“ANALYSIS OF AUTISM SPECTRUM DISORDER DETECTION TECHNIQUE”

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

SARNITHA G U

19BEC0575

Under the guidance of

Dr. Renuga Devi S

School Name

VIT, Vellore.

April, 2023

DECLARATION

I hereby declare that the thesis entitled "Analysis of Autism Spectrum Disorder Detection Technique" submitted by me, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Dr. Renuga Devi S.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore
Date : 11/04/2023

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Analysis of Autism Spectrum Disorder Detection Technique" submitted by **Sarnitha G U 19BEC0575**, **SENSE**, VIT, for the award of the degree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by her under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 11/04/2023

Signature of the Guide

Internal Examiner

External Examiner

Head of the Department

Electronics and Communication Engineering

ACKNOWLEDGEMENTS

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I would like to express my gratitude towards Dr. Renuga Devi S [Proctor], Dr. Thanikaiselvan, My Respectable HOD, My professors, my peers & my parents for their kind co-operation and encouragement which helped me in completion of this project.

Student Name
Sarnitha G U

Executive Summary

A neurological and developmental disorder called Autism Spectrum Disorder (ASD) frequently coexists with sensory problems such as excessive or reduced sensitivity to noises, smells, or touch. Autism Spectrum Disorder (ASD) is more common now than ever and the numbers are increasing more quickly than ever before.Early autism diagnosis can open the door to early intervention and subsequent treatments. This takes the form of therapies that can enhance a child's social and communicative abilities. As a result, as a child grows older, their quality of life may be enhanced. Screening testing for autistic features is very expensive and time-consuming. The development of machine learning (ML) and artificial intelligence has made it possible to predict autism relatively early. The project's primary goal of this thesis is to examine and analyze different machine learning algorithms, including SVM (support Vector machine), Random Forest, Decision Trees, and Logistic Regression, and evaluate the outcomes based on their efficacy and accuracy. Dataset from Kaggle and python is used to find the accuracy and outcomes of various Machine Learning algorithms.

CONTENTS

Page No.

List of Figures

List of Tables

List of Abbreviations

ML	**Machine Learning**
ASD	**Autism Spectrum Disorder**
PCA	**Principal component analysis**
SVM	**Support Vector Machine**
LR	**Logistic Regression**

Symbols and Notations

N/A

1. INTRODUCTION

Autism is a neurodevelopmental condition that impairs brain function. Although it can happen to anyone at any age, it most frequently affects toddlers under the age of three. It happens both due to the combination of hereditary and environmental factors or variables. Autism is not an illness or a disease rather it is a neurological disorder in which the affected is unable to concentrate, think, learn, focus and solve problems as simple as which neurotypical people face everyday. They experience trouble in communicating ideas through facial expression or by using gestures. It is the developmental disability with the fastest growth rate worldwide, including in India. One in every 68 children, according to the Autism Centre for Excellence (ACE), has ASD. So, it is necessary to identify this disability as early as possible.

There are numerous factors that can cause ASD in children. Among them are: Genetic - One or both parents have this disorder, another family member has autism, and Environmental - complications during pregnancy, improper or delayed vaccination of the infant, etc. The child suffering from ASD faces various challenges like

1. Lack of concentration
2. Repetition of same words
3. Limited to no interaction
4. Lack of understanding in social practices
5. Extreme ends of sensitivity in feel, touch, speech or smell

The maintenance of the subject's physical and mental health can be considerably helped by early diagnosis of this neurological disorder. With the increased use of machine learning-based models for illness prediction, it is now possible to identify disorders early based on a variety of physiological, health and psychological parameters. This element encouraged us to become more interested in the identification and examination of ASD in order to develop more effective treatment approaches. Diagnosing ASD becomes a little harder, because many neural and mental disorders share the same symptoms of ASD.

One of the most crucial actions to be taken to lessen the symptoms of autism spectrum disorder are early detection and treatment. Monitoring and observing is usually how ASD symptoms are

identified. ASD symptoms are mostly recognised in older children and adolescents who attend school by their parents and teachers. As certain ASD symptoms overlap with those of other mental health conditions, adults have a much harder time diagnosing ASD symptoms than older children and adolescents do.

1.1. OBJECTIVE

The major objective of this work is to thoroughly analyze several machine learning approaches used to predict autism symptoms, what are the various techniques used by the different researchers to recognise the child with ASD, and how effective and accurate each methodology is.

Most of the used and existing systems still rely on older machine learning techniques like logistic regression and decision trees, which are fast at processing but produce inaccurate results. There are several algorithms that employ deep learning to diagnose autism, but they demand a lot of processing power and are a little difficult to use. They provide results which have low accuracy, higher level of complexity and are usually difficult to scale.

The development of machine learning (ML) and artificial intelligence has made it possible to predict autism relatively early. The primary goal of this project is to evaluate the results of different machine learning algorithms, including SVM (support Vector machine), Random Forest, Decision Trees, and Logistic Regression, and determine which ones perform more accurately and efficiently and have low complexity and are small scale.

1.2 Motivation

Early autism diagnosis can help the affected to get early intervention. This takes the form of therapy that can enhance a child's social and communicative abilities. As a result, as a child becomes older, their quality of life may be enhanced. Programs for early intervention can begin as soon as an ASD is suspected. These may start when the child is 2 or 3 years old, when the brain is still developing. Early intervention is more likely to be effective than therapy that is initiated later in life. Early intervention programs can help children gain basic life skills that never may be delayed if not diagnosed early, such as: communication skills. social interaction, physical strength and movement, thinking and emotional skills.

The timing of an autism diagnosis may be influenced by emotional, behavioral, and social issues. Prior to receiving an autism diagnosis, late-diagnosed autistic children frequently struggled greatly with their mental health and social skills, and they frequently got worse as they approach adolescence.

1.3 Background

The purpose is to make detection of ASD easier and viable and also get results which are accurate and credible. As wrong diagnosis is more dangerous than the disorder itself, the model should be as credible as possible.

2. PROJECT DESCRIPTION AND GOALS

2.1 Block Diagram

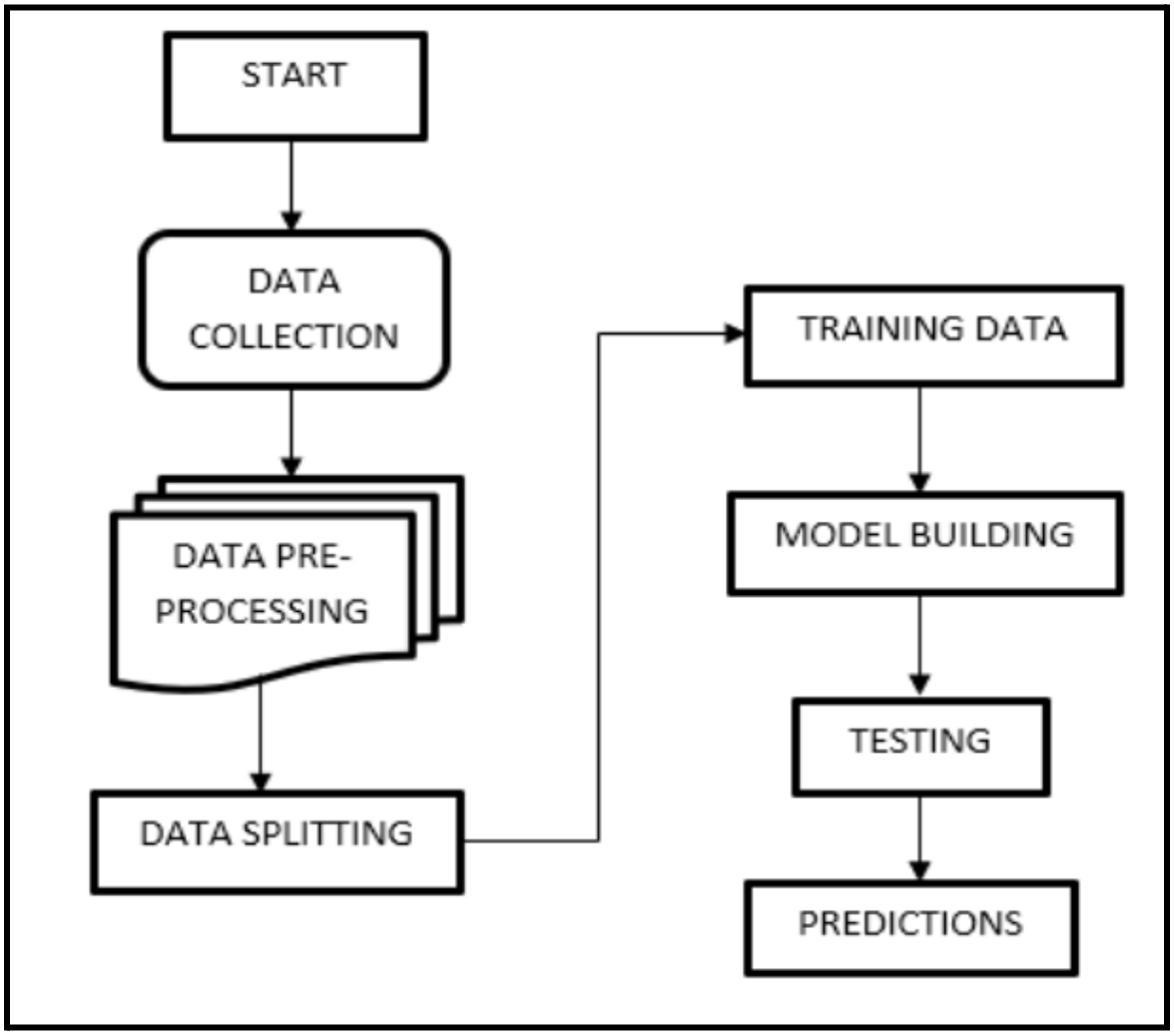

Figure 2.1: Basic Block Diagram

Data Collection: The required data for the project is collected and stored as per the need. In this project the required data is stored in csv format.

Data Pre-processing: Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. Data Preprocessing is important to get this quality data, without which it would just be a *Garbage In, Garbage Out* scenario.

Data Splitting: Data splitting is when data is divided typically into two subsets,with which one part is used to evaluate or test the data and the other to train the model.

Training Data: Training data is the data that is used to train the machine learning model to predict the outcome the model is designed to predict.

Model Building: In machine learning, generalizing and learning from training data results in the creation of a mathematical representation. The created machine learning model is then used to analyze new data and derive predictions.

Testing: Test data are used to assess the effectiveness of the algorithm used to train the computer, such as its accuracy or efficiency. Test data can also be used to determine how effective the model, which is built on training data, can predict new outcomes. Machine learning models must be improved and validated using both training and test data.

Predictions: When predicting the likelihood of a specific outcome, "prediction" refers to the result of an algorithm that has been trained on past data and applied to current data.
For every record in the new data, the algorithm will produce possible values for an unknown variable, enabling the model builder to determine what that value will most likely be.

2.2 Goal

By the end of the project, all of the mentioned machine learning models (SVM, Random Forest, Decision Tree, Logistic Regression) will be trained and tested and the results will be predicted and also the Accuracy, Sensitivity, Specificity and Precision will be calculated.

3. TECHNICAL SPECIFICATION

S/W System Configuration:

- **Operating System : Windows 11**
- **Server Script : Python**
- **IDE : Anaconda, Jupyter Notebook**
- **Packages : Sklearn, Pandas, Numpy, Matplotlib, Seaborn**

3.1 Python IDE

The project is made in Jupyter Notebook Python IDE. By structuring the data in a step-by-step fashion, such as code, photos, text, and output, Jupyter notebooks may show the analysis process in detail. A data scientist will benefit from writing down their ideas as they improve their analysis technique. The outcome can also be written down and included in the notebook.

3.2 Python Libraries

3.2.1 Numpy

The Python package NumPy is used to work with arrays. Additionally, it has matrices, fourier transform, and functions for working in the domain of linear algebra.The equivalent of arrays in Python are lists, although they take a long time to execute. The goal of NumPy is to offer array objects that are up to 50 times faster than Python lists.

3.2.2 Pandas

Python's Pandas library is used to work with data collections.

It offers tools for data exploration, cleaning, analysis, and manipulation.With the help of Pandas, we can examine large data sets and draw conclusions based on statistical principles. Pandas can organize data sets, making them understandable and useful. Rows that are irrelevant or contain incorrect data, such as empty or NULL values, can also be deleted by Pandas.

3.2.3 Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

3.2.4 Seaborn

A matplotlib-based Python data visualization package is called Seaborn. Seaborn uses Matplotlib as its foundation to plot graphs. In order to see random distributions, it will be utilized. It offers a sophisticated drawing tool for creating eye-catching and educational statistics visuals.

3.2.5 Scikit-learn

The most crucial library for the Python programming language that is frequently used in machine learning applications is called Scikit-learn. Scikit-learn is a machine learning toolkit that focuses on general-purpose, statistical, and mathematical algorithms that serve as the foundation for many machine learning technologies. The modules used in the Scikit-learn libraries are as follows:

PCA or principal component analysis - Reduce linear dimensionality with Singular Value data decomposition in order to project it into a lower dimensional space.

LogisticRegression - The module used to implement logistic regression.

RandomForestClassifier - The module used to implement random forest classifier.

DecisionTreeClassifier - The module used to implement decision tree classifier.

SVC -The module used to implement support vector machine classifier.

accuracy_score - The set of labels predicted for a sample must perfectly match the corresponding set of labels in y_true for this function to calculate subset accuracy.

confusion_matrix - to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

sensitivity_score - function to calculate subset sensitivity.

train_test_split - function to split the dataset into 2 parts for training and testing.

4 DESIGN APPROACH AND DETAILS

4.1 Design

4.1.1 Flow of the project

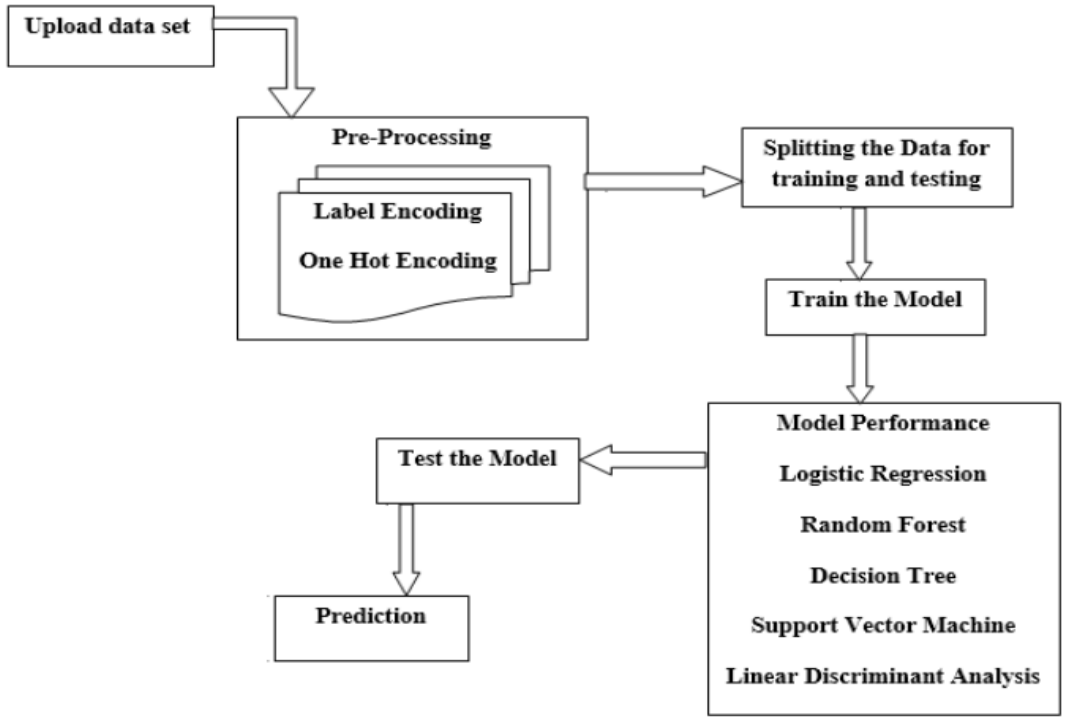

Figure 4.1 Flow of the project

4.1.2Dataset

Dataset for this research purpose has been collected from Kaggle which is publicly available. Below is the columns description of the datasets

- A1_Score to A10_Score - Score based on Autism Spectrum Quotient (AQ) 10 item screening tool
- age - Age of the patient in years
- gender - Gender of the patient
- ethnicity - Ethnicity of the patient
- jaundice - Whether the patient had jaundice at the time of birth
- autism - Whether an immediate family member has been diagnosed with autism
- contry_of_res - Country of residence of the patient
- used_app_before - Whether the patient has undergone a screening test before

- result - Score for AQ1-10 screening test
- age_desc - Age of the patient
- relation - Relation of patient who completed the test
- Class/ASD - Classified result as 0 or 1. Here 0 represents No and 1 represents Yes. This is the target column, and during submission submit the values as 0 or 1 only.

4.1.3 Preprocessing

Label encoding - Label encoding is the process of transforming labels into a numeric form so that they may be read by machines. The operation of such labels can then be better determined by machine learning techniques. It is a significant supervised learning preprocessing step for the structured dataset.

One hot encoding - One hot encoding is a method used in machine learning models to encode categorical variables as numerical values. In models that call for numerical input, it enables the incorporation of categorical variables. By giving the model extra data about the category variable, it can enhance model performance.

4.1.4 **Models**

4.1.4a **Logistic regression**

Predictive analytics and categorization frequently make use of this kind of statistical model, commonly referred to as a logit model. Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. Given that the result is a probability, the dependent variable's range is 0 to 1. In logistic regression, the odds—that is, the likelihood of success divided by the probability of failure—are transformed using the logit formula.

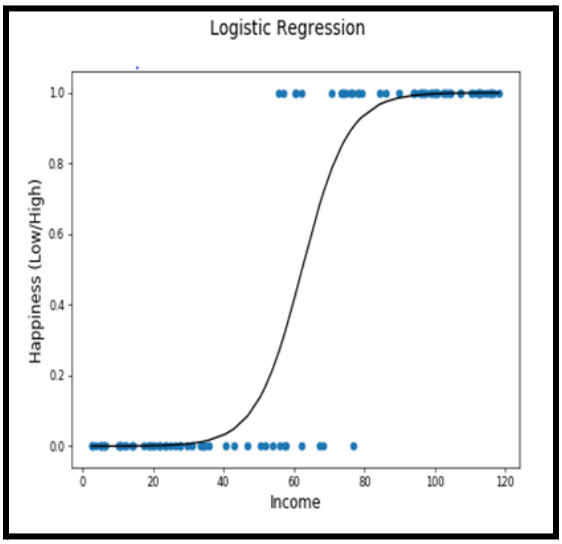

Figure 4.1.4a Logistic regression

4.1.4b DECISION TREE:

Non-parametric supervised learning techniques called decision trees are utilised for classification and regression. The objective is to learn straightforward decision rules derived from the data features in order to build a model that predicts the value of a target variable.

An upside-down decision tree is depicted, with the root at the top. On the left figure, the bold writing in black denotes an internal node or condition upon which the tree's branches and edges are based. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

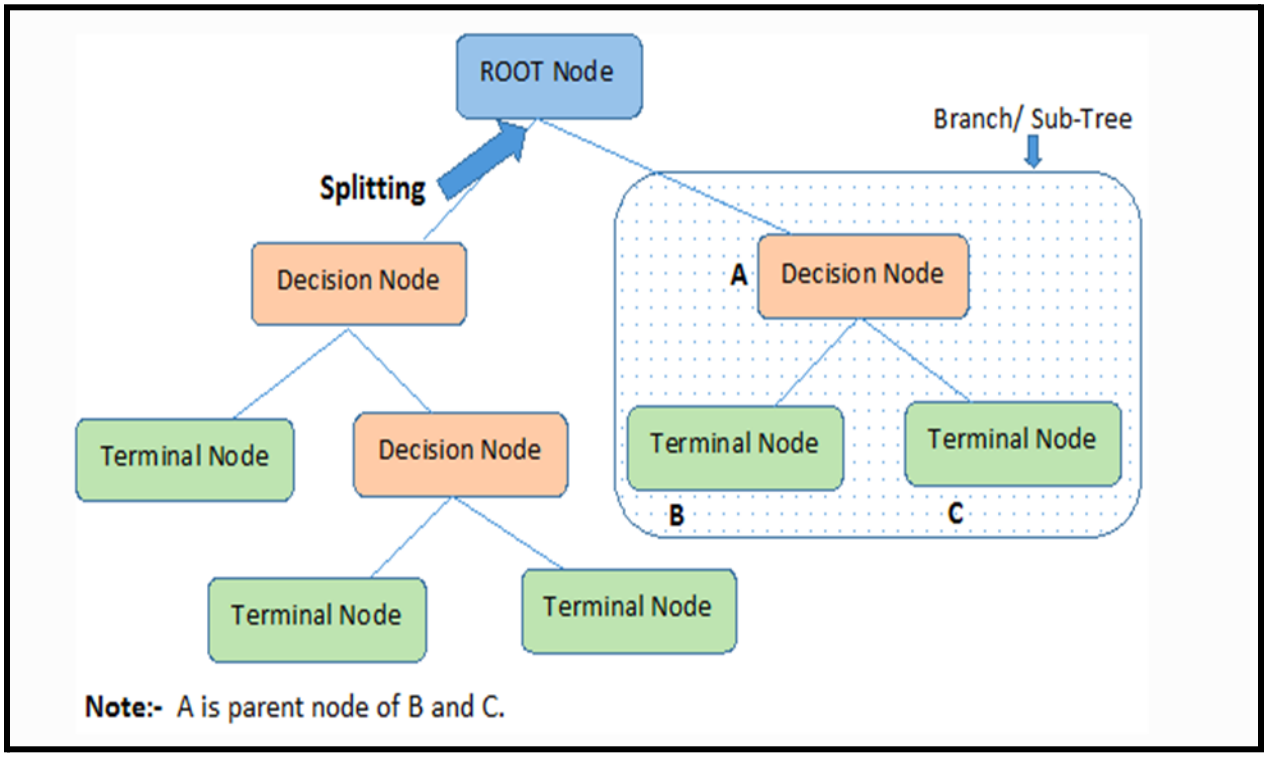

Figure 4.1.4b Block Diagram for Decision Tree Algorithm:

4.1.4c RANDOM FOREST CLASSIFIER:

A supervised learning approach called random forest is employed for both classification and regression. Yet it is primarily employed for classification issues. As is common knowledge, a forest is made up of trees, and a forest with more trees will be more sturdy. Similar to this, the random forest algorithm builds decision trees on data samples, obtains predictions from each one, and then uses voting to determine the optimal option. Because it averages the results, the ensemble method—which is superior to a single decision tree—reduces over-fitting.

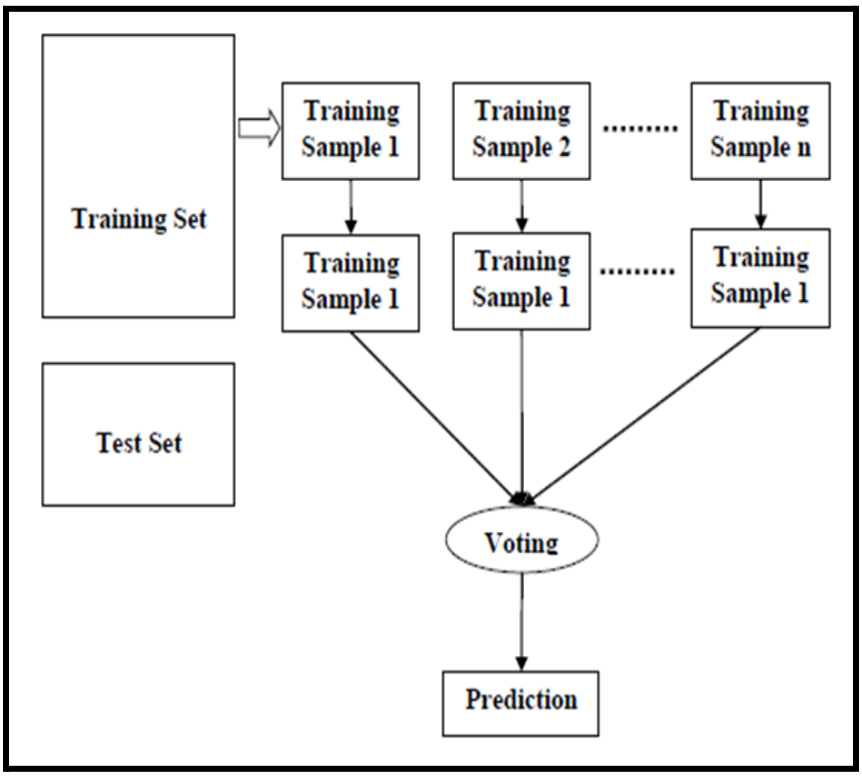

Figure 4.1.4b Block Diagram for Random Forest Classifier

4.1.4d SUPPORT VECTOR MACHINE:

Finding a hyper plane in an N-dimensional space (N is the number of features) that clearly distinguishes between the two classes of data points is the goal of the support vector machine method. There are numerous potential hyper planes from which to choose. Finding a plane with the greatest margin—that is, the greatest separation between data points from both classes—is our goal. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

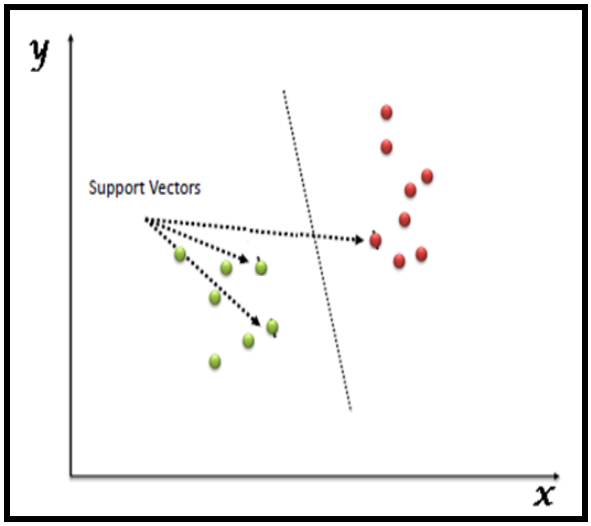

Figure 4.1.4d Block Diagram for SVM

4.2 Codes and Standards

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
from imblearn.metrics import sensitivity_score
from sklearn.model_selection import train_test_split

df=pd.read_excel("Autism child data.xlsx")
df.head()

df.isnull().sum()

df.describe()

df.shape
df.info()
df = df.dropna()

df.columns = [
"A1_Score","A2_Score","A3_Score","A4_Score","A5_Score","A6_Score","A7_Score","A
8_Score","A9_Score","A10_Score","age","gender","ethnicity","jaundice","autism","countr
y_of_res","used_app_before","result","age_desc","relation","Class_ASD",]
df.head()

sns.countplot(x='relation',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(5, 5)})

sns.countplot(x='gender',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(5, 5)})

sns.countplot(x='autism',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(5, 5)})
```

```
sns.countplot(x='country_of_res',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(7, 13)})

sns.countplot(x='Class_ASD',data=df)
plt.xticks(rotation=90)

sns.countplot(x='jaundice',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(4, 4)})

sns.countplot(x='ethnicity',data=df)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(5,5)})

df = df[df['age']!='?']
org_data = df[[ "A1_Score", "A2_Score", "A3_Score", "A4_Score", "A5_Score",
"A6_Score", "A7_Score", "A8_Score", "A9_Score", "A10_Score", "age", ]]

label_encoded_data = df[["gender", "autism", "jaundice", "Class_ASD"]]
label_encoded_data["gender"] = label_encoded_data["gender"].apply(lambda x: 1 if x ==
"m" else 0)
label_encoded_data["autism"] = label_encoded_data["autism"].apply(lambda x: 1 if x ==
"yes" else 0)
label_encoded_data["jaundice"] = label_encoded_data["jaundice"].apply(lambda x: 1 if x
== "yes" else 0)
label_encoded_data["Class_ASD"] = label_encoded_data["Class_ASD"].apply(lambda x:
1 if x == "YES" else 0)
one_hot_encoded_data = df[["ethnicity"]]
one_hot_encoded_data = pd.get_dummies(one_hot_encoded_data)

fixed_data = pd.concat([org_data,label_encoded_data,one_hot_encoded_data],axis=1)

fixed_data.head()

X = fixed_data.drop(columns=['Class_ASD'])
y = fixed_data[['Class_ASD']]

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=42)
X_test.shape
y_train.shape

lr=LogisticRegression()
lr.fit(X_train,y_train)
y_pred=lr.predict(X_test)
```

```
lraccuracy=accuracy_score(y_test, y_pred)
lraccuracy
lrsensitivity=sensitivity_score(y_test, y_pred)
lrsensitivity

train_acc = lr.score(X_train, y_train)
test_acc = lr.score(X_test, y_test)
print("Training Accuracy: ", train_acc)
print("Test Accuracy: ", test_acc)
sns.barplot(x=["Training", "Test"], y=[train_acc, test_acc])

conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion matrix:\n", conf_matrix)
sns.heatmap(conf_matrix, annot=True, cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()

RF=RandomForestClassifier(n_estimators=1000, max_depth=None, min_samples_split=2,
random_state=0)
RF.fit(X_train,y_train)
y_pred=RF.predict(X_test)
RFaccuracy=accuracy_score(y_test, y_pred)
RFaccuracy
RFsensitivity=sensitivity_score(y_test, y_pred)
RFsensitivity

train_acc_RF = RF.score(X_train, y_train)
test_acc_RF = RF.score(X_test, y_test)
print("Training Accuracy: ", train_acc_RF)
print("Test Accuracy: ", test_acc_RF)
sns.barplot(x=["Training", "Test"], y=[train_acc_RF, test_acc_RF])

conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion matrix:\n", conf_matrix)
sns.heatmap(conf_matrix, annot=True, cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()

DT=DecisionTreeClassifier(max_depth=None, min_samples_split=2, random_state=0)
DT.fit(X_train,y_train)
y_pred=DT.predict(X_test)
DTaccuracy=accuracy_score(y_test, y_pred)
DTaccuracy
DTsensitivity=sensitivity_score(y_test, y_pred)
```

DTsensitivity

```
train_acc_DT = DT.score(X_train, y_train)
test_acc_DT = DT.score(X_test, y_test)
print("Training Accuracy: ", train_acc_DT)
print("Test Accuracy: ", test_acc_DT)
sns.barplot(x=["Training", "Test"], y=[train_acc_DT, test_acc_DT])

conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion matrix:\n", conf_matrix)
sns.heatmap(conf_matrix, annot=True, cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()

svc=SVC(kernel='linear')
svc.fit(X_train,y_train)
y_pred=svc.predict(X_test)
svcaccuracy=accuracy_score(y_test, y_pred)
svcaccuracy
svcsensitivity=sensitivity_score(y_test, y_pred)
svcsensitivity

train_acc_svc = svc.score(X_train, y_train)
test_acc_svc = svc.score(X_test, y_test)
print("Training Accuracy: ", train_acc_svc)
print("Test Accuracy: ", test_acc_svc)
sns.barplot(x=["Training", "Test"], y=[train_acc_svc, test_acc_svc])

conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion matrix:\n", conf_matrix)
sns.heatmap(conf_matrix, annot=True, cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()

svc.predict([[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]])

lr.predict([[1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]])

data=pd.DataFrame({"Algorithm":['SUPPORT VECTOR MACHINE','LOGISTIC
REGRESSION', 'RANDOM FOREST', 'DECISION TREE'],
                   "Accuracy":[svcaccuracy, lraccuracy, RFaccuracy, DTaccuracy]})

data
```

```
x=data['Algorithm']
y=data['Accuracy']
sns.barplot(x, y, data=data)
plt.xticks(rotation=90)
sns.set(rc={'figure.figsize':(4, 4)})


from sklearn.metrics import roc_curve, roc_auc_score

plt.figure(figsize=(12, 8))

# Logistic Regression (overlapping)
lr_probs = lr.predict_proba(X_test)[:, 1]
lr_fpr, lr_tpr, _ = roc_curve(y_test, lr_probs)
lr_auc = roc_auc_score(y_test, lr_probs)
plt.plot(lr_fpr, lr_tpr, label='Logistic Regression (AUC = %0.2f)' % lr_auc)

# Random Forest (overlapping)
rf_probs = RF.predict_proba(X_test)[:, 1]
rf_fpr, rf_tpr, _ = roc_curve(y_test, rf_probs)
rf_auc = roc_auc_score(y_test, rf_probs)
plt.plot(rf_fpr, rf_tpr, label='Random Forest (AUC = %0.2f)' % rf_auc)

# Decision Tree
dt_probs = DT.predict_proba(X_test)[:, 1]
dt_fpr, dt_tpr, _ = roc_curve(y_test, dt_probs)
dt_auc = roc_auc_score(y_test, dt_probs)
plt.plot(dt_fpr, dt_tpr, label='Decision Tree (AUC = %0.2f)' % dt_auc)

# Support Vector Machine (overlapping)
svc_probs = svc.decision_function(X_test)
svc_fpr, svc_tpr, _ = roc_curve(y_test, svc_probs)
svc_auc = roc_auc_score(y_test, svc_probs)
plt.plot(svc_fpr, svc_tpr, label='Support Vector Machine (AUC = %0.2f)' % svc_auc)

# Plot formatting
plt.plot([0, 1.5], [0, 1.5], 'k--', label='Random Guessing')
plt.xlim([0, 1.5])
plt.ylim([0, 1.5])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve')
plt.legend(loc="lower right")
plt.show()
```

5. SCHEDULE, TASKS AND MILESTONES

There are four major milestones with respect to the project. The four milestones are:

1. Preprocessing and splitting the data in the dataset.

2. Training the dataset with the above mentioned ML algorithms.

3.Calculating the accuracy, sensitivity, specificity and precision using the confusion matrix.

4. Plotting and comparing the training and testing accuracy.

6. PROJECT DEMONSTRATION

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	result
count	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000	498.000000
mean	0.682731	0.514056	0.746988	0.570281	0.751004	0.710843	0.622490	0.477912	0.542169	0.730924	6.345382
std	0.465881	0.500305	0.435175	0.495534	0.432866	0.453827	0.485252	0.500014	0.502738	0.443926	2.360214
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	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	5.000000
50%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	7.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	8.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	10.000000

Figure 6 Average of the data in the dataset

6.1 Logistic Regression

```
In [27]: lraccuracy=accuracy_score(y_test, y_pred)
         lraccuracy

Out[27]: 0.9865771812080537

In [28]: lrsensitivity=sensitivity_score(y_test, y_pred)
         lrsensitivity

Out[28]: 1.0
```

Figure 6.1.1 Accuracy and sensitivity of Logistic Regression

```
In [29]: train_acc = lr.score(X_train, y_train)
         test_acc = lr.score(X_test, y_test)

         print("Training Accuracy: ", train_acc)
         print("Test Accuracy: ", test_acc)

         sns.barplot(x=["Training", "Test"], y=[train_acc, test_acc])

         Training Accuracy:  1.0
         Test Accuracy:  0.9865771812080537
```

Figure 6.1.2 Training and Test accuracy of LR

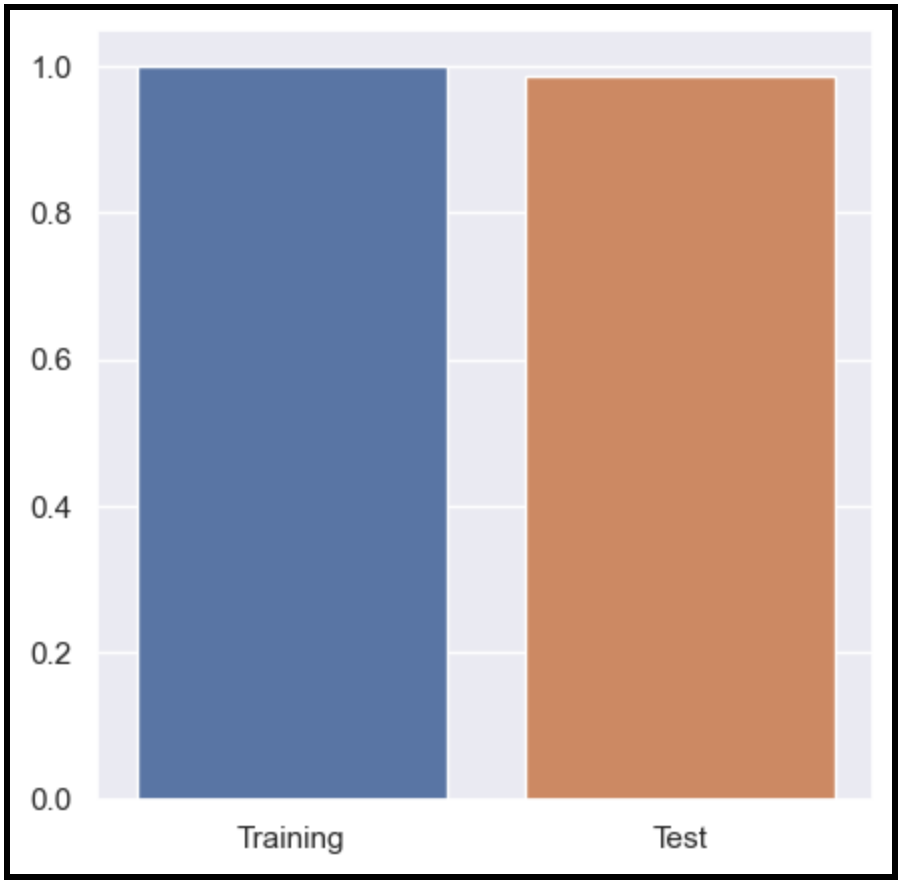

Figure 6.1.3 Training and Test accuracy plot of LR

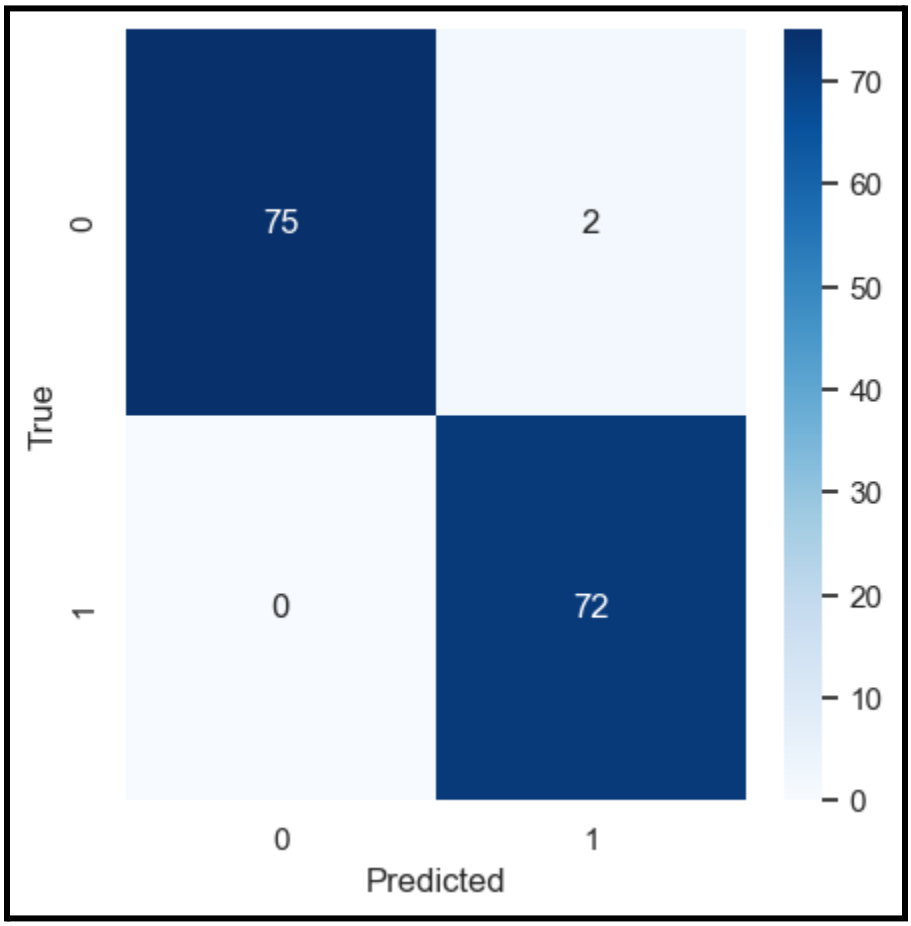

Figure 6.1.4 Confusion matrix of LR

6.2 Random Forest

```
In [32]: y_pred=RF.predict(X_test)
         RFaccuracy=accuracy_score(y_test, y_pred)
         RFaccuracy

Out[32]: 0.9865771812080537

In [33]: RFsensitivity=sensitivity_score(y_test, y_pred)
         RFsensitivity

Out[33]: 0.9722222222222222
```

Figure 6.2.1 Accuracy and sensitivity of Random Forest

```
In [34]: train_acc_RF = RF.score(X_train, y_train)
         test_acc_RF = RF.score(X_test, y_test)

         print("Training Accuracy: ", train_acc_RF)
         print("Test Accuracy: ", test_acc_RF)

         sns.barplot(x=["Training", "Test"], y=[train_acc_RF, test_acc

         Training Accuracy:  1.0
         Test Accuracy:  0.9865771812080537
```

Figure 6.2.2Training and Test accuracy of Random Forest

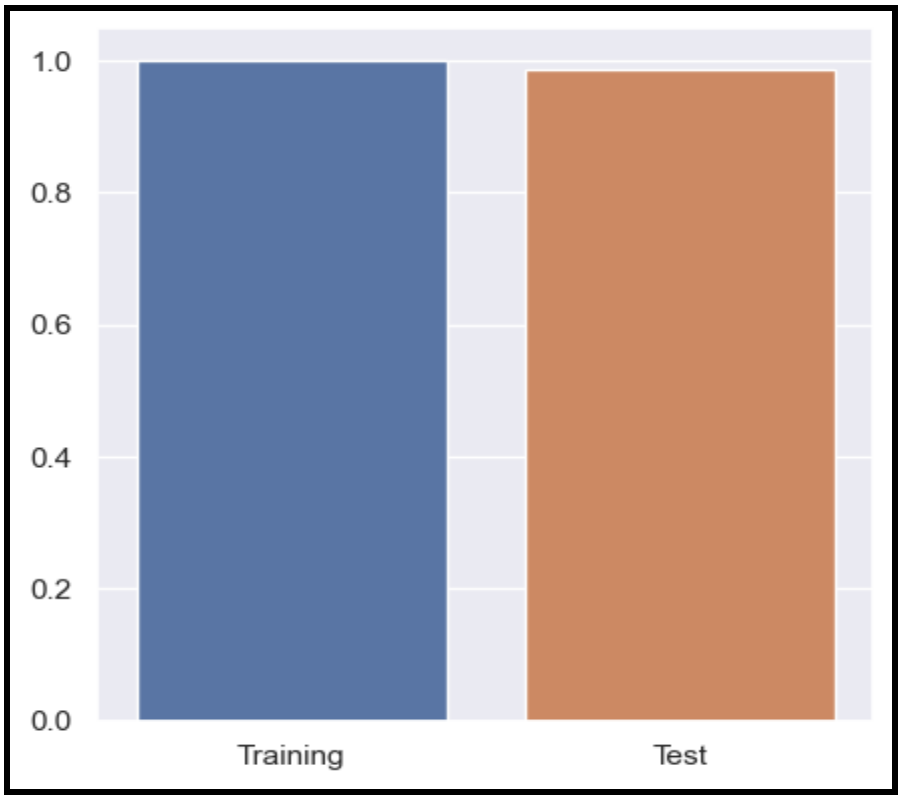

Figure 6.2.3 Training and Test accuracy plot of Random Forest

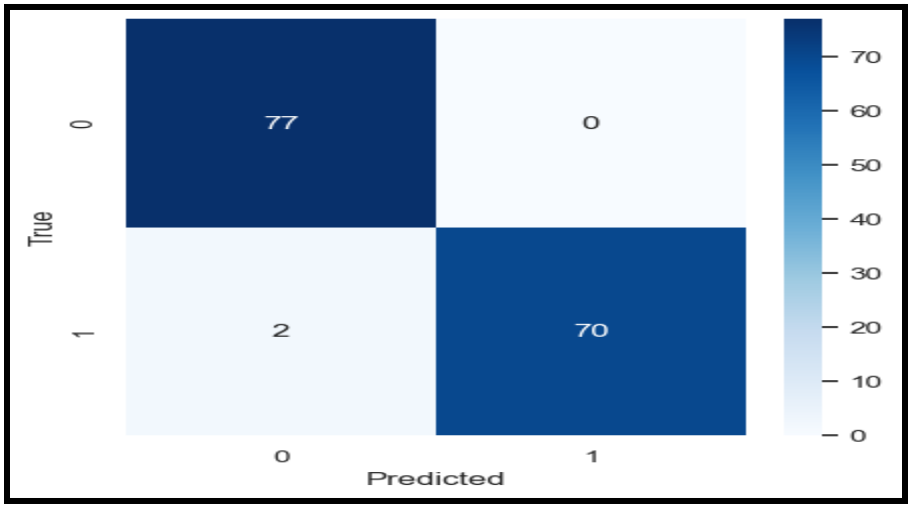

Figure 6.2.4 Confusion matrix of SVM

6.3 Decision Tree

```
In [37]: y_pred=DT.predict(X_test)
         DTaccuracy=accuracy_score(y_test, y_pred)
         DTaccuracy

Out[37]: 0.9731543624161074

In [49]: DTsensitivity=sensitivity_score(y_test, y_pred)
         DTsensitivity

Out[49]: 1.0
```

Figure 6.3.1 Accuracy and sensitivity of Decision Tree

```
In [38]: train_acc_DT = DT.score(X_train, y_train)
         test_acc_DT = DT.score(X_test, y_test)

         print("Training Accuracy: ", train_acc_DT)
         print("Test Accuracy: ", test_acc_DT)

         sns.barplot(x=["Training", "Test"], y=[train_acc_DT, test_acc

         Training Accuracy:  1.0
         Test Accuracy:  0.9731543624161074
```

Figure 6.3.2 Training and Test accuracy of Decision tree

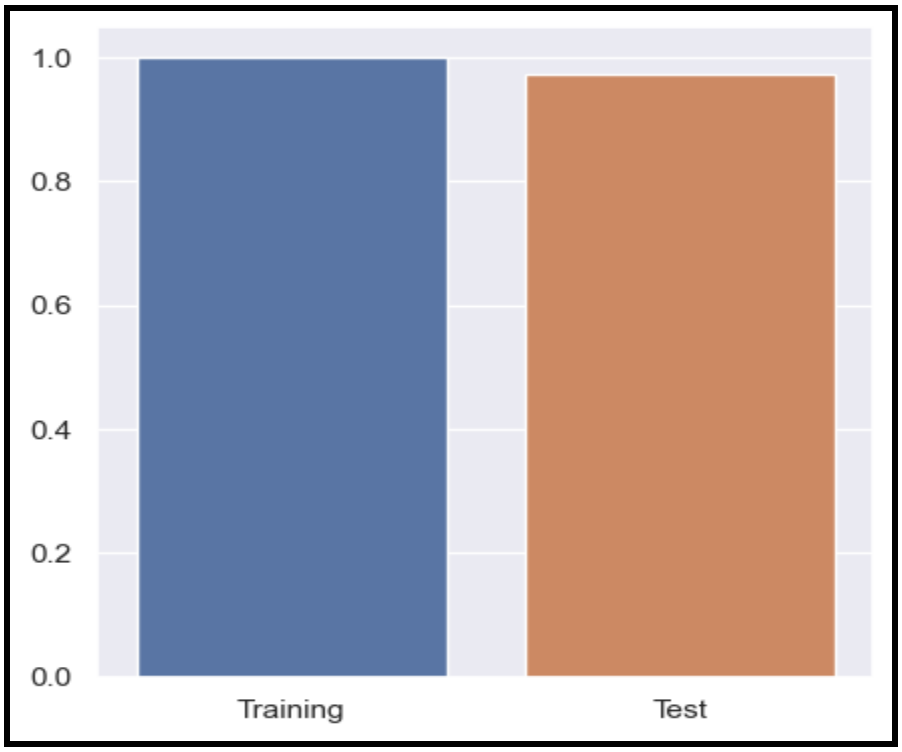

Figure 6.3.3 Training and Test accuracy plot of Decision tree

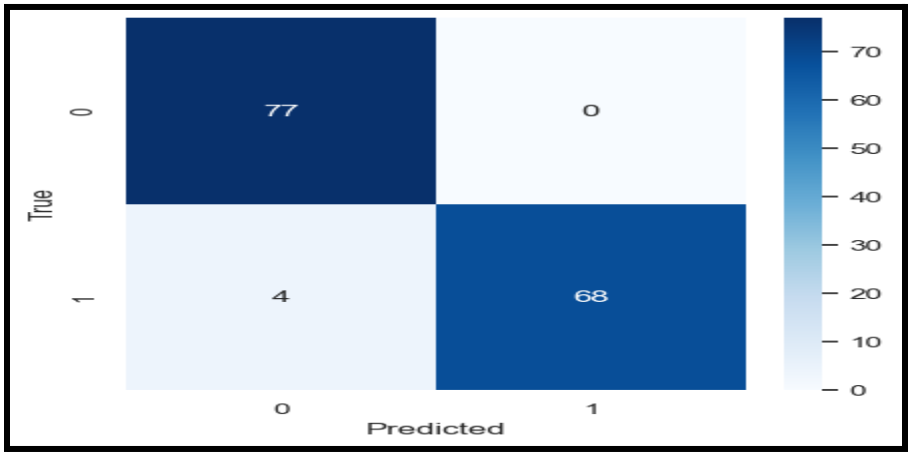

Figure 6.3.4 Confusion matrix of Decision tree

6.4 SVM

```
In [41]: y_pred=svc.predict(X_test)
         svcaccuracy=accuracy_score(y_test, y_pred)
         svcaccuracy

Out[41]: 1.0

In [50]: svcsensitivity=sensitivity_score(y_test, y_pred)
         svcsensitivity

Out[50]: 1.0
```

Figure 6.4.1 Accuracy and sensitivity of SVM

```
In [42]: train_acc_svc = svc.score(X_train, y_train)
         test_acc_svc = svc.score(X_test, y_test)

         print("Training Accuracy: ", train_acc_svc)
         print("Test Accuracy: ", test_acc_svc)

         sns.barplot(x=["Training", "Test"], y=[train_acc_svc, test_ac

         Training Accuracy:  1.0
         Test Accuracy:  1.0
```

Figure 6.4.2 Training and Test accuracy of SVM

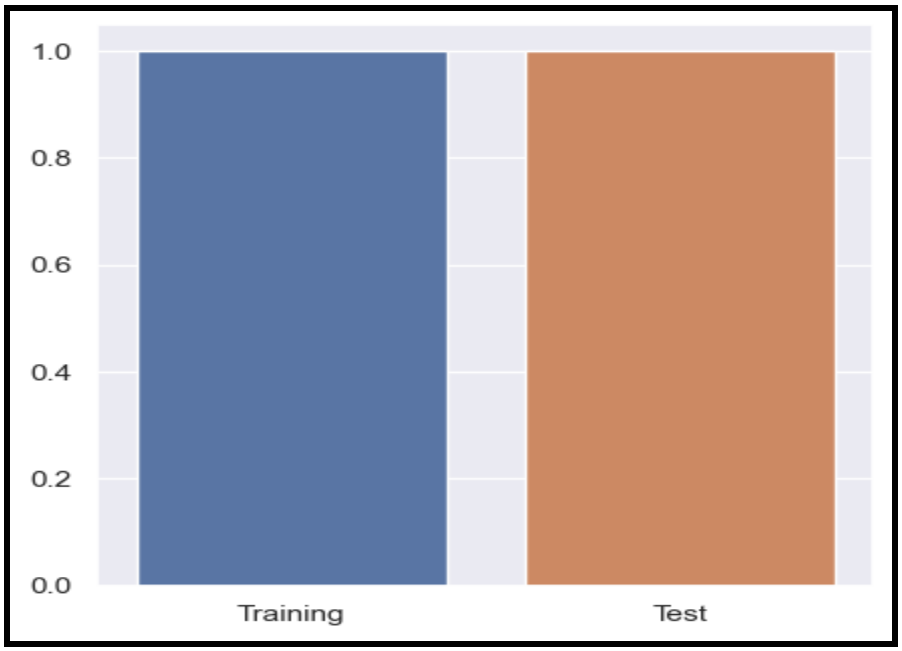

Figure 6.4.3 Training and Test accuracy plot of SVM

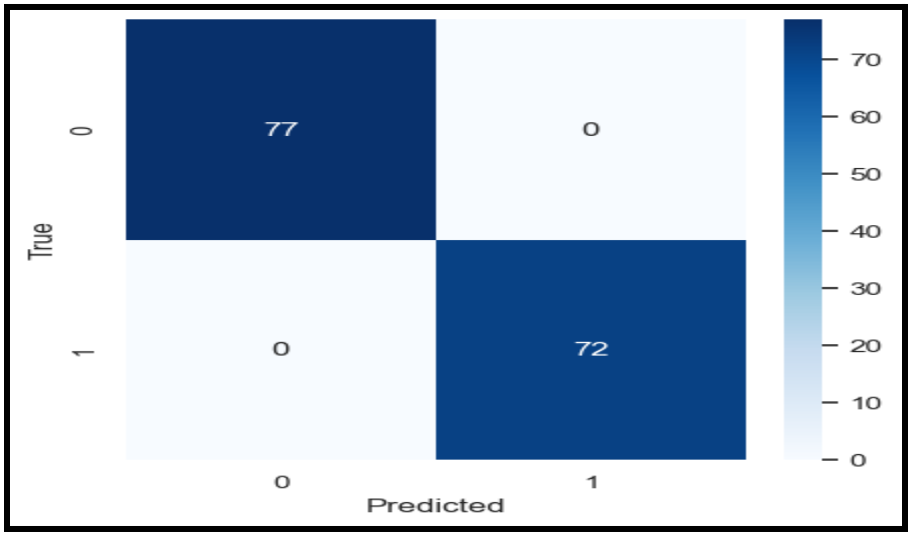

Figure 6.1.4 Confusion matrix of SVM

7. Result and Discussion

Calculations

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Sensitivity = TP / (TP + FN)

Specificity = TN / (TN + FP)

Precision = TP / (TP + FP)

From the obtained confusion matrix for each ML algorithm model analyzed in this project the above mentioned values can be calculated.

1. Logistic Regression

TP = 75 FP = 0

FN = 2 TN = 72

2. Random Forest

TP = 77 FP = 2

FN = 0 TN = 70

2. Random Forest

TP = 77 FP = 2

FN = 0 TN = 70

3. Decision Tree

TP = 77 FP = 4

FN = 0 TN = 68

3.SVC

TP = 77 FP = 0

FN = 0 TN = 72

The results concluded from the project is

	Algorithm	Accuracy	Sensitivity	Specificity	Precision
0	SUPPORT VECTOR MACHINE	1.00000	1.00000	1.00000	1.00000
1	LOGISTIC REGRESSION	0.98657	1.00000	0.97402	0.97292
2	RANDOM FOREST	0.986577	0.97222	1.00000	1.00000
3	DECISION TREE	0.973154	0.94444	1.00000	1.00000

Table 7 Result comparison

8.SUMMARY

Autism is one of the vital issues which cannot be prevented but can be treated and it's a big challenge for any family to have a child with autistic disorder. Therefore, it's important to diagnose it, as early as possible. One of the major issues in today's research is to have improved diagnostic tools to have faster, effective and accurate results. Machine learning techniques have shown favorable results. Machine learning with its leverage methods can be used to diagnose ASD. Use of machine Learning is to create algorithms that are vigorous and have engineer instruments. By using different ML algorithms researchers were able to build such a model which shows improved results in terms of accuracy and precisions. In this analysis various classifiers have been used like SVM, Random Forest, decision tree, Logistic Regression and so on. Among all, SVM has shown the best result with an accuracy of 100%. This might not be as accurate as the dataset used in this project is relatively smaller than the data all around the world.

9.REFERENCES

Omar, K. S., Mondal, P., Khan, N. S., Rizvi, M. R. K., & Islam, M. N. (2019, February). A machine learning approach to predict autism spectrum disorder. In 2019 International conference on electrical, computer and communication engineering (ECCE) (pp. 1-6). IEEE

Raj, S., & Masood, S. (2020). Analysis and detection of autism spectrum disorder using machine learning techniques. *Procedia Computer Science*, *167*, 994-1004.

Nogay, H. S., & Adeli, H. (2020). Machine learning (ML) for the diagnosis of autism spectrum disorder (ASD) using brain imaging. *Reviews in the Neurosciences*, *31*(8), 825-841.

Ahmed, I. A., Senan, E. M., Rassem, T. H., Ali, M. A., Shatnawi, H. S. A., Alwazer, S. M., & Alshahrani, M. (2022). Eye tracking-based diagnosis and early detection of autism spectrum disorder using machine learning and deep learning techniques. *Electronics*, *11*(4), 530.

Liu, Wenbo & Li, Ming & Yi, Li. (2016). Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework. Autism research : official journal of the International Society for Autism Research.

Md. Shahriare Satu , Farha Farida Sathi, Md. Sadrul Arifen, Md. Hanif Ali and Mohammad Ali Moni, “Early Detection of Autism by Extracting Features:A Case Study in Bangladesh”, “International Conference on Robotics,Electrical and Signal Processing Techniques” , 2019

JA Kosmicki, V Sochat, M Duda and DP Wall, “Searching for a minimal set of behaviors for autism detectionthrough feature selection-based machine learning”, www.nature.com/tp, vol 5, 2014, doi:10.1038/tp.2015.7

Rello, L., & Ballesteros, M. (2015, May). Detecting readers with dyslexia using machine learning with eye tracking measures. In *Proceedings of the 12th International Web for All Conference* (pp. 1-8).

Yetton, B. D., Niknazar, M., Duggan, K. A., McDevitt, E. A., Whitehurst, L. N., Sattari, N., & Mednick, S. C. (2016). Automatic detection of rapid eye movements (REMs): A machine learning approach. *Journal of neuroscience methods*, *259*, 72-82.

Ahmed, I. A., Senan, E. M., Rassem, T. H., Ali, M. A., Shatnawi, H. S. A., Alwazer, S. M., & Alshahrani, M. (2022). Eye tracking-based diagnosis and early detection of autism spectrum disorder using machine learning and deep learning techniques. *Electronics*, *11*(4), 530.

Thabtah, F., & Peebles, D. (2020). A new machine learning model based on induction of rules for autism detection. *Health informatics journal*, *26*(1), 264-286.

Wall, D. P., Kosmicki, J., Deluca, T. F., Harstad, E., & Fusaro, V. A. (2012). Use of machine learning to shorten observation-based screening and diagnosis of autism. *Translational psychiatry*, *2*(4), e100-e100.

Heinsfeld, A. S., Franco, A. R., Craddock, R. C., Buchweitz, A., & Meneguzzi, F. (2018). Identification of autism spectrum disorder using deep learning and the ABIDE dataset. *NeuroImage: Clinical*, *17*, 16-23.

Abbas, H., Garberson, F., Glover, E., & Wall, D. P. (2018). Machine learning approach for early detection of autism by combining questionnaire and home video screening. *Journal of the American Medical Informatics Association*, *25*(8), 1000-1007.

Vaishali, R., & Sasikala, R. (2018). A machine learning based approach to classify autism with optimum behaviour sets. *Int. J. Eng. Technol*, *7*(4), 18.