ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

**Research Paper**

# EDUCATIONAL INTERVENTIONS FORAUTISM SPECTRUM DISORDER: PAST, PRESENT, AND FUTURE PERSPECTIVES

[1]N.PRASANNA [2]Mrs.P.SHRADDHA
*[1]M.Tech Student, [2]Assistant Professor*
*Department of Computer Science And Engineering*
*KLR College Of Engineering And Technology, B.C.M Road, Paloncha, Bhadradri Kothagudem Dist.,Telangana,5071*

**Abstract—**

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition characterized by challenges in social interaction, communication, behavior, and adaptive functioning. Early identification is critical, as prompt intervention can significantly improve language skills, learning, behavioral adaptation, and social integration. However, conventional diagnostic approaches are often resource-intensive, time-consuming, and dependent on specialist availability. These constraints limit the accessibility and timeliness of ASD screening, particularly in settings with limited resources.
To address these challenges, this project introduces a web-based predictive platform: A Machine Learning Framework for Early Detection of Autism Spectrum Disorders. Built with the Django web framework, the platform utilizes powerful Python libraries—including Pandas, NumPy, Matplotlib, and Scikit-learn—for robust data processing and predictive analysis. It incorporates four distinct real-world datasets (Toddler, Child, Adolescent, and Adult), each featuring behavioral questionnaire responses, demographic details (such as age, gender, and history of jaundice), and the ASD_traits outcome variable. Data preparation steps include label encoding, handling missing values, feature scaling, and splitting datasets for training and validation.
The system employs a comprehensive set of supervised machine learning algorithms: AdaBoost, Random Forest, Decision Tree, K-Nearest Neighbors, Gaussian Naive Bayes, Logistic Regression, Support Vector Machine, and Linear Discriminant Analysis. An ensemble Voting Classifier is used to enhance prediction accuracy. The framework is guided by two primary objectives: (1) to rigorously evaluate and compare model performance across different age-group datasets, and (2) to deliver an intuitive user interface where individuals can input behavioral and demographic information to obtain a preliminary ASD risk assessment. The ensemble classifier generates the final assessment, with outcomes presented using metrics such as accuracy, precision, recall, and F1-score, as well as visualizations to aid interpretation and education.
This machine learning framework demonstrates the potential of AI-driven tools to support healthcare decision-making while complementing—rather than replacing—professional clinical expertise. Designed as an initial screening resource, the system streamlines the assessment process, promotes consistency, and broadens access to decision support across all age groups. Ultimately, it offers a scalable, cost-effective, and accessible solution for early ASD screening by leveraging machine learning techniques on questionnaire-based data.

## INTRODUCTION

Autism Spectrum Disorder (ASD) is a lifelong neurodevelopmental condition marked by ongoing difficulties in social communication, interaction, restricted interests, and repetitive behaviors. Being a spectrum disorder, ASD manifests differently in each individual: some may only show subtle symptoms that go unnoticed for years, while others display clear developmental differences early in life. ASD can impact language, cognitive abilities, adaptive skills, educational attainment, and social functioning. This makes early identification crucial for effective intervention. Early-start interventions—such as educational therapy, speech and language support, family counseling, and behavioral training—yield the most favorable outcomes. As a result, both healthcare professionals and computational researchers are focusing on developing fast, reliable screening tools.
Traditional assessment for ASD typically includes structured observation, developmental history interviews, caregiver questionnaires, and expert analysis. Although these clinically validated methods are effective, they are resource-intensive and require specialized professionals. Short-form questionnaire-based screeners are more accessible, but manual scoring can be inconsistent and slow. In areas with limited access to specialists, wait times between initial concern and formal evaluation may be long. This highlights the need for intelligent computational solutions that can process behavioral screening data efficiently and deliver consistent, timely results. Machine learning is particularly well-suited for this

role, as it can uncover complex patterns in structured data and accurately classify new cases.

Problem Statement:

This project addresses these challenges by developing a web-based autism prediction system for multiple age groups. Unlike approaches limited to a single dataset, this framework integrates four datasets tailored to toddlers, children, adolescents, and adults. This distinction is important, as behavioral screening questions and demographic patterns differ across developmental stages. The platform provides a unified interface for loading datasets, applying feature scaling, training various machine learning models, comparing their performance, and predicting ASD risk for new cases. This makes the project academically relevant, technically instructive, and practically valuable as an introduction to healthcare analytics.

The system is built with Django as the web framework and Python for backend logic. Key libraries include Pandas for data processing, NumPy for numerical operations, Scikit-learn for data preprocessing and modeling, and Matplotlib for visualizing model performance. Data is sourced from four CSV files—Toddler.csv, Children.csv, Adolescent.csv, and Adults.csv. Instead of using every available column, the system selects core features from each dataset. Behavioral items A1 to A10 are the primary input features, supported by demographic variables like age, gender/sex, and jaundice history. The ASD_traits column serves as the target label for model training.

Data preprocessing is a vital step in the workflow. The datasets contain both numerical and categorical variables, yet machine learning models require numeric input. To address this, the system uses label encoding for categorical features, fills missing values with zeros, and applies feature scaling. Two scaling methods are used: the Normalizer for toddler and child datasets, and the QuantileTransformer for adolescent and adult datasets. After preprocessing, the data is split into training and testing sets, and multiple classifiers are trained and evaluated based on predictive accuracy.

Limitations:

- **Questionnaire Reliance:** The system depends on self- or caregiver-reported questionnaires, which can introduce bias or inaccuracies and may not capture behaviors outside the survey context.
- **Dataset Scope:** The four datasets (Toddler, Child, Adolescent, Adult) are limited in size and diversity, potentially affecting how well results generalize across cultures, languages, and socioeconomic backgrounds.
- **Feature Constraints:** Only demographic data (age, gender, jaundice history) and behavioral responses are used; important information such as medical, genetic, or environmental factors is not included.
- **Model Transparency:** Ensemble classifiers may achieve higher accuracy but can reduce interpretability, making it harder for clinicians and educators to understand individual predictions.
- **Not a Diagnostic Tool:** The system is meant for pre-screening and decision support only and should not replace professional medical assessment or diagnosis.
- **Resource Requirements:** Although web-based, the system requires internet access, computational resources, and basic digital literacy, which may limit its use in under-resourced settings.
- **Static Assessment:** The system assesses ASD traits at a single point in time and does not support ongoing monitoring or adaptive learning.

## LITERATURE SURVEY

### •Comprehensive Machine Learning Frameworks for ASD Detection

*Authors: Tania Akter, Md. Shahriare Satu, Md. Imran Khan, Mohammad Hanif Ali, Shahadat Uddin, Pietro Lio, Julian M. W. Quinn, and Mohammad Ali Moni*

This study introduces a robust machine learning framework for early autism detection, leveraging screening data from multiple age groups. The authors systematically evaluated various classification algorithms, measuring their effectiveness using standard performance metrics. Their findings confirm that combining questionnaire responses with demographic information enables accurate identification of ASD risk patterns at early stages. This work directly supports the current project by validating the use of multi-age datasets and comparative classifier analysis.

### •Efficient Machine Learning Models for ASD Detection Across Age Groups

*Authors: Mousumi Bala, Mohammad Hanif Ali, Md. Shahriare Satu, Khondokar Fida Hasan, and Mohammad Ali Moni*

This paper presents optimized machine learning models for autism detection by analyzing screening data spanning toddlers to adults. It emphasizes the critical role of feature scaling, selection, and interpretability in enhancing model performance. The study compares several classifiers and investigates how individual features impact predictions across different age groups. These strategies closely align with the present project, particularly the focus on developmental groupings and tailored preprocessing before classification.

**•Integrating Questionnaires and Home Videos in ASD Screening**

*Authors: Halim Abbas, Ford Garberson, Eric Glover, and Dennis P. Wall*

This research explores combining short caregiver questionnaires with home video-based behavioral tagging to improve early autism screening. Utilizing machine learning, the authors developed a screening tool that reduces both cost and assessment time while maintaining high predictive accuracy. Their approach addresses real-world issues such as data sparsity and class imbalance, demonstrating that computational methods can expedite access to early intervention. The study underscores the value of machine learning as an assistive layer within diagnostic workflows.

**•Reducing Observation-Based Screening with Machine Learning**

*Authors: Dennis P. Wall, J. Kosmicki, T. F. DeLuca, E. Harstad, and V. A. Fusaro*

This paper demonstrates that machine learning can significantly streamline autism observation protocols by identifying a minimal yet highly informative subset of behavioral features. Analysis of clinical data shows that a small group of key features achieves high sensitivity and specificity in ASD classification. This research laid the groundwork for feature reduction techniques, supporting the current project's emphasis on targeted behavioral indicators for efficient early-stage prediction.

**•Computational Intelligence for Rapid and Cost-Effective Autism Screening**

*Authors: Fadi Thabtah, Firuz Kamalov, and Khairan Rajab*

This study introduces a computational intelligence-driven approach to autism screening, focusing on structured questionnaire data and predictive modeling. The authors illustrate how well-organized responses can be efficiently converted into reliable classification results, enabling rapid, scalable, and affordable screening. This methodology is directly relevant to the current project, which also relies on questionnaire datasets and machine learning for pre-diagnostic support. The work further highlights the importance of straightforward input formats for broad community screening.

## SYSTEM ANALYSIS

### EXISTING SYSTEM

Autism identification in most settings is still rooted in manual processes, relying on screening questionnaires, developmental observations, parent interviews, and comprehensive clinical evaluations by specialists such as psychologists, pediatricians, or behavioral experts. Typically, evaluation is only initiated after caregivers or educators notice differences in a child's communication, social interaction, or behavioral patterns. Although standardized screening tools exist, their results are usually interpreted manually, followed by referrals to specialized centers for further diagnosis.

Because autism can present very differently depending on age and severity, the assessment process often requires several sequential steps, leading to substantial delays from initial concern through screening, diagnosis, and, ultimately, the start of intervention. While this traditional method is thorough and vital for clinical confirmation, it is often slow, resource-heavy, and may not be accessible to all communities.

From a systems perspective, the conventional approach is limited by its dependence on specialist expertise and the availability of dedicated services. Many settings lack immediate specialist access, and while questionnaires are frequently used, the resulting data is not always leveraged for data-driven or automated decision-making. Manual scoring is repetitive, time-consuming, and difficult to standardize across various environments. Large-scale screening efforts require significant investment in data management, risk assessment by age group, and consistent identification of high-priority cases. As the number of individuals screened rises, the burden on professionals grows, further straining the system.

Therefore, although conventional methods are essential for final clinical validation, there is an evident need for automated pre-screening tools capable of rapidly processing structured data and supporting faster, more equitable referral decisions.

### Feasibility Study

Feasibility Assessment

During this phase, the viability of the proposed project is thoroughly evaluated. This includes the preparation of a preliminary business proposal and a high-level project plan, complete with initial cost estimates. The feasibility study is designed to confirm that the proposed solution will not place unnecessary strain on the organization. A detailed understanding of the system's core requirements underpins this analysis.

The feasibility assessment focuses on three key areas:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

**Economic Feasibility**

This area considers the financial impact of the proposed system on the organization. With strict research and development budget limitations, every expense must be well-justified. The project plans to remain cost-effective by utilizing open-source or freely accessible technologies, with expenditures limited to essential custom-built components only.

**Technical Feasibility**

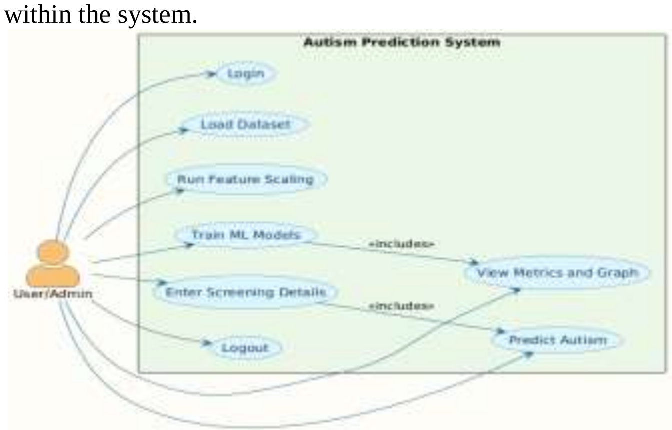
Technical feasibility evaluates whether the organization's current technical resources can support the new system. The design emphasizes minimal technical requirements, allowing for easy implementation with little to no changes required to the client's existing infrastructure.

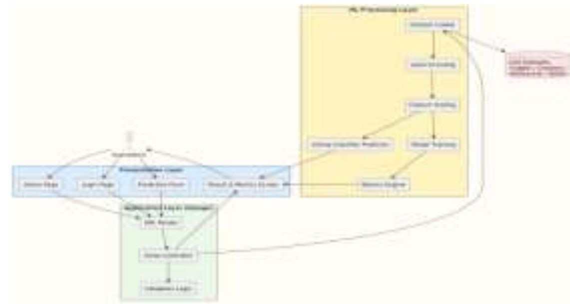
**Social Feasibility**

Social feasibility measures the probability of user acceptance and the successful adoption of the system. This involves providing thorough training to ensure users are confident and capable in operating the new system. The aim is for users to recognize the system as a beneficial tool rather than a disruption. Effective education and familiarization efforts will be made to encourage positive feedback and engagement from key stakeholders.

**SYSTEM ARCHITECTURE:**

A use case diagram in the Unified Modeling Language (UML) is a behavioral diagram that originates from use-case analysis. It visually represents the system's functionality by illustrating actors, their objectives (use cases), and the relationships among those use cases. The primary goal of a use case diagram is to display which system functions are associated with each actor. It also highlights the roles played by actors within the system.



**UML DIAGRAM'S:**

UML, or Unified Modeling Language, is a standardized modeling language widely used in object-oriented software engineering. Developed and managed by the Object Management Group, UML provides a common, general-purpose language for modeling the structure and behavior of software systems. Its primary aim is to facilitate clear communication and shared understanding among developers, designers, and stakeholders in software projects.

Currently, UML consists of two core components: a meta-model that defines the language's underlying structure, and a graphical notation for visually representing system components. Future enhancements may introduce additional methodologies or processes associated with UML.

UML is not limited to software systems—it is also applicable to business modeling and other non-software domains. It supports the specification, visualization, construction, and documentation of system artifacts, and embodies a collection of best practices proven effective in modeling large and complex systems.

By relying primarily on graphical notations, UML helps teams design, communicate, and document their software projects efficiently. It has become a fundamental part of the object-oriented software development process.

**Primary Goals of UML Design:**

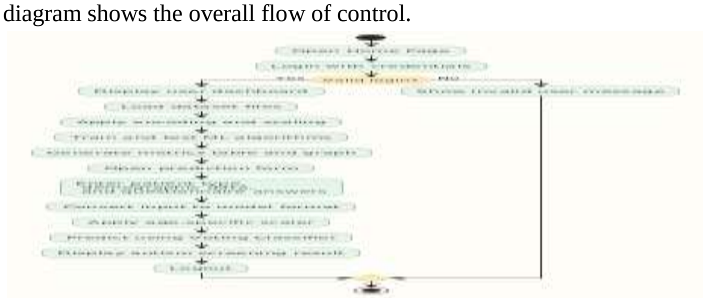
1. Provide a comprehensive and expressive visual modeling language for creating and sharing meaningful models.
2. Enable extension and specialization of core concepts to meet various modeling requirements.

3. Remain independent of specific programming languages and development processes.
4. Establish a formal basis for understanding and applying the modeling language.
5. Promote the advancement and evolution of the object-oriented tools market.

**USE CASE DIAGRAM:**

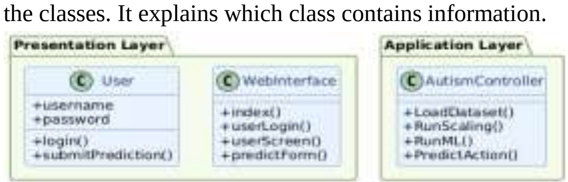
**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.




**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) 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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

**INPUT DESIGN**

**Input Design**

Input design serves as the crucial bridge between an information system and its users, focusing on how data enters the system. This process involves establishing clear procedures for preparing and entering data so that transactional information can be efficiently converted into a format suitable for computer processing. Data input can occur through scanning physical documents or direct user entry via keyboards and other devices.

The main objectives of effective input design are to control the volume of data entered, minimize errors, reduce input delays, eliminate unnecessary steps, and streamline the process. Key considerations also include security, usability, and privacy.

Key Elements of Input Design:

- Identifying essential data to be captured
- Determining data structure and coding methods
- Designing intuitive, user-friendly data entry interfaces
- Implementing validation checks and error-handling procedures

Goals of Input Design:

1. Convert user data entry needs into a computer-based process that ensures accuracy and supports management in accessing relevant information.
2. Develop intuitive and efficient data entry screens capable of managing large data volumes, simplifying input, and minimizing errors. These interfaces should facilitate easy editing and review of records.
3. Integrate real-time data validation and provide clear feedback messages, making data entry straightforward and user-focused. The ultimate objective is to build input layouts that are logical, user-friendly, and efficient.

---

Output Design

Output design is concerned with presenting processed information to users in a clear and meaningful manner. It ensures that outputs—whether displayed on screens or printed as reports—convey system results effectively to users or other systems. Well-designed outputs are essential for informed decision-making and enhancing user satisfaction.

Principles of Output Design:

1. Take a systematic approach to output design, carefully identifying necessary outputs and ensuring each element is clear and usable.

IJERST International Journal of Engineering Research and Science & Technology

2. Select the most suitable format for presenting information, whether as digital displays or printed documents.
3. Create reports and documents that present system-generated information in a clear, concise, and effective manner.

Goals of Output Design:

- Provide information on past events, current status, or future projections
- Highlight significant events, opportunities, issues, or warnings
- Support or validate actions based on system results

A strong output design ensures that information systems communicate effectively with users, fulfill their requirements, and support sound decision-making.

**SCREEN SHOTS**

To run project double click on 'run.bat' file to start python server and then will get below page

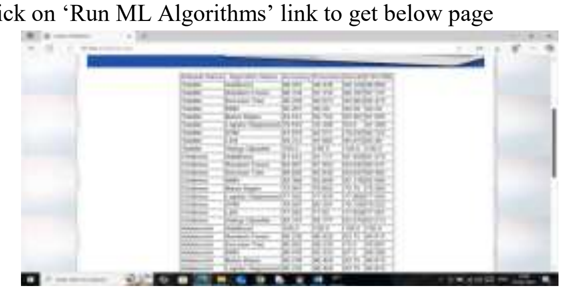
In above screen python server started and now open browser and enter URL as http://127.0.0.1:8000/index,html and then press enter key to get below page


In above screen click on 'User Login' link to get below page

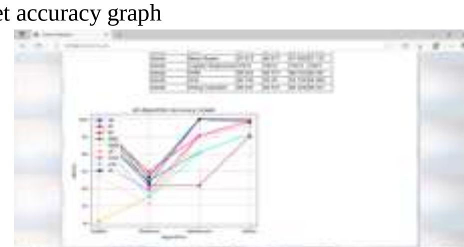

In above screen user is login and after login will get below page


In above screen all 4 datasets loaded and can see number of rows and features available with each dataset and now click on 'Run Features Scaling & Selection' link to process features and then will get below page


In above screen features scaling process completed and now click on 'Run ML Algorithms' link to get below page



In above two screens displaying performance of each algorithm on dataset wise and in all algorithms voting classifier got high accuracy for all dataset and then further scroll down above page to get accuracy graph



In above accuracy comparison graph x-axis represents "dataset names" and y-axis represents accuracy and then each line represents different algorithms and in above graph can see Black line is for Voting Classifier and its top on graph for maximum datasets. Now click on 'Predict Autism' link to get below page


In above screen selecting some input values and then press button to get below page


In above screen for given input ML algorithm predicted 'No Autism Detected' and similarly you can input parameters and get prediction. Below is another sample


In above screen gave some other input values and below is the output


In above screen for given input ML predicted 'Autism detected.

**IMPLEMENTATION:**

This project is centered on a Django-based web application, AutismApp, developed as an academic prototype for early-stage autism spectrum disorder (ASD) screening using machine learning techniques. The application orchestrates incoming requests through Autism/urls.py, while core user interactions and backend logic are managed in AutismApp/views.py. The user interface leverages modular HTML templates and a static CSS file for styling. Machine learning functionality is embedded directly within the application's views, resulting in a streamlined architecture where request handling and predictive analytics are tightly integrated.

**Module Descriptions**

**Login Module**

User authentication is implemented via the UserLogin and UserLoginAction modules. The system presents a straightforward login page, validating credentials against hard-coded values (admin/admin). Upon successful authentication, users are redirected to the main dashboard.

**Dataset Loading Module**

The LoadDataset module loads four separate CSV datasets using Pandas and displays row and column counts for each dataset. This feature allows users to quickly understand the scope and structure of the available data for analysis and model training.

**Preprocessing Module**

Preprocessing logic is primarily handled by the processDataset function. This module identifies categorical features, applies label encoding, replaces missing values with zeros, extracts the target variable (ASD_traits), and constructs the feature matrix for classification. These steps ensure that raw questionnaire data is systematically prepared for machine learning analysis.

**Feature Scaling Module**

Feature scaling is indicated to users via the RunScaling interface option, with the actual scaling steps executed during model training. Normalizer is applied to toddler and children datasets, while QuantileTransformer is used for adolescent and adult datasets. This standardizes input data, improving model performance and prediction reliability.

**Machine Learning Training Module**

The core analytical process is managed by the RunML function, which organizes data by age group and trains multiple machine learning algorithms, including:

- AdaBoost Classifier
- Random Forest Classifier
- Decision Tree Classifier
- K-Nearest Neighbors
- Gaussian Naive Bayes
- Logistic Regression
- Support Vector Machine
- Linear Discriminant Analysis
- Voting Classifier

Each model's performance is evaluated using accuracy, precision, recall, and F1-score. Results are stored in global lists and displayed in a comparative results table for easy reference.

**Metrics and Visualization Module**

The functions calculateMetrics, getMetrics, and getResults collect and organize model performance metrics. Visual comparison is enabled through Matplotlib-generated graphs, which are encoded in Base64 for direct integration into the web interface. This enhances interpretability and facilitates model comparison.

**Prediction Module**

Prediction functionality is handled by the Predict, PredictAction, and createDataFrame modules. Users input relevant features, including age group, gender, age, jaundice status, and responses to ten screening questions. The system selects the appropriate encoder, scaler, and voting classifier for the patient type, and displays the final result as either "Autism Detected" or "No Autism Detected."

**User Interface Module**

The application uses four main HTML templates to provide a simple, menu-driven interface:

- index.html: Home page
- UserLogin.html: Login form
- UserScreen.html: Dashboard and results
- Predict.html: Prediction input form

This intuitive interface ensures the system is accessible for academic demonstration and workflow evaluation.

**Workflow Summary**

1. The user visits the home page.
2. The user logs in through the authentication form.
3. The system loads and displays dataset information.
4. Selected training data is preprocessed and scaled.
5. Multiple machine learning models are trained.
6. The system presents comparative metrics and visualizations.
7. The user inputs a new screening case.
8. The system predicts the autism screening outcome.

**Conclusion**

This project delivers a comprehensive academic prototype for autism screening, integrating structured data analytics and machine learning within a web-based interface. By supporting datasets for toddlers, children, adolescents, and adults, the system demonstrates the flexibility of machine learning models across different developmental stages, unified under one architecture.

The application encompasses every step of the data science workflow: loading, preprocessing, encoding, scaling, model training, evaluation, visualization, and prediction. By comparing multiple classifiers and leveraging an ensemble voting classifier for final screening, the system ensures robust results. Standard performance metrics and visualizations increase transparency and model interpretability.

Importantly, this project shows how questionnaire-based screening data can be transformed into actionable machine learning inputs. While the system does not replace clinical diagnosis, it offers valuable decision support for rapid pre-screening—potentially reducing manual workload and helping identify high-risk cases earlier. This demonstrates the potential of open-source technologies and structured behavioral data in building accessible, cost-effective healthcare tools.

**Future Work**

Several enhancements can further improve the system's technical sophistication and practical value:

- Implement secure, database-backed user management in place of hard-coded credentials.
- Add model persistence to retain trained classifiers between sessions.
- Introduce advanced features such as automated feature selection, hyperparameter tuning, cross-validation, and explainable AI techniques (e.g., SHAP) to improve model trust and transparency.
- Expand the dataset for increased diversity and reduced population bias.
- Design a more interactive, clinically oriented user interface for healthcare professionals.
- Provide multilingual and mobile-friendly support for broader accessibility.
- Integrate with hospital information systems or cloud platforms for scalability.
- Incorporate audio, video, or behavioral observation data for multimodal prediction.

These improvements would help transition the system from an academic prototype to a robust, clinically valuable decision-support tool for autism screening.

Project Implementation Overview

This project presents AutismApp, a Django-based web application designed to facilitate early-stage screening for autism spectrum disorder using machine learning. The application's request routing is managed through Autism/urls.py, while the main user interactions and core logic reside in AutismApp/views.py. The front end is built with HTML templates and styled via a static CSS file. Machine learning operations are embedded directly in the views, resulting in a streamlined prototype that closely integrates request handling and predictive analytics.

**Module Overview**

Login Module

User authentication is managed by the UserLogin and

UserLoginAction modules. The system uses a simple login form, validating credentials against hard-coded values (admin/admin). Upon successful login, users are redirected to the main dashboard.

**Dataset Loading Module**
Implemented by the LoadDataset module, this component loads four CSV datasets using Pandas and displays their row and column counts. This provides users with a quick overview of data scope and structure for subsequent analysis and training.

**Preprocessing Module**
The processDataset function handles data preprocessing. It detects categorical features, applies LabelEncoder transformations, fills missing values with zeros, extracts the ASD_traits target variable, and constructs the feature matrix for classification. These steps convert raw questionnaire data into a machine-readable form.

**Feature Scaling Module**
Feature scaling is initiated via the RunScaling option, with operations executed during model training. A Normalizer is applied to toddler and children datasets, while a QuantileTransformer is used for adolescent and adult datasets, ensuring standardized inputs for machine learning.

**Machine Learning Training Module**
RunML serves as the main analytical engine, preparing age-specific datasets and training multiple models:

**AdaBoost Classifier**

Random Forest Classifier

Decision Tree Classifier

K-Nearest Neighbors

Gaussian Naive Bayes

Logistic Regression

Support Vector Machine

Linear Discriminant Analysis

Voting Classifier

Each model is evaluated using accuracy, precision, recall, and F1-score. Results are stored in global lists and compiled into a comparative results table.

**Metrics and Visualization Module**
The functions calculateMetrics, getMetrics, and getResults aggregate performance metrics and generate comparative graphs with Matplotlib. These graphs are encoded in Base64 for seamless web integration, providing visual insight for model comparison.

**Prediction Module**
Prediction is conducted via the Predict, PredictAction, and createDataFrame components. Users input demographic and screening data, including age group, gender, age, jaundice status, and ten questionnaire responses. The system automatically applies the appropriate encoder, scaler, and voting classifier based on patient type, returning a result of either “Autism Detected” or “No Autism Detected.”

**User Interface Module**
The application employs four main HTML templates:

index.html: Home page

UserLogin.html: Login form

UserScreen.html: Dashboard/results page

Predict.html: Prediction input form

This menu-driven layout supports intuitive navigation and is well-suited for academic demonstration and system evaluation.

Workflow Summary

User navigates to the home page.

User logs in using the authentication form.

The system loads and displays dataset statistics.

Selected data is preprocessed and scaled.

Multiple machine learning models are trained.

Comparative performance metrics and visualizations are generated.

User submits a new screening case.

The system returns an autism screening prediction.

***Conclusion***

Project Overview

This project presents a comprehensive academic prototype designed for early-stage autism screening. By seamlessly integrating structured data analytics and machine learning within an intuitive web interface, the system accommodates data from four key developmental stages—toddlers, children, adolescents, and adults—demonstrating the flexibility of machine learning models within a unified platform.

Key Features

Core platform functionalities include:

- Data loading and preprocessing
- Feature encoding and scaling
- Model training, evaluation, and visualization
- Predictive capabilities

The system allows users to compare multiple classifiers and leverages an ensemble voting classifier to enhance decision robustness. Its user-friendly web interface ensures that the platform is accessible for both educational purposes and evaluative scenarios.

A noteworthy contribution of this work is the conversion of questionnaire data into meaningful machine learning inputs, coupled with the use of standard performance metrics to ensure transparency and interpretability. While the system is not intended to replace clinical diagnosis, it provides a valuable decision-support tool for rapid pre-screening, potentially minimizing manual effort and enabling earlier identification of individuals at risk. This project exemplifies how open-source technologies and structured behavioral data can be harnessed to create scalable, cost-effective healthcare informatics solutions.

Future Directions

To further increase technical sophistication and practical impact, the following enhancements are proposed:

- Implement secure, database-driven user authentication instead of hard-coded credentials
- Enable model persistence to retain trained classifiers between sessions
- Introduce advanced features such as automated feature selection, hyperparameter tuning, cross-validation, and explainable AI techniques (e.g., SHAP) for greater trust and understanding
- Expand datasets to improve diversity and reduce bias
- Develop a more interactive and clinically focused interface for healthcare professionals
- Support multilingual and mobile-friendly access to widen usability
- Integrate with hospital information systems or cloud platforms to ensure scalability
- Incorporate audio, video, or behavioral observation data for multimodal predictions

These improvements will help evolve the project from an academic prototype into a robust, clinically valuable decision-support tool for autism screening.

## References

T. Akter, M. S. Satu, M. I. Khan, M. H. Ali, S. Uddin, P. Lio, J. M. W. Quinn, and M. A. Moni, “Machine learning-based models for early-stage detection of autism spectrum disorders,” *IEEE Access*, vol. 7, pp. 166509-166527, 2019, doi: 10.1109/ACCESS.2019.2952609. Available: https://ieeexplore.ieee.org/document/8895818/

[2] M. Bala, M. H. Ali, M. S. Satu, K. F. Hasan, and M. A. Moni, “Efficient machine learning models for early stage detection of autism spectrum disorder,” *Algorithms*, vol. 15, no. 5, p. 166, 2022, doi: 10.3390/a15050166.Available: https://www.mdpi.com/1999-4893/15/5/166

[3] H. Abbas, F. Garberson, E. Glover, and D. P. Wall, “Machine learning approach for early detection of autism by combining questionnaire and home video screening,” *Journal of the American Medical Informatics Association*, vol. 25, no. 8, pp. 1000-1007, 2018, doi: 10.1093/jamia/ocy039.Available: https://pubmed.ncbi.nlm.nih.gov/29741630/

[4] D. P. Wall, J. Kosmicki, T. F. DeLuca, E. Harstad, and V. A. Fusaro, “Use of machine learning to shorten observation-based screening and diagnosis of autism,” *Translational Psychiatry*, vol. 2, p. e100, 2012, doi: 10.1038/tp.2012.10. Available: https://www.nature.com/articles/tp201210