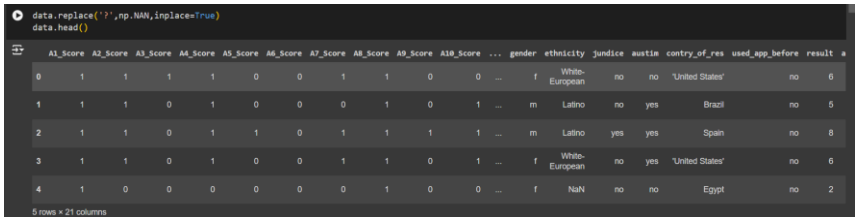


Data Collection and Preprocessing Phase

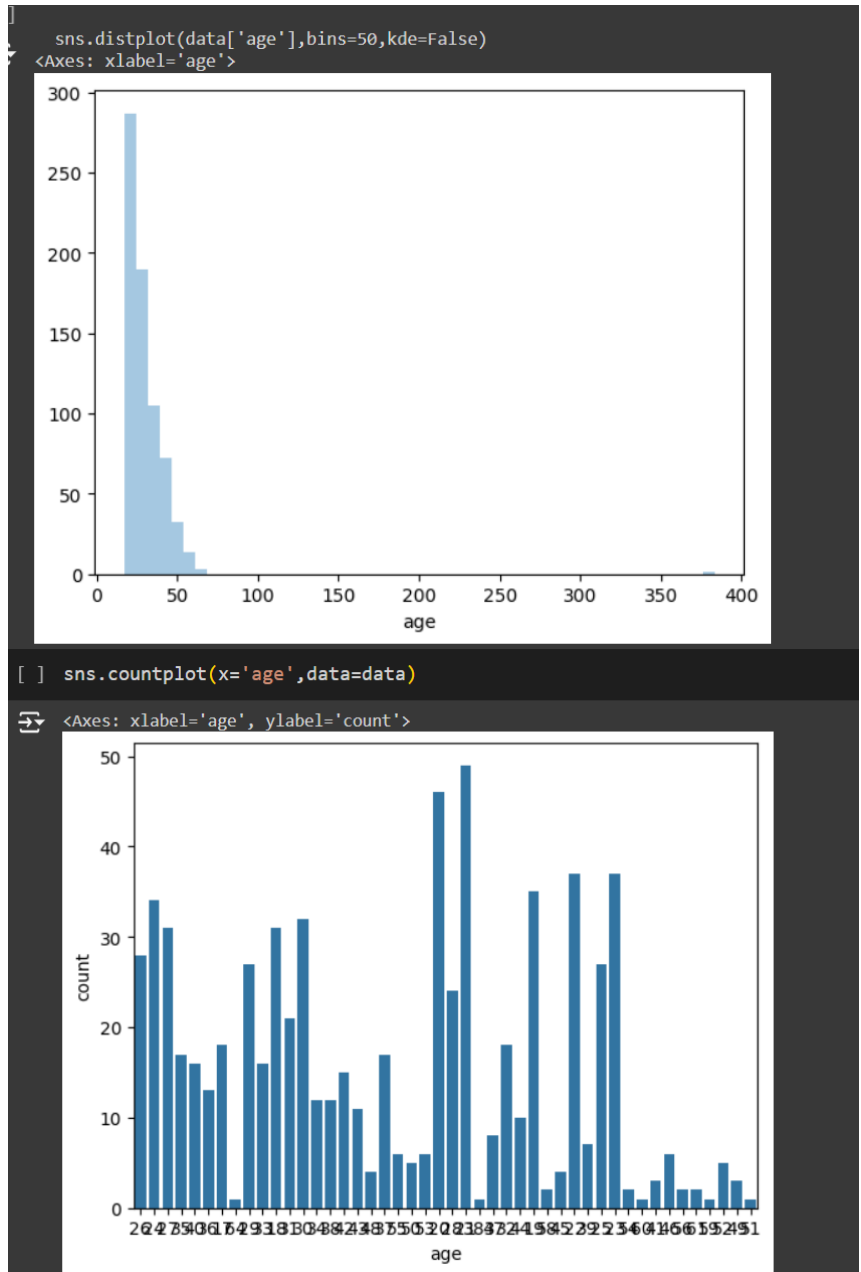
Date	15 JULY 2024
Team ID	740075
Project Title	Detection Of Autistic Spectrum Disorder: Classification
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

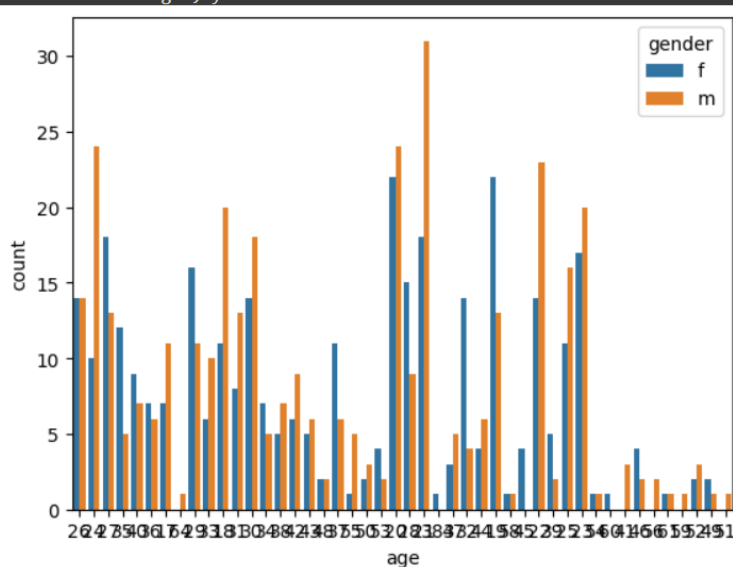
Section	Description
Data Overview	<p>#Structure of the data:</p> <pre>data.replace({'':np.NaN,inplace=True) data.head()</pre>  <pre>data.shape</pre> <p>(704, 21)</p> <p>#Descriptive Statistical:</p> <p>Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.</p>

	<div><div><div>data.describe()</div><div><table><thead><tr><th></th><th>A1_Score</th><th>A2_Score</th><th>A3_Score</th><th>A4_Score</th><th>A5_Score</th><th>A6_Score</th><th>A7_Score</th><th>A8_Score</th><th>A9_Score</th><th>A10_Score</th><th>result</th></tr></thead><tbody><tr><td>count</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td><td>704.000000</td></tr><tr><td>mean</td><td>0.721591</td><td>0.453125</td><td>0.457386</td><td>0.495739</td><td>0.498580</td><td>0.284091</td><td>0.417614</td><td>0.649148</td><td>0.323864</td><td>0.573864</td><td>4.875000</td></tr><tr><td>std</td><td>0.448535</td><td>0.498152</td><td>0.498535</td><td>0.500337</td><td>0.500353</td><td>0.451301</td><td>0.493516</td><td>0.477576</td><td>0.468281</td><td>0.494866</td><td>2.501493</td></tr><tr><td>min</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td></tr><tr><td>25%</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>3.000000</td></tr><tr><td>50%</td><td>1.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>0.000000</td><td>1.000000</td><td>0.000000</td><td>1.000000</td><td>4.000000</td></tr><tr><td>75%</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>7.000000</td></tr><tr><td>max</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>1.000000</td><td>10.000000</td></tr></tbody></table></div></div></div>		A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	result	count	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	mean	0.721591	0.453125	0.457386	0.495739	0.498580	0.284091	0.417614	0.649148	0.323864	0.573864	4.875000	std	0.448535	0.498152	0.498535	0.500337	0.500353	0.451301	0.493516	0.477576	0.468281	0.494866	2.501493	min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.000000	50%	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	4.000000	75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	7.000000	max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	10.000000
	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	result																																																																																																		
count	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000	704.000000																																																																																																		
mean	0.721591	0.453125	0.457386	0.495739	0.498580	0.284091	0.417614	0.649148	0.323864	0.573864	4.875000																																																																																																		
std	0.448535	0.498152	0.498535	0.500337	0.500353	0.451301	0.493516	0.477576	0.468281	0.494866	2.501493																																																																																																		
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000																																																																																																		
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.000000																																																																																																		
50%	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	4.000000																																																																																																		
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	7.000000																																																																																																		
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	10.000000																																																																																																		
Univariate Analysis	<p>Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.</p> <p>Univariate Analysis:</p> <p>In simple words, univariate analysis is understanding the data with a single feature. We have displayed three different types of graphs and plots.</p> <p>For simple visualizations we can use the matplotlib. pyplot library. Here the plt. figure() command is used to determine the size of the plot.</p> <p>We have histogram for all features of the dataset which include phosphorus, humidity, temperature as well . The histogram shows the distribution of nitrogen fertilizers for crop.</p> <div><div>Code cell output actions</div><div>[] sns.distplot(data['age'],bins=50,kde=False)</div></div>																																																																																																												



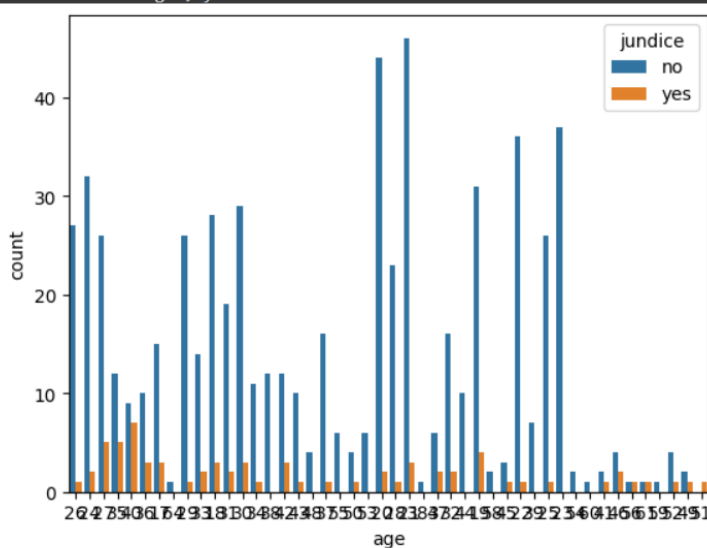
```
[ ] sns.countplot(x='age',hue='gender',data=data)
```

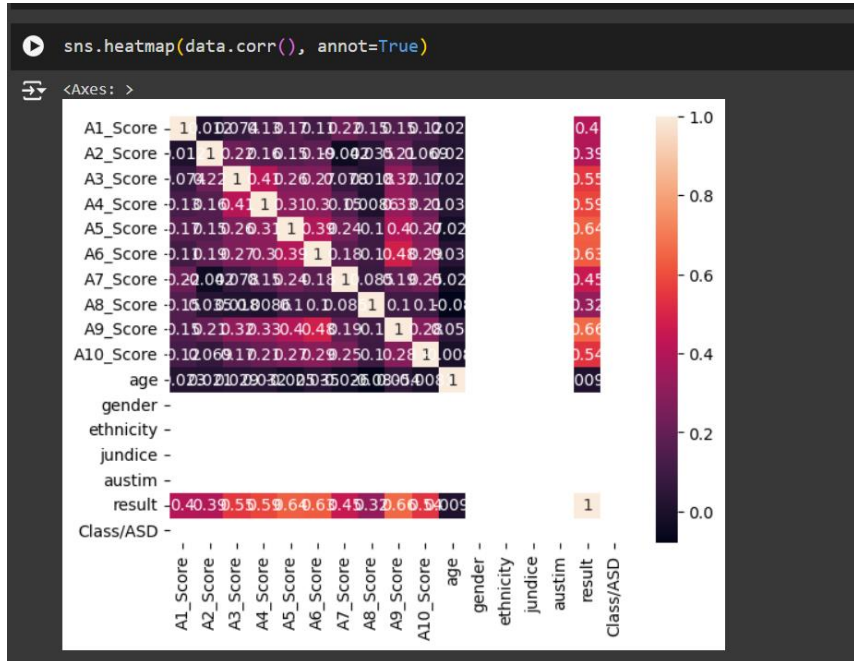
```
<Axes: xlabel='age', ylabel='count'>
```



```
> sns.countplot(x='age',hue='jundice',data=data)
```

```
<Axes: xlabel='age', ylabel='count'>
```



	 <pre>sns.heatmap(data.corr(), annot=True)</pre> <p><Axes: ></p> <p>A1_Score - 1.0002070410.170.110.220.150.150.020.4</p> <p>A2_Score - 0.0110.20.10.150.190.040.230.200.090.39</p> <p>A3_Score - 0.070.270.10.40.20.20.070.010.30.10.020.55</p> <p>A4_Score - 0.10.10.410.310.30.150.080.30.20.030.59</p> <p>A5_Score - 0.10.10.20.370.30.240.10.40.20.020.64</p> <p>A6_Score - 0.10.10.270.30.390.10.180.10.40.290.030.63</p> <p>A7_Score - 0.20.090.270.150.240.180.080.190.260.020.45</p> <p>A8_Score - 0.150.050.180.080.10.10.080.10.10.10.00.32</p> <p>A9_Score - 0.150.20.30.330.40.40.190.110.280.050.66</p> <p>A10_Score - 0.100.690.10.20.20.290.250.10.280.100.54</p> <p>age - 0.000.200.100.290.320.020.350.260.060.0410.09</p> <p>gender -</p> <p>ethnicity -</p> <p>jundice -</p> <p>austim -</p> <p>result - 0.40.30.50.50.60.60.40.30.60.50.0091</p> <p>Class/ASD -</p>
Outliers and Anomalies	There is no Outliers in our project.
Outliers and Anomalies	There is no Outliers in our project.
Data Preprocessing Code Screenshots	
Loading Data	#Loading the data

Handling Missing Data

```
data = pd.read_csv('/content/Autism_Data.arff')
data
```

	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
0	1	1	1	1	0	0	1	1	0	0	...	f	White-European	no	no	'United States'	no	6
1	1	1	0	1	0	0	0	1	0	1	...	m	Latino	no	yes	Brazil	no	5
2	1	1	0	1	1	0	1	1	1	1	...	m	Latino	yes	yes	Spain	no	8
3	1	1	0	1	0	0	1	1	0	1	...	f	White-European	no	yes	'United States'	no	6
4	1	0	0	0	0	0	0	1	0	0	...	f	?	no	no	Egypt	no	2
...
699	0	1	0	1	1	0	1	1	1	1	...	f	White-European	no	no	Russia	no	7
700	1	0	0	0	0	0	0	1	0	1	...	m	Hispanic	no	no	Mexico	no	3
701	1	0	1	1	1	0	1	1	0	1	...	f	?	no	no	Russia	no	7
702	1	0	0	1	1	0	1	0	1	1	...	m	'South Asian'	no	no	Pakistan	no	6

```
[ ] data.shape
```

```
(704, 21)
```

```
data.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                   704 non-null   object
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   int64
18  age_desc              704 non-null   object
19  relation              704 non-null   object
20  Class/ASD            704 non-null   object
dtypes: int64(11), object(10)
memory usage: 115.6+ KB
```

```
data.isnull()
```

	A1_score	A2_score	A3_score	A4_score	A5_score	A6_score	A7_score	A8_score	A9_score	A10_score	...	gender	ethnicity	jundice	austin	contry_of_res	used_app_before	result
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
...
699	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
700	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
701	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
702	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False
703	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False

704 rows x 21 columns

For checking the null values . isnull() function is used. To sum those null values we use. sum() function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.