DETECTION OF AUTISM SPECTRUM DISORDER APPLYING DEEP NEURAL NETWORK

A research
Submitted in partial fulfillment of the requirements for the Degree of
Masters of Science in Computer Science and Engineering.

Submitted by

NASRIN AKTER	2018-3-96-004

Supervised by

Dr. Md. Nawab Yousuf Ali
Associate Professor
Department of Computer Science and Engineering
East West University

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

September 2019

CANDIDATES' DECLARATION

I, hereby, declare that the research presented in this report is the outcome of the investigation performed by me under the supervision of Dr. Md. Nawab Yousuf Ali, Associate Professor, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh.

It is also declared that neither this research nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Countersigned

_____

Dr. Md. Nawab Yousuf Ali

_____

Nasrin Akter
ID: 2018-3-96-004

ACKNOWLEDGEMENT

I recognize myself being fortunate enough to have completed this academic thesis whilst in sound health. In the beginning, I express gratitude to my parents, in whose patronage I could come this far. Most importantly, I thank my supervisor Dr. Md. Nawab Yousuf Ali, whose constant guidance proved invaluable. Whenever I came up with complicated issues, he guided me the simple way to resolve the issues. Besides, I am grateful to all our course directors for providing me with necessary contemporary insights from the field of system development and implementations.

Dhaka

September 2019

Nasrin Akter

ABSTRACT

Autistic Spectrum Disorder (ASD) is a neurological condition associated with communication, repetitive and social challenges. ASD screening is the process of detecting potential autistic traits in individuals using tests conducted by the medical professionals, caregivers or parents. These tests often contain large numbers of items to be covered by the user and they generate score based on scoring functions designed by psychologists and behavioral scientists. Potential technologies that may improve the reliability and accuracy of ASD tests are Artificial Intelligence and Machine Learning. . The goal of this paper is to research and develop a prediction model to detect autism among toddlers. This paper presents a Classifier to detect ASD applying Deep Neural Network. The neural network applied is of four layers with three ReLU hidden layers and a sigmoid classification layer. The best accuracy of 92.73% obtained on the cross validation set upon trained parameters. Additionally, perfect recall 97.99%, precision of 99.32% and F Score of 98.65% has been achieved. Since machine learning is the future for the next generation's problem solicitor to detect and decide key decisions, hope this work contributes significantly in Autism Spectrum Disorder detection.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1:

INTRODUCTION

Autism spectrum disorder (ASD) is a developmental disorder that involves persistent challenges in social interaction, speech and nonverbal communication, and restricted/repetitive behaviors. These problems can be mild, severe, or somewhere in between. The effects of ASD and the severity of symptoms are different in each person. Although autism can be diagnosed at any age, it is said to be a “developmental disorder” because symptoms generally appear in the first two years of life. According to the Diagnostic and Statistical Manual of Mental Disorders (DSM-5), a guide created by the American Psychiatric Association used to diagnose mental disorders, people with ASD have difficulty with communication and interaction with other people, restricted interests and repetitive behaviors, symptoms that hurt the person’s ability to function properly in school, work, and other areas of life.

Early diagnosis is important, because early treatment can make a big difference. Until recently, experts talked about different types of autism, such as autistic disorder, Asperger’s syndrome, pervasive developmental disorder not otherwise specified (PDD-NOS). But now they are all called “autism spectrum disorders’’ because there is wide variation in the type and severity of symptoms people experience. ASD occurs in all ethnic, racial, and economic groups. Although ASD can be a lifelong disorder, treatments and services can improve a person’s symptoms and ability to function.

1.1 The Prevalence and Severity of ASD

ASDs continue to be an important public health concern. It is estimated that worldwide 1 in 160 children has an ASD. This estimate represents an average figure,

and reported prevalence varies substantially across studies. Some well-controlled studies have, however, reported figures that are substantially higher. The prevalence of ASD in many low- and middle-income countries is so far unknown.

Based on epidemiological studies conducted over the past 50 years, the prevalence of ASD appears to be increasing globally. There are many possible explanations for this apparent increase, including improved awareness, expansion of diagnostic criteria, better diagnostic tools and improved reporting.

1.2 Importance of Early identification of ASD

Early detection is the key in helping a child with autism live a more normal life in society. Since autism can be seen as early as 18 months of age, children should be watched throughout their development for any warning signs of autism. Early identification of an ASD is crucial, as it means early intervention services can begin, making a huge impact on a child's behavior, functioning and future well-being. Without early intervention, the symptoms of autism can worsen, resulting in more costly treatment over the course of a lifetime. The estimated lifetime cost of caring for someone with autism ranges from $1.4-2.4 million, but this cost can be reduced by two-thirds through early diagnosis and intervention.

Currently, the average age of diagnosis is between 3 and 6 years of age, though some children can be diagnosed as young as 2. It is important for parents to discuss the diagnosis with their medical practitioner(s) and devise a treatment plan that best addresses the needs of child and family. The Autism Society encourages applied research to identify the most effective early intervention approaches and also encourage the sharing of research advances worldwide so all people with autism can benefit.

Research has shown that early intervention can improve a child's overall development. Children who receive autism-appropriate education and support at key developmental stages are more likely to gain essential social skills and react better in society. Essentially, early detection can provide an autistic child with the potential for

a better life. Parents of autistic children can learn early on how to help their child improve mentally, emotionally, and physically throughout the developmental stages with assistance from specialists.

Lastly, catching autism and working through it early also benefits parental relationships. The strain of caring for an autistic child can be an everyday challenge, but with early preparation and intervention, parents can prepare themselves for the road ahead emotionally and mentally.

1.3 Screening and Medical Diagnosis of ASD

Diagnosing autism spectrum disorder (ASD) can be difficult, since there is no medical test, like a blood test, to diagnose the disorders. Doctors look at the child's behavior and development to make a diagnosis.

ASD can sometimes be detected at 18 months or younger. By age 2, a diagnosis by an experienced professional can be considered very reliable. However, many children do not receive a final diagnosis until much older. This delay means that children with an ASD might not get the help they need.

Diagnosing an ASD takes two steps:

- **Developmental Screening:** Developmental screening is a short test to tell if children are learning basic skills when they should, or if they might have delays. During developmental screening the doctor might ask the parent some questions or talk and play with the child during an exam to see how she learns, speaks, behaves, and moves.
- **Comprehensive Diagnostic Evaluation:** The second step of diagnosis is a comprehensive evaluation. This thorough review may include looking at the child's behavior and development and interviewing the parents. It may also include a hearing and vision screening, genetic testing, neurological testing, and other medical testing.

1.4 Organization of the Project

The organization of this project dictates the second chapter as presentations of related work, the third chapter as an introduction to theoretical foundations, the fourth chapter as preparation of the dataset, the fifth chapter as a narration of proposed methodology, the sixth chapter as a tabulation of results and the final section as concluding remarks.

Chapter 2:

LITERATURE REVIEW

Rahman, 2010 urged that Increasing Intelligibility within the Speech of the Autistic Children by an Interactive Computer Game. There is no definite treatment for autism. Serving to autistic children by providing games and teaching facilities to improve their skills. In the year 2013 Santos examines the first detection of Autism means that taking the symptoms of patient during childhood supported by preverbal vocalization by using the classification technique supervised learning SVM (support vector machine). Chaminade, 2012 started a shot to use MRI study of young adults with autism interacting with a humanoid robot (Emily T. Prud'hommeaux 2011) examines the difficulties for classification of non-standardized text of machine learning techniques.

Bishop-Fitzpatrick et al. (2018) leveraged ICD-9 codes, V-codes, and E-codes from the electronic health records of 91 decedents with an ASD (or related) diagnosis and 6186 control decedents to build a random forest classifier. The goal of this study was not only to distinguish ASD from the control decedent but also to examine the lifetime health problems of those with ASD. From the first RF model, the top 50 ICD-9 codes, V-codes, and E-codes were chosen and were used to build a second, smaller random forest model. The model had an accuracy of 93%, sensitivity of 75%, specificity of 94%, and an AUC of 0.88. The 50 codes were then ranked in order of importance. The authors report that, overall, decedents with an ASD diagnosis have higher rates of nearly all 50 ICD-9 codes, V-codes, and E-codes.

Bussu et al. (2018) used longitudinal data captured at multiple points in development (8 and 14 months of age) for high-risk siblings to increase accuracy of predicting ASD diagnosis at 36 months.

Chen et al. (2015) used machine learning models, including RF, to analyze neuroimaging data for diagnostic classification purposes. Low-motion resting-state

functional MRI (rs-fMRI) scans were used for a sample of 126 individuals with ASD and 126 individuals with TD. Participants were matched based on age, non-verbal IQ, and head motion. Diagnostic classification was based on a matrix of functional connectedness between 220 identified regions of interest. Using the top 100 regions of interest, an RF produced the greatest level of accuracy of the models tested for diagnostic classification at 91%, with a sensitivity of 89% and a specificity of 93%. When applied to the top 10 regions of interest, the RF achieved an accuracy of 75%, with a sensitivity of 75% and a specificity of 75%. The high number of regions required to produce a strong accuracy in detecting ASD may imply that brain biomarkers for ASD are scattered rather than localized.

Hyde et al. (2018) utilized a decision tree to predict a path for individuals with ASD to successfully find employment. The model was built using 17 independent variables created from the responses of 154 representatives of various employers who have hired an individual with ASD in the past and 142 from those who have never hired an individual with ASD. The model was able to predict whether an employer has hired an individual with ASD with 75% accuracy and 82% specificity as well as identify some important features that lead to the decision.

Zhang et al. (2018) sought to identify male children with ASD from TD children though diffusion magnetic resonance imaging (dMRI). Using the data of 70 children diagnosed with ASD and 79 typically developed controls obtained from the Center for Autism Research through Children's Hospital of Philadelphia, this study analyzed their whole brain white matter connectivity with the help of a SVM and 10-fold cross validation. By extracting multiple diffusion features from each fiber cluster of each subject, they were able to classify subjects as ASD or TD. The model with the highest accuracy, 78.33%, occurred with 4697 valid fiber clusters. This model produced a sensitivity of 84.81% and specificity of 72.86%.

Table 2.1 : Summarization of Reviewed Literature

Authors	Sample size	Data type	Prediction goal	Methods	Performance Evaluation Parameters	
(Bishop-Fitzpatrick 2018)	n = 91 ASD; n = 6186 control	ICD; V-codes; E-codes	ASD/TD	RF	Accuracy %	93
					Sensitivity %	75
					Specificity %	94
					AUC	0.88
(Narayanan 2016)	n = 1264 ASD; n = 462 other DD	ADI-R; SRS	ASD/other DD	SVM	Accuracy %	–
					Sensitivity %	[87, 89]
					Specificity %	[53, 59]
					AUC	–
(G. Bussu 2018)	n = 104 HR; n = 71 LR	MSEL; VABS; AOSI	High risk/low risk	SVM	Accuracy %	–
					Sensitivity %	–
					Specificity %	–
					AUC	0.65–0.71
(Chen CP1 2015)	n = 126 ASD; n = 126 TD	rs-fMRI	ASD/TD	RF	Accuracy %	91
					Sensitivity %	89
					Specificity %	93
					AUC	–
(Alessandro Crippa 2015)	n = 15 ASD; n = 15 TD	Kinematic data	ASD/TD	SVM	Accuracy %	85
					Sensitivity %	82
					Specificity %	89
					AUC	–
(Anibal Sólon Heinsfeld 2018)	n = 505 ASD; n = 530 TD	rs-fMRI	ASD/TD	Deep learning	Accuracy %	70
					Sensitivity %	74
					Specificity %	63
					AUC	–
(Kayleigh Hyde 2018)	n = 263 employers	Employer survey	ASD employment	Decision Tree	Accuracy %	75
					Sensitivity %	–
					Specificity %	82
					AUC	–
(Li 2017)	n = 16 ASD; n = 14 TD	Kinematic data	ASD/TD	SVM; RF; Naïve Bayes; Decision tree	Accuracy %	86.7
					Sensitivity %	85.7
					Specificity %	87.5
					AUC	–
					AUC	0.9
(Zhang F1 2018)	n = 70 ASD; n = 79 TD	dMRI	ASD/TD	SVM	Accuracy %	78
					Sensitivity %	85
					Specificity %	73
					AUC	–
(Yongxia Zhou 2014)	n = 127 ASD; n = 153 TD	MRI	ASD/TD	Random Tree	Accuracy %	70
					Sensitivity %	–
					Specificity %	–
					AUC	–

The literature review gave me insights about the present state of ASD's detection mechanisms and I could accumulate points necessary for designing the model. I could know that Neural Networks (both conventional and convolutional) are able to extract out important features from training data, the parameters learned upon which can generate accurate predictions on test data.

Chapter 3:

THEORETICAL FOUNDATION

Neural networks are presently in great popularity in the machine learning arena for their excellent performance. This chapter describes the essential concepts we studied for efficient designing of neural networks.

3.1 Classification Problem

In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category-membership is known. Classes are sometimes called targets, labels or categories. Classification predictive modeling is the task of approximating a mapping function f from input variables *x* to discrete output variables *y*.

Often, the individual observations are analyzed into a set of quantifiable properties, known as explanatory variables or features. These properties may variously be categorical (e.g., ‘A’, ‘B’, ‘AB’ or ‘O’, for blood type), ordinal (e.g., ‘large’, ‘medium’ or ‘small’, for sizes), integer-valued (e.g., the number of occurrences of a particular word in an email) or real-valued (e.g., measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

In the terminology of machine learning, classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering that involves grouping data into categories based on some measure of inherent similarity or distance. An algorithm that implements

classification concretely is known as a classifier. The term 'classifier' sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Types of classification:

- **Logistic regression** (binary)—analyzes a set of data points and finds the best fitting model to describe them. Easy to implement and very effective for input variables that are well known, and closely correlated with the outcome.

- **Decision tree** (multiclass)—classifies using a tree structure with if-then rules, running the input through a series of decisions until it reaches a termination condition. Able to model complex decision processes and is highly intuitive, but can easily overfit the data.

- **Random forest** (multiclass)—an ensemble of decision trees, with automatic selection of the best performing tree. Provides the strength of the decision tree algorithm without the problem of overfitting.

- **Naive Bayes classifier** (multiclass)—a probability-based classifier. Calculates the likelihood that each data point exists in each of the target categories. Simple to implement and accurate for a large set of problems, but sensitive to the set of categories selected.

- **K-Nearest neighbor** (multiclass)—classifies each data point by analyzing its nearest neighbors among the training examples. Simple to implement and understand effective for many problems, especially those with low dimensionality. Provides lower accuracy compared to supervised algorithms, and is computationally intensive.

3.2 Supervised Learning

Supervised machine learning incorporates an external teacher (output labels) so that each output unit is told what its desired response to input signals ought to be. During the learning process, global information may be required. Paradigms of supervised

learning include error-correction learning, reinforcement learning and stochastic learning.

An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. A well-known method is the least mean square (LMS) convergence for regression problems. However, classification problems optimize cross-entropy loss.

3.3 Neural Networks and Deep Learning

3.3.1 Artificial Neural Networks (ANNs)

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An ANN is configured for pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

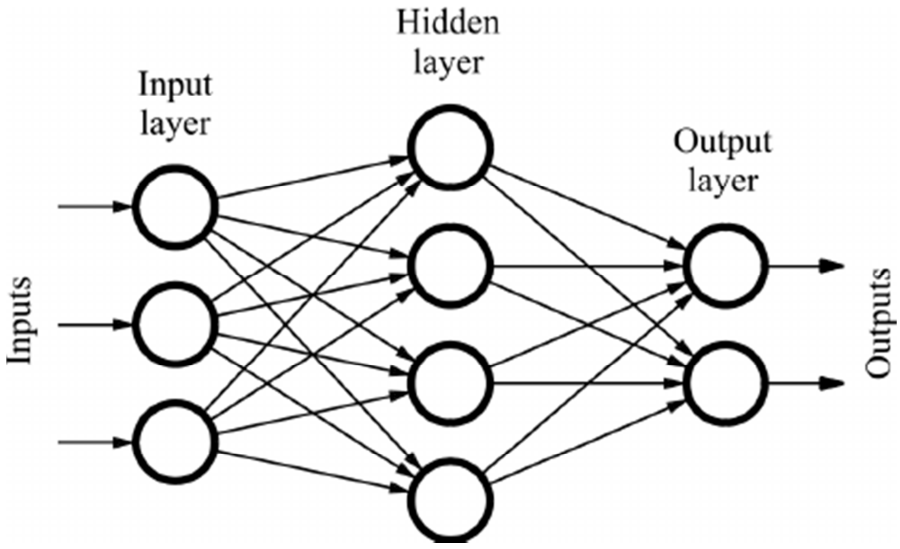

Figure 3.1: a feed-forward artificial neural network

The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. The signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Feed-forward ANNs (Figure 3.1) allow signals to travel one way only; from input to output. There is no feedback loop i.e., the output of one layer does not affect that same layer. .Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Signals travel from the first layer of neurons (input layer), to the last (output layer), possibly after traversing the layers multiple times.

3.3.2 Feed-forward Neural Networks

Feed-forward ANNs (Figure 3.1) allow signals to travel one way only; from input to output. There is no feedback loop i.e., the output of one layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

3.3.3 Elements of a Neural Network

Neural networks are composed of several layers of nodes. A node is a place where computation happens when it encounters sufficient stimuli.

Nodes: A node (Figure 3.2) combines input from the data with a set of coefficients, or weights that either amplify or dampen that input, thereby assigning significance to inputs for the task the algorithm is trying to learn. These input-weight products are summed and the sum is passed through a

node's activation function, to determine whether and to what extent that signal progresses further through the network to affect the ultimate outcome.

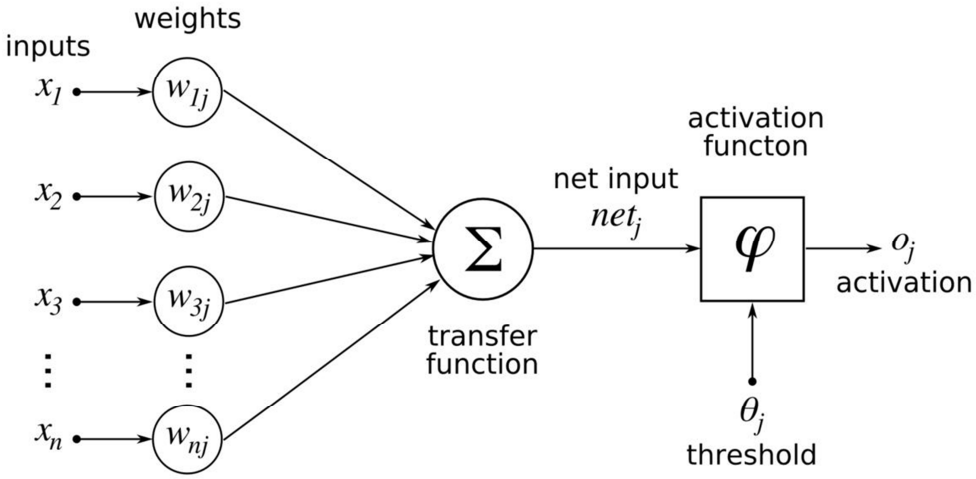

Figure 3.2: structure of a single node of an NN (a neuron)

Layers: A layer (Figure 3.3) is a row of neuron switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving data.

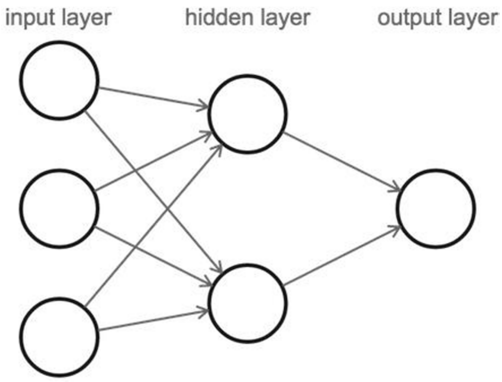

Figure 3.3 : simplified structure of a neural network

Weights: Pairing adjustable weights with input features is how significance is assigned to features with regard to how the network classifies and clusters input.

3.3.4 Layers of a Neural Network

The commonest type of artificial neural network consists of three layers of units: a layer of 'input' units is connected to a layer of 'hidden' units, which is connected to a layer of 'output' units (Figure 3.1).

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

3.3.5 Deep Learning

Deep learning (deep structured learning or hierarchical learning) (Figure 3.4) is a part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Deep learning architectures such as deep neural networks (neural networks with three or more hidden layers), deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

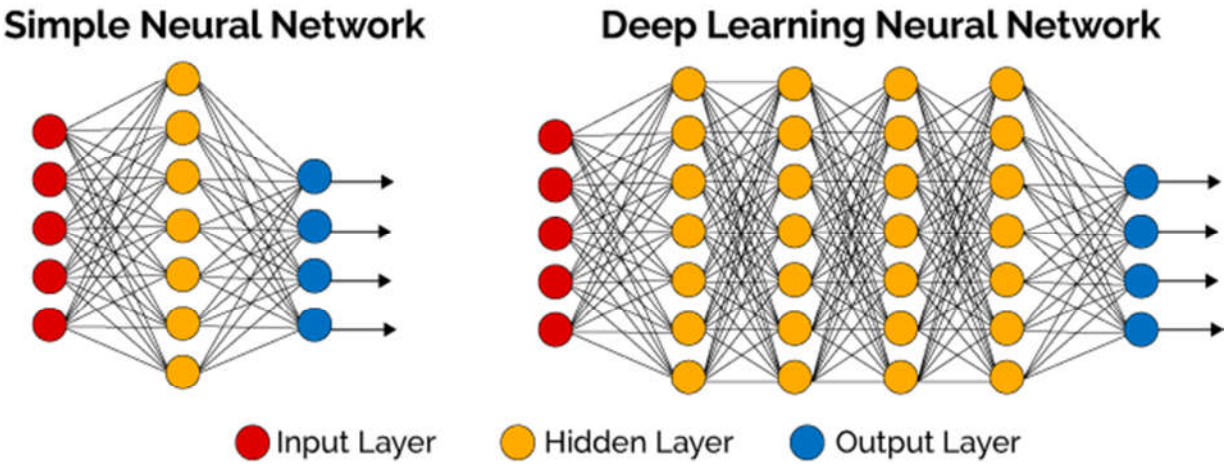

Figure 3.4 : Shallow vs. Deep Neural Network

Deep learning models, vaguely inspired by information processing and communication pat-terns in biological nervous systems, yet have various differences from the structural and functional properties of biological brains, which make them incompatible with neuroscience evidence.

3.4 Transfer Functions

An activation function (Figure 3.5) is a node that is added to the output end of any neural network. It is also known as a transfer function. It can also be attached between two neural networks. It is used to determine the output of neural network such as ‘yes’ or ‘no’. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending on the function). The nonlinear activation functions are the most used activation functions. This function typically falls into three categories:

- For linear units, the output activity is proportional to the total weighted output.
- For threshold units the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

- For Sigmoid units, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

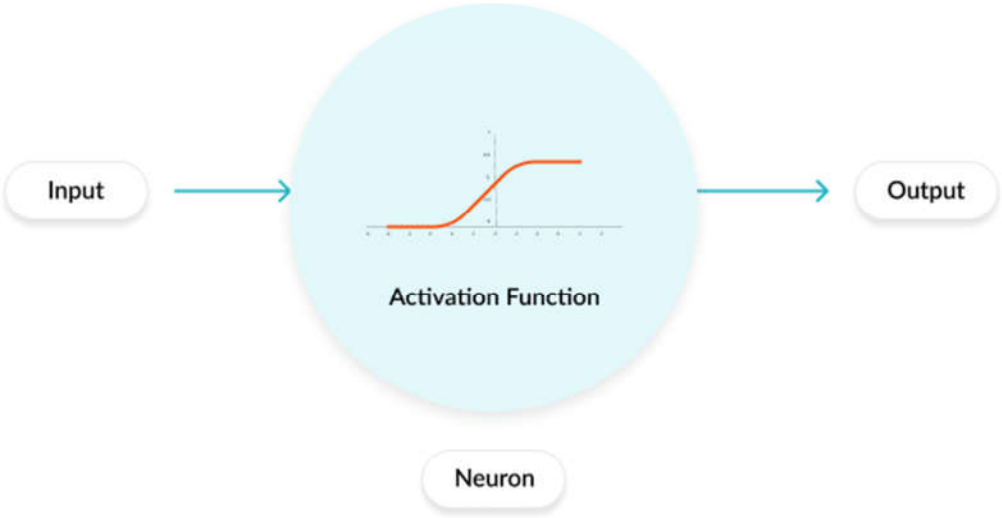

Figure 3.5 : Transfer Function

3.4.1 ReLU (Rectified Linear Unit) Activation Function

The ReLU is the most used activation function in the world right now. All the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly.

- As it can be seen (Figure 3.6), the ReLU is half rectified (from bottom). *f (z)* is zero when *z* is less than zero and *f(z)* is equal to *z* when *z* is above or equal to zero.
- Range: [0 to infinity)
- The function and its derivative, both are monotonic.

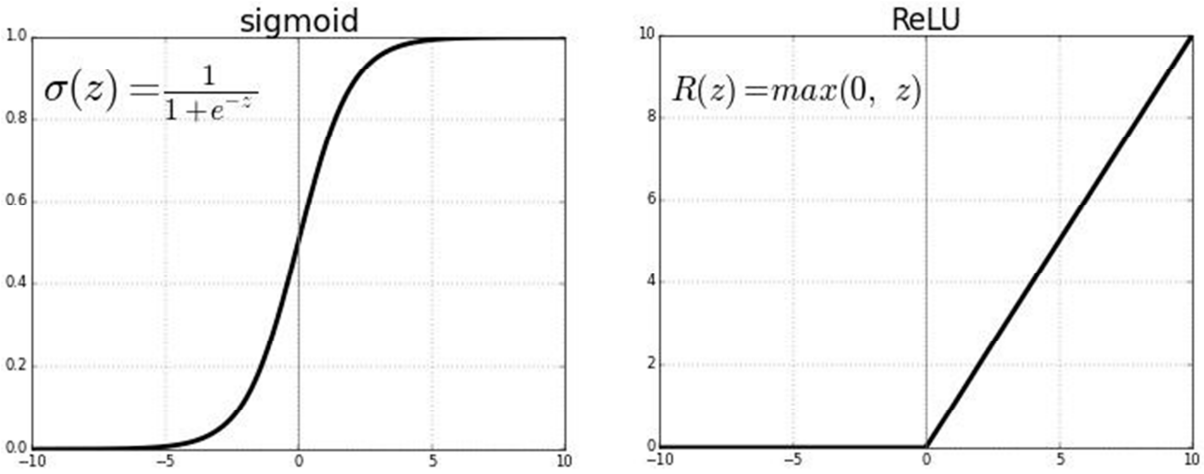

Figure 3.6 : ReLU vs. logistic Sigmoid

3.4.2 Common Activation Functions

- **The sigmoid function** has a smooth gradient and outputs values between zero and one. For very high or low values of the input parameters, the network can be very slow to reach a prediction, called the *vanishing gradient* problem.
- **The TanH function** is zero-centered making it easier to model inputs that are strongly negative strongly positive or neutral.
- **The ReLu function** is highly computationally efficient but is not able to process inputs that approach zero or negative.
- **The Leaky ReLu** function has a small positive slope in its negative area, enabling it to process zero or negative values.
- **The Parametric ReLu** function allows the negative slope to be learned, performing backpropagation to learn the most effective slope for zero and negative input values.
- **Softmax** is a special activation function use for output neurons. It normalizes outputs for each class between 0 and 1, and returns the probability that the input belongs to a specific class.
- **Swish** is a new activation function discovered by Google researchers. It performs better than ReLu with a similar level of computational efficiency.

3.4.3 Sigmoid or Logistic Activation Function

The Sigmoid function curve looks like an s-shape (Figure 3.7). The SoftMax function is a more generalized logistic activation function which is used for multiclass classification.

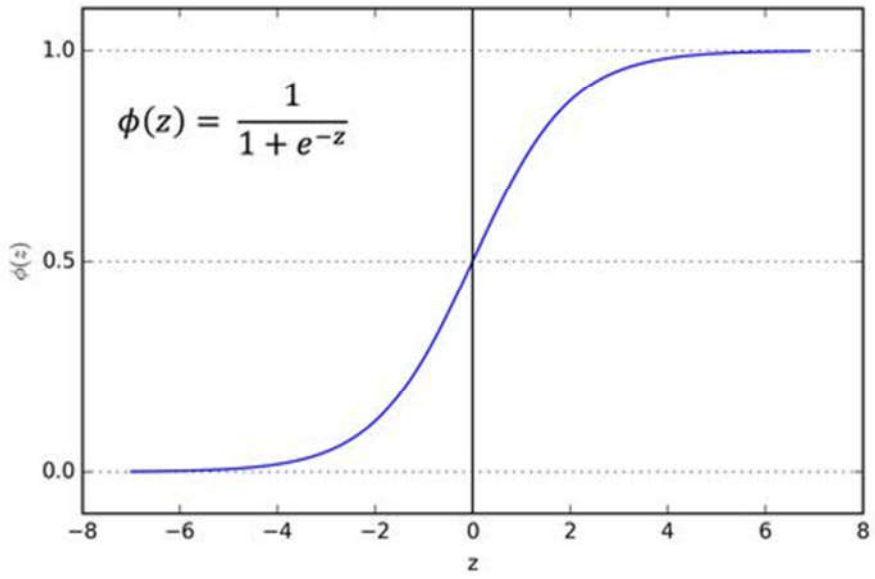

Figure 3.7: Sigmoid curve

- The main reason why sigmoid function is used is it exists in between (0 to 1).
- Therefore, it is especially used for models where one has to predict the probability as an output.
- The function is differentiable. That means the slope of the sigmoid curve can be found at any two points.
- The function is monotonic but the function's derivative is not.
- The logistic sigmoid function can cause a neural network to get stuck at the training time.

3.4.4 TanH Activation Function

The range of the tanh function is from (-1 to 1). tanh (Figure 3.8) is also Sigmoidal (s-shaped). Both tanh and logistic Sigmoid activation functions are used in feed-forward NNs.

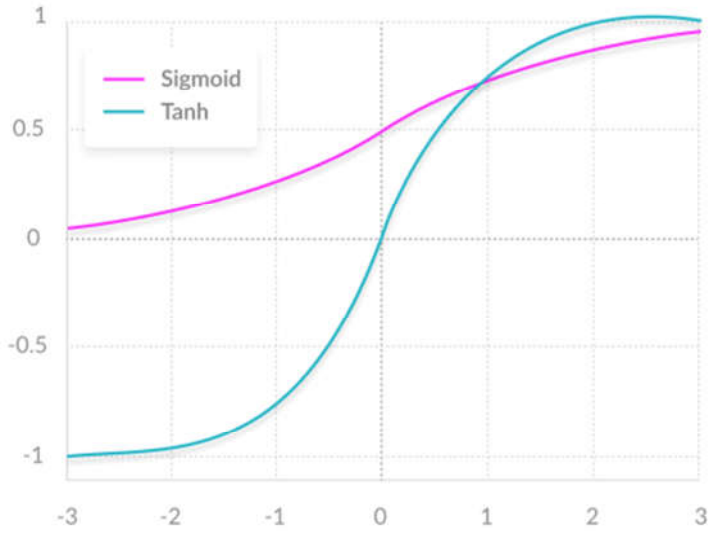

Figure 3.8 : tanh vs. logistic Sigmoid

- The advantage is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph.
- The function is differentiable.
- The function is monotonic while its derivative is not monotonic.
- The tanh function is mainly used for classification between two classes.

3.5 Division of Training Data

Training data: The model is initially fit on a training dataset that is a set of examples used to fit the parameters of the model. The classifier is trained on the training dataset using supervised learning. The training dataset often consists of pairs of an input vector and the corresponding output, which is commonly denoted as the target label. The current model is run with the training dataset and produces a result, which

is then compared with the target for each input vector. Based on the comparison and the specific learning algorithm being used, the parameters of the model are adjusted.

Cross-validation data: Successively, the fitted model is used to predict the responses for the observations in a second dataset. This validation dataset provides an unbiased evaluation of a model while tuning the model's hyper-parameters (e.g. the number of hidden units in a neural network). Validation datasets can be used for regularization by early stopping: stop training when the error on the validation dataset increases, as this is a sign of over-fitting the training data.

Test data: Finally, the test data is used to provide an unbiased evaluation of the final model fit on the training dataset. When the data in the test dataset has never been used in training (for example in cross-validation), the test dataset is also called a holdout dataset.

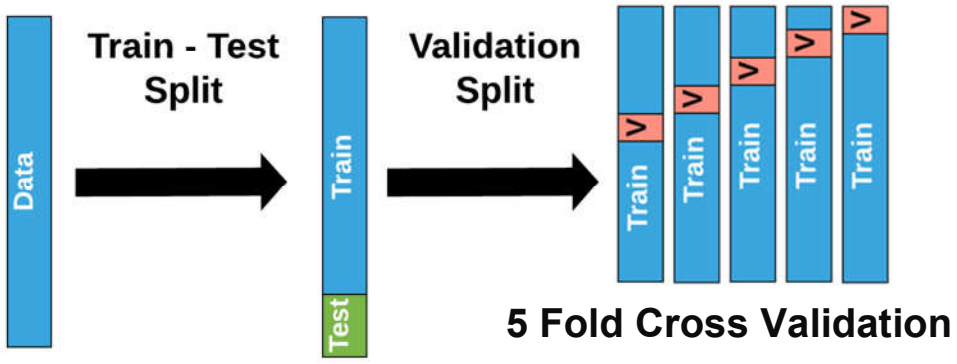

Figure 3.9 : Data Splitting into Train, Test and Validation set.

Division of Dataset into Training, Validation and Test Sets: The following data partitioning method has been used for this research (Figure 3.9):

- 65% of the entire Dataset for training (Training data)
- 15% of the entire Dataset for validation (Validation data)
- 20% of the entire Dataset for testing (Testing data)

In most articles, it is 70% vs. 30% for training and testing set respectively. Better terminology is to call it a 'dev' set as it is used in the development.

3.6 Weight Initialization of NNs: Random Normal

Careful initialization of weights helps signals reach deep into the network. If the weights in a network start too small, then the signal shrinks as it passes through each layer until it is too tiny to be useful.

If the weights in a network start too large, then the signal grows as it passes through each layer until it is too massive to be useful. For example, if we use the values of 0.0 for all weights the equations of the learning algorithm would fail to make any changes to the network weights, and the model will be stuck. It is important to note that the bias weight in each neuron is set to zero by default, not a small random value.

Specifically, nodes that are side-by-side in a hidden layer connected to the same inputs must have different weights for the learning algorithm to update the weights.

kernel_initializer is the function that initializes the weights. It is used for when fitting the deep learning model the weights will be initialized to numbers close to zero, but not zero. To achieve this RandomNormal initializer has been used. It uses randomness in order to find a good enough set of weights for the specific mapping function from inputs to outputs in the data that is being learned. It means that a specific network on a specific training data will fit a different network with a different model skill each time the training algorithm is run.

3.7 Optimizer: Adam Gradient Descent

In order to train an NN, weights of each unit must be adjusted in such a way that the error be-tween the desired output and the actual output is reduced. This requires the neural network to compute the error derivative of the weights *(EW)*. In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back-propagation algorithm is the most widely used method for determining *EW*.

Optimization algorithms help to minimize an error function *E(x)* which is a mathematical function dependent on a model's internal learnable parameters used in computing the target values *(y)* from the set of predictors *(x)*. The weights *(W)* and biases *(b)* of a neural network are its internal learnable parameters used in computing the output values. These are learned and updated in the direction of the optimal solution through minimizing the loss during the network's training process. Optimization plays a major role in the training process of an NN.

To apply gradient descent to the neural network an optimization strategy is used that works to reduce errors during the training process. Gradient descent is how randomly assigned weights in a neural network are adjusted by reducing the cost function, which is a measure of how well a neural network performs based on the output expected from it.

The aim of a gradient descent is to get the point where the error is at its least. This is done by finding where the cost function is at its minimum, which is referred to as a local minimum. In gradient descent, we differentiate to find the slope at a specific point and find out if the slope is negative or positive.

The method is straightforward, computationally efficient, has little memory requirements, invariant to a diagonal rescaling of the gradients and well-suited for problems that are large in terms of data and/or parameters.

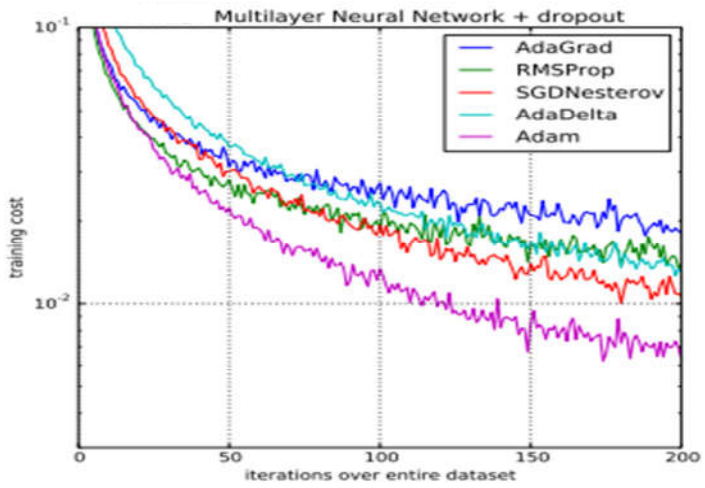

Figure 3.10 : Comparison of Adam to Other Optimization Algorithms.

Adam configuration parameters are:

- **Alpha-** Also referred to as the learning rate or step size. The proportion that weights are updated (e.g. 0.001). Larger values (e.g. 0.3) results in faster initial learning before the rate is updated. Smaller values (e.g. 1.0E-5) slow learning right down during training
- **Beta1-** The exponential decay rate for the first moment estimates (e.g. 0.9).
- **beta2-** The exponential decay rate for the second-moment estimates (e.g. 0.999). This value should be set close to 1.0 on problems with a sparse gradient (e.g. NLP and computer vision problems).
- **Epsilon-** Is a very small number to prevent any division by zero in the implementation (e.g. 10E-8).

Further, learning rate decay can also be used with Adam.

Chapter 4

PROPOSED METHOD

In this chapter, the workflow and the model experimented with for successful detection of early Autism Spectrum Disorder (ASD) has been presented. The methodology is also mentioned in this chapter.

4.1 Methodology

The post-processing has been performed, to select the best version, with the following aims:

- NNs are versed in finding complex patterns within datasets. A set of weights determine the mapping from one layer to the other and each neuron accounts for a more complex feature than the ones in its previous layer. Thus gradual layers detect more sophisticated features and the final sigmoid classification provides the output label with the highest probability. This capability of the algorithm is in harmony with our purpose, thus justified is the use of NNs.

- Moreover, plain NNs are suitable for a structured dataset where features are numerical values, rather than sequence data or images. Such architecture creates the scope for individual numerical features as input and delicately propagates them through the network while making refinements to the parameters.

- Furthermore, being a parametric approach, a NN knows of its functional form. This reduces the workload of assigning different kernels and selecting the right kernel for determining the decision boundary. This simplicity also makes NNs a suitable choice for early ASD's classification.

4.2 Proposed Strategy

For the model to give good predictions, it was customary (Figure 4.1) to select the right, predictive, impactful features. So, EDA has been implemented to find the correlation of the features with the categories (e.g., ASD Traits Present, and ASD Traits Absent). This led to visualizations of different combinations (e.g., Univariate, bivariate and multivariate) of features and helped to extract the most significant ones. The irrelevant features have also been eliminated to make the best use of available data.

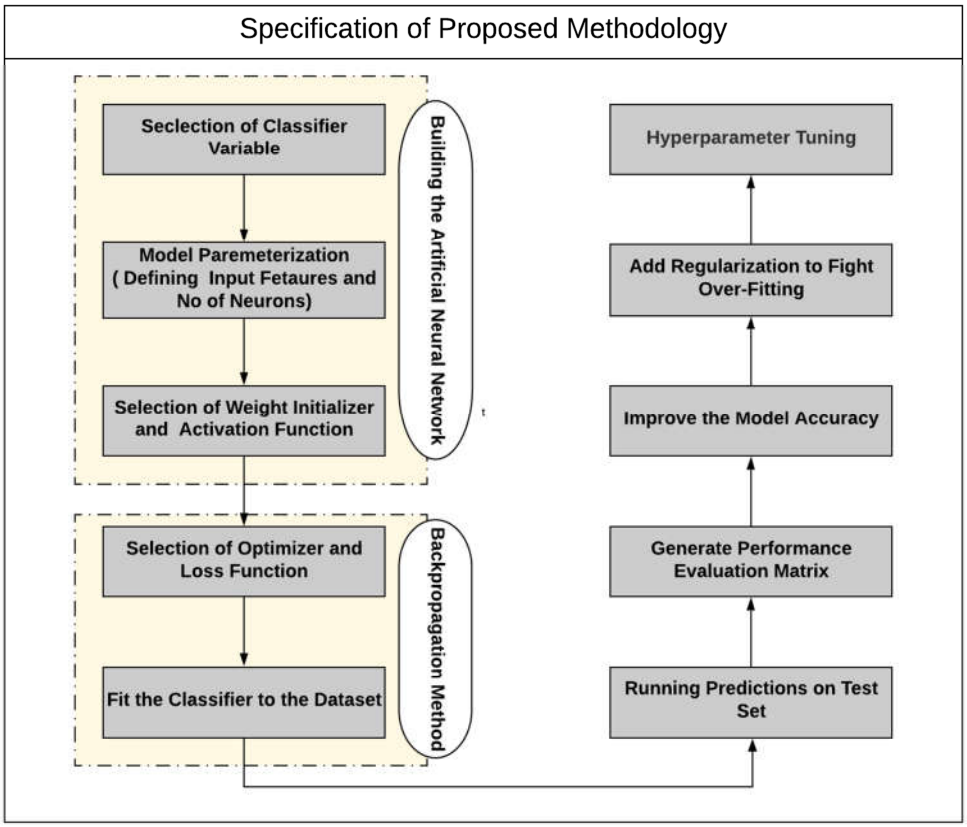

Figure 4.1 : workflow of the proposed methodology

4.3 Motivation Behind the Four-layer Model

Despite of the limitation of the size of the Dataset I wanted to develop a Deep Neural Network which will give decent accuracies for the task.

For achieving the aims mentioned earlier, this model served the targeted purpose.

4.4 Four-Layer Architecture

This is the visualization of the model that has been implemented to the Dataset. The model has a total of four layers and hyperparameters (learning rate, number of epochs, Adam Optimization parameters, and size of batches) which were tuned for this model throughout the study (Figure 4.2).

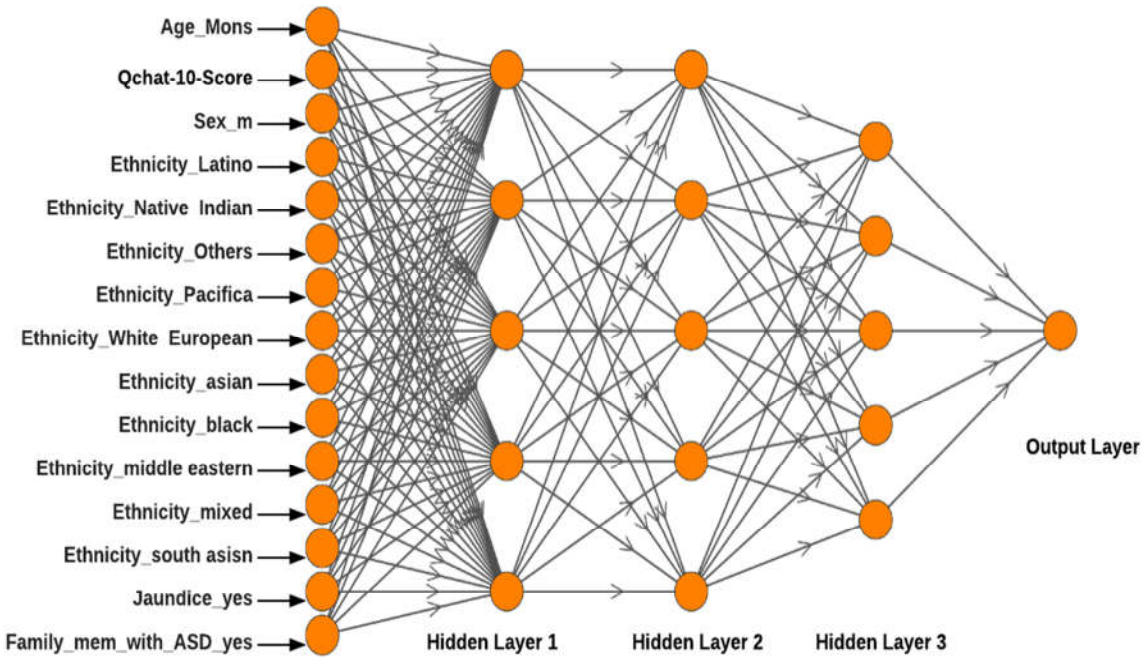

Figure 4.2 : The Four-layer architecture

Medical data being scarce and classified, the dataset used for this research has fewer (only 1054) tuples than what is called ‘big data’ in modern data science. Ironically, neural networks, being intrinsically a complex architecture compared to logistic regression and other simpler learning, have a hunger for huge amounts of

data to generate excellent predictions. So, the challenge was to select a model capable enough to fit the comparatively miniature dataset that would supersede in Performance conventional learning approaches.

This is why three hidden layers have been chosen, so as to not make the model overfit. Conversely, I keep the number of hidden neurons in each layer identical to each other, in order not to underfit the data.

4.5 Performing Random Normal Initialization to Chosen Model

This serves the process of symmetry-breaking and gives much better accuracy. In this method, the weights are initialized very close to zero, but randomly. This helps in breaking symmetry and every neuron is no longer performing the same computation.

4.6 Defining the Cross-entropy Loss Function

The cross-entropy loss function has been optimized for the two-class classification problem with a view to obtaining the greatest refinement of the parameters. Represented here is precisely the cross-entropy, summed over all training examples:

$$-\log L\left(\{y^n\},\{\hat{y}^{(n)}\}\right) = \sum_n [-\sum_i y_{i \log \hat{y}_i^{(n)}} = \sum_n H(y^{(n)}, y^{(n)})$$

Where n indicates the number of training examples, y^n denotes the ground-truth value for an individual example, $\hat{y}^n$ is the prediction of the model and i represents the sequence of activation within a layer.

4.7 Minimization of Loss using Gradient Descent

Gradient descent is an optimization strategy that works to reduce errors during the training process. Gradient descent is how randomly assigned weights in a neural network are adjusted by reducing the cost function, which is a measure of how well a neural network performs based on the output expected from it.

The aim of a gradient descent is to get the point where the error is at its least. This is done by finding where the cost function is at its minimum, which is referred to as a local minimum. In gradient descent, strategy is to find the slope at a specific point by differentiates and find out if the slope is negative or positive.

A set of parameters θ is to be chosen so as to minimize error J (θ). The gradient descent algorithm starts with some initial θ, and then repeatedly performs the update.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

This update is simultaneously performed for all features, i.e., j = 0, 1... n where α is the learning rate. This is a very natural algorithm that repeatedly takes a step in the direction of the steepest decrease of J (θ). To implement the algorithm, the partial derivative term has to be computed. If there is only one training example (x, y), we have,

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2$$

$$= 2.\frac{1}{2}(h_\theta(x) - y).\frac{\partial}{\partial \theta_j}(h_\theta(x) - y))$$

$$= (h_\theta(x) - y).\frac{\partial}{\partial \theta_j}(\sum_{i=0}^{n} \theta_j x_i - y)$$

$$= (h_\theta(x) - y).x_j$$

Therefore, $\frac{\partial}{\partial \theta_j} J(\theta) = (h_\theta(x) - y).x_j$

To modify this method for a training set of more than one example, it is to be replaced with the following algorithm:

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \qquad (for\ every\ j)$$

}

4.8 Application of Adam Optimization to Gradient Descent

Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The parameters for Adam Optimization are as follows.

• α: The learning rate or step size. Learning rate decay, permissible in Adam, has not been used for ASD's classification.

• β1: The exponential decay rate for the first moment estimates (e.g. 0.9).

• β2: The exponential decay rate for the second-moment estimates (e.g. 0.999). This value is set close to 1.0 on problems with a sparse gradient.

• ε: A very small number to prevent any division by zero in the implementation (e.g. 10E-8).

4.9 Running Predictions on the Test Set

According to the division of the dataset after training the model with the train data, checked the prediction probability of the model through test data. A single prediction was also performed to check if the model is executing well.

4.10 Generate Performance Evaluation Matrix

A confusion matrix generated to check the number of correct and incorrect predictions. It is also known as an error matrix, is a square matrix that reports the number of true positives(tp), false positives(fp), true negatives(tn), and false negatives(fn) of a classifier.

- A **true positive** is an outcome where the model correctly predicts the positive class (also known as sensitivity or recall).
- A **true negative** is an outcome where the model correctly predicts the negative class.
- A **false positive** is an outcome where the model incorrectly predicts the positive class.
- A **false negative** is an outcome where the model incorrectly predicts the negative class.

4.11 Improve the Model Accuracy

When the model is trained many times it will give different results. The accuracies of each training have a high variance. In order to solve this problem, K-fold cross-validation is used. The model is trained on the first 4 folds and tested on the last fold. This iteration continues until all folds have been used. Each of the iterations gives its own accuracy. The accuracy of the model becomes the average of all these accuracies.

4.12 Weight Regularization to Fight Over-Fitting

Predictive models are prone to a problem known as overfitting. This is a scenario whereby the model memorizes the results in the training set and isn't able to generalize on data that it hasn't seen. Typically overfitting is observed when a very high variance on accuracies.

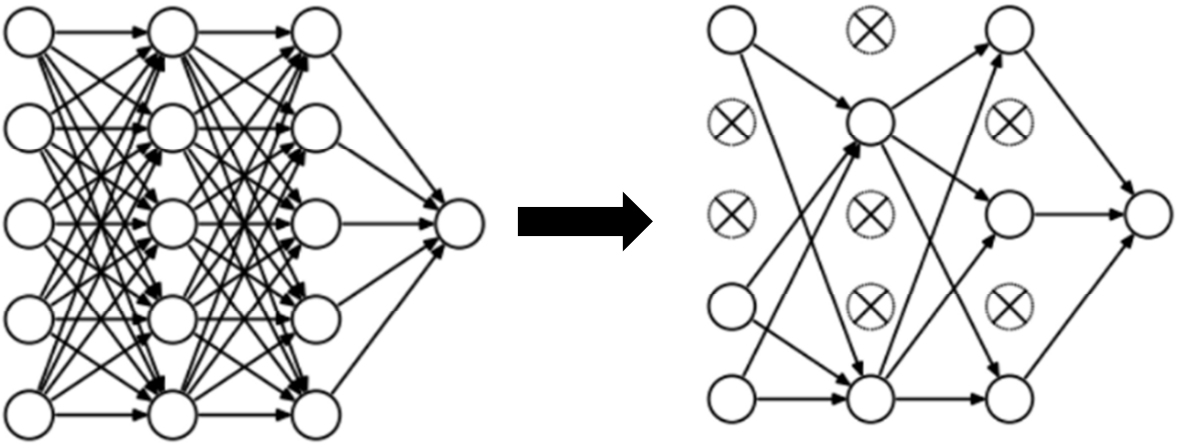

Figure 4.3 : Weight Regularization in NN

To help fight over-fitting in a layer is added to the model. In neural networks, dropout regularization (Figure 4.3) is the technique that fights overfitting by adding a Dropout layer in the neural network. It has a rate parameter that indicates the number of neurons which will be deactivated during each iteration. The process of deactivating neurons is usually random. In this case, specify 0.5 as the rate meaning that 5% of the neurons will deactivate during the training process.

4.13 Final Model: Tuning of Hyperparameters

Factors and specifications that determine the efficacy of a learning model other than the weights or parameters are called hyperparameters. It was extremely important to tune and find the appropriate hyperparameters for ensuring the most favorable outcome for the proposed models. Below are hyperparameters of the most promising model (Table 4.1).

Table 4. 1: Tuned hyperparameters for the proposed method

Hyperparameter	**Tuned value**
number of input features	15
number of hidden layers	3
number of hidden units	first hidden layer:5, second hidden layer: 5 third hidden layer: 5
activation function applied to linear function	ReLU
number of output classes	2; namely, ASD traits yes (1), ASD traits no (0)
division of dataset	training-set: 80%, test-set: 20%
loss function to optimize	Binary Crossentropy Loss function
learning rate, α	0.006
number of epochs	20
size of batch	10 examples per batch
Adam optimization parameters	learning rate decay: not used
NN evaluation metrics	accuracy, precision, recall, F1-score

Chapter 5

PREPARATION OF THE DATASET

5.1 Gathering Data

Dataset for this research paper has been collected from Kaggle Website. The dataset in the study was collected through a mobile application called ASDTests screening app, which implemented a screening method called Q-Chat-10 to collect primary data related to autism screening. The ASD Tests application contains ten questions related to behavioral traits besides other features such as age, gender, ethnicity, jaundice and family history of autism. The participants undergo autism screening using the ASDTests screening app and, at the end of the screening, a score is calculated for the individual based on the answers and using the Q-Chat-10 scoring functions.

Often parents, caregivers, teachers, or medical professionals take the test on behalf of the child.

The dataset features are depicted in Table 1 in which features A1 -A10 are answers to the Q-Chat-10 questions. A1-A10: Items within Q-Chat-10 in which questions possible answers: “Always, Usually, Sometimes, Rarely & Never” items’ values are mapped to “1” or “0” in the dataset. For questions 1-9 (A1-A9) in Q-chat-10, if the response was sometimes / rarely / never “1” are assigned to the question (A1-A9). However, for question 10 (A10), if the response was Always / usually / Sometimes then “1” is assigned to that question. If the user obtained More than 3, Add points together for all ten questions. If a child scores more than 3 (Q-chat-10- score) then there is a potential ASD traits otherwise no ASD traits are observed.

The remaining features in the datasets are collected from the “submit” screen in the ASDTests screening app. It should be noted that the class variable was assigned automatically based on the score obtained by the user while undergoing the screening process using the ASDTests app.

Table 5. 1: Details of variables mapping to the Q-Chat-10 screening methods

Variable in Dataset	**Corresponding Q-chat-10-Toddler Features**
A1	Does your child look at you when you call his/her name?
A2	How easy is it for you to get eye contact with your child?
A3	Does your child point to indicate that s/he wants something? (e.g. a toy that is out of reach)
A4	Does your child point to share interest with you? (e.g. poin9ng at an interes9ng sight)
A5	Does your child pretend? (e.g. care for dolls, talk on a toy phone)
A6	Does your child follow where you're looking?
A7	If you or someone else in the family is visibly upset, does your child show signs of warning to comfort them? (e.g. stroking hair, hugging them)
A8	Would you describe your child's first words as:
A9	Does your child use simple gestures? (e.g. wave goodbye)
A10	Does your child stare at nothing with no apparent purpose?

Table 5. 2: Features collected and their descriptions

Feature	**Type**	**Description**
A1: Question 1 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A2: Question 2 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A3: Question 3 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A4: Question 4 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A5: Question 5 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A6: A6: Question 6 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A7: Question 7 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A8: Question 8 Answer	Binary (0, 1)	The answer code of the question based on the screening method used

A9: Question 9 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
A:10 Question 10 Answer	Binary (0, 1)	The answer code of the question based on the screening method used
Age	Number	Toddlers (months)
Score by Q-chat-10	Number	1-10 (Less than or equal 3 no ASD traits; > 3 ASD traits
Sex	Character	Male or Female
Ethnicity	String	List of common ethnicities in text format
Born with jaundice	Boolean (yes or no)	Whether the case was born with jaundice
Family member with ASD history	Boolean (yes or no)	Whether any immediate family member has a PDD
Who is completing the test	String	Parent, self, caregiver, medical staff, clinician ,etc.
Class variable	String	ASD traits or No ASD traits (automatically assigned by the ASDTests app). (Yes / No)

5.2 Data Preprocessing

The dataset (Figure 5.1) has undergone several preprocessing (Figure 5.2) and exploratory analysis, later was divided into test and train dataset to feed the neural network.

Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Score	Sex
1	0	0	0	0	0	0	1	1	0	1	28	3	f
2	1	1	0	0	0	1	1	0	0	0	36	4	m
3	1	0	0	0	0	0	1	1	0	1	36	4	m
4	1	1	1	1	1	1	1	1	1	1	24	10	m
5	1	1	0	1	1	1	1	1	1	1	20	9	f
6	1	1	0	0	1	1	1	1	1	1	21	8	m
7	1	0	0	1	1	1	0	0	1	0	33	5	m
8	0	1	0	0	1	0	1	1	1	1	33	6	m
9	0	0	0	0	0	0	1	0	0	1	36	2	m
10	1	1	1	0	1	1	0	1	1	1	22	8	m
11	1	0	0	1	0	1	1	0	1	1	36	6	m

Figure 5.1 : Snapshot of the labeled dataset

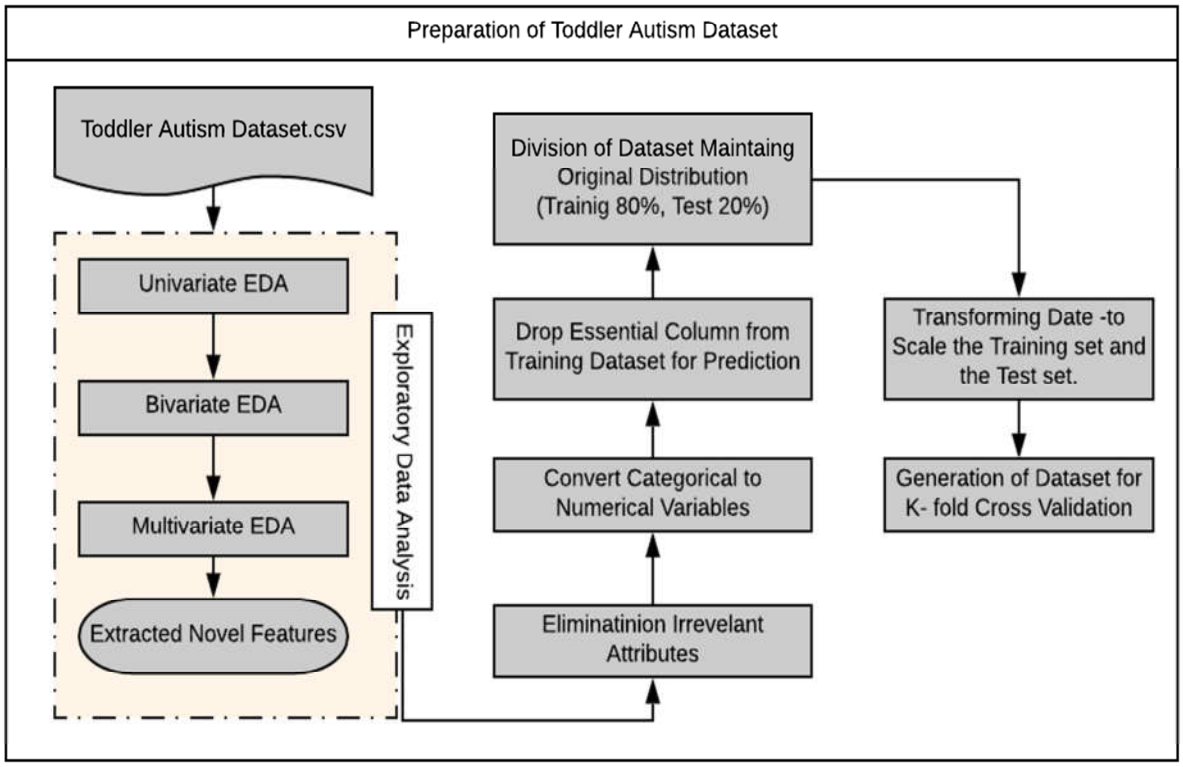

Figure 5.2 : Preprocessing of the Autism dataset

5.3 Selection of Features using Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

5.4 Variable Identification

The very first step in exploratory data analysis is to identify the type of variables in the dataset. Variables are two types: numerical and categorical.

5.5 Graphical Analysis

Variable under Analysis: ASD Traits (ASD Traits Absent and ASD Traits Present)

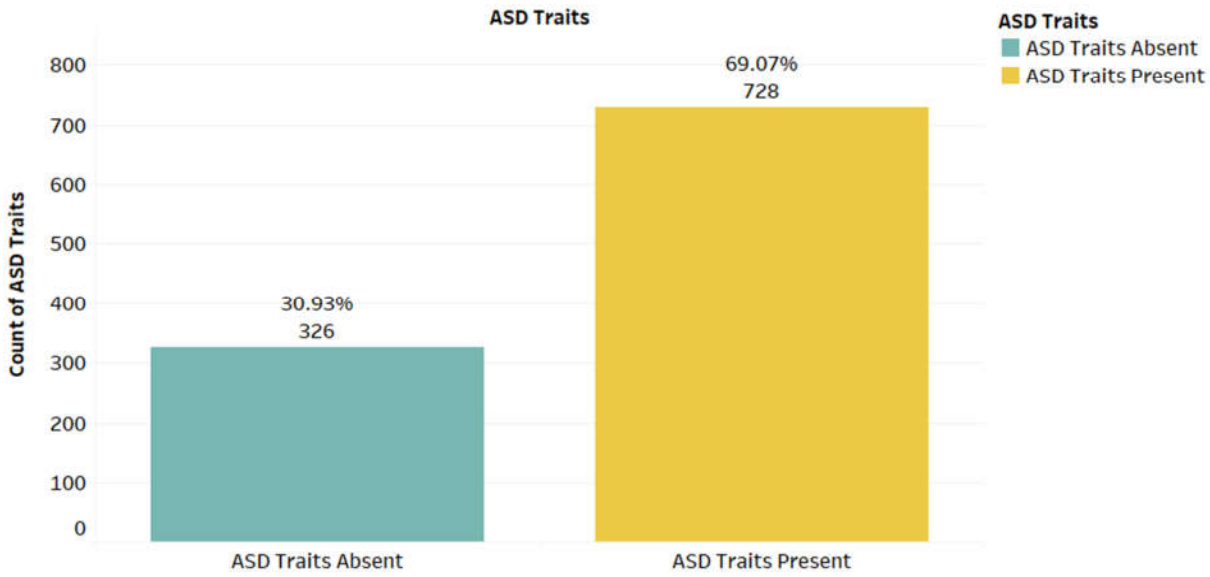

Figure 5.3 : Distribution of ASD Traits in the dataset

This distribution (Figure 5.3) gives the insights that the percentage of subjects in which ASD traits present is higher (69.07%) compare to the subjects which does not contains any ASD traits.

5.6 Univariate Exploratory Data Analysis (EDA)

Variable under Analysis: Ethnicity

Distribution of ethnicity in our dataset is more for White European (31.69%), Asian (28.37%) and Middle Eastern (17.84%) respectively (Figure: 5.4).

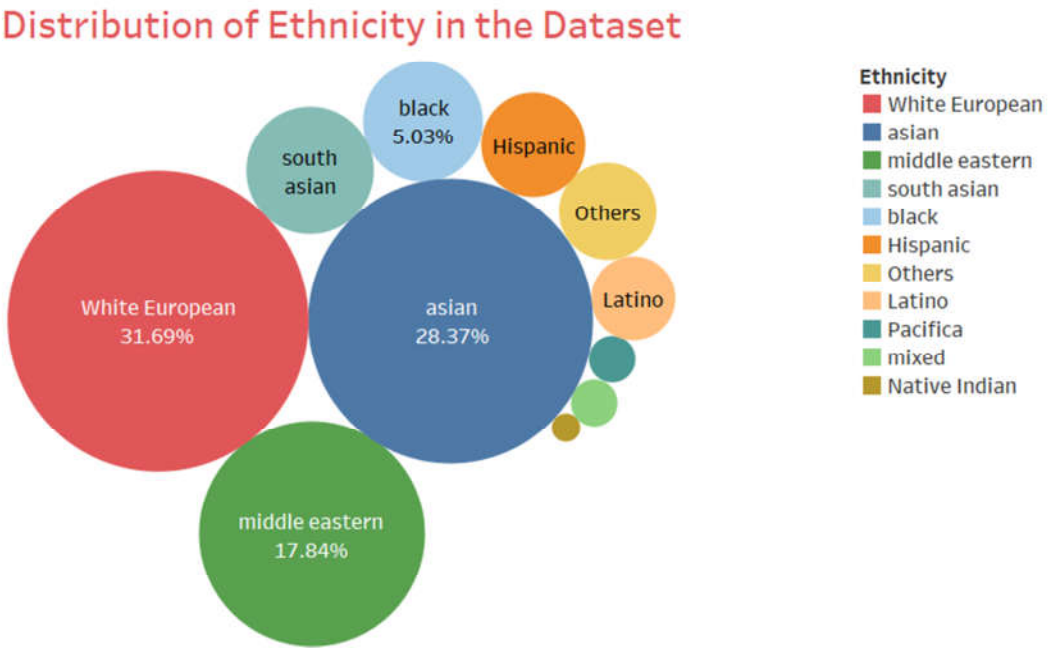

Figure 5.4 : Univariate analysis of Ethnicity

5.7 Bivariate Exploratory Data Analysis (EDA)

Variable under Analysis: Ethnicity and ASD Traits

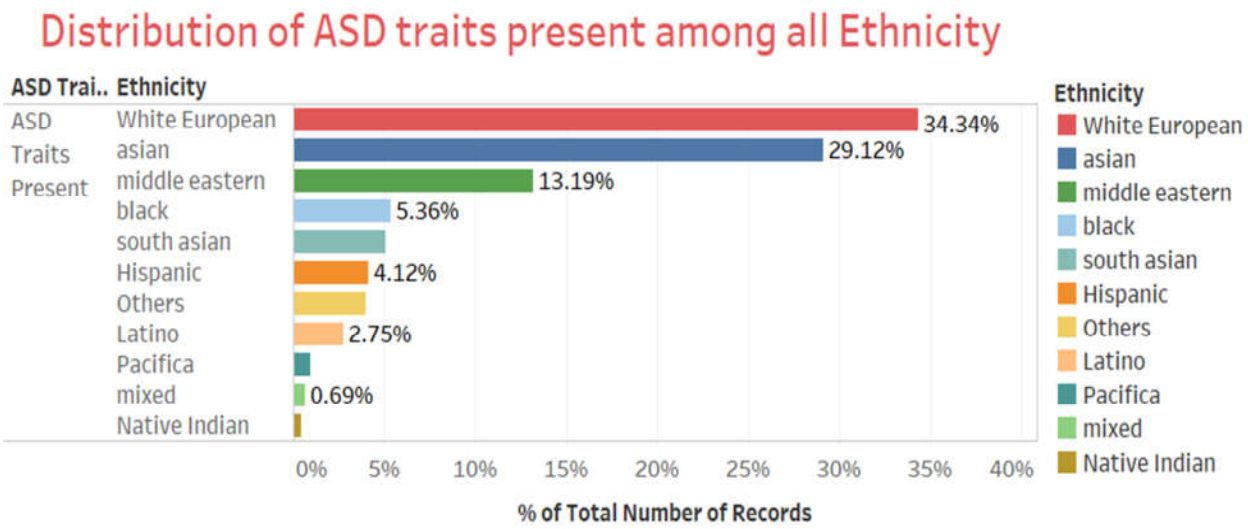

Figure 5.5 : Bivariate analysis of Ethnicity and ASD Traits

(Figure: 5.5) shows White European (34.34%), Asian (29.217%) and Middle Eastern (13.19%) respectively has the highest probability to be ASD positive compare to the Non-ASD.

Variable under Analysis: Family ASD record and ASD Traits

While 67.65% of toddlers has shown (Figure 5.6) ASD symptoms due to their family members' past ASD records, 69.34% toddlers has shown ASD symptoms despite their family members' past ASD histories. Therefore, past family ASD records have less significance while detecting ASD traits in toddlers.

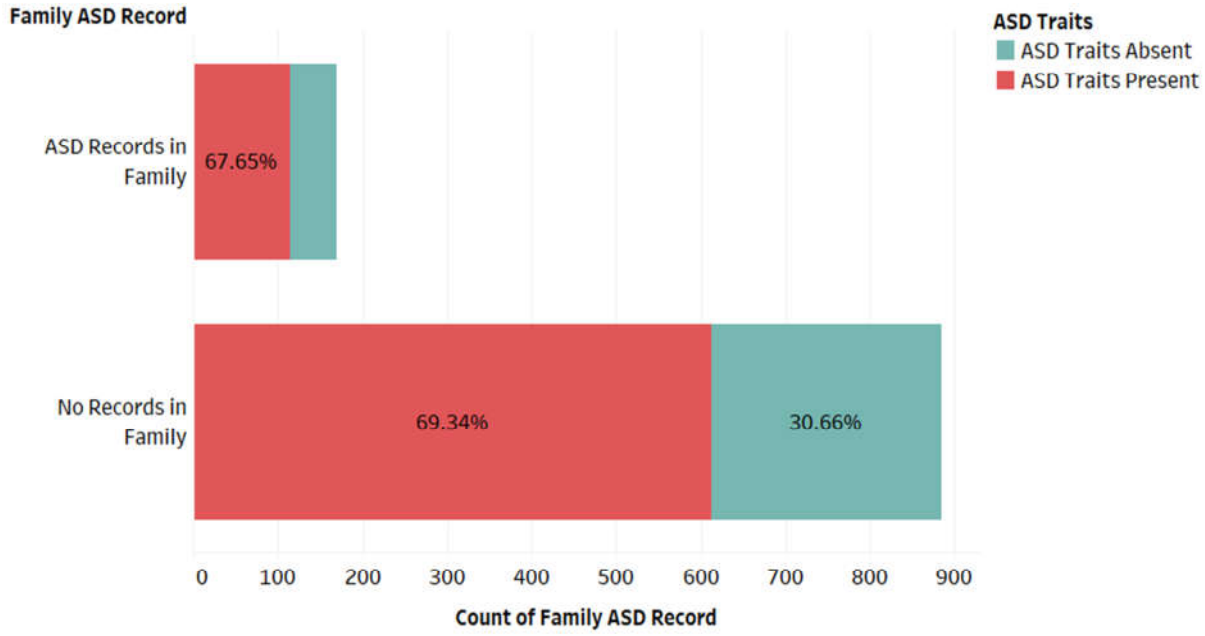

Figure 5.6 : Bivariate analysis of Family ASD records and ASD trait

Variable under Analysis: Jaundice record and ASD Traits

While 74.65% of toddlers has shown (Figure 5.7) ASD symptoms due to their past jaundice records, 66.97% toddlers has shown ASD symptoms despite any past jaundice histories.

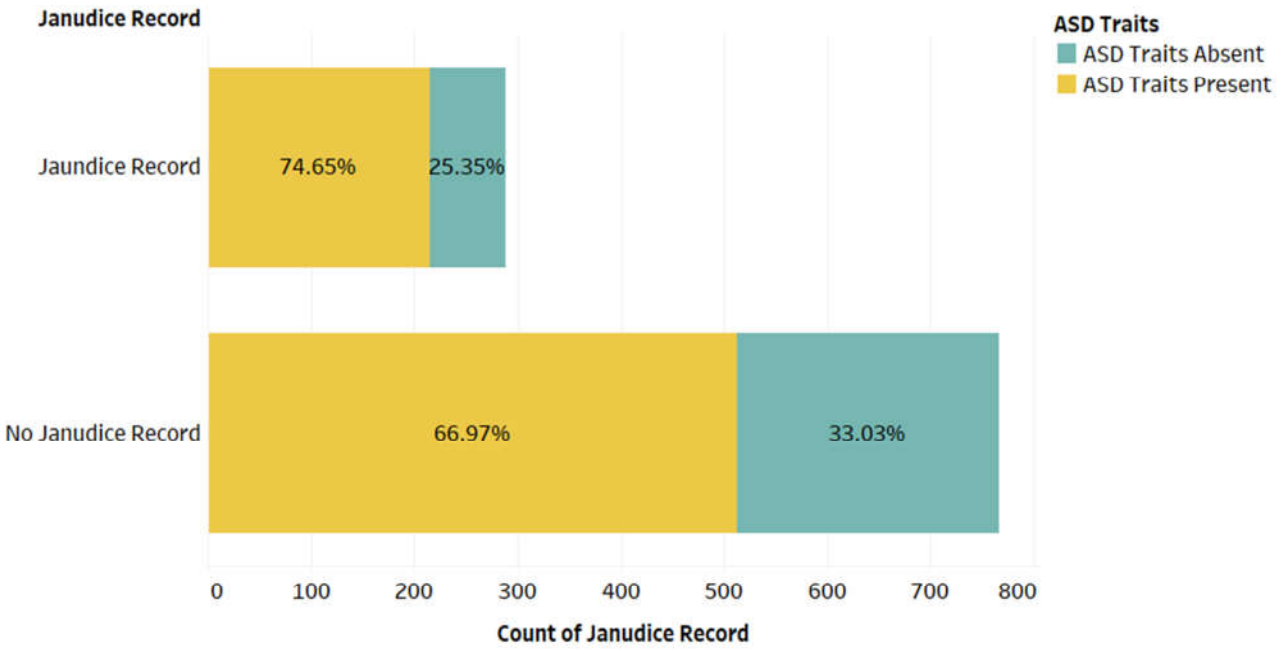

Figure 5.7 : Bivariate analysis of Jaundice records and ASD trait

Variable under Analysis: Q Chat 10 and ASD Traits

From the (Figure: 5.8) it is evident that only Q chat 10 score bigger than 3 are directly related to ASD traits in toddlers.

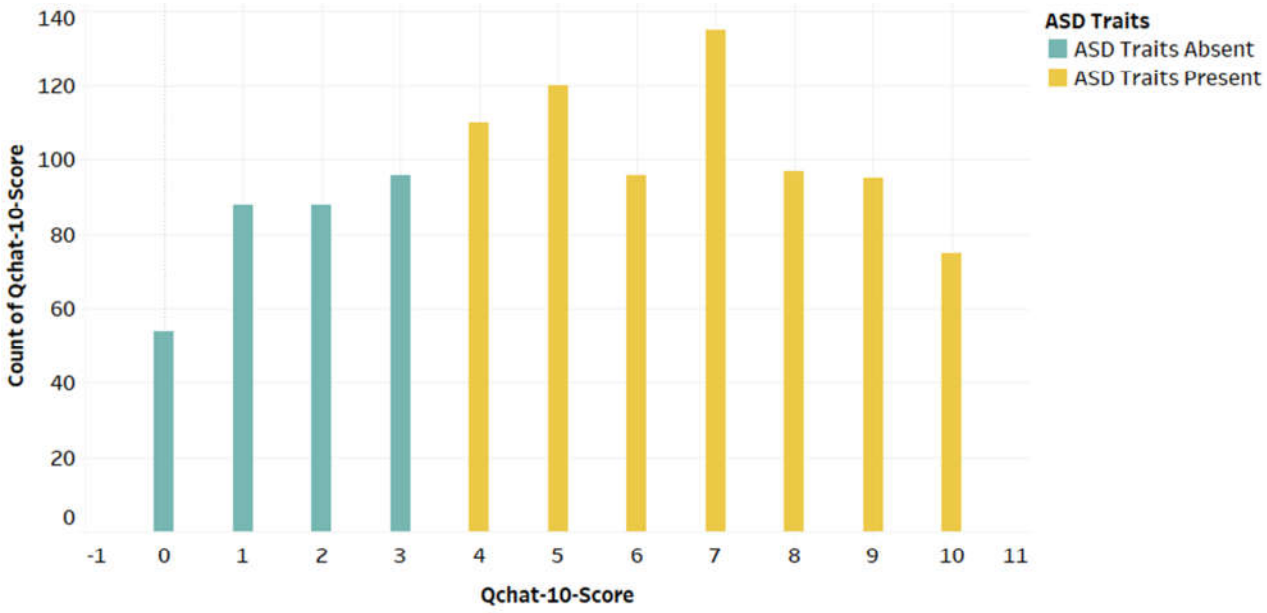

Figure 5.8 : Bivariate analysis of Jaundice records and ASD trait

Moreover, scores 7, 4 and 5 are the more prominent while scores 10 is least possible outcome among all. So, it is evident that medium scores are more responsible to detect ASD symptoms among toddlers.

5.8 Multivariate Exploratory Data Analysis (EDA)

Variable under Analysis: Ethnicity, Gender and ASD Traits

(Figure 5.9) shows for all ethnicities, ASD traits are prominent in males compare to the number of females.

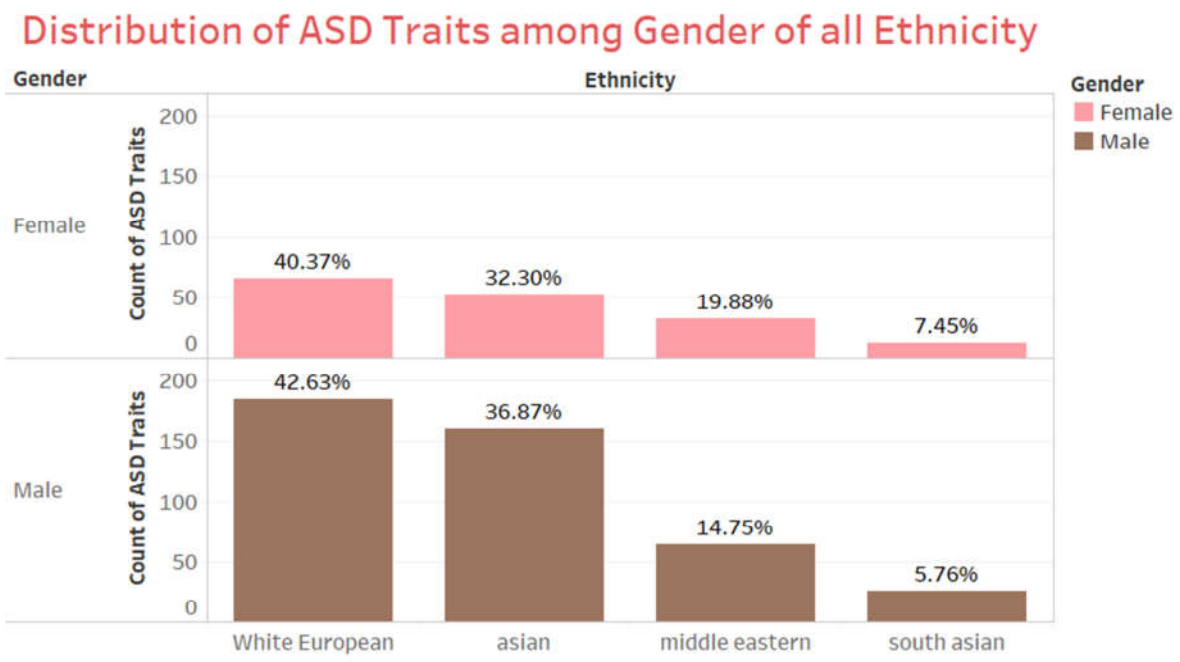

Figure 5.9 : Multivariate analysis of Ethnicity with Gender and ASD traits

Variable under Analysis: Ethnicity, Age Group and ASD Traits

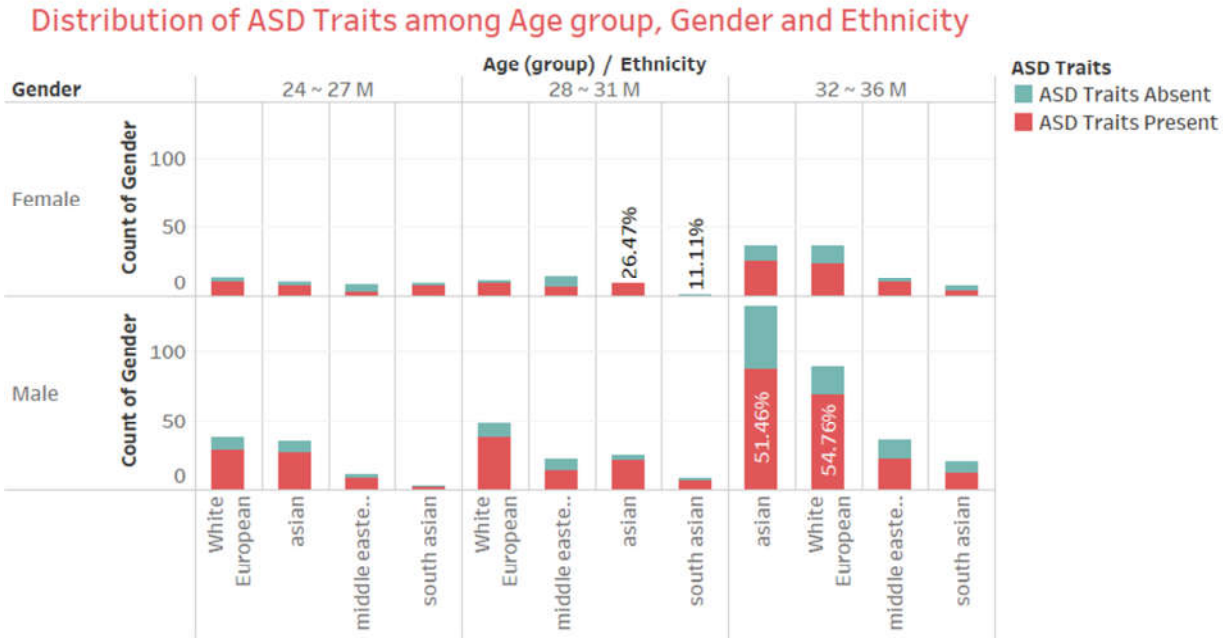

Figure 5.10 : Multivariate analysis of Ethnicity with Age group and ASD traits

Variable under Analysis: Jaundice record, Gender and ASD Traits

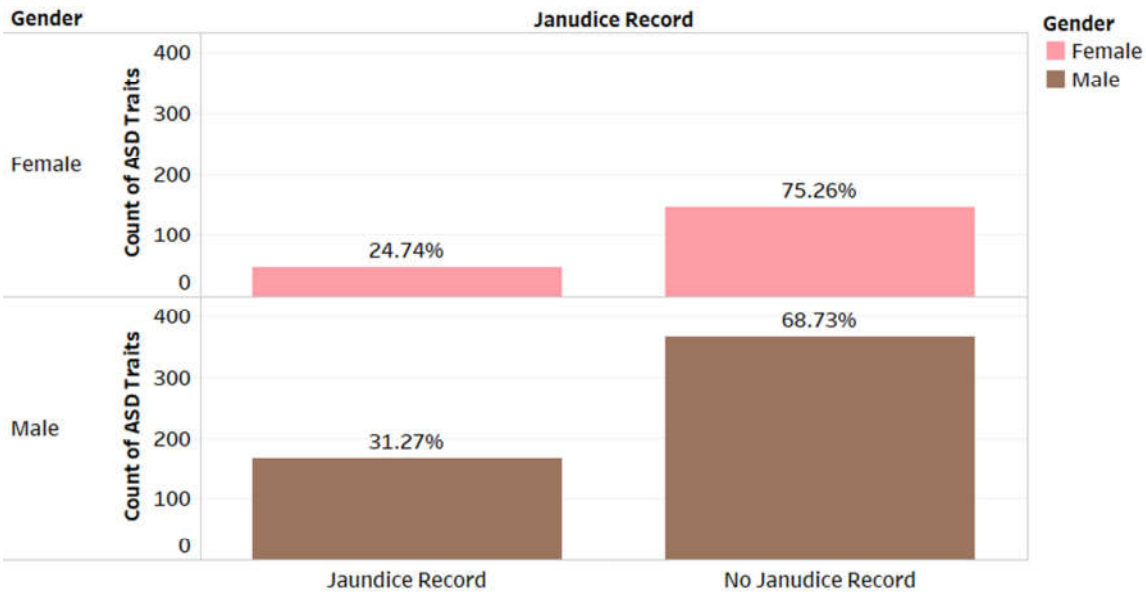

Figure 5.11: Multivariate analysis of Ethnicity with Age group and ASD traits

(Figure 5.11) show that 75.26 % of the females and 68.73% of the male has detected as ASD despite they have no sign of jaundice before. So it is evident that past jaundice records is less significant to determine ASD among toddlers.

5.9 Chi-square Test:

The Chi-Square statistic is most commonly used to evaluate Tests of Independence when using a cross tabulation (also known as a bivariate table). Cross tabulation presents the distributions of two categorical variables simultaneously, with the intersections of the categories of the variables appearing in the cells of the table. The Test of Independence assesses whether an association exists between the two variables by comparing the observed pattern of responses in the cells to the pattern that would be expected if the variables were truly independent of each other.

5.9.1 State the Hypotheses

Suppose that Variable A has r levels, and Variable B has c levels. The null hypothesis states that knowing the level of Variable A does not help to predict the level of Variable B. That is, the variables are independent.

Ho: Variable A and Variable B are independent.
Ha: Variable A and Variable B are not independent.

The alternative hypothesis is that knowing the level of Variable A can help to predict the level of Variable B.

5.9.2 Formulate Significance level.

Often, researchers choose significance levels equal to 0.01, 0.05, or 0.10; but any value between 0 and 1 can be used.

5.9.3 Analyze Sample Data

Using sample data, analyzed the degrees of freedom, expected frequencies, test statistic, and the P-value associated with the test statistic.

5.9.4 Degrees of freedom

The degree of freedom (DF) is equal to:

$$DF = (r - 1) * (c - 1)$$

Where r is the number of levels for one categorical variable, and c is the number of levels for the other categorical variable.

5.9.5 Expected Frequencies

The expected frequency counts are computed separately for each level of one categorical variable at each level of the other categorical variable. Compute r * c expected frequencies, according to the following formula.

$$E_{r,c} = (n_r * n_c) / n$$

where $E_{r,c}$ is the expected frequency count for level r of Variable A and level c of Variable B, n_r is the total number of sample observations at level r of Variable A, n_c is the total number of sample observations at level c of Variable B, and n is the total sample size.

5.9.6 Test Statistic

The test statistic is a chi-square random variable (X^2) defined by the following equation.

$$X^2 = \Sigma [(O_{r,c} - E_{r,c})^2 / E_{r,c}]$$

where $O_{r,c}$ is the observed frequency count at level r of Variable A and level c of Variable B, and $E_{r,c}$ is the expected frequency count at level r of Variable A and level c of Variable B.

5.9.7 P-value

The P-value (highly statistically significant) is the probability of observing a sample statistic as extreme as the test statistic.

5.9.8 Interpreting Results

If the sample findings are unlikely, given the null hypothesis, the researcher rejects the null hypothesis. Typically, this involves comparing the P-value to the significance level, and rejecting the null hypothesis when the P-value is less than the significance level.

Table 5.3: Results of Chi-Square Test

Discrete Features	**Significance level**	**Degree of Freedom**	**Chi-square Statistic**	**p-value**	**Ho: no association**	**Ha: association exists**
Sex, ASD traits	0.05	1	14.59242154	0.00013345	Reject	Retain
Ethnicity, ASD traits	0.05	10	43.57129272	3.93E-06	Reject	Retain
Jaundice, ASD traits	0.05	1	5.781024078	1.62E-02	Reject	Retain
Family_mem_with_ASD, ASD traits	0.05	1	0.192163458	6.61E-01	Retain	Reject
Who completed the test, ASD traits	0.05	3	1.415313128	0.701949207	Retain	Reject

First decided on a null hypothesis, H0: There is no significant association between subjects' Sex, Jaundice and Ethnicity with ASD Traits. Conversely, alternative hypothesis, Ha: There does a significant association between subjects' Sex, Jaundice and Ethnicity with ASD Traits.

5.10 T Test:

The independent samples t-test is an inferential statistical test to determine whether the difference between two groups' (ASD traits present or ASD traits absent) means are statistically significant. If so, an attribute's values can constitute a feature for the classification. The null and alternative hypotheses are the same as the Chi-squared test. The t-statistic is calculated as:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

$$where\ \bar{x}_1 = mean\ of\ sample - 1$$

$$where\ \bar{x}_2 = mean\ of\ sample - 2$$

$$n_1 = number\ of\ subjects\ in\ sample - 1$$

$$n_2 = number\ of\ subjects\ in\ sample - 2$$

$$s_1^2 = variance\ of\ sample - 1 = \frac{\sum(x_1 - \bar{x}_1)^2}{n_1}$$

$$s_2^2 = variance\ of\ sample - 1 = \frac{\sum(x_2 - \bar{x}_2)^2}{n_2}$$

Table 5. 4: Results of Welch Two Sample t-test

Welch Two Sample t-test

continuous features	t-score	degrees of freedom	p-value	Ho: $\mu 1 = \mu 2$	Ha: $\mu 1 \neq \mu 2$
Qchat-10-Score	-55.072	1006.4486	5.70E-306	reject	retain
Age_Mons	-2.0323	537.9457	0.042618	reject	retain

5.11 Elimination of Irrelevant Attributes

Attributes that do not contribute to the determination of the ASD Traits are irrelevant and thus have been removed for the following reasons:

- They contribute to the curse of dimensionality by requiring a lot more training data than what is available.

- They make gradient descent shoot unnecessarily along the path of convergence.

- They require training for longer only to find no useful parameters while also over fitting useful parameters.

So the following attributes have been scissor out of the dataset.

- **Case_No:** the primary key of the subject.
- **A1 to A10:** because these are the questions that sum up to score Q-Chat 10 that is the significant feature for the ASD Traits. So these features are irrelevant and redundant.
- **Family_mem_with_ASD:** Past family members' ASD record does not contribute to the ASD traits of the subjects.
- **Who completed the test:** Past family members' ASD record does not contribute to the ASD traits of the subjects.

5.12 Normalization of Input Features

When building deep learning models it is usually good practice to scale dataset in order to make the computations more efficient. In this step, we have scaled the data using the StandardScaler; this will ensure that dataset values have a mean of zero

and a unit variable. This transforms the dataset to be normally distributed. We use the scikit-learn StandardScaler to scale the features to be within the same range. This will transform the values to have a mean of 0 and a standard deviation of 1. This step is important because we are comparing features that have different measurements; so it is typically required in machine learning. Normalization rescales all numeric in the range [0, 1] using the formula below:

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

5.13 Generation of K-fold Datasets for Cross-Validation

When model is trained many times, we keep getting different results. The accuracies of each training have a high variance. In order to solve this problem, we use K-fold cross-validation. Usually, K is set to 5. In this technique, the model is trained on the first 4 folds and tested on the last fold. This iteration continues until all folds have been used. Each of the iterations gives its own accuracy. The accuracy of the model becomes the average of all these accuracies.

Chapter 6:

IMPLEMENTATION AND RESULTS

6.1 Tools Utilized for Implementation

The tools and their justified application are stated below:

- **Programming language:** Python 3
- This modern version of Python, being constantly updated, is perfect for DNN implementation and has therefore been chosen for the research. **libraries:** Pandas, matplotlib, NumPy , Seaborn has been chosen. **Neural networks modeling framework:** Keras
- This highly optimized framework has been used for defining the model, compiling it, running it on the provided dataset and producing results.
- **Integrated development environment (IDE):** Jupiter iPython Notebook with Python
- **Kernel:** This online IDE has been chosen for a perfect reproducibility of the implementation and its results.
- **Hardware:** a 3.20GHz CPU, 8.00 GB RAM
  Training necessary for the implementation was done by an x-64 based CPU, rather than a GPU owing to that the dataset is not as large. The fair amount of computations needed for forward and backpropagation was handled by an Intel processor: Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz, 3201 MHz, 4 Core(s), and 4 Logical Processor(s). The total Installed Physical Memory (RAM) was 8.00 GB with a total of 7.89 GB in use.
- **Platform:** Windows 10 Microsoft Windows 10 Home, manufactured by Microsoft Corporation has been our platform. The version used for this task is Version 10.0.17763 Build 17763.

6.2 Generated Confusion Matrices

In the jargon of machine learning, concretely in the problem of statistical classification, a confusion matrix (Figure 6.1) is a specific tabular layout used to explain the performance of a classification model on a set of cross-validation data for which the true labels are available.

Rows of the matrix represent the instances in a predicted class while columns represent the instances in an actual class (or vice versa). The name originates from that it makes viable to see if the system is confusing the classes (i.e. commonly mislabeling one as another).

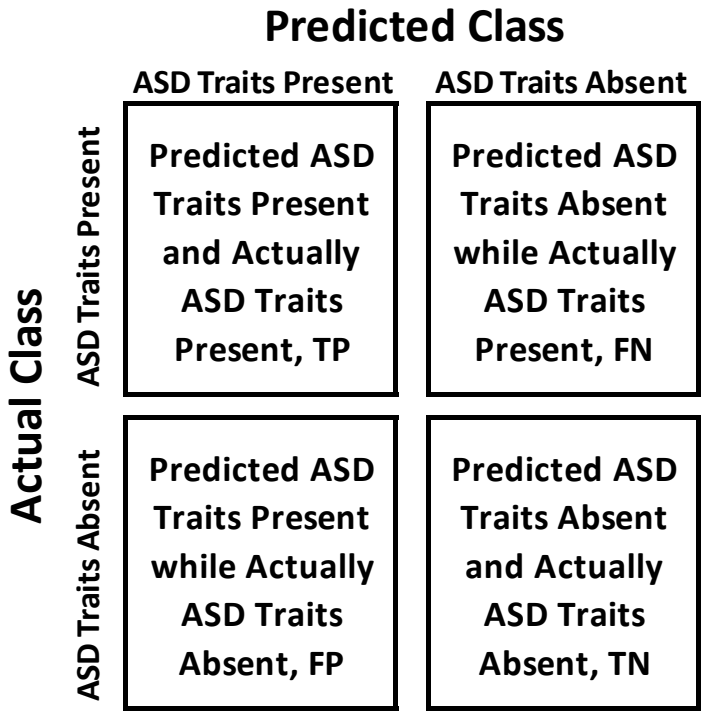

Figure 6.1: Confusion matrix for ASD classification problem

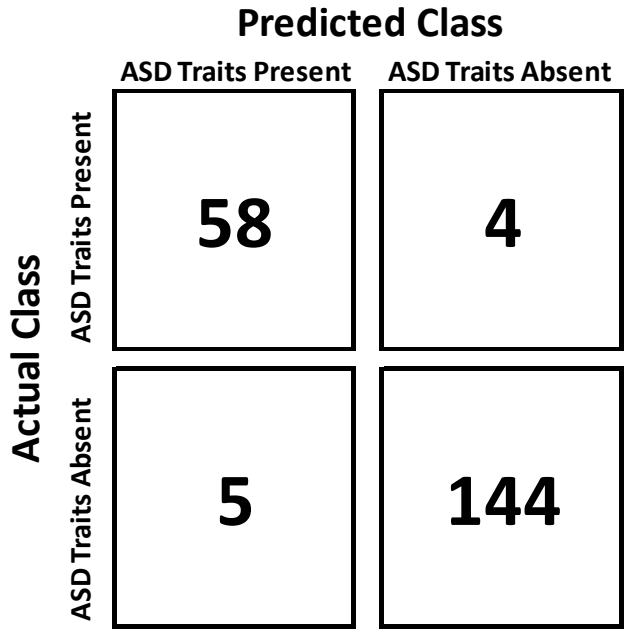

Figure 6.2: Confusion Matrix for 4-Layer ASD Classifier.

The matrix (Figure 6.2) is a special kind of contingency table, with two dimensions and identical sets of classes in both dimensions.

6.3 Performance Evaluation Metrics

For our medical diagnosis problem, we select accuracy, precision and recall and F1-score as evaluation metrics using the terms calculated in the confusion matrix.

Accuracy

• Accuracy attempts to answer the following question:

What proportion of predictions (both ASD Traits Present and ASD Traits Absent) was actually correct?

• Accuracy is mathematically defined as follows: accuracy = (TP+TN)/ (TN+FP+FN+TP)

• A model that produces no false predictions provides an accuracy of 1.0.

Precision

• Precision attempts to answer the following question:

What proportion of 'ASD Traits Present' identifications was actually correct?

• Precision has been calculated as follows: precision = TP/ (TP+FP)

• A model that produces no false positives renders a precision of 1.0.

Recall

• Recall attempts to answer the following question:

What proportion of actual 'ASD Traits Present' was identified correctly?

• Mathematically, recall has been defined as follows: recall = TP/ (TP+FN)

• A model that produces no false negatives delivers a recall of 1.0.

F1-score

• F1-score is a trade-off between accuracy and precision in that it is the harmonic mean of the two. It is mathematically defined as (2*TP)/ (2*TP+FP+FN)

• the optimal value for this metric is 1.00, signifying perfect precision and recall.

For more than two classes, the abovementioned metrics are calculated separately for each

class like the following:

$$Precision_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$Recall_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

6.4 K-Fold Cross-Validated Results

K-fold cross-validation is a technique of evaluating statistical predictive models' performance on K independent test-sets. From the whole gamut of data, different

Subsets are iteratively, randomly selected and K test-sets are formed.

While the model is trained many times different results has been generated. Each of the trainings has accuracy with high variance. In order to solve this problem, K-fold cross-validation has been used in this research. K is set to 5. In this technique, the model is trained on the first 4 folds and tested on the last fold. This iteration continues until all folds have been used. Each of the iterations gives its own accuracy. The accuracy of the model becomes the average of all these accuracies. Thus the Model Accuracy has been improved.

6.5 Results upon Training Four-layer Model

The training was first performed with incomplete tuples eliminated. This gave consistent accuracy 98%. As the second step of experimentation, applying prediction on test set and gets accuracy 92%.

Table 6. 1: Results on Training Four-Layer Model

Model	1
Weight Initializer	Random Normal
Learning Rate	0.005
Number of Layer	4
Number of Nodes	5,5,5,1
Total Parameter	146
Batch Size	10
Epoch	30
Test Accuracy	98.10%
K fold	5
Cross Validation Mean Accuracy	92.73%
Variance	0.000609108
Precision	99.32%
Recall	97.99%
F Score	98.65%

6.6 Convergence to Minimum Error

The four-layer model have been trained for 30 epochs each, for K=5. After much tuning, a learning rate of 0.005 has been chosen for the model to converge to minimal error using Adam-optimized gradient descent.

The cross entropy loss has reduced with each epoch and the 30th epoch ensured the most refined parameters with the least error (Figure 6.3) and (Figure 6.4).

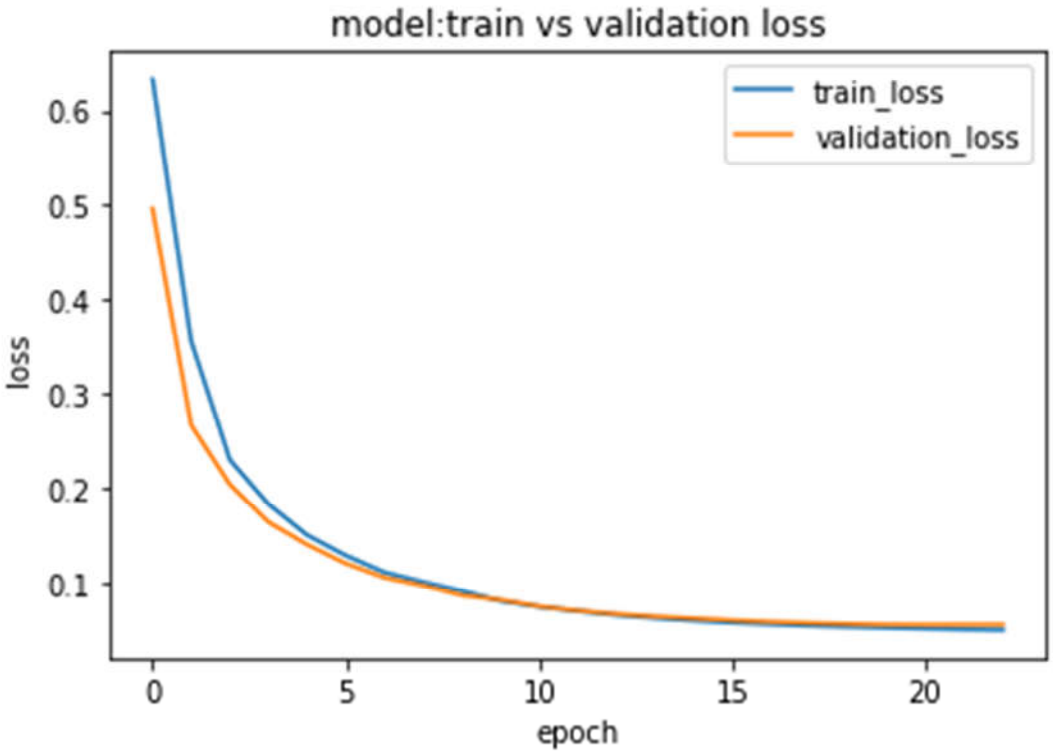

Figure 6.3: Epoch vs. Loss Curve for 4-Layer ASD Classifier.

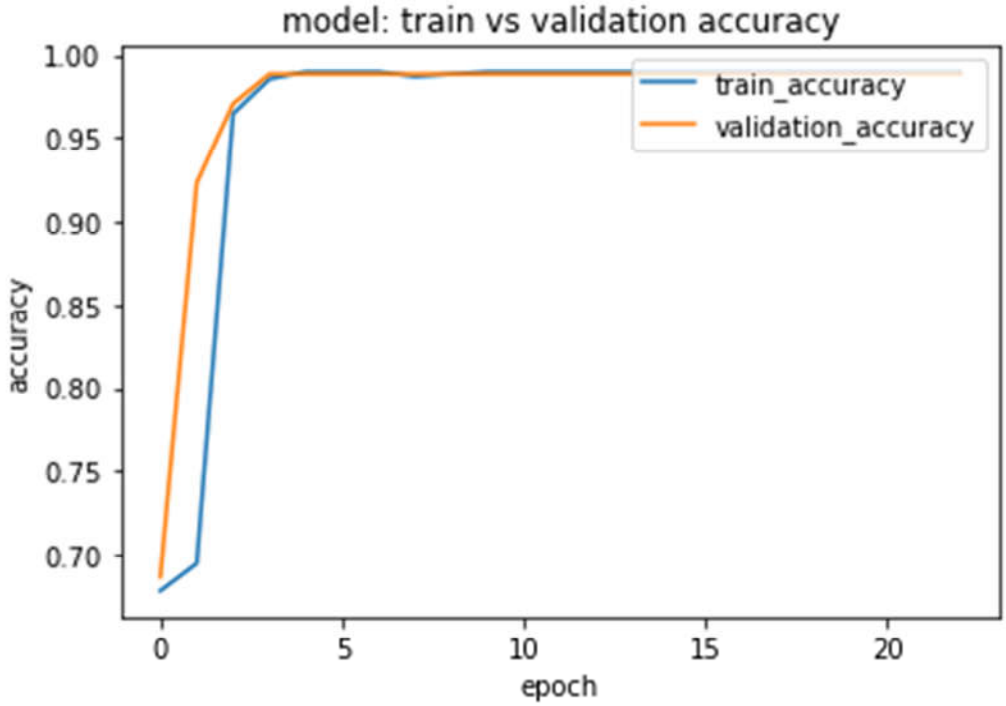

Figure 6.4: Epoch vs. Accuracy Curve for 4-Layer ASD Classifier.

With each epoch the loss is decreasing and the accuracy is increasing which ensure that the model is working completely as expected.

Chapter 7:

CONCLUSION

A growing percentage of toddlers are now threatened to have ASD. It has become a great concern among different countries while research works to detect ASD within earliest possible time is also significantly increasing. My research works have done a limited effort to understand ASD symptoms in toddlers at the same time have introduced a neural network based machine learning system that may not resolve issues but would help detecting them effortlessly. However, my work was limited to simpler data, where more complex dataset may have been posed intricate model but allowing more access to find autism disorder.

My desire to research in detecting autism among toddlers comes from my social responsibilities. It has made me enthusiastic to bring it further to promise more acceptability towards people. Since machine learning is the future for the next generation's problem solicitor to detect and decide key decisions, it is worth to extend my research further.

7.1 Achievements of the Work

The research yielded the following benefits:

• **Usage of structured data:** The usage of a structured database has omitted the necessity for costly radiological medical images. Structured data is cheaper to obtain and of greater business value.

• **Simple, unique model:** The study proposed simple, multi-layer neural networks for the diagnosis of the disease which took less time to train and to tune.

• **No RAM shortage:** Running out of primary memory is a common complaint of researchers. The study comes round this by conducting training using few batch sizes.

• **Reproducible work:** The study has been carried out on state-of-the-art platforms and all the hyper parameters are explicitly stated. This makes the research reproducible, which is of utmost importance to the research community.

7.2 Limitations

Every scientific study is permitted to have some limitations, upon which further researches expand roots. This work is no exception and has the following limitations.

- **Scarcity of Data:** For deep NNs, the larger the data, the greater the results. The dataset used here is not large enough. If the more large dataset can be used the more robust model can be built.
- **Less Features:** Number of features was limited to experiment the model strength. If number of features
- **Absence of clinical Data:** Since this dataset was not based on clinical test and observations, pragmatic outcomes are limited.

7.3 Future Scopes

The work has explored new dimensions for current and future researchers conducting research using medical data. Some are stated below.

- **Augmentation of data:** In future, this work can be extended by collecting data from different hospitals and diagnostic centers which will enrich the existing dataset. More and more features would be explored and more classes of ASD would have been incorporated if a diversity of data could have been introduced. This will also enable the proposed models to perform even better.

- **Implementation of deeper models:** Overcoming the limitation of scarce data, deeper models can be trained for improved metrics. It has been proved that deeper architectures are efficient in recognizing greatly complex patterns hidden in data.

- **Different optimizations:** This work has used Adam gradient descent optimization for optimizing the objective error function. Many modern optimizations are now in use such that momentum, Xavier, RMSProp etc. These are proved to optimize the parameters in a short time to a great level of refinement and can be experimented with.

- **Evaluation of other metrics:** This study chose the most compelling metrics for medical diagnosis systems. However, other metrics can also be experimented with, such as false positive rate, false negative rate, specificity etc. These will enable researchers to look at the performance from different perspectives of utility.

REFERENCES

Alessandro Crippa, Christian Salvatore, Paolo Perego, Sara Forti, Maria Nobile, Massimo Molteni, Isabella Castiglioni. "Use of Machine Learning to Identify Children with Autism and Their Motor Abnormalities." *Journal of Autism and Developmental Disorders*, 2015.

Anibal Sólon Heinsfeld, a Alexandre Rosa Franco,b,c,d R. Cameron Craddock,f,g Augusto Buchweitz,b,d,e and Felipe Meneguzzia,b,*. "Identification of autism spectrum disorder using deep learning and the ABIDE dataset." *Neuroimage Clin*, 2018.

Bishop-Fitzpatrick, Movaghar, Greenberg , DaWalt , Brilliant , Mailick MR. "Using machine learning to identify patterns of lifetime health problems in decedents with autism spectrum disorder." *Official Journal for the International Society for Autism Research*, 2018: 1120-1128.

Chen CP1, Keown CL2, Jahedi A3, Nair A4, Pflieger ME5, Bailey BA6, Müller RA1. "Diagnostic classification of intrinsic functional connectivity highlights somatosensory, default mode, and visual regions in autism." *Neuroimage Clin*, 2015.

Emily T. Prud'hommeaux, Brian Roark, Lois M. Black, and Jan van Santen. "Classification of atypical language in autism." *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics.* Portland, Oregon: Association for Computational Linguistics, 2011. 88–96.

G. Bussu, corresponding author E. J. H. Jones, T. Charman, M. H. Johnson,J. K. Buitelaar, and the BASIS Team. " Prediction of Autism at 3 Years from Behavioural and Developmental Measures in High-Risk Infants: A Longitudinal Cross-Domain Classifier Analysis." *Journal of Autism and Development Disorders*, 2018.

Kayleigh Hyde, Amy-Jane Griffiths, Cristina Giannantonio, Amy Hurley-Hanson, Erik Linstead. "Predicting Employer Recruitment of Individuals with Autism Spectrum Disorders with Decision Trees." 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

Li, Baihua Sharma, Arjun Meng, James Purushwalkam, Senthil Gowen, Emma. "Applying machine learning to identify autistic adults using imitation: An exploratory study." *US: Public Library of Science*, 2017.

Narayanan, Daniel Bone Somer L. Bishop Matthew P. Black Matthew S. Goodwin Catherine Lord Shrikanth S. "Use of machine learning to improve autism screening and diagnostic instruments: effectiveness, efficiency, and multi-instrument fusion." *The Journal of Child Psychology and Psychiatry*, 2016.

Yongxia Zhou, Fang Yu,Timothy Duong. "Multiparametric MRI Characterization and Prediction in Autism Spectrum Disorder Using Graph Theory and Machine Learning." *PLOS ONE*, 2014.

Zhang F1, Savadjiev P2, Cai W3, Song Y3, Rathi Y2, Tunç B4, Parker D4, Kapur T2, Schultz RT5, Makris N2, Verma R4, O'Donnell LJ2. "Whole brain white matter connectivity analysis using machine learning: An application to autism." *Neuroimage*. , 2018 .