

# Detection of Autistic Spectrum Disorder :Classification

## Final Project Report

### 1. Introduction

#### a. Project overviews :

This project, "Detection of Autistic Spectrum Disorder: Classification," aims at developing a robust system for identifying and classifying Autism Spectrum Disorder in individual entities using machine learning techniques. ASD is one of the developmental conditions which impacts communication, behavior, and social interaction. Thus, early and correct detection may ensure timely interventions that turn out to have better outcomes for the affected people.

The project focuses basically on the use of data science and machine learning to analyze behavioral patterns and indicative features in a way that could classify people with ASD or not. Of course, all this is done on a web application basically powered by Flask. The user interface provided, of course, supports easy screening and getting results for classification. This application could stand in front of clinicians, researchers, probably parents, and caregivers for them to use; hence, providing a fine tool in the early detection and understanding of ASD.

#### b. Objectives :

- Develop a Reliable Classification Model.
- Data Preprocessing and Feature Engineering.
- Implement a Flask Web Application.
- Validation and Evaluation.
- Enhance Accessibility and Usability

## 2. Project Initialization and Planning Phase

### a. Define Problem Statement

Abnormalities in ASD development include social interaction and communication, together with repetitive behaviors. While it is possible that very early detection and intervention might make a huge difference in the long-term outcome for those with ASD, diagnosis is currently very time-consuming and subjective, often requiring specialized expertise that may not be readily available. This, in turn, calls for a reliable, efficient, and accessible method that aids in the early detection that shall complement the traditional ASD diagnosis approaches.

### b. Project Proposal (Proposed Solution)

Proposed Solution:

The solution proposed to the problem of early and accurate detection of Autism Spectrum Disorder (ASD) will be based on a well-structured approach: data preprocessing, feature selection, model development, and deployment via a web application. The following are the key steps for this project:

#### 1. Data Preprocessing:

Clean and standardize the dataset. Streaming or batch processing – Handling missing values, Normalization of data. Perform exploratory data analysis.

#### 2. Feature Selection:

Use statistical methods and machine learning techniques to find relevant features. Implement feature selection algorithms including RFE and PCA.

#### 3. Model Development:

Develop models with selected features and train them.

Algorithm investigations: Logistic Regression, SVM, Random Forest, Neural Networks. Cross-validate for robustness.

#### 4. Model Evaluation:

Calculate accuracy, precision, recall, and the F1-score as measures for the model's performance. Improve the features and model parameters in the light of evaluation results.

#### 5. Web Application Implementation:

Write a user-friendly web application using Flask. Integrate this trained model into the real-time classification of ASD. Ensure proper processing of data and display of results.

#### 6. Optimization:

Optimizing the model and application by making it computationally efficient. Make backend and frontend performance better. Implement security measures to protect the user's data.

### **Key Features:**

- **Machine Learning Model** : This robust classification model has been designed and trained on a rich dataset for the correct identification of ASD. WindSpeed (m/s): Directly influences the amount of energy generated.
- **User-Friendly Interface** : A web application to provide ease and user-friendly experience.
- **Real-Time Results** : Provides immediate feedback of the likelihood of ASD, given users' input data.
- **Data privacy and security** : Be assured that this is where confidentiality and security of the user's data are maintained.

### **c. Initial Project Planning :**

Project scope would identify target users, data requirements and sources, and the outline of the functionalities of the web application. The resources are to be allocated by identification of the tools and technologies that are to be used. Tools and technologies include programming languages such as Python and Flask, and machine learning libraries. Risks are mostly concentrated on quality and model bias, with a focus on security. It will be well documented, with regular progress reports. The key phases in this project include: research and data collection, model development, web application development, and testing and deployment.

- **Project Kickoff**: Define project objectives, outline deliverables, and establish initial meetings to discuss project scope and planning.
- **Data Collection**: Source various data sets related to ASD detection and contextualize them into a single consolidated data set, checking for completeness and reliability.
- **Preliminary Analysis**: Perform an initial exploration of the data to understand its structure, distribution, and identify any potential issues.
- **Feature Selection Strategy**: Develop a strategy for selecting key features, utilizing techniques such as correlation analysis, feature importance ranking, and input from domain experts.
- **Modeling Framework**: Choose and implement appropriate modeling techniques and tools for developing predictive models, such as machine learning algorithms and statistical methods.
- **Timeline and Milestones**: Create a detailed project timeline with key phases, milestones, and deadlines to guide the project's progress.
- **Risk Assessment**: Identify risks and develop strategies to address them.

- Stakeholder Communication: Plan regular updates and communication with stakeholders to ensure project alignment, manage expectations, and address any concerns.

### **3. Data Collection and Preprocessing Phase**

#### **a. Data Collection Plan and Raw Data Sources Identified**

We will collect data from various reliable sources, including:

- Clinical Records: Old records from health care providers w.r.t. diagnosis for ASD and assessment related to the same.
- Behavioral Assessment: Treated data from standardized behavior evaluations and questionnaires used in ASD screening
- Research Dataset: Available datasets from academic studies and research institutions on ASD.
- Surveys and Questionnaires: Data from surveys and questionnaires capturing relevant behavioral and development information.

#### **b. Data Quality Report**

A data quality report will be generated that will assess the completeness, accuracy, consistency, and validity of the data. Completeness: Ensuring no significant gaps in the data.

- Accuracy: Verifying data correctness against known standards.
- Completeness: This will ensure that there are no large gaps or missing values in this data.
- Accuracy: It refers to the correctness of the data against known standards or benchmarks.
- Consistency: This checks if formats, values, and units used are uniform across the different sources.
- Validity: It ensures the data falls within expected ranges, adheres to constraint and definitions particular to ASD detection.
- Timeliness: Check that the data is current and relevant for appropriate ASD classification.

### **c. Data Exploration and Preprocessing**

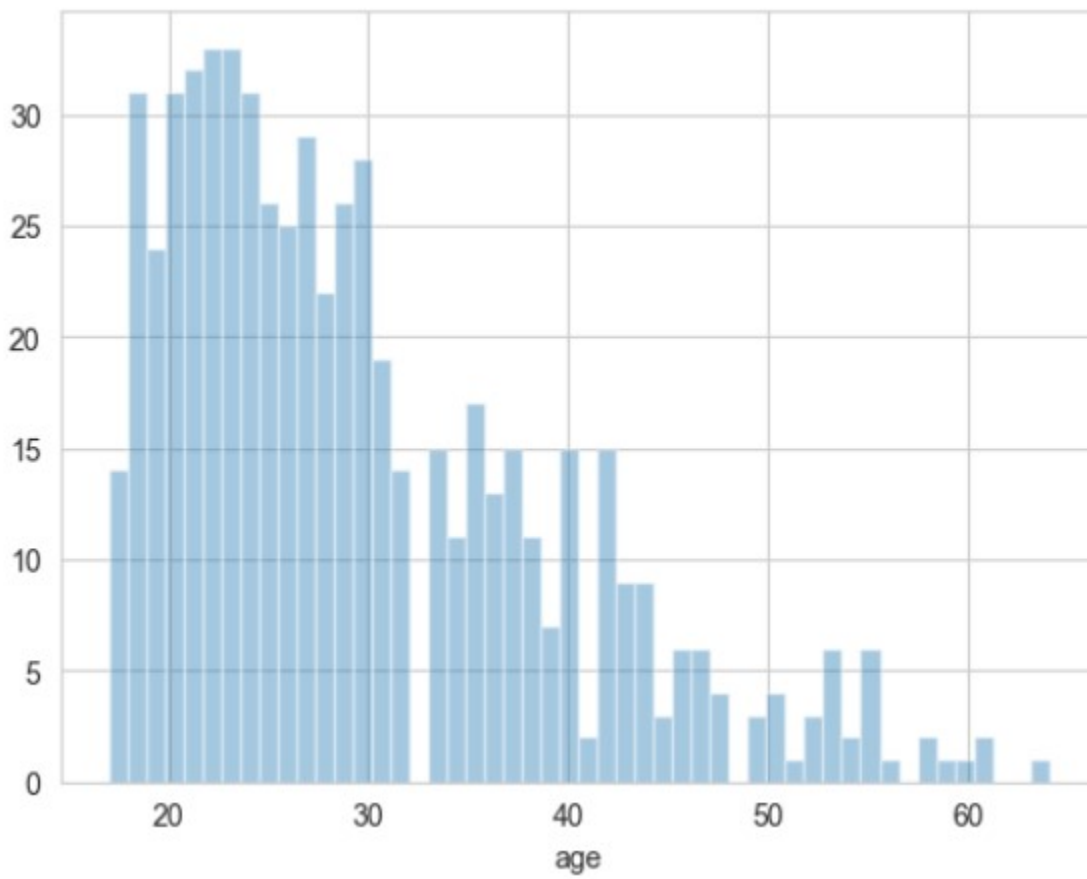
In the initial exploration of data, the following will be studied:

- Descriptive Statistics: The distributions, central trends, and variability of the data.
- Visualization: Charts and graphs will show users patterns and trends in the data and point out the outliers.

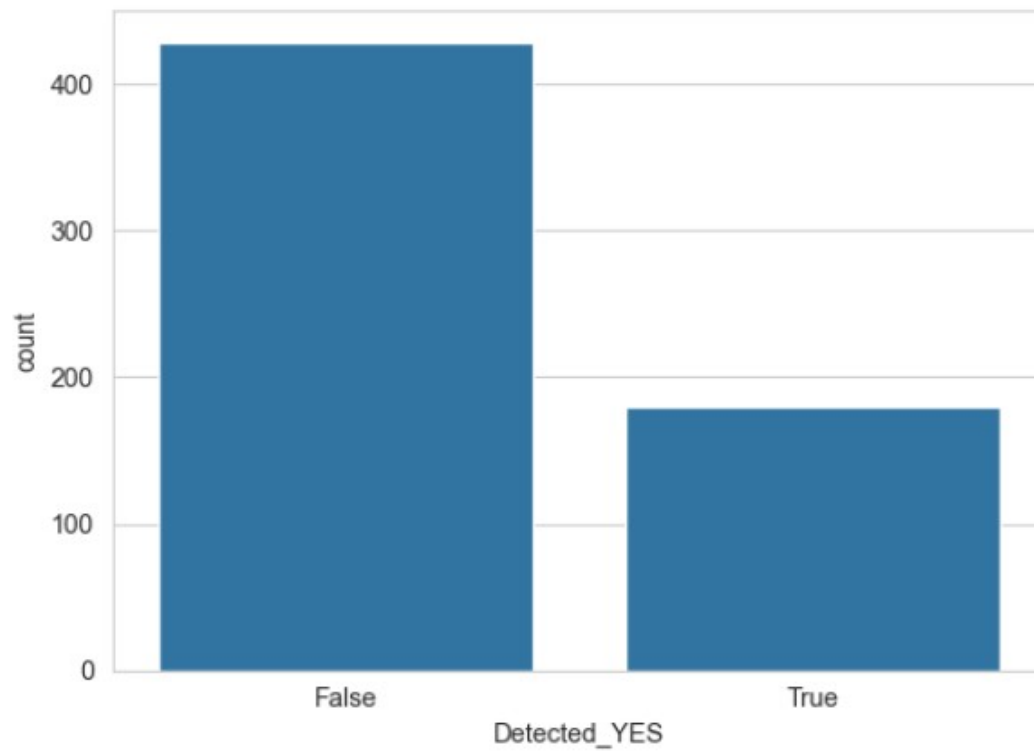
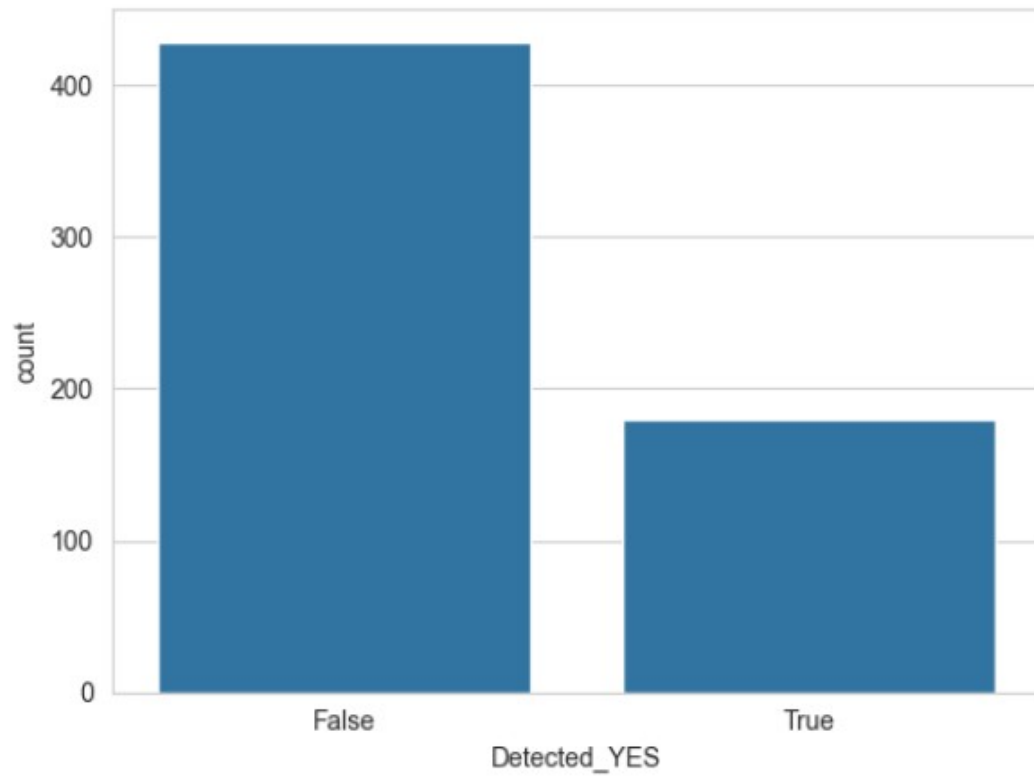
Pre-processing will include:

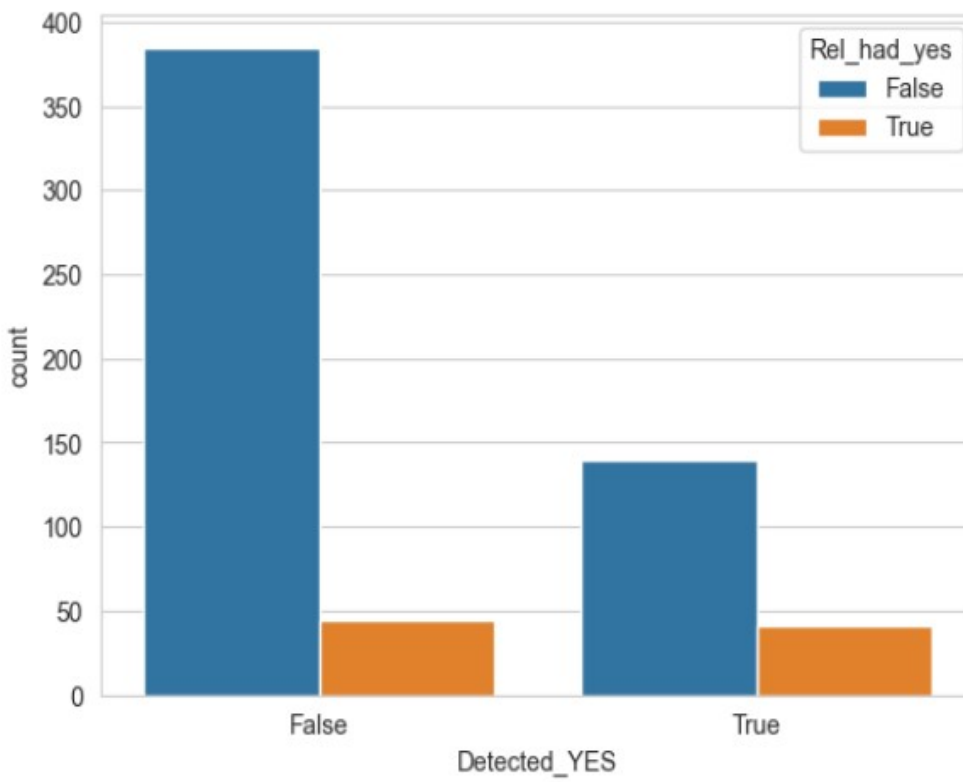
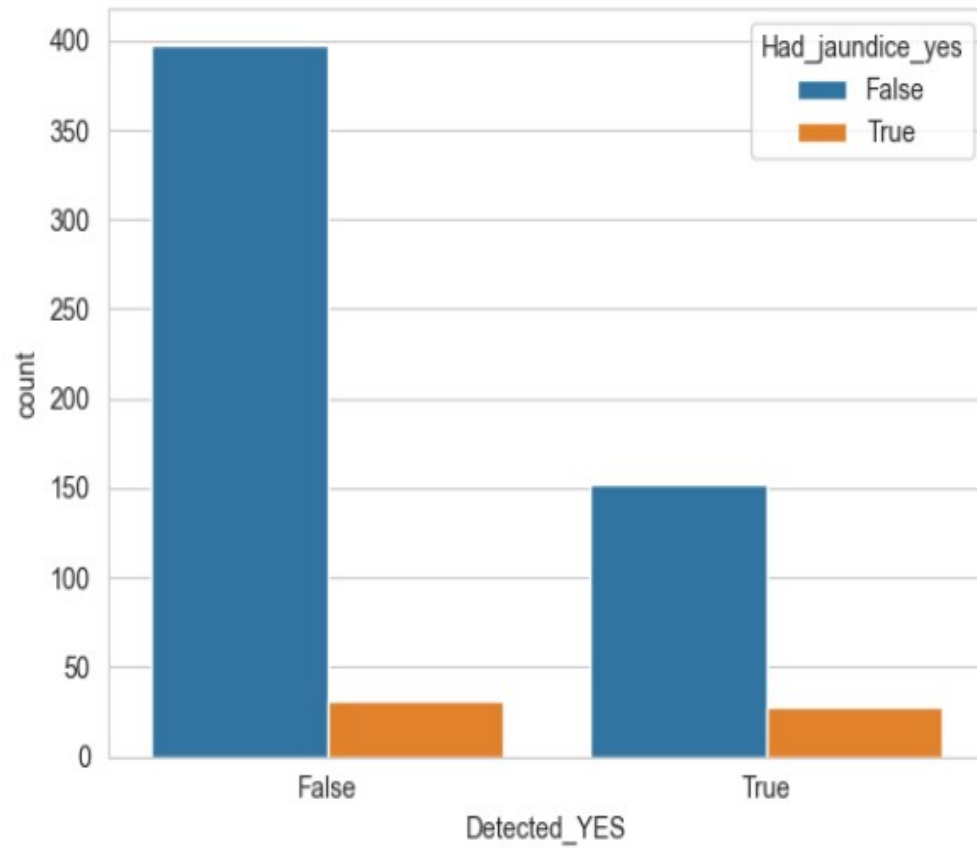
- Cleaning: Handling missing values, errors, and duplicates.
- Transformation: Ensuring consistency; scaling or normalizing data for better performance of the model.
- Feature Engineering: It creates new features or modifies existing ones based on domain knowledge to improve model accuracy.
- Data Split: This means a division in the dataset into training, validation, and test sets for training models effectively and assessing their performance.

distplot:



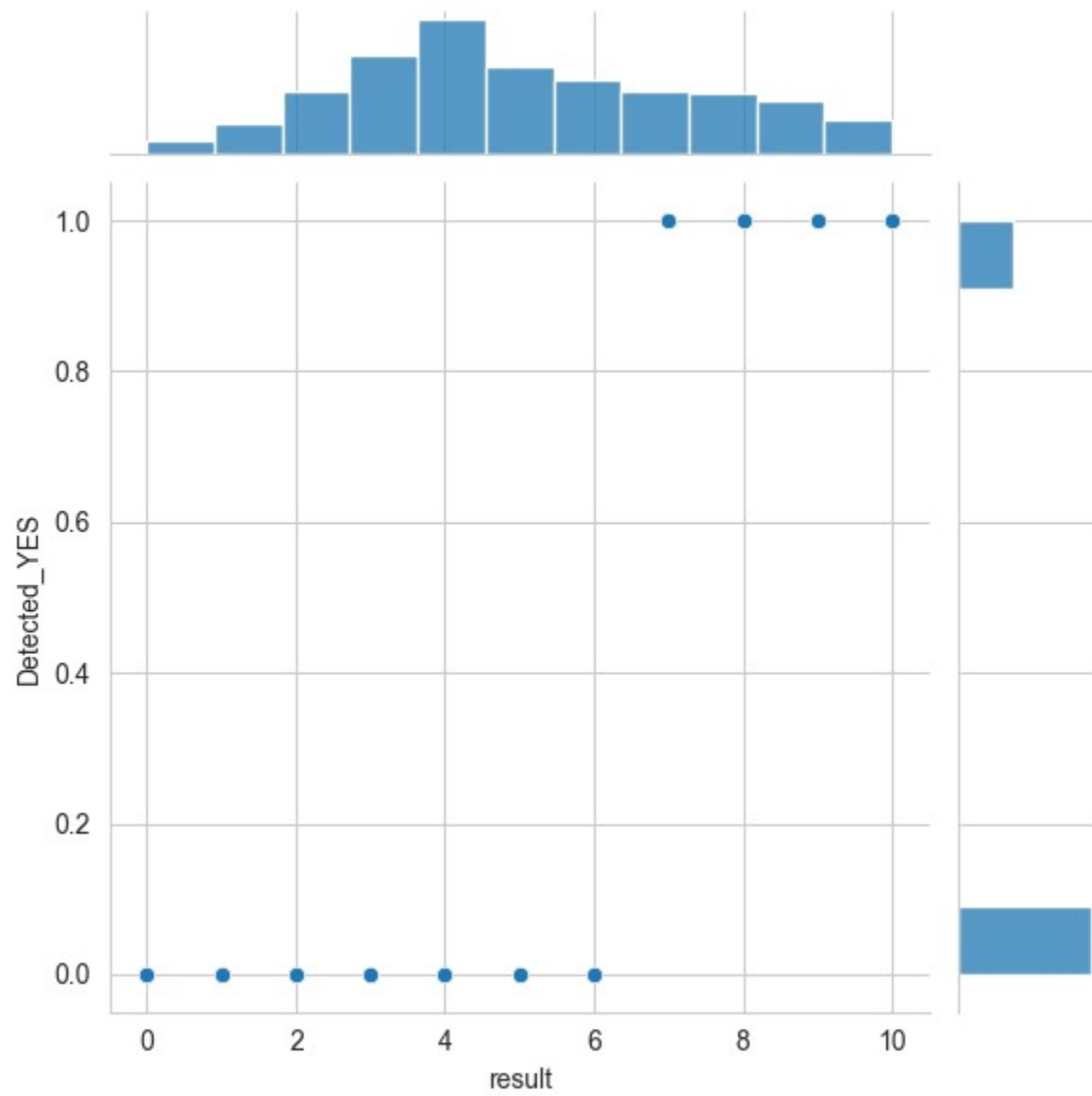
Countplot:



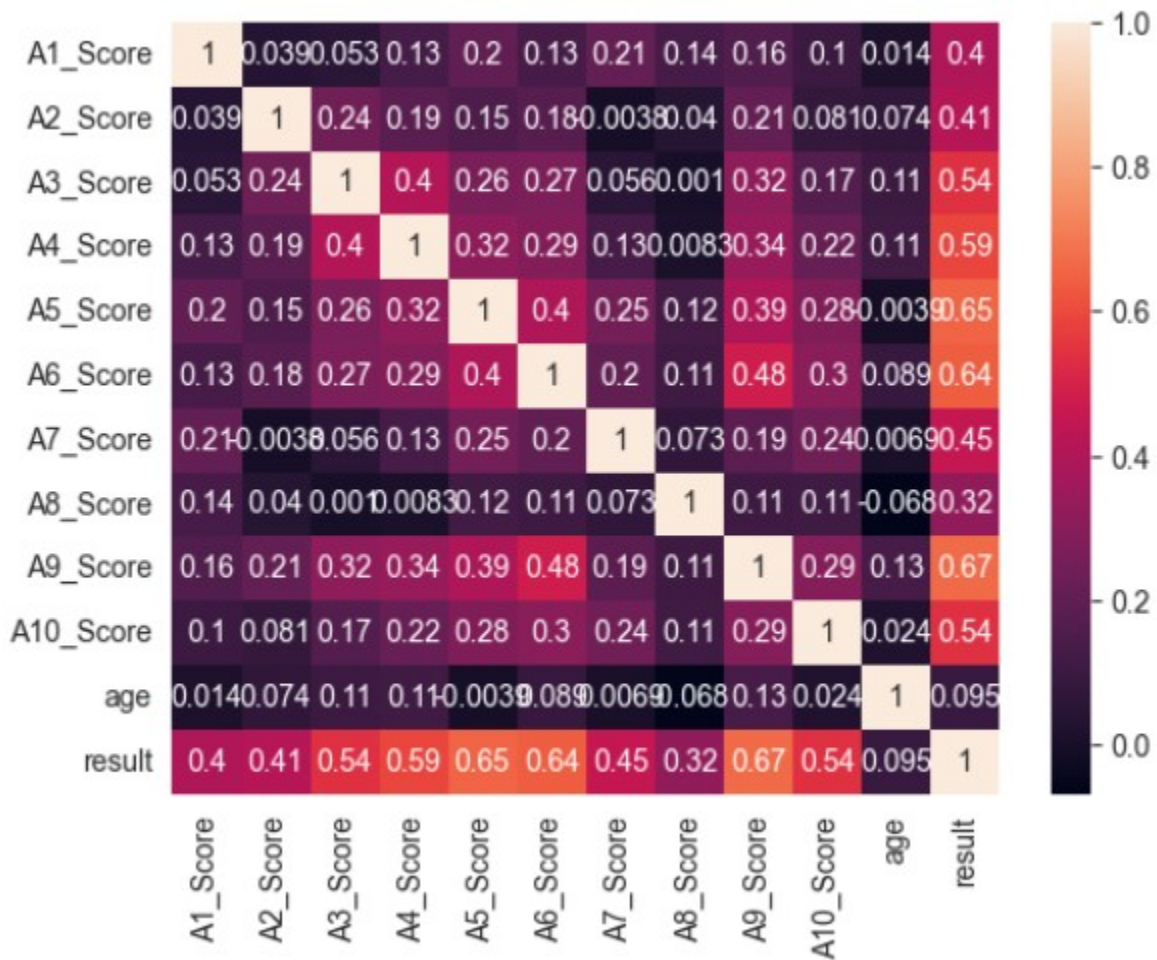




Jointplot:



Heatmap :



Loading Data :

```
data=pd.read_csv("Autism_Data.arff")
```

Handling Missing Data:

```
data.replace("?",np.nan,inplace=True)
```

```
data.head(20)
```

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	...	gender	ethnicity	jundice	austim	contry_of_res	used_app_before	result	age_desc
0	1	1	1	1	0	0	1	1	0	0	...	f	White-European	no	no	'United States'	no	6	'18 and more'
1	1	1	0	1	0	0	0	1	0	1	...	m	Latino	no	yes	Brazil	no	5	'18 and more'
2	1	1	0	1	1	0	1	1	1	1	...	m	Latino	yes	yes	Spain	no	8	'18 and more'
3	1	1	0	1	0	0	1	1	0	1	...	f	White-European	no	yes	'United States'	no	6	'18 and more'
4	1	0	0	0	0	0	0	1	0	0	...	f	NaN	no	no	Egypt	no	2	'18 and more'
5	1	1	1	1	1	0	1	1	1	1	...	m	Others	yes	no	'United States'	no	9	'18 and more'
6	0	1	0	0	0	0	0	1	0	0	...	f	Black	no	no	'United States'	no	2	'18 and more'
7	1	1	1	1	0	0	0	0	1	0	...	m	White-European	no	no	'New Zealand'	no	5	'18 and more'
8	1	1	0	0	1	0	0	1	1	1	...	m	White-European	no	no	'United States'	no	6	'18 and more'
9	1	1	1	1	0	1	1	1	1	0	...	m	Asian	yes	yes	Bahamas	no	8	'18 and more'
10	1	1	1	1	1	1	1	1	1	1	...	m	White-European	no	no	'United States'	no	10	'18 and more'
11	0	1	0	1	1	1	1	0	0	1	...	f	'Middle Eastern '	no	no	Burundi	no	6	'18 and more'
12	0	1	1	1	1	1	0	0	1	0	...	f	NaN	no	no	Bahamas	no	6	'18 and more'
13	1	0	0	0	0	0	1	1	0	1	...	m	NaN	no	no	Austria	no	4	'18 and more'

Creating dummy variables :

```
sex=pd.get_dummies(data['gender'],drop_first=True)
jaund=pd.get_dummies(data['jundice'],drop_first=True,prefix="Had_jaundice")
rel_autism=pd.get_dummies(data['austim'],drop_first=True,prefix="Rel_had")
detected=pd.get_dummies(data['Class/ASD'],drop_first=True,prefix="Detected")
```

Updating the dataset:

```
data=data.drop(['gender','jundice','austim','Class/ASD'],axis=1)
data_featured=pd.concat([data,sex,jaund,rel_autism,detected],axis=1)
data_featured.head()
```

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	age	result	m	Had_jaundice_yes	Rel_had_yes	Detected_YES
0	1	1	1	1	0	0	1	1	0	0	26.0	6	False	False	False	False
1	1	1	0	1	0	0	0	1	0	1	24.0	5	True	False	True	False
2	1	1	0	1	1	0	1	1	1	1	27.0	8	True	True	True	True
3	1	1	0	1	0	0	1	1	0	1	35.0	6	False	False	True	False
5	1	1	1	1	1	0	1	1	1	1	36.0	9	True	True	False	True

#### 4. Model Development Phase

##### a. Model Selection Report

Model	Description
Logistic Regression	A linear model used for binary classification. It calculates the probability of a sample belonging to a particular class using a logistic function.
Support Vector Machine (SVM)	A classification model that finds the hyperplane that best separates the classes. It can handle non-linearity using kernel functions.
Decision Tree	A tree-based model that splits the data based on feature values to make predictions. It's easy to visualize and interpret.
Random Forest	An ensemble method that combines multiple decision trees to improve performance and reduce overfitting. Each tree is trained on a subset of the data

K-Nearest Neighbors (KNN)	A non-parametric method that classifies samples based on the majority label of their nearest neighbors in the feature space.
---------------------------	--

## a. Initial Model Training Code, Model Validation and Evaluation Report

Training code :

### 1) Logistic Regression:

```
from sklearn.linear_model import LogisticRegression

lgr=LogisticRegression()

lgr.fit(X_train,y_train)

LogisticRegression()

pred=lgr.predict(X_test)

y_pred_lgr = lgr.predict(X_test)

from sklearn.metrics import classification_report

accuracy_lgr = accuracy_score(y_test,y_pred_lgr)
print('Accuracy LGR:', accuracy_lgr*100)

Accuracy LGR: 100.0
```

## 2) SVM:

# SVM

```
from sklearn.svm import SVC  
svm=SVC(kernel='rbf', random_state=0)  
svm.fit(X_train, y_train)
```

▼ SVC ⓘ ?  
SVC(random\_state=0)

```
y_pred_svc=svm.predict(X_test)
```

```
print('Training Set: ', svm.score (X_train,y_train))
```

```
print('Testing Set:',svm.score(X_test,y_test))
```

Training Set: 0.9530516431924883

Testing Set: 0.9453551912568307

```
accuracy_SVC=svm.score(X_test,y_test)  
print('Accuracy_SVM:', accuracy_SVC*100)
```

Accuracy\_SVM: 94.53551912568307

### 3) Decision Tree:

## Decision Tree

```
dt = DecisionTreeClassifier()  
  
dt.fit(X_train,y_train)
```

▼ DecisionTreeClassifier ⓘ ?  
DecisionTreeClassifier()

```
y_pred_dt=dt.predict(X_test)
```

```
print('Training Set: ',dt.score(X_train,y_train))  
  
print('Test Set: ',dt.score(X_test,y_test))
```

Training Set: 1.0  
Test Set: 1.0

```
print("Accuracy:", metrics.accuracy_score(y_test, y_pred_dt)*100)
```

Accuracy: 100.0

```
accuracy_dt=accuracy_score(y_test,y_pred_dt)  
print('Accuracy DT:', accuracy_dt*100)
```

Accuracy DT: 100.0

#### 4) Random Forest:

# Random Forest

```
rand_forest = RandomForestClassifier(random_state=42)
```

```
rand_forest.fit(X_train, y_train)
```

```
▼ RandomForestClassifier ⓘ ?  
RandomForestClassifier(random_state=42)
```

```
y_pred_rf=dt.predict(X_test)
```

```
predictionRF = rand_forest.predict(X_test)  
  
print('Training set: ',rand_forest.score(X_train, y_train))  
print('Testing set: ',rand_forest.score(X_test, y_test))
```

```
Training set:  1.0  
Testing set:  1.0
```

```
accuracy_RF=rand_forest.score(X_test, y_test)  
print ("Accuracy_RF:",accuracy_RF*100)
```

```
Accuracy_RF: 100.0
```



5) KNN:

# KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
knn.fit(X_train, y_train)
```

▼ KNeighborsClassifier ⓘ ?  
KNeighborsClassifier()

```
y_pred = knn.predict(X_test)
```

```
#Calculate accuracy of the model

from sklearn.metrics import accuracy_score
accuracy_KNN = accuracy_score(y_test, y_pred)
print(f'Accuracy_KNN: {accuracy_KNN*100}')
```

Accuracy\_KNN: 96.17486338797814

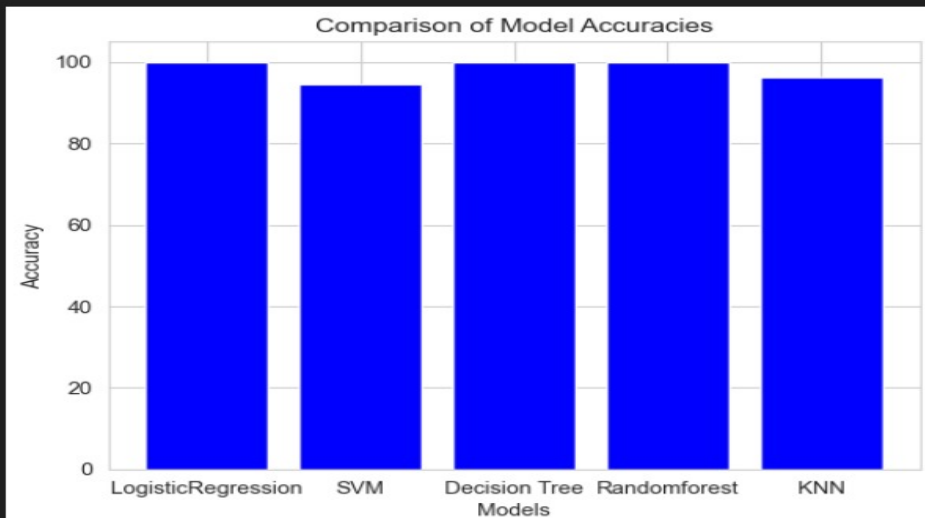
## b .Performance Metrics Comparison Report :

```
accuracy_df = pd.DataFrame({  
    'Model': ['LogisticRegression', 'SVM', 'DecisionTree', 'Randomforest', 'KNN'],  
    'Accuracy': [accuracy_lgr*100, accuracy_SVC*100, accuracy_dt*100, accuracy_RF*100, accuracy_KNN*100]  
})  
  
print(accuracy_df)
```

	Model	Accuracy
0	LogisticRegression	100.000000
1	SVM	94.535519
2	DecisionTree	100.000000
3	Randomforest	100.000000
4	KNN	96.174863

```
models = ['LogisticRegression', 'SVM', 'Decision Tree', 'Randomforest', 'KNN']  
accuracies = [accuracy_lgr*100, accuracy_SVC*100, accuracy_dt*100, accuracy_RF*100, accuracy_KNN*100]  
plt.bar(models, accuracies, color='blue')  
  
#Add title and axis Labels  
  
plt.title('Comparison of Model Accuracies')  
plt.xlabel('Models')  
plt.ylabel('Accuracy')
```

Text(0, 0.5, 'Accuracy')



## 5. Model Optimization and Tuning Phase

### b. Final Model Selection Justification

Model : Random Forest Classifier

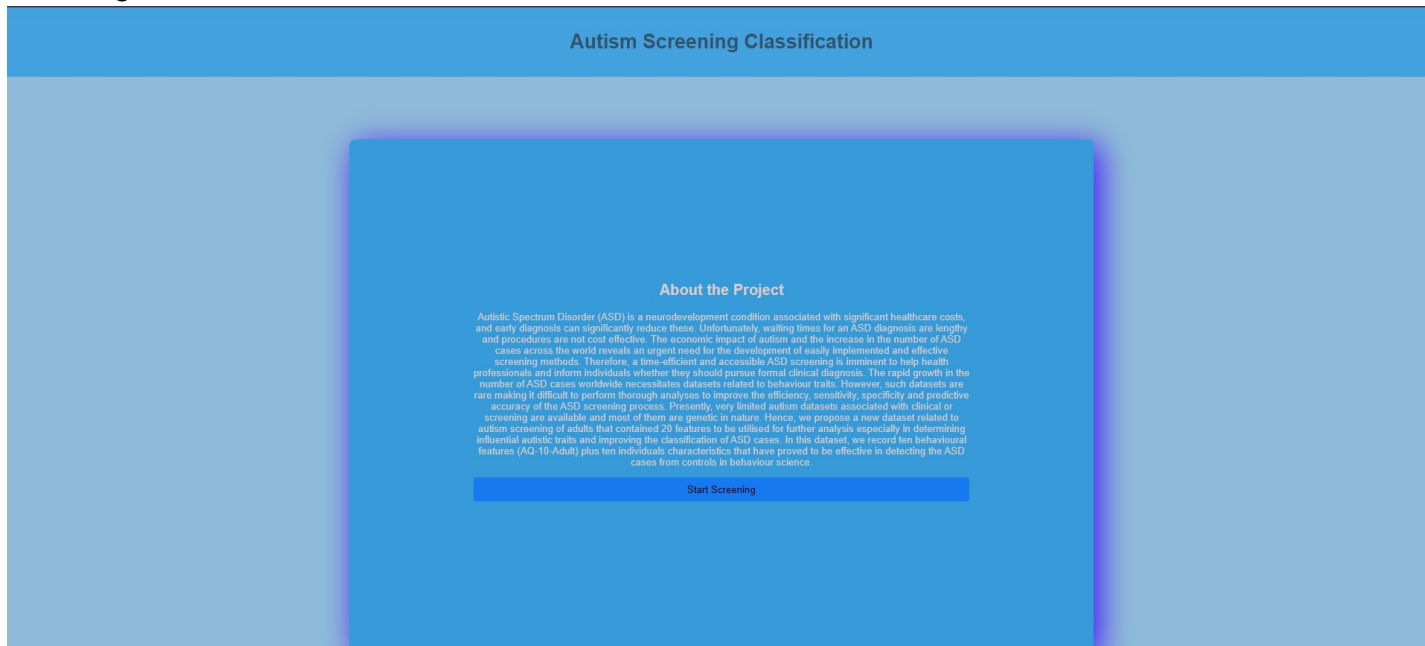
Reasoning :

The final model chosen to be used was the Random Forest Classifier because of its robust performance in handling complex datasets with multiple features. It concatenates the predictions of a large number of decision trees, thereby reducing overfitting and improving accuracy. It works well with nonlinear relationships and feature interactions, which are important in the correct classification of Autism Spectrum Disorder. High-performance metrics, such as accuracy, precision, and recall of the random forest classifier, establish its reliability and efficiency for making accurate ASD predictions.

## 6. Results

### a. Output Screenshots

Home Page :



Input :

ASD Screening

A1\_Score

A2\_Score

A3\_Score

A4\_Score

A5\_Score

A6\_Score

A7\_Score

A8\_Score

A8\_Score

A9\_Score

A10\_Score

Age

Result

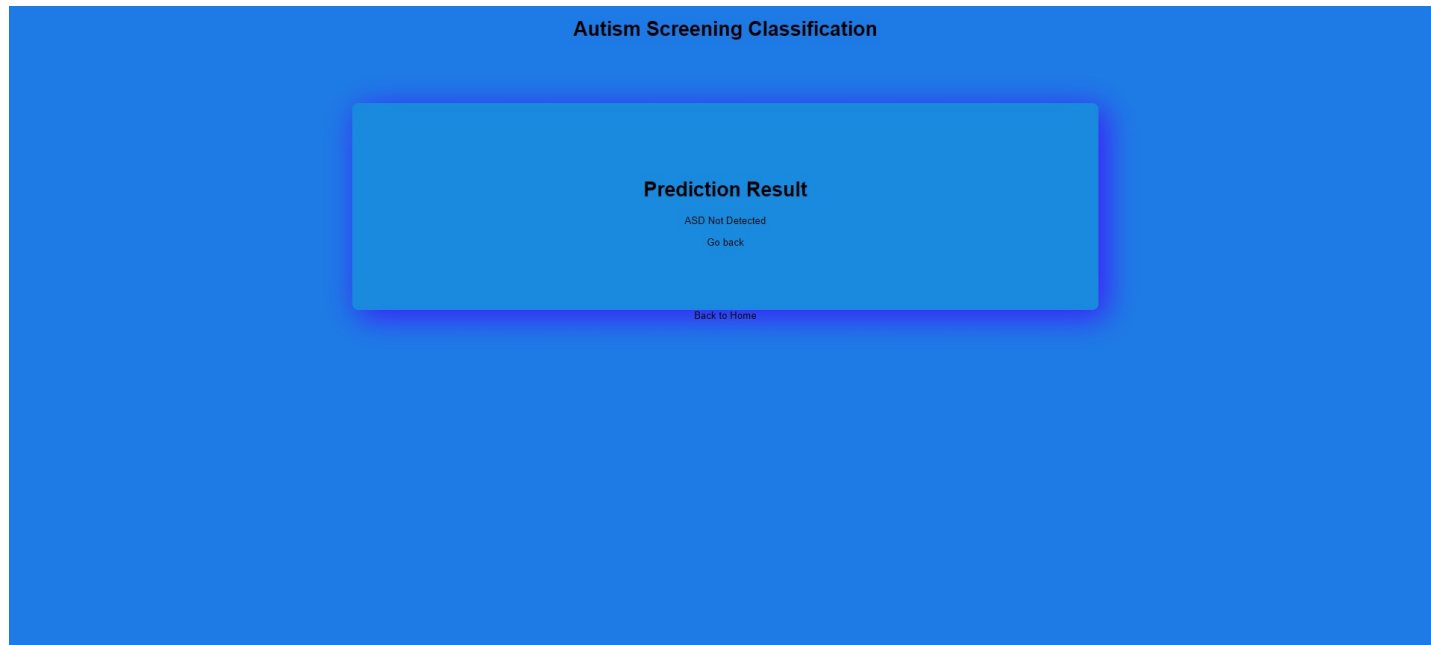
Gender (Male=1, Female=0)

Had Neonatal Jaundice (Yes=1, No=0)

Relative with Autism (Yes=1, No=0)

Predict

Output :



## 7. Advantages & Disadvantages:

### **Advantages:**

1. **Robust Performance:** This Random Forest model gives high accuracy and reliability by taking an aggregate for the predictions of multiple decision trees, hence reducing overfitting and variance.
2. **Handling Complex Relationships:** It effectively handles nonlinear relationships and interaction among features, which is very important to handle complex classification tasks such as autism spectrum disorder detection.
3. **Feature Importance:** It can be used to identify and rank different features by their importance to help in interpretation as to what factors influence classification the most.
4. **Flexibility:** Works well with categorical and numerical data, quite versatile for various types of input data.

### **Disadvantages:**

1. **Complexity:** Depending on the number of trees and features, it could become rather computationally intensive, hence longer training and prediction times.
2. **Interpretability:** Random Forests provide feature importance but are still difficult to interpret as models compared to simple models like decision trees, making it harder to understand the process of decision making.
3. **Memory Usage:** Storing multiple decision trees can be quite memory-consuming. This might create issues in case of large datasets or small computational resources.
4. **Overfitting Risk:** Although RF reduces overfitting risk compared to individual decision trees, they can still overfit in case of lack of tuning or a very noisy dataset.

## 8. Conclusion

The random forest classifier has also done well in autism spectrum disorder classification. Its resilient performance is because of its ability to manage complex and nonlinear relationships and interactions between features, hence making the technique very vital for ASD detection. By aggregating predictions from multiple decision trees, this model reduces over-fitting and enhances accuracy, making it suitable for the task at hand. Therefore, even with the challenges in reduced interpretability and computational complexity, this model of random forest has the edge as a result of its high accuracy and insight derived from feature importance, thus establishing it as a reliable and valuable tool in this classification problem.

## 9. Future Scope

- **Incorporate Additional Data:** Integrate additional variables such as genetic data, longitudinal behavioral assessments, and environmental factors to enhance the model's prediction accuracy for Autism Spectrum Disorder.
- **Explore Advanced Models:** Investigate more sophisticated models, such as Gradient Boosting Machines or Neural Networks, to potentially improve classification performance and handle more complex patterns in the data.
- **Real-Time Predictions:** Develop and deploy systems for real-time data integration and ASD classification to provide timely assessments and support in clinical settings.
- **Model Interpretability:** Enhance model interpretability with techniques like SHAP or LIME to better understand feature impacts and improve decision-making.
- **Automated Retraining:** Implement automated systems for model retraining and updates to adapt to new data and evolving patterns, ensuring continued accuracy and relevance in ASD classification.

## 10. Appendix

- a. Source Code:
- b. GitHub Link : <https://github.com/ChinmayDandekar7/Detection-of-Autistic-Spectrum-Disorder-Classification.git>
- c. Project demo Link : [https://drive.google.com/file/d/1-TU9QyOzBOMyav\\_7XNyab7lpgvz33Qio/view?usp=sharing](https://drive.google.com/file/d/1-TU9QyOzBOMyav_7XNyab7lpgvz33Qio/view?usp=sharing)

