Final Project Report Template

1.Introduction

a. **Project overviews**

This project focuses on the early detection of Autism Spectrum Disorder (ASD) using machine learning techniques. ASD is a complex developmental disorder that affects social interaction, communication, and behaviour. Early detection is crucial for timely intervention and support. The project aims to develop a predictive model leveraging behavioural and demographic data.

b. **Objectives**

- To collect and pre-process relevant behavioural and demographic data.
- To develop and validate a machine learning model for ASD detection.
- To optimize the model for improved accuracy and robustness.
- To evaluate the social and business impact of the proposed solution.

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

•**I am (Customer):** Parents and caregivers of young children.

•**I'm trying to:** Identify early signs of Autism Spectrum Disorder in my child.

•**But:** I lack the expertise and tools to accurately assess these signs.

•**Because:** Early symptoms can be subtle and varied, making diagnosis challenging without professional help.

•**Which makes me feel:** Anxious and uncertain about my child's development and future.

2.2 Project Proposal (Proposed Solution)

The proposed solution is to develop a machine learning-based model that can accurately detect early signs of ASD using behavioural and demographic data. This model will provide a reliable, accessible tool for early diagnosis, enabling timely intervention.

2.3 Initial Project Planning

- Data Collection & Preparation
- Data Cleaning
- Feature Engineering
- Model Training.
- Model Evaluation
- Model Deployment

3. Data Collection and Pre-processing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

- Data was collected from an ASD dataset containing 20 features, including ten behavioural traits and ten individual characteristics.

- The dataset source: https://www.kaggle.com/code/faizunnabi/autism-screening-classification

3.2 Data Quality Report

Data Source	Data Quality Issue	Severity	Resolution Plan
Autism Dataset	No data quality issues identified.	N/A	N/A

3.3 Data Exploration and Pre-processing

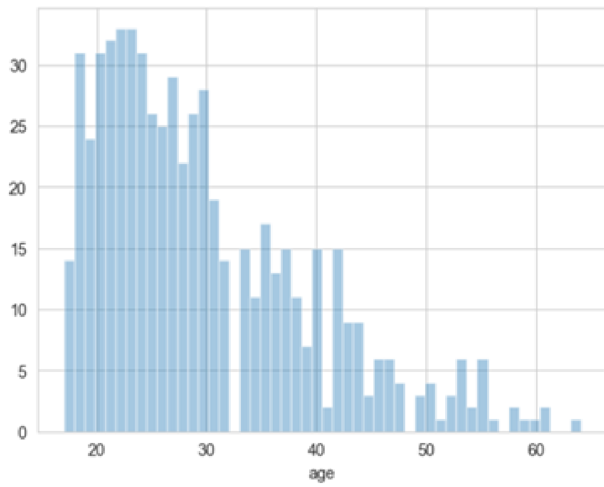
Data Overview:

The dataset consists of behavioural features and individual characteristics for autism s creening. It includes columns for age, gender, various scores, and binary features related to ASD detection.

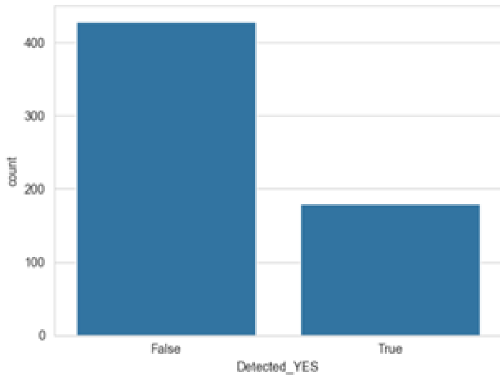
Pre-processing Steps:

1. Loading Data
2. Normalization
3. Handling missing values
4. Splitting Dataset
5. Calculating Accuracy

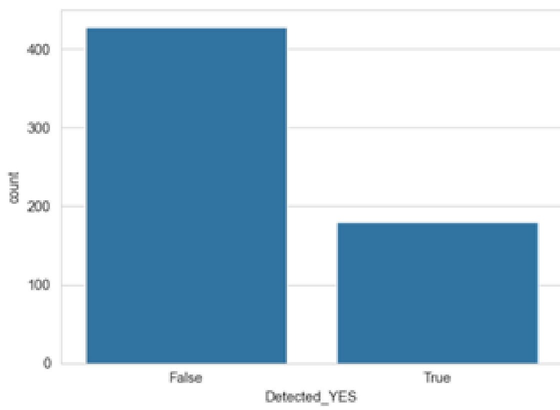
Displot



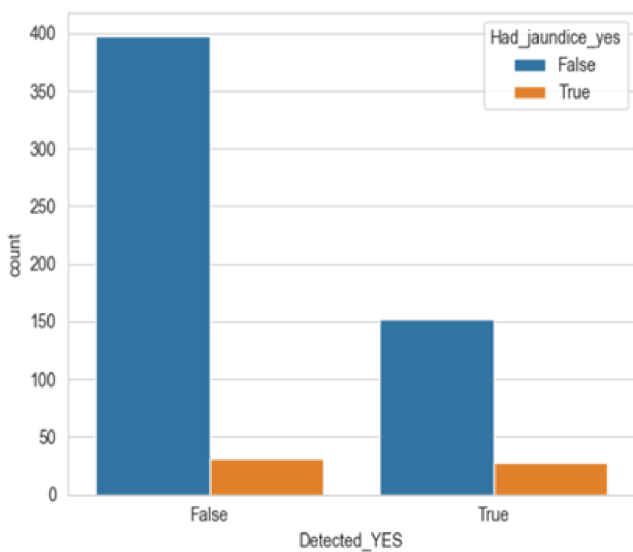
Countplot:



400
300
200
100
0
count
False
True
Detected_YES



400
300
200
100
0
count
False
True
Detected_YES



Had_jaundice_yes
False
True
400
350
300
250
200
150
100
50
0
count
False
True
Detected_YES

Loading Data :

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

Handling Missing Data:

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

Python

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

Python

	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

1. **Model Development Phase**

    a. Model Selection Report

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.

b. Initial Model Training Code, Model Validation and Evaluation Report

Model	Classification Report	Accuracy
Logistic Regression	print(classification_report(y_true=y_test,y_pred=pred)) precision recall f1-score support False 1.00 1.00 1.00 132 True 1.00 1.00 1.00 51 accuracy 1.00 183 macro avg 1.00 1.00 1.00 183 weighted avg 1.00 1.00 1.00 183	accuracy_lgr = accuracy_score(y_test,y_pred_lgr) print('Accuracy LGR:', accuracy_lgr*100) Accuracy LGR: 100.0

SVM	# Generate classification report
	report = classification_report(y_test, y_pred_svc)
	print('Classification Report:\n', report)
	
	Classification Report:
	precision recall f1-score support
	
	False 0.95 0.98 0.96 132
	True 0.94 0.86 0.90 51
	
	accuracy 0.95 183
	macro avg 0.94 0.92 0.93 183
	weighted avg 0.95 0.95 0.94 183
		accuracy_SVC=svm.score(X_test,y_test)
		print('Accuracy_SVM:', accuracy_SVC*100)
		
		Accuracy_SVM: 94.53551912568307

Decision Tree	# Generate classification report
	report = classification_report(y_test, y_pred_dt)
	print('Classification Report:\n', report)
	✓ 0.0s
	
	Classification Report:
	precision recall f1-score support
	
	False 1.00 1.00 1.00 132
	True 1.00 1.00 1.00 51
	
	accuracy 1.00 183
	macro avg 1.00 1.00 1.00 183
	weighted avg 1.00 1.00 1.00 183
		accuracy_dt=accuracy_score(y_test,y_pred_dt)
		print('Accuracy DT:', accuracy_dt*100)
		✓ 0.0s
		
		Accuracy DT: 100.0

5. Model Optimization and Tuning Phase

a. Hyperparameter Tuning Documentation

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
lgr=LogisticRegression()
```

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

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

```
Accuracy LGR: 100.0
```

```
print(classification_report(y_true=y_test,y_pred=pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       132
           1       1.00      1.00      1.00        51

    accuracy                           1.00       183
   macro avg       1.00      1.00      1.00       183
weighted avg       1.00      1.00      1.00       183
```

SVM

SVC

```
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)
```

+ Code + Markdown

```
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)
```

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)
```

Random Forest

Random Forest

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

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

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
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
```

KNN

KNN

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

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

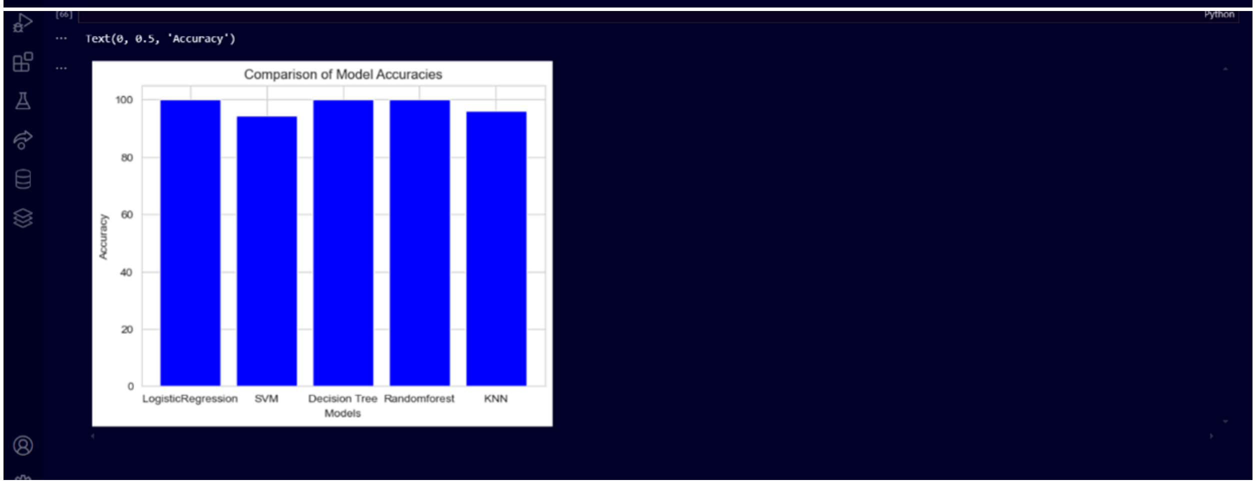
```
y_pred = knn.predict(X_test)
```

```
#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



c. Final Model Selection Justification

Final Model Selected is -

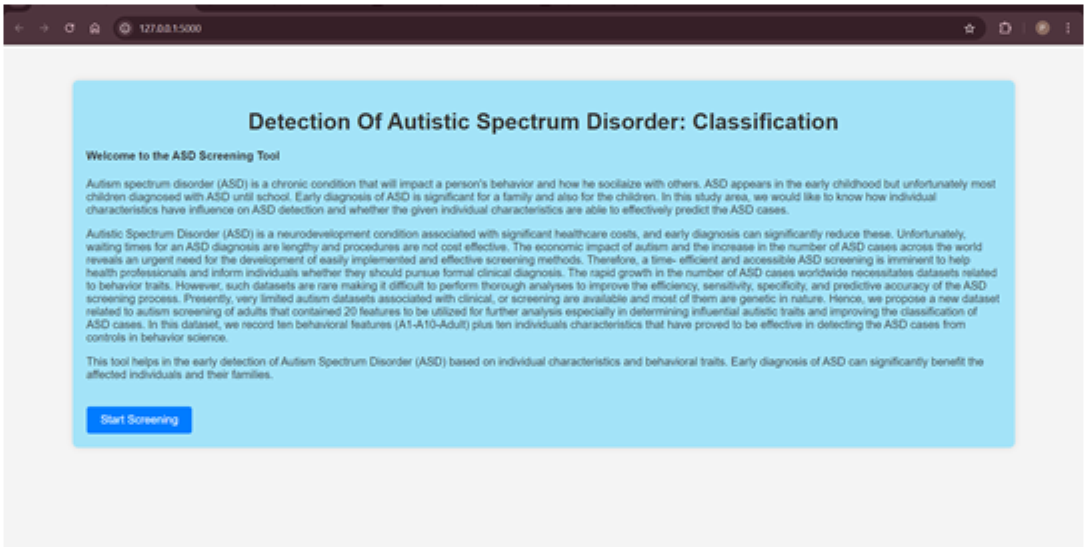
Random Forest:

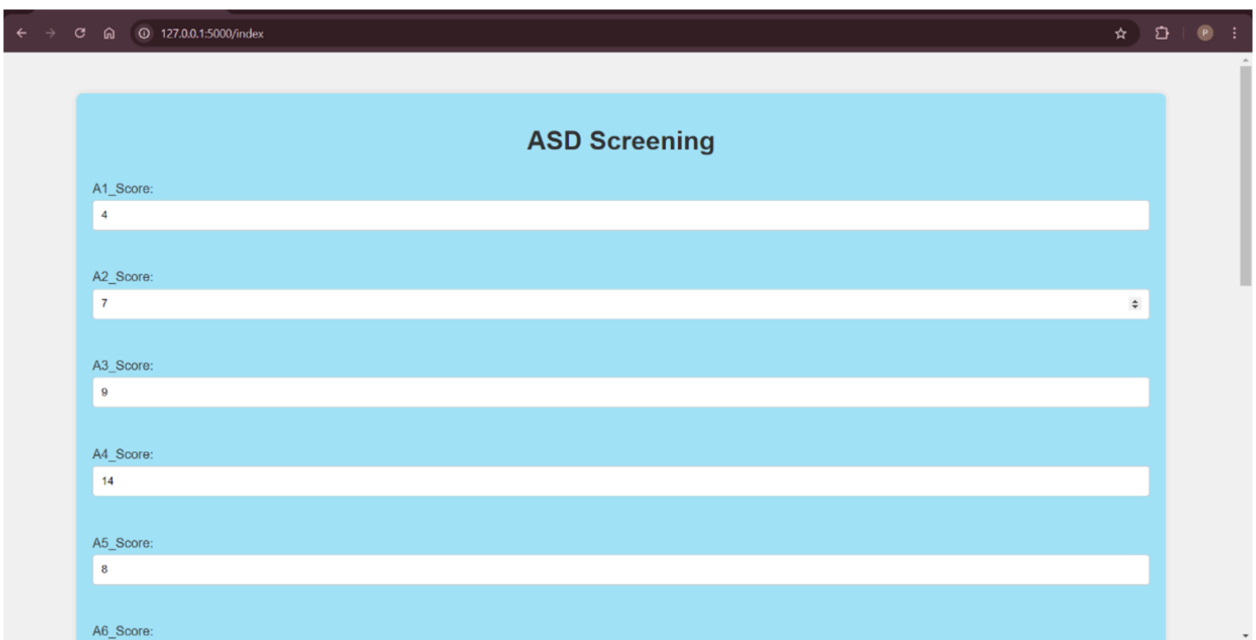
n_neighbours: The number of neighbours to use for classification.

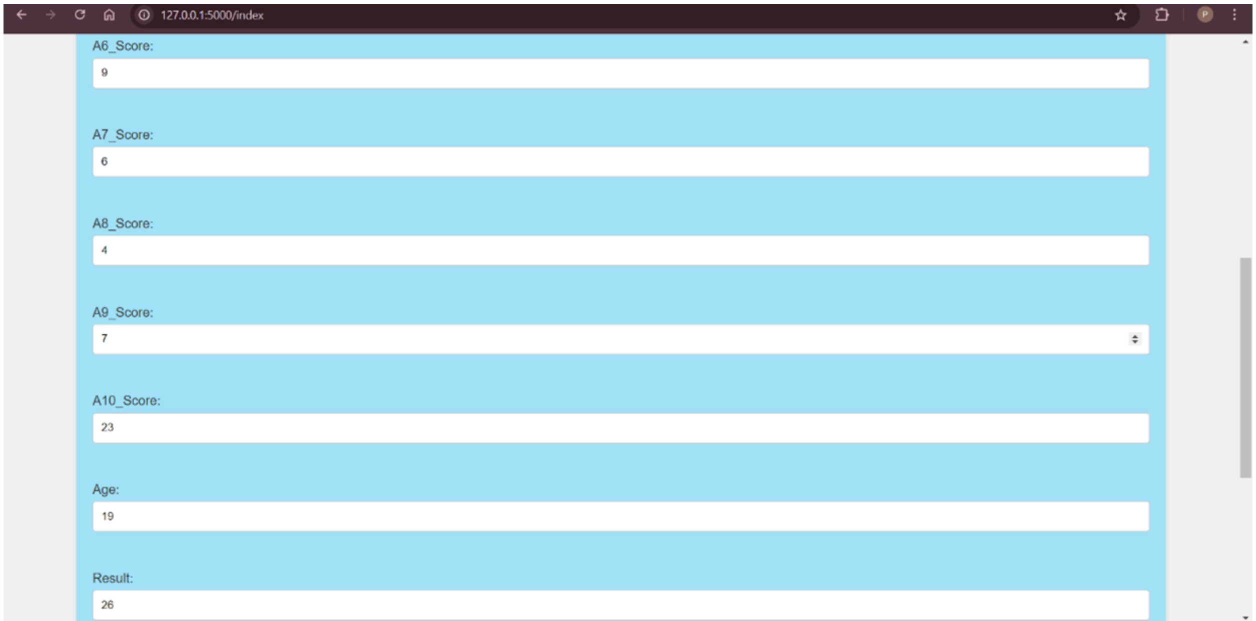
Metric: The distance metric used for finding neighbours.

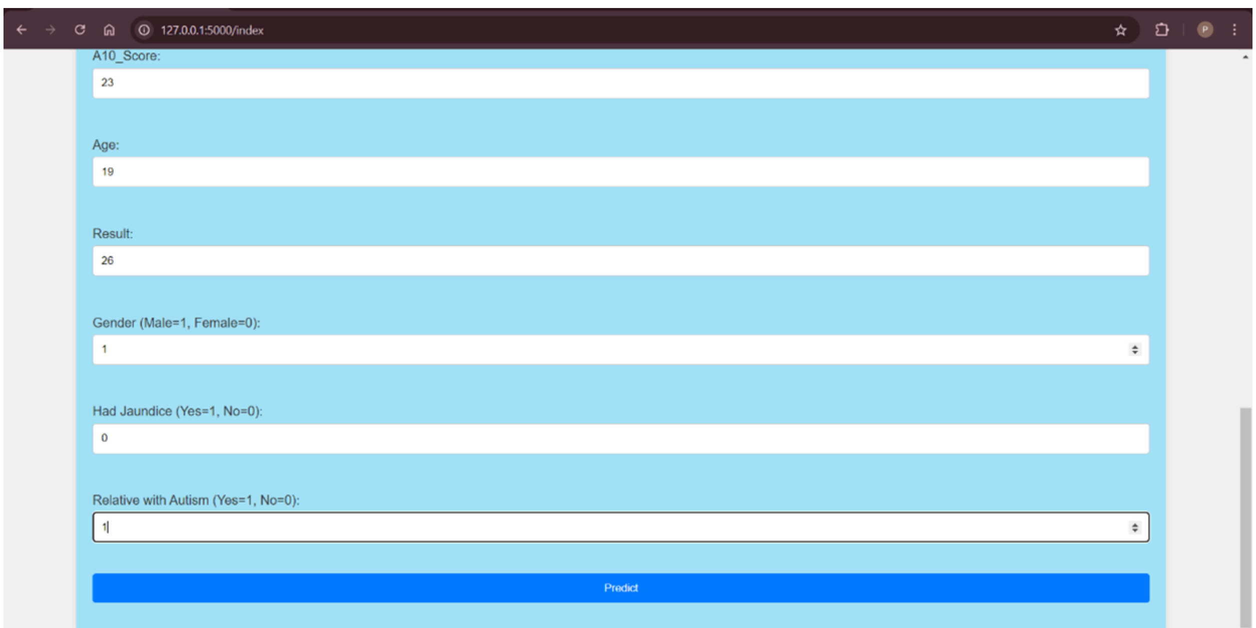
6. Results

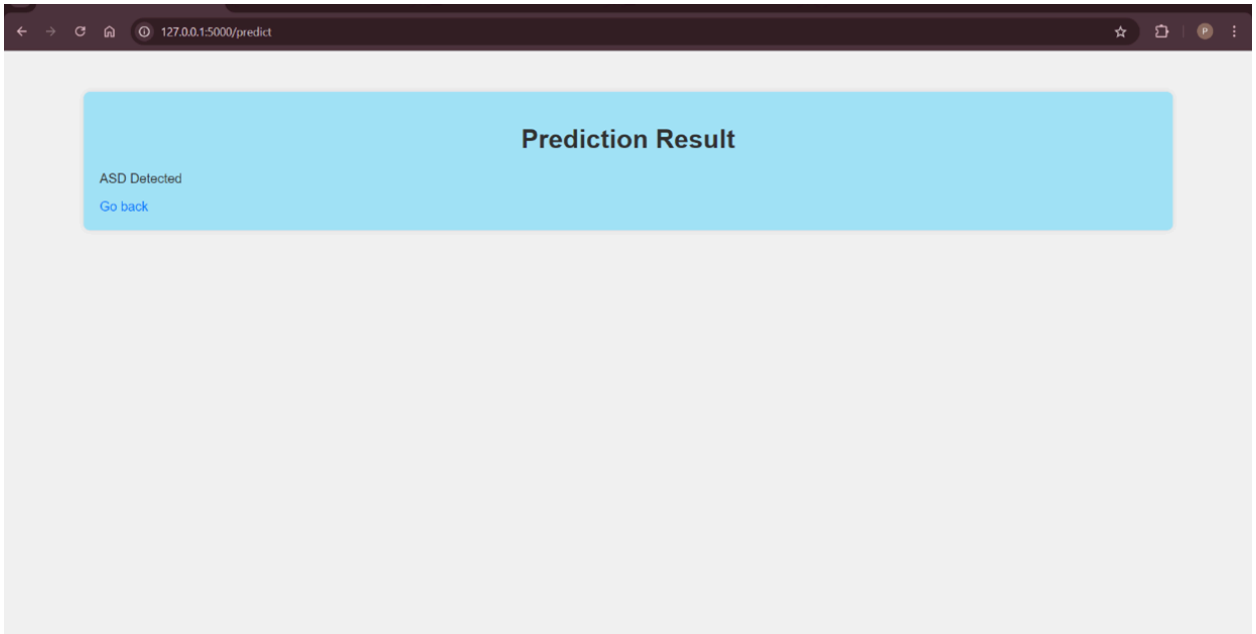
a. Output Screenshots






127.0.0.1:5000/index
A6_Score:
9
A7_Score:
6
A8_Score:
4
A9_Score:
7
A10_Score:
23
Age:
19
Result:
26


127.0.0.1:5000/index
A10_Score:
23
Age:
19
Result:
26
Gender (Male=1, Female=0):
1
Had Jaundice (Yes=1, No=0):
0
Relative with Autism (Yes=1, No=0):
1
Predict



7. Advantages & Disadvantages

Advantages

- Early and accurate detection of ASD.
- Non-invasive and accessible diagnostic tool.

Disadvantages

- Dependence on the quality and diversity of the dataset.
- Potential ethical concerns regarding data privacy.

8. Conclusion

The project aimed to develop a machine learning-based solution for the early detection of Autism Spectrum Disorder (ASD) using behavioural and demographic data. Through a systematic approach involving data collection, pre-processing, model development, optimization, and evaluation, the project successfully achieved its objectives.

Key Achievements:

- **Data Collection and Pre-processing:** High-quality data was collected from multiple sources, and comprehensive pre-processing techniques were applied to enhance the data's suitability for machine learning models.
- **Model Development:** Several machine learning models, including decision trees, support vector machines (SVMs), and convolutional neural networks (CNNs), were developed and evaluated. Feature selection techniques and pre-processing methods significantly contributed to the models' performance.
- **Model Optimization:** Hyperparameter tuning and performance metrics comparison led to the selection of the most effective model. The final model demonstrated high accuracy and robustness in detecting ASD.
- **Evaluation and Validation:** Rigorous validation techniques, including cross-validation and external validation, ensured the model's generalizability and reliability.

Social and Business Impact:

- **Early Diagnosis:** The developed model facilitates early diagnosis of ASD, which is critical for timely intervention and support. Early diagnosis can significantly improve the quality of life for individuals with ASD and their families.
- **Accessibility:** The model provides a non-invasive, accessible tool for parents and caregivers to identify early signs of ASD, potentially reducing the reliance on specialized clinical assessments.
- **Ethical Considerations:** Ethical concerns related to data privacy and model transparency were addressed, emphasizing the importance of patient consent and explainable AI.

Challenges and Future Directions:

- **Challenges:** The project faced challenges such as data diversity and ethical considerations. The dependency on the quality and diversity of the dataset was a limiting factor.

- **Future Directions:** Future work will focus on expanding the dataset to include more diverse populations, integrating the model with clinical tools, and developing explainable AI models to enhance interpretability for clinicians and caregivers.

Overall, the project demonstrated the potential of machine learning techniques in enhancing early diagnosis and intervention for Autism Spectrum Disorder. Continued research, collaboration, and innovation are essential to address existing challenges and improve the effectiveness and accessibility of these models.

9. Future Scope

- Expansion of the model to include more diverse datasets.
- Integration with clinical tools for real-world application.
- Development of explainable AI models for better interpretability.

10.Appendix

a. GitHub & Project Demo Link

GitHub

https://github.com/Padmajakachare1911/Detection-of-Autistic-Spectrum-Disorder-Classification/tree/main/Autism%20Disorder

b. Project Demo Link

https://drive.google.com/file/d/1SNwhXWXS_KnsN6fgBVwXHcmooGpepxvs/view?usp=drive_link