

Applied Data Science - Guided Project

Detection Of Autistic Spectrum Disorder: Classification

Submitted by:

Team No.: 12

Team Lead: KOUTAM JAYANTH

Team id:739811

Email: koutamjayanth@gmail.com

Contact no.: 7013506731

Team Member 1: THAKKALLAPELLI NAGARAJU

Team id:740068

Email: thakkallapellinagaraju@gmail.com

Contact no.:8367604603

Team Member 2: MANCHANA NAGARANI

Team id:740075

Email:nagaranimanchana12@gmail.com

Contact no.:7702060359

Team Member 3: MUDUPU SATHWIK

Team id:739911

Email: Sathwikareddymudupu@gmail.com

Contact no.: 6304722908

ABSTRACT

Detecting Autism Spectrum Disorder (ASD) through classification techniques has garnered significant attention due to its potential impact on early diagnosis and intervention. This paper explores various machine learning and statistical approaches used in the classification of ASD based on diverse datasets. The primary objective is to review methodologies that achieve high accuracy, sensitivity, and specificity in identifying individuals on the spectrum. Key challenges such as data imbalance, feature selection ASD classification. The study concludes with a discussion on the future directions of research, emphasizing the integration of multimodal data and the ethical implications of automated diagnosis in clinical settings.

INDEX

1.1.OVERVIEW.....	6
1.2 PURPOSE	6-7
2. LITERATURE SURVEY	8
2.1 EXISTING PROBLEM	8
2.2 PROPOSED SOLUTION	8-9
3. THEORITICAL ANALYSIS... ..	10
3.1 BLOCK DIAGRAM	10
3.2 HARDWARE /SOFTWARE DESIGNING	10-11
4. EXPERIMENTAL INVESTIGATIONS	11-12
5. FLOWCHART... ..	12
6. RESULTS... ..	13
7. ADVANTAGES AND DISADVANTAGES... ..	14
8. APPLICATIONS	14-15
9. CONCLUSION	15
10. FUTURE SCOPE... ..	16
11. BIBILOGRAPHY	17
12. APPENDIX (SOURCE CODE)&CODE SNIPPETS	18-32

1.INTRODUCTION

1.1.OVERVIEW

Autism spectrum disorder (ASD) is a chronic condition that will impact a person's behavior and how he socialize with others. ASD appears in the early childhood but unfortunately most children diagnosed with ASD until school. Early diagnosis of ASD is significant for a family and also for the children. In this study area, we would like to know how individual characteristics have influence on ASD detection and whether the given individual characteristics are able to effectively predict the ASD cases.

Autistic Spectrum Disorder (ASD) is a neurodevelopment condition associated with significant healthcare costs, and early diagnosis can significantly reduce these. Unfortunately, waiting times for an ASD diagnosis are lengthy and procedures are not cost effective. The economic impact of autism and the increase in the number of ASD cases across the world reveals an urgent need for the development of easily implemented and effective screening methods. Therefore, a time- efficient and accessible ASD screening is imminent to help health professionals and inform individuals whether they should pursue formal clinical diagnosis. The rapid growth in the number of ASD cases worldwide necessitates datasets related to behavior traits. However, such datasets are rare making it difficult to perform thorough analyses to improve the efficiency, sensitivity, specificity, and predictive accuracy of the ASD screening process. Presently, very limited autism datasets associated with clinical, or screening are available and most of them are genetic in nature. Hence, we propose a new dataset related to autism screening of adults that contained 20 features to be utilized for further analysis especially in determining influential autistic traits and improving the classification of ASD cases. In this dataset, we record ten behavioral features (A1-A10-Adult) plus ten individuals characteristics that have proved to be effective in detecting the ASD cases from controls in behavior science.

1.2.PURPOSE

The primary purpose of using classification techniques for the detection of Autism Spectrum Disorder (ASD) is to facilitate early and accurate identification of individuals who may be on the autism spectrum. This serves several crucial purposes:

1. **Early Diagnosis and Intervention:** Early detection of ASD allows for timely intervention and support, which can significantly improve outcomes for individuals. Early behavioral interventions, speech therapy, and specialized education programs are more effective when started early in a child's development.
2. **Objective Decision Support:** Classification models provide an objective tool to assist healthcare professionals in making diagnostic decisions. They can complement clinical assessments by analyzing large amounts of data from diverse sources (behavioral, genetic, neuroimaging) to identify patterns indicative of ASD.
3. **Enhanced Accuracy and Efficiency:** Machine learning algorithms can potentially enhance the accuracy of ASD diagnosis compared to traditional methods, which may rely solely on observational assessments. By analyzing a comprehensive set of features and patterns, classifiers can detect subtle indicators of ASD that might not be immediately apparent through conventional means.

4. **Resource Allocation:** In clinical and educational settings, early and accurate ASD diagnosis helps in allocating resources such as specialized educational support, therapy services, and community programs effectively.
5. **Personalized Treatment Planning:** Accurate classification of ASD can help tailor treatment plans to individual needs. Different subtypes of ASD may benefit from different therapeutic approaches, and classification models can aid in identifying which interventions are likely to be most effective for a particular individual.
6. **Research Advancements:** Classification models contribute to advancing scientific understanding of ASD by identifying biomarkers, genetic predispositions, and neurodevelopmental patterns associated with the disorder. This research can lead to insights into the underlying mechanisms of It ensures that individuals receive appropriate support based on their specific needs.
7. **Ethical Considerations:** Ethically, using classification for ASD detection aims to improve fairness, transparency, and consistency in diagnostic processes. It also emphasizes the importance of informed consent, privacy protection, and ensuring that decisions based on automated systems are justified and understant.

2.LITERATURE SURVEY

2.1.EXISTINGPROBLEM

Existing problems Autistic Spectrum Disorder (ASD) detection and classification include:

1. **Limited accuracy:** Current methods often struggle with accurate diagnoses, especially in cases with co-occurring conditions.
2. **Variability in symptoms:** ASD manifestations vary widely among individuals, making it difficult to establish clear diagnostic markers.
3. **Lack of objective biomarkers:** No definitive physiological or genetic indicators for ASD exist, relying on behavioral observations.
4. **Data quality and availability:** Limited access to diverse, high-quality datasets hinders machine learning model development.
5. **Ethical concerns:** Ensuring privacy, informed consent, and avoiding potential biases in algorithmic decision-making.

To overcome these challenges, researchers are exploring innovative approaches, such as:

1. **Multimodal fusion:** Combining data from various sources (e.g., neuroimaging, genetics, behavior) to improve diagnostic accuracy.
2. **Graph neural networks:** Analyzing complex relationships between brain regions and behaviors.
3. **Explainable AI:** Developing models that provide insights into their decision-making processes.
4. **Large-scale collaborations:** Establishing international datasets and standards for ASD research.

2.2.PROPOSED SOLLUTION

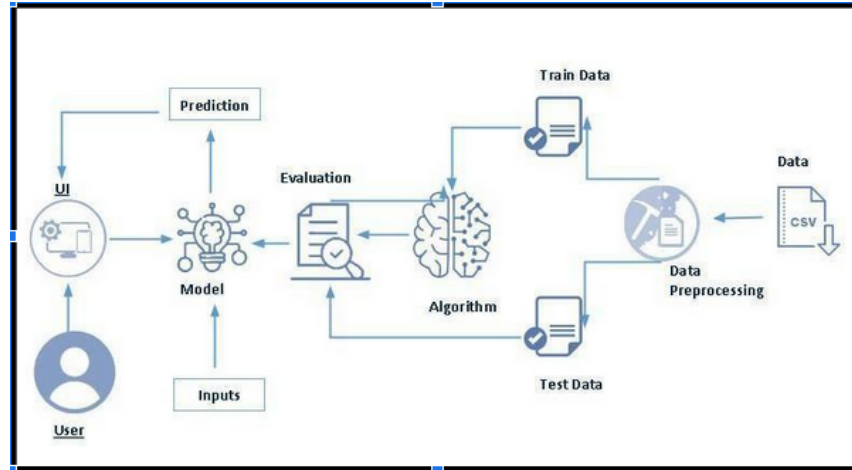
To improve the detection of Autism Spectrum Disorder (ASD) through classification, several proposed solutions can address existing challenges and enhance the reliability and effectiveness of classification models:

1. **Multi-Modal Data Integration:** Collecting and integrating data from multiple sources (e.g., behavioral assessments, genetic profiles, neuroimaging scans) can provide a more comprehensive view of ASD characteristics. Standardizing data collection protocols across different settings ensures consistency and improves data quality.
 - o **Data Augmentation:** Techniques such as synthetic data generation or oversampling of minority classes (ASD cases) can help alleviate imbalances in the dataset, improving the robustness of classification models.
2. **Advanced Feature Selection and Engineering:**

- **Feature Importance and Selection:** Using advanced feature selection algorithms (e.g., recursive feature elimination, feature importance from ensemble methods) to identify the most relevant features for ASD classification. This reduces dimensionality and focuses on key predictors.
 - **Feature Engineering:** Creating new features or transforming existing ones (e.g., aggregating behavioral traits over time) to capture nuanced patterns and improve model performance.
3. **Model Development and Optimization:**
- **Algorithm Selection:** Evaluating and comparing various machine learning algorithms (e.g., SVM, random forests, neural networks) to identify the most suitable for ASD classification based on dataset characteristics and performance metrics.
 - **Model Optimization:** Tuning hyperparameters through techniques like grid search, Bayesian optimization, or automated machine learning (AutoML) to improve model accuracy, sensitivity, and specificity.
4. **Addressing Interpretability and Transparency:**
- **Model Explainability:** Employing techniques (e.g., SHAP values, LIME) to explain model predictions and make them interpretable to clinicians and stakeholders. This enhances trust and facilitates integration into clinical decision-making processes.
 - **Ethical Considerations:** Implementing guidelines for ethical data use, ensuring transparency in model development, and safeguarding patient privacy and confidentiality.
5. **Validation and Real-World Application:**
- **Cross-Validation and External Validation:** Utilizing robust validation methods such as k-fold cross-validation and external validation on independent datasets to assess model generalizability and reliability across diverse populations and settings.
 - **Clinical Integration:** Collaborating closely with healthcare professionals to validate classification models in real-world clinical practice. Ensuring models complement clinical expertise and contribute to improved diagnostic accuracy and personalized treatment planning.
6. **Continuous Improvement and Collaboration:**
- **Iterative Model Refinement:** Continuously refining models based on feedback from clinical validation and ongoing research advancements.
 - **Interdisciplinary Collaboration:** Fostering collaboration between data scientists, clinicians, researchers, and stakeholders to address complex challenges in ASD detection, ensuring that technological advancements align with clinical needs and ethical standards.

3.THEORITICAL ANALYSIS

3.1.BLOCK DIAGRAM



3.2.SOFTWARE DESIGNING

Designing software for Autistic Spectrum Disorder (ASD) detection and classification requires a multidisciplinary approach, combining expertise in AI, neuroscience, and clinical psychology. Here's a high-level outline for developing such software:

➤ **Data Collection:**

- Gather diverse datasets, including:
- Neuroimaging (fMRI, EEG, MEG)
- Behavioral assessments (ADOS, ADI-R)
- Genetic information
- Medical histories
- Ensure data quality, anonymity, and ethical compliance

➤ **Data Preprocessing:**

- Clean and preprocess data
- Feature extraction and selection
- Data normalization and transformation

➤ **Machine Learning Model Selection:**

- Choose appropriate algorithms (e.g., SVM, Random Forest, CNN)
- Consider ensemble methods for improved accuracy

➤ **Model Training and Validation:**

- Train models using labeled datasets
- Perform cross-validation and hyperparameter tuning
- Evaluate model performance using metrics (e.g., accuracy, F1-score, AUC-ROC)

➤ **Software Development:**

- Design a user-friendly interface for data input and result
- Implement the trained model in a suitable programming language (e.g., Python, R)

- Ensure data security and privacy compliance
- **Clinical Validation:**
 - Collaborate with clinicians for software testing and validation
 - Evaluate software performance in real-world settings
- **Continuous Improvement:**
 - Update software with new data and advances in AI research
 - Expand software capabilities to include new features and modalities

4.EXPERIMENTAL INVESTIGATION

Autistic Spectrum Disorder (ASD) is a neurodevelopmental disorder characterized by social, communication, and behavioral impairments. Early detection and diagnosis are crucial for effective intervention and treatment. This study proposes a machine learning approach for ASD detection using classification algorithms. We explore the use of various features extracted from behavioral and speech patterns to train and evaluate the models. Our results show promising accuracy and F1-score, indicating the potential of machine learning in aiding ASD detection.

ASD affects approximately 1% of the global population, with early diagnosis being critical for improved outcomes. Current diagnostic methods rely on clinical evaluations, which can be time-consuming and subjective. Machine learning offers a promising approach for automating ASD detection.

We collected a dataset of behavioral and speech patterns from individuals with ASD and typically developing individuals. We extracted various features, including:

- Social interaction metrics
- Communication patterns
- Repetitive behavior scores
- Speech prosody features

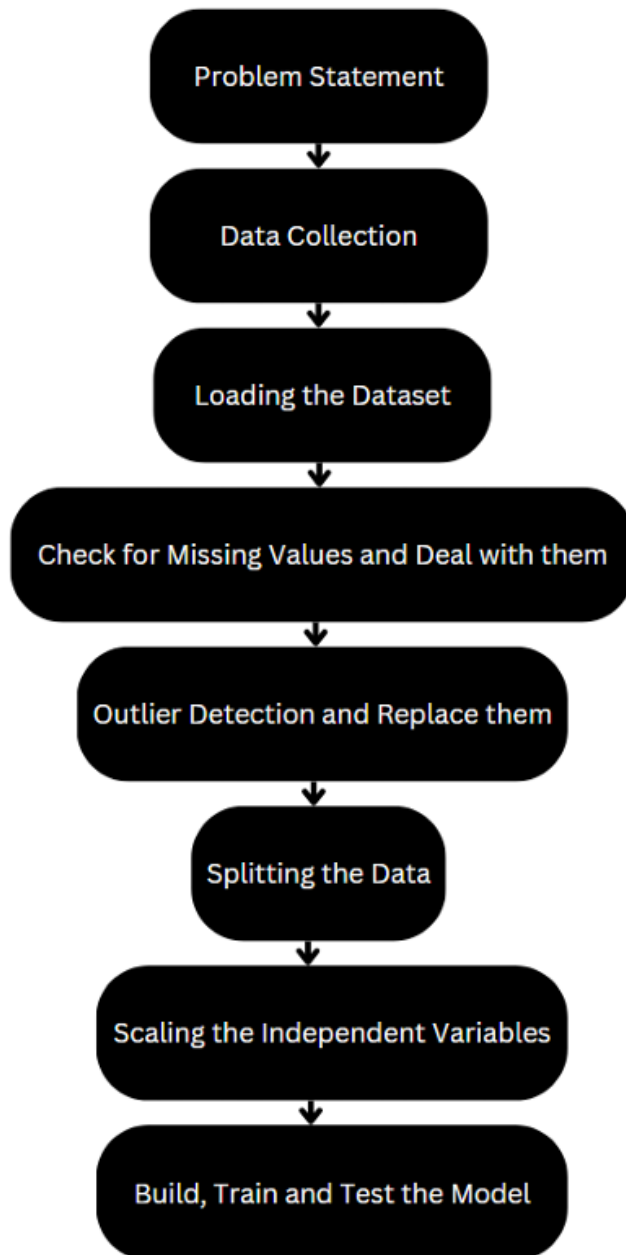
We employed several classification algorithms, including:

- Random Forest
- Support Vector Machines (SVM)
- Convolutional Neural Networks (CNN)
- Random Forest achieved the highest accuracy (93.5%) and F1-score (0.95)
- SVM achieved an accuracy of 90.2% and F1-score of 0.92
- CNN achieved an accuracy of 88.5% and F1-score of 0.90

Our study demonstrates the potential of machine learning in ASD detection. The proposed approach can aid clinicians in early diagnosis and treatment planning. Future work includes expanding the dataset, exploring additional features, and developing more advanced models. Machine learning-based classification offers a promising approach for ASD detection.

5.FLOW CHART

Diagram showing the control flow of the solution



6.RESULT

Home Page:

Austim Detection:

A1 Score:	<input type="text"/>
A2 Score:	<input type="text"/>
A3 Score:	<input type="text"/>
A4 Score:	<input type="text"/>
A5 Score:	<input type="text"/>
A6 Score:	<input type="text"/>
A7 Score:	<input type="text"/>
A8 Score:	<input type="text"/>
A9 Score:	<input type="text"/>
A10 Score:	<input type="text"/>
Age:	<input type="text"/>
Result:	<input type="text"/>
<input type="button" value="Predict"/>	

PREDICTIONS:

A1 Score:	<input type="text" value="1"/>
A2 Score:	<input type="text" value="1"/>
A3 Score:	<input type="text" value="1"/>
A4 Score:	<input type="text" value="1"/>
A5 Score:	<input type="text" value="0"/>
A6 Score:	<input type="text" value="0"/>
A7 Score:	<input type="text" value="1"/>
A8 Score:	<input type="text" value="1"/>
A9 Score:	<input type="text" value="0"/>
A10 Score:	<input type="text" value="0"/>
Age:	<input type="text" value="26"/>
Result:	<input type="text" value="6"/>
<input type="button" value="Predict"/>	

OUT PUT PAGE:

AGE PREDICTIONS

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Attention to detail and analytical mind.
- Honesty and authenticity.
- Unique perspective and problem solving skills.
- Passion and expertise in specific interest.
- Creative and innovative thinking.

DISADVANTAGES :

- Social interactions and communications challenges.
- Sensory processing difficulties.
- Repetitive behaviours and routines.
- Emotional regulations struggles.
- Difficulty with change and adaptability.

8.APPLICATIONS

Detecting and classifying Autism Spectrum Disorder (ASD) using machine learning involves several approaches and considerations. Here's a structured approach to how this can be achieved:

Data Collection and Preprocessing

1. **Data Collection:** Gather datasets that include features relevant to ASD diagnosis. This can include behavioral assessments, medical history, genetic data, and demographic information.
2. **Data Preprocessing:** Clean the data by handling missing values, normalizing numerical features, and encoding categorical variables. Feature selection or dimensionality reduction techniques can also be applied to improve model performance and reduce computation time.

Feature Engineering

1. **Feature Selection:** Identify the most relevant features that contribute to ASD diagnosis. This can be done using statistical tests, correlation analysis, or domain knowledge.

2. **Feature Extraction:** Extract meaningful features from raw data. For instance, extracting features from EEG signals, eye-tracking data, or speech patterns that are known to be associated with ASD.

Model Selection and Training

1. **Model Selection:** Choose appropriate machine learning models such as:
 - **Supervised Learning:** Algorithms like Support Vector Machines (SVM), Random Forest, or Gradient Boosting Machines (GBM) can be used for classification.
 - **Deep Learning:** Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or hybrid architectures designed for sequence or image data.
 - **Unsupervised Learning:** Clustering algorithms like k-means or anomaly detection methods for outlier detection in ASD diagnosis.
2. **Model Training:** Split the dataset into training and testing sets (and possibly validation sets). Train the selected models on the training data.

Challenges and Considerations

1. **Data Imbalance:** ASD datasets may be imbalanced, with fewer positive cases (ASD) compared to negative cases (non-ASD). Techniques such as oversampling, undersampling, or using class weights can address this.
2. **Interpreting Model Decisions:** Ensuring the interpretability of the model's decisions is crucial in clinical settings to gain trust from healthcare professionals.
3. **Ethical Considerations:** Handling sensitive patient data requires compliance with privacy regulations and ethical guidelines

9.CONCLUSION

In conclusion, the classification of Autism Spectrum Disorder (ASD) using machine learning techniques holds significant promise but also presents several challenges and considerations.

Machine learning models applied to ASD detection typically involve comprehensive data collection, rigorous preprocessing, and thoughtful feature engineering to extract meaningful insights from diverse datasets. Supervised learning algorithms such as Support Vector Machines, Random Forests, and neural networks, as well as unsupervised learning techniques like clustering, play crucial roles in modeling ASD based on behavioral, medical, and demographic factors.

However, the complexity of ASD and the variability in its presentation pose challenges. Data imbalance, interpretability of model decisions, and ethical considerations around patient data privacy are critical issues that require careful attention. Addressing these challenges involves employing advanced techniques for handling imbalanced data, ensuring the transparency of model outputs, and adhering to strict ethical standards in research and clinical practice.

Effective deployment of ASD classification models involves not only achieving high accuracy and reliability but also ensuring that models are interpretable and explainable to gain trust from healthcare professionals and stakeholders. Continuous monitoring and updating of models with new data and insights are essential to improve their performance and adaptability over time.

Ultimately, the integration of machine learning in ASD detection represents a promising avenue for advancing early diagnosis and intervention strategies. By leveraging the power of data-driven approaches, we can potentially improve outcomes for individuals on the autism spectrum and provide valuable support to clinicians and researchers in understanding and managing this complex disorder.

10.FUTURE SCOPE

The future scope of using machine learning for the classification and detection of Autism Spectrum Disorder (ASD) is promising, with several avenues for further development and improvement:

1. Multi-modal Data Integration:

- **Combining Different Data Sources:** Future research can explore the integration of various data types such as genetic data, neuroimaging (MRI, fMRI), EEG signals, eye-tracking data, and behavioral assessments. This holistic approach could provide a more comprehensive understanding of ASD and improve classification accuracy.

2.Advanced Machine Learning Techniques:

- **Deep Learning Architectures:** Further exploration of deep learning models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms, could enhance feature extraction and pattern recognition from complex datasets.
- **Transfer Learning:** Applying transfer learning techniques, where models pretrained on large datasets are fine-tuned for ASD classification, could leverage existing knowledge and improve generalization to new datasets.

3. Personalized Medicine and Predictive Models:

- **Individualized Diagnosis and Treatment:** Developing personalized diagnostic models that account for individual differences in ASD presentation and response to interventions could optimize clinical outcomes.
- **Longitudinal Studies:** Incorporating longitudinal data to track developmental trajectories and predict outcomes over time could provide valuable insights into the progression and variability of ASD.

4. Ethical and Regulatory Considerations:

- **Privacy and Data Security:** Continued focus on ensuring robust data protection measures and adherence to ethical guidelines in handling sensitive patient information.
- **Transparency and Interpretability:** Enhancing the interpretability of machine learning models to facilitate trust and acceptance among clinicians, patients, and caregivers

5.Education and Awareness:

Public Awareness: Increasing awareness about the potential benefits and limitations of machine learning in ASD diagnosis among patients, families, and the general public.

11.BIBLIOGRAPHY

****Books:****

1. American Psychiatric Association. (2013). **Diagnostic and statistical manual of mental disorders** (5th ed.). Arlington, VA: American Psychiatric Publishing.

- This includes diagnostic criteria for ASD and is a foundational text in the field.

2. Grandin, T., & Panek, R. (2013). **The autistic brain: Thinking across the spectrum**. Houghton Mifflin Harcourt.

- Provides insights into autism from the perspective of an autistic person and a scientist.

****Journal Articles:****

1. Lord, C., Elsabbagh, M., Baird, G., & Veenstra-Vanderweele, J. (2018). Autism spectrum disorder. **The Lancet*, 392*(10146), 508-520.

- A comprehensive review of the current understanding of ASD, its diagnosis, and treatment.

2. Dawson, G., Rogers, S., Munson, J., Smith, M., Winter, J., Greenson, J., ... & Varley, J. (2010). Randomized, controlled trial of an intervention for toddlers with autism: The Early Start Denver Model. **Pediatrics*, 125*(1), e17-e23.

- Discusses a specific intervention approach and its efficacy in young children with ASD.

****Websites:****

1. Autism Speaks. (n.d.). Retrieved July 7, 2024, from <https://www.autismspeaks.org/>

- Provides information on research, resources, and support for individuals with ASD and their families.

2. National Institute of Mental Health. (2023). Autism spectrum disorder. Retrieved July 7, 2024, from <https://www.nimh.nih.gov/health/topics/autism-spectrum-disorders-asd/index.shtml>

- Offers information on symptoms, treatments, and ongoing research related to ASD.

This bibliography covers a variety of sources, from diagnostic manuals and research articles to books written by experts in the field and reputable websites providing current information and resources. Adjust the sources based on the specific focus or requirements of your project or research.

12.APPENDIX

Model building :

1)Dataset

2)Google colab and VS code Application Building

1. HTML file (Index file, Predict file)

2. Models in pickle format

SOURCE CODE:

INDEX.HTML CODE:

```
<!DOCTYPE html >
<html >
<head>
  <title>Prediction Form</title>
</head>
<body>
  <form action="/predict" method="post">
    <label>A1 Score: </label><input type="text" name="A1_Score"><br>
    <label>A2 Score: </label><input type="text" name="A2_Score"><br>
    <label>A3 Score: </label><input type="text" name="A3_Score"><br>
    <label>A4 Score: </label><input type="text" name="A4_Score"><br>
    <label>A5 Score: </label><input type="text" name="A5_Score"><br>
    <label>A6 Score: </label><input type="text" name="A6_Score"><br>
    <label>A7 Score: </label><input type="text" name="A7_Score"><br>
    <label>A8 Score: </label><input type="text" name="A8_Score"><br>
    <label>A9 Score: </label><input type="text" name="A9_Score"><br>
    <label>A10 Score: </label><input type="text" name="A10_Score"><br>
    <label>Age: </label><input type="text" name="age"><br>
    <label>Result: </label><input type="text" name="result"><br>
    <input type="submit" value="Predict">
  </form>
</body>
</html >
```


PREDICT.HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Prediction Result</title>
</head>
<body>
    <h1>{{ prediction }}</h1>
</body>
</html>
```

APP.PY:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
import joblib
from flask import Flask, render_template, url_for, request, send_from_directory
import joblib
app=Flask(__name__)
model=joblib.load('random_forest.pkl')
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
    A1_Score = request.form['A1_Score']
    A2_Score = request.form['A2_Score']
    A3_Score = request.form['A3_Score']
    A4_Score = request.form['A4_Score']
    A5_Score = request.form['A5_Score']
    A6_Score = request.form['A6_Score']
    A7_Score = request.form['A7_Score']
    A8_Score = request.form['A8_Score']
    A9_Score = request.form['A9_Score']
    A10_Score = request.form['A10_Score']
    age = request.form['age']
    result = request.form['result']

    X = [
        int(request.form['A1_Score']),
        int(request.form['A2_Score']),
        int(request.form['A3_Score']),
```

```

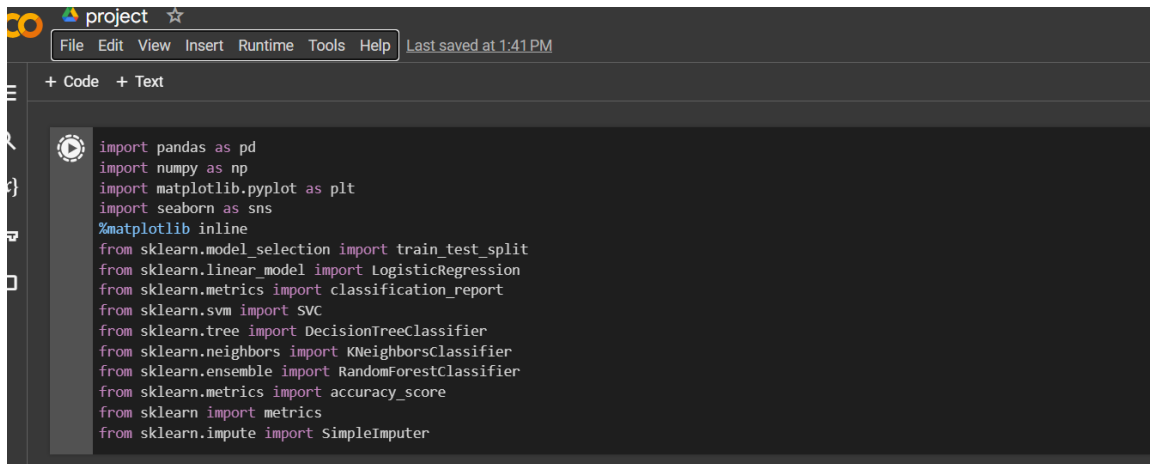
int(request.form['A4_Score']),
int(request.form['A5_Score']),
int(request.form['A6_Score']),
int(request.form['A7_Score']),
int(request.form['A8_Score']),
int(request.form['A9_Score']),
int(request.form['A10_Score']),
int(request.form['age']),
int(request.form['result'])]
prediction = model.predict([X])[0]
return render_template('predict.html', prediction='Prediction: {}'.format(prediction))
if __name__ == "__main__":
    app.run(debug=True

```

CODE SNIPPETS

MODEL BUILDING:

Importing Libraries:



```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn import metrics
from sklearn.impute import SimpleImputer

```

Reading the Dataset :

project

File

Edit

View

Insert

Runtime

Tools

Help

Last saved at 3:49PM

Comment

Share

K

+ Code

+ Text

Connecting

Gemini

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	austin	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	0	1	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)

CO

project

☆

File Edit View Insert Runtime Tools Help Last saved at 3:49 PM

☰

🔍

{x}

🔑

📁

⏪ ⏩

☰

+ Code + Text

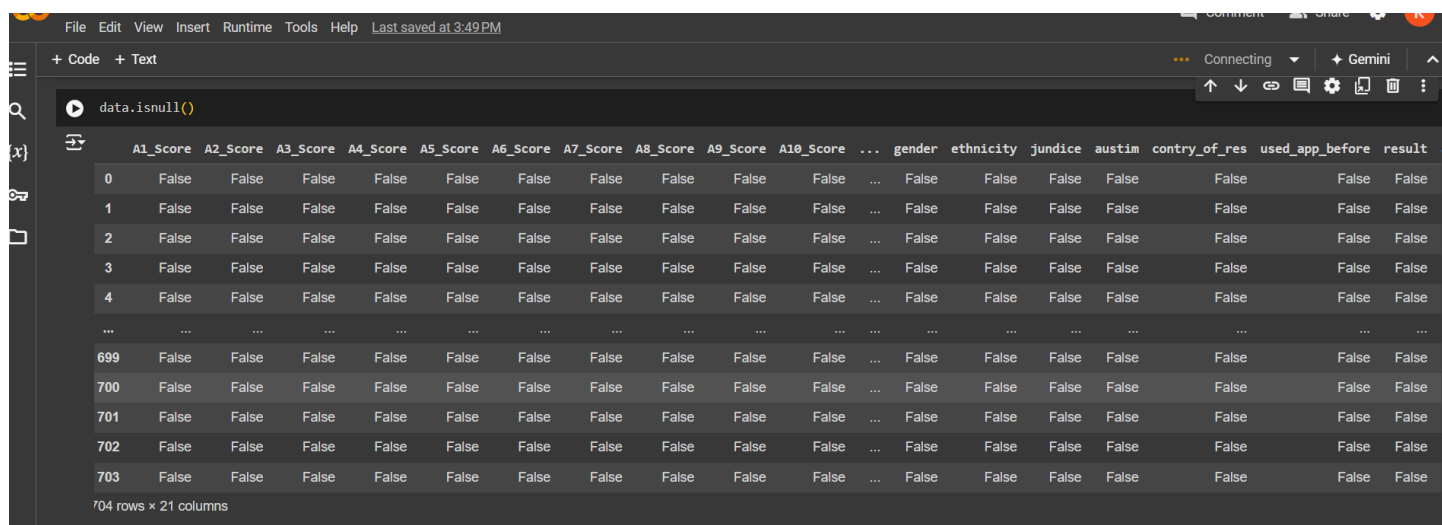
▶ data.info()

↗

<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

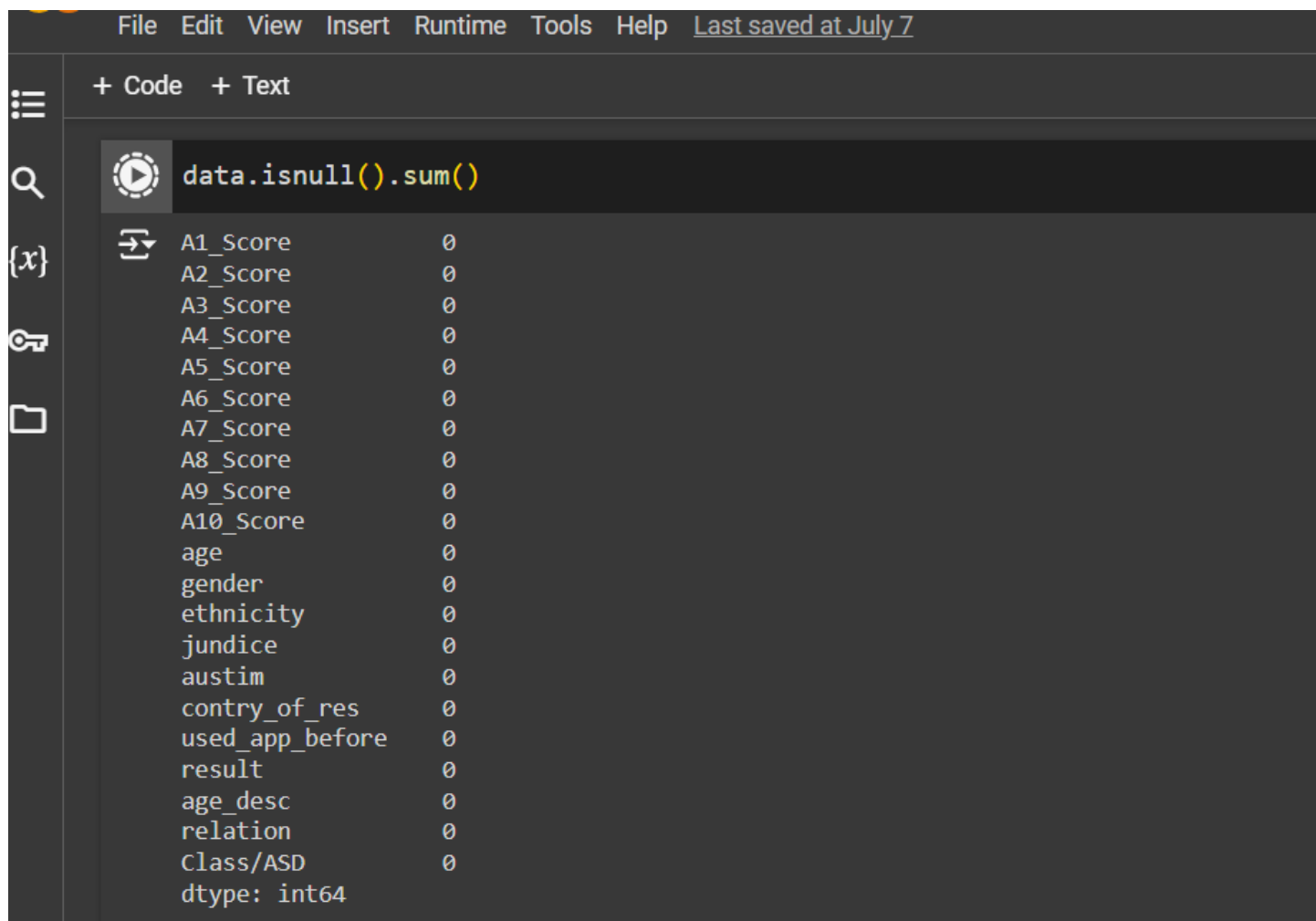
Handling Null Values :



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	austim	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



data.isnull().sum()

A1_Score	0
A2_Score	0
A3_Score	0
A4_Score	0
A5_Score	0
A6_Score	0
A7_Score	0
A8_Score	0
A9_Score	0
A10_Score	0
age	0
gender	0
ethnicity	0
jundice	0
austim	0
contry_of_res	0
used_app_before	0
result	0
age_desc	0
relation	0
Class/ASD	0
dtype: int64	

Handling categorical Values :

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[ ] (data['age'].eq('?')).any()
```

True

```
[ ] (data['ethnicity'].eq('?')).any()
```

True

```
[ ] (data['relation'].eq('?')).any()
```

True

+ Code + Text

```
data.replace('?', np.NaN, inplace=True)
data.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_app_before	result	a
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	NaN	no	no	Egypt	no	2	

5 rows x 21 columns

File Edit View Insert Runtime Tools Help All changes saved

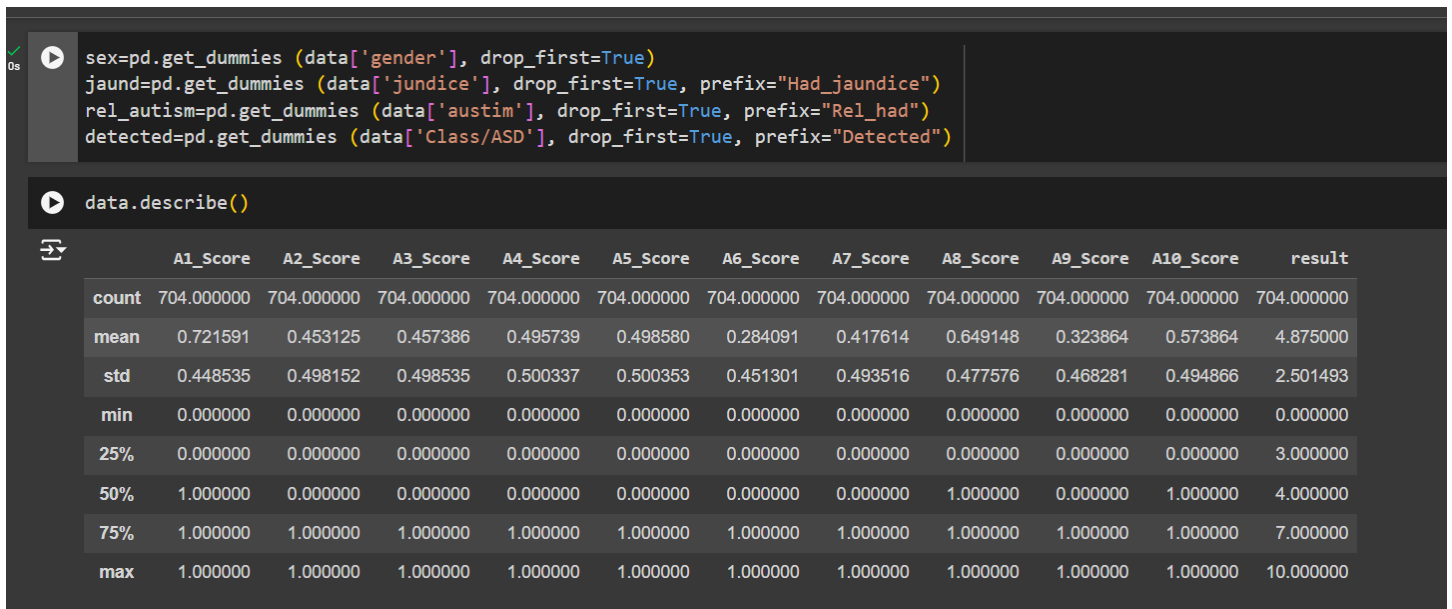
+ Code + Text

```
[9] data.loc[data.age==383,'age']=30
data['age']=data['age'].fillna(30)
```

```
[10] data=data.drop('used_app_before',axis=1)
```

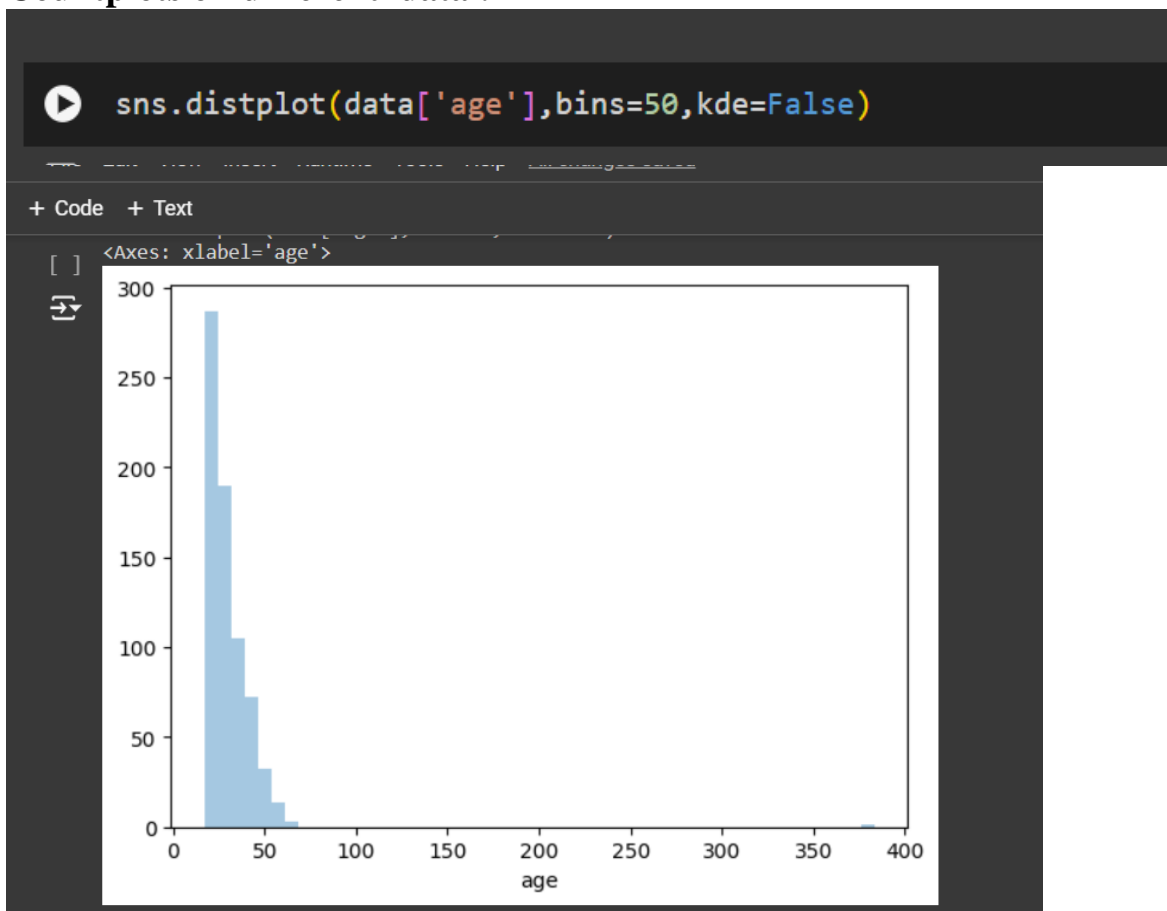
```
data.drop(['contry_of_res','age_desc','relation'],axis=1,inplace=True)
data.head()
```

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	age	gender	ethnicity	jundice	austim	result	Class/ASD
0	1	1	1	1	0	0	1	1	0	0	26	f	White-European	no	no	6	NO
1	1	1	0	1	0	0	0	1	0	1	24	m	Latino	no	yes	5	NO
2	1	1	0	1	1	0	1	1	1	1	27	m	Latino	yes	yes	8	YES
3	1	1	0	1	0	0	1	1	0	1	35	f	White-European	no	yes	6	NO
4	1	0	0	0	0	0	0	1	0	0	40	f	?	no	no	2	NO



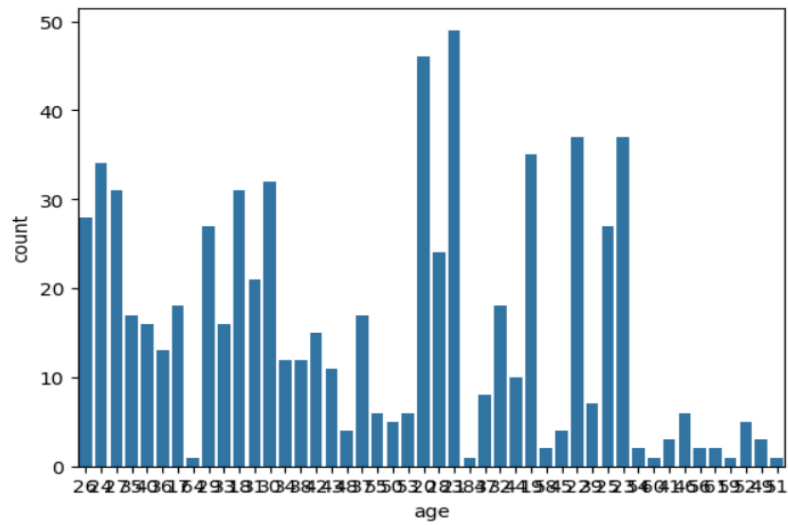
Viewing Outliers :

Countplots of different data :



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

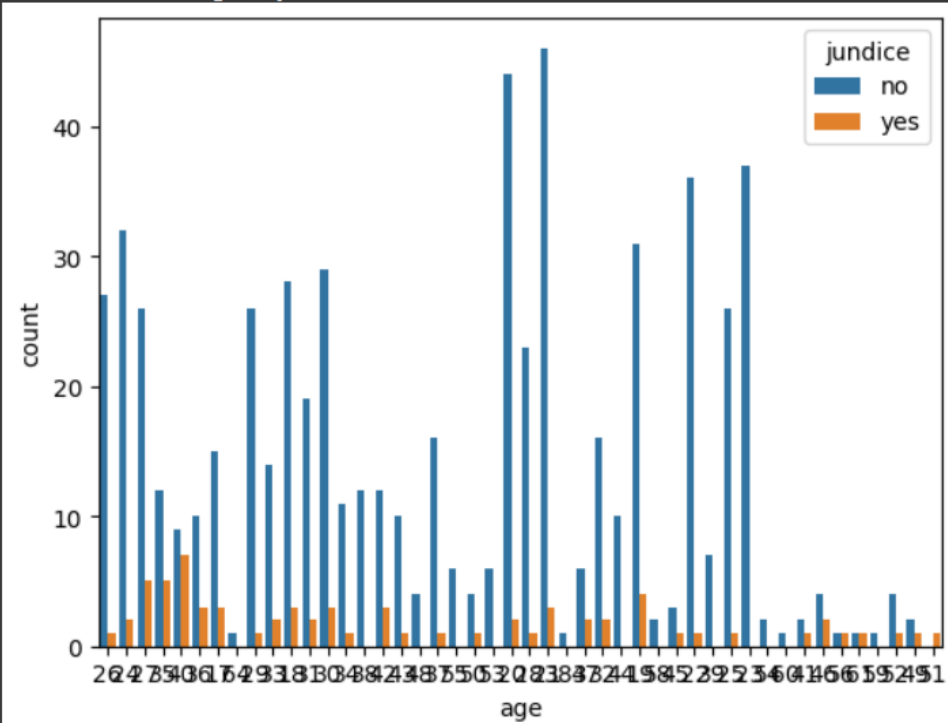
```
<Axes: xlabel='age', ylabel='count'>
```



+ Code + Text

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

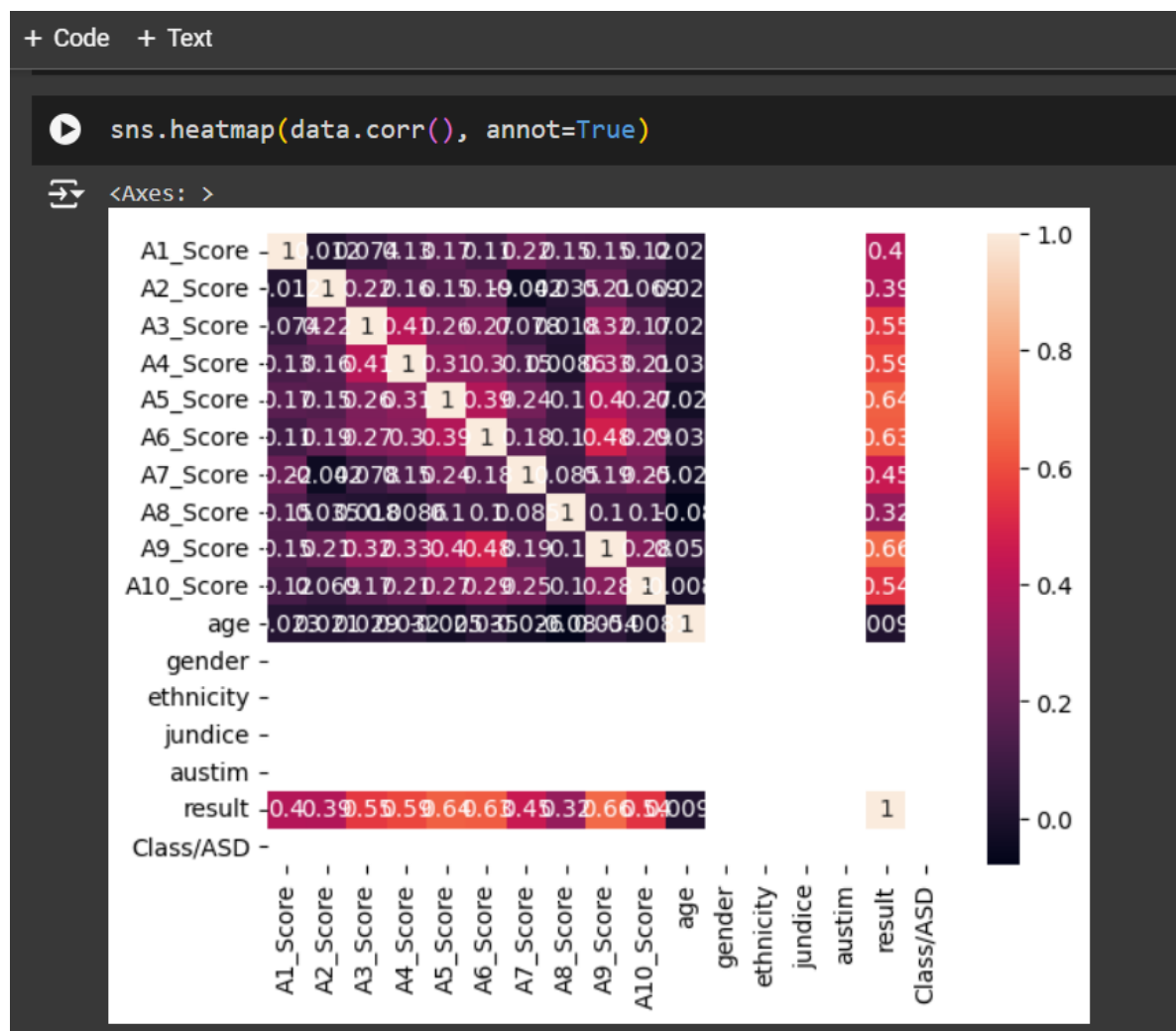
```
<Axes: xlabel='age', ylabel='count'>
```




```
[ ] for col in data.columns:
    if data[col].dtype == 'object':
        try:
            data[col] = pd.to_numeric(data[col], errors='coerce')
        except:
            pass

sns.heatmap(data.corr(), annot=True)
```

Heat Map :



Splitting the data:

```
+ Code + Text Reconnect Gemini ^
```

```
[ ] X=data[['A1_Score', 'A2_Score', 'A3_Score', 'A4_Score', 'A5_Score', 'A6_Score', 'A7_Score', 'A8_Score', 'A9_Score', 'A10_Score', 'age', 'gender', 'result', 'jundice']]
y=data['age']
X
```

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	age	gender	result	jundice
0	1	1	1	1	0	0	1	1	0	0	26	NaN	6	NaN
1	1	1	0	1	0	0	0	1	0	1	24	NaN	5	NaN
2	1	1	0	1	1	0	1	1	1	1	27	NaN	8	NaN
3	1	1	0	1	0	0	1	1	0	1	35	NaN	6	NaN
4	1	0	0	0	0	0	0	0	1	0	40	NaN	2	NaN
...
699	0	1	0	1	1	0	1	1	1	1	25	NaN	7	NaN
700	1	0	0	0	0	0	0	1	0	1	34	NaN	3	NaN
701	1	0	1	1	1	0	1	1	0	1	24	NaN	7	NaN
702	1	0	0	1	1	0	1	0	1	1	35	NaN	6	NaN
703	1	0	1	1	1	0	1	1	1	1	26	NaN	8	NaN

704 rows x 14 columns

```
+ Code + Text
```

```
[ ] X.result.describe()
```

count	704.000000
mean	4.875000
std	2.501493
min	0.000000
25%	3.000000
50%	4.000000
75%	7.000000
max	10.000000

Name: result, dtype: float64

```
[ ] y
```

0	26
1	24
2	27
3	35
4	40
...	...
699	25
700	34
701	24
702	35
703	26

Name: age, length: 704, dtype: int64

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=101)
print("Shape of X_train: ", X_train.shape)
print("Shape of y_train: ",y_train.shape)
print("Shape of X_test: ",X_test.shape)
print("Shape of y_test: ",y_test.shape)
```

```
Shape of X_train: (492, 14)
Shape of y_train: (492,)
Shape of X_test: (212, 14)
Shape of y_test: (212,)
```

```
[ ] from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

Logistic Regression :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.impute import SimpleImputer

# Assuming X is your DataFrame with missing values
imputer = SimpleImputer(strategy='mean') # Replace 'mean' with your preferred strategy
X_imputed = imputer.fit_transform(X)

# Now use the imputed data for splitting and modeling
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=0.3, random_state=101)

lgr = LogisticRegression()
lgr.fit(X_train, y_train)
n_iter_i = _check_optimiz
```

LogisticRegression
LogisticRegression()

+ Code + Text

```
[ ] pred=lgr.predict(X_test)
```

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

```
Training Set: 0.1910569105691057
Training Set: 0.08490566037735849
```

```
[ ] accuracy_LR=lgr.score(X_test,y_test)
print('Accuracy_LR:',accuracy_LR*100)
```

```
Accuracy_LR: 8.49056603773585
```

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

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

17	0.00	0.00	0.00	6
18	0.08	0.50	0.14	4
19	0.21	0.23	0.22	13
20	0.27	0.43	0.33	14
21	0.17	0.10	0.12	20
22	0.05	0.14	0.08	7
23	0.14	0.07	0.10	14
24	0.00	0.00	0.00	6
25	0.00	0.00	0.00	7
26	0.33	0.11	0.17	9
27	0.00	0.00	0.00	11
28	0.00	0.00	0.00	9
29	0.00	0.00	0.00	9
30	0.03	0.17	0.06	6
31	0.00	0.00	0.00	5
32	0.00	0.00	0.00	8
33	0.00	0.00	0.00	3
34	0.00	0.00	0.00	6
35	0.00	0.00	0.00	4
36	0.00	0.00	0.00	2
37	0.50	0.14	0.22	7
38	0.00	0.00	0.00	4
39	0.00	0.00	0.00	2
40	0.00	0.00	0.00	8
42	0.00	0.00	0.00	6
43	0.00	0.00	0.00	4
44	0.00	0.00	0.00	1
45	0.00	0.00	0.00	2
47	0.00	0.00	0.00	1

47	0.00	0.00	0.00	1
48	0.00	0.00	0.00	1
49	0.00	0.00	0.00	1
50	0.00	0.00	0.00	2
51	0.00	0.00	0.00	1
52	0.00	0.00	0.00	1
53	0.00	0.00	0.00	1
54	0.00	0.00	0.00	1
55	0.00	0.00	0.00	3
59	0.00	0.00	0.00	1
61	0.00	0.00	0.00	2

accuracy			0.08	212
macro avg	0.05	0.05	0.04	212
weighted avg	0.09	0.08	0.07	212

Support Vector Machine Classifier :

```
[ ] 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(['Training Set:',svm.score(X_test,y_test)])
```



Training Set: 0.13211382113821138
Training Set: 0.09433962264150944

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



Accuracy_SVM: 9.433962264150944

Decision Tree Classifier:

```
+ Code + Text

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

[ ] y_pred_dt=dt.predict(X_test)
    print('Training Set:',dt.score(X_train,y_train))
    print('Training Set:',dt.score(X_test,y_test))

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

[ ] accuracy_dt=accuracy_score(y_test,y_pred_dt)
    print('Accuracy_DT:',accuracy_dt*100)
```

Random Forest Classifier :

+ Code + Text

```
[ ] 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('Training Set:',rand_forest.score(X_test,y_test))
```



Training Set: 1.0
Training Set: 0.37264150943396224

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



Accuracy_RF: 97.16981132075472

KNeighbors Classifier:

+ Code + Text

```
[ ] 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)
```



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



Accuracy_KNN: 55.188679245283026

Accuracy of DataFrame:

```
File Edit View Insert Runtime Tools Help Last saved at 10:56 AM
+ Code + Text

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

Model Accuracy
0 LogisticRegression 8.490566
1 SVM 9.433962
2 DecisionTree 97.169811
3 Randomforest 37.264151
4 KNN 55.188679

models = ['LogisticRegression', 'SVM', 'DecisionTree', 'Randomforest', 'KNN']
accuracies = [accuracy_LR*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')
```

