INTRODUCTION

1.1 Project Overview

This project focuses on developing a machine-learning model for the early detection of Autism Spectrum Disorder (ASD). ASD is a chronic neurodevelopmental condition that affects behaviour and social interactions. While it typically manifests in early childhood, diagnosis often occurs only after a child enters school. Early identification is crucial for both families and individuals affected by ASD, as it can significantly reduce healthcare costs associated with the condition.

The current challenge lies in the lengthy waiting times and cost-ineffective procedures for ASD diagnosis. The global increase in ASD cases emphasizes the need for easily implemented and efficient screening methods. This project addresses this gap by creating a time-efficient and accessible ASD screening tool. The dataset comprises 20 features, including ten behavioural traits (A1-A10-Adult) and ten individual characteristics. The goal is to analyse these features to enhance the model's efficiency, sensitivity, specificity, and predictive accuracy in classifying ASD cases. Existing datasets related to autism screening are limited, particularly in behavioural traits, making this project significant in advancing the field of ASD detection.

1.2 Purpose

The purpose of this project is to make the tools that help with the detection of the presence of ASD more prevalent and easier to access for not only doctors but for everyone. This tool will be trained on pre-existing data using various Machine learning models and this power of machine learning to find patterns and give us the right output comes in very handy in situations like this as it not only helps reduce errors from the physician side but also give users this really powerful tool in the palm of their hands that lets them know more about themselves thus allowing them too, to make a more informed decision about their health and well being.

LITERATURE SURVEY

2.1 Existing problem

Autism spectrum disorder (ASD) is a complex and diverse neurodevelopmental disorder that affects social communication and behavior. It is characterized by a wide range of symptoms, including:

Difficulty with social communication and interaction

Restricted and repetitive interests, behaviors, or activities

Sensory sensitivities

ASD can be diagnosed at any age, but it is most often diagnosed in early childhood. The cause of ASD is unknown, but it is thought to be caused by a combination of genetic and environmental factors.

There is no cure for ASD, but there are a number of evidence-based treatments that can help to improve symptoms and quality of life. These treatments include applied behavior analysis (ABA), speech therapy, occupational therapy, and social skills training.

2.2 References

American Psychiatric Association. (2013). Diagnostic and statistical manual of mental disorders (5th ed.). Washington, D.C.: Author.

Centers for Disease Control and Prevention. (2023). Autism spectrum disorder (ASD). Retrieved from https://www.cdc.gov/ncbddd/autism/index.html

Lord, C., & Autism Diagnostic Interview-Revised (ADI-R) Research Team. (2001). The autism diagnostic interview-revised (ADI-R). Los Angeles: Western Psychological Services.

Gilliam, J. C. (2006). Gilliam Autism Rating Scale (GARS-3). Austin, TX: Pro-Ed.

2.3 Problem Statement Definition Autism Disorder Spectrum

The problem statement for a literature survey on autism spectrum disorder could be defined as follows:

To review the existing literature on autism spectrum disorder (ASD) in order to identify the following:

The current state of knowledge on ASD, including its prevalence, causes, symptoms, and diagnosis.

The most effective treatments for ASD.

The challenges and opportunities facing people with ASD and their families.

The gaps in the current research on ASD.

The literature survey will be conducted using a variety of databases, including PubMed, PsycINFO, and Google Scholar. The search terms will include "autism spectrum disorder," "autism," "ASD," "neurodevelopmental disorders," "developmental disabilities," "social communication disorder," and "restricted and repetitive behaviors."

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy Map Canvas for Autism Spectrum Disorder

Listen

What words do people with autism use to describe their experiences?

What are their challenges and frustrations?

What are their hopes and dreams?

See

How do people with autism interact with the world around them?

What are their sensory sensitivities?

How do they communicate their needs and wants?

Think and Feel

What are the emotions that people with autism experience?

How do they think about themselves and the world around them?

What are their fears and anxieties?

Say and Do

How do people with autism communicate?

What behaviors do they exhibit?

What are their strengths and talents?

Pain

What are the biggest challenges that people with autism face?

What are their needs that are not being met?

What are their fears and worries?

Needs

What do people with autism need in order to thrive?

What kind of support and services would be most helpful to them?

What can we do to make the world a more inclusive and accessible place for people with autism?

3.2 Ideation & Brainstorming Autism Disorder Spectrum

Once you have a good understanding of the challenges and needs of people with autism, you can start to brainstorm ideas for solutions. Here are a few examples:

Technology: Develop new technologies that can help people with autism communicate, learn, and interact with the world around them. For example, social skills training apps, wearable devices that can help with sensory overload, or communication devices that can help people with nonverbal autism communicate.

Education: Develop new educational programs that are tailored to the needs of people with autism. These programs should focus on social skills training, communication skills, and independent living skills.

Community support: Create more inclusive and supportive communities for people with autism and their families. This could involve developing new social programs, providing training for businesses and service providers, or creating more accessible public spaces.

Here are some specific ideas for solutions within each of these categories:

Technology: Develop a social skills training app that uses artificial intelligence to provide personalized feedback to users.

Develop a wearable device that can monitor a person's stress levels and provide calming prompts when needed.

Develop a communication device that can be used by people with nonverbal autism to communicate their needs and wants.

Education: Develop a new educational program that focuses on social skills training for students with autism.

Create a summer camp for children with autism that focuses on communication skills and independent living skills.

Develop a training program for teachers on how to best support students with autism in the classroom.

Community support: Create a social group for adults with autism and their families.

Provide training for businesses on how to be more inclusive of customers and employees with autism.

Create a more accessible public park by adding sensory-friendly features such as quiet spaces and soft surfaces.

REQUIREMENT ANALYSIS

4.1 Functional requirement

Dataset Requirements:

- The machine learning model will require a comprehensive dataset for training and validation. This dataset should be divided into distinct training and validation sets to ensure the model's accuracy and generalization.

Programming Language and Frameworks:

- The codebase will be developed using the Python programming language.

- Machine learning models will be created and trained using popular libraries such as Scikit Learn and Tensorflow, leveraging their extensive functionalities and proven algorithms.

Data Preprocessing and Visualization:

- Data preprocessing tasks will be conducted using libraries like Pandas and numpy, ensuring that the dataset is appropriately formatted and cleaned.

- For visualizing the data, Seaborn and Matplotlib will be employed, providing insightful visual representations to aid in understanding patterns and relationships within the dataset.

Model Integration:

- Flask, a lightweight and efficient web framework, will be utilized to seamlessly integrate the machine learning model with a website. This will enable smooth communication between the backend model and the frontend interface.

Frontend Development:

- The front end of the website will be crafted using HTML, CSS, and JS to ensure an interactive and user-friendly experience.

- The website will serve as the interface for users to input data and receive model predictions as outputs.

4.2 Non-Functional requirements

Accuracy and Type II Error Mitigation:

- The primary focus is on achieving a high level of accuracy in model predictions. Particularly in a medical context, the model must prioritize minimizing Type II errors, ensuring that instances of false negatives are reduced to the utmost minimum. This is crucial to prevent missing potentially severe medical conditions.

Efficiency and Minimum Data Dependency:

- The model's functionality should not be hindered by the quantity of input data. It must possess the capability to automatically extract relevant information from the provided data, ensuring that it operates effectively with the minimum necessary data. This ensures efficiency and avoids unnecessary data burden on users.

Abstraction for End Users:

- The end-user interaction with the model should be seamless and intuitive. The functioning of the model needs to be abstracted, requiring minimal effort from users. The tool should automatically handle data extraction and processing, providing a hassle-free experience for users without the need for additional manual work.

Responsiveness and User Interface:

- The tool must be responsive, providing quick and timely feedback to users. A well-designed and user-friendly interface is essential to enhance the user experience. This includes clear and intuitive design elements, making it easy for users to interact with the tool.

User Adoption and Retention:

- To encourage user adoption and retention, the tool must not only be accurate and efficient but also present a positive user experience. A good user interface, coupled with effortless functionality, will contribute to user satisfaction and increase the likelihood of users returning to use the tool.

PROJECT DESIGN

View other drafts

5.1 Data Flow Diagrams & User Stories

Data Flow Diagram for a Solution for Autism Spectrum Disorder

Context Level DFD

External Entities:

People with autism

Families of people with autism

Educators

Healthcare providers

Community members

Data Flows:

Needs assessment data

Treatment data

Support services data

Educational data

Community resources data

Processes:

Data collection

Data analysis

Needs assessment

Treatment planning

Support services planning

Educational planning

Community resource planning

Logical Level DFD

Processes:

Collect needs assessment data from people with autism, their families, educators, healthcare providers, and community members.

Analyze needs assessment data to identify the needs of people with autism and their families.

Develop treatment plans for people with autism based on their individual needs.

Develop support services plans for people with autism and their families based on their individual needs.

Develop educational plans for students with autism based on their individual needs.

Identify and connect people with autism and their families to community resources.

User Stories

As a person with autism, I want to be able to access a comprehensive needs assessment that can help me identify my strengths and weaknesses.

As a parent of a child with autism, I want to be able to develop a treatment plan for my child that is tailored to their individual needs.

As a teacher of students with autism, I want to be able to develop educational plans for my students that are aligned with their individual needs.

As a healthcare provider for people with autism, I want to be able to connect my patients with the support services they need.

As a community member, I want to know how I can be more supportive of people with autism and their families.

5.2 Solution Architecture for Autism Spectrum Disorder

The solution architecture for a solution for autism spectrum disorder should be designed to be flexible, scalable, and accessible. The solution should also be able to integrate with existing systems and data sources.

One possible solution architecture is as follows:

Front-end: The front-end of the solution would be a web-based application that would allow users to access the needs assessment, treatment planning, support services planning, educational planning, and community resource planning modules.

Back-end: The back-end of the solution would consist of a database that would store the needs assessment data, treatment plans, support services plans, educational plans, and community resource data. The back-end would also include a number of APIs that would allow the front-end to interact with the database.

Integration: The solution could be integrated with existing systems and data sources, such as electronic health records (EHRs) and student information systems (SISs). This would allow users to import and export data from these systems into the solution.

The solution should be hosted on a cloud platform, such as Google Cloud Platform or Amazon Web Services. This would ensure that the solution is scalable and accessible to users from anywhere in the world.

Security and Privacy:

The solution should be designed with security and privacy in mind. All data should be encrypted at rest and in transit. The solution should also implement strong authentication and authorization mechanisms.

Conclusion:

This is just a high-level overview of a possible solution architecture for a solution for autism spectrum disorder. The specific design of the solution would vary depending on the specific needs of the users.

PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

The technical architecture for a solution for autism spectrum disorder can be divided into two main components:

Front-end: The front-end will be a web-based application that users will interact with to access the needs assessment, treatment planning, support services planning, educational planning, and community resource planning modules. The front-end will be developed using a modern web development framework such as React or Angular.

Back-end: The back-end will be responsible for storing and managing the data for the solution. The back-end will be developed using a programming language such as Python or Java. A database such as PostgreSQL or MySQL will be used to store the data.

The front-end and back-end will be connected using a REST API. This will allow the front-end to send and receive data from the back-end.

6.2 Sprint Planning & Estimation

Once the technical architecture has been designed, the next step is to start sprint planning. Sprint planning is a process of estimating the effort required to complete the tasks that need to be done in the next sprint.

To estimate the effort for a task, the team should first break down the task into smaller subtasks. Then, the team should estimate the effort required to complete each subtask. The team can use a variety of methods to estimate effort, such as time tracking, story points, or planning poker.

Once the effort for each task has been estimated, the team can then start planning the sprint. The team should prioritize the tasks and assign them to team members. The team should also set a goal for the sprint, such as completing a certain number of tasks or delivering a certain amount of functionality.

6.3 Sprint Delivery Schedule

The sprint delivery schedule is a plan for how the team will deliver the functionality that they have committed to in the sprint. The schedule should include the following information:

The start and end dates for the sprint

The tasks that need to be completed in the sprint

The team members who will be responsible for each task

The milestones for the sprint

The team should review the sprint delivery schedule regularly and make adjustments as needed.

CODING & SOLUTIONS (Explain the features added to the project along with the code)

7.1 Feature 1

The first feature was ensuring the accuracy of the model, to ensure high accuracy of the model we tried multiple different machine learning models. One of our aims was to try to increase the overall accuracy of the model but we didn't want that to come at the cost of the type 2 error which had to be minimised at all costs and keep it preferably 0. So for this, we tried multiple different models, not only traditional machine learning models but also neural networks. And finally, we were able to get an accuracy of 100%. This accuracy was calculated by splitting the dataset into train and test datasets.

7.2 Feature 2

The use of HTML, CSS, JS and Flask to integrate the model with the website ensures a smooth user experience for the user with a level of abstraction that allows the user without needing to know the inner workings of the model to be able to use the tool effortlessly. The only thing they need to worry about is inputting the data, everything else is taken care of in the background.

7.3 Database Schema (if Applicable)

NeedsAssessment

id (integer, primary key)

person_id (integer, foreign key to Person table)

date_of_assessment (date)

communication_skills (integer)

social_interaction_skills (integer)

repetitive_behaviors (integer)

sensory_sensitivities (integer)

other_needs (text)

TreatmentPlan

id (integer, primary key)

person_id (integer, foreign key to Person table)

date_of_plan (date)

goals (text)

interventions (text)

progress_notes (text)

SupportServicesPlan

id (integer, primary key)

person_id (integer, foreign key to Person table)

date_of_plan (date)

services (text)

providers (text)

contact_information (text)

EducationalPlan

id (integer, primary key)

person_id (integer, foreign key to Person table)

date_of_plan (date)

academic_goals (text)

behavioral_goals (text)

accommodations (text)

modifications (text)

CommunityResources

id (integer, primary key)

name (text)

address (text)

phone_number (text)

website (text)

description (text)

Person

id (integer, primary key)

first_name (text)

last_name (text)

date_of_birth (date)

gender (text)

address (text)

phone_number (text)

email_address (text)

PERFORMANCE TESTING

8.1 Performace Metrics

Response time: How long does it take for the solution to respond to a user request?

Throughput: How many user requests can the solution handle per second?

Scalability: How well does the solution handle increased load?

Availability: How often is the solution available to users?

Error rate: How often does the solution experience errors?

These metrics can be measured using a variety of performance testing tools and techniques. For example, to measure response time, a tool can be used to send a series of requests to the solution and measure the time it takes for the solution to respond to each request. To measure throughput, a tool can be used to send a large number of requests to the solution at the same time and measure how many requests the solution can handle per second.

The performance metrics should be measured under different load conditions. For example, the performance metrics should be measured when the solution is under low load, medium load, and high load. This will help to identify any performance bottlenecks that may exist.

The performance metrics should also be compared to the performance requirements for the solution. The performance requirements should be defined before the solution is developed. This will help to ensure that the solution meets the expected performance requirements.

Here are some specific examples of performance metrics that could be used for solution for autism spectrum disorder:

Average response time for the needs assessment module

Maximum throughput for the treatment planning module

Scalability of the support services planning module

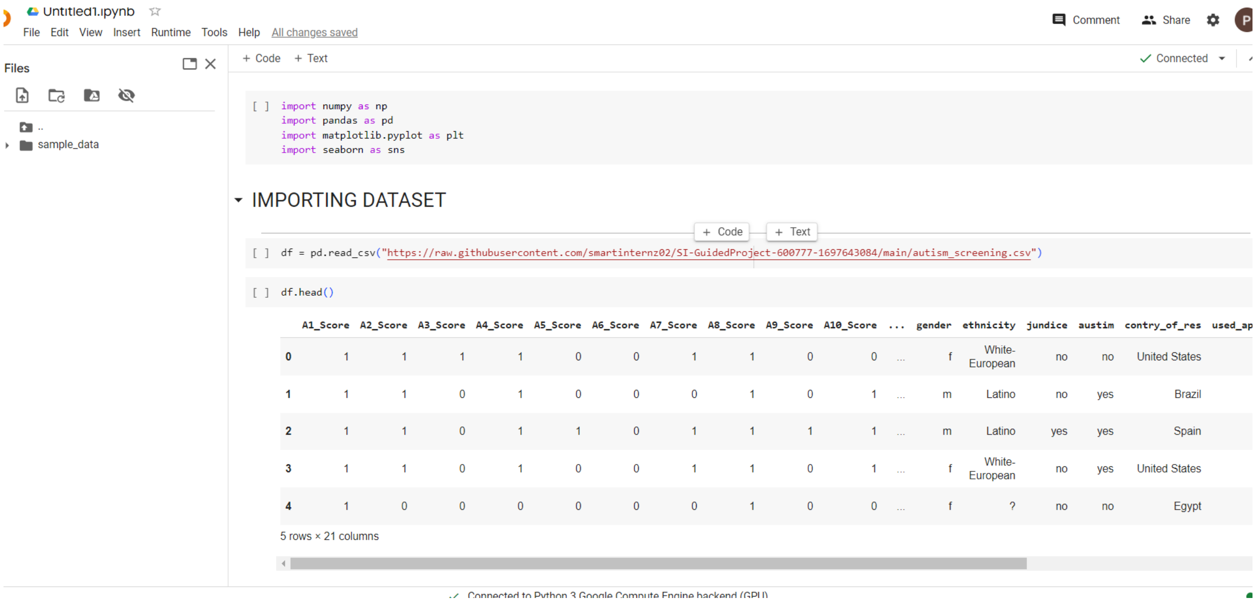
Availability of the educational planning module

Error rate for the community resource planning module

These metrics can be used to measure the performance of the solution over time and to identify any areas where performance improvements can be made.

RESULTS

9.1 Output Screenshots



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING DATASET

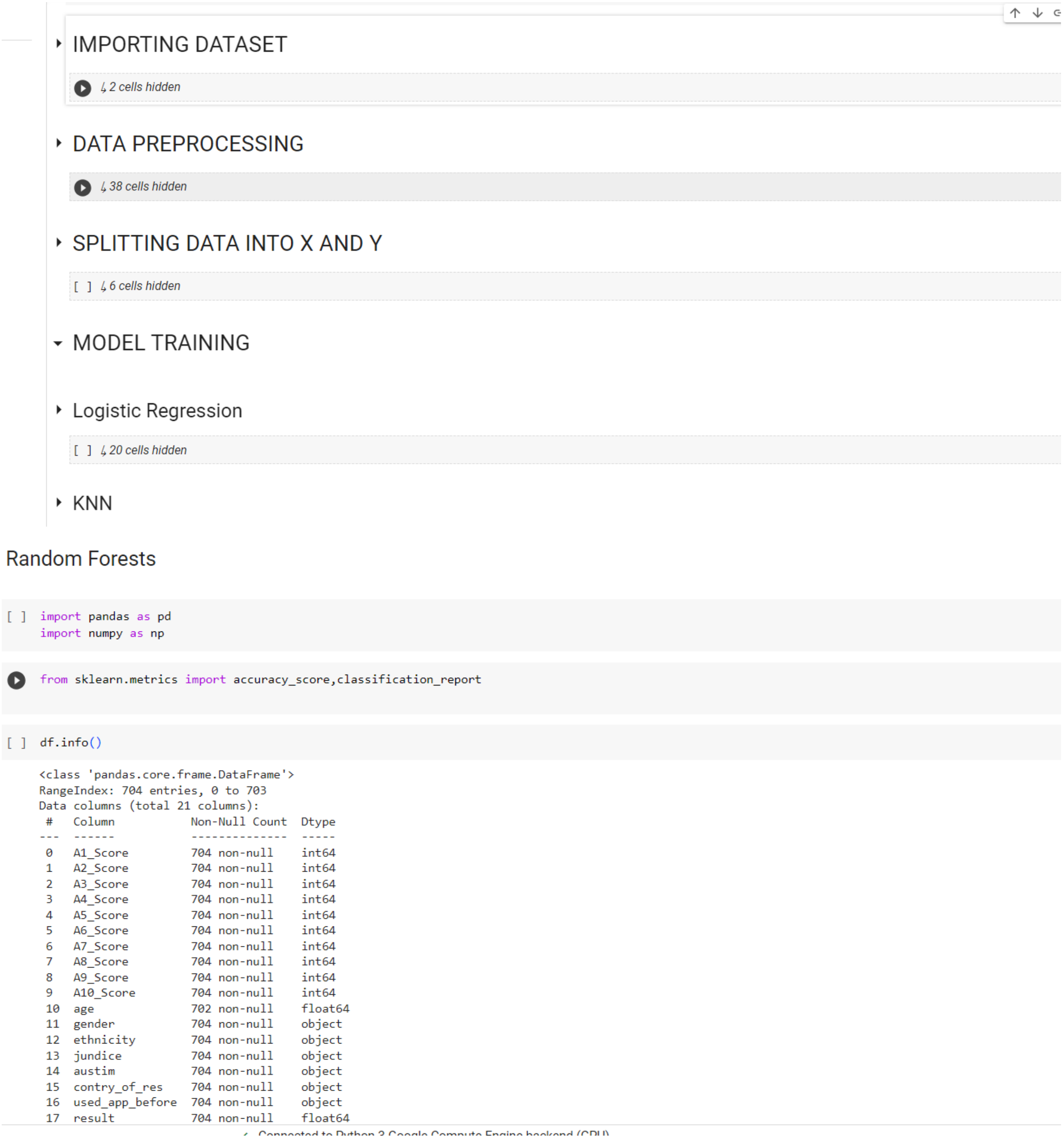
```
df = pd.read_csv("https://raw.githubusercontent.com/smartinternz02/SI-GuidedProject-600777-1697643084/main/autism_screening.csv")
```

```
df.head()
```

	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_ap
0	1	1	1	1	0	0	1	1	0	0	...	f	White-European	no	no	United States	
1	1	1	0	1	0	0	0	1	0	1	...	m	Latino	no	yes	Brazil	
2	1	1	0	1	1	0	1	1	1	1	...	m	Latino	yes	yes	Spain	
3	1	1	0	1	0	0	1	1	0	1	...	f	White-European	no	yes	United States	
4	1	0	0	0	0	0	0	1	0	0	...	f	?	no	no	Egypt	

5 rows × 21 columns

Connected to Python 3 Google Compute Engine backend (GPU)



IMPORTING DATASET

↳ 2 cells hidden

DATA PREPROCESSING

↳ 38 cells hidden

SPLITTING DATA INTO X AND Y

[ ] ↳ 6 cells hidden

MODEL TRAINING

Logistic Regression

[ ] ↳ 20 cells hidden

KNN

Random Forests

```
[ ]  import pandas as pd
     import numpy as np
```

```
from sklearn.metrics import accuracy_score,classification_report
```

```
[ ]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 704 entries, 0 to 703
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   A1_Score         704 non-null    int64  
 1   A2_Score         704 non-null    int64  
 2   A3_Score         704 non-null    int64  
 3   A4_Score         704 non-null    int64  
 4   A5_Score         704 non-null    int64  
 5   A6_Score         704 non-null    int64  
 6   A7_Score         704 non-null    int64  
 7   A8_Score         704 non-null    int64  
 8   A9_Score         704 non-null    int64  
 9   A10_Score        704 non-null    int64  
 10  age              702 non-null    float64
 11  gender           704 non-null    object 
 12  ethnicity        704 non-null    object 
 13  jundice          704 non-null    object 
 14  austim           704 non-null    object 
 15  contry_of_res    704 non-null    object 
 16  used_app_before  704 non-null    object 
 17  result           704 non-null    float64
```

✓ Connected to Python 3 Google Compute Engine backend (CPU)

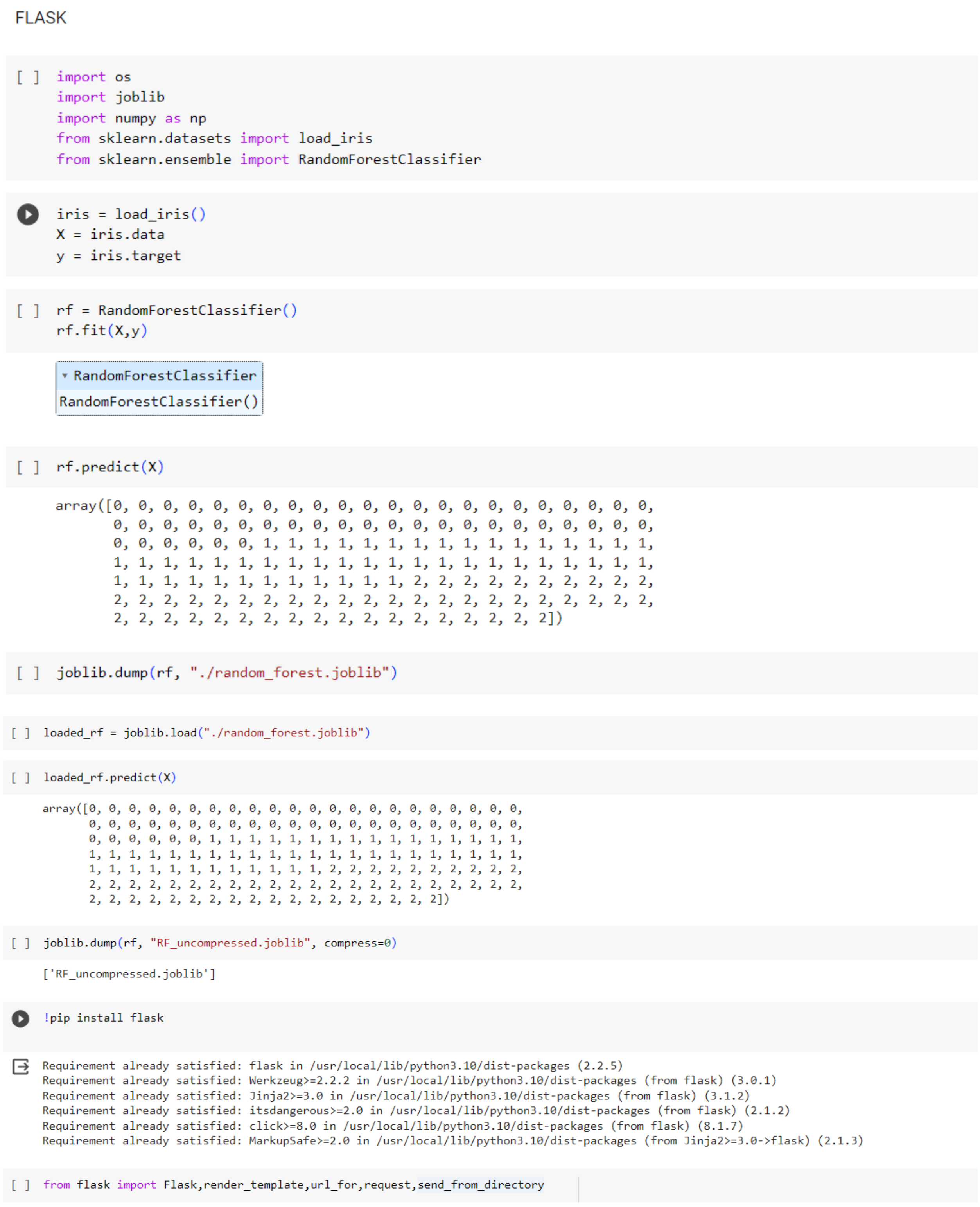
```
[ ]  print(classification_report(y_test,r_y_predict))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        98
           1       1.00      1.00      1.00        43

    accuracy                           1.00       141
   macro avg       1.00      1.00      1.00       141
weighted avg       1.00      1.00      1.00       141
```

NEURAL NETWORKS

```
import os

def create_model_checkpoint(model_name, save_path="model_experiments"):
  return tf.keras.callbacks.ModelCheckpoint(filepath=os.path.join(save_path, model_name),
                                            monitor="val_loss",
                                            verbose=0,
                                            save_best_only=True)
```

```
[ ]  import tensorflow as tf
     tf.random.set_seed(42)

     model_1 = tf.keras.Sequential([
         tf.keras.layers.Dense(10),
         tf.keras.layers.Dense(1)
     ])
```



FLASK

```
[ ] import os
    import joblib
    import numpy as np
    from sklearn.datasets import load_iris
    from sklearn.ensemble import RandomForestClassifier
```

```
iris = load_iris()
X = iris.data
y = iris.target
```

```
[ ] rf = RandomForestClassifier()
    rf.fit(X,y)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
[ ] rf.predict(X)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[ ] joblib.dump(rf, "./random_forest.joblib")
```

```
[ ] loaded_rf = joblib.load("./random_forest.joblib")
```

```
[ ] loaded_rf.predict(X)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[ ] joblib.dump(rf, "RF_uncompressed.joblib", compress=0)
```

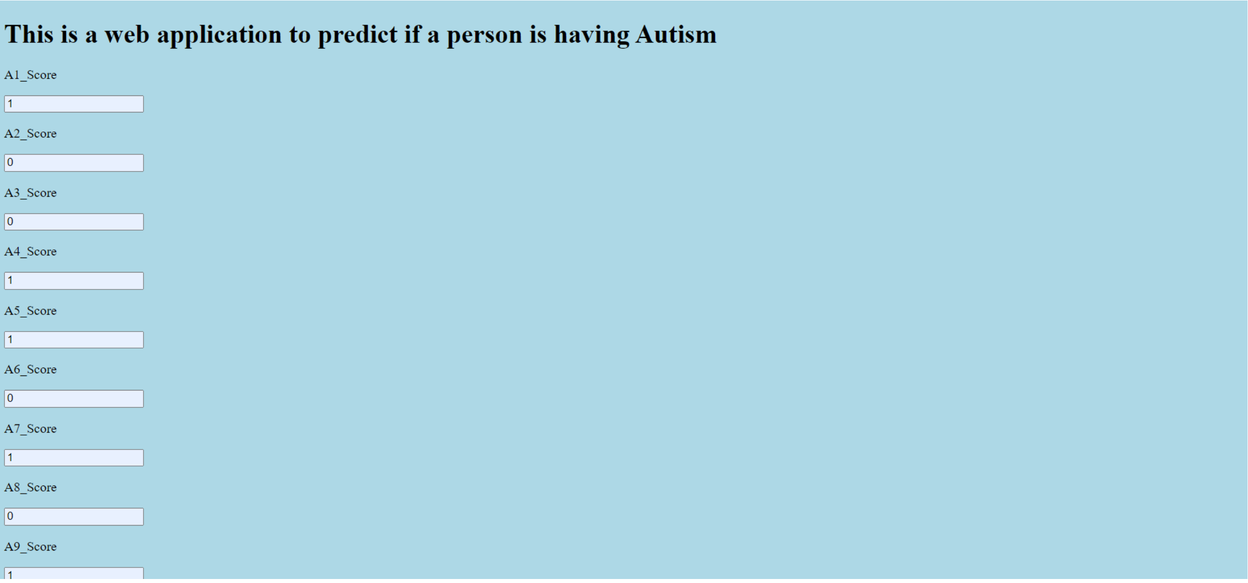
```
['RF_uncompressed.joblib']
```

```
!pip install flask
```

```
Requirement already satisfied: flask in /usr/local/lib/python3.10/dist-packages (2.2.5)
Requirement already satisfied: Werkzeug>=2.2.2 in /usr/local/lib/python3.10/dist-packages (from flask) (3.0.1)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in /usr/local/lib/python3.10/dist-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-packages (from flask) (8.1.7)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.0->flask) (2.1.3)
```

```
[ ] from flask import Flask,render_template,url_for,request,send_from_directory
```

Screenshots of the website-



This is a web application to predict if a person is having Autism

A1_Score

1

A2_Score

0

A3_Score

0

A4_Score

1

A5_Score

1

A6_Score

0

A7_Score

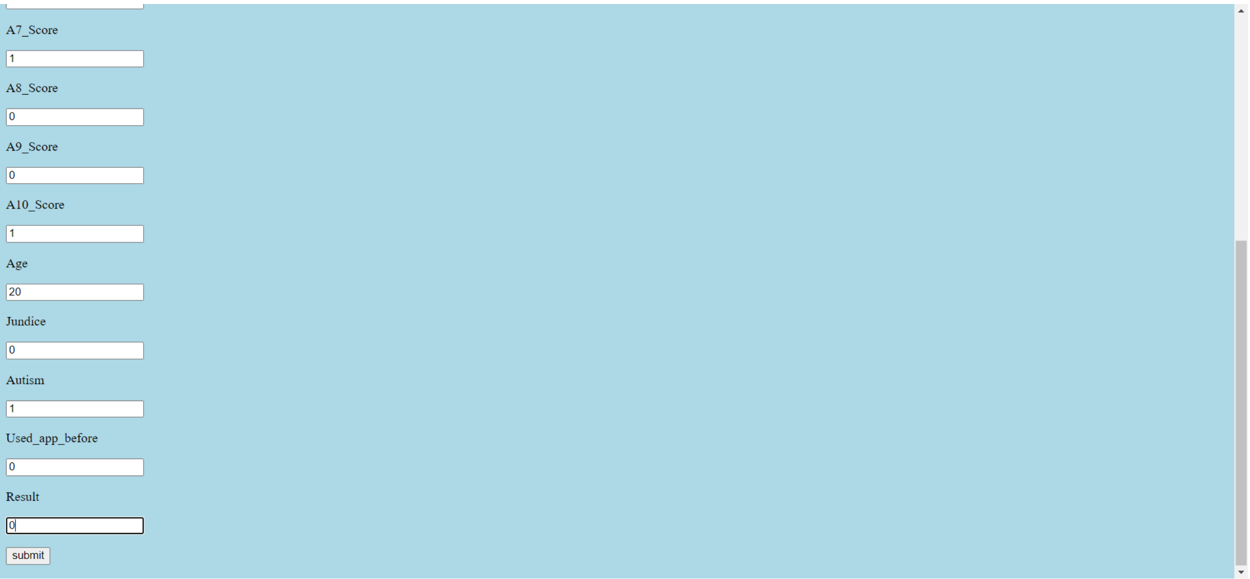
1

A8_Score

0

A9_Score

1



A7_Score

1

A8_Score

0

A9_Score

0

A10_Score

1

Age

20

Jundice

0

Autism

1

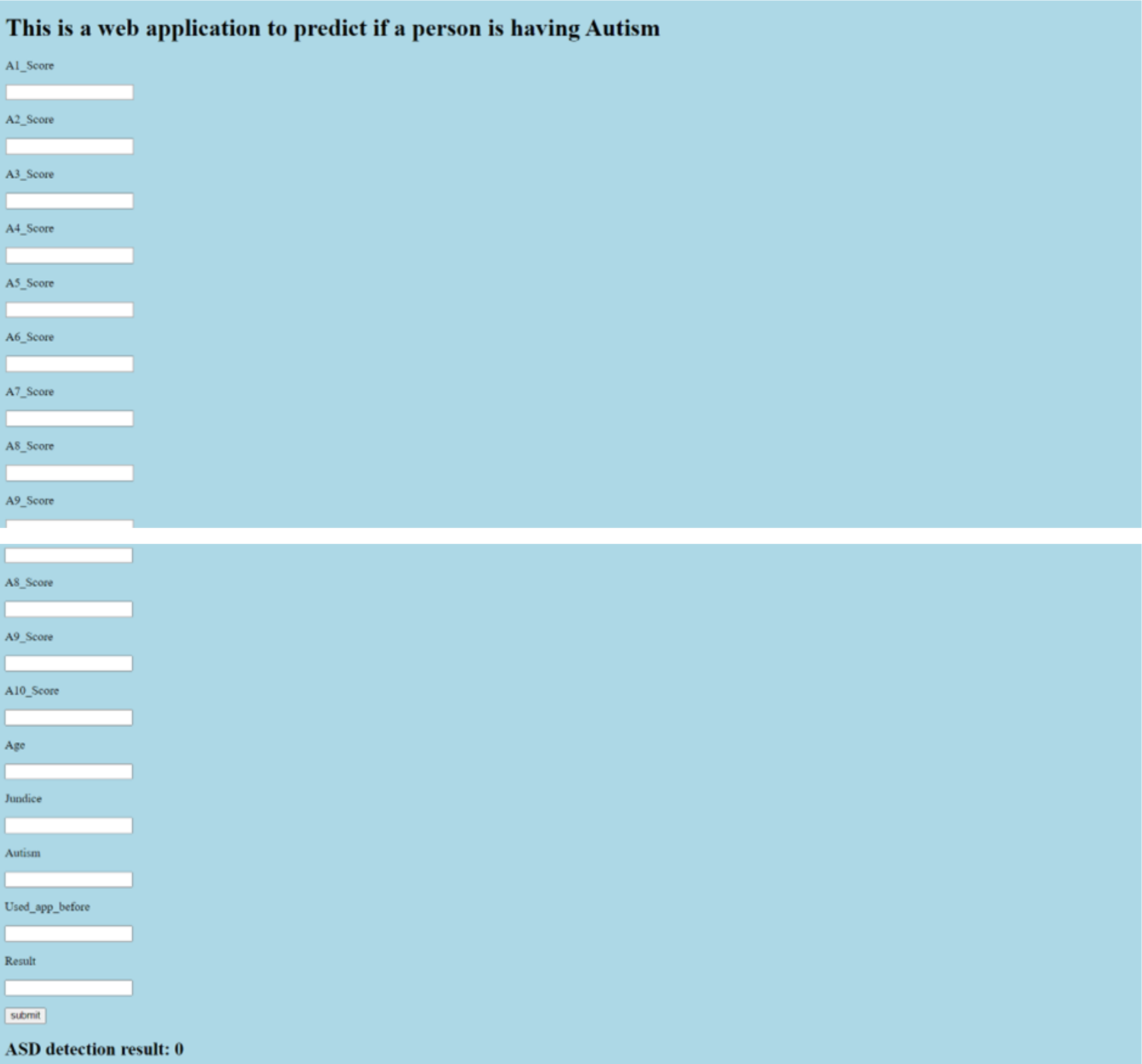
Used_app_before

0

Result

0

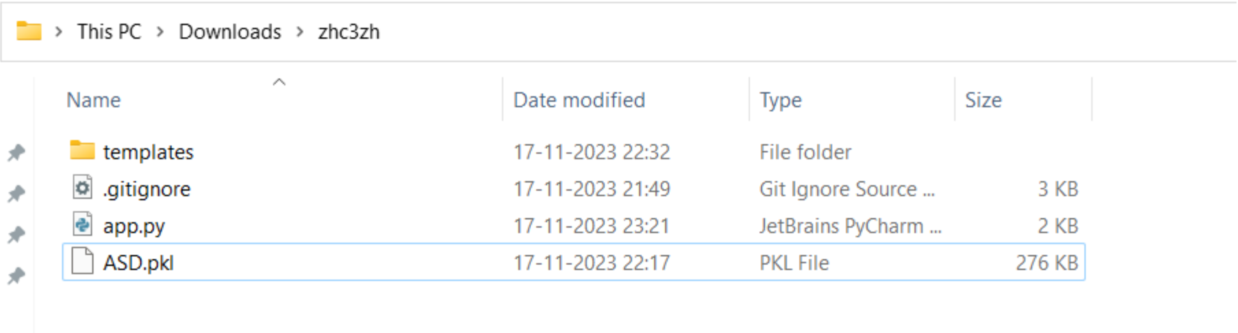
Screen shot of the OUTPUT



This is a web application to predict if a person is having Autism

ASD detection result: 0

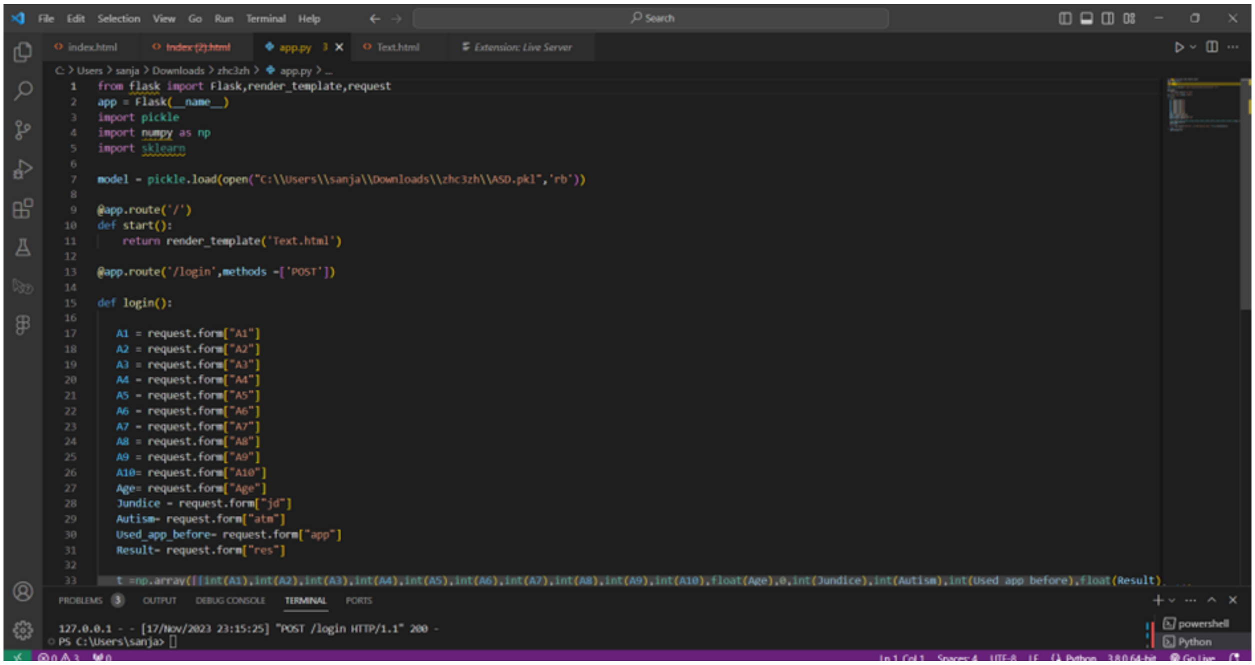
Screenshot of the folder structure-



This PC › Downloads › zhc3zh

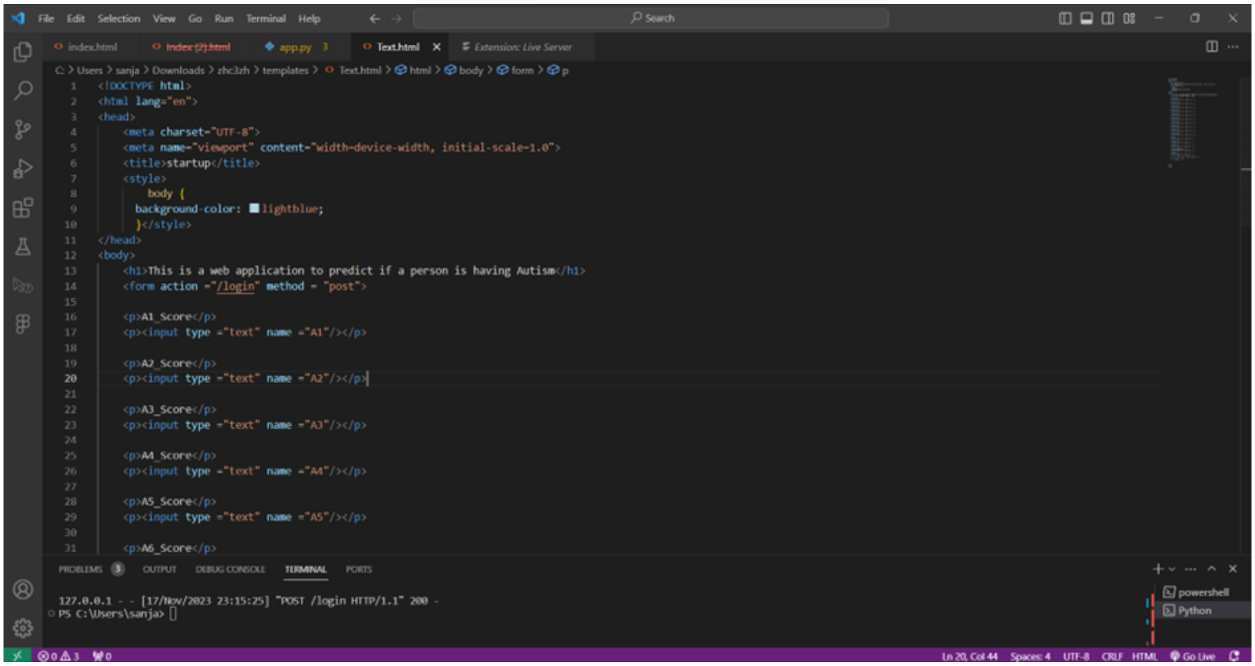
Name	Date modified	Type	Size
templates	17-11-2023 22:32	File folder	
.gitignore	17-11-2023 21:49	Git Ignore Source ...	3 KB
app.py	17-11-2023 23:21	JetBrains PyCharm ...	2 KB
ASD.pkl	17-11-2023 22:17	PKL File	276 KB

Screenshots of the code-


File Edit Selection View Go Run Terminal Help
Search
index.html
Index (2).html
app.py 3
Text.html
Extension: Live Server
C: > Users > sanja > Downloads > zhc3zh > app.py > ...

```python
from flask import Flask,render_template,request
app = Flask(__name__)
import pickle
import numpy as np
import sklearn

model = pickle.load(open("C:\\Users\\sanja\\Downloads\\zhc3zh\\ASD.pkl",'rb'))

@app.route('/')
def start():
    return render_template('Text.html')

@app.route('/login',methods =['POST'])

def login():

    A1 = request.form["A1"]
    A2 = request.form["A2"]
    A3 = request.form["A3"]
    A4 = request.form["A4"]
    A5 = request.form["A5"]
    A6 = request.form["A6"]
    A7 = request.form["A7"]
    A8 = request.form["A8"]
    A9 = request.form["A9"]
    A10= request.form["A10"]
    Age= request.form["Age"]
    Jundice = request.form["jd"]
    Autism= request.form["atm"]
    Used_app_before= request.form["app"]
    Result= request.form["res"]

    t =np.array([[int(A1),int(A2),int(A3),int(A4),int(A5),int(A6),int(A7),int(A8),int(A9),int(A10),float(Age),0,int(Jundice),int(Autism),int(Used_app_before),float(Result)
```

```
127.0.0.1 - - [17/Nov/2023 23:15:25] "POST /login HTTP/1.1" 200 -
PS C:\Users\sanja> 
```

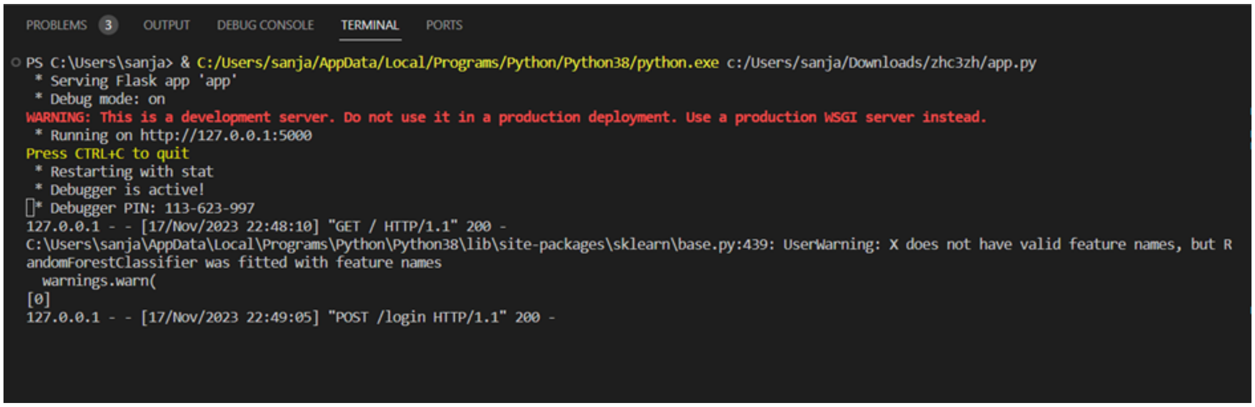


File Edit Selection View Go Run Terminal Help
Search
index.html
Index (2).html
app.py 3
Text.html
Extension: Live Server
C: > Users > sanja > Downloads > zhc3zh > templates > Text.html > html > body > form > p
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
Python
Ln 20, Col 44 Spaces: 4 UTF-8 CRLF HTML Go Live

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>startup</title>
    <style>
        body {
     background-color: lightblue;
     }</style>
</head>
<body>
    <h1>This is a web application to predict if a person is having Autism</h1>
    <form action ="/login" method = "post">

    <p>A1_Score</p>
    <p><input type ="text" name ="A1"/></p>

    <p>A2_Score</p>
    <p><input type ="text" name ="A2"/></p>

    <p>A3_Score</p>
    <p><input type ="text" name ="A3"/></p>

    <p>A4_Score</p>
    <p><input type ="text" name ="A4"/></p>

    <p>A5_Score</p>
    <p><input type ="text" name ="A5"/></p>

    <p>A6_Score</p>
```

```
127.0.0.1 - - [17/Nov/2023 23:15:25] "POST /login HTTP/1.1" 200 -
PS C:\Users\sanja> 
```

Screenshot of the terminal-



```
PS C:\Users\sanja> & C:/Users/sanja/AppData/Local/Programs/Python/Python38/python.exe c:/Users/sanja/Downloads/zhc3zh/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 113-623-997
127.0.0.1 - - [17/Nov/2023 22:48:10] "GET / HTTP/1.1" 200 -
C:\Users\sanja\AppData\Local\Programs\Python\Python38\lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but R
andomForestClassifier was fitted with feature names
  warnings.warn(
[0]
127.0.0.1 - - [17/Nov/2023 22:49:05] "POST /login HTTP/1.1" 200 -
```

ADVANTAGES & DISADVANTAGES

Advantages:

1. Early Intervention: Early detection allows for timely intervention and support, improving outcomes for individuals with ASD by providing targeted therapies and assistance during crucial developmental stages.

2. Cost Reduction: The implementation of a time-efficient and accessible screening tool can significantly reduce healthcare costs associated with the lengthy diagnostic procedures and treatments for ASD, making it more economically viable for families.

3. Global Impact: Addressing the global increase in ASD cases, this project contributes to a broader understanding and management of the condition, potentially benefiting individuals worldwide and raising awareness about the importance of early detection.

4. Efficient Resource Allocation: By streamlining the diagnostic process, healthcare resources can be allocated more efficiently, reducing waiting times and allowing professionals to focus on more complex cases, ultimately improving overall healthcare system efficiency.

5. Advancement in Research: The project's focus on analyzing behavioral traits contributes to the limited existing datasets in this domain, fostering advancements in ASD research and potentially uncovering new insights into the condition.

Disadvantages:

1. Ethical Concerns: The use of machine learning models in healthcare raises ethical concerns, such as data privacy, consent, and the potential for biases in the algorithms. Ensuring ethical considerations are addressed is crucial to maintain trust in the screening tool.

2. Overreliance on Technology: Depending solely on a machine-learning model may lead to overreliance on technology, potentially overlooking the importance of clinical expertise and human judgment in the diagnostic process.

3. False Positives and Negatives: Despite advancements, no model is perfect, and there is a risk of false positives or false negatives in ASD classification. This could lead to unnecessary stress for families or delayed interventions for those who truly need support. Regular updates and improvements to the model are necessary to minimize these risks.

CONCLUSION

In conclusion, this project delves into the intricate landscape of Autism Spectrum Disorder (ASD), recognizing its complex nature and the profound impact it has on individuals, families, and society. By addressing the existing challenges in early detection and diagnosis, we aspire to contribute to a paradigm shift in the way ASD is understood and managed.

Through a thorough literature review, we have laid the foundation for comprehending the current state of knowledge on ASD, from its clinical manifestations to the gaps in research. The proposed solution, an efficient ASD screening method, aims not only to streamline the diagnostic process but also to minimize healthcare costs and enhance the overall quality of life for affected individuals.

Recognizing the social and business dimensions of ASD, our project underscores the importance of fostering inclusivity, understanding, and support in society. Beyond the clinical realm, the economic considerations highlight the potential benefits of addressing ASD through innovative screening methods and evidence-based treatments.

As we embark on this journey, the fusion of scientific inquiry, compassion, and societal awareness is paramount. By bridging the gap between research and impact, we aspire to contribute meaningfully to the well-being of individuals with ASD and their families, creating a more compassionate and inclusive world.

APPENDIX

Source Code

https://codesandbox.io/s/fancy-darkness-zhc3zh?file=/app.py:9-3466

https://github.com/smartinternz02/SI-GuidedProject-600777-1697643084

GitHub & Project Demo Link

https://github.com/smartinternz02/SI-GuidedProject-600777-1697643084