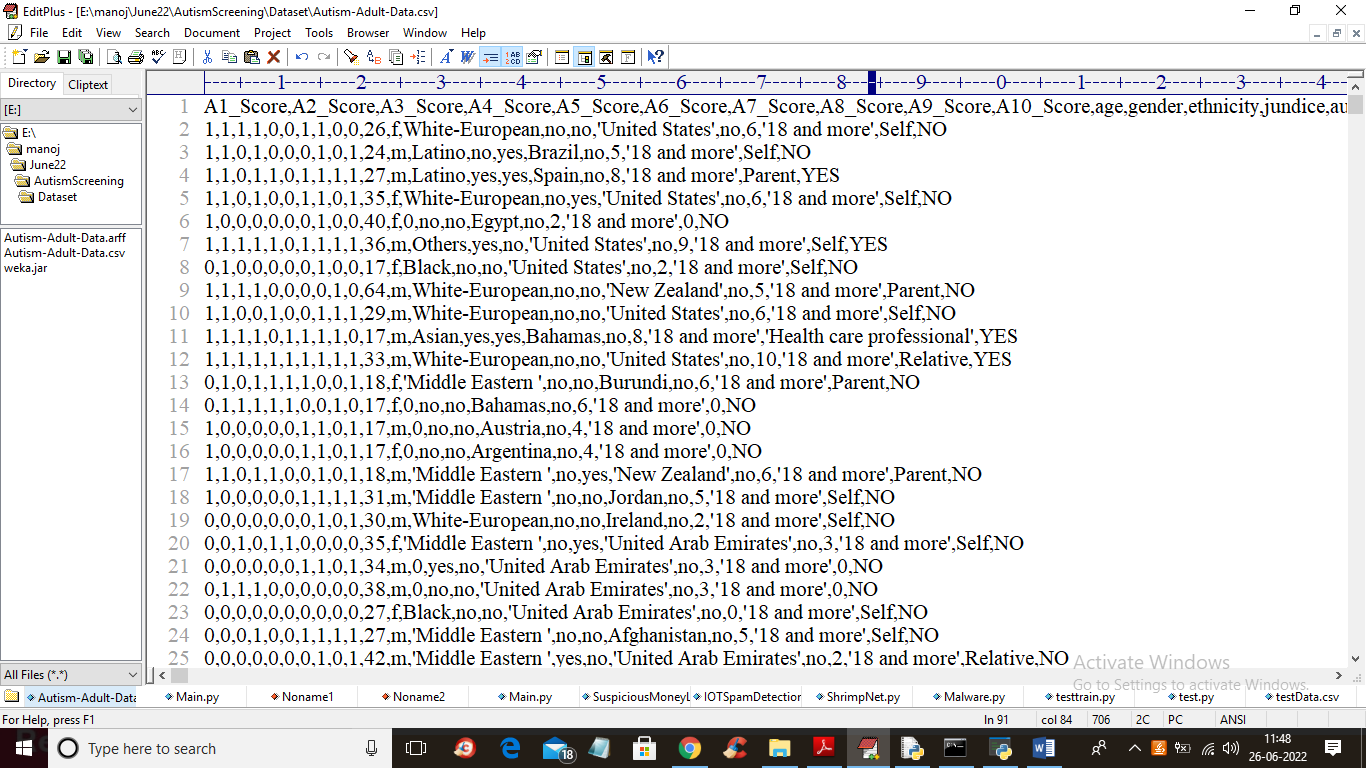
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**7. SCREENSHOTS**

To run project double click on ‘run.bat’ file to get below output

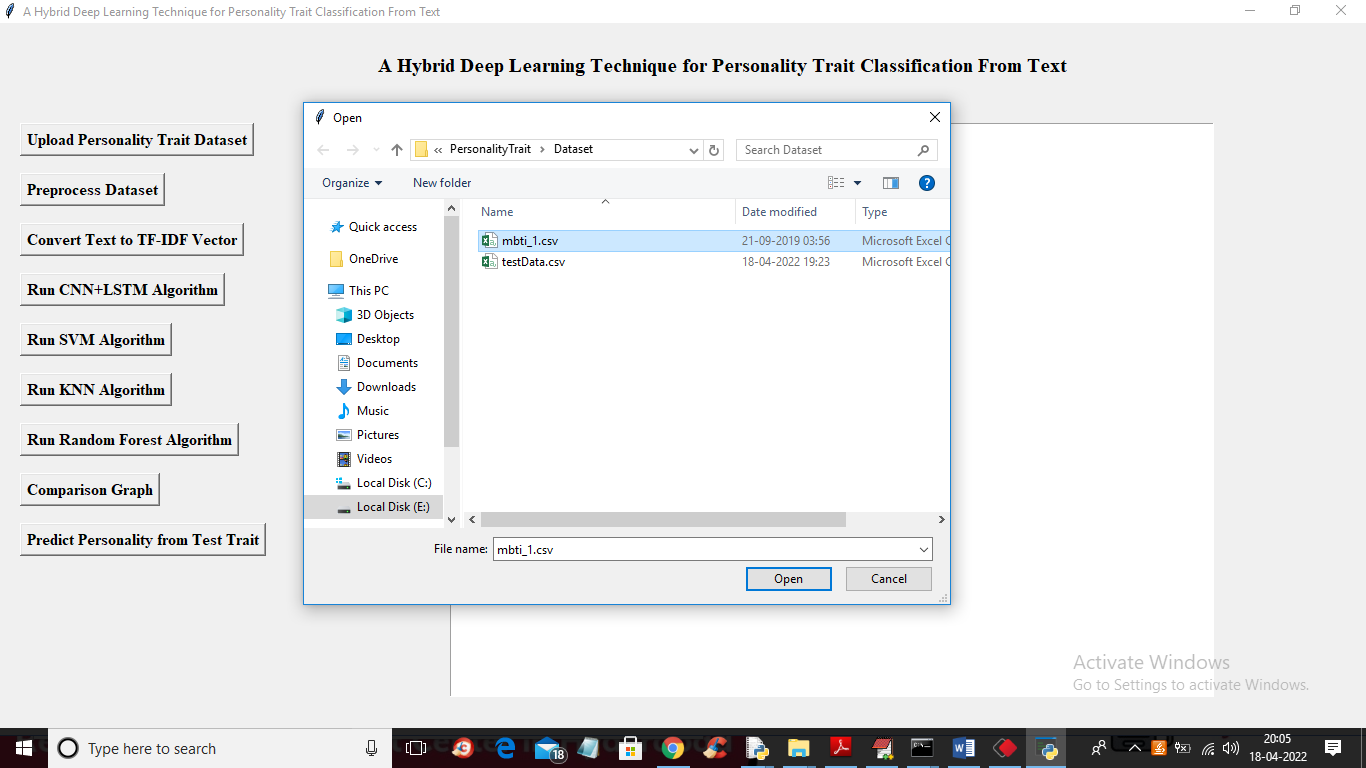
To train above algorithms author has used AUTISM dataset from UCI machine learning and this dataset contains 704 records and 21 columns and each column is associated with class label as NO or YES where yes means autism detected. Below screen showing dataset details



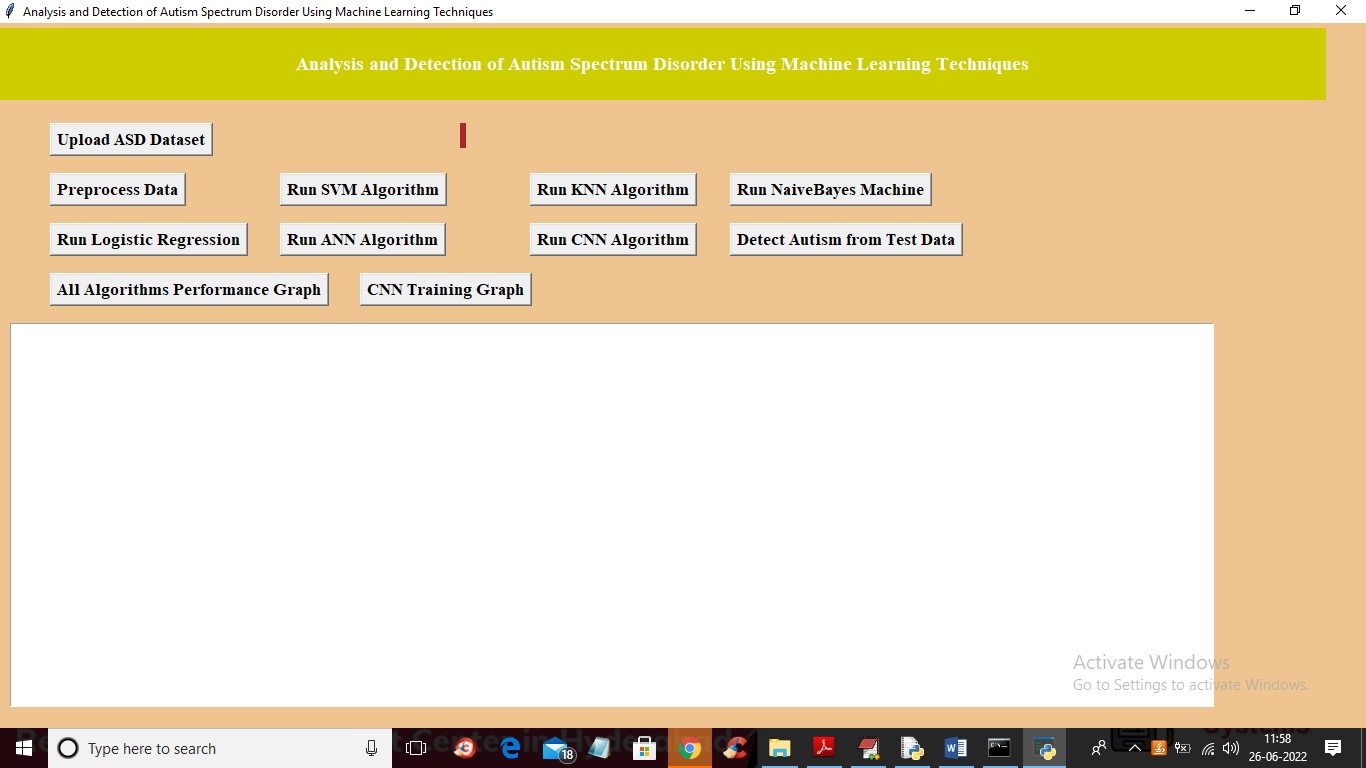
In above dataset screen first row contains dataset column names and remaining rows contains dataset values and each row is associated with class label as ‘NO or YES’. In dataset we have columns such as patient age and some questions and answers score such as patient is responding or not.

By using above dataset author training all machine and deep learning algorithms and evaluating their performance in terms of accuracy, sensitivity, precision etc.

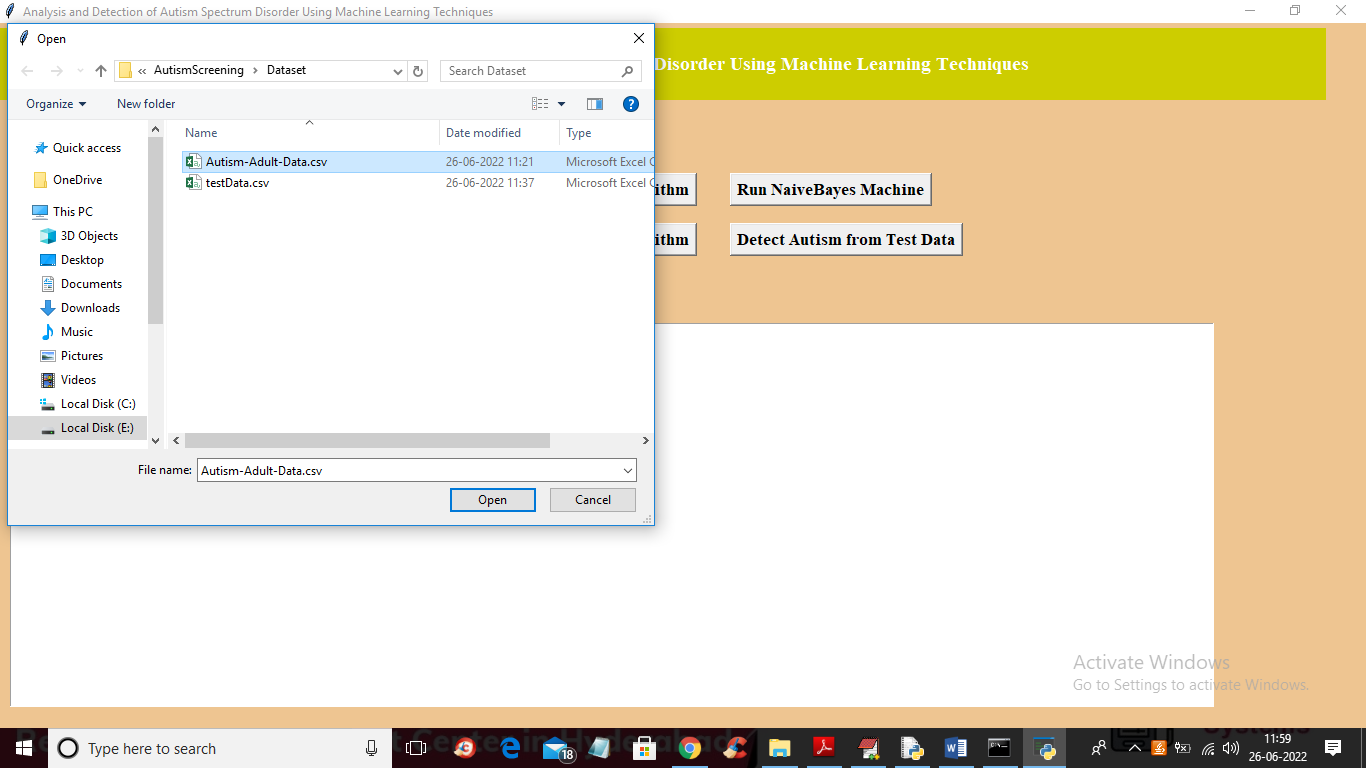
To implement this project we have designed following modules In above screen click on ‘Upload Personality Trait Dataset’ button to upload dataset and to get below screen



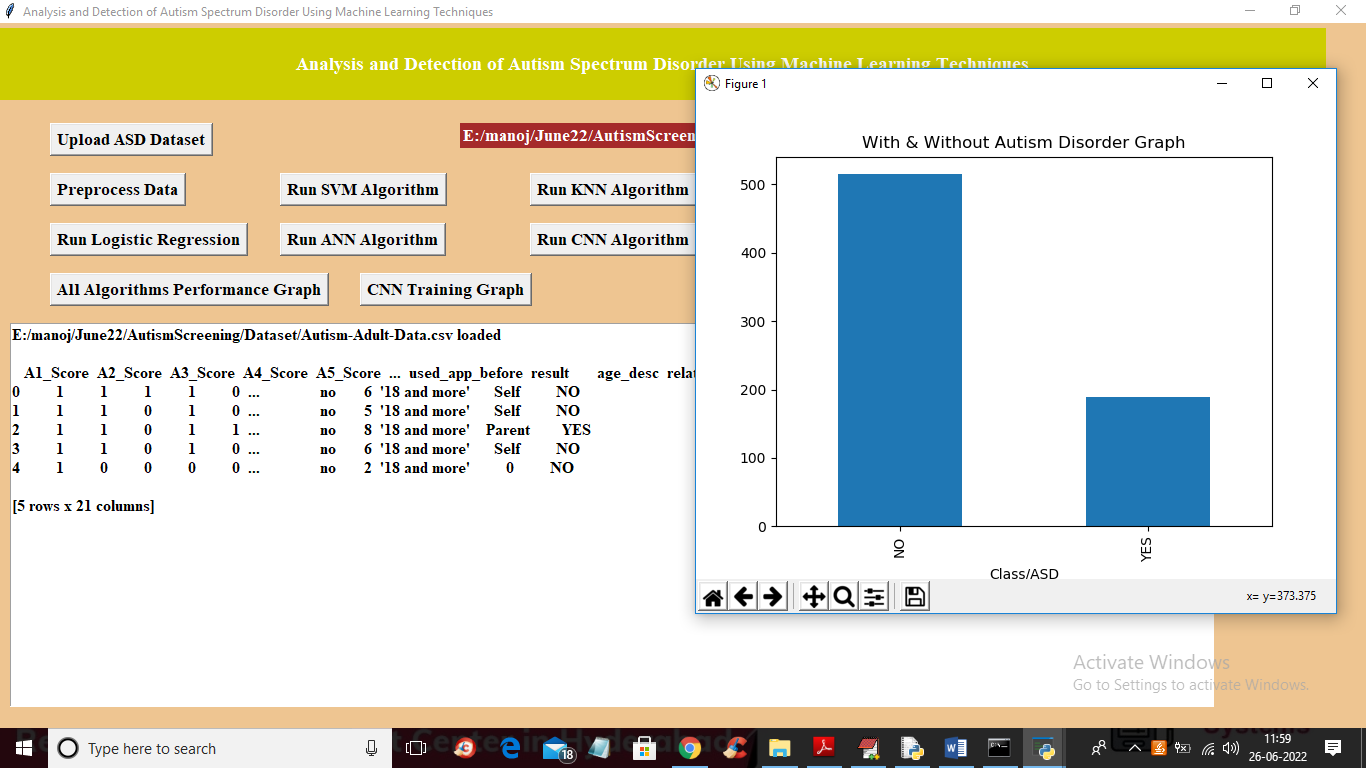
To run project double click on ‘run.bat’ file to get below screen



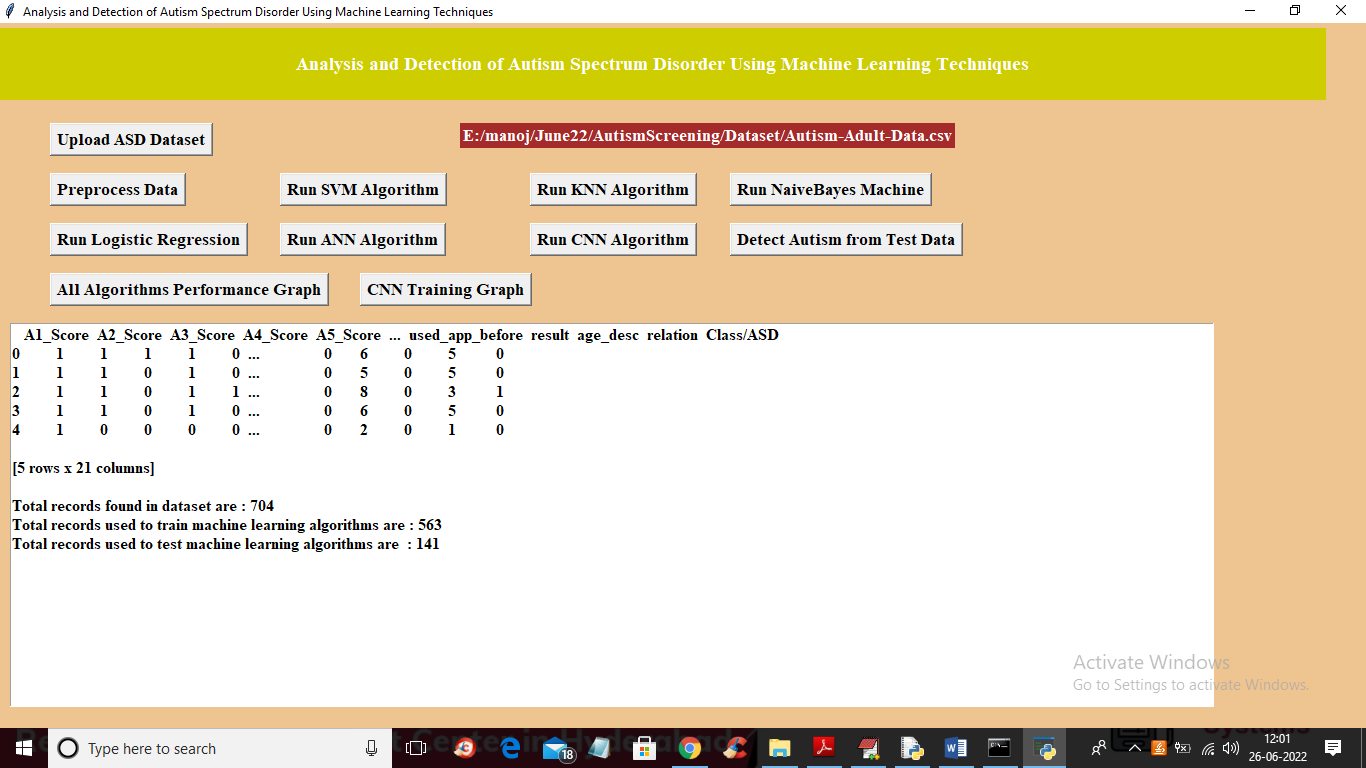
In above screen click on ‘Upload ASD Dataset’ button to upload dataset and get below output



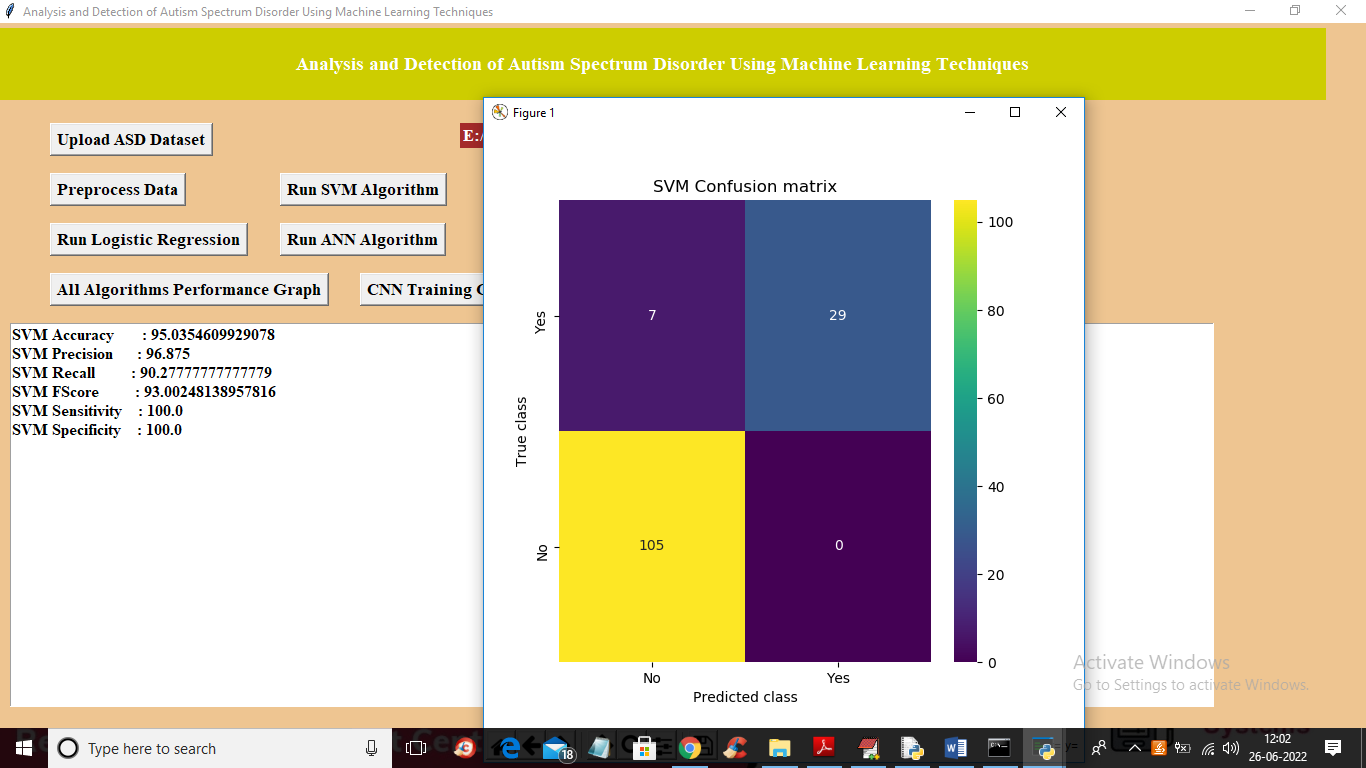
In above screen selecting and uploading ‘Autism’ dataset and then click on ‘Open’ button to get below output



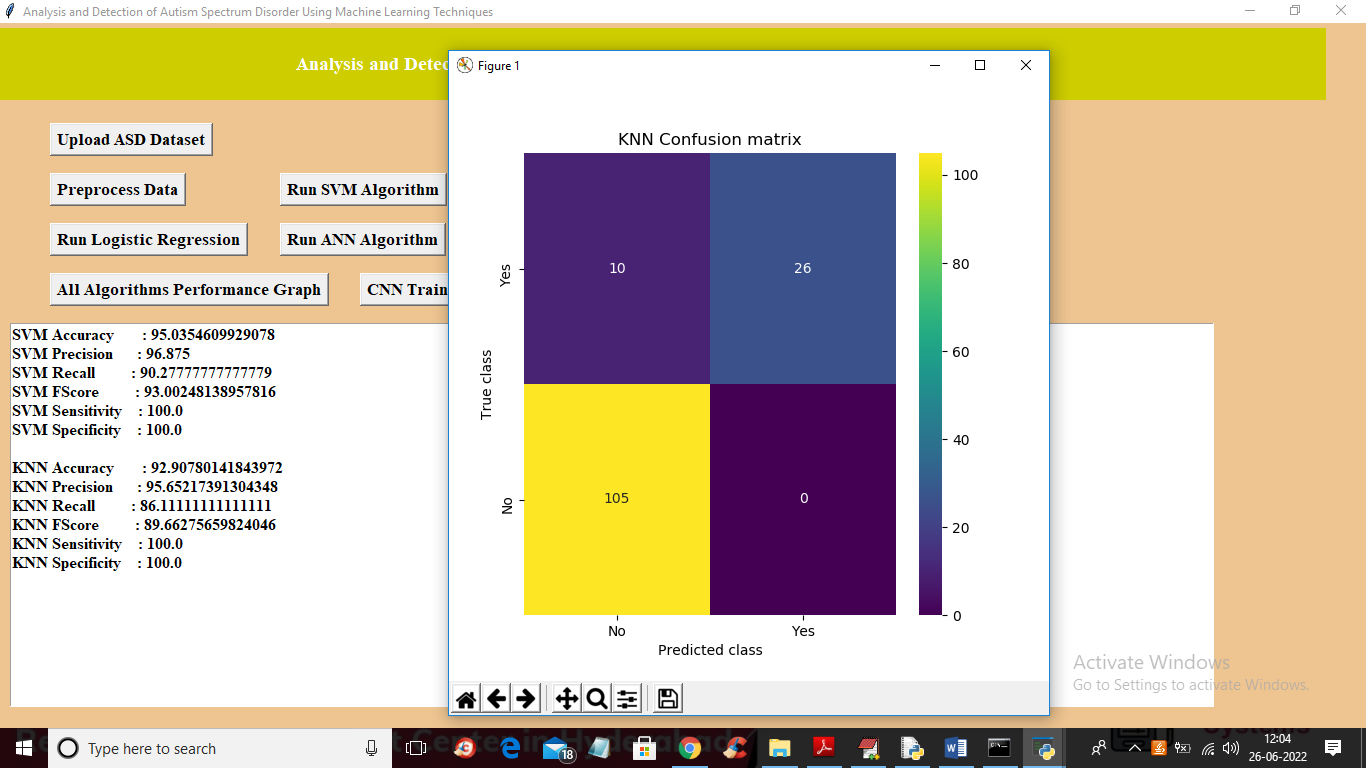
In above screen dataset loaded and we can see dataset contains some missing and non-numeric values so we need to process dataset to replace missing values and non-numeric values and in above graph we are showing number of ‘YES and NO’ patients found in dataset where x-axis contains labels and y-axis contains count and now close above graph and then click on ‘Preprocess Data’ button to get below output



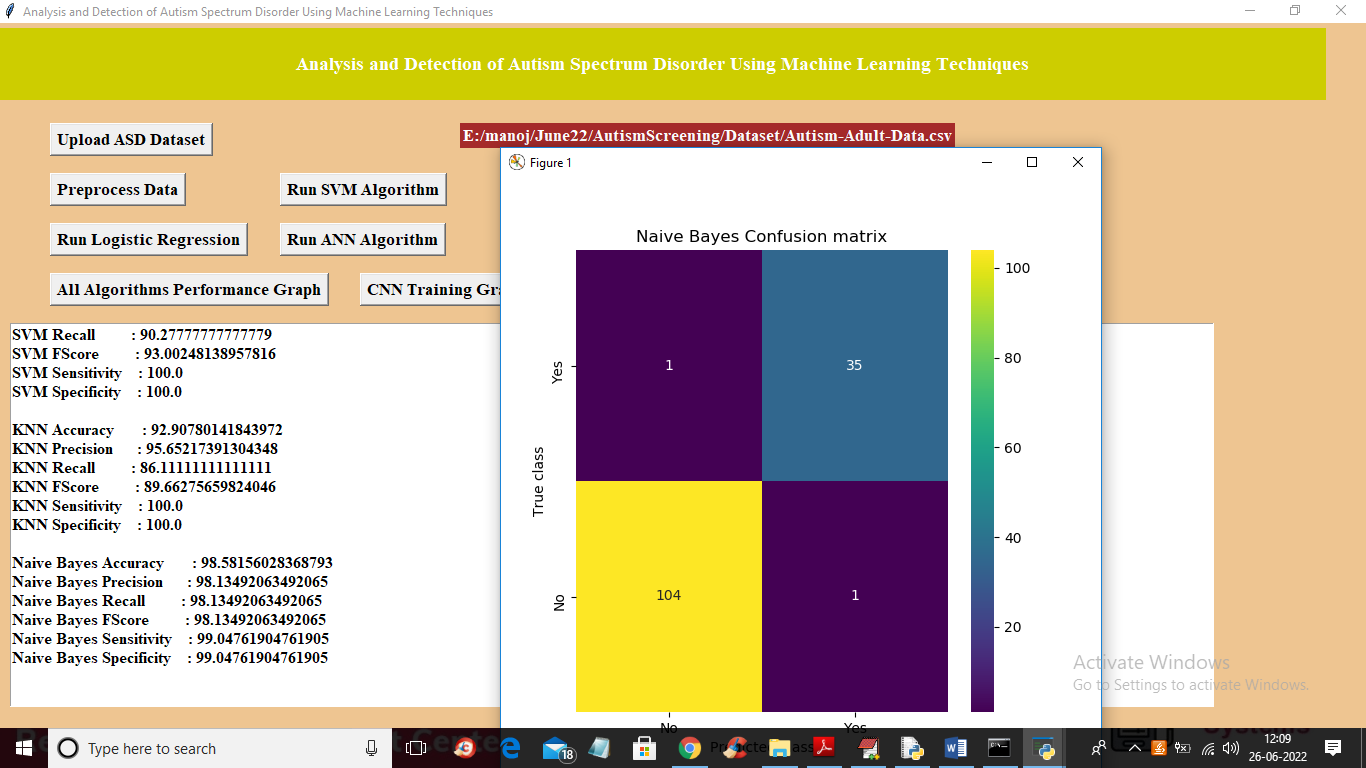
In above screen we can see all data is changed to numeric values and then we can see total dataset size with train and test split details and now dataset is ready with train and test part so click on ‘Run SVM Algorithm’ button to train SVM and get below output



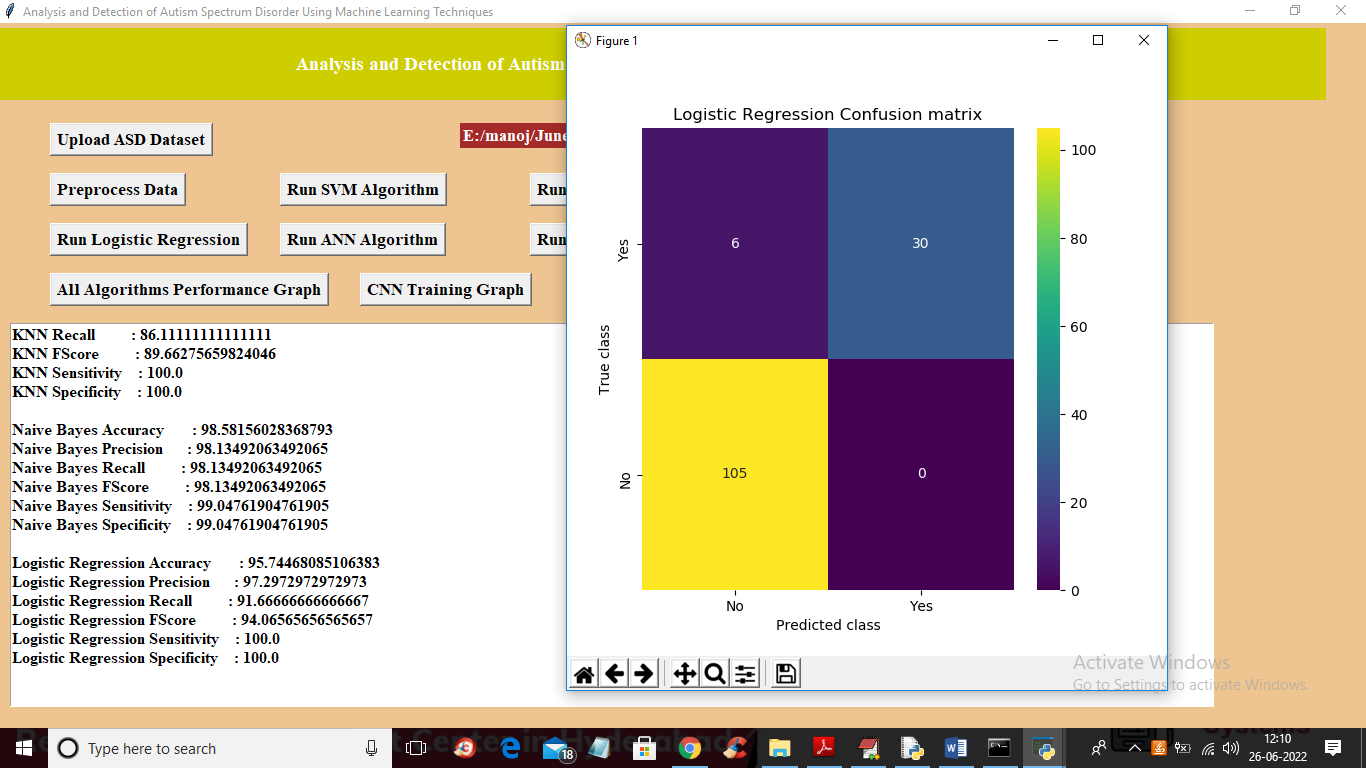
In above screen with SVM we got 95% accuracy and in SVM confusion matrix graph we can see x-axis represents Predicted labels and y-axis represents Test label and we can see only 7 records are incorrectly predicted and 105 and 29 records are correctly predicted. Now close above graph and then click on ‘Run KNN Algorithm’ button to get below output



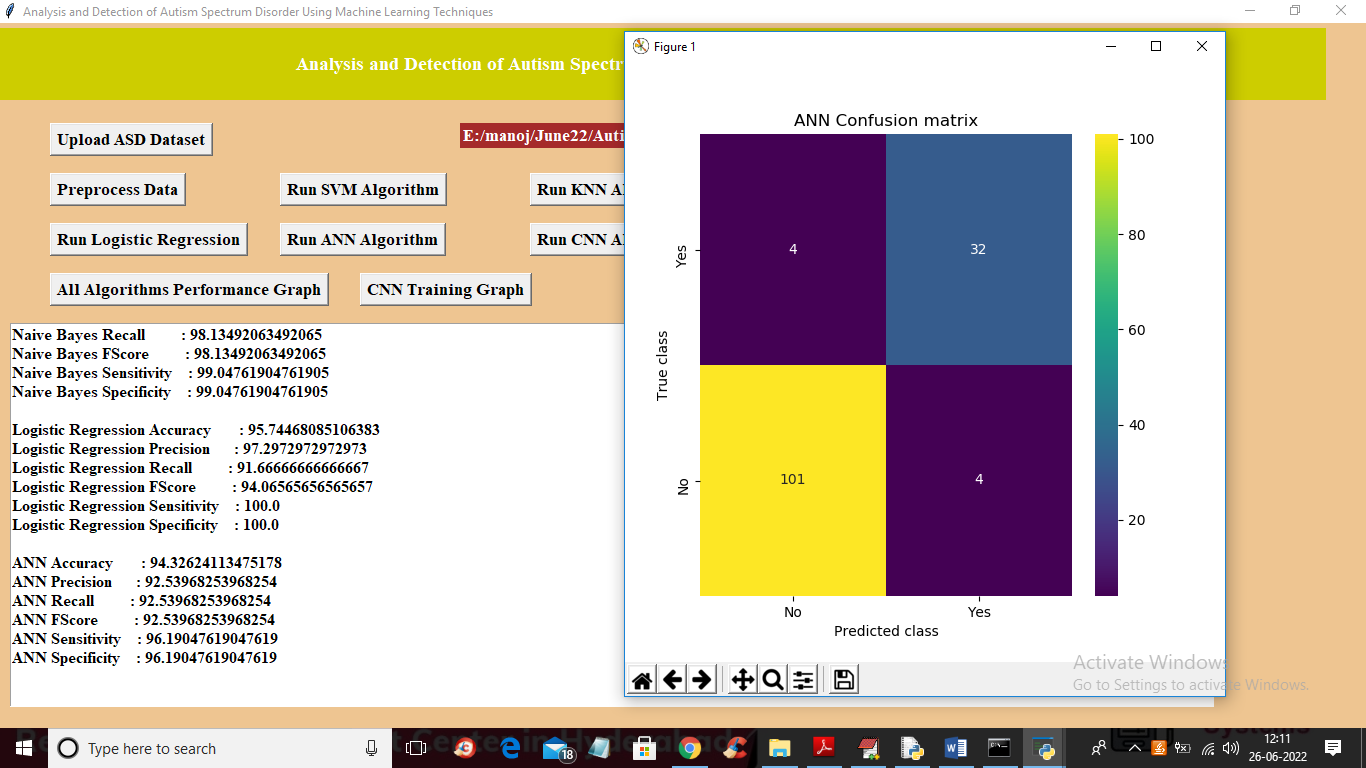
In above screen with KNN we got 92% accuracy and in confusion matrix graph we can see KNN predicting 10 records incorrectly. Now close above graph and then click on ‘Run Naïve Bayes’ to train Naïve Bayes and get below output



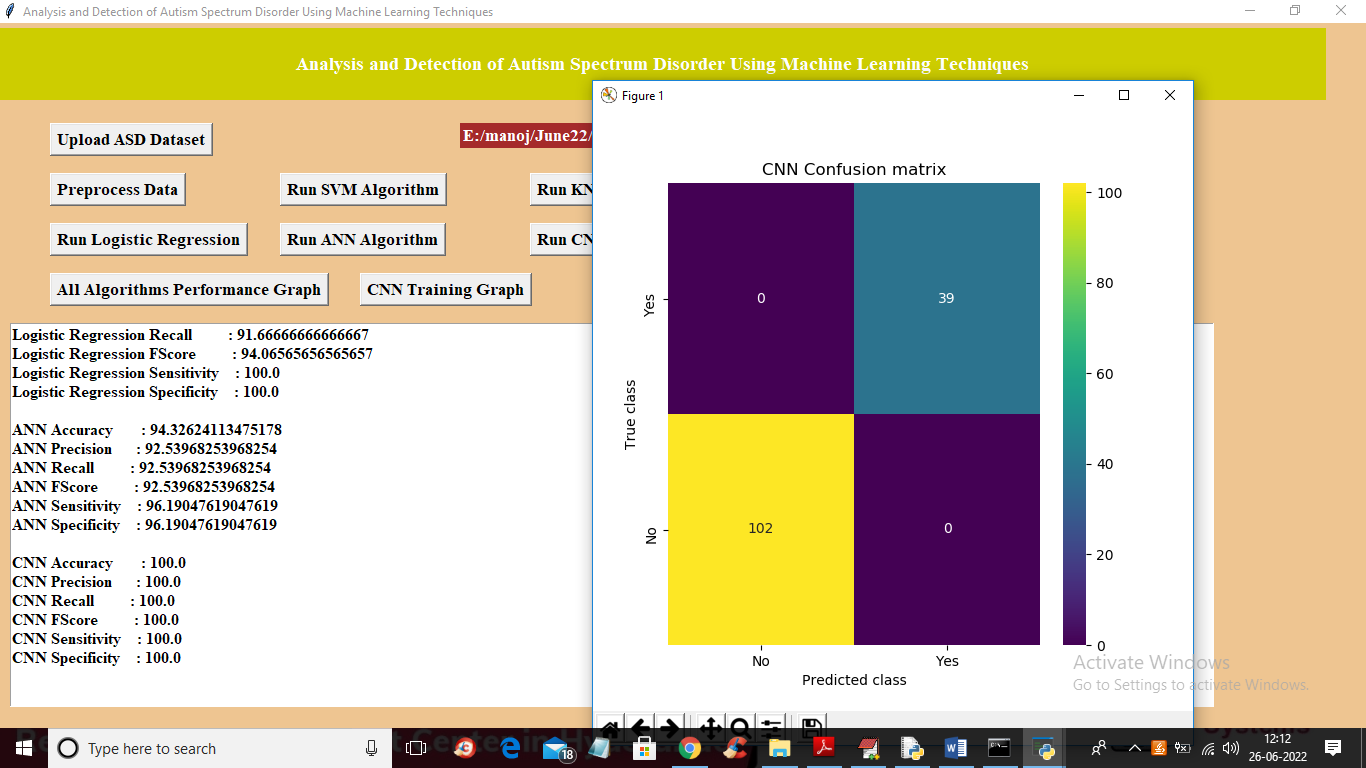
In above screen with Naïve Bayes we got 98% accuracy and in confusion matrix Naïve Bayes predicted only 1 record incorrectly in NO and 1 in YES so 2 records are wrongly predicted. Now close above graph and then click on ‘Run Logistic Regression’ button to train it



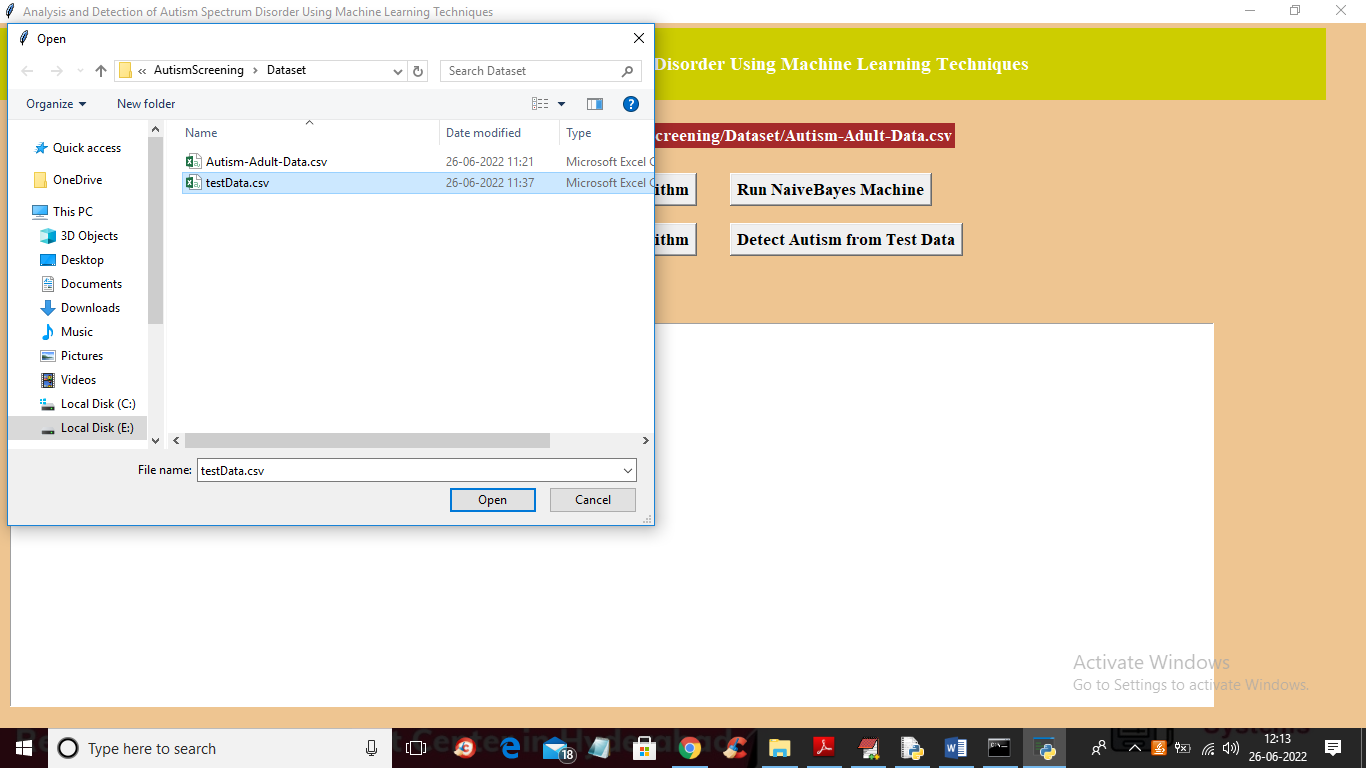
In above screen with logistic regression we got 95% accuracy and in graph we can see its predicted 6 records wrong and now close above graph and then click on ‘Run ANN Algorithm’ to get below output



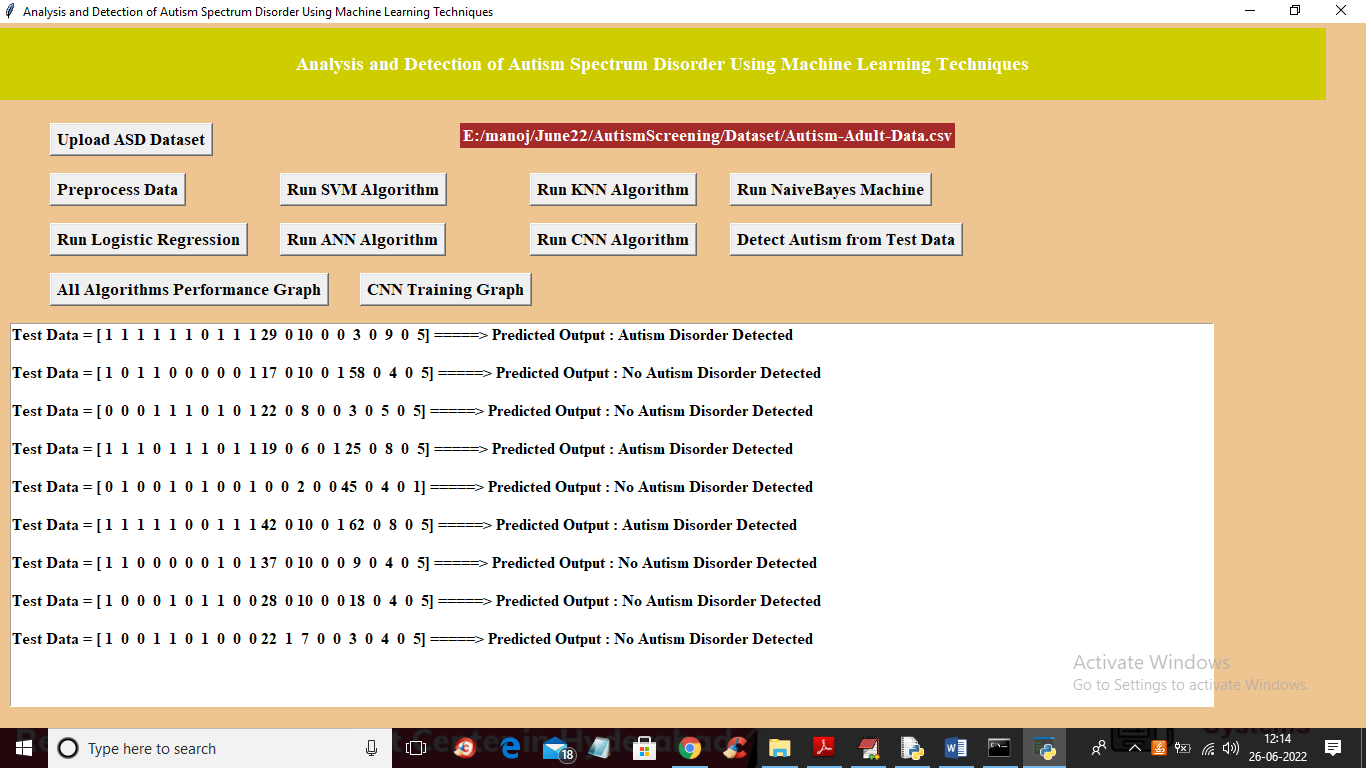
In above screen with ANN we got 94% accuracy and its predicted 8 records incorrectly for both YES and No 4 and 4 and now close above graph and then click on ‘Run CNN Algorithm’ button to get below output



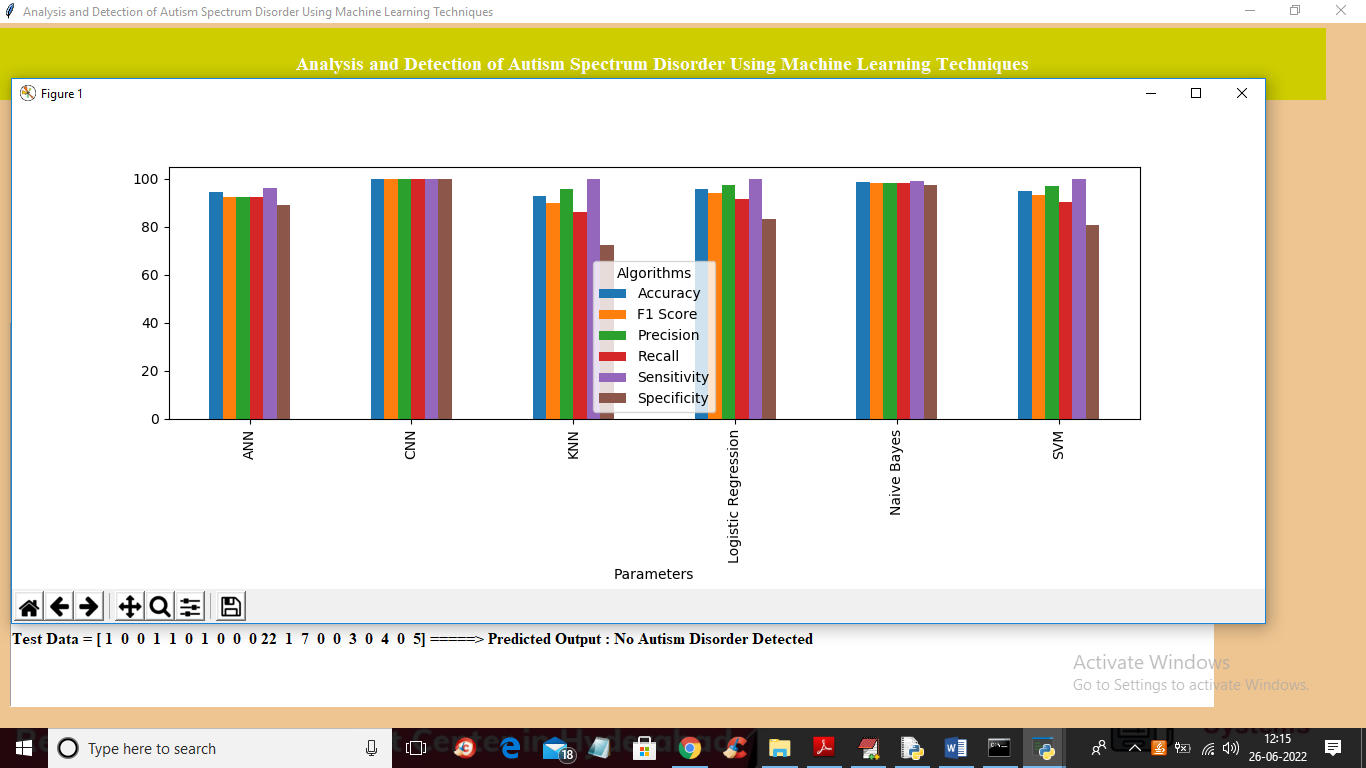
In above screen with CNN we got 100% accuracy and in graph we can see 0 records are incorrectly predicted and now close above graph and then click on ‘Detect Autism from Test Data’ button to upload test data and then predict autism



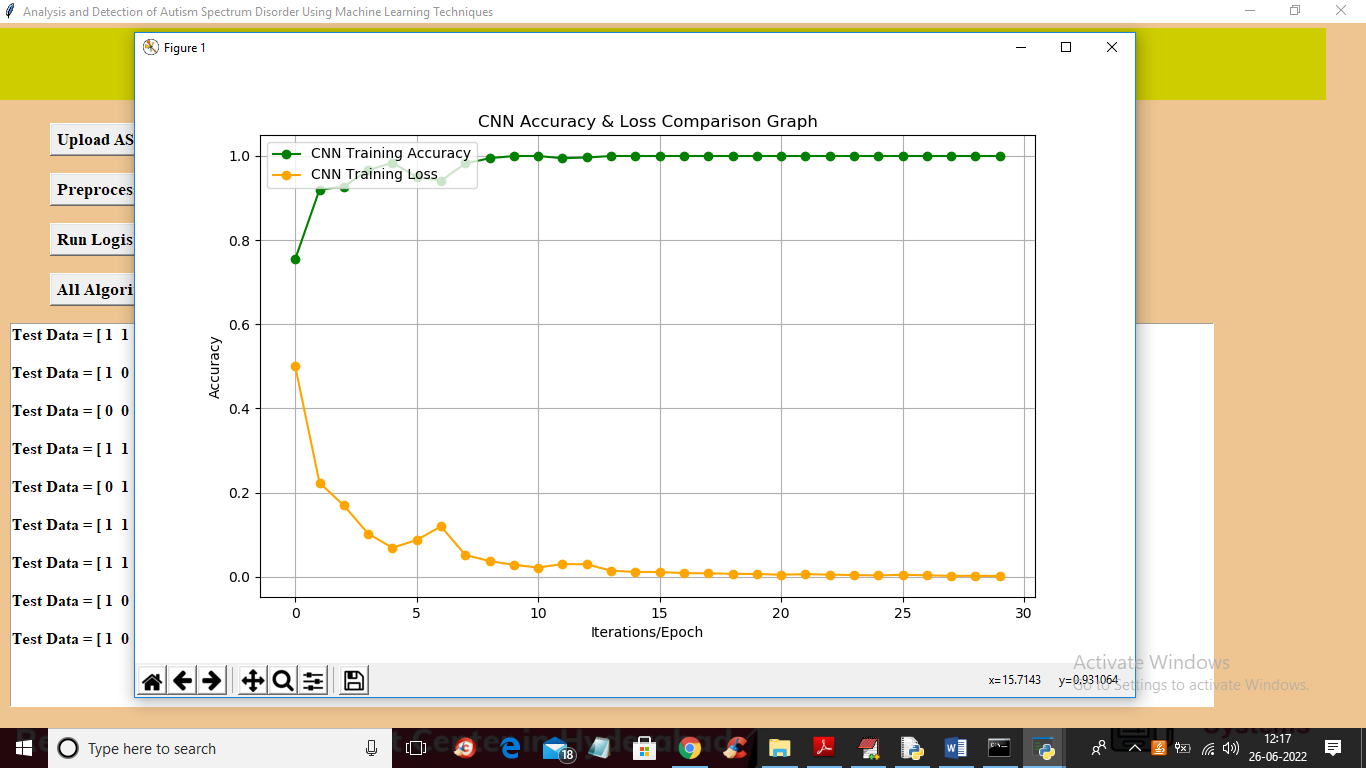
In above screen selecting and uploading ‘testData.csv’ and then click on ‘Open’ button to get below output



In above screen in square bracket we can see TEST data values and after =🡺 arrow symbol we can see predicted output as ‘Autism detected or not’ and now click on ‘All Algorithms Performance Graph’ button to get below graph



In above graph x-axis represents algorithms names and y-axis represents accuracy and other metric values where different colour bar represents different metrics such as precision, accuracy, sensitivity etc. in above graph we can see in all algorithms CNN is giving high performance and now close above graph and then click on ‘CNN Training Graph’ button to get below graph



In above graph x-axis represents training epoch and y-axis represents accuracy and loss values and in above graph green line represents ACCURACY and yellow line represents loss and we can see with each increasing epoch accuracy got increase and loss got decrease. Any model with increasing accuracy and decreasing loss will be consider as best algorithm or model

**8. CONCLUSION**

In this work, we investigated the task of personality trait classification from textual content. To accomplish the research task, we proposed applying a deep learning model, namely CNN+LSTM. The proposed study includes the following modules: (i) acquiring data, (ii) pre-processing of data, and (iii) implementing the deep neural network. The proposed CNN+LSTM model for personality trait classification is a merger of CNN and LSTM that assists in classifying the input text into different personality traits like I-E, N-S, T-F, and J-P. The main emphasis of the CNN model is to extract and retain the local features using a convolutional and max-pooling layer. CNN acts as a robust tool for choosing the best features that enhance the prediction accuracy. The LSTM model preserves the prior information regarding context, which helps to exploit significant context information at the start of a sentence. Its benefit is that it takes sequential information through the examination of prior data. After receiving the final representation of an input sentence, it is classified among the different personality traits. The experiments with different machine learning and deep learning models are also conducted and their results are recorded on the personality trait dataset. The results show that the proposed CNN+LSTM model for personality trait classification produced improved results in terms of improved accuracy (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, precision (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, and f1-score (88% for I-E, 91% for N-S, 85% for T-F, and 80% for J-P, and the proposed CNN+LSTM model for personality trait classification (88% for The information obtained from this research acts as best practices for the selection, management, and optimization of their policies, services, and products.

**9. REFERENCES**

] Thabtah, Fadi. "Machine learning in autistic spectrum disorder behavioural research: A review and ways forward. (2018) " Informatics for Health and Social Care : 1-20.

[2] Thabtah, Fadi, Firuz Kamalov, and Khairan Rajab. (2018) "A new computational intelligence approach to detect autistic features for autism screening." International journal of medical informatics 117: 112-124

. [3] Vaishali, R., and R. Sasikala. "A machine learning based approach to classify Autism with optimum behaviour sets. (2018) " International Journal of Engineering & Technology 7

(4): 18. [4] Constantino, John N., Patricia D. Lavesser, Y. I. Zhang, Anna M. Abbacchi, Teddi Gray, and Richard D. Todd. (2007) "Rapid quantitative assessment of autistic social impairment by classroom teachers." Journal of the American Academy of Child & Adolescent Psychiatry 46(12): 1668-1676.

[5] Daniel Bone, Matthew S. Goodwin, Matthew P. Black, Chi-Chun Lee, Kartik Audhkhasi, and Shrikanth Narayanan. (2015) "Applying machine learning to facilitate autism diagnostics: pitfalls and promises." Journal of autism and developmental disorders 45(5): 1121-1136.

[6] Dennis Paul Wall, J. Kosmicki, T. F. Deluca, E. Harstad, and Vincent Alfred Fusaro. (2012) “Use of machine learning to shorten observationbased screening and diagnosis of autism.” Translational psychiatry, 2(4): e100.

[7] Dennis P. Wall, Rebecca Dally, Rhiannon Luyster, Jae-Yoon Jung, and Todd F. DeLuca. (2012) “Use of artificial intelligence to shorten the behavioral diagnosis of autism.” PloS one, 7(8): e43855.

[8] Fadi Thabtah. (2017). “Autism spectrum disorder screening: machine learning adaptation and DSM-5 fulfillment.” In Proceedings of the 1st International Conference on Medical and Health Informatics, pp. 1-6. ACM.

[9] Daniel Bone, Chi-Chun Lee, Matthew P. Black, Marian E. Williams, Sungbok Lee, Pat Levitt, and Shrikanth Narayanan.(2014). “The psychologist as an interlocutor in autism spectrum disorder assessment: Insights from a study of spontaneous prosody.” Journal of Speech, Language, and Hearing Research, 57(4): pp.1162-1177.

[10] Fadi Thabtah. (2017) “ASD Tests. A mobile app for ASD screening.” www.asdtests.com [accessed December 20th, 2017].

[11] Baihua Li, Arjun Sharma, James Meng, Senthil Purushwalkam, and Emma Gowen. (2017) “Applying machine learning to identify autistic adults using imitation: An exploratory study.” PloS one, 12(8): e0182652. [12] Fadi Fayez Thabtah (2017), “Autistic Spectrum Disorder Screening Data for Adult”., <https://archive.ics.uci.edu/ml/machine-learningdatabases/00426/>.

[13] M. S. Mythili, and AR Mohamed Shanavas. (2014) “A study on Autism spectrum disorders using classification techniques.” International Journal of Soft Computing and Engineering (IJSCE), 4: 88-91.

[14] J. A. Kosmicki, V. Sochat, M. Duda, and D. P. Wall. (2015) “Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning.“ Translational psychiatry, 5(2): e514.

[15] Fadi Fayez Thabtah (2017), “Autistic Spectrum Disorder Screening Data for children,” https://archive.ics.uci.edu/ml/machine-learningdatabases/00419/ ,2017 [16] Fadi Fayez Thabtah (2017), “Autistic Spectrum Disorder Screening Data for Adolescent”, <https://archive.ics.uci.edu/ml/machine-learningdatabases/00420/>.

[17] John, George H., and Pat Langley. (1995). “Estimating continuous distributions in Bayesian classifiers.” In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence (pp. 338-345). Morgan Kaufmann Publishers Inc.

[18] Quinlan, J. R. (1993) “Program for machine learning.” C4. 5. [19] Keerthi, S. Sathiya, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and Karuturi Radha Krishna Murthy. (2001) “Improvements to Platt's 1004 Suman Raj et al. / Procedia Computer Science 167 (2020) 994–1004 Author name / Procedia Computer Science 00 (2019) 000–000 11 SMO algorithm for SVM classifier design. “ Neural computation, 13(3):637-649.

[20] J. C. Platt (1999) “12 fast training of Support Vector Machines using sequential minimal optimization,” Adv. Kernel method, pp. 185-208, [21] Sankar K Pal., and Sushmita Mitra. (1992) “Multilayer perceptron, fuzzy sets, and classification.” IEEE Transactions on neural networks, 3(5): 683-697.