AUTISM PREDICTION IN ADULTS

A Research Report

Submitted by

STUDENT

Guggilam Leela Naga Sai Sri Saketh – AP23110010510

Under the Supervision of

Mr. B. L. V. SIVA RAMA KRISHNA

Assistant Professor

Department of Computer Science and Engineering

SRM University-AP

In partial fulfilment for the requirements of the Research

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY-AP

NEERUKONDA

MANAGALAGIRI - 522503

ANDHRA PRADESH, INDIA

July 2025

Mr. Boddu L V Siva Rama Krishna
Assistant Professor
Computer Science and Engineering
School of Engineering and Sciences (SEAS)
Contact: +91 9885050551
sivaramakrishna.b@srmap.edu.in

SRM UNIVERSITY AP
Andhra Pradesh

Date:13-08-2025

To Whomsoever It May Concern

Sub: Summer Research Internship – Joining Letter

This is to certify that **Mr. GUGGILAM LEELA NAGA SAI SRI SAKETH(AP23110010510)** a student of Computer Science and Engineering at SRM University-AP, has been selected to participate in the **Summer Research Internship Program** under my guidance in the **Department of Computer Science and Engineering**, SRM University – AP.

The internship will be conducted from **June 2, 2025 to July 25, 2025**. During this period, the intern will be engaged in research activities related to **Autism Prediction in Adults using Machine Learning Techniques**, involving literature review, experimentation, data analysis, and preparation of reports.

We welcome **Mr. GUGGILAM LEELA NAGA SAI SRI SAKETH(AP23110010510)** to this program and look forward to their valuable contribution.

Faculty Mentor

Mr. Boddu L. V. Siva Rama Krishna
Assistant Professor
Department of Computer Science and Engineering
SRM University – AP

Acknowledgement

I would like to say thank you to everyone who helped and supported me while I worked on this project.

First, I want to thank **Prof. Manoj K Arora**, the Vice Chancellor of SRM University–AP, for creating a learning environment that made this project possible.

I sincerely thank my faculty mentor, **Mr. B. L. V. Siva Rama Krishna**, for his unwavering support and insightful guidance throughout the duration of this project. His suggestions greatly contributed to my progress and refinement of the work.

This project was successful because of the support from all these people, and I truly appreciate it.

Abstract

This study presents the development of a machine learning framework for predicting Autism Spectrum Disorder (ASD) in adult populations at early stages. The research utilized a comprehensive dataset from the UCI Machine Learning Repository, which included participant demographic data and responses from validated screening instruments. The investigation compared the performance of several classification algorithms, encompassing Logistic Regression, Decision Tree analysis, Support Vector Machines, Random Forest ensemble methods, and Gaussian NB classifiers, to identify the most effective approach for ASD prediction in adults.

The data underwent preprocessing including cleaning, label encoding, and missing value handling. Training and testing subsets were created from the data through application of the train-test-split function. Four-fifths of the data supported model development, with one-fifth retained for testing purposes. Key performance indicators such as overall correctness rate, true positive identification rate, sensitivity, their harmonic mean, confusion matrix, and AUC-ROC were employed for comprehensive evaluation. The Random Forest Classifier demonstrated superior performance in prediction accuracy and generalization capabilities, making it the most suitable model for ASD classification in adults.

This study contributes to automated screening tools that could assist healthcare professionals in early ASD identification, potentially improving diagnostic efficiency and patient outcomes.

This work was conducted by GUGGILAM LEELA NAGA SAI SRI SAKETH under the supervision of B.L.V. Siva Rama Krishna as part of a machine learning research project.

TABLE OF CONTENTS

A brief introduction of the organization's business sector

Overview of the Higher Education Sector in India:

India's higher education sector is among the largest in the world and plays a key role in building a skilled workforce. It covers a wide range of fields such as engineering, medicine, science, management, law, arts, and new areas like data science and artificial intelligence.

The system is made up of universities, institutes, and colleges run by both government and private bodies. With millions of students enrolling every year, India stands second globally in terms of higher education enrollment. To ensure quality, organizations like the University Grants Commission (UGC) and the All-India Council for Technical Education (AICTE) regulate and guide institutions.

In recent years, the National Education Policy (NEP) 2020 has aimed to make education more flexible, multidisciplinary, and focused on research and skills. Digital learning and online platforms have also grown, making education more accessible.

Higher education in India is expanding quickly and is central to preparing young people for careers, supporting innovation, and helping the country grow as a knowledge-driven economy.

Overview of the organization

SRM University Andhra Pradesh:

1. Brief History

SRM University Andhra Pradesh was officially established in 2017 by the SRM Trust. Although based in Amaravati, its roots trace back to the Nightingale School founded in 1969. Over time, SRM expanded into multiple campuses, leading to the creation of SRM-AP as a new-age, globally connected institution.

2. Business Size

While SRM-AP is not a commercial entity with stocks or commodities, its scale can be understood in academic terms:

Campus size: approximately 100 acres in Amaravati.

Academic scope: Offers undergraduate, postgraduate, and doctoral programs across multiple disciplines.

Research output: Over 3,300 total publications, including 1,268 in elite Q1 journals, 526 patents filed and 60 granted.

3. Product Lines (Services Offered)

SRM-AP provides a wide range of academic and professional offerings:

Undergraduate, Postgraduate, and Ph.D. programs in:

Engineering & Applied Sciences

Liberal Arts & Basic Sciences

Business & Management (now Paari School of Business, offering BBA (Honours), MBA—including specialized MBAs in Banking & Finance, Marketing, HR—and EMBA)

Interdisciplinary Minors and Specializations, integrated via the IDEAL (Inter-Disciplinary Experiential Active Learning) model.

Advanced Research Facilities: across domains like nanotechnology, energy storage, geospatial tech, consumer research, drone tech, etc.

Executive Education through DEEPS (Directorate of Executive Education and Professional Studies), offering skill-development programs for professionals and corporate clients.

Career Development Support: CDC (Career Development Centre) focuses on employability via training in quantitative aptitude, coding, communication, mock interviews, and domain-specific skills.

4. Competitors

SRM University–AP operates in a competitive higher education sector where institutions strive to excel in academics, research, global exposure, and student outcomes. The university positions itself through its interdisciplinary approach, strong research culture, and industry collaborations.

5. Brief summary of all departments

SRM-AP comprises three main academic schools, each with multiple departments:

School of Engineering and Sciences (SEAS): Offers degrees like BTech, MTech, BSc, MSc, and PhD. Departments include Civil Engineering, CSE, Electronics & Communication (with Microelectronics), Electrical & Electronics Engineering, Mechanical, Biological Sciences, Chemistry, Environmental Science & Engineering, Mathematics, and Physics.

Easwari School of Liberal Arts: Offers BA (Honours) and BSc (Honours) programs in disciplines like Economics, Literature & Languages, History, Psychology, Sociology & Anthropology, Political Science, and Media Studies. It supports a flexible curriculum with choices of minors across disciplines.

Paari School of Business (formerly SEAMS): Includes BBA (Honours), MBA (specialized programs), EMBA, and PhD programs. The business curriculum emphasizes practical application, AI/machine learning, entrepreneurial skills, and experiential learning through Action Learning Programmes (ALPs).

Plan of internship program

- The research was carried out under the Department of Computer Science and Engineering (CSE).
- Start and end dates of internship are June 2 2025 and July 25 2025
- Duties and Responsibilities Performed:
  - Conducted background research on Autism Spectrum Disorder and reviewed related machine learning studies to understand its significance.
  - Defined the research problem as a binary classification task (ASD vs Non-ASD).
  - Acquired and explored the UCI ASD in adults dataset, studied feature descriptions, and examined missing values, data types, and class distribution.
  - Preprocessed the dataset by handling missing values, encoding categorical variables, normalizing numerical features, and encoding the target variable.
  - Performed Exploratory Data Analysis (EDA) with visualizations (histograms, box plots, heatmaps) to identify feature patterns, correlations, and class imbalance.
  - Implemented baseline machine learning models including Logistic Regression, Decision Tree, Random forest Classifier, Support Vector Machine, Gaussian Naive Bayes and compared their performance using accuracy, precision, recall, and F1-score.
  - Optimized models through hyperparameter tuning and K-fold cross-validation, and evaluated using ROC, AUC, and confusion matrices.
  - Selected Random Forest as the best-performing model after optimization.
  - Interpreted the final model by analyzing feature importance and identifying key predictors of ASD.
  - Saved the final trained model in .pkl format for deployment.
  - Prepared a research report and presentation slides summarizing the entire workflow, results, and insights.

Introduction

Background:

Autism Spectrum Disorder affects how individuals communicate, interact socially, and behave, often leading to distinctive patterns in development and response. It is a “spectrum” because its manifestation varies widely among individuals. While autism is often diagnosed in childhood, many adults remain undiagnosed due to subtle symptoms or lack of awareness. In adults, late or missed diagnosis can result in prolonged challenges in personal, professional, and social domains.

Traditional diagnostic methods depend on clinical interviews and behavioural assessments, which are often time-consuming, costly, and limited by the availability of trained specialists. These methods also rely heavily on subjective interpretation, leading to inconsistencies in diagnosis and delays in intervention.

In recent years, researchers have explored the use of machine learning for autism detection, particularly in children, by analysing behavioural and questionnaire-based data. However, limited studies focus on adult autism prediction, where early intervention remains critical. This study aims to address that gap by building and evaluating machine learning models specifically for adult screening.

Importance of Early Diagnosis

Early identification of ASD in adults is crucial for promoting self-understanding, reducing anxiety, and enabling timely access to therapeutic interventions and workplace accommodations. It improves quality of life and supports independent living, while also reducing long-term healthcare and societal costs through efficient resource allocation.

Challenges in Current diagnosis Methods:

In current diagnostic approaches in autism prediction face many limitations

- Lengthy assessment processes that can take months to complete
- Limited availability of qualified diagnosticians
- Subjective interpretation of behavioural indicators
- Inconsistent screening protocols across different healthcare settings
- Barriers to accessing specialized diagnostic services

Why Machine Learning Can help:

In recent years, machine learning (ML) has shown promise in healthcare for developing predictive models using structured data. By training ML models on standardized autism screening questionnaires (e.g., A1 to A10) and demographic features (age, gender, jaundice, autism, ethnicity etc.), we can build efficient tools to screen individuals at risk of ASD. These tools can act as a supportive aid to clinicians by flagging high-risk cases for further evaluation.

Proposed Work and Objectives

This research proposes a machine learning-based approach to predict autism in adults using UCI "Autism Screening for Adults" dataset. The goal is to create a most accurate model to help with ASD Screening and early detection in adult populations. This study explores and compares different machine learning algorithms to identify the most reliable model for better accuracy and predictive performance.

Dataset & Methods:

Dataset:

- Name: Autism Screening on Adults
- Source: UCI Machine Learning repository

Dataset Overview:

- Total number of rows: 704
- Total number of columns: 21
- Target variable: Class/ASD – weather the person has ASD or not

Column Description:

Column Name	Description	Type
A1Scoreto A10Score	These are the 1 to 10 ASD screening test scores.	Object
age	Patient age in years	Floating
gender	The patient gender.	Object
ethnicity	cultural or ancestral background of the individual.	Object
jaundice	At the time of birth weather, the patient has jaundice or not.	Object
autism	Does the participant have an immediate family member with an autism diagnosis?	Object
country_of_res	The patient belongs to which country of residence.	Object
used_app_before	Does the patient go to any screening test before or not.	Object
result	It is the sum of all the AQ scores from 1 to 10.	Floating
age_desc	The patient belongs to which age category	Object

Column Name	Description	Type
relation	Relation of the patient who finished the test.	Object
Class/ASD	Weather the patient has ASD or not	Object

Data Preprocessing:

The dataset obtained for this research contained several inconsistencies, missing values, and formatting issues that required comprehensive preprocessing before applying machine learning algorithms. The preprocessing steps undertaken are detailed below:

1. Handling Missing Values and Inconsistencies:

Missing entries denoted by the character "?" were uniformly replaced with NaN values using NumPy to facilitate consistent processing.

```
adult = adult.replace('?', np.nan)
```

- age: Missing values in the age column were imputed using the median of available values. As the median returned a float, it was converted to an integer to reflect realistic age representation.

```
median_age = adult['age'].median()

adult['age'] = adult['age'].fillna(median_age)

adult['age']=adult['age'].astype(int)
```

- ethnicity:

- Missing values were replaced with the label "others".

```
adult['ethnicity'] = adult['ethnicity'].replace({None:"others","Others":"others"})
```

- Inconsistent labels such as "Others" and "others" were standardized to lowercase.

  adult['ethnicity'] = adult['ethnicity'].str.strip().str.lower()

- All the entries are converted to lowercase and there is a in between entries and those spaces were replaced with '_'

  adult['ethnicity'] = adult['ethnicity'].str.replace(" ","_")

  - relation:

- Similar to ethnicity, missing values were replaced with "others".

  adult['relation'] = adult['relation'].replace({None:"Others"})

- All the entries are converted to lowercase and there is a in between entries and those spaces were replaced with '_'.

  adult['relation'] = adult['relation'].str.strip().str.lower()

  adult['relation'] = adult['relation'].str.replace(" ","_")

2. Correction of Data Formatting:

   Spelling Corrections: Column names containing typographical errors were corrected to maintain consistency.

```
adult.rename(columns={'jundice': 'jaundice'}, inplace=True)
adult.rename(columns={'austim': 'autism'}, inplace=True)
adult.rename(columns={'contry_of_res': 'country_of_res'}, inplace=True)
```

   country_of_residence: Values were converted to lowercase and formatted with underscores to standardize input for label encoding.

3. Data Type Conversion:

Screening Test Scores: Originally stored as strings, they were converted to integer types to enable mathematical operations and modelling.

```
adult['A1_Score']=adult['A1_Score'].astype(int)
adult['A2_Score']=adult['A2_Score'].astype(int)
adult['A3_Score']=adult['A3_Score'].astype(int)
adult['A4_Score']=adult['A4_Score'].astype(int)
adult['A5_Score']=adult['A5_Score'].astype(int)
adult['A6_Score']=adult['A6_Score'].astype(int)
adult['A7_Score']=adult['A7_Score'].astype(int)
adult['A8_Score']=adult['A8_Score'].astype(int)
adult['A9_Score']=adult['A9_Score'].astype(int)
adult['A10_Score']=adult['A10_Score'].astype(int)
```

result: This column was in float format and was cast to integer for consistency.

adult['result']=adult['result'].astype(int)

4. Feature Elimination:

The age_desc column contained only a single unique value across all entries, providing no variance or useful information for prediction. Therefore, it was removed from the dataset.

adult.drop('age_desc',axis=1,inplace=True)

The result column was observed to be highly correlated with the target variable, posing a risk of data leakage. To preserve model validity, this column was excluded from model training.

5. Label Encoding:

Categorical variables with object data types underwent label encoding transformation to generate numerical representations compatible with machine learning algorithms.

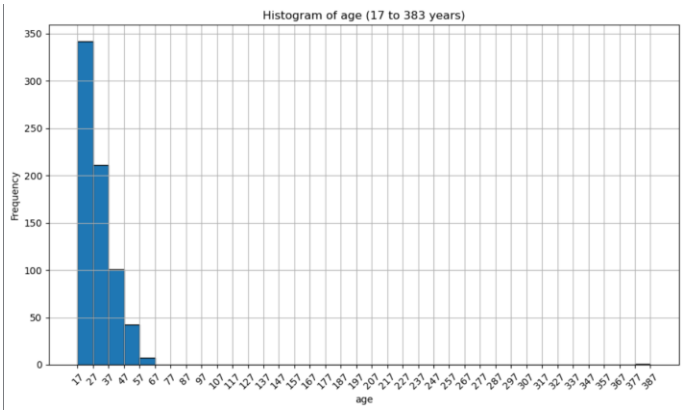
```
from sklearn.preprocessing import LabelEncoder
encoders = {}
# Apply label encoding and store encoders
for col in cols_to_encode:
    le = LabelEncoder()
    adult1[col] = le.fit_transform(adult1[col])
    encoders[col] = le

# Print mappings: string -> number
for col in cols_to_encode:
    print(f"\nLabel Encoding Mapping for '{col}':")
    mapping = dict(zip(encoders[col].classes_, encoders[col].transform(encoders[col].classes_)))
    for key, value in mapping.items():
        print(f"{key} --> {value}")
```

6. Exploratory data analysis:

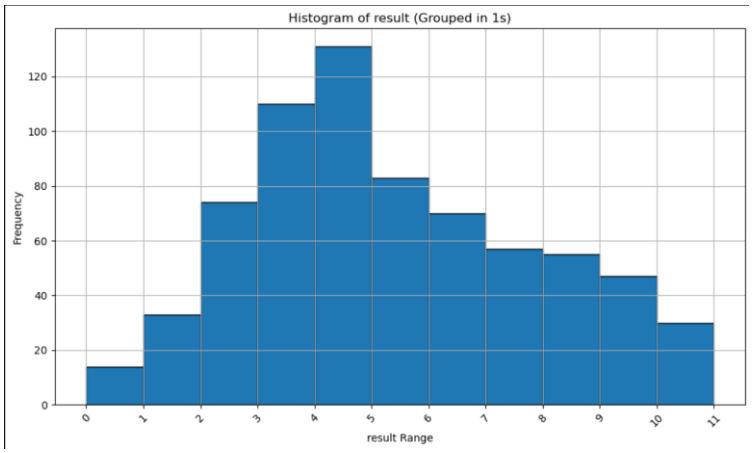
➢ Data Distribution & Patterns:

Age:



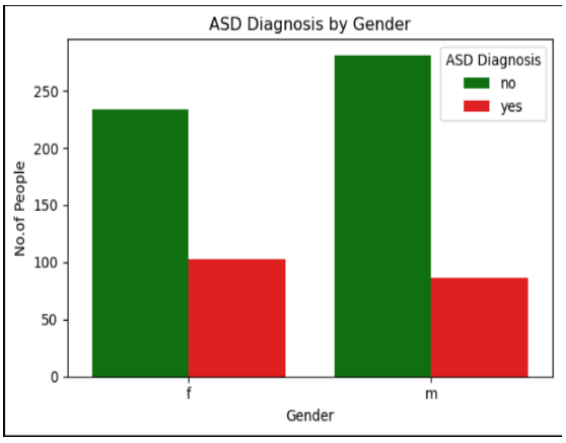
Histograms and boxplots show a right-skewed distribution with some high-end outliers. Majority of participants are aged 20–40.

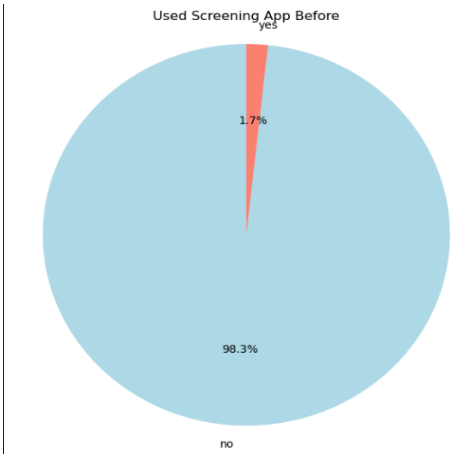
Result Scores:



- Histogram (bin size 1) shows participant score distribution.
- Boxplot highlights mild outliers and score spread.

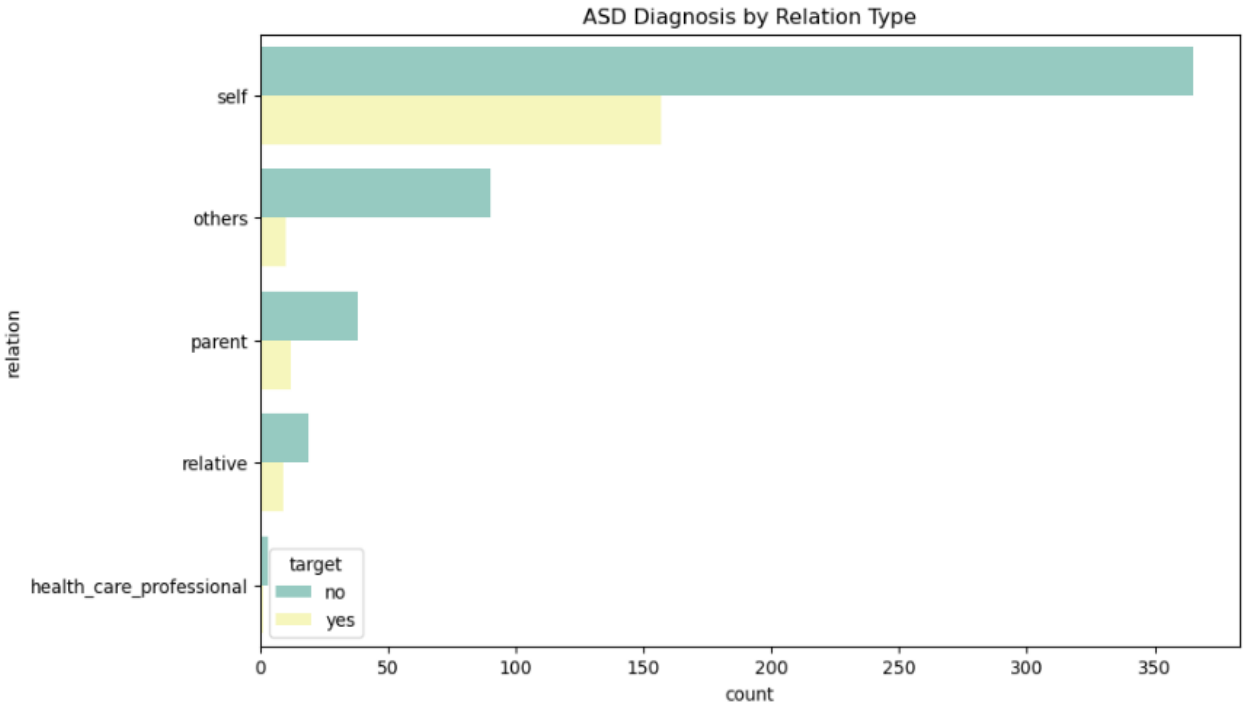
Categorical Features:

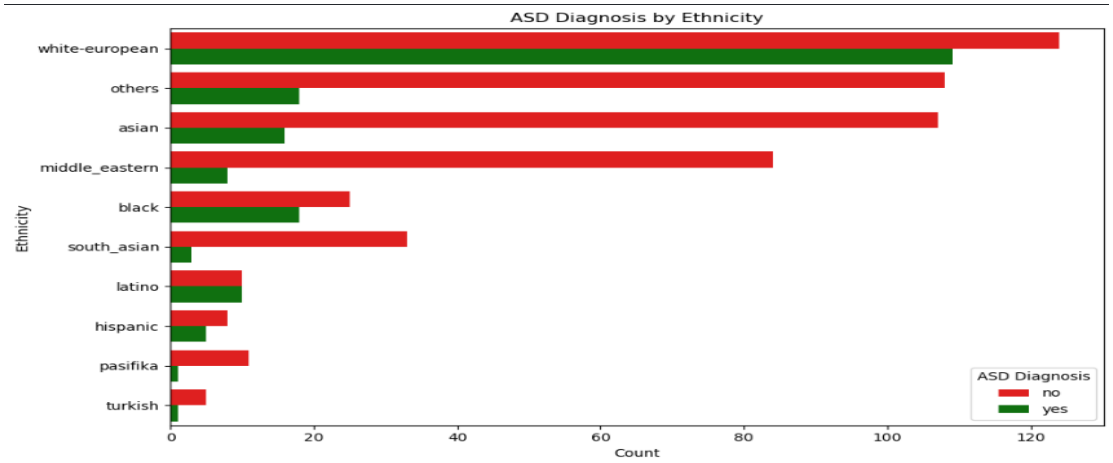




Bar graphs and pie charts were used for Gender and Used App Before and Most users hadn't used the app before (pie chart) and Gender participation was fairly balanced.

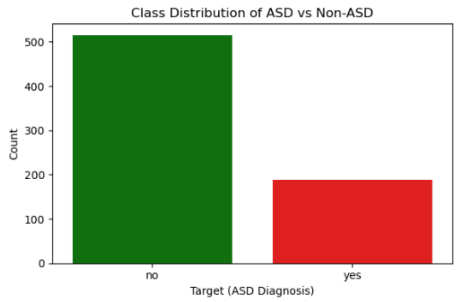
Ethnicity and Relation:





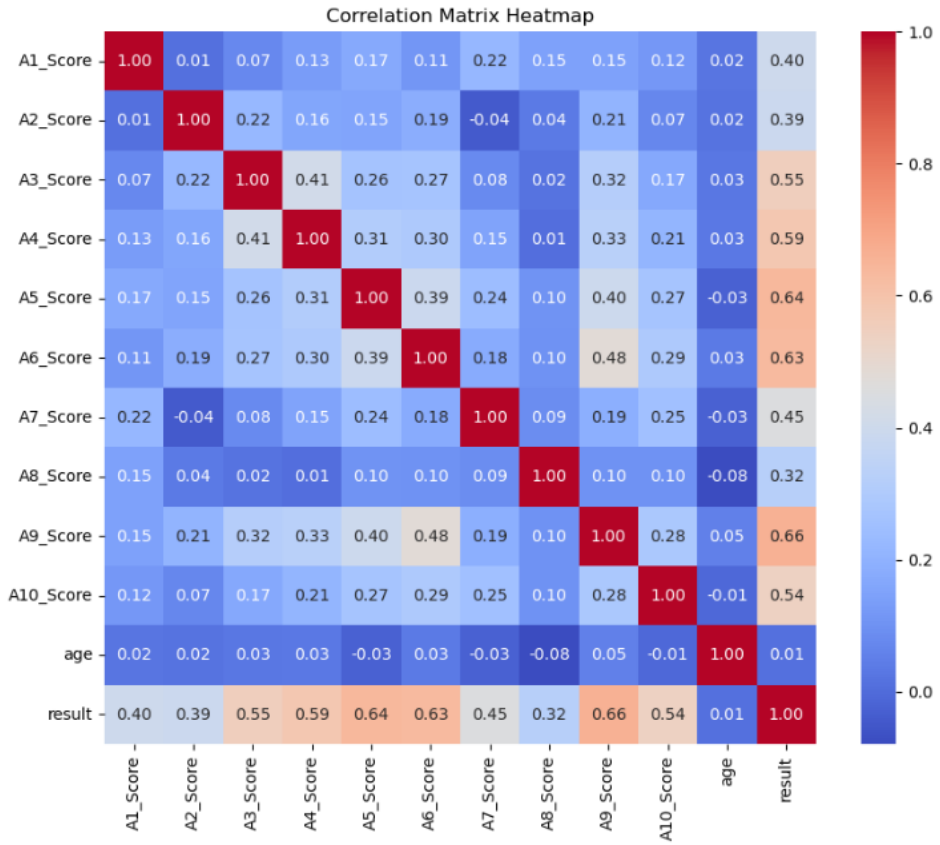
Horizontal bar plots were used for ethnicity and relation due to long category names and many values. Ethnicity showed a few underrepresented groups, while most assessments were self-reported, indicating adult self-screening.

➢ Class Imbalance (Target: ASD vs Non-ASD):



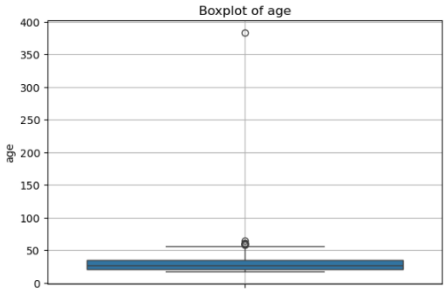
The target column in the given bar chart shows class imbalance, with significantly more Non-ASD (NO) than ASD (YES) participants.

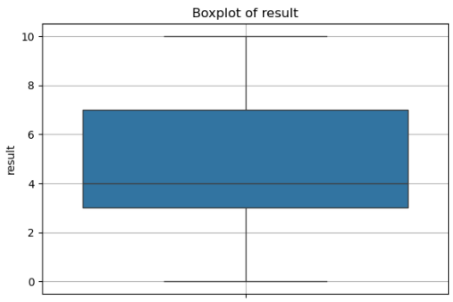
➢ Correlation Heatmap:



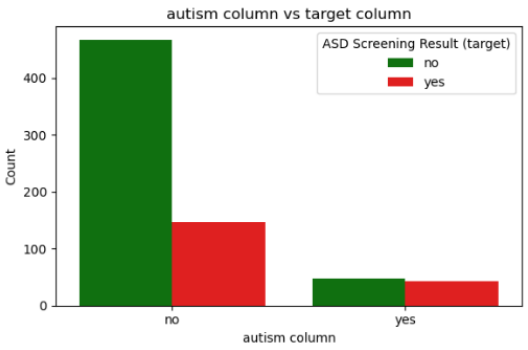
Most A1–A10 scores show moderate to strong positive correlation with the result, especially A9 (0.66), A5 (0.64), A6 (0.63), and A4 (0.59). This is expected as the result is derived from A1–A10 scores.

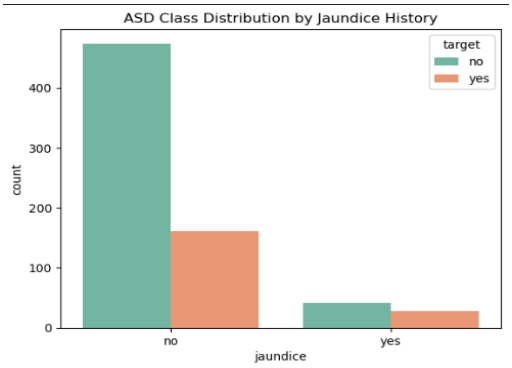
Outliers Detection:

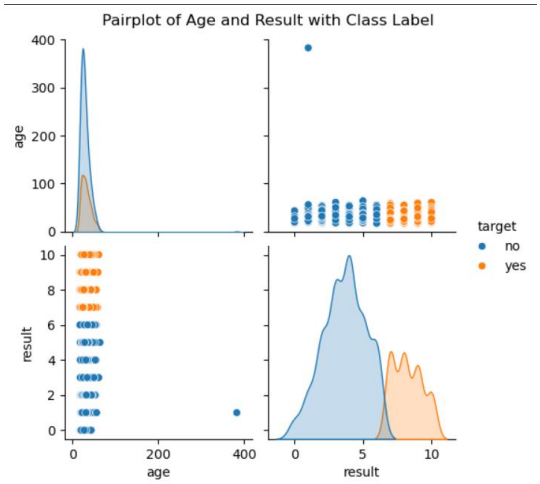


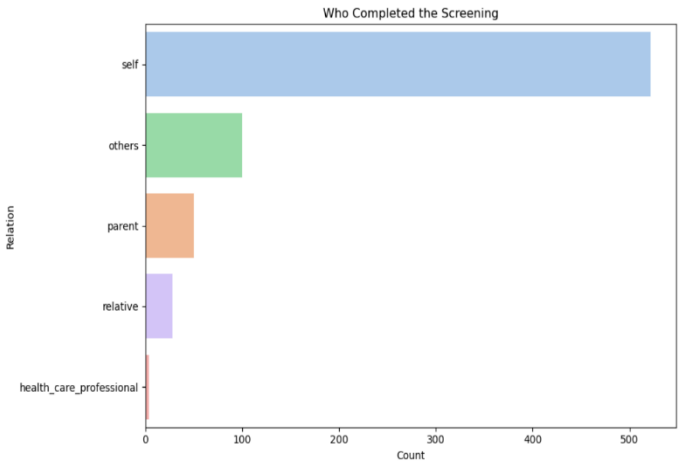


Z-score method detected outliers in numeric columns like age and result, with extreme outliers noted in age (e.g., 383).









2.2 Model Building:

Outliers Removing: Unrealistic entries (e.g., negative values or ages above 120) were identified in the age column. These outliers were replaced by the column's median value to retain reasonable age boundaries.

```
median_age = df[(df['age'] >= 0) & (df['age'] <= 120)]['age'].median()
df.loc[(df['age'] < 0) | (df['age'] > 120), 'age'] = median_age
```

Data Splitting:

To find the performance of each model the given dataset is divided in to training and testing sets. An 80-20 split was implemented, with four-fifths allocated for model development and one-fifth reserved for testing. This division strategy prevents performance inflation and ensures the model's applicability to practical applications

```
# Train split test
from sklearn.model_selection import train_test_split
X = df.drop('target',axis=1)
y = df.target
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

Handling Class Imbalance:

An imbalanced distribution was observed in the target variable, which could result in skewed model performance. The training dataset was processed using SMOTE to address this concern. This technique creates synthetic examples for minority classes, achieving better class equilibrium and improving the model's ability to generalize across different scenarios

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

Models Used:

To identify the most suitable model for predicting Autism Spectrum Disorder (ASD), the following supervised machine learning algorithms were trained and evaluated:

1. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
model1 = LogisticRegression(max_iter=1000)
model1.fit(X_train_data,y_train_data)
```

2. Decision Tree Classifier

```
from sklearn import tree
model2 = tree.DecisionTreeClassifier(random_state=42)
model2.fit(X_train_data,y_train_data)
```

3. Support Vector Machine (SVC)

```
from sklearn.svm import SVC
model3 = SVC()
model3.fit(X_train_data,y_train_data)
```

4. Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
model4 = RandomForestClassifier()
model4.fit(X_train_data,y_train_data)
```

5. Gaussian NB

```
from sklearn.naive_bayes import GaussianNB
model5 = GaussianNB()
model5.fit(X_train_data,y_train_data)
```

Feature Scaling:

Feature scaling was applied using StandardScaler, especially for models sensitive to feature magnitude, such as Logistic Regression, SVM, and Gaussian NB. Standardization ensures all features contribute equally and prevents bias due to varying scales.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Hyperparameter Tuning

To optimize the performance of each ML model, hyperparameter tuning was performed using Grid Search in conjunction with Stratified K-Fold Cross-Validation.

- GridSearchCV:

  GridSearchCV from scikit-learn was employed to exhaustively search over a specified parameter grid for each model. The tuning was done with accuracy as the major assessment criterion.

- Stratified K-Fold Cross- Validation.

  A 5-fold stratified cross-validation was used to ensure that each fold maintained the same class distribution as the full dataset. This approach reduces variance and provides a more reliable estimate of model performance.

```
# Grid Search with Stratified K-Fold
scores_lr = []
CV_lr = StratifiedKFold(n_splits=5, shuffle=True,random_state=42)

for model_name, mp in model_params_lr.items():
    clf_lr = GridSearchCV(mp['model'], mp['params'], cv=CV_lr, scoring='accuracy', return_train_score=False)
    clf_lr.fit(X_train_scaled, y_train)
    scores_lr.append({
        'model': model_name,
        'best_score': clf_lr.best_score_,
        'best_params': clf_lr.best_params_
    })
```

Hyperparameters Considered:

Each model was tuned with specific hyperparameters as follows:

- Logistic Regression:

  For logistic regression the hyperparameters used are penalty and solver. The penalty is used to regularize the model and prevent from overfitting and solver is an algorithm to optimize cost function

```
param_grid = []
param_grid.append({
    'penalty': ['l2'],
    'solver': ['liblinear', 'lbfgs', 'sag', 'newton-cg']
})
param_grid.append({
    'penalty': ['l1'],

    'solver': ['liblinear', 'saga']
})
model_params_lr = {
    'logistic_regression': {
        'model': LogisticRegression(max_iter=3000),
        'params': param_grid
    }
}
```

- Decision Tree Classifier

  For Decision Tree Classifier the hyperparameters used are criterion to check the quality of split, max depth for maximum depth, min samples split for minimum samples required to split the internal node, min samples leaf for minimum number of samples at the leaf node, max features for the number of features to consider when looking for the best split, class weight for weight is balanced or not, ccp alpha is the post-pruning parameter that balances the size of the tree and its performance.

```
model_params_dt = {
    'Decision_tree': {
        'model': DecisionTreeClassifier(random_state=42),
        'params': {
            'criterion': ['gini', 'entropy', 'log_loss'],
            'max_depth': [5, 10, 15, 20, None],
            'min_samples_split': [2, 5, 10],
            'min_samples_leaf': [1, 2, 4],
            'max_features': [None, 'sqrt', 'log2'],
            'class_weight': [None, 'balanced'],
            'ccp_alpha': [0.0, 0.001, 0.01]
        }
    }
}
```

- Support Vector Machine (SVC)

For SVC the hyper parameters used are C, kernel and gamma. The C is a regularization operator and kernel is a function to project data into higher dimensions if not linearly separable and gamma is floating positive number. If low gamma value, then smoother decision boundary otherwise complex decision boundary.

```
model_params_svc = {
    'svm':{
        'model':SVC(),
        'params':{
            'C': [0.01, 0.1, 1, 10, 100],
            'kernel': ['linear', 'rbf'],
            'gamma': ['scale', 'auto', 0.001, 0.01, 0.1, 1]
        }
    }
}
```

- Random Forest Classifier

In this study to increase the performance of the model, we used the parameters the n estimators, max depth and criterion.

The n estimators is used to keep the maximum no. of trees in the forest. The criterion function used to measure the quality of decision tree in each split and max depth is used to for the max depth of each decision tree.

```
model_params_rfc = {
    'random_forest': {
        'model': RandomForestClassifier(random_state=42),
        'params': {
            'n_estimators': [100, 200,500,1000],
            'criterion': ['gini', 'entropy', 'log_loss'],
             'max_depth':[None,20,50,100]

        }
    }
}
```

- Gaussian Naive Bayes

The var_smoothing parameter was configured for the Gaussian Naive Bayes model. This technique adds a negligible amount to each feature's variance, eliminating zero-division problems and maintaining consistent prediction reliability.

In Gaussian Naive Bayes, each feature is assumed to follow a normal distribution. When the variance is very small, the model might make weird predictions or crash due to numerical problems. So var_smoothing helps make the model stable.

```
model_params_gnb = {
    'GaussianNB': {
        'model': GaussianNB(),
        'params': {
           'var_smoothing': [1e-9, 1e-8, 1e-7, 1e-6]

        }
    }
}
```

Results and Discussion:

Before applying hyper parameter tuning the performance metrics of each of model is,

Model Name	Accuracy	Class type	Precision value	Recall value	F1 Score	Support
Logistic Regression model	0.93	0	1.00	0.90	0.95	105
		1	0.78	1.00	0.88	36
Decision Tree	0.82	0	0.94	0.80	0.87	105
		1	0.60	0.86	0.70	36
Support Vector Machine	0.74	0	0.87	0.76	0.81	105
		1	0.49	0.67	0.56	36
Random Forest Classifier	0.95	0	0.99	0.94	0.97	105
		1	0.85	0.97	0.91	36
Gaussian NB	0.94	0	1.00	0.92	0.96	105
		1	0.82	1.00	0.90	36

To increase the performance of the model we have done the hyper parameter tuning and we found the best parameters that increases the performance of the model which are obtained from Grid Search CV for each model

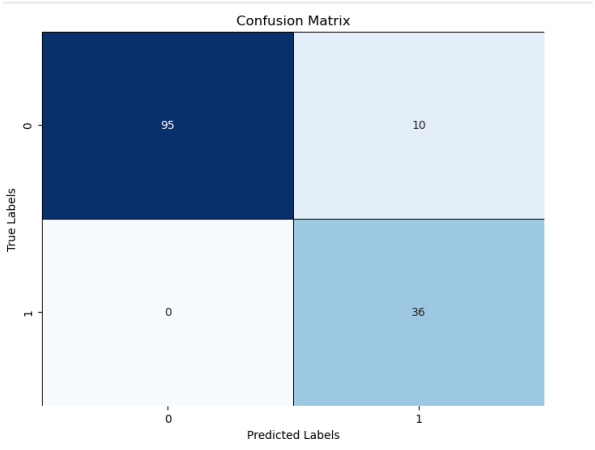
1. Logistic Regression: penalty = l2, solver = liblinear
2. Decision Tree: ccp_alpha = 0.0, class_weight = None, criterion = 'gini', max_depth = 10, max_features = None, min_samples leaf = 1, min_samples_split = 5
3. Support Vector Machine: C = 10, gamma = 0.1, kernel = rbf
4. Random Forest Classifier: criterion = gini, max_depth = None, n_estimators = 200
5. Gaussian NB: var_smoothing = 1e-09

After training each model with these hyper parameters the performance metrics are

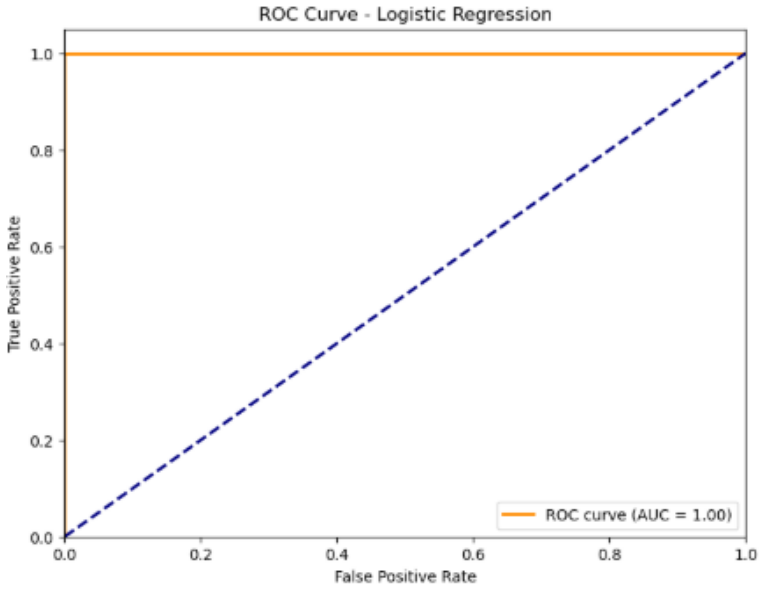
Model Name	Accuracy	Class type	Precision value	Recall value	F1 Score	Support
Logistic Regression model	0.93	0	1.00	0.90	0.95	105
		1	0.78	1.00	0.88	36
Decision Tree	0.82	0	0.93	0.81	0.87	105
		1	0.60	0.83	0.70	36
Support Vector Machine	0.93	0	0.99	0.91	0.95	105
		1	0.80	0.97	0.88	36
Random Forest Classifier	0.95	0	1.00	0.93	0.97	105
		1	0.84	1.00	0.91	36
Gaussian NB	0.94	0	1.00	0.92	0.96	105
		1	0.82	1.00	0.90	36

Evaluation Metrics:

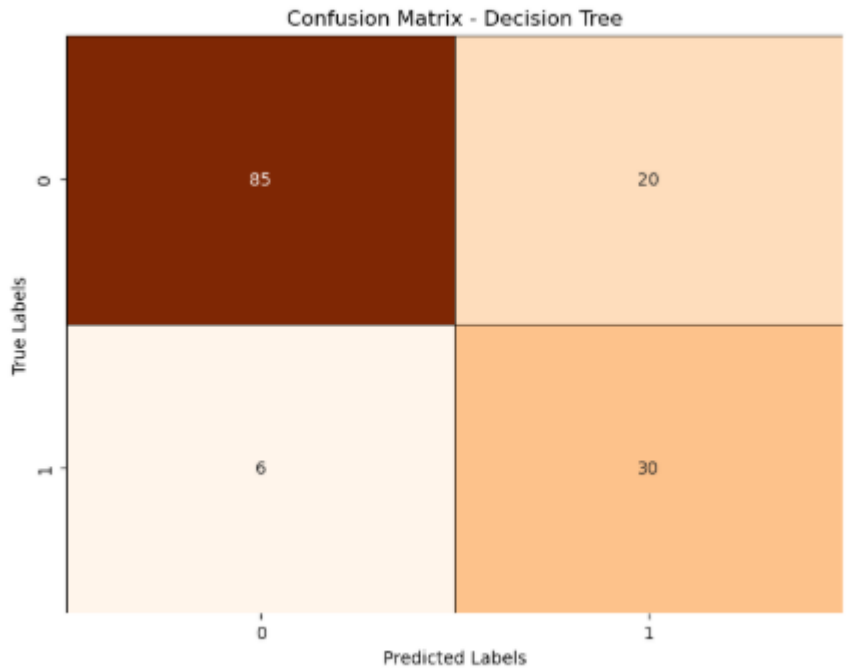
1. Logistic Regression:
    - The tuned Logistic Regression model's performance was assessed through the application of specific evaluation metrics:
    - Confusion matrix:



- AUC Score: 1.00
- ROC Curve:

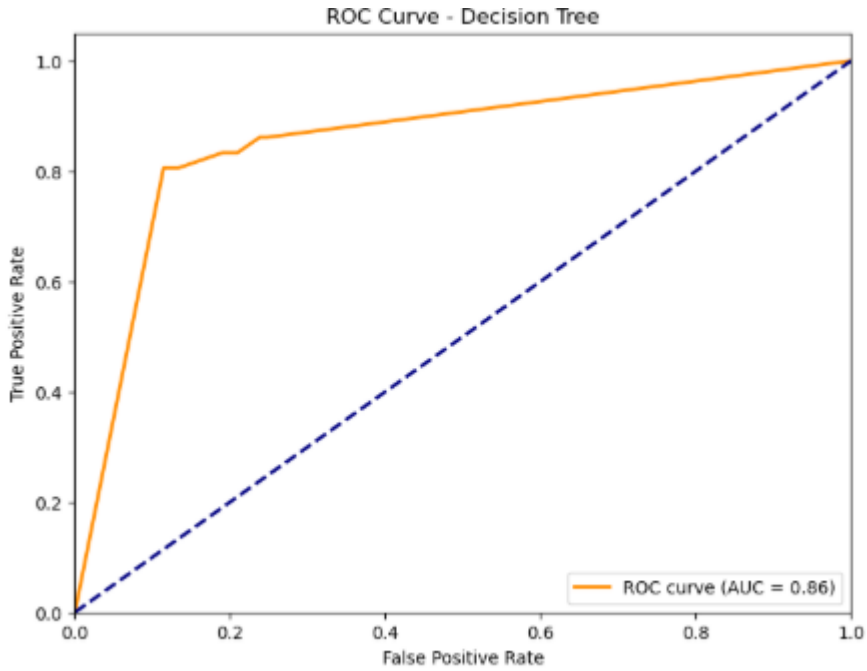


2. Decision Tree Classifier:
    - Performance analysis of the fine-tuned Decision Tree Classifier was conducted using these measurement indicators:
    - The Confusion Matrix:



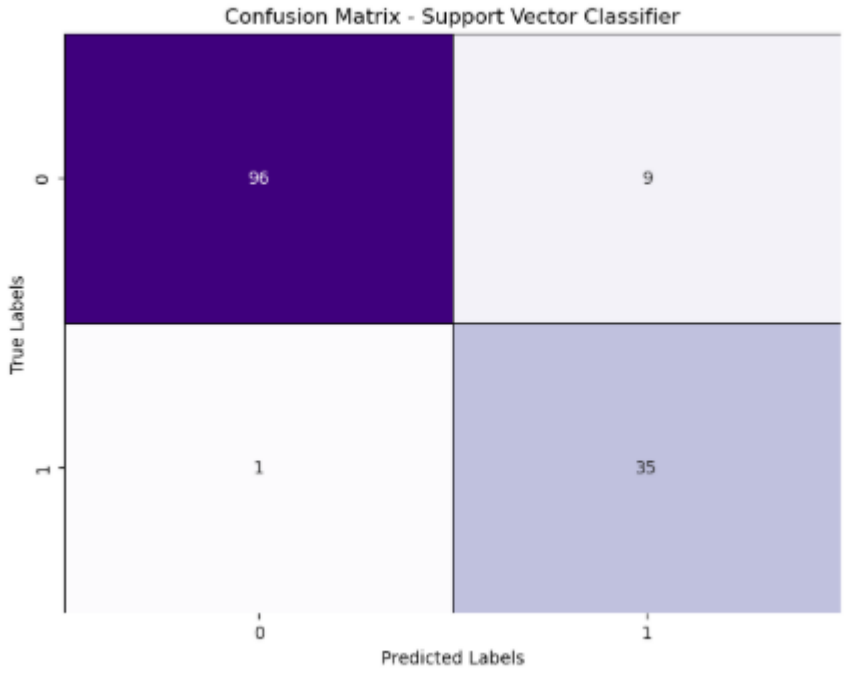
- AUC Score: 0.8566

- ROC Curve:



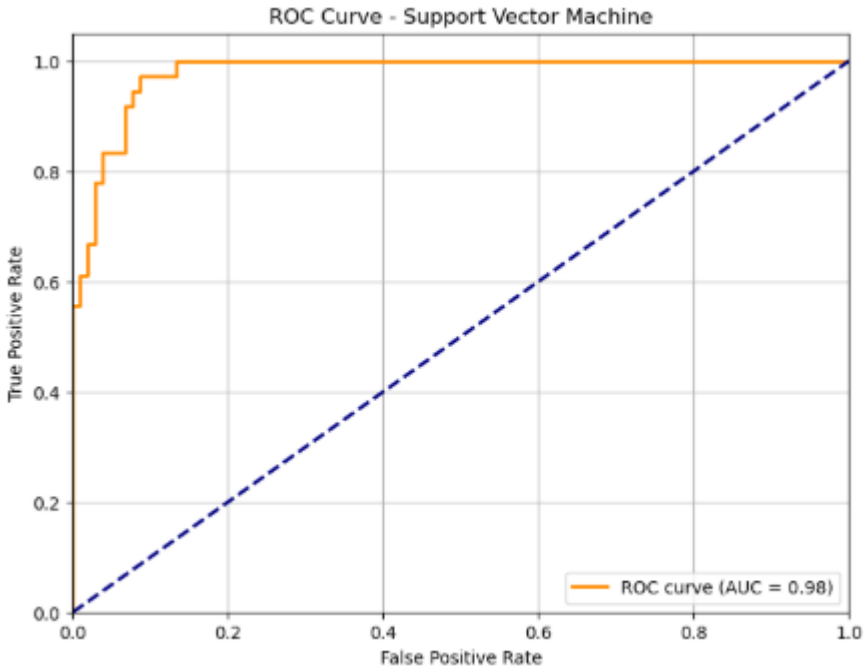
3. Support Vector Machine:

- To evaluate the performance of the tuned Support Vector Machine model, we used the following metrics:
- Confusion Matrix:



- AUC Score: 0.9794

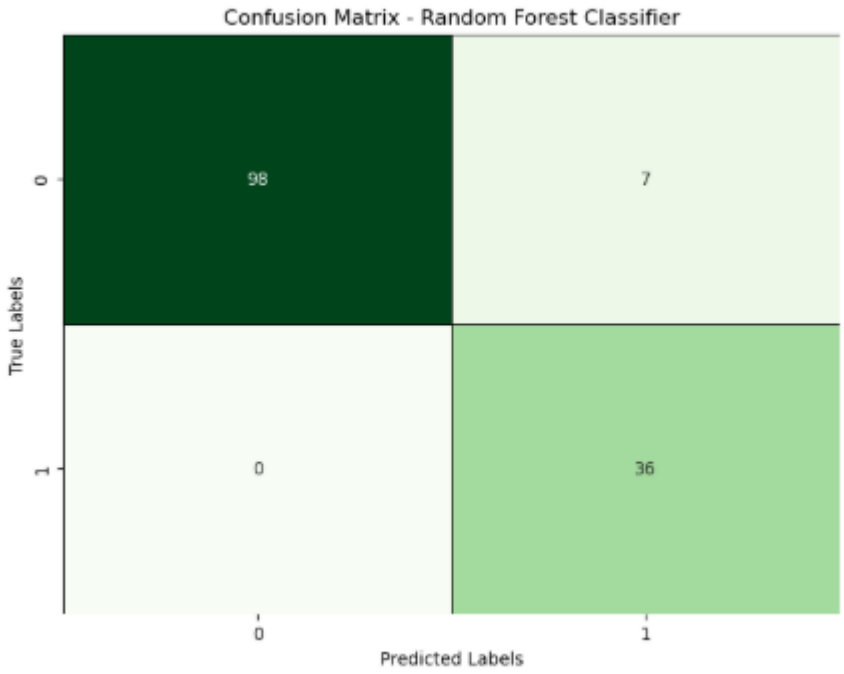
- ROC Curve:



4. Random Forest Classifier:

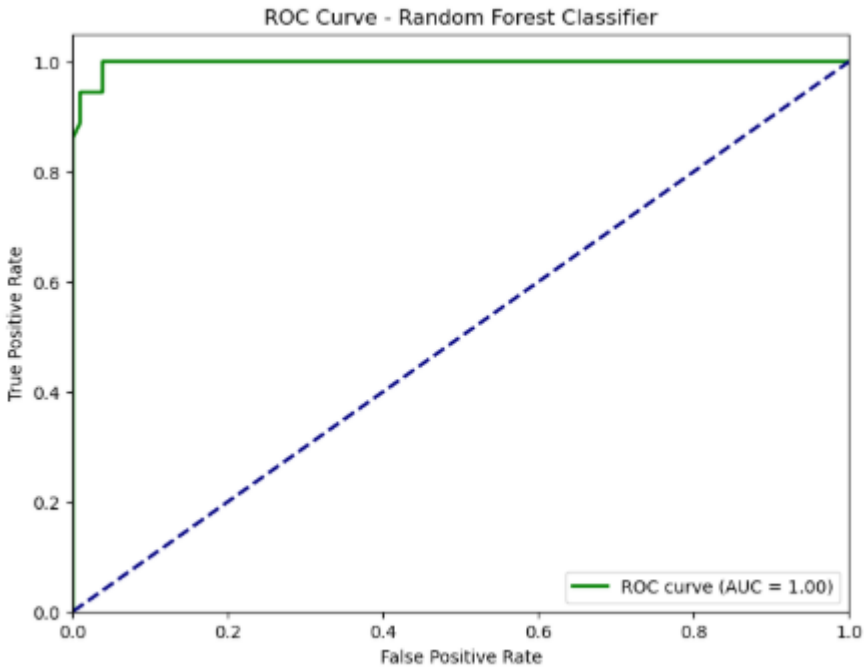
- The following evaluation measures were employed to analyze the optimized Random Forest Classifier's effectiveness:

- Confusion Matrix:



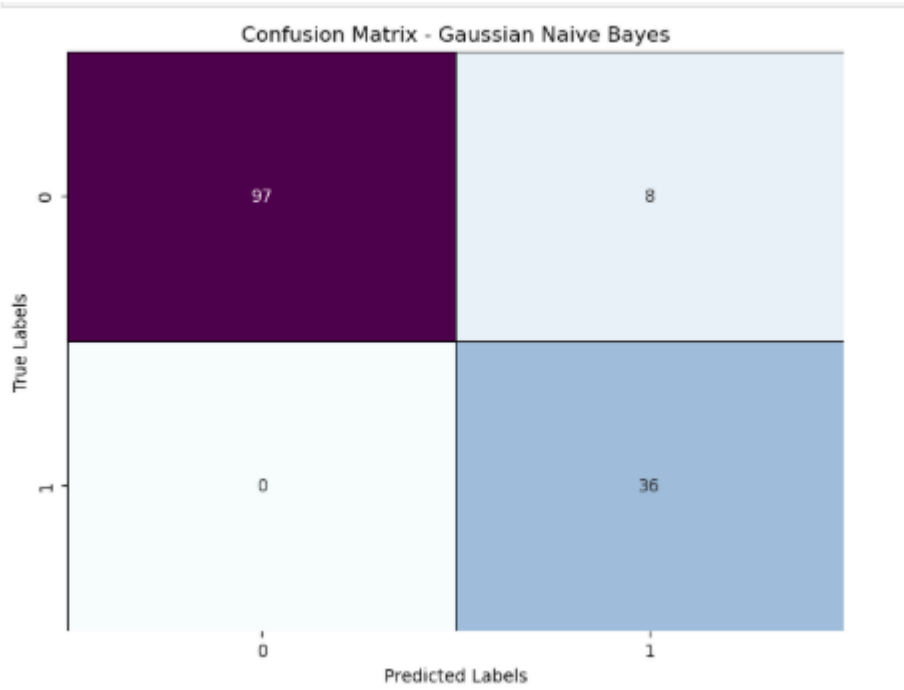
- AUC Score: 0.9972

- ROC Curve:



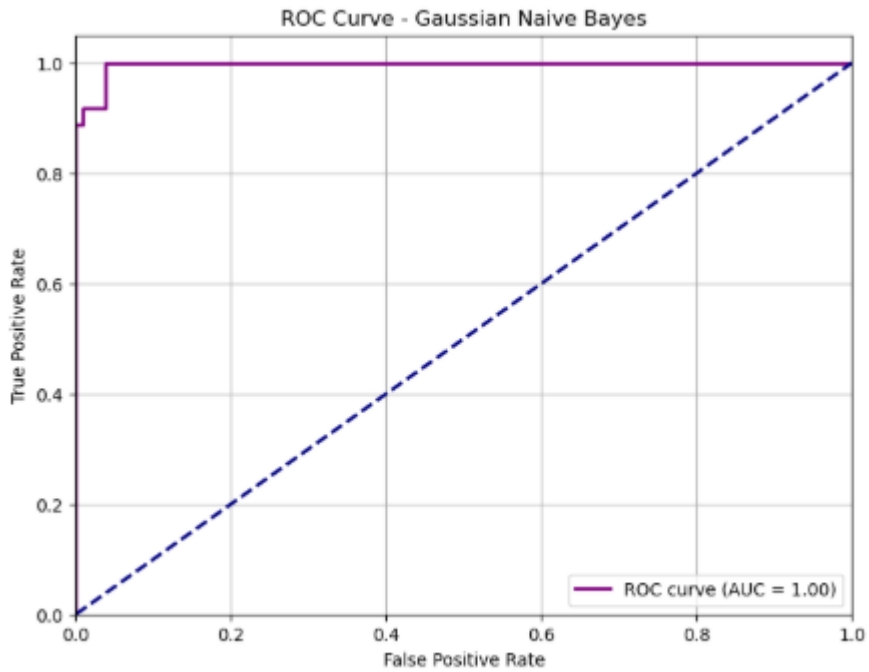
5. Gaussian NB:

- To find the best model of the Gaussian NB model, we decieded based on the following
- Confusion Matrix:



- AUC Score: 0.9966

- ROC Curve:



Best Model Selection:

Based on the comprehensive evaluation of all trained machine learning models, the Random Forest Classifier emerged as the most effective and reliable model for predicting autism in adults. Each model's effectiveness was assessed through standard evaluation measures including accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). Performance Metrics for Random Forest Classifier: Accuracy: 95%

- AUC Score: 0.9972
- Class-wise Performance:
  - The adult individual has no autism (class 0):
    - Precision value: 1.00
    - Recall value: 0.93
    - F1-Score: 0.97
  - If the adult individual has autism (class 1):
    - Precision value: 0.84
    - Recall value: 1.00
    - F1-Score: 0.91

These results indicate that the Random Forest model demonstrates a strong balance in identifying both positive (autistic) and negative (non-autistic) cases.

Reasons for Selecting Random Forest as the Best Model:

- It achieved the highest overall accuracy (95%), outperforming all other models tested.
- It demonstrated a near-perfect AUC score (0.9972), reflecting excellent discriminative power between the two classes across all thresholds.
- The ensemble nature of Random Forest inherently mitigates overfitting, providing reliable generalization on unseen data.
- The model demonstrated balanced performance across all classes, which is crucial in healthcare predictions where both false positives and false negatives can have serious consequences.

Comparison with Other Models:

- Logistic Regression exhibited a perfect AUC score (1.00) but had a lower accuracy (93%) and recall for Class 0 (0.90), making it less effective for balanced predictions.
- Gaussian B is also performed good, with 94% accuracy and an AUC score of 0.9966, but slightly underperformed in F1-score and class-wise consistency compared to Random Forest.
- Support Vector Machine and Decision Tree showed reasonable accuracy but failed to match the consistent and high scores delivered by Random Forest across all evaluation metrics.

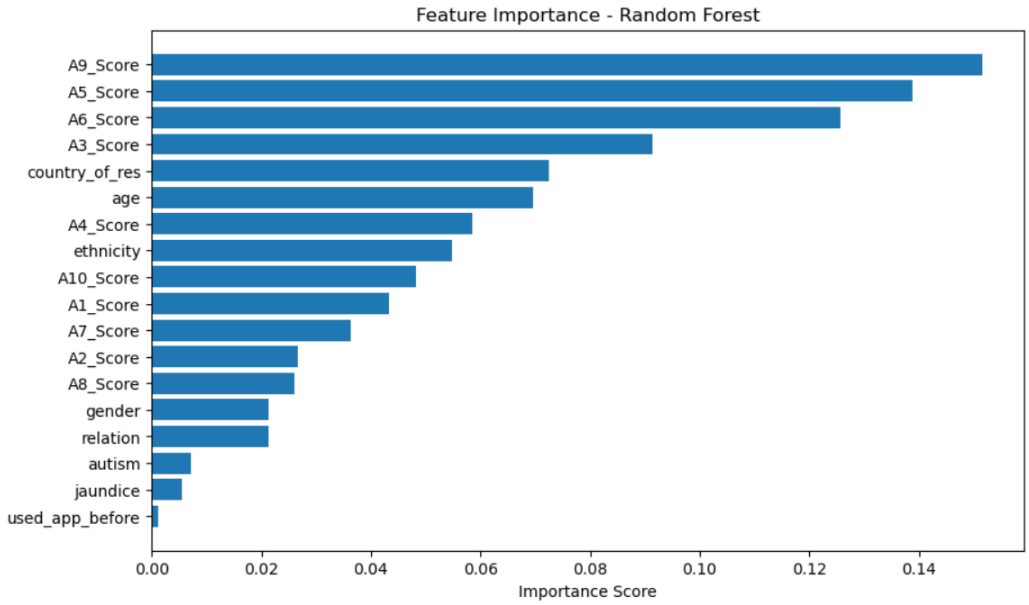
The Random Forest Classifier provides the most optimal balance among precision, recall, and accuracy, making it the most appropriate choice for adult autism prediction in this study. Its strong performance, ability to scale, and ease of interpretation make it a promising candidate for practical implementation in real-world screening applications, delivering dependable and well-rounded diagnostic support.

Outcomes

- The Random Forest Classifier provided the best predictive results for detecting Autism in adults, with:
  - Accuracy: 95%
  - AUC Score: 0.9972
- High precision and recall scores for both autistic and non-autistic classes, demonstrating robust classification capabilities.

Feature Analysis:

- Based on the feature importance graph, the top features contributing to the model's decision-making were:




```
importances_rf = model_rf.feature_importances_
feature_names = X_train.columns
importance_df_rf = pd.DataFrame({'Feature': feature_names, 'Importance': importances_rf})
importance_df_rf = importance_df_rf.sort_values(by='Importance', ascending=False)
importance_df_rf['Importance (%)'] = (importance_df_rf['Importance'] * 100).round(2)
print(importance_df_rf)
```

- A9_Score, A5_Score, and A6_Score — indicating specific behavioral patterns are highly predictive.
- A3_Score, country_of_res, and age also significantly influenced predictions.

- Lesser but still relevant features included ethnicity, gender, relation, and autism (which are related to the family history).
- Least impactful features included used_app_before and jaundice

Interpretability & Insights

- The feature importance results provide interpretability to the model, giving psychological researchers and practitioners clear insight into which screening questions (especially A5, A6, A9) are most indicative of potential autism in adults.
- The model also suggests demographic factors like age and country of residence play a secondary but notable role.
- The final trained model is saved using pickle file.

```
import pickle
```

```
with open ('model.pkl','wb') as f:
    pickle.dump(model_rfc,f)
```

```
with open ('model.pkl','rb') as f:
    final_model = pickle.load(f)
```

- The final trained model is integrated into a web-based screening application developed using HTML, CSS and Flask. Where the individual enters the screening scores and demographical data it will predict that the adult individual has autism or not.

Conclusions/Recommendations

This study effectively designed and assessed a machine learning-driven system to predict Autism Spectrum Disorder (ASD) in adults, utilizing the UCI Autism Screening dataset. After comparing multiple models—including Logistic Regression, Decision Tree, SVM, Gaussian NB, and Random Forest—the Random Forest Classifier emerged as the best-performing algorithm, achieving 95% accuracy and an AUC score of 0.9972. It also demonstrated high precision and recall across both classes, making it reliable for balanced prediction, especially important in a healthcare context.

By identifying key screening questions (notably A5, A6, and A9) and demographic factors (such as age and country of residence) as top predictors, the model offers valuable insights that align with clinical understanding. Furthermore, integrating the final model into a user-friendly web application enables practical deployment for preliminary autism screening in adults, particularly in areas with limited access to professional diagnostic services.

Recommendations:

1. Clinical Testing:

   Future work should include testing the model with real patients under the guidance of doctors or psychologists to check how well it works in real-life situations.

2. Use of More Data:

   Adding more data from people of different ages, regions, and backgrounds can help make the model more accurate and suitable for everyone.

3. Better Understanding of Results:

   Including techniques like SHAP or LIME can help doctors understand why the model gave a particular prediction, making the system more transparent and trustworthy.

4. Mobile App Version:

   Creating a mobile app version of the tool will make it easier to use, especially in rural or remote areas where people may not have access to computers or clinics.

5. Work with Experts:

   Regularly improving the features used in the model by working with mental health professionals can make the tool more accurate and reduce any chances of bias.

   This project shows how machine learning can support early and accessible autism screening in adults, offering potential for timely intervention and improved outcomes.

References

- Ibanga, J. (Department of Computing, Bournemouth University, Dorset, UK). Machine Learning Approaches for Early Detection and Identification of Autism Spectrum Disorders.
- Utilizing Machine Learning Models to Predict Autism Spectrum Disorder Using Limited Medical History and Demographic Data." Published in JAMA Network Open.
- Detection of autism spectrum disorder (ASD) in children and adults using machine learning-Scientific Reports, 13, 9605
- Ahmad, S., Wardhani, L. K., Megantara, R. A., & Fransiscus, Y. (2023). "Application of Machine Learning Algorithms for Identifying Autism Spectrum Disorder Across Pediatric and Adult Populations
- Machine learning-based three-stage sequential diagnosis method for autism spectrum disorder

Mr. Boddu L V Siva Rama Krishna

Assistant Professor

Computer Science and Engineering

School of Engineering and Sciences (SEAS)

Contact: +91 9885050551

sivaramakrishna.b@srmap.edu.in

SRM UNIVERSITY AP
Andhra Pradesh

Date:13-08-2025

To Whomsoever It May Concern

Sub: Summer Research Internship – Completion Certificate

This is to certify that **Mr. GUGGILAM LEELA NAGA SAI SRI SAKETH(AP23110010510)**, a student of Computer Science and Engineering at SRM University-AP, has successfully completed the **Summer Research Internship Program** under my guidance in the **Department of Computer Science and Engineering**, SRM University – AP.

The internship was conducted from **June 2, 2025 to July 25, 2025**. During this period, the intern worked on **Autism Prediction in Adults using Machine Learning Techniques**, contributing to various stages of the research process, including literature review, methodology development, experimentation, result analysis, and preparation of reports/presentations.

We appreciate their dedication, enthusiasm, and active participation in the project, and we wish them success in their future academic and professional endeavors.

Faculty Mentor

Mr. Boddu L. V. Siva Rama Krishna
Assistant Professor
Department of Computer Science and Engineering
SRM University – AP

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

Exclusions

- 1 Excluded Match

Match Groups

13 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
1% Publications
3% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 38 - Integrity Overview Submission ID trn:oid:::8044:108010071

Match Groups

- **13** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
1% Publications
3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	deepai.org <1%
2	Internet	www.coursehero.com <1%
3	Internet	epublications.vu.lt <1%
4	Internet	github.com <1%
5	Internet	www.textileassociationindia.org <1%
6	Internet	randr19.nist.gov <1%
7	Submitted works	Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering and Com... <1%
8	Submitted works	srmap on 2024-08-19 <1%
9	Submitted works	srmap on 2024-11-28 <1%
10	Submitted works	Dublin City University on 2025-07-30 <1%

turnitin Page 3 of 38 - Integrity Overview Submission ID trn:oid:::8044:108010071

11 Internet

bmjpaedsopen.bmj.com <1%

12 Internet

srmap.edu.in <1%

13 Internet

studenttheses.uu.nl <1%

turnitin Page 4 of 38 - Integrity Overview Submission ID trn:oid:::8044:108010071