TECHNO INTERNATIONAL NEW TOWN

(Formerly known as Techno India College of Technology)

Block-DG, Action Area 1, New Town, Kolkata -700156, West Bengal, India

Department of Computer Science & Engineering

Eight Semester Project-III Report (PROJ-CS881)

AI-Enhanced Diagnosis of Autism Spectrum Disorder Using Support Vector Classifier: A Data-Driven Approach

Prepared by

Anirban Pal (Roll No: -18700120122)

Bishal Kumar Ghosh (Roll No: -18700120100)

Under the Guidance of

Prof. Chinmoy Kar

Batch: - 2020-2024   Semester :8th (2024 –EVEN)   Year : January 2024 – June 2024

Stream: - Computer Science & Engineering   Year of Study: 4th

Affiliated to

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

(FORMERLY KNOWN AS WEST BENGAL UNIVERSITY OF TECHNOLOGY)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Prof. Chinmoy Kar of the Department of Computer Science & Engineering, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not least we convey our gratitude to all the teachers for providing us with the technical skill that will always remain our asset and to all non-teaching staff for the cordial support they offered.

Place: Techno International New Town

Date: 06/06/2024

Anirban Pal
(Roll No: - 18700120122)

Bishal Kumar Ghosh
(Roll No: - 1870120100)

Department of Computer Science & Engineering,
Techno International New Town
Kolkata – 700 156
West Bengal, India.

Approval

This is to certify that the project report entitled “AI-Enhanced Diagnosis of ASD Using SVC: A Data-Driven Approach” prepared under my supervision by Anirban Pal (18700120122), Bishal Kumar Ghosh (18700120100), be accepted in partial fulfillment for the degree of Bachelor of Technology in Computer Science & Engineering which is affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Formerly known as West Bengal University of Technology).

It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

…………………………………………
Prof. Chinmoy Kar

…………………………………………
Prof. Swagata Paul,
HOD, Computer Science & Engineering,
Techno International New Town

Abstract

Autism Spectrum Disorder (ASD) is a complex developmental condition marked by persistent challenges in social interaction, communication, and restrictive, repetitive behaviors. These challenges can significantly impact various aspects of an individual's life, including education, employment, and social relationships. Early diagnosis and intervention are crucial, as they can greatly improve long-term outcomes for children with ASD.

In this project, we present a novel application designed to assist in the early detection and diagnosis of ASD in children aged 1.5 to 5 years. Our app leverages a Support Vector Classifier (SVC) model trained on relevant data to assess the likelihood and typicalness of ASD. The user-friendly interface allows parents and healthcare providers to input necessary data, which the SVC model analyzes to provide a preliminary diagnosis. Furthermore, based on the model's assessment, the app recommends a customized level of therapy for the child, promoting early intervention and personalized treatment plans.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

Autism Spectrum Disorder (ASD) is a complex and lifelong neurodevelopmental disorder that presents a heterogeneous set of symptoms across individuals. It is primarily characterized by challenges in social communication and interaction, alongside restricted and repetitive behaviors, interests, and activities [1-3]. Early detection is critical for effective intervention, but diagnosing ASD traditionally relies on behavioral assessments, typically conducted around the age of two. However, the exact timing can vary depending on the severity of symptoms. Despite the availability of clinical tools for early detection, these tools are often used only when there is a significant suspicion or high risk of ASD due to their intricate diagnostic processes**.**

Globally, approximately 1 in 100 children are diagnosed with ASD [4]. In the United States, the Centers for Disease Control and Prevention (CDC) estimates that about 1 in 44 children aged 8 are identified with ASD [5]. In India, a 2021 study indicated an estimated prevalence of 1 in 68 children under 10 years old being diagnosed with ASD [8]. Without appropriate treatment, children with ASD may encounter difficulties in developing crucial skills related to learning, speech, and social interactions [6, 7]. Adults with untreated ASD often face challenges in independent living, employment, and forming relationships, and they have a higher risk of experiencing both physical and mental health issues [6]**.**

Identifying behavioral changes due to ASD in children is generally simpler than in adolescents and adults. This complexity arises because ASD symptoms can overlap with other mental health disorders as individuals age. This research focuses on classifying ASD as "ASD" or "no ASD" across different age groups: toddlers (up to 36 months), children (4-11 years), adolescents (12-16 years), and adults (17 years and over). The "Qualitative Checklist for Autism in Toddlers (Q-CHAT-10)" [9] and "Autism Spectrum Quotient (AQ-10)" [10, 11] form the basis of investigation in this approach. These screening questionnaires are also present in the ASD datasets used for this study, developed by Fadi Fayez Thabtah *et. al.* and have created a mobile-based ASD Test app, which is a screening application that provided the toddler, child, adolescent, and adult ASD datasets utilized in this research. The toddler dataset includes 18 attributes with an output ASD class, whereas the child, adolescent, and adult datasets each contain 21 attributes with an output ASD class**.**

While successful classification of ASD was achieved for child, adolescent, and adult cases, the toddler dataset presented challenges due to a significant imbalance, with a majority of instances classified as no ASD. This study aims to address this imbalance and emphasizes the classification of ASD across all age categories by using a minimal number of features from the datasets. By leveraging machine learning (ML) techniques, classifier models are built to classify ASD classes. These models are trained on training data and tested on test data. Ongoing analysis highlights the importance of feature selection, dimension reduction, the evolution of advanced ML approaches for ASD classification, enhancement of evaluation parameters, and reduction of diagnosis time for ASD [12].

A. Importance of Early-Stage Detection:

Early detection of Autism is transformative for several reasons. It enables the implementation of early intervention strategies tailored to a child's specific needs, significantly enhancing their developmental progress in areas such as communication, social skills, and behavior. During the early years, the brain's high plasticity allows for the formation of new connections and better adaptation, providing a prime opportunity to positively influence brain development and boost cognitive performance. Additionally, early diagnosis equips parents and caregivers with crucial knowledge and resources, helping them understand and support their child's needs more effectively. Therefore, recognizing the telltale signs and symptoms of ASD in children is essential for leveraging these benefits [9].

B. Prognosis of Early-Stage Detected ASD:

The prognosis for ASD detected at an early stage is generally more favorable compared to later detection. Early diagnosis and intervention can lead to several positive outcomes:

1. Improved Developmental Outcomes: Children who receive early intervention often exhibit significant improvements in communication, social skills, and behavior. Customized therapies can address specific developmental needs, helping children achieve critical milestones [10].

2. Enhanced Cognitive Function: During the early years, the brain's high plasticity allows for more effective formation of neural connections. Early intervention can leverage this plasticity to boost cognitive development and learning abilities [11].

3. Better Adaptation and Independence: Early support helps children with ASD develop better coping mechanisms and adaptive skills, leading to increased independence and a higher quality of life as they grow [10, 11].

4. Parental Support and Education: Early detection provides parents with a better understanding of their child's condition, enabling them to access appropriate resources and support, reducing stress and improving family dynamics [12].

5. Long-Term Benefits: Early intervention can result in long-term benefits, including improved educational outcomes, better employment prospects, and enhanced social relationships in adulthood. Individuals diagnosed early are more likely to lead fulfilling, productive lives [12].

C. Support Vector Classifier (SVC):

Support Vector Classification (SVC) is a supervised machine learning algorithm that optimizes the separation of data points into classes by finding the optimal hyperplane with the maximum margin, influenced by support vectors. It excels in handling high-dimensional, complex datasets and non-linear relationships using the kernel trick. Compared to Logistic Regression, which is linear and less effective in high-dimensional spaces, and Random Forests and Gradient Boosting, which can be computationally intensive and require extensive tuning, SVC stands out for its robust performance. Its ability to generalize well to unseen data, achieving accuracy scores of 95.986% in training and 93.439% in validation, makes it a preferred choice for various classification tasks.

D. Dataset and Supporting Data:

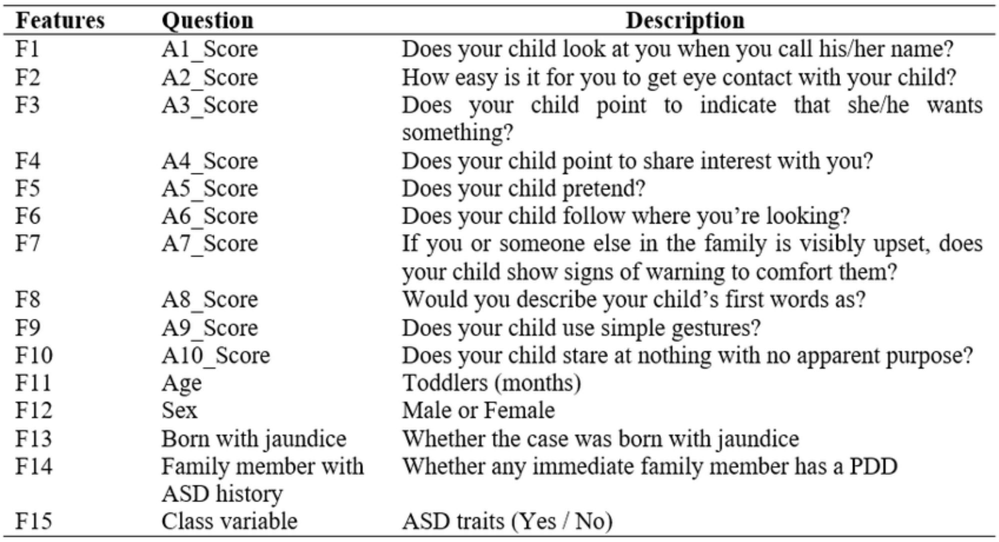
The dataset consists of 2,718 real samples and 728 augmented samples, collected from sources including Kaggle and TASD-Dataset from Mendeley Data. The questionnaire data, as illustrated in ***Figure 1***, was obtained from the University of Cambridge. The dataset includes fields such as gender, age in months, ethnicity, neonatal jaundice history, presence of ASD in family members, and responses to ten questions (Q1-Q10) concerning the child's behavior.

Links to the Data:

1. Kaggle: https://www.kaggle.com/datasets/fabdelja/autism-screening-for-toddlers

2. Mendley Data:
https://data.mendeley.com/datasets/87s2br3ptb/1#:~:text=The%20TASD%2DDataset%2C%20created%20for,associated%20with%20specific%20toddler%20behaviors.

3. Our Merged Data:
https://docs.google.com/spreadsheets/d/1zBZvQgvnkmSg4edfbFGv1HFp4TaqITgeIbL491GEde4/edit?usp=sharing



Features	Question	Description
F1	A1_Score	Does your child look at you when you call his/her name?
F2	A2_Score	How easy is it for you to get eye contact with your child?
F3	A3_Score	Does your child point to indicate that she/he wants something?
F4	A4_Score	Does your child point to share interest with you?
F5	A5_Score	Does your child pretend?
F6	A6_Score	Does your child follow where you're looking?
F7	A7_Score	If you or someone else in the family is visibly upset, does your child show signs of warning to comfort them?
F8	A8_Score	Would you describe your child's first words as?
F9	A9_Score	Does your child use simple gestures?
F10	A10_Score	Does your child stare at nothing with no apparent purpose?
F11	Age	Toddlers (months)
F12	Sex	Male or Female
F13	Born with jaundice	Whether the case was born with jaundice
F14	Family member with ASD history	Whether any immediate family member has a PDD
F15	Class variable	ASD traits (Yes / No)

Figure 1: A1-A10 Questions as per University of Cambridge

II. Requirement Specifications

1. Hardware specification-

    a. CPU: Utilizes approximately 1.7-2.1% of Intel's capacity and 0.9-1.6% of AMD's capacity

    b. RAM: Utilizes approximately 147 MBs on Intel and 152 MBs on AMD

    c. Hard Disk: Needs no setup and installation process and occupies 247 MBs on the root directory and 2-5 KBs on the temp directory

2. Software specification –

    a. Operating system(version): Supported on Windows XP to newer versions

    b. Software(version): Version 1.0 has been released and will receive automatic updates which includes the new model (i.e., the prepared SVC joblib model stored in the Firestore Cloud bucket). Additionally, for newer application updates, the system will check for new models. If a compressed file is fetched from the bucket, the application will display a message instructing users to remove the current version and unpack and use the updated version.

    c. Base Language: Python3.12

    d. Cloud Storage: Firestore

III. Background

In [13] paper by Ahmed El-Gazzar, Mirjam Quaak, e*t. al.*, an end-to-end algorithm is proposed that can extract spatiotemporal features from 4-D fMRI scan data using 3-D CNNs and 3-D Convolutional LSTMs. The results of this algorithm are evaluated using the publicly available ABIDE dataset.

In [14] by Md. Delowar Hossain, Muhammad Ashad Kabir, *et. al.*, the primary focus is on identifying optimal traits and automating the diagnosis process using various classification techniques. The authors analyze datasets from toddlers, children, adolescents, and adults to determine ASD using MLP (Multilayer Perceptron).

Ibrahim Abdulrab Ahmed, Ebrahim Mohammed Senan, *et. al.* in [15] explore ASD determination using eye-tracking techniques and visual behavior analysis through a hybrid model and employed three distinct methods for diagnosing ASD. The first method leverages Feedforward Neural Networks (FFNNs) and Artificial Neural Networks (ANNs) for feature classification, utilizing a hybrid approach with Local Binary Pattern (LBP). The second method uses a pre-trained Convolutional Neural Network (CNN) model. The third method integrates deep learning and machine learning techniques.

In [16], the use of various machine learning models, including Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), K-Nearest Neighbours (KNN), Logistic Regression (LR), and Support Vector Machines (SVM), to determine ASD is explored by Suman Raja and Sarfaraz Masood. The performance of these models is compared, and it is found that the CNN-based prediction model delivers superior results.

The application of Federated Learning (FL) for autism detection is explored by Muhammad Shoaib Farooq, Rabia Tehseen, Maidah Sabir, and Zabihullah Atal in [17]. This innovative technique is employed to train two different machine learning classifiers: Logistic Regression and Support Vector Machine.

In [18], M. Parvathi, M. Ravikanth, and M. Neelakantappa address ASD detection by integrating K-means clustering with Support Vector Machine (SVM) techniques. They transform the supervised classification of SVM into an unsupervised clustering problem and observe improved accuracy levels.

The paper [19] by Tania Akter, Md. Shahriare Satu, Md. Imran Khan, Mohammad Hanif Ali, Shahadat Uddin, Pietro Liò, Julian M.W. Quinn, and Mohammad Ali Moni employs the Q-chart 10 and AQ-10 tools for data collection from candidates. Subsequently, the data is evaluated using a pre-trained Support Vector Machine (SVM) model to derive the results.

In [20], Random Forest Classifier (RFC), Naive Bayes (NB), Logistic Regression (LR), and K-Nearest Neighbours (KNN) were employed by Kaushik Vakadkar, Diya Purkayastha, and Deepa Krishnan to analyse a dataset and construct predictive models. The primary objective of this study is to identify early susceptibility to ASD in children, thereby facilitating the streamlining of the diagnostic process.

Diagnostic models for determining ASD based on regional thickness measurements derived from Surface-based Morphometry (SBM) were constructed by authors Yun Jiao, Rong Chen, Xiaoyan Ke, Kangkang Chu, Zuhong Lu, and Edward H. Herskovits in [21]. These models were compared to diagnostic models based on volumetric morphometry. Support Vector Machines (SVM), Multilayer Perceptrons (MLPs), Function Trees (FTs), and Logistic Model Trees (LMTs) were utilized to develop the diagnostic models. Among these, the highest accuracy for thickness-based classification was achieved by LMT. The diagnostic model generated by LMT for thickness-based classification includes seven structures.

The primary objective of the paper by K. K. Mujeeb Rahman and M. Monica Subashini in [22] is to develop and implement algorithms based on deep neural networks (DNN) for identifying patients using the QCHART datasets.

A predictive model for diagnosing Autism Spectrum Disorder (ASD) was developed by the authors of this paper [23], employing Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forest (RF). To mitigate computational complexity, a feature selection algorithm based on Recursive Feature Elimination (RFE) was utilized.

A. Comparative study:

Several data-driven approaches have been studied and analyzed, including DNN and CNN-based models, as well as various computer vision techniques implemented in prior research. ***Table 1*** presents a comparative study between previous years' research and our current research.

Ref	Tech Stack & Algorithm	Accuracy	Cons
14	Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO) with Relief-F	92.58%	Data imbalance, complexity, feature engineering, generalization, limited dataset access.
15	Hybrid Technique (CNN + SVM)	GoogleNet + SVM: 91.5%, ResNet-18 + SVM: 93.5%	Generalization, Limited Dataset Access
17	Logistic Regression (LR), Support Vector Machine (SVM), Meta Classifier	Children: 93%, Adults: 81%	Limited model complexity, Data heterogeneity, Communication overhead, Lack of transparency
Our Task	Support Vector Classifier (SVC), Logistic Regression (LR), Random Forest, Gradient Boosting	93.439%	Limited Dataset Availability, Lack of Regressive Learning

Table 1: Comparative Study

IV. Problem Statement

The early and accurate detection of Autism Spectrum Disorder (ASD) is crucial for effective intervention and support. Traditional diagnostic methods are often time-consuming and require specialized expertise. This research aims to develop a robust predictive model for ASD detection using the Support Vector Classification (SVC) algorithm, known for handling high-dimensional data and non-linear relationships effectively.

The study utilizes a comprehensive dataset, including real and augmented samples from sources like Kaggle, the CDC Autism Data Visualization Tool, and the University of Cambridge. Key behavioral and demographic features contributing to ASD diagnosis will be identified. Recursive Feature Elimination (RFE) will be employed to optimize feature selection, enhancing model performance and reducing computational complexity.

The primary objectives are to:

1. Evaluate SVC's effectiveness in ASD detection: Assessing the model's accuracy, precision, recall, and overall performance.
2. Identify significant predictive features: Using RFE to determine the most important features for ASD diagnosis.
3. Compare SVC with other machine learning approaches: Benchmarking against Artificial Neural Networks (ANN) and Random Forest (RF) to identify the most effective method.

This research aims to develop an accurate and efficient machine learning-based approach for earlier and more accessible ASD diagnosis.

V. Proposed Methodology

The entire system employs a packaged application developed using the Tkinter GUI library built on Python. The system was trained on the dataset using Logistic Regression, Random Forest, Gradient Boosting, and SVC classifiers with varying parameters. In the developed evaluative model, the process follows the sequence illustrated in ***Figure 2***.

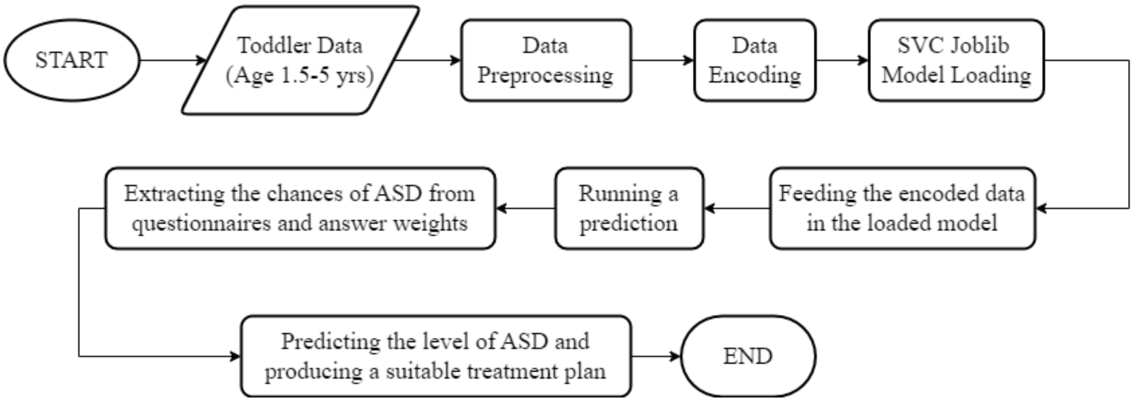

START
Toddler Data (Age 1.5-5 yrs)
Data Preprocessing
Data Encoding
SVC Joblib Model Loading
Feeding the encoded data in the loaded model
Running a prediction
Extracting the chances of ASD from questionnaires and answer weights
Predicting the level of ASD and producing a suitable treatment plan
END

Figure 2: Flow of the mechanism

The flowchart depicts a structured process for predicting Autism Spectrum Disorder (ASD) in toddlers aged 1.5 to 5 years. The process begins with the collection of toddler data, which likely includes behavioral observations and questionnaire responses. This raw data undergoes preprocessing to clean and prepare it for analysis. Next, the data is encoded into a format suitable for machine learning algorithms. Following this, an SVC (Support Vector Classifier) model, previously trained and saved using Joblib, is loaded to make predictions.

Once the model is ready, the encoded data is fed into it to run the prediction. The model processes the data to extract the likelihood of ASD based on the questionnaire responses and associated weights. The final steps involve predicting the level of ASD and producing a suitable treatment plan based on the prediction. The flowchart concludes with the output of the prediction and treatment plan, providing a clear and actionable result for further medical or therapeutic intervention.

A. Model Training and Exporting:

The dataset, formatted as a CSV file, is loaded and cleaned by addressing data type formats and empty values. Boolean values (i.e., 'Yes' and 'No') are converted to binary (0/1) format, and consistent data entries are removed from the dataset. The corrected, cleaned, and encoded data is then fed into the classifiers previously mentioned: Random Forest, Gradient Boosting, Logistic Regression, and SVC. The accuracy levels of these classifiers are compared with varying hyperparameters. The SVC classifier has demonstrated exceptional performance, achieving an accuracy of 95.986% in training and 93.439% in validation, as shown in ***Table 2*** which mentions the best 2 of the considerations. Based on the dataset analysis, Logistic Regression showed poor performance, resulting in the selection of SVC without any kernel for the final model.

	Models	Training Accuracy	Validation Accuracy
Case 1	Logistic Regression	89.151	82.633
	Gradient Boosting (subsample=0.5, eta=0.07)	87.452	92.751
	SVC	93.472	91.257
	SVC (kernel=rvf, degree=2)	91.367	92.027
Case 2	Logistic Regression (random state=10)	90.173	91.647
	Gradient Boosting (subsample=0.5, eta=0.06, max_depth=7)	83.713	85.434
	SVC	95.986	93.439
	SVC (kernel=rvf, degree=2, gamma=scale)	92.447	91.025

Table 2: Accuracy Comparison on Varying Models

B. Data Capture:

The data is captured from users, who can be the child's parent, paediatrician, or any relative, and includes features F1-F14 as shown in ***Error! Reference source not found.***, which are obtained through the application interface. The data entered, initially in string format, is appended to a list. The questionnaire responses are then converted to binary values (1) based on the following criteria: for Questions 1-9, responses with options 3-5 are assigned a value of 1, and for Question 10, responses with options 1-3 are assigned a value of 1. The responses to each question carry a weight that determines the importance of the question in assessing ASD. This weighted scoring helps in applying an overlay to determine the ASD level, as indicated in ***Table 3***. This assessment will further aid in evaluating the appropriate treatment for the child. These questions as mentioned in ***Figure 1*** are standardized questions which have been assigned particular weights to determine the chances of ASD in the diagnosed child.

Questions		Weight
Q1	***Does your child look at you when you call his/her name?***	9
Q2	***How easy is it for you to get eye contact with your child?***	10
Q3	***Does your child point to indicate that she/he wants something? (e.g., a toy that is out of reach)***	8
Q4	***Does your child point to share interest with you? (e.g., pointing at an interesting sight)***	8
Q5	***Does your child pretend? (e.g., care for dolls, talk on a toy phone)***	7
Q6	***Does your child follow where you're looking?***	7
Q7	***If you or someone else in the family is visibly upset, does your child show signs of wanting to comfort them? (e.g., stroking hair, hugging them)***	6
Q8	***Would you describe your child's first words as:***	6

Q9	***Does your child use simple gestures? (e.g., wave goodbye)***	7
Q10	***Does your child stare at nothing with no apparent purpose or have some emotional attachment to a certain object (e.g., a pen, a cup, a teddy which he/she considers as a safety net)?***	5

Table 3: Weight per Question

C. Data Preprocessing and Encoding:

The input data undergoes processing in accordance with the standard Scalar format. Categorical 'yes' and 'no' responses are transformed into numerical values, with 'yes' converted to 1 and 'no' converted to 0. Additionally, columns that contain 'object' data types are encoded using the *LabelEncoder()* function from the *sklearn.preprocessing* library. The process of evaluating the level of ASD is derived from and based on the data presented in ***Table 3***, as described below:

1. Assign Scores to Responses:
- For each question, assign 1 point if the response indicates a potential ASD trait (based on the criteria provided earlier).

2. Calculate Total Score:
- Sum the scores from all questions.

3. Convert Score to Percentage:
- Calculate the percentage of ASD likelihood by dividing the total score by the maximum possible score (73) and then multiplying by 100.

4. Determine ASD Level:
- **Mild ASD**: 0% - 33%
- **Moderate ASD**: 34% - 66%
- **Severe ASD**: 67% - 100%

Example Calculation:

Assume the responses indicate the following potential ASD traits:

- Questions 1-9: 5 questions answered with options indicating ASD traits.
- Question 10: Answered with an option indicating an ASD trait.

This would give a total score of 6.

Step-by-Step Calculation:

1. Total Score: 6
2. ASD Percentage: $\frac{6}{73} \times 100 = 8.22\%$
3. ASD Level: 8.22% falls into the Mild ASD category.

VI. Result Analysis and Discussion

As per the analysis on the data available by various sources it has been noticed that children of certain ethnic groups have higher chances of ASD as in ***Figure 3***.

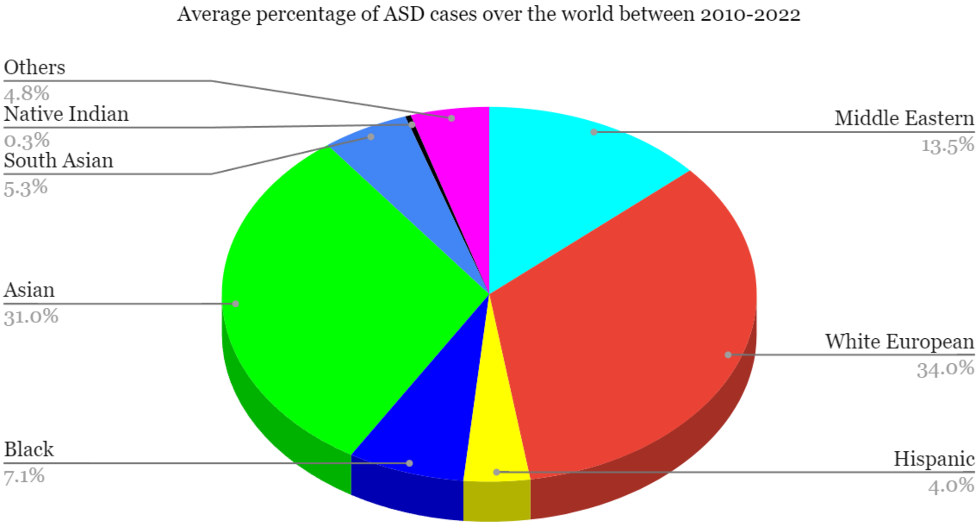


Figure 3: Average percentage of ASD cases over the world between 2010-2022

The theory that Asperger, a German researcher, put forth regarding autism suggested an "extreme variant of male intelligence" associated with the condition. This idea would likely face controversy in today's discourse surrounding gender identity and equality, as it touches upon the sensitive subject of cognitive differences between sexes. The definition of intelligence remains elusive, and perceptions of it are subject to various interpretations.

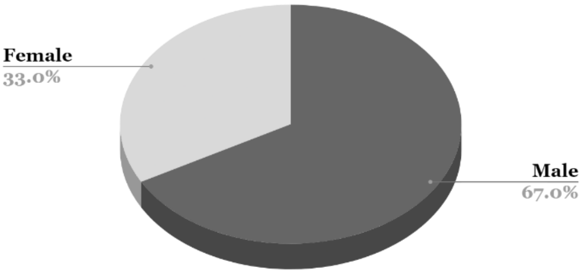


Figure 4: Correlation between Sex and ASD

Research has shown that, at a fundamental level, male and female brains exhibit differences. Male brains tend to be larger than female brains, and on standard IQ tests, men and women typically score the same average. However, the male population displays a broader range of scores, with more men at both the top and bottom of the scale.

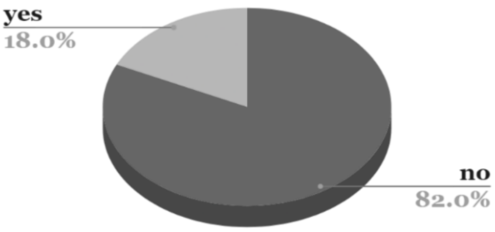


Figure 5: Correlation Between ASD in Children and Family Members with

Interestingly, when examining specific components of IQ tests, disparities between men and women become apparent. Women often score higher in verbal abilities, while men excel in visuospatial abilities. This pattern bears resemblance to the verbal communication deficits observed in ASD patients, regardless of sex. Furthermore, men generally outperform women in math and analytical analysis problems, which also aligns with typical ASD characteristics. The tested data always establishes that ASD is quite a trait for males as in

Figure 4. The peak diagnosis of ASD occurs within the age ranges of 35-38 months and 23-26 months, as demonstrated in ***Figure 6***. This is likely because developmental milestones become more pronounced during these periods, making it easier to identify delays or atypical behaviors associated with ASD. At these ages, children typically begin to exhibit more complex social interactions, communication skills, and play behaviors, which can highlight the presence of developmental differences that prompt further evaluation and diagnosis.

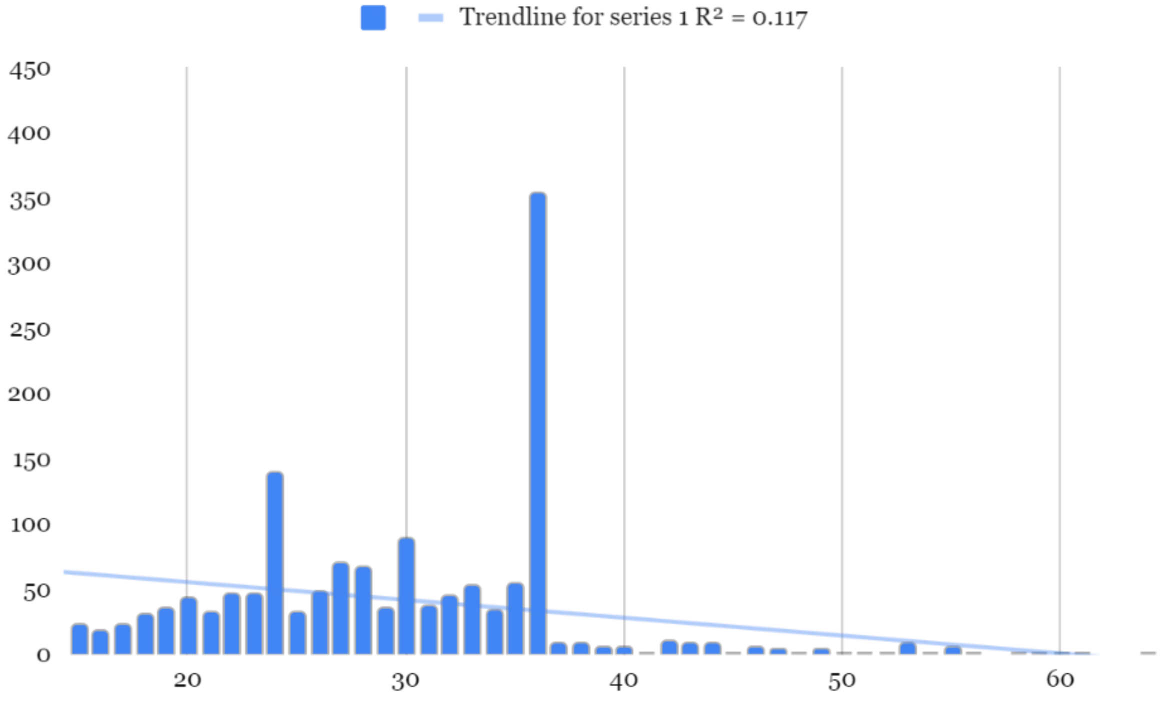


Figure 6: ASD Cases diagnosed and detected with respect to age in months

No correlations have been identified between family members with ASD and the occurrence of ASD in children as in ***Figure 5***. However, ASD is a heterogeneous genetic disorder with a male-to-female ratio of 4:1. It involves over 800 recognized ASD-related genes, numerous chromosome aberrations, and dozens of identified syndromes. The disorder is influenced by a complex interplay of genetic inheritance, environmental factors, and epigenetics.

We have also provided examples of some test cases in ***Table 4*** as follows which are the sample validation test data:

QCHAT Score	QCHAT Weight	Age (Mons)	Sex	Ethnicity	Neonatal Jaundice	Family member ASD	Output	Percent	Typical Type
8	57	21	M	White European	No	No	Yes	78.03	Severe ASD
5	38	33	M	Asian	Yes	No	Yes	52.04	Moderate ASD
6	41	33	M	Asian	Yes	No	Yes	56.16	Moderate ASD

2	11	36	M	Asian	No	No	Yes	15.07	Mild ASD
3	21	29	F	Middle Eastern	No	No	No	28.77	Mild Chance of ASD

Table 4: Validation Test Result

The evaluated results include data tested from real life data. The tested output is based on the SVC model that was trained on the custom dataset. ***Table 4*** presents real-life test results for ASD diagnosis across various demographic and clinical factors. The data includes QCHAT scores, weights, ages, sex, ethnicity, presence of neonatal jaundice, family history of ASD, output scores, percentages, and the typical type of ASD diagnosed. The results indicate that higher QCHAT scores and weights generally correlate with more severe ASD diagnoses, as seen in the first case with a severe ASD diagnosis and an output score of 78.03%. Moderate ASD cases (cases 2 and 3) show mid-range QCHAT scores and weights with output scores around 50-56%. A mild ASD case (case 4) has the lowest QCHAT score and weight, reflecting a significantly lower output score of 15.07%. The only female case (case 5), with intermediate scores and weights, shows a mild chance of ASD, indicating that various factors, including age and QCHAT scores, play crucial roles in determining the severity of ASD.

VII. Conclusion

Treating Autism Spectrum Disorder (ASD) at a younger age is vital for optimizing developmental outcomes. Early intervention facilitates timely identification of symptoms, leveraging the brain's plasticity to enhance neural adaptation and skill acquisition crucial for social interaction and communication. While commendable, the Windows application developed using Python and SVC machine learning model has limitations, including restricted platform support, a small dataset potentially leading to overfitting, a lack of regressive learning mechanisms hindering adaptability, and the absence of live physician support, limiting personalized guidance and expertise. Addressing these limitations through broader platform support, dataset expansion, incorporating adaptive learning mechanisms, and integrating physician support could significantly enhance the application's efficacy in supporting individuals with ASD.

VIII. Future Scope

The future development of the developed system aims to enhance its diagnostic capabilities and accessibility. One significant advancement will be the integration of live doctor consultations and recommendations, particularly beneficial when the test is administered by a parent or relative. This feature will ensure immediate professional guidance and support, offering personalized medical advice and next steps based on the child's test results. Additionally, incorporating a Sinc-Net module will automate the analysis of certain traits directly from the child. This includes assessing the child's response to their name, tracking eye movements while viewing engaging videos for toddlers, and analyzing other behaviors such as facial expressions, gestures, and vocalizations. These enhancements will provide deeper insights into social attention and visual engagement, critical markers for ASD.

To further improve the model's effectiveness, we plan to develop its self-learning capabilities. By implementing reinforcement learning techniques, the model can continuously learn from new data and outcomes, enhancing its predictive accuracy over time. A feedback loop where the system learns from consultations and professional recommendations will refine its assessment algorithms, making it more adaptive and accurate. Additionally, we will build a web application to make the ASD evaluation tool widely accessible. This will allow users from various regions to easily access the tool, increasing its reach and impact. The application will feature a user-friendly interface compatible with smartphones, tablets, and computers, ensuring a seamless user experience. Robust data security and privacy measures will also be implemented to protect users' information, fostering trust and regulatory compliance. These future enhancements will significantly improve the diagnostic capabilities of the ASD evaluation system, providing comprehensive support and accessibility to users worldwide.

IX. APPENDIX I

a. *Pseudocode of the Working Prototype:*

```
 // Function to calculate age from DOB
function calculateAge(DOB):
    today = currentDate()
    age = today.year - DOB.year
    if (today.month < DOB.month) or (today.month == DOB.month and today.day < DOB.day):
        age -= 1
    return age

// Function to convert Ethnicity to index
function convertEthnicityToIndex(ethnicity):
    // Predefined mapping of ethnicity to index (0-11)
    ethnicityMap = {"Ethnicity1": 0, "Ethnicity2": 1, ..., "Ethnicity12": 11}
    return ethnicityMap[ethnicity]

// Function to convert Yes/No to 1/0
function convertYesNoToBinary(value):
    if value == "Yes":
        return 1
    else:
        return 0

// Function to score QCHAT-10 answers
function scoreQCHAT10(answers):
    scores = []
    for i = 1 to 9:
        if answers[i] in [3, 4, 5]:
            scores.append(1)
        else:
            scores.append(0)
    if answers[10] in [1, 2, 3]:
        scores.append(1)
    else:
        scores.append(0)
    return scores

// Function to calculate ASD percentage and level
function calculateASDLevel(totalScore, maxScore = 73):
    asdPercentage = (totalScore / maxScore) * 100
    if asdPercentage <= 33:
        asdLevel = "Mild ASD"
    elif asdPercentage <= 66:
        asdLevel = "Moderate ASD"
    else:
        asdLevel = "Severe ASD"
    return asdPercentage, asdLevel

// Main function to process the evaluation
function evaluateASD():
    // Input details
    childName = input("Enter the name of the child:")
    childDOB = input("Enter the DOB of the child (DD-MM-YYYY):")
    ethnicity = input("Enter the ethnicity of the child:")
    neonatalJaundice = input("Did the child have neonatal jaundice? (Yes/No):")
    familyASD = input("Does any family member have ASD? (Yes/No):")
    parentEmail = input("Enter the email address of the parent/doc/relative:")

    // Calculate age
    age = calculateAge(childDOB)

    // Convert inputs to appropriate format
    ethnicityIndex = convertEthnicityToIndex(ethnicity)
    neonatalJaundiceBinary = convertYesNoToBinary(neonatalJaundice)
    familyASDBinary = convertYesNoToBinary(familyASD)

    // Input QCHAT-10 answers
    qchatAnswers = []
    for i = 1 to 10:
        answer = input("Enter answer for Q" + i + ":")
        qchatAnswers.append(answer)

    // Score QCHAT-10 answers
    qchatScores = scoreQCHAT10(qchatAnswers)

    // Calculate total score
    importanceLevels = [9, 10, 8, 8, 7, 7, 6, 6, 7, 5]
    totalScore = 0
    for i = 0 to 9:
        if qchatScores[i] == 1:
            totalScore += importanceLevels[i]

    // Calculate ASD percentage and level
    asdPercentage, asdLevel = calculateASDLevel(totalScore)

    // Display result
    print("ASD Percentage: " + asdPercentage + "%")
    print("ASD Level: " + asdLevel)

    // Generate treatment plan based on ASD level
     treatmentPlan = generateTreatmentPlan(asdLevel)
     // Implement this function to return a treatment plan based on the level

    // Send treatment plan to email
    sendEmail(parentEmail, "ASD Evaluation Result and Treatment Plan", treatmentPlan)

    // Display confirmation
    print("The treatment plan has been sent to " + parentEmail)

// Execute the evaluation
evaluateASD()
```

X. APPENDIX II

a. Source Code:

input_form.py

```
from tkinter import Tk, PhotoImage, Frame, Menu, Label, Entry, Button, CENTER
from tkinter.ttk import Combobox
from tkcalendar import DateEntry
from datetime import datetime, date
from dateutil.relativedelta import relativedelta
from tkinter import messagebox
from re import match
from webbrowser import open
from smtplib import SMTP
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from config import mail_id, password
from subprocess import run
confirm_color = '#90EE90'

window_width = 380
window_height = 490
root = Tk()
root.geometry(f'{window_width}x{window_height}')
root.title('Input Form')
root.resizable(False, False)
icon = PhotoImage(file = '.images/form-icon.png')
root.iconphoto(False, icon)


def center_screen():
    """ gets the coordinates of the center of the screen """
    global screen_height, screen_width, x_coordinate, y_coordinate
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()
        # Coordinates of the upper left corner of the window to make the window appear in
the center
    x_coordinate = int((screen_width/2) - (window_width/2))
    y_coordinate = int((screen_height/2) - (window_height/2))
    root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_coordinate,
y_coordinate))
center_screen()

## Outline
frame1 = Frame(root, highlightbackground="blue", highlightthickness=1,width=370,
height=590, bd=2)
frame1.pack(pady=10, padx=10)


## Menu Bar
menubar = Menu(root)

# About and Help
help = Menu(menubar, tearoff=0)
menubar.add_cascade(label ='Help', menu = help)
def show_about():
    messagebox.showinfo('ASD', "Autism spectrum disorder is a significant developmental
condition that hinders communication and social interaction abilities. It affects various
aspects of an individual's cognitive, emotional, social, and physical well-being due to
its impact on the nervous system. Symptoms can range in severity, commonly including
challenges in communication and social interactions, along with repetitive behaviors and
intense interests. Early identification and access to behavioral, educational, and family
interventions can help alleviate symptoms, promote development, and enhance learning
outcomes.\n\nWhile treatment can provide support, autism spectrum disorder cannot be
cured. It is a chronic condition that can persist for years or throughout life. Diagnosis
typically requires medical evaluation, although lab tests or imaging studies are rarely
necessary.")
def show_affect():
    messagebox.showinfo('Effects', 'ASD, or autism spectrum disorder, is a neuro-
developmental condition that can lead to various social, communication, and behavioral
difficulties in children. While certain children may exhibit ASD symptoms during infancy,
others may initially develop typically before later displaying withdrawal or
aggression.')
def show_eval():
    messagebox.showinfo('Evaluation effects', "Early evaluation of autism spectrum
disorder (ASD) is crucial as it enables timely initiation of targeted intervention
programs tailored to a child's specific needs. These interventions, such as speech
therapy, occupational therapy, and behavioral interventions, can address communication
challenges, improve social interactions, and manage behavioral issues effectively.
Additionally, early diagnosis allows families to access education and support services,
empowering them to understand ASD better and actively participate in their child's
development. Ultimately, early intervention significantly improves long-term outcomes by
promoting optimal development, school readiness, and overall well-being for children with
ASD.")
def show_q_chat():
    messagebox.showinfo('Q-CHAT-10', "The Q-CHAT measures ASD symptoms and language
impairments in toddlers. The Q-CHAT-10 is a uni-dimensional measure of autistic traits.")
def exit_app():
    exit(0)
help.add_command(label ='About', command = show_about)
help.add_command(label ='How it might affect your child?', command = show_affect)
help.add_command(label ='How the evaluation helps?', command = show_eval)
help.add_command(label ='What is Q-Chat-10?', command = show_q_chat)
help.add_separator()
help.add_command(label ='Exit', command = exit_app)

# More Info
info = Menu(menubar, tearoff=0)
menubar.add_cascade(label='Info', menu=info)
def know_more():
    open('https://www.cdc.gov/ncbddd/autism/facts.html')
info.add_command(label='More', command=know_more)


## NAME SECTION ##
name_label = Label(root, text="Name:")
name_label.place(x=30, y=50)
name_entry = Entry(root, borderwidth=.5, relief="solid")
name_entry.place(x=85, y=50, width=197)
name_entry.focus_force()
name = ''
def name_clicked():
    global name
    name = name_entry.get()
    if name=='':
        messagebox.showerror('Error', 'Name is empty!')
    else:
        name_button.config(state='disabled')
        name_entry.config(state='disabled')
        name_button.config(background=confirm_color)
        messagebox.showinfo('Information', "Candidate\'s Name will not be recorded!")
name_button = Button(root, text='Confirm', command=name_clicked, relief='groove')
name_button.place(x=290, y =47)



## AGE AND DOB SECTION ##
current_date = datetime.now()
reverse_date_upper = current_date + relativedelta(months=-18)
reverse_date_lower = current_date + relativedelta(months=-70)
cal = DateEntry(root, borderwidth=1, relief="solid", width=12, date_pattern='dd/mm/y',
year = current_date.year, month = current_date.month, day = current_date.day,
background='darkblue', foreground='white')
dob_label = Label(root, text="DOB:")
dob_label.place(x=30, y=90)
dob = ''
def dob_clicked():
    global dob
    dob = cal.get()
    dob  = dob.split('/')
    if date(year=reverse_date_lower.year, month=reverse_date_lower.month,
day=reverse_date_lower.day)< date(year=int(dob[2]), day=int(dob[0]), month=int(dob[1])) <
date(year=reverse_date_upper.year, month=reverse_date_upper.month,
day=reverse_date_upper.day):
        dob_button.config(state='disabled')
        cal.config(state='disabled')
        dob_button.config(background=confirm_color)
    else:
        messagebox.showerror("Age Error", "Date range should be from 18 months to 5
years!")
dob_button = Button(root, text='Confirm', command=dob_clicked, relief='groove')
dob_button.place(x=290, y =88)
cal.place(x=85, y=90, width=197)
```

```
## GENDER SECTION ##
gender_label = Label(root, text="Gender:")
gender_drop = Combobox(state="readonly", values=["None", "Male", "Female"], width=29)
gender_drop.current(0)
gender = ''
gender_label.place(x=30, y=130)
gender_drop.place(x=85, y=130)
def gender_clicked():
    global gender
    gender = gender_drop.get()
    if gender == 'None':
        messagebox.showerror("Gender Error", "Not selected!")
    else:
        gender_button.config(state='disabled')
        gender_drop.config(state='disabled')
        gender_button.config(background=confirm_color)
gender_button = Button(root, text='Confirm', command=gender_clicked, relief='groove')
gender_button.place(x=290, y =128)


## Ethnicity Selection ##
ethnicity_label = Label(root, text="Ethnicity:")
ethnicity_types = ["None", 'Asian', 'Black', 'Hispanic', 'Latino', 'Middle Eastern',
'Mixed', 'Native Indian', 'Pacifica', 'South Asian', 'White European', 'Others',
'Unknown']
ethnicity_drop = Combobox(state="readonly", values=ethnicity_types, width=29)
ethnicity_drop.current(0)
ethnicity = ''
ethnicity_label.place(x=30, y=170)
ethnicity_drop.place(x=85, y=170)
def ethnicity_clicked():
    global ethnicity
    ethnicity = ethnicity_drop.get()
    if ethnicity == 'None':
        messagebox.showerror("Ethnicity Error", "Not selected!")
    else:
        ethnicity_button.config(state='disabled')
        ethnicity_drop.config(state='disabled')
        ethnicity_button.config(background=confirm_color)
ethnicity_button = Button(root, text='Confirm', command=ethnicity_clicked,
relief='groove')
ethnicity_button.place(x=290, y =168)


## Neonatal jaundice Entry ##
jaundice_label = Label(root, text="Neonatal Jaundice:")
jaundice_ans = ["None", "Yes", "No"]
jaundice_drop = Combobox(state="readonly", values=jaundice_ans, width=20)
jaundice_drop.current(0)
jaundice = ''
jaundice_label.place(x=30, y=210)
jaundice_drop.place(x=140, y=210)
def jaundice_clicked():
    global jaundice
    jaundice = jaundice_drop.get()
    if jaundice == 'None' or '':
        messagebox.showerror("Error", "Not selected!")
    else:
        jaundice_button.config(state='disabled')
        jaundice_drop.config(state='disabled')
        jaundice_button.config(background=confirm_color)
jaundice_button = Button(root, text='Confirm', command=jaundice_clicked, relief='groove')
jaundice_button.place(x=290, y =208)

## Family Member with ASD ##
family_label = Label(root, text="Any family member with ASD:")
family_ans = ["None", "Yes", "No"]
family_drop = Combobox(state="readonly", values=family_ans, width=10)
family_drop.current(0)
family = ''
family_label.place(x=30, y=250)
family_drop.place(x=200, y=250)
def family_clicked():
    global family
    family = family_drop.get()
    if family == 'None' or '':
        messagebox.showerror("Error", "Not selected!")
    else:
        family_button.config(state='disabled')
        family_drop.config(state='disabled')
        family_button.config(background=confirm_color)
family_button = Button(root, text='Confirm', command=family_clicked, relief='groove')
family_button.place(x=290, y =248)


## Person Taking the test ##
user_label = Label(root, text="User taking the test:")
user_ans = ["None", "Parent", "Doc/Pediatrician", "Relative"]
user_drop = Combobox(state="readonly", values=user_ans, width=20)
user_drop.current(0)
user = ''
user_label.place(x=30, y=290)
user_drop.place(x=140, y=290)
def user_clicked():
    global user
    user = user_drop.get()
    if user == 'None' or '':
        messagebox.showerror("User Error", "Please select the person taking the test!")
    else:
        user_button.config(state='disabled')
        user_drop.config(state='disabled')
        user_button.config(background=confirm_color)
user_button = Button(root, text='Confirm', command=user_clicked, relief='groove')
user_button.place(x=290, y =288)


## Separator ##
l = Label(root, text = "CONTACT DETAILS")
l.place(relx = 0.5, rely = .7, anchor = CENTER)
l.configure(font=("Courier", 14, 'bold'))

## Email ##
email = ''
email_label = Label(root, text='Email:')
email_entry = Entry(root)
email_entry.config(width=50, relief='solid')
def get_email():
    global email
    email= email_entry.get()
    if match('^[_a-z0-9-]+(\\.[_a-z0-9-]+)*@[a-z0-9-]+(\\.[a-z0-9-]+)*(\\.[a-z]{2,4})$',
email):
        email_button.config(state='disabled')
        email_entry.config(state='disabled')
        email_button.config(background=confirm_color)
        if name!='' and len(dob)!=0 and (gender!='None' or gender!='') and
(ethnicity!='None' or ethnicity!='') and (jaundice!='None' or jaundice!='') and
(family!='' or family!='None') and (user!='None' or user!=''):
            submit_button.configure(state='active')
    else:
        messagebox.showerror('Mail Error', 'Invalid Email!')
email_button = Button(root, text='Confirm', command=get_email, relief='groove')
email_label.place(x=30, y=380)
email_entry.place(x=85, y=380, width=197)
email_button.place(x=290, y=378)

## Function to calculate age
def calculate_age(birthdate):
    today = date.today()
    age = today.year - birthdate.year - ((today.month, today.day) < (birthdate.month,
birthdate.day))
    return age * 12

## submit button ##
def submit_action():
    global user, dob, gender, ethnicity, jaundice, family, ethnicity_types, user_ans

    sender_email = mail_id
    recipient_email = email
    sender_password = password
    subject = "Welcome to NeuroVantage AI"
    note = 'DISCLAIMER: \nName and email will not be stored and used. Name has been taken
so as to address the child in the application and email. Email has been taken for
communication and result mailing purposes.'
    body = f"Welcome {user} for participating in the ASD assessment program.
Thank you for connecting with us. We hope to serve you the best cure and help.\n\nThe
details filled out are:\nName: {name}\nDOB: {'-'.join(dob)}\nGender: {gender}\nEthnicity:
```

```
{ethnicity}\nNeonatal Jaundice: {jaundice}\nAny family member diagnosed with ASD: 
{family}\nUser taking the test: {user}\n\nThanking You,\nTeam Cognitive 
Healers\n\n{note}"

    # Calculating parameters
    age = date(int(dob[2]), int(dob[1]), int(dob[0]))
    age = calculate_age(age)
    gender = 1 if gender == 'Male' else 0
    ethnicity_types =  ethnicity_types[1:]
    ethnicity_types.sort()
    print(ethnicity_types)
    ethnicity = ethnicity_types.index(ethnicity)
    jaundice = 0 if jaundice.lower()=='no' else 1
    user_ans = user_ans[1:]
    user_ans.sort()
    user = user_ans.index(user)
    family = 0 if family.lower()=='no' else 1
    # messagebox.showinfo('Data_Collected', f'Age in Months: {age}\n, DOB: {dob}\n, 
Gender: {gender}\n, Ethnicity: {ethnicity}\n, Neonatal Jaundice: {jaundice}\n, User 
taking the test: {user}\n, Any Family member with ASD: {family}\n')

    # Create a MIMEText object\
    message = MIMEMultipart()
    message['From'] = sender_email
    message['To'] = recipient_email
    message['Subject'] = subject
    message.attach(MIMEText(body, 'plain'))

    try:
        # Connect to the SMTP server
        server = SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(sender_email, sender_password)

        # Send email
        server.sendmail(sender_email, recipient_email, message.as_string())
        server.quit()
        messagebox.showinfo("Success", "Verification E-mail sent successfully!")
        root.destroy()
        run(['python', 'questionnaire.py', f"{recipient_email}, {age}, {gender}, 
{ethnicity}, {jaundice}, {user}, {family}, {name}"])
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred: {str(e)}")

submit_button = Button(root, text='S  U  B  M  I  T', command=submit_action, 
state='disabled', relief='groove', font=('Arial', 11, 'bold'), background='#dee9fa')
submit_button.place(x=30, y=420, width=315)

root.config(menu = menubar)
root.mainloop()
print(f'Name: {name}\nDOB: {dob}\nGender: {gender}\nEthnicity: {ethnicity}\nNeonatal 
Jaundice: {jaundice}\nUser: {user}\nEmail: {email}')
```

questionnaire.py

```
from tkinter import Tk, Label, PhotoImage, Frame, Radiobutton, IntVar, Button
from sys import argv
from tempfile import gettempdir
from uuid import uuid4
from subprocess import run

questions = {"Does your child look at you when you call his/her name?": ['Always', 
'Usually', 'Sometimes', 'Rarely', 'Never'],
             "How easy is it for you to get eye contact for your child?": ['Very easy', 
'Quite easy', 'Quite difficult', 'Very difficult', 'Impossible'],
             "Does your child point to indicate that he/she wants something? (e.g. a 
toy\nthat is out of reach)": ["Many times a day", "A few times a day", "A few times a 
week", "Less than once a week", "Never"],
             "Does your child point to share interest with you? (e.g. pointing at 
an\ninteresting sight)": ["Many times a day", "A few times a day", "A few times a week", 
"Less than once a week", "Never"],
             "Does your child pretend? (e.g. care for dolls, talk on a toy phone)": 
["Many times a day", "A few times a day", "A few times a week", "Less than once a week", 
"Never"],
             "Does your child follow where you're looking?": ["Many times a day", "A few 
times a day", "A few times a week", "Less than once a week", "Never"],
             "If you or someone else in the family is visibly upset, does your child\n
show signs of wanting to comfort them? (e.g. stroking hair, hugging them)": ['Always',
'Usually', 'Sometimes', 'Rarely', 'Never'],
             "Would you describe your child's first words as:": ["Very typical", "Quite
typical", "Slightly unusual", "Very unusual", "My child doesn't speak"],
             "Does your child use simple gestures? (e.g. wave goodbye)": ["Many times a
day", "A few times a day", "A few times a week", "Less than once a week", "Never"],
             "Does your child stare at nothing with no apparent purpose?": ["Many times a
day", "A few times a day", "A few times a week", "Less than once a week", "Never"]
}
times_new_roman = 'Times New Roman'
ans_values = [0,0,0,0,0,0,0,0,0,0]
accept_ans = [0,0,0,0,0,0,0,0,0,0]
weights = [9,10,8,8,7,7,6,6,7,5]
temp = gettempdir()
hex_val = str(uuid4())

# Accepting from input form
data = argv[1]
data = data.split(', ')
email = data[0]
age = int(data[1])
sex = int(data[2])
ethnicity = int(data[3])
jaundice = int(data[4])
user = int(data[5])
family = int(data[6])

# Creating a window
root = Tk()
root.title("Data form")
icon = PhotoImage(file = '.images/form-icon.png')
root.iconphoto(False, icon)
window_width = 700
window_height = 400
root.geometry(f'{window_width}x{window_height}')
root.resizable(False, False)


# Centering the main window
def center_screen():
    global screen_height, screen_width, x_coordinate, y_coordinate
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()

    # Coordinates of the upper left corner of the window to make the window appear in the
center
    x_coordinate = int((screen_width/2) - (window_width/2))
    y_coordinate = int((screen_height/2) - (window_height/2))
    root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_coordinate,
y_coordinate))
center_screen()

frame1 = Frame(root, highlightbackground="blue", highlightthickness=1,width=680,
height=680, bd=2)
frame1.pack(pady=10, padx=10)
key_vals = list(questions.keys())
c = 0

# Question Numbers and Questions
q_no = Label(root, text=f'{c+1}.', font=(times_new_roman, 14, 'bold'))
q = Label(root, text=key_vals[c], font=(times_new_roman, 15), justify='left')
q_options = questions[key_vals[c]] # Options

ans = 0
# Radio Button function on click
def select_value():
    global ans, var, accept_ans
    ans = var.get()
    accept_ans [c] = var.get()
    if c==9 and (ans==1 or ans==2 or ans==3):
        ans_values[c] = 1
    elif c!=9 and (ans==3 or ans==4 or ans==5):
        ans_values[c] = 1
    else:
        ans_values[c] = 0
    # var.set(0)
var = IntVar()
```

```
# Radio buttons
r1 = Radiobutton(root, text=q_options[0], variable=var, value=1, command=select_value,
font=(times_new_roman, 14, 'italic'))
r2 = Radiobutton(root, text=q_options[1], variable=var, value=2, command=select_value,
font=(times_new_roman, 14, 'italic'))
r3 = Radiobutton(root, text=q_options[2], variable=var, value=3, command=select_value,
font=(times_new_roman, 14, 'italic'))
r4 = Radiobutton(root, text=q_options[3], variable=var, value=4, command=select_value,
font=(times_new_roman, 14, 'italic'))
r5 = Radiobutton(root, text=q_options[4], variable=var, value=5, command=select_value,
font=(times_new_roman, 14, 'italic'))

r1.place(x=80, y=100)
r2.place(x=80, y=140)
r3.place(x=80, y=180)
r4.place(x=80, y=220)
r5.place(x=80, y=260)

q_no.place(x=30, y=40)
q.place(x=80, y=40)
# q_part.place(x=30, y=50)

def update_question():
    global q_no, q, q_options, c
    q_no.config(text=f'{c+1}.')
    q.config(text=key_vals[c])

def update_options():
    global q_options, r1, r2, r3, r4, r5
    q_options= questions[key_vals[c]] # Extract Options

    # Reinitialize the radio-buttons
    r1.config(text=q_options[0])
    r2.config(text=q_options[1])
    r3.config(text=q_options[2])
    r4.config(text=q_options[3])
    r5.config(text=q_options[4])

def next_question():
    print('Ans 0/1:', ans_values, 'Ans options:', accept_ans)
    global ans, c, var
    if ans!=0:
        if c!=9:
            prev.configure(state='active', command=prev_question)
            var.set(0)
            c+=1
            update_question()
            update_options()
            if accept_ans[c] != 0:
                var.set(accept_ans[c])
                if var.get() != accept_ans[c]:
                    accept_ans[c] = var.get()
            if c==9:
                submit.config(state='active')
                next.configure(command='none', state='disabled')
            select_value()


def prev_question():
    print('Ans 0/1:', ans_values, 'Ans options:', accept_ans)

    global c
    if c!=0:
        if c == 9:
            next.configure(command=next_question, state='active')
            submit.configure(state='disabled')
        c-=1
        update_question()
        update_options()
        var.set(accept_ans[c])
        if c==0:
            prev.config(command='none', state='disabled')
        select_value()

prev_img = PhotoImage(file=r'.images/prev.png').subsample(3,3)
prev = Button(root, command='none', state='disabled', image=prev_img, borderwidth=0)
prev.place(x=80,y=325)
next_img = PhotoImage(file=r'.images/next.png').subsample(3,3)
```

```
next = Button(root, command=next_question, state='active',  image=next_img,
borderwidth=0)
next.place(x=580,y=325)
submit_img = PhotoImage(file=r'.images/submit.png').subsample(3,3)


def create_file(data):
    file = open(f'{temp}/data_{hex_val}.csv', 'w')
    data = ','.join(data)
    file.write(f'A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,Qchat-10-
Score,Age_Mons,Sex,Ethnicity,Jaundice,Who completed the
test,Family_mem_with_ASD\n{data}\n')
    file.close()
    file = open(f'{temp}/data_{hex_val}.csv', 'r')
    for i in file:
        print(i)


def submit_action():
    root.destroy()
    global ans_values
    print('Ans 0/1:', ans_values, 'Ans options:', accept_ans)
    q_chat_10_sum = sum(ans_values)
    weight_sum = 0
    for i in range(len(ans_values)):
        if ans_values[i] == 1:
            weight_sum += weights[i]
    ans_values = list(map(str, ans_values))
    ans_values1 = list(map(str, [q_chat_10_sum, age, sex, ethnicity, jaundice, user,
family]))
    name = data[7]
    create_file(ans_values+ans_values1)
    run(['python', 'model_loading_output.py', name, f'{temp}/data_{hex_val}.csv',
str(weight_sum)])

submit = Button(root, command=submit_action, state='disabled', image=submit_img,
borderwidth=0)
submit.place(x=275, y=325)
root.mainloop()
```

config.py

```
mail_id = "noreplyasddetection@gmail.com"
password = "oriv wqnc skrn jsgw"
```

download_model.py

```
import firebase_admin
from firebase_admin import credentials, storage
from google.oauth2 import service_account
from google.cloud import storage
from tempfile import gettempdir
from os.path import exists
from os import mkdir

def download():
    temp = gettempdir()
    folder_path = f'{temp}/models'
    if not exists(f'{temp}/models'):
        mkdir(f'{temp}/models')
    cred = credentials.Certificate(".keys/key.json")
    firebase_admin.initialize_app(cred,{'storageBucket': 'asd-model.appspot.com'})

    cred_var = service_account.Credentials.from_service_account_file(".keys/key.json")

storage.Client(credentials=cred_var).bucket(firebase_admin.storage.bucket().name).blob('s
vc_new.joblib').download_to_filename(f'{folder_path}/svc_new.joblib')
    return f'{folder_path}/svc_new.joblib'
```

model_loading_output.py

```
from pandas import read_csv
from joblib import load
from tkinter import Tk, PhotoImage,Label,Button
from sys import argv
from download_model import download
from sklearn.preprocessing import LabelEncoder
from numpy import log
from time import sleep
from threading import Thread
import firebase_admin
from firebase_admin import credentials, storage
from google.oauth2 import service_account
from google.cloud import storage
from os.path import exists
from os import mkdir
from tkinter import filedialog
from tkinter import messagebox

per = 0
dir_name = filedialog.askdirectory()
folder_path = f'{dir_name}/Treatment_Plan'
if not exists(folder_path):
    mkdir(folder_path)
cred = credentials.Certificate(".keys/key.json")
cred_var = service_account.Credentials.from_service_account_file(".keys/key.json")

# Global variables
counter = 3
running = True

name = argv[1]
data_to_eval = argv[2]
weight = int(argv[3])

root = Tk()
root.title("Evaluation window")
icon = PhotoImage(file = '.images/result.png')

window_width = 700
window_height = 400

root.iconphoto(False, icon)
root.geometry(f'{window_width}x{window_height}')
root.resizable(False, False)


def center_screen():
    """ gets the coordinates of the center of the screen """
    global screen_height, screen_width, x_coordinate, y_coordinate
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()
    # Coordinates of the upper left corner of the window to make the window appear in the
center
    x_coordinate = int((screen_width/2) - (window_width/2))
    y_coordinate = int((screen_height/2) - (window_height/2))
    root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_coordinate,
y_coordinate))
center_screen()


def encode_labels(data):
    for col in data.columns:
        # Here we will check if datatype
        # is object then we will encode it
        if data[col].dtype == object:
            le = LabelEncoder()
            data[col] = le.fit_transform(data[col])
    return data


label_img_1 = PhotoImage(file=".images/evaluating.png").subsample(2, 2)
label1 = Label(image=label_img_1, state='active')
label1.place(anchor='center', relx=0.5, rely=0.5)


def update_label():
    global counter, running, label1
    while counter > 0 and running:
        counter -= 1
        sleep(1)

def send_treatment_plan():
    global per
    if result == [1]:
        if 10<=per<=33:

storage.Client(credentials=cred_var).bucket(firebase_admin.storage.bucket().name).blob('M
ild ASD.pdf').download_to_filename(f'{folder_path}/Mild ASD.pdf')
            messagebox.showinfo('Info', 'File Downloaded!\nClose the window')
        elif per<=66:

storage.Client(credentials=cred_var).bucket(firebase_admin.storage.bucket().name).blob('M
oderate ASD.pdf').download_to_filename(f'{folder_path}/Moderate ASD.pdf')
            messagebox.showinfo('Info', 'File Downloaded!\nClose the window')
        else:

storage.Client(credentials=cred_var).bucket(firebase_admin.storage.bucket().name).blob('S
evere ASD.pdf').download_to_filename(f'{folder_path}/Severe ASD.pdf')
            messagebox.showinfo('Info', 'File Downloaded!\nClose the window')


def evaluate_model():
    global label1, label2, result, running, per
    sleep(3)
    if running:
        file_Path = download()
        loaded_model = load(file_Path)
        df1 = read_csv(data_to_eval)
        df1['Age_Mons'] = df1['Age_Mons'].apply(lambda x: log(x))
        df1 = df1.replace({'yes':1, 'no':0, 'Yes':1, 'No':0, '?':'others',
'Others':'others'})
        df1 = encode_labels(df1)
        # print(name,'\n', df1)
        result = loaded_model.predict(df1)
        text = ''
        per = (weight/73)*100

        if per<10:
            text='no'
        elif 10<=per<=33:
            text = 'mild'
        elif 34<=per<=66:
            text = 'moderate'
        else:
            text = 'severe'

        b = Button(root, text='Get treatment plan',  font=('Times New Roman', 14,
'italic'), command=send_treatment_plan, relief='groove', state='disabled')
        b.place(anchor='n', relx=0.5, rely=.6)

        if result == [0]:
            text += ' chance of'
        else:
            b.configure(state='active')

        per = "{:.2f}".format(per)
        label1.configure(text=f'{name} has {per}% {text} ASD.', image="", font=('Times
New Roman', 14, 'italic'))
        per = float(per)
        print(folder_path)

# Start the threads
countdown_thread = Thread(target=update_label)
evaluating_thread = Thread(target=evaluate_model)

countdown_thread.start()
evaluating_thread.start()

root.mainloop()
```

send_treatment_plan.py

```
# All imports below are part of python built packages no need to install any exras

# smtplib provides functionality to send emails using SMTP.
import smtplib
# MIMEMultipart send emails with both text content and attachments.
from email.mime.multipart import MIMEMultipart
# MIMEText for creating body of the email message.
from email.mime.text import MIMEText
# MIMEApplication attaching application-specific data (like CSV files) to email messages.
from email.mime.application import MIMEApplication
from config import mail_id, password
from sys import argv


subject = "Treatment plan"
body = "Please adhere to the plan as sent by the team.\nThanking You,\nTeam Cognitive
Healers"
sender_email = mail_id
recipient_email = argv[1]
sender_password = password
smtp_server = 'smtp.gmail.com'
smtp_port = 465
path_to_file = argv[2]

# MIMEMultipart() creates a container for an email message that can hold
# different parts, like text and attachments and in next line we are
# attaching different parts to email container like subject and others.
message = MIMEMultipart()
message['Subject'] = subject
message['From'] = sender_email
message['To'] = recipient_email
body_part = MIMEText(body)
message.attach(body_part)

# section 1 to attach file
with open(path_to_file,'rb') as file:
    # Attach the file with filename to the email
    message.attach(MIMEApplication(file.read(), Name="treatment_plan.pdf"))

# secction 2 for sending email
with smtplib.SMTP_SSL(smtp_server, smtp_port) as server:
   server.login(sender_email, sender_password)
   server.sendmail(sender_email, recipient_email, message.as_string())
```

XI. APPENDIX III

a. Execution manual:

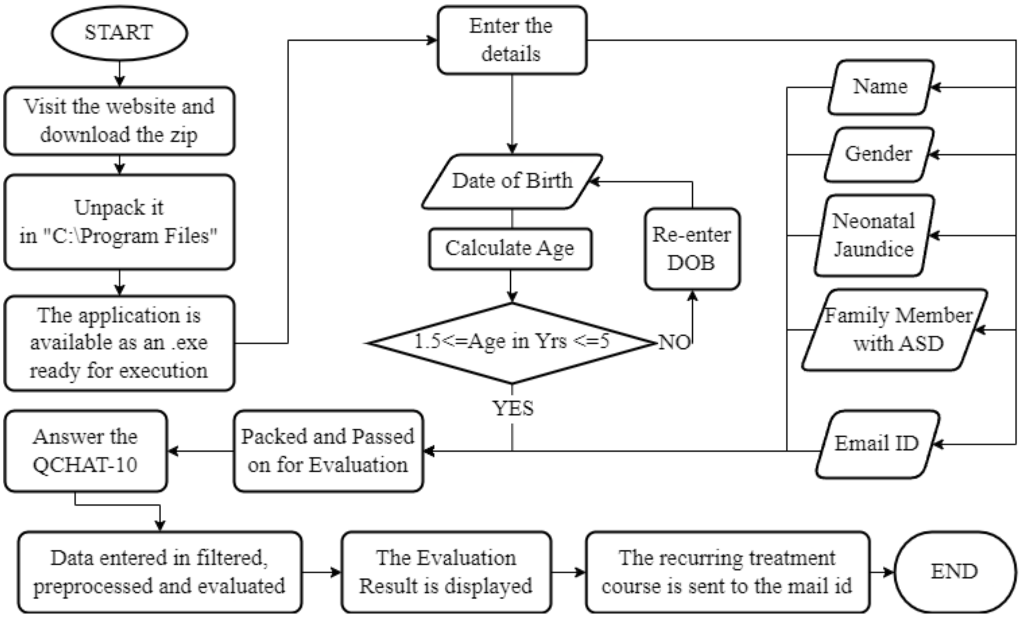
a. Note:

i. The Name of the child and email will not be stored due to patient's privacy concerns.

ii. The application relies on internet connectivity for two main purposes: firstly, to send a confirmation email after the evaluation is submitted, and secondly, to deliver a tailored treatment plan via email once the assessment of ASD level is completed for the respective test taker.

b. Steps to execute the application:

As depicted in ***Figure 7***, the flow progresses as follows:



START
Visit the website and download the zip
Unpack it in "C:\Program Files"
The application is available as an .exe ready for execution
Enter the details
Date of Birth
Calculate Age
1.5<=Age in Yrs <=5
NO
Re-enter DOB
YES
Name
Gender
Neonatal Jaundice
Family Member with ASD
Email ID
Packed and Passed on for Evaluation
Answer the QCHAT-10
Data entered in filtered, preprocessed and evaluated
The Evaluation Result is displayed
The recurring treatment course is sent to the mail id
END

Figure 7: Flow of Application Usage on User's End

Step 1: - Start the Process: Begin the ASD evaluation process.

Step 2: - Visit the Website: Go to the specified website and download the zip file.

Step 3: - Unpack the Zip File: Unpack the downloaded zip file into "C:\Program Files".

Step 4: - Install the Application: The application is available as an executable (.exe) file, ready for execution.

Step 5: - Enter User Details: Input the following details:
Name, Gender, Date of Birth, Ethnicity, Neonatal Jaundice (Yes/No), Family Member with ASD (Yes/No), Email ID

Step 6: - Calculate Age: Calculate the child's age based on the Date of Birth provided.

Step 7: - Check Age Validity:
- If the age is between 1.5 and 5 years, proceed to the next step.

- If the age is outside this range, prompt to re-enter the Date of Birth.

Step 8: - Answer QCHAT-10: Answer the QCHAT-10 questionnaire.

Step 9: - Pack Data for Evaluation: The entered data is packed and passed on for evaluation.

Step 10: - Data Processing: The data is filtered, preprocessed, and evaluated.

Step 11: - Display Evaluation Result: The evaluation result is displayed to the user.

Step 12: - Send Treatment Plan: A recurring treatment plan is generated and sent to the provided email ID.

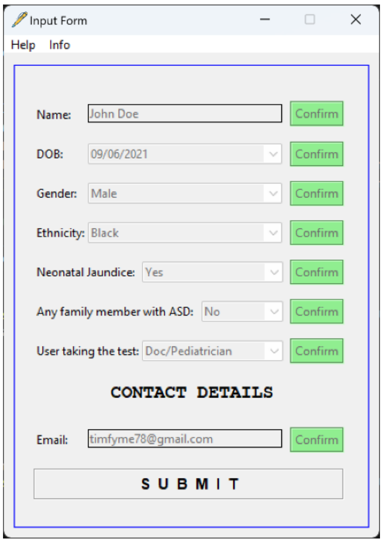
Step 13: - End the Process: Conclude the ASD evaluation process.

XII. APPENDIX IV

a. Interface

Input Form:

Enter the details as follows: This will capture the name of the patient, gender, DOB, ethnicity, whether the child has neonatal jaundice or not, whether there is a family history of ASD, and who is the user taking the test and email for confirmation and treatment plan.



Name: John Doe

DOB: 09/06/2021

Gender: Male

Ethnicity: Black

Neonatal Jaundice: Yes

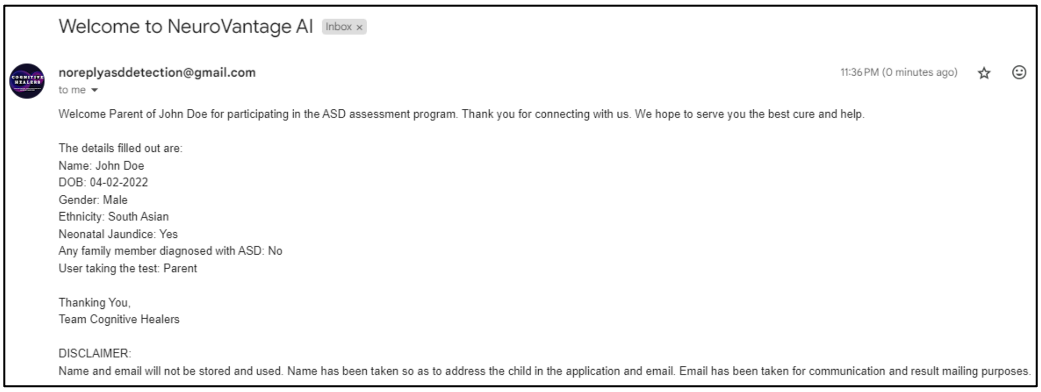
Any family member with ASD: No

User taking the test: Doc/Pediatrician

Email: timfyme78@gmail.com

Confirmation Mail:

The mail sent will the hold the data for confirmation.



Welcome to NeuroVantage AI

noreplyasddetection@gmail.com
to me

Welcome Parent of John Doe for participating in the ASD assessment program. Thank you for connecting with us. We hope to serve you the best cure and help.

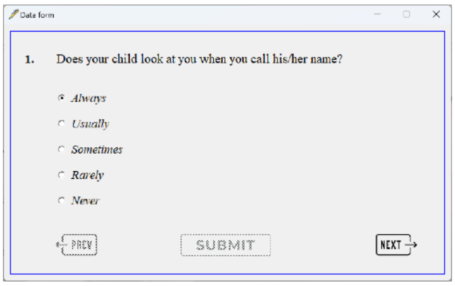
The details filled out are:
Name: John Doe
DOB: 04-02-2022
Gender: Male
Ethnicity: South Asian
Neonatal Jaundice: Yes
Any family member diagnosed with ASD: No
User taking the test: Parent

Thanking You,
Team Cognitive Healers

DISCLAIMER:
Name and email will not be stored and used. Name has been taken so as to address the child in the application and email. Email has been taken for communication and result mailing purposes.

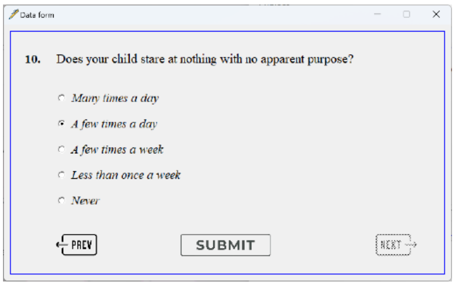
Questionaries:

The questions are answered as a click on the radio button and the value captured is either in format of 0/1. For questions 1-9 if options 3-5 are answered the value captured is 1 else 0 and for question 10 if option 1-3 are answered the value captured is 1 else 0.



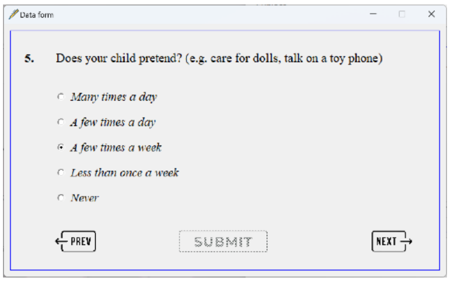
1. Does your child look at you when you call his/her name?

● *Always*
○ *Usually*
○ *Sometimes*
○ *Rarely*
○ *Never*



10. Does your child stare at nothing with no apparent purpose?

○ *Many times a day*
● *A few times a day*
○ *A few times a week*
○ *Less than once a week*
○ *Never*

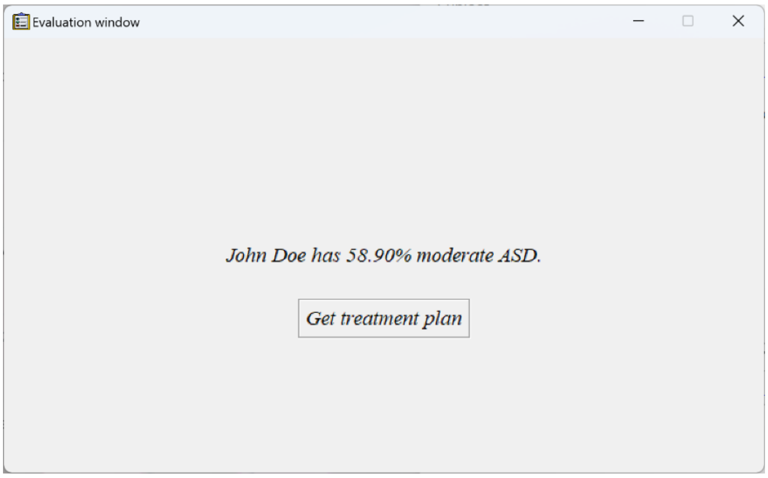


5. Does your child pretend? (e.g. care for dolls, talk on a toy phone)

○ *Many times a day*
○ *A few times a day*
● *A few times a week*
○ *Less than once a week*
○ *Never*

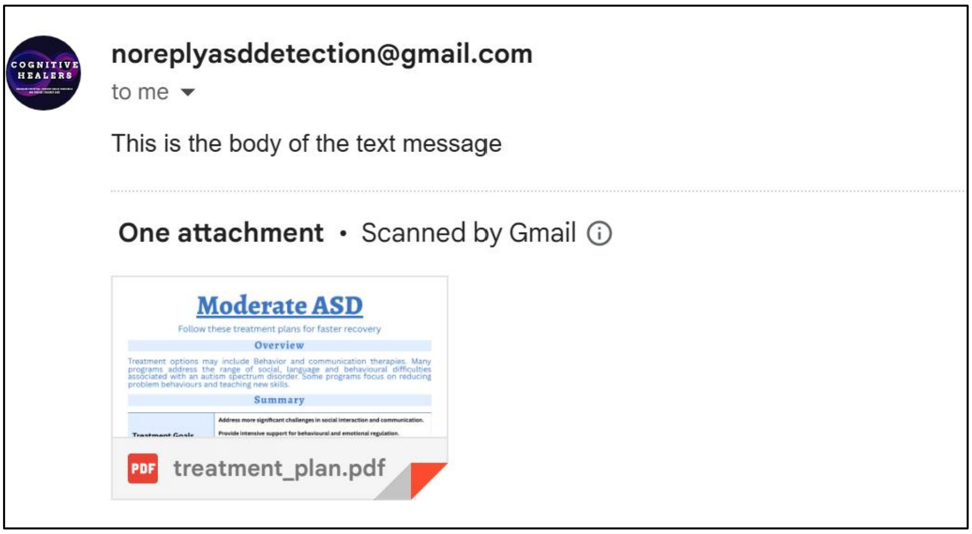
Treatment Plan and Mail:

This image demonstrates the evaluation output window and on clicking the Get Treatment Plan the Treatment Plan mail will be directed to the patient on the entered email.



John Doe has 58.90% moderate ASD.

The Evaluation Output Window



noreplyasddetection@gmail.com

to me

This is the body of the text message

One attachment • Scanned by Gmail

Moderate ASD

Follow these treatment plans for faster recovery

Overview

Treatment options may include Behavior and communication therapies. Many programs address the range of social, language and behavioural difficulties associated with an autism spectrum disorder. Some programs focus on reducing problem behaviours and teaching new skills.

Summary

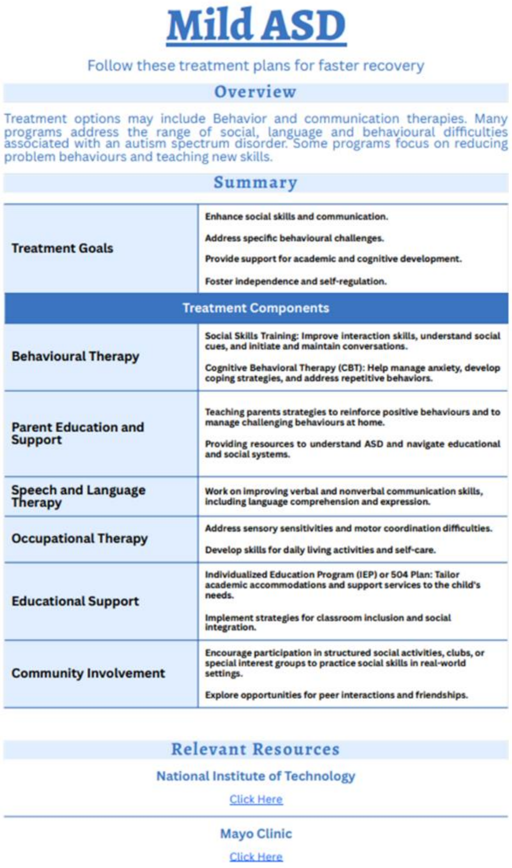
Treatment Goals	Address more significant challenges in social interaction and communication. Provide intensive support for behavioural and emotional regulation.

treatment_plan.pdf

Treatment Plan Mail

Treatment Plans:

For mild autism spectrum disorder (ASD), treatment typically focuses on educational interventions and therapies that enhance social skills, communication abilities, and behavioral management. Applied Behavior Analysis (ABA) is commonly used to teach specific skills and reduce problem behaviors. Speech therapy may also be employed to improve communication. Parental involvement and support are crucial in implementing strategies both at home and in educational settings.



Mild ASD

Follow these treatment plans for faster recovery

Overview

Treatment options may include Behavior and communication therapies. Many programs address the range of social, language and behavioural difficulties associated with an autism spectrum disorder. Some programs focus on reducing problem behaviours and teaching new skills.

Summary

Treatment Goals	Enhance social skills and communication. Address specific behavioural challenges. Provide support for academic and cognitive development. Foster independence and self-regulation.
Treatment Components	
Behavioural Therapy	Social Skills Training: Improve interaction skills, understand social cues, and initiate and maintain conversations. Cognitive Behavioral Therapy (CBT): Help manage anxiety, develop coping strategies, and address repetitive behaviors.
Parent Education and Support	Teaching parents strategies to reinforce positive behaviours and to manage challenging behaviours at home. Providing resources to understand ASD and navigate educational and social systems.
Speech and Language Therapy	Work on improving verbal and nonverbal communication skills, including language comprehension and expression.
Occupational Therapy	Address sensory sensitivities and motor coordination difficulties. Develop skills for daily living activities and self-care.
Educational Support	Individualized Education Program (IEP) or 504 Plan: Tailor academic accommodations and support services to the child's needs. Implement strategies for classroom inclusion and social integration.
Community Involvement	Encourage participation in structured social activities, clubs, or special interest groups to practice social skills in real-world settings. Explore opportunities for peer interactions and friendships.

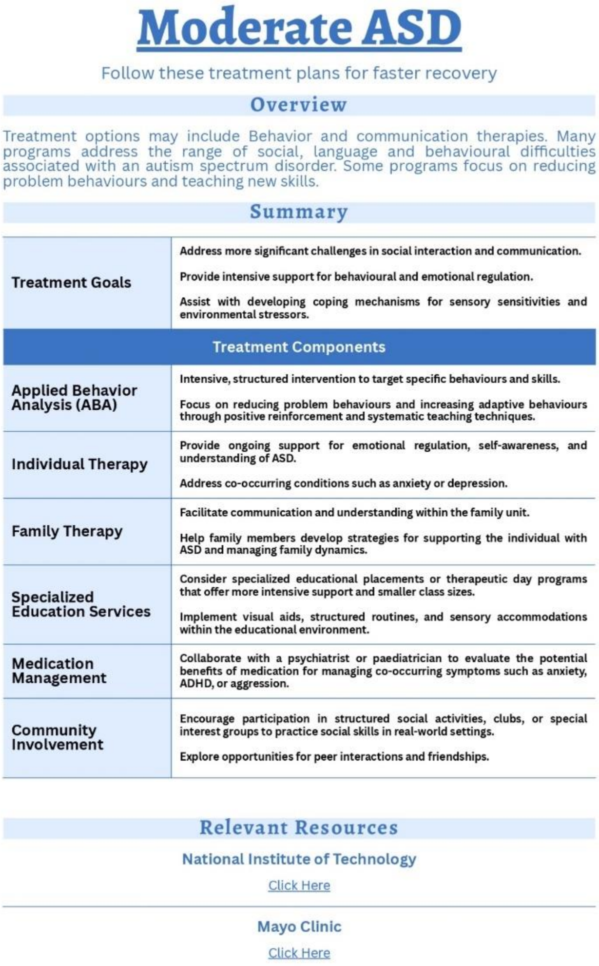
Relevant Resources

National Institute of Technology

Click Here

Mayo Clinic

Click Here



Moderate ASD

Follow these treatment plans for faster recovery

Overview

Treatment options may include Behavior and communication therapies. Many programs address the range of social, language and behavioural difficulties associated with an autism spectrum disorder. Some programs focus on reducing problem behaviours and teaching new skills.

Summary

Treatment Goals	Address more significant challenges in social interaction and communication. Provide intensive support for behavioural and emotional regulation. Assist with developing coping mechanisms for sensory sensitivities and environmental stressors.
Treatment Components	
Applied Behavior Analysis (ABA)	Intensive, structured intervention to target specific behaviours and skills. Focus on reducing problem behaviours and increasing adaptive behaviours through positive reinforcement and systematic teaching techniques.
Individual Therapy	Provide ongoing support for emotional regulation, self-awareness, and understanding of ASD. Address co-occurring conditions such as anxiety or depression.
Family Therapy	Facilitate communication and understanding within the family unit. Help family members develop strategies for supporting the individual with ASD and managing family dynamics.
Specialized Education Services	Consider specialized educational placements or therapeutic day programs that offer more intensive support and smaller class sizes. Implement visual aids, structured routines, and sensory accommodations within the educational environment.
Medication Management	Collaborate with a psychiatrist or paediatrician to evaluate the potential benefits of medication for managing co-occurring symptoms such as anxiety, ADHD, or aggression.
Community Involvement	Encourage participation in structured social activities, clubs, or special interest groups to practice social skills in real-world settings. Explore opportunities for peer interactions and friendships.

Relevant Resources

National Institute of Technology

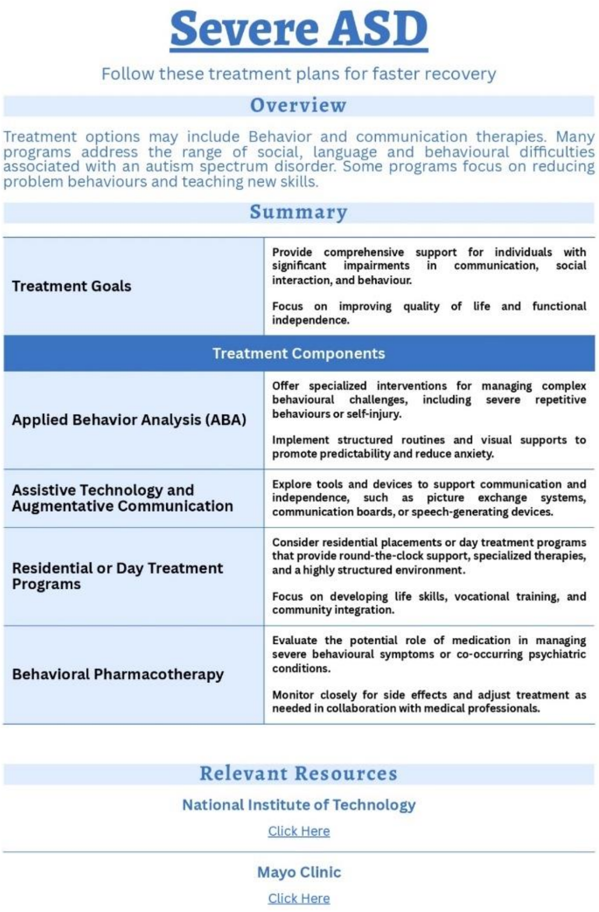
Click Here

Mayo Clinic

Click Here

Moderate ASD involves more intensive interventions, often including a combination of behavioral therapies (such as ABA), speech therapy, occupational therapy, and social skills training. Individualized education plans (IEPs) are developed to address academic, social, and behavioral needs. Specialized support services in school settings help facilitate learning and social integration. Family counseling and support groups play a significant role in providing strategies and emotional support for parents and caregivers.

Severe ASD treatment plans are highly individualized and intensive, typically involving comprehensive therapies and structured environments tailored to the person's specific needs. This may include 24/7 support services, residential treatment programs, and specialized educational settings designed to address significant developmental challenges. Behavioral interventions, communication strategies (including augmentative and alternative communication methods), sensory integration therapy, and medical management of co-occurring conditions are key components. Collaborative efforts among caregivers, educators, therapists, and healthcare professionals are essential in managing severe ASD effectively.



Severe ASD

Follow these treatment plans for faster recovery

Overview

Treatment options may include Behavior and communication therapies. Many programs address the range of social, language and behavioural difficulties associated with an autism spectrum disorder. Some programs focus on reducing problem behaviours and teaching new skills.

Summary

Treatment Goals	Provide comprehensive support for individuals with significant impairments in communication, social interaction, and behaviour. Focus on improving quality of life and functional independence.
Treatment Components	
Applied Behavior Analysis (ABA)	Offer specialized interventions for managing complex behavioural challenges, including severe repetitive behaviours or self-injury. Implement structured routines and visual supports to promote predictability and reduce anxiety.
Assistive Technology and Augmentative Communication	Explore tools and devices to support communication and independence, such as picture exchange systems, communication boards, or speech-generating devices.
Residential or Day Treatment Programs	Consider residential placements or day treatment programs that provide round-the-clock support, specialized therapies, and a highly structured environment. Focus on developing life skills, vocational training, and community integration.
Behavioral Pharmacotherapy	Evaluate the potential role of medication in managing severe behavioural symptoms or co-occurring psychiatric conditions. Monitor closely for side effects and adjust treatment as needed in collaboration with medical professionals.

Relevant Resources

National Institute of Technology

Click Here

Mayo Clinic

Click Here

BIBLIOGRAPHY

[1] APA—DSM—Diagnostic and Statistical Manual of Mental Disorders. (2013). https://www.appi.org/dsm

[2] Hus, Yvette, and Osnat Segal. "Challenges surrounding the diagnosis of autism in children." Neuropsychiatric disease and treatment (2021): 3509-3529.

[3] Lord, Catherine, Mayada Elsabbagh, Gillian Baird, and Jeremy Veenstra-Vanderweele. "Autism spectrum disorder." The lancet 392, no. 10146 (2018): 508-520.

[4] Zeidan, Jinan, Eric Fombonne, Julie Scorah, Alaa Ibrahim, Maureen S. Durkin, Shekhar Saxena, Afiqah Yusuf, Andy Shih, and Mayada Elsabbagh. "Global prevalence of autism: A systematic review update." Autism research 15, no. 5 (2022): 778-790.

[5] Maenner, Matthew J. "Prevalence of autism spectrum disorder among children aged 8 years—autism and developmental disabilities monitoring network, 11 sites, United States, 2016." MMWR. Surveillance Summaries 69 (2020).

[6] Kuo, Alice A., Kristy A. Anderson, Tara Crapnell, Lynette Lau, and Paul T. Shattuck. "Introduction to transitions in the life course of autism and other developmental disabilities." Pediatrics 141, no. Supplement_4 (2018): S267-S271.

[7] Shattuck, Paul, and Lindsay Shea. "Improving care and service delivery for autistic youths transitioning to adulthood." Pediatrics 149, no. Supplement 4 (2022).

[8] Kumar, P., & Kumar, P. (2024, May 13). All you need to know about early detection and diagnosis of autism. India Autism Center. https://rb.gy/nyxqzx

[9] Bosl, William J., Helen Tager-Flusberg, and Charles A. Nelson. "EEG analytics for early detection of autism spectrum disorder: a data-driven approach." Scientific reports 8, no. 1 (2018): 6828.

[10] Fernell, Elisabeth, Mats Anders Eriksson, and Christopher Gillberg. "Early diagnosis of autism and impact on prognosis: a narrative review." Clinical epidemiology (2013): 33-43.

[11] AMIHAESEI, Ioana Cristina, and Elena Stefanachi. "Autism, an overwhelming condition: history, etiopathogenesis, types, diagnosis, therapy and prognosis." The Medical-Surgical Journal 117, no. 3 (2013): 654-661.

[12] Witwer, Andrea N., and Luc Lecavalier. "Examining the validity of autism spectrum disorder subtypes." Journal of autism and developmental disorders 38 (2008): 1611-1624.

[13] El-Gazzar, Ahmed, Mirjam Quaak, Leonardo Cerliani, Peter Bloem, Guido van Wingen, and Rajat Mani Thomas. "A hybrid 3DCNN and 3DC-LSTM based model for 4D spatio-temporal fMRI data: an ABIDE autism classification study." In International Workshop on OR 2.0 Context-Aware Operating Theaters, pp. 95-102. Cham: Springer International Publishing, 2019.

[14] Hossain, Md Delowar, Muhammad Ashad Kabir, Adnan Anwar, and Md Zahidul Islam. "Detecting autism spectrum disorder using machine learning techniques: An

experimental analysis on toddler, child, adolescent and adult datasets." Health Information Science and Systems 9 (2021): 1-13.

[15] Ahmed, Ibrahim Abdulrab, Ebrahim Mohammed Senan, Taha H. Rassem, Mohammed AH Ali, Hamzeh Salameh Ahmad Shatnawi, Salwa Mutahar Alwazer, and Mohammed Alshahrani. "Eye tracking-based diagnosis and early detection of autism spectrum disorder using machine learning and deep learning techniques." Electronics 11, no. 4 (2022): 530.

[16] Raj, Suman, and Sarfaraz Masood. "Analysis and detection of autism spectrum disorder using machine learning techniques." Procedia Computer Science 167 (2020): 994-1004.

[17] Farooq, Muhammad Shoaib, Rabia Tehseen, Maidah Sabir, and Zabihullah Atal. "Detection of autism spectrum disorder (ASD) in children and adults using machine learning." Scientific Reports 13, no. 1 (2023): 9605.

[18] Parvathi, M. "Early detection support mechanism in ASD using ML classifier." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, no. 10 (2021): 4543-4549.

[19] Akter, Tania, Md Shahriare Satu, Md Imran Khan, Mohammad Hanif Ali, Shahadat Uddin, Pietro Lio, Julian MW Quinn, and Mohammad Ali Moni. "Machine learning-based models for early-stage detection of autism spectrum disorders." IEEE Access 7 (2019): 166509-166527.

[20] Vakadkar, Kaushik, Diya Purkayastha, and Deepa Krishnan. "Detection of autism spectrum disorder in children using machine learning techniques." SN computer science 2 (2021): 1-9.

[21] Jiao, Yun, Rong Chen, Xiaoyan Ke, Kangkang Chu, Zuhong Lu, and Edward H. Herskovits. "Predictive models of autism spectrum disorder based on brain regional cortical thickness." Neuroimage 50, no. 2 (2010): 589-599.

[22] Mujeeb Rahman, K. K., and M. Monica Subashini. "A deep neural network-based model for screening autism spectrum disorder using the quantitative checklist for autism in toddlers (QCHAT)." Journal of Autism and Developmental Disorders 52, no. 6 (2022): 2732-2746.

[23] Kumar, Chandan Jyoti, and Priti Rekha Das. "The diagnosis of ASD using multiple machine learning techniques." International Journal of Developmental Disabilities 68, no. 6 (2022): 973-983.

[24] Yücelbaş, Şule, and Cüneyt Yücelbaş. "Autism spectrum disorder detection using sequential minimal optimization-support vector machine hybrid classifier according to history of jaundice and family autism in children." Concurrency and Computation: Practice and Experience 34, no. 1 (2022): e6498.