**AUTISM SPECTRUM DISORDER DETECTION USING MULTIMODAL DATA AND VR THERAPY**

Kabileshwaran SD, Praveen V, Sumathi K, Harish M

Guide: Dr. P. Varalakshmi

**AIM:**

The aim of this project is to develop an efficient and interactive system for the early detection and intervention of Autism Spectrum Disorder (ASD) in children. This system will leverage both machine learning and deep learning to accurately identify signs of ASD using multimodal data, including EEG signals, images, and Behavioural-based questions and Eye Tracking data.

**INTRODUCTION:**

Autism spectrum disorder is a condition related to brain development that impacts how a person perceives and socializes with others, causing problems in social interaction and communication. Globally, approximately 1 in 100 children are diagnosed with Autism Spectrum Disorder (ASD), with boys being 4 to 5 times more likely than girls to receive a diagnosis. The United States has the highest prevalence, with nearly 1 in 36 children diagnosed with ASD, reflecting advanced awareness and diagnostic capabilities. In contrast, India’s estimated prevalence is around 1 in 68 children, with many cases potentially going undiagnosed due to limited awareness and social stigma. These statistics underscore the importance of early diagnosis, increased awareness, and support systems to improve the quality of life for individuals with autism across the world.

Our project aims to enhance ASD detection and therapy using machine learning and deep learning to analyze multimodal data, including EEG signals, images, Behavioural-based questions and Eye Tracking data.

**PROPOSED SOLUTION:**

Our solution involves a multi-faceted approach to enhance the detection and management of Autism Spectrum Disorder (ASD).

1. **Advanced Detection**: We will use machine learning and deep learning models to analyze multimodal data, including EEG signals, facial images, and IQ-based responses, to accurately diagnose ASD.
2. **Continuous Improvement**: Implement incremental learning to refine models with new data, ensuring accuracy and adaptability over time.

This integrated approach aims to provide a more effective, personalized, and accessible solution for ASD diagnosis and therapy.

***ASD AQ10 Dataset***

**1. Data Preprocessing**

*a. Replace Strings with Integers:*

Replaced categorical string values with integers for binary variables: 'YES' was replaced with 1, and 'NO' with 0. Uncertain or other values, like '?' and 'others', were standardized to 'Others'

*b. Correlation Matrix and Feature Selection:*

A correlation matrix helps identify how strongly each feature is correlated with the target variable and other features. Based on this matrix, selected the most relevant features for r model to prevent multicollinearity and improve performance.

*c. One-Hot Encoding:*

Applied one-hot encoding for categorical features that had more than two categories. This converts categorical variables into binary columns.

*d. Remove Outliers in Age Using IQR:*

Outliers in the age column were removed using the Interquartile Range (IQR) method. The IQR is the range between the first and third quartiles, and values outside the range [Q1 - 1.5 \* IQR, Q3 + 1.5 \* IQR] are considered outliers and removed.

*e. Replace NaN Values:*

Missing values in the age column were filled with 0, indicating unknown or missing ages.

*f. Drop Unwanted Columns:*

dropped columns that were unnecessary for model training, such as 'contry\_of\_res', 'used\_app\_before', and 'result'.

*g. Train-Test Split:*

After preprocessing, the data was split into training and testing sets using train\_test\_split.

**2. Model Building and Hyperparameter Tuning Using GridSearchCV**

a. Logistic Regression

b. Random Forest

c. Support Vector Machine (SVM)

d. K-Nearest Neighbors (KNN)

e. Artificial Neural Network (ANN)

f. Decision Tree (DT)

g. Extra Trees Classifier

**3. GridSearchCV:**

Purpose: It automates hyperparameter tuning by searching over a grid of hyperparameters and selecting the best combination based on cross-validation performance.

*How GridSearchCV Works:*

For each combination of hyperparameters, it trains the model multiple times (using cross-validation) and evaluates it based on a specified metric (e.g., accuracy or F1-score). After trying all combinations, it returns the best hyperparameters.

**4. Model Prediction and Evaluation:**

For each model, after fitting, predicted on the test set and evaluated the performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrices.

**Eye Tracking Dataset**

***1.Data Preprocessing***

1. *Removed Duplicates*: Ensured dataset uniqueness to maintain analysis integrity and avoid skewed results.
2. *Handled Null Values:* Addressed missing values through imputation or removal to improve model quality.
3. *Categorized Features:* Differentiated between discrete, categorical, and continuous variables for tailored analysis and visualization.
4. *Analyzed and Visualized Distributions:* Used visualization techniques to understand variable distributions, identify patterns, and detect anomalies.
5. *Label Encoding:* Converted categorical values into numerical format for compatibility with machine learning algorithms.
6. *Removed Outliers:* Identified and handled outliers to enhance model accuracy and robustness.
7. *Box Plot to Check Outliers:* Utilized box plots to visualize data spread and detect outliers.
8. *Correlation Matrix:* Analyzed relationships between continuous variables to inform feature selection and engineering.

**2. Model Building and Hyperparameter Tuning Using GridSearchCV**

a. Random Forest

b. LSTM

c. K-Nearest Neighbors (KNN)

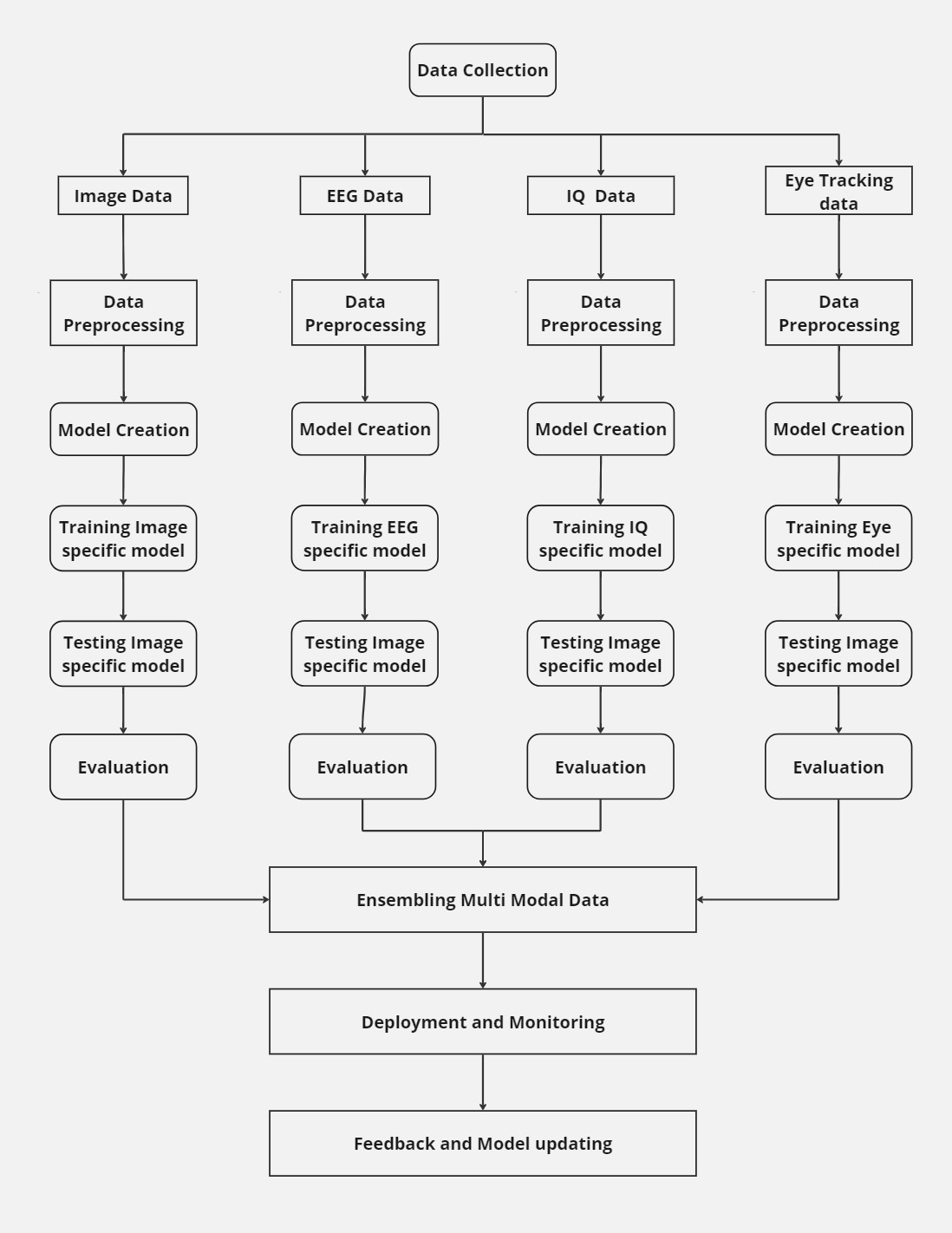
d. Artificial Neural Network (ANN)

e. Decision Tree (DT)

**3. Model Prediction and Evaluation:**

For each model, after fitting, predicted on the test set and evaluated the performance using metrics such as accuracy, precision, recall, F1-score, and confusion matrices.

**Architecture Diagram**

****

**TOOLS AND TECHNOLOGIES USED:**

* Python for data preprocessing, analysis, and model creation/generation (using Pandas, Numpy, Matplotlib, scikit-learn, and TensorFlow libraries)
* Google Colab for training the model
* Flask to run and manage backend services
* Unity/Vuforia for AR and VR technology
* Git/GitHub for version control
* HMT Device for implementing AR and VR experiences