Machine Learning-Based Early Screening System for Autism Spectrum Disorder in Toddlers

A **Support Vector Machine(SVM)** Approach for Automated Autism Spectrum Disorder Detection.

Team Members

Lavanya A	**A2212651066 (Team Lead)**
Shreya N	**A22126510110**
Meghana S	**A22126510201**
Afeefa	**A22126510065**
Parimala T	**A22126510202**

Guided by: Dr.G.Srinivas

(Dean of CSA)

The Need for Earlier Detection

What is ASD?

AutismSpectrum Disorder affects social interaction and communication development.

Why Detect Early?

Early detection significantly improves intervention and therapy outcomes.

Traditional Diagnosis

Relies heavily on expert evaluation and subjective behavioural observation.

ML Automation

Machine Learning offers objective, data-driven detection via behavioural patterns.

Problem Statement: Addressing Subjectivity

Late & Subjective Diagnosis

Autism is often diagnosed later than optimal due to variability in observation.

Need for a Data-Driven Tool

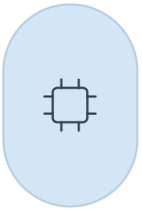
Requires a reliable, automated system to support and accelerate early detection.

Prediction Goals

Predict autism traits accurately using structured questionnaire responses.

Our core goal: delivering an easy-to-use web interface for rapid, data-driven predictions.

Key Objectives of the Project



Intelligent System Development

Create a robust system for predicting ASD traits in toddlers.

SVM Classification

Implement Support Vector Machine (SVM) as the core classification algorithm.

Efficient Data Handling

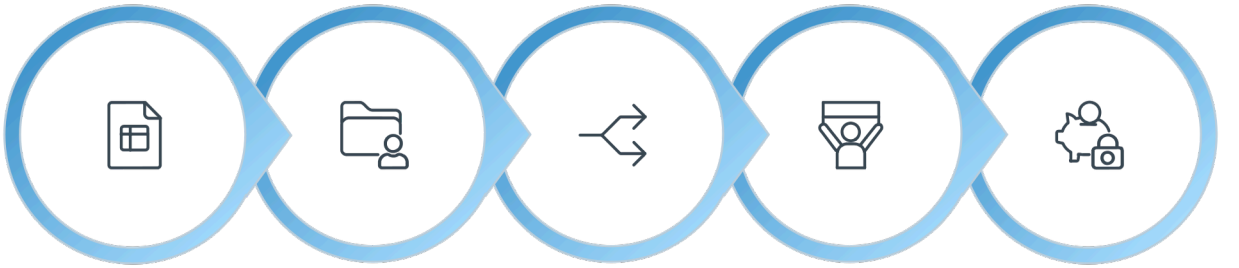
Execute efficient data preprocessing and strategic encoding methods.

User Interface (Gradio)

Build an accessible,web-based UI using Gradio for real-time predictions.

> **Focus:** Achieving high accuracy, full automation, and seamless user interaction.

System Architecture: End-to-End Workflow



The system transforms raw questionnaire data into actionable predictions via a trained SVM model and an interactive web front-end.

System Components: Layered Approach

Dataset Layer

Structured Autism dataset in CSV format.

Preprocessing Layer

Handles feature scaling and categorical data encoding.

Model Layer

The core Support Vector Machine (SVM) classifier.

Interface Layer

User-facing prediction application built with Gradio.

Persistence Layer

Utilises Joblib for saving the trained SVM model and encoders.

Each component is designed for reliability and reusability within the screening system.

Modular Division for Implementation

1. Data Collection

Loading the raw dataset and removing irrelevant columns (e.g., Case_No).

2. Preprocessing

Encoding text features and applying Standard Scaling to numeric data.

3. Model Training

Training the SVM classifier using the prepared data via Scikit-learn.

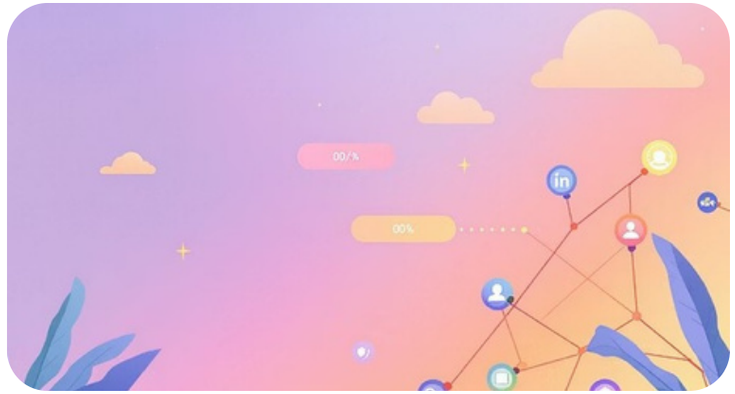
4. Prediction

Generating predictions and associated confidence scores.

5. User Interface

Gradio UI to collect inputs and display final prediction results clearly.

Detailed Data Preprocessing Steps



Column Management

Dropped unnecessarycolumns(e.g., `Case_No`).

Categorical Encoding

Used **LabelEncoder** to convert text data into numerical formats suitable for SVM.

Feature Scaling

StandardScaler applied to ensure uniform feature contribution.

Data Partition

Split the dataset:80% for trainingand 20% for testing.

Algorithm Used: Support Vector Machine (SVM)

Key Characteristics

- Supervised classification algorithm.
- Effective at finding the **optimal boundary (hyperplane)** between classes.
- Highly suitable for complex and moderately sized binary classification datasets.

Hyperparameters

KernelType:	**RBF** (Radial Basis Function)
Regularisation Parameter (C):	1.0
Gamma:	Scale

SVM is ideal for the binary task: **Autism vs. No Autism**.

Project Methodology and Workflow Summary

1. Load Dataset

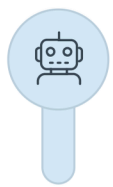
Acquire and initiate dataset.

2. Preprocess & Encode

Clean data, handle categorical variables, and scale features.

3. Split Data

Partition into Training (80%) and Testing (20%) sets.



4. Train SVM Model

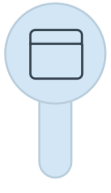
Fit the SVM classifier to the training data.

5. Evaluate Performance

Measure accuracy and analyse the confusion matrix.

6. Save Model

Use Joblib for model persistence.



7. Build Gradio UI

Finalise the user-friendly prediction interface.

Full Traceability: From raw data input to a usable, evaluated prediction tool.

Technology Stack: Leveraging Open-Source Tools

Our solution relies exclusively on robust, open-source tools ideal for rapid prototyping and effective implementation.

Python

The primary programming language for data manipulation, modelling, and back-end logic.

Scikit-learn

The essential library for implementing the Support Vector Machine (SVM) classifier.

Gradio

Used to build the intuitive and easily shareable web-based user interface.

Joblib

Employed for efficiently saving and reloading the trained model and associated components.

Data Management & Analysis Libraries

A suite of specialised libraries were used for data handling, cleaning, and exploratory data analysis (EDA).

- **Pandas:** Crucial for data loading, structuring, and cleaning the raw dataset.
- **NumPy:** Provided foundational support for high-performance numerical operations and array processing.
- **Seaborn & Matplotlib:** Used extensively for generating visualisations during the data exploration and model evaluation phases.
- **Jupyter Notebook:** Served as the primary development environment for interactive coding and testing.

User Interaction: Input and Expected Output

The Gradio interface is designed to simplify the input process ,requiring only standard observations on child behaviour.

System Input

The user responds to a series of diagnostic questions concerning the child's behaviour.

- Responses are either Yes/No binary choices or scored numerical inputs.
- *Example Input:* "Does the child respond to their name when called?"
- *Example Input:* "Does the child smile back during interactions?"

System Output

The model processes the input and provides a clear classification result.

- Primary prediction: "Autistic Traits" or "No Autism"
- Confidence Score: A percentage indicating the models certainty (e.g., **92%** confidence).

> The UI allows instant predictions, making the screening process simple and accessible for users.

Future Enhancements: Expanding Capabilities

To improve both accuracy and system accessibility, we have defined several key areas for future development.

Database Integration

Implement persistent storage for user results and screening history, enabling longitudinal studies and tracking.

Deep Learning Models

Explore sophisticated deep learning architectures to potentially achieve even higher levels of predictive accuracy.

Cross-Platform Deployment

Develop dedicated mobile and desktop versions of the application to maximise accessibility for clinicians.

Analytics & Visualisation

Introduce interactive dashboards to display results trends, model performance, and detailed analysis.

Strategic Applications: Impacting Early Identification

This machine learning model serves as a valuable tool with wide-ranging potential in clinical and research settings.

Early Screening

Provides an initial, rapid screening tool for toddlers, facilitating earlier intervention efforts.

Clinical Support

Assists paediatricians and psychologists in their diagnostic processes, acting as an objective second opinion.

Research Utility

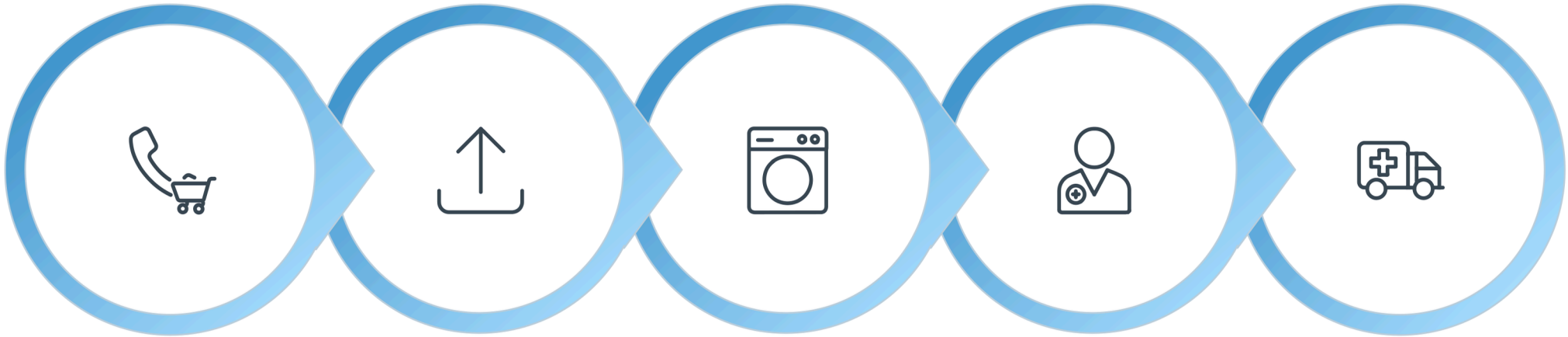
A useful resource for child development centres and researchers studying Autism Spectrum Disorder (ASD) identification.

Reducing Error

Minimises the risk of human subjectivity and error in the crucial early stages of ASD identification.

Core Function: Assisting Healthcare Professionals

The system is designed to seamlessly integrate into the clinical workflow ,offering objective data to support complex diagnostic decisions.



Child Screened | **Data Sent** | **ML Prediction** | **Clinician Review** | **Final Decision**

> Our system is intended to assist, not replace, healthcare professionals, providing quantitative support for qualitative assessments.

References: Foundational Resources

Our project's methodology, data, and implementation were guided by established research and documentation.

Data Source

Toddler Autism Dataset, UCI ML Repository (2018), providing the core training data.

Key Research

IEEE Paper: *Early Detection of Autism Using Machine Learning Techniques* (2020), defining the application context.

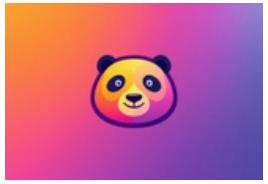
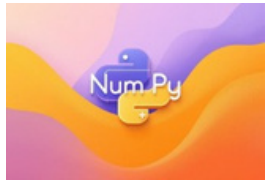
Modelling Documentation

Scikit-learn Documentation (**https://scikit-learn.org**), for effective model deployment.

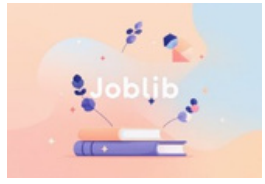
Interface Documentation

Gradio Documentation (**https://www.gradio.app**), for UI development and hosting.

Summary of Key Libraries





- **Data Handling:** Pandas and NumPy were essential for efficient data manipulation.
- **Visualisation:** Matplotlib and Seaborn enabled comprehensive data exploration and result presentation.
- **Machine Learning:** Scikit-learn provided the reliable framework for our SVM classifier.
- **Model Persistence:** Joblib ensured the easy and correct saving of the entire pipeline.

THANK YOU