

**AUTISM SPECTRUM DISORDER
DETECTION USING MACHINE LEARNING
TECHNIQUES**

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and
Engineering

By

KALYANAPU SRIDHAR (Reg No – 39110439)

SAGAM NIKHIL REDDY (Reg No - 39110864)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

APRIL - 2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with 'A' grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119
www.sathyabama.ac.in



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KALYANAPU SRIDHAR (39110439)** and **SAGAM NIKHIL REDDY (39110864)** who carried out the Project Phase-2 entitled “**AUTISM SPECTRUM DISORDER DETECTION USING MACHINE LEARNING TECHNIQUES**” under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr. R. AROUL CANESSANE, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on -----

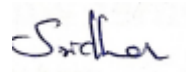
Internal Examiner

External Examiner

DECLARATION

I, **KALYANAPU SRIDHAR (39110439)**, hereby declare that the Project Phase-2 Report entitled **AUTISM SPECTRUM DISORDER DETECTION USING MACHINE LEARNING TECHNIQUES**” done by me under the guidance of **Dr. R. AROUL CANESSANE, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 25-04-2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **DR.R. AROUL CANESSANE M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 project work.

I wish to express my thanks total Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Autism Spectrum Disorder (ASD) is a neurodevelopmental ailment related to large healthcare expenses, and early analysis can significantly lessen expenses. Unfortunately, inside the outdoors, the time and fee of the product isn't always well worth it. With the financial impact of autism and the prevalence of ASD international growing daily, there's a pressing need to expand strategies that are easy to put into effect and powerful for screening. Thus, speedy, and low-cost screening for ASD is necessary to help health experts and tell humans whether or not they need to have a proper clinical diagnosis. The speedy increase inside the quantity of ASD instances global has brought about demands for applicable behavioral statistics. However, such records are rare, making it hard to carry out a detailed analysis to improve the performance, sensitivity, specificity, and accuracy of the predictive procedure of ASD screening. There are presently very confined facts available on autism for clinical trials or screening, and maximum of them are genetic in nature. We consequently advise a new dataset relevant to adult screening for autism, which includes 20 characteristics that can be used for similarly analysis, especially for figuring out influential autistic traits and improving the classification of ASD instances. In this dataset, we document ten behavioral traits (AQ-10-Adult) plus ten man or woman traits that have been shown to be powerful in evaluating cases of ASD among technological know-how controls.

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		vi
1	INTRODUCTION		1
2	LITERATURE SURVEY 2.1 Inferences from Literature Survey		3
	2.2 Open problems in Existing System		5
3	REQUIREMENTS ANALYSIS		7
	3.1	Feasibility Studies/Risk Analysis of the Project	7
	3.2	Software Requirements Specification Document	8
4	DESCRIPTION OF PROPOSED SYSTEM		9
	4.1	Selected Methodology or process model	9
	4.2	Architecture / Overall Design of Proposed System	13
	4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	13
	4.4	Project Management Plan	19
	4.5	Transition/ Software to Operations Plan	29
5	IMPLEMENTATION DETAILS		30
	5.1	Development and Deployment Setup	30
	5.2	Algorithms	31
	5.3	Testing	34
6	RESULTS AND DISCUSSION		42

7	CONCLUSION		44
	7.1	Conclusion	44
	7.2	Future work	47
	REFERENCES		48
	APPENDIX		50
	A. SOURCE CODE		50
	B. SCREENSHOTS		54
	C. RESEARCH PAPER		58

LIST OF FIGURES

FIGURE	NAME	Page
No.		
4.1	missing values in ASD data	9
4.2	k-fold Cross Validation	11
4.3	Histograms for original & ID3 predicted ASD classes.	12
4.4	Architecture	13
4.5	Complexity graph	14
4.6	Learning Curves	15
4.7	Random Forest Diagram	24
4.8	support vector machine	24
4.9	Logistic Regression Model	26
4.10	MLP Architecture	27
5.1	A gini diagram of Decision Tree.	32
5.2	Factor plot: kind = 'boxplot'	35
5.3	Factor plot: kind = 'swamp'	35
5.4	jaundice	36
5.5	A violine plot depicting multi-feature relationship.	37
5.6	target class of individual with ASD	39
5.7	normalized weights	40
7.1	Free-Form Visualization	44

LIST OF TABLES

FIGURE	NO	NAME	Page
4.1		Confusion Matrix for ASD Data Set	15
4.2		Confusion Matrix	17
4.3		List of Attributes in ASD dataset.	20
5.1		Confusion Matrix for ASD Data Set	31
5.2		Comparison of metrics using different learning algorithm	41

CHAPTER 1

INTRODUCTION

Autistic Spectrum Disorder (ASD) is the name for a group of developmental disorders impacting the nervous system. ASD symptoms range from mild to severe: mainly language impairment, challenges in social interaction, and repetitive behaviors. Many other possible symptoms include anxiety, mood disorders and Attention Deficit/Hyperactivity Disorder (ADHD). ASD has a significant economic impact in the healthcare domain, both due to the increase in the number of ASD cases, and because of the time and costs involved in diagnosing a patient. Early detection of ASD can help both patients and the healthcare sector by prescribing patients the therapy and/or medication they need and thereby reducing the long-term costs associated with delayed diagnosis. Thus, health care professionals across the globe have an urgent need for easy, time-efficient, robust, and accessible ASD screening methods that can accurately predict whether a patient with certain measured characteristics has ASD and inform individuals whether they should pursue formal clinical diagnosis. However, challenges remain. Pursuing such research necessitates working with datasets that record information related to behavioral traits and other factors such as gender, age, ethnicity, etc. Such datasets are rare, making it difficult to perform thorough analyses to improve the efficiency, sensitivity, specificity, and predictive accuracy of the ASD screening process. At present, very limited autism datasets associated with clinical, or screening are available and most of them are genetic in nature. These data are extremely sensitive and hard to collect for social and personal reasons and the regulations around them.

Machine learning is a field of artificial intelligence that allows a computer to learn and analyze data—new and old—without human supervision. Various applications like e commerce and information technology exist to gather an enormous amount of data and make predictions. In the case of medicine, machine learning could scan single nucleotide polymorphisms (SNPs) for the prevalence of specific genes that are linked to autism spectrum disorders (ASD). ASD is neurodevelopmental, is diagnosable in the first two to three years of one's life, and is characterized by social, communication, and behavioral deficits. SNPs are genetic sequence variations that take place at a position where a single DNA nucleotide is switched to another.

the genes that are linked to autism spectrum disorders (ASD) may serve as useful biomarkers for ASD diagnosis and help to understand the exact genetic causes of ASD. The purpose of this project is to identify ASD-related SNPs based on SNP genotyping in genomic DNA in a large cohort of ASD patients and unaffected related individuals. The dataset retrieved from the Gene Expression Omnibus database (GSE6754) contains more than 6,000 samples from 1,400 families. The SNPs are ranked by the distance in three-dimensional genotype count space between all the affected and unaffected subjects in the cohort. The results demonstrate that the SNPs with the highest-ranking distances are likely to be linked to ASD. High-ranking SNPs that currently have no known links to ASD can potentially become novel biomarkers. The involvement of the genes containing these SNPs in the biological pathways that might be relevant to ASD is analyzed using a pathway database. The top-ranking SNPs can be used as attributes for machine learning models to identify ASD patients based on their genetic sequencing data.

CHAPTER 2

LITERATURE SURVEY

2.1. INFERENCES FROM LITERATURE SURVEY

In 2018, the CDC Autism and Developmental Disabilities Monitoring Network reports that about 1 in 59 U.S. children and counting are diagnosed with ASD. When machine learning plays a role in transforming several industry sectors, most executives think that artificial intelligence maximizes productivity in the job economy.

Likewise, data science can solve real-world problems in natural science, international development, humanities, and many other disciplines that involve very large sets of information. Since clinical autism research is being produced over the years, machine learning.

Machine learning emerges from the latest technologies that can rely on frequently used computations to make decisions based on self-adaptable new data. Companies that generate large volumes of data are usually excited about data mining, algorithm design, and cheaper data storage and processing. When machine learning is used in bioinformatics, scientists most likely use neural networks, genetic algorithms, and fuzzy logic. One of the greatest problems this project contributes to is the classification of genes that are impacted by an illness or disorder and are distinguishable from normal genes.

Researchers cannot clearly tell how autism is being structured, but linkage scans and copy number variations (CNV) in over 1,000 families with at least two affected individuals can explain the possibility of autism risk loci being chromosome 11p12-p13 under linkage analysis and neurexins under CNV. Obviously, linkage screening can become a diagnostic tool for autism ancestry.

In P.P. Sans research paper, a combination of a deep neural network with support vector machine (SVM) classifier at the last layer was also proposed. Other previous works are based on intra-subject approaches. Some cross- subject approaches have also been proposed by combining EEG samples from all subjects, followed by

3

splitting them into training and testing randomly, like H. Zeng et al. This approach is naturally random and so it ends up mixing some training subjects samples with the testing ones, which is not cross-subject.

In Y. Liuet all s research paper, the authors perform domain adaptation, a branch of transfer learning, to adapt the data distributions of source and target so that the classification could be more efficient in a cross- subject scenario. Md. Yousuf Hossain

et al [3] proposed a non-intrusive system using the eye closure ratio as the input parameter. In Y. Liu et al's paper, EEG features, statistics, higher order crossing, fractal dimension, signal energy, and spectral power were extracted and combined with several classifiers, such as logistic regression, linear discriminate analysis, 1-nearest neighbor, linear SVM, and naïve Bayes. Mika Sunagawa et al [4] proposed a model that was accurately capable of sensing the entire range of stages of distraction, from weak to strong.

Monagi H. Alkinani et al [5] published an extensive analysis of comparisons between various deep learning-based techniques for recognizing distraction, drowsiness, fatigue and aggressiveness of a driver. Whereas Joao Ruivo Paulo et al [7] explored the methodology for observing distraction using EEG signals in a sustained attention driving task, with results showing a better-balanced accuracy of 75.87% and higher for leave-one-out cross-validation. Wang et al., N. Hatami et al., and Z. Zhao et al's methodologies used (recurrence plots and gramian angular fields), they have been successfully applied in computer vision algorithms combined with deep learning, these have been used in recent works in the EEG research domain, but still are relatively unexplored.

Several studies have made use of machine learning in various ways to improve and speed up the diagnosis of ASD. Duda et al. [5] applied forward feature selection coupled with under sampling to differentiate between autism and ADHD with the help of a Social Responsiveness Scale containing 65 items. Deshpande et al. [4] used metrics based on brain activity to predict ASD. Soft computing techniques such as probabilistic reasoning, artificial neural networks (ANN), and classifier combination have also been used [15]. Many of the studies performed have talked of automated ML models which only depend on characteristics as input features. A few studies

4

relied on data from brain neuroimaging as well. In the ABIDE database, Li et al. [14], extracted 6 personal characteristics from 851 subjects and performed the implementation of a cross-validation strategy for the training and testing of the ML models. This was used to classify between patients with and without ASD, respectively. Thabtah et al. [21] proposed a new ML technique called Rules-Machine Learning (RML) that offers users a knowledge base of rules for understanding the underlying reasons behind the classification, in addition to detecting ASD traits. Al Banna et al. [1] made use of a personalized AI-based system which assists with the monitoring and support of ASD patients, helping them cope with the COVID-19 pandemic.

2.2. OPEN PROBLEMS IN EXISTING SYSTEM

I was able to find open-source data available at UCI Machine Learning Repository. The data was made available to the public recently on December 24th, 2017. The data set, which I will be referring to as the ASD data set from here on out, came with a .csv file that contains 704 instances that are described by 21 attributes, a mix of numerical and categorical variables. A short description of ASD dataset can be found on this page. This data set was denoted by Prof Fadi Fayez Thabtah, Department of Digital Technology, MIT, Auckland, New Zealand, With the available ASD data on individuals my goal is to make predictions regarding new patients and classify them into one of two categories: patient has ASD or patient does not have ASD.

In other words, we are working on a binary classification problem with the ultimate goal of being able to classify new instances, i.e., when we have a new adult patient with certain characteristics, we would like to be able to predict whether or not that individual has high probability of having ASD.

Machine learning emerges from the latest technologies that can rely on frequently used computations to make decisions based on self-adaptable new data. Companies that generate large volumes of data are usually excited about data mining, algorithm design, and cheaper data storage and processing. When machine learning is used in bioinformatics, scientists most likely use neural networks, genetic algorithms, and fuzzy logic. One of the greatest problems this project contributes to is the

classification of genes that are impacted by an illness or disorder and are distinguishable from normal genes. Researchers cannot clearly tell how autism is being structured, but linkage scans and copy number variations (CNV) in over 1,000 families with at least two affected individuals can explain the possibility of autism risk loci being chromosome 11p12-p13 under linkage analysis and neurexins under CNV. Obviously, linkage screening can become a diagnostic tool for autism ancestry.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

ASD Symptoms usually recognized by observation. In Older and adolescents who go to school, ASD symptoms are usually identified by their parents and teachers. After that ASD symptoms are evaluated by a special education team of the school. These school team suggested these children visit their health care doctor for required testing. In adults identifying ASD symptoms is very difficult than older children and adolescents because some symptoms of ASD may be overlap with other mental health disorders.

There are some social interaction and communication problems like as:

- Inappropriate laughing and giggling
 - No sensitivity of pain
 - Not able to make eye contact properly
 - No proper response to sound
 - May not have a wish for cuddling
 - Not able to express their gestures
 - No interaction with others
 - Inappropriate objects attachment
 - Want to live alone
 - Using echo words etc.
 - Less sensitive than another person in some cases like light, noise People with ASD also have difficulty with constrained interests and consistently repetition of behaviors. The following list presents specific examples of the types of behaviors.
 - Repeating certain behaviors like repeating words or phrases much time. •
- The Person will be upset when a routine is going to change.
- Having a little interest in certain matters of the topic like numbers, facts, etc. •
- Less sensitive than another person in some cases like light, noise, etc.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

We will use supervised machine learning to refer to creating and using models that

are learned from data, i.e., there is a set of data labeled with the correct answer for the model to learn from. I will also apply a feature selection algorithm to figure out which of the 20 variables are most important in determining whether an individual has ASD or not. This work aims to explore several competing supervised machine learning classification techniques namely:

- Decision Trees
- Random Forests
- Support Vector Machines (SVM)
- k-Nearest Neighbors (KNN)
- Naive Bayes
- Logistic Regression
- Linear Discriminant Analysis (LDA)
- Multi-Layer Perceptron (MLP)

Machine learning emerges from the latest technologies that can rely on frequently used computations to make decisions based on self-adaptable new data. Companies that generate large volumes of data are usually excited about data mining, algorithm design, and cheaper data storage and processing. When machine learning is used in bioinformatics, scientists most likely use neural networks, genetic algorithms, and fuzzy logic. One of the greatest problems this project contributes to is the classification of genes that are impacted by an illness or disorder and are distinguishable from normal.

4.1. SELECTED METHODOLOGY OR PROCESS MODEL

4.1.1 Data Preprocessing

Unfortunately, this data set does have a lot of invalid or missing entries that are represented with question marks. Thus, we must preprocess our data so it is ready to be used as input for machine learning algorithms. Here, we begin, by replacing entries with the symbol “?” and convert them into ‘NAN’ (not a number). We come to find that there are quite a few missing values in the data set, but the missing values are random (verified through Python code, see Figure 13). In other words, it seems appropriate to simply drop every row that contains the missing data and we will not run the risk of biasing our data.

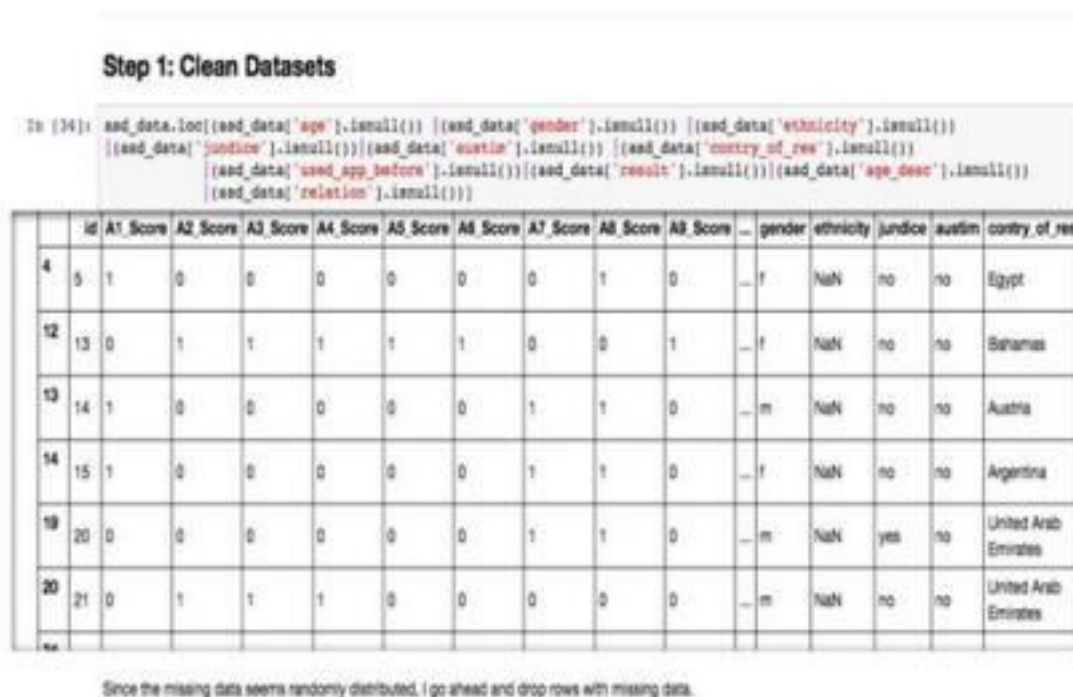


figure 4.1: missing values in ASD data

Next, we split the data into features and target label and normalize the numerical variables ‘age’ and ‘result’, using the Min Max Scaler feature in Python. For the categorical variables we use the one-hot encoding scheme. This results in 94 total

9

features after one-hot encoding. Additionally, we need to convert the non-numeric target variable ‘Class/ASD’ to numerical values for the learning algorithm to work. Since there are only two possible categories for this label (“yes” or “no”), we can avoid using one-hot encoding and simply encode these two categories as 0 and 1, respectively.

4.1.2 Implementation

Assuming that the available data for analytics fairly represents the real-world process that we wish to model and the process is expected to remain relatively stable, then the data we currently have should be a reasonable representation of the data we expect to see in the future. As a result, withholding some of the current data for testing is a fair and justified way to provide an honest assessment of our model. Thus, we split the given data into two parts. 80% of the data will be used to train the model and this data will be referred to as the training data set and 20% of the data will be reserved for testing the accuracy and effectiveness of the model on data that the model has never seen before and will be referred as the testing data set. Thus, our training set has 487 samples, and the testing set has 122 samples. Below is a flowchart that represents the whittling process of the raw data:

4.1.3 Cross Validation

Cross-validation is a model validation technique for assessing how the results of a machine learning algorithm will generalize to an unseen data set. The goal of cross validation is to define a dataset to “test” the model in the training phase (i.e., the cross-validation dataset), in order to limit problems like under fitting or overfitting, and give an insight on how the model will generalize to an independent unknown dataset.

4.1.4 k-Fold Cross Validation

One of the disadvantages in performing cross-validation is that we lose quite a bit of data that could have been used to train the model and thus possibly arrive at more correct predictions. In a traditional train-test split, the error metric can have high variance, i.e., the error may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made. To overcome these

10

difficulties, another popular technique for model assessment with the same flavor as cross-validation but slight variation is called k-Fold Cross Validation is used.

The advantage of k-fold cross validation is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once and gets to be in a training set (k-1) time. The variance of the resulting error estimate is reduced as k is increased. Since we preprocessed the ASD dataset, we didn't run into any problems or difficulties. To implement each of the mentioned methods, we imported and used the following Python.



Figure 4.2: k-fold Cross Validation

4.1.5 modules from Scikit Learn.

- from sklearn.tree import Decision Tree Classifier
 - from sklearn. ensemble import Random Forest Classifier
 - from sklearn import svm
 - from sklearn import neighbors
 - from sklearn. Naïve bayes import Multinomial NB
 - from sklearn. Linear model import Logistic Regression
- from sklearn. discriminant_analysis import
 LinearDiscriminantAnalysis • from keras models import Sequential

11

As we can see in the Figure 5, there is a clear class imbalance in the target class of individual with ASD. In this situation, we decided to apply AUC-score as well as F1-score in addition to k-Cross validation accuracy as accuracy may not be a proper metric to use in cases like ours where we can observe a definitive class imbalance.

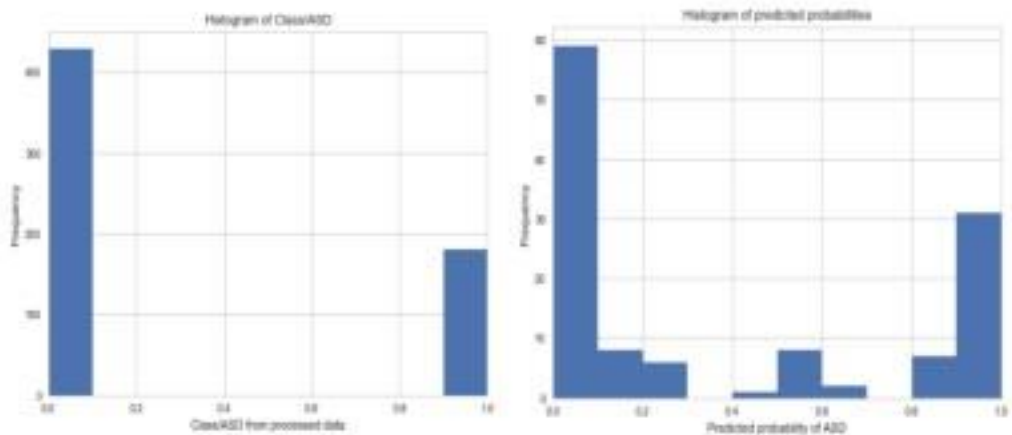
To calculate the desired metrics, I imported and used the following Python

```
modules: • from sklearn. cross_validation import cross_val_score
cross_val_score (*, features_final, asd_classes, cv=10, scoring='roc_auc').
mean() • from sklearn. metrics import fbeta_score
• from sklearn.model_selection import cross_val_scoreclf =
Classifier(random_state=1)
cv_scores = cross_val_score(clf, features_final, asd_classes, cv=10) cv_
scores.mean()
```

```

• model = Sequential ()
model.add(Dense(8, activation='relu', input_dim= 94))
model.add(Dropout(0.2))
model.add(Dense(1, kernel_initializer='normal',
activation='sigmoid')) model.summary()
score = model.evaluate(X_test, y_test, verbose=0)

```

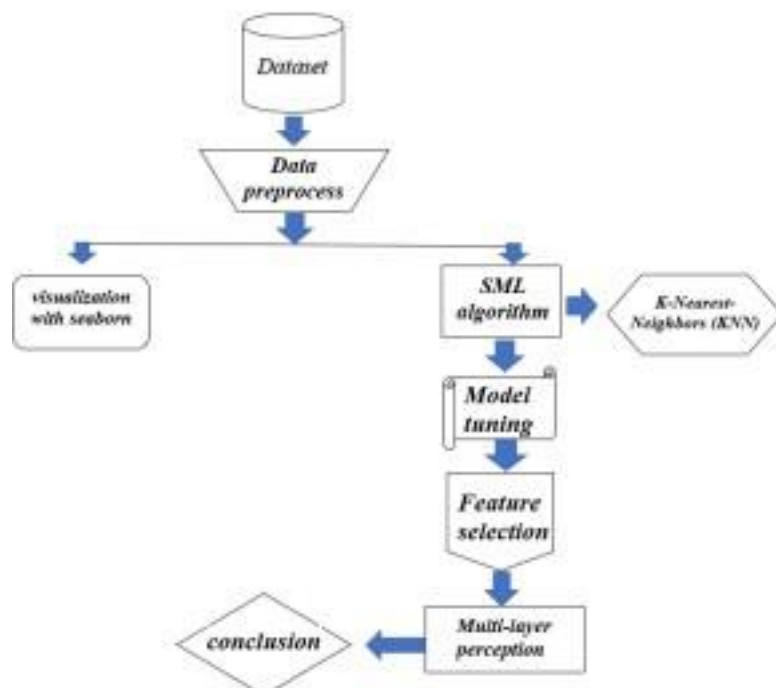


(a) Histogram for ASD classes for cleaned data. (b) Histogram for predicted ASD classes with DT.

Figure 4.3: Histograms for original & ID3 predicted ASD classes.

12

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED



SYSTEM

Figure 4.4: Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

In order to choose the appropriate model that avoids Underfitting or Overfitting the data we will analyze the Bias-Variance Trade-Off, Model Complexity Graph, Learning Curves and Receiver Operator Characteristic Curves (ROC). To measure the effectiveness of each classification model we will study the accuracy score along with the precision, recall, F-Beta Score 3 and confusion matrix.

Definition 1.1. Model Complexity Graph

Our goal is to always find the model that will generalize well to unseen data. A model complexity graph plots the training error and cross validation error as the model's complexity varies, i.e., the x-axis represents the complexity of the model (such as degree of the polynomial for regression or depth of a decision tree) and the y-axis measures the error in training and cross validation. Note that the size of the data set remains constant even

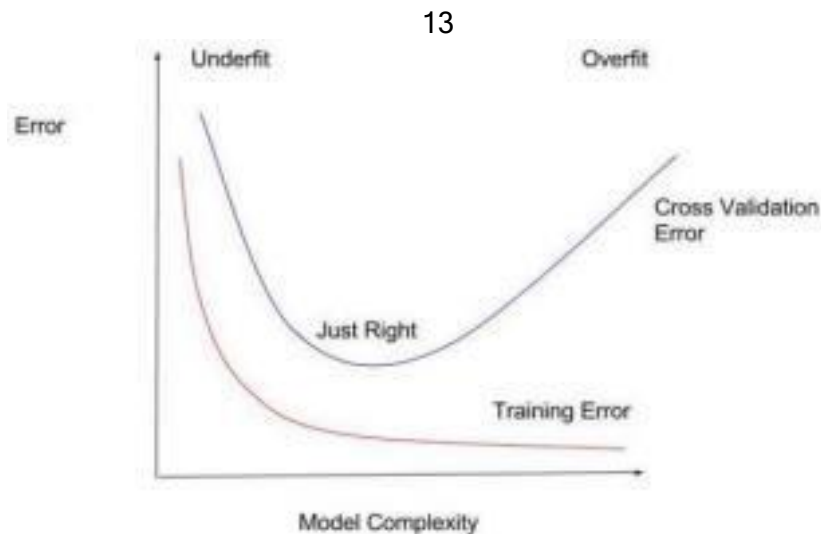


Figure 4.5: Complexity graph

while model complexity varies. Figure (1) below shows a typical model complexity graph. On the left we have a model that underfits and we see high training and cross validation errors, while on the right we have a model that overfits and gives us low training error but high cross validation error. The model in the middle is just right, with relatively low training and testing errors, and this is the model we should select. The best predictive and fitted model (neither underfitting nor overfitting) would be where

the cross-validation error achieves its global minimum.

Learning Curves Learning curves provide a way for us to distinguish between underfitting, overfitting, or the model that is just right. Learning curves are a plot of both the training error and cross validation error versus the size of the training set. In other words, a learning curve in machine learning is a graph that compares the errors of a model on training and cross validation data over a varying number of training instances. By separating training and cross validation sets and graphing errors of each, learning curves help us understand how well the model will potentially generalize to the unseen testing data. Typically, we observe that the training error increases with the size of the training set, since we have more points to fit the model to. A learning curve allows us to verify when a model has learned as much as it can about the data. When this occurs, both the training and cross validation errors reach a plateau and there is a consistent gap between the two error rates. In the case of

14

underfitting or a high-bias model the two curves converge to a high point. In the perfect model the curves converge to a low point. In the case of overfitting or a high variance models, the curves do not converge; the training curve stays low and the cross-validation error stays high, since models that overfit tend to memorize the training data.

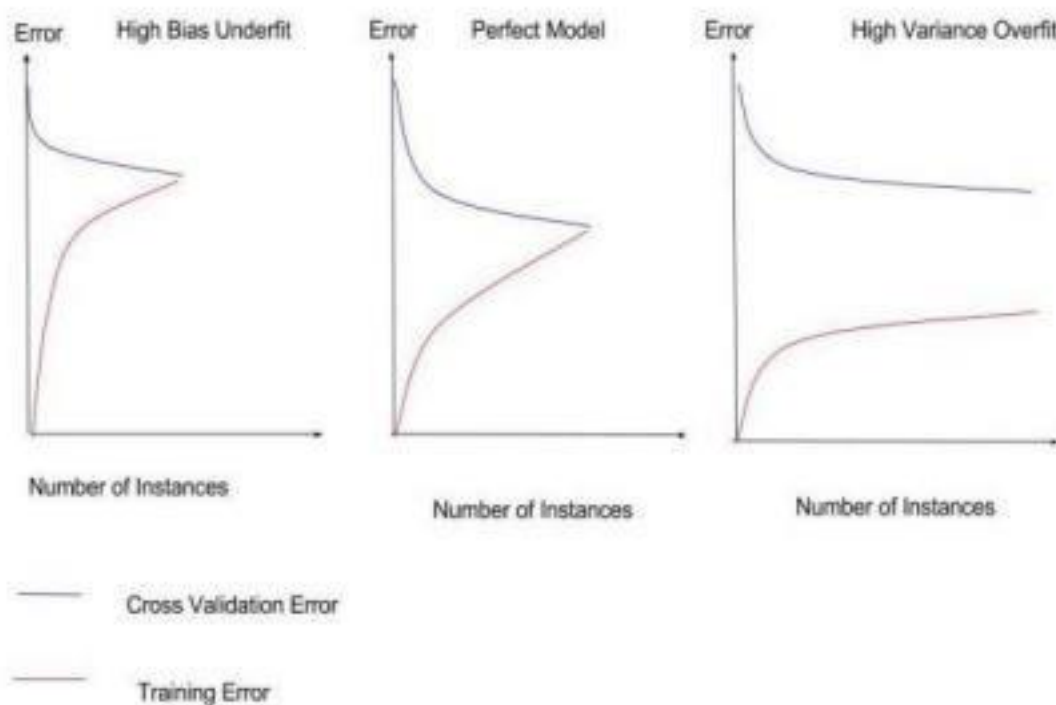


Figure 4.6: Learning Curves

4.3.1 Decision Trees: ID3 Algorithm.

The confusion matrix in this case takes the form:

	individual with ASD	individual with no ASD
predicted: ASD = 'YES'	79	0
predicted: ASD = 'NO'	0	43

Table 4.1: Confusion Matrix for ASD Data Set

The accuracy of the Decision Tree classifier which measures overall how often this Classifier is correct is:

15

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = 0.9508$$

and the precision of the Decision Tree Classifier which measures how often the prediction is correct when a positive outcome is predicted is:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 0.9302$$

The sensitivity of the method measures how often the prediction is correct given that the actual outcome is positive:

$$\text{sensitivity} = \frac{\text{TP}}{\text{FN} + \text{TP}} = 0.9302$$

and the specificity measures how often the prediction is correct given that the actual outcome is negative:

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 0$$

Finally, we calculate the false positive rate as

$$\text{false positive rate} = \frac{\text{FP}}{\text{TN} + \text{FP}} = 0.0379$$

which measures how often the prediction is correct given that the outcome was actually negative. Note the values of the following evaluation metrics.

- cross-validation score=1.0

- AUC Score=1.0

- F-Beta Score=1.0 with Beta=0.5.

Note that the Decision Tree Classifier will serve as our Benchmark Model. This means that all future models that we study will be compared to the Decision Tree classifier accuracy only as good as but not better than the Decision Tree classifier. In other words, no future model can surpass the Decision Tree classifier in terms of performance statistics.

4.3.2 Receiver Operating Characteristic (ROC) Curves:

The receiver operating characteristic curve is a commonly used summary for assessing the diagnostic ability of a binary classifier over all possible thresholds. It is a plot of the true positive rate (also known as sensitivity or recall) versus the false positive rate (also known as specificity). The area under the ROC curve gives an idea about the overall performance of a classifier, summarized. The bigger the area under its curve, the better the overall performance of the binary classifier. Given a set of labeled data and a predictive model, every data point will lie in one of the four categories: True positive: The adult individual DID have ASD and we correctly predicted that the individual would have ASD. True negative: The individual did NOT have ASD and we correctly predicted that the individual would NOT have ASD. • • 5 False positive (Type 1 Error): The individual did NOT have ASD, but we incorrectly predicted that the individual would have ASD. False negative (Type 2 Error): The individual DID have ASD, but we incorrectly predicted that the individual would NOT have ASD. These counts can also be represented in a confusion matrix (see Table 4.2) that allows us to visualize the performance of a supervised machine learning algorithm:

Table 4.2: Confusion Matrix

4.3.3 Confusion Matrix

Accuracy:

Accuracy measures how often the classifier makes the correct prediction. In other words, it is the ratio of the number of correct predictions to the total number of predictions (the number of test data points). Accuracy is defined as the fraction of correct predictions.

Precision:

Precision measures how accurate our positive predictions were i.e., out of all the points predicted to be positive how many of them were actually positive

Recall

Recall measures what fraction of the positives our model identified, i.e., out of the points that are labelled positive, how many of them were correctly predicted as positive. Another way to think about this is what fraction of the positive points were my model able to catch?

For classification problems that are skewed in their classification distributions like ours where we have a total of 609 records (after data preprocessing) with 180 individuals diagnosed with ASD and 429 individuals not diagnosed with ASD, accuracy by itself is not a very good metric. Thus, in this case precision and recall come in very handy.

These two metrics can be combined to get the F1 score, which is weighted average(harmonic mean) of the precision and recall scores. This score can range from 0 to 1, with 1 being the best possible F1 score (we take the harmonic mean as we are dealing with ratios).

F1 – Score

The F1– Score is the harmonic mean of precision and recall and thus must lie between them.

F1– Score is closer to the smaller of precision and recall than the higher number. As a result, if one of the precision or recall value is low the F1 – Score raises a flag

4.4. PROJECT MANAGEMENT PLAN

4.4.1 Data set

Our data set involves ten behavioral features (“AQ-10-Adult”) (binary data) and ten individual characteristics such as “Gender”, “Ethnicity”, “Age”, etc (categorical data) and one numerical data (“result”) below lists all variables involved in the ASD data set.

Our raw data set contains 704 instances with 189 individuals diagnosed with ASD. Thus, the percentage of individuals diagnosed with ASD is 26.85%.

4.4.2 Feature set Exploration

This data contains 704 instances, and contains the following attributes:

- **age:** *number* (Age in years).
- **gender:** *String* [Male/Female].
- **ethnicity:** *String* (List of common ethnicities in text format).
- **Born with jaundice:** *Boolean* [yes or no].
- **Family member with PDD:** *Boolean* [yes or no].
- **Who is completing the test:** *String* [Parent, self, caregiver, medical staff, clinician etc.].
- **Country of residence:** *String* (List of countries in text format).
- **Used the screening app before:** *Boolean* [yes or no] (Whether the user has used a screening app)
- **Screening Method Type:** *Integer* [0,1,2,3] (The type of screening methods chosen based on age category (0=toddler, 1=child, 2= adolescent, 3= adult).
- **Question 1-10 Answer:** *Binary* [0, 1] (The answer code of the question based on the screening method used).
- **Screening Score:** *Integer* (The final score obtained based on the scoring algorithm of the screening method used. This was computed in an automated manner).

Now we are going see the plan used to develop the project.

Table 4.3: List of Attributes in ASD dataset.

4.4.3 Road Map:

- [Step 0](#): Import Datasets.
- [Step 1](#): Clean Datasets (The data needs to be cleaned; many rows contain missing data, and there may be erroneous data identifiable as outliers).
- [Step 2](#): A quick visualization with *Seaborn*.
- [Step 3](#): At First, I applied several Supervised Machine Learning (SML) techniques on the data for classification purpose.
- [Step 4](#): Next, I experimented with different topologies, optimizers, and hyperparameters for different models.

- [Step 5](#): Model tuning.
- [Step 6](#): Feature Selection.
- [Step 7](#): Then I built a Multi-Layer Perceptron and train it to classify individual with ASD based on its features.
- [Step 8](#): Conclusion.

4.4.4 Import dataset

We will import the datasets from the UCI machine learning repository, The data set, which I will be referring to as the ASD data set from here on out, came with a .csv file that contains 704 instances that are described by 21 attributes, a mix of numerical and categorical variables. Before data can be used as input for machine learning algorithms, it must be cleaned, formatted, and maybe even restructured this is typically known as pre-processing. Unfortunately, for this dataset, there are many invalid or missing entries, we must deal with, moreover, there are some qualities about certain features that must be adjusted. This pre-processing can help tremendously with the outcome and predictive power of nearly all learning algorithms.

4.4.5 CLEANING THE DATASET (data preprocessing):

Data cleaning, also known as data cleansing or preparation, is the process of identifying and correcting or removing errors, inconsistencies, and inaccuracies in data. It is an essential step in preparing data for machine learning as it ensures that the data is of high quality and that the machine learning model can learn from it effectively. Data cleaning improves the accuracy of the machine learning model by reducing the likelihood of errors in the model's output and ensuring that the model is making accurate predictions. Best practices for data cleaning in machine learning include setting up a quality plan, filling missing values, removing rows, and reducing data size. The main steps involved in data cleaning are handling missing data, removing duplicates, and handling irrelevant data. Data preprocessing is required to

21

clean the data and make it suitable for a machine learning model, which increases the accuracy and efficiency of the model.

4.4.5 A QUICK VISUALIZATION WITH SEABORN:

Before proceeding to apply any algorithm, we take a moment to visualize the ASD data set using the Seaborn module of Python. We first generate side-by-side box plots, which present various distribution of the feature 'result' with respect to 'gender' and 'relation'. In Figure 5, the boxplots in red show the distributions of the data which belongs to the 'ASD class' whereas the blue one showing the distributions of the data which are non autistic. This gives us our first impression of the internal connections of some of the above mentioned features that are present in our dataset.

4.4.6 APPLYING ALGORITHMS:

We are gonna apply different algorithms to find out the best accuracy among those algorithm by getting accurate output. Total 7 algorithms are used here • Decision Tree

- Random forest
- Logistic Regression
- Support Vector Machines(SVM)
- K-Nearest Neighbors (KNN)
- Naïve bayes
- Linear Discriminant Analysis
- Multi-layer perceptron
- Decision Trees.

We will begin with creating a Decision Tree Classifier, also known as ID3 Algorithm and fit our training data. A Decision Tree uses a tree structure to represent a number of possible decision paths and an outcome for each path. They can also perform regression tasks and they form the fundamental components of Random Forests which will be applied to this data set in the next section.

Decision Tree models are easy to use, run quickly, are able to handle both categorical and numerical data, and graphically allow you to interpret the data. Further, we don't have to worry about whether the data is linearly separable or not. On the other hand, Decision Trees are highly prone to overfitting but one solution to that is pruning the branches so that not too many features are included and the use of ensemble methods like random forests. Another weakness of Decision Trees is they don't support online learning, so you have to rebuild your tree when new

examples come in.

A Decision Tree model is a good candidate for this problem as they are particularly adept at binary classification, however it may run into problems due to the high number of features so care will have to be taken with regards to feature selection. Due to these advantages and the ease of interpretation of the results, we will use the Decision Tree Classifier as the benchmark model.

Random Forests

One way of avoiding overfitting that Decision Trees are prone to, is to apply a technique called Random Forests, in which we build multiple decision trees and let them vote on how to classify inputs. Random forests (or random decision forests) are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. A depiction of the algorithm is presented in Figure 4.7.

Figure 4.7: Random Forest Diagram

Support Vector Machines (SVM):

Figure 4.8: support vector machine

Next, we move on to the Support Vector Machine algorithm (SVM) which is, by far, my favorite machine learning algorithm. Support Vector Machine is a supervised machine learning algorithm that is commonly used in classification problems. It is based on the idea of finding the hyperplane that 'best' splits a given data set into two classes. The algorithm gets its name from the support vectors (the data points closest to the hyperplane), which are points of a data set that if removed would alter the position of the separating hyperplane. (See Figure 4.8)

The distance between the hyperplane and the nearest training data point from either set is known as the margin. Mathematically, the SVM algorithm is designed to find the hyperplane that provides the largest minimum distance to the training instances. In

other words, the optimal separating hyperplane maximizes the margin of the training data.

k-Nearest Neighbors (KNN):

The k Nearest Neighbor (KNN) algorithm is based on mainly two ideas: the notion of a distance metric and that points that are close to one another are similar. Let x be the new data point that we wish to predict a label for. The k Nearest Neighbor algorithm works by finding the k training data points x_1, x_2, \dots, x_k closest to x using a Euclidean distance metric. KNN algorithm then performs majority voting to determine the label for the new data point x . In the case of binary classification, it is customary to choose k as odd. In the situation where we encounter a tie as a result of majority voting there are couple things we can do. First of all we could randomly choose the winner among the labels 15 that are tied. Secondly, we could weigh the votes by distance and choose the weighted winner and last but not least, we could lower the value of k until we find a unique winner.

Naive Bayes:

We proceed the study of supervised machine learning algorithms by applying Naive Bayes (NB), which is based around conditional probability (Bayes theorem) and counting. The name "naive" comes from its core assumption of conditional independence i.e. all input features are independent from one and another. If the NB conditional independence assumption actually holds, a NB classifier will converge quicker than discriminative models like logistic regression, so one needs less training data. And even if the NB assumption doesn't hold, a NB classifier still often does a great job in practice. It's main disadvantage is that it can't learn interactions between features. It only works well with limited number of features. In addition, there is a high bias when there is a small amount of data.

Logistic Regression:

The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory)

variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest.

Figure 4.9: Logistic Regression Model

Logistic regression can be called binary classification problems. A key point to note here is that Y can have 2 classes only and not more than that. If response variable has more than 2 classes, it would become a multi class classification and you can no longer use the vanilla logistic regression for that. Yet, Logistic regression is a classic predictive modelling technique and still remains a popular choice for modelling binary categorical variables.

Another advantage of logistic regression is that it computes a prediction probability score of an event. More on that when you actually start building the models. Logistic regression achieves this by taking the log odds of the event $\ln(P)$, where P is the probability of event. So P always lies between 0 and 1.

Linear Discriminant Analysis (LDA):

A classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes theorem. The model fits a Gaussian density to each class, if all classes share the same covariance matrix. The fitted model can also be used to reduce the dimensionality of the input by projecting it to the most discriminative directions.

Multi-Layer Perception (MLP):

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. Multilayer perceptron's are sometimes colloquially referred to as 'vanilla' neural networks, especially when they have a single hidden.

Figure 4.10: MLP Architecture

4.4.7 Model tuning:

Now I will fine tune the chosen model. For this I use grid search (GridSearchCV) with at least one important parameter tuned with at least 3 different values. I will need to use the entire training set for this. In the code cell below, I will need to implement the following:

27

- Import `sklearn.grid_search.GridSearchCV` and `sklearn.metrics.make_scorer`.

Initialize the classifier you've chosen and store it in `clf`.

- Set a `random_state` if one is available to the same state you set before.

Create a dictionary of parameters you wish to tune for the chosen model.

Example: `parameters = {'parameter' : [list of values]}`.

- **Note:** Avoid tuning the `max_features` parameter of your learner if that parameter is available!

- Use `make_scorer` to create an `fbeta_score` scoring object (with $\beta=0.5$ \diamond $=0.5$).

Perform grid search on the classifier `clf` using the 'scorer', and store it in `grid_obj`.

- Fit the grid search object to the training data (`X_train`, `y_train`), and store it in `grid_fit`.

Note that, `svm.SVC` may perform differently with different kernels. The choice of kernel is an example of a "hyper parameter." Here I experimented with different kernels such as 'rbf', 'sigmoid', and 'poly' and found that the best-performing kernel is linear.

4.4.8 Feature selection:

Choose a scikit-learn supervised learning algorithm that has a feature importance attribute available for it. This attribute is a function that ranks the importance of each feature when making predictions based on the chosen algorithm.

In the code cell below, I will implement the following:

- Import a supervised learning model from sklearn if it is different from the three used earlier.
- Train the supervised model on the entire training set.
- Extract the feature importances using `'.feature_importances_'`.

We need to ask ourselves how does a model perform if we only use a subset of all the available features in the data? With less features required to train, the expectation is that training and prediction time is much lower — at the cost of performance metrics. From the visualization above, we see that the top five most important features(in order with their weightage factor) contribute more than half of the importance of **all** features present in the data. These 5 features are:

- 'result'
- 'relation_self'
- 'country_of_residence'
- 'jundice_no'
- 'jundice_yes'

This hints that we can attempt to *reduce the feature space* and simplify the information required for the model to learn. Although looking at those weight factor it seems like 'result' feature is clearly dominating its influence on the algorithms over all other features

4.5 TRANSITION/SOFTWARE TO OPERATION PLAN

- For executing the data, we are gonna use jupyter notebook.
- Data preprocessing and feature selection are done by using the applications in python like pandas.

- Pursuing such research necessitates working with datasets that record information related to behavioral traits and other factors such as gender, age, ethnicity, etc.
- Such datasets are rare, making it difficult to perform thorough analyses to improve the efficiency, sensitivity, specificity and predictive accuracy of the ASD screening process.
- At present, very limited autism datasets associated with clinical or screening are available and most of them are genetic in nature.
- Central Processing Unit (CPU) Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal. RAM 8 GB minimum, 16 GB or higher is recommended. Graphics Processing Unit (GPU) NVIDIA GeForce GTX 960 or higher.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

- Assuming that the available data for analytics fairly represents the real world process that we wish to model and the process is expected to remain relatively stable, then the data we currently have should be a reasonable representation of the data we expect to see in the future.
- As a result, withholding some of the current data for testing is a fair and justified way to provide an honest assessment of our model. Thus, we split the given data into two parts. 80% of the data will be used to train the model and this data will be referred to as the training data set and 20% of the data will be reserved for testing the accuracy and effectiveness of the model on data that the model has never seen before and will be referred to as the testing data set.

- Thus, our training set has 487 samples, and the testing set has 122 samples. Below is a flowchart that represents the whittling process of the raw data: The random partitioning of data into testing and training data also helps us determine whether our model is underfitting (too simple, high bias, low variance) or overfitting (too complicated, high variance, low bias). A model that has high training and testing error is a model that underfits.
- This means our model is too simplistic. A model that has low training error, but high testing error is one that overfits. This means that our model is memorizing the data rather than trying to understand the intrinsic trends or patterns in the data.
- In other words, it's important to test our model and see how it generalizes to unseen data by applying the model with testing data which was not a part of the model creation.
-

30

5.2 ALGORITHMS

5.2.1 *Decision Trees:*

ID3 Algorithm.

The confusion matrix in this case takes the form:

Table 5.1: Confusion Matrix for ASD Data Set

The accuracy of the Decision Tree classifier which measures overall how often this classifier is correct is:

and the precision of the Decision Tree Classifier which measures how often the

prediction is correct when a positive outcome is predicted is:

The sensitivity of the method measures how often the prediction is correct given that the actual outcome is positive:

and the specificity measures how often the prediction is correct given that the actual outcome is negative:

31

Finally, we calculate the false positive rate as

which measures how often the prediction is correct given that the outcome was actually negative. Note the values of the following evaluation metrics. • cross-validation score=1.0

- AUC Score=1.0
- F-Beta Score=1.0 with Beta=0.5.

Note that the Decision Tree Classifier will serve as our Benchmark Model. This means that all future models that we study will be compared to the Decision Tree classifier and may have an accuracy only as good as but not better than the Decision Tree classifier. In other words, no future model can surpass the Decision Tree classifier in terms of performance statistics.

Figure 5.1: A gini diagram of Decision Tree.

5.2.2 Random Forest.

- cross-validation score=0.9933
- AUC Score=0.9988
- F-Beta Score=1.0.

32

5.2.3 Support Vector Machine (SVM).

We start the SVM algorithm with a linear kernel and gamma=2 and find:

- cross-validation score=1.0
- AUC Score=1.0
- F-Beta Score=1.0

5.2.4 K-Nearest Neighbors (KNN).

We apply K-nearest neighbor algorithm with an initial value of K = 10 and observe:

- cross validation score=0.94745
- AUC Score=0.9930
- F-Beta Score=0.9148

5.2.5 Naive Bayes.

On applying NB to our data set we find:

- cross-validation score= 0.885
- AUC Score=0.9445 and

- F-Beta Score=0.8370.

5.2.6 Logistic Regression.

We note the following values for evaluation metrics:

- cross validation score=0.9704,
- AUC Score=0.9974, and
- F-Beta Score=0.9307.

5.2.7 Linear Discriminant Analysis.

- cross validation score= 0.9326
- AUC Score=0.9850
- F-Beta Score=0.9148

33

5.2.8 Multi-Layer Perception (MLP).

- training accuracy =0.9979
- testing accuracy=0.9836

5.2.9 SVM:

- I found that a non-linear kernel did not yield a good cross-validation score.

5.2.10 KNN:

- Choosing K is always tricky, and we decided to run KNN algorithm using different values of K running through a loop with values ranging from K = 11 to K = 99. The results show that there was no major improvement in the cross-validation scores for different values of K. In fact, the best we achieve is K = 94 with a cross validation score of 0.9606.

5.3 TESTING

5.3.1 Visualization with seaborn

- Before proceeding to apply any algorithm, we take a moment to visualize the ASD data set using the Seaborn module of Python.

- We first generate side-by-side box plots, which present various distribution of the feature 'result' with respect to 'gender' and 'relation'.
- In Figure (5), the boxplots in red show the distributions of the data which belongs to the 'ASD class whereas the blue one showing the distributions of the data which are non-augitic.
- This gives us our first impression of the internal connections of some of the above-mentioned features that are present in our dataset.

Figure 5.2: Factor plot: kind = 'boxplot

Next we present Figure 5.2, a special form of Factor plot where we display the connection between several attributes from our data set and how they are related with our target class. In this case, we can see when 'jaundice' is present at birth, an individual with a higher 'result' score will have autism irrespective of their gender.

Figure 5.3: Factor plot: kind = 'swamp'

35

We move on to a series of violin plots to compare how different features contribute to the likelihood of autism.

Figure 5.4a and Figure 5.4b show a similar relationship between 'result' and 'gender' versus 'result' and 'jaundice'. In both cases, an individual with a higher 'result' score is more likely to have autism, independent of the other feature.

The variables 'jaundice' or 'gender' do not seem to have a lot of influence in deciding the ASD class. Lastly we present another variation of a violin plot in Figure 5.5, where we look at how the distribution of autism varies by relation (Self, Parent, ...), subdivided by whether the patient was born with jaundice ('jaundice') and the patient's gender.

Figure 5.4: jaundice

- (a) A violine plot showing how the ASD classes is related with the attributes 'result' & 'jundice'.
- (b) A violine plot showing how the ASD classes is related with the attributes 'result' & 'gender'.

This diction exercise is a great way to understand the data anatomy before we decide to apply algorithms which will be most suitable for our goal.

5.3.2 IMPLEMENTATION

Assuming that the available data for analytics fairly represents the real world process that we wish to model and the process is expected to remain relatively stable, then the data we currently have should be a reasonable representation of the data we expect to see in the future.

As a result, withholding some of the current data for testing is a fair and justified way to provide an honest assessment of our model. Thus, we split the given data into two parts.

37

80% of the data will be used to train the model and this data will be referred to as the training data set and 20% of the data will be reserved for testing the accuracy and effectiveness of the model on data that the model has never seen before and will be referred to as the testing data set.

Thus, our training set has 487 samples, and the testing set has 122 samples. Below is a flowchart that represents the whittling process of the raw data:

The random partitioning of data into testing and training data also helps us determine whether our model is underfitting (too simple, high bias, low variance) or overfitting (too complicated, high variance, low bias).

A model that has high training and testing error is a model that underfits. This means our model is too simplistic. A model that has low training error, but high testing error is one that overfits.

This means that our model is memorizing the data rather than trying to understand the intrinsic trends or patterns in the data.

In other words, it's important to test our model and see how it generalizes to unseen data by applying the model with testing data which was not a part of the model creation.

Since we pre processed the ASD dataset, we didn't run into any problems or difficulties. To implement each of the mentioned methods, we imported and used the following Python modules from Scikit Learn.

- `from sklearn.tree import DecisionTreeClassifier`
- `from sklearn.ensemble import RandomForestClassifier`
- `from sklearn import svm`
- `from sklearn import neighbors`
- `from sklearn.naive_bayes import MultinomialNB`
- `from sklearn.linear_model import LogisticRegression`
- `from sklearn.discriminant_analysis import LinearDiscriminantAnalysis`

38

- `from keras.models import Sequential`
`from keras.layers import Dense, Dropout,`
`Activation`

As we can see in the Figure 5.6, there is a clear class imbalance in the target class of individual with ASD. In this situation, we decided to apply AUC-score as well as F1-score in addition to k-Cross validation accuracy as accuracy may not be a proper metric to use in cases like ours where we can observed a definitive class imbalance. To calculate the desired metrics I imported and used the following Python modules: `from sklearn.cross_validation import cross_val_score`

```
cross_val_score(*, features_final, asd_classes, cv=10, scoring='roc_auc').mean()
```

- `from sklearn.metrics import fbeta_score`
- `from sklearn.model_selection import cross_val_score`
`clf =`

```
Classifier(random_state=1)
```

```
cv_scores = cross_val_score(clf, features_final, asd_classes,  
cv=10)cv_scores.mean()
```

- `model = Sequential()`

```
model.add(Dense(8, activation='relu', input_dim= 94))
```

```
model.add(Dropout(0.2)) model.add(Dense(1, kernel_initializer='normal',  
activation='sigmoid')) model.summary()
```

```
score = model.evaluate(X_test, y_test, verbose=0)
```

Figure 5.6: target class of individual with ASD

39

5.3.3 Feature Importance:

An important task when performing supervised learning on a dataset like the autistic data we study here is determining which features provide the most predictive power. By focusing on the relationship between only a few crucial features and the target label we simplify our understanding of the phenomenon, which is almost always a useful thing to do. In case of this project, that means we wish to identify a small number of features that most strongly predict whether an adult individual has ASD or not.

Choose a scikit-learn classifier (such as, gradient Boosting, adaboost, random forests) that has a `feature_importances_` attribute, which is a function that ranks the importance of features according to the chosen classifier.

In Figure 5.7 shows the top 5 most important features for the ASD dataset using two different classifiers which has that feature importance attribute. We need to ask ourselves how does a model perform if we only use a subset of all.

Figure 5.7: normalized weights

40

Figure 5.7: Feature importances with two Classifiers the available features in the data? With less features required to train, the expectation is ,the training and prediction time will be lowered at the cost of performance metrics. From the visualization presented in Figur5.7, we see that the top most important feature 'result'(in term of their weightage factor) contribute heavily compared to the other features.

Here we list the first 5 most important features according to their weightage

order: 1. 'result'

2. 'relation self'

3. 'country of residence'

4. 'jundice-no'

5. 'jundice-yes'

This hints that we can attempt to reduce the feature space and simplify the information required for the model to learn. By doing so we can train the model with even with big dataset with reduced time.

Table 5.2: Comparison of metrics using different learning algorithm

41

CHAPTER 6

RESULTS AND DISCUSSION

In this section, we present the numerical results from different machine learning algorithms applied to our ASD data set. In all these numerical computations, I used SciKit Learn module which is written in Python.

As important as it is to find the right model it is equally important to establish which models may not be the best choice. Evaluation metrics such as AUC Score(AUC), F Beta-score with $\beta=0.5$ (F-Beta), Cross-validation score(CVS) for each method are summarized.

Machine initially seeks to discover how machine learning and mathematics can reveal the visibility and measurability of the genetic makeup of ASD. Hypothetically, those disciplines would help make predictions of novel biomarkers in an ASD patient since they are useful for analysing vast amounts of available data. The distance formula model demonstrates that when the genotype counts for ASD patients are much higher than those for unaffected relatives, the distance suggests that one or more genes are likely to be linked to autism.

6.1 Model Evaluation and Validation

After exploring the ASD dataset with different learning algorithms, we have arrived at the conclusion that all of our model work extremely well with the data. we have used

three different metrics (accuracy, AUC score and F-score) to measure the performance of the models, and it seems like all of the metrics indicated an almost perfect classification of the ASD cases.

Recall that we had decided to use the Decision Tree Classifier as the Benchmark Model for this problem. Looking over the Table 5.1, it is abundantly clear that the

42

SVM algorithm with a linear kernel does the best job in classifying new instances into one of the two categories: “patient has ASD” and “patient does not have ASD”.

Hence, we should consider SVM as our final model, with the Logistic Regression and Random Forest being a close second.

On the other hand, the Naive Bayes classifier would not be the method of choice for prediction of new instances given the previously discussed disadvantages of the method and the metric performances that can be observed from Table 4.

Here it is reasonable to infer that all the model performances are so high because of the fact that only one feature(‘result’) is predominant over all others features which we have discussed in the Feature Importance section.

6.2 Justification

Our SVM model achieves the same results as the benchmark model. However, the reader should also note that using the Decision Tree Classifier as the benchmark model which has an accuracy of 1 implies that no other model can supersede the Decision Tree Classifier, but can only achieve the same accuracy.

43

CHAPTER 7

CONCLUSION

7.1 Free-Form Visualization

As in “Feature Selection” exploration, we have seen the attribute named ‘result’ has such a powerful presence in the ASD dataset, all other attributes have little to no contribution in deciding the final class. In the Figure 18 below, we have drawn a swarm plot with ‘result’ as xaxis and ASD-class (say ‘yes’ is 1 and ‘no’ is 0) as y-axis, and reconfirmed the underlying association between the given variables where the target class is easily separable.

Figure 7.1: Free-Form Visualization

7.1.1 Reflection

- During pre-processing of the data, we dropped about 95 of rows of data due to its ‘NaN’ entries. Ideally one should try to retain as much data as possible, as there could always be some valuable insight that could be lost.
- we usually try to fill the missing entries by ‘median filling’ or with a fancier approach, one can also run a supervised learning model to predict the NaN value. But that approach is not applicable for our dataset as many inputs that

44

was missing are of categorical type and it is not feasible to make a median of a category).

- While there are some advanced procedures like ‘imputation process’ to take care of these issues of missing values of categorical types without having them dropped, we didn’t explore those techniques in this current work.
- Even without implementing more sophisticated techniques like ‘imputation’ in order to retain valuable information, we managed to find very efficient models that can classify new instances with accuracy score 1.
- The reason scores close to 1.0 is because a single feature is almost entirely responsible for deciding the output. This means that the output can be determined by a high confidence.
- The present work is certainly quite interesting and is a noble area where technology can be used to actually change lives.

7.1.2 Improvement

- Thus, to summarize, we set out with the hopes of applying machine learning algorithms, specifically, supervised machine learning techniques that can classify new patients (new instances) with certain measurable characteristics (the variables) into one of two categories “patient has ASD” or “patient does not have ASD”.
- Cleaning the data set (which documented the characteristics associated with ASD), was challenging in that we had mostly categorical variables and just two numerical variables, but ultimately we were able to build such models and found that the algorithm that performs the best in all aspects is the SVM machine learning algorithm, using a ‘linear kernel’.
- The SVM outperformed all other models with respectt to crossvalidation score, AUC Score, and F-Beta Score, all of score were 1, and thus was as good as our benchmark model.

- Although the data association made the prediction very simple, but we feel this work can certainly serve as an invaluable aid for physicians for detection of new autistic cases.

- In our consideration, to build an accurate and robust model, one needs to have larger datasets. Here the number of instances after cleaning the data were not sufficient enough to claim that this model is optimum.
- Looking at the performances of our learning models, nothing can be improved with this current data set as models are already at their best.
- After discussing this issue with a researcher directly working on adult autism, we have realised that it is extremely difficult to collect a lot of well documented data related to ASD.
- This ASD dataset has recently been made public (available from December 2017), and thus not much work has been done.
- With this in consideration, our research has resulted in well developed models that can accurately detect ASD in individuals with given attributes regarding the persons behavioral and medical information.
- These models can serve as benchmarks for any machine learning researcher/practitioner who is interested in exploring this dataset further or other data sets related to Autism screening disorder

7.2 FUTURE WORK

This project will extend to the top 100-300 SNP ranking differences that implicate ASD-linked genes as well as high-ranking SNPs in balanced populations of affected and unaffected individuals.

In addition, it will establish connections to relevant research conducted in the field of

mathematical and computational neuroscience.

Ideally, the genetic makeup of ASD would be further analysed if a full understanding of mirror neurons genetics is achieved and if such neural simulators as Neng/Neural Engineering Object (a Python-based software package for simulating large-scale neuronal.

REFERENCES

[1]. A. Geron, Hands-On Machine Learning with Scikit-Learn & Tensor Flow, O'Reilly
ISBN: 9781491962299

[2] Ching, M. S. L., et al. (2010).Autism prevalence slightly higher in CDC' s ADDM Network | CDC Online Newsroom | CDC. (2018, June 29). Retrieved July 31, 2018, from <https://www.cdc.gov/media/releases/2018/p0426-autismprevalence.html>

[3] Brian Godsey, Think Like a Data Scientist Manning, ISBN: 9781633430273

[4] D. Cielen, A. Meysman, M. Ali, Introducing Data Science, Manning, ISBN: 9781633430037.

[5]. G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer, ISBN 9781461471370

[6] H. Brink, J. Richards, M. Fetherolf, Real World Machine Learning, Manning, ISBN :9781617291920

[7] J. Grus, Data Science from Scratch First Principles With Python, O'Reilly, ISBN: 9781491901427

[8] Kou, Y., Betancur, C., Xu, H., Buxbaum, J. D., & Ma 'Ayan, A. (2012). Network and Attribute-Based Classifiers Can Prioritize Genes and Pathways for Autism Spectrum Disorders and for Intellectual Disability. American Journal of Medical Genetics. Part C, Seminars in Medical Genetics, 160C (2), 130-142

[9]. T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Second Edition, Springer

[10]. Tabtah, F. (2017). Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment. Proceedings of the 1st International Conference on Medical and Health Informatics 2017, pp.1-6. Taichung City, Taiwan, ACM.

48

[11]. Thabtah, F. (2017). ASD Tests. A mobile app for ASD screening. www.asdtests.com [accessed December 20th, 2017].

[12]. Thabtah, F. (2017). Machine Learning in Autistic Spectrum Disorder Behavioral Research: A Review. Informatics for Health and Social Care Journal. December 2017 (in press)

49

APPENDIX

SOURCE CODE

```
[{"metadata": {}, "cell_type": "markdown", "source": "\n\n## *Machine learning approaches to the classification problem for autism spectrum disorder*\n\n"}]
```

```
import numpy as np
import pandas as pd
from time import time
from IPython.display import display # Allows the use of display() for
```

```
DataFrames import visuals as vs
```

```
%matplotlib inline
data = pd.read_csv("\Users\kalya\Downloads\Nikhil-c3.zip\Nikhil
- c3\autism_screening.csv')
display(data.head(n=5))
```

```
n_records = len(data.index)
```

```
n_asd_yes = len(data[data['Class/ASD'] == 'YES'])
```

```
n_asd_no = len(data[data['Class/ASD'] == 'NO'])
```

```
yes_percent = float(n_asd_yes) / n_records * 100
```

```
print "Total number of records: {}".format(n_records)
print "Individuals diagnosed with ASD: {}".format(n_asd_yes)
print "Individuals not diagnosed with ASD: {}".format(n_asd_no) print "Percentage
of individuals diagnosed with ASD: {:.2f}%".format(yes_percent)
```

```
asd_data.loc[(asd_data['age'].isnull()) |(asd_data['gender'].isnull())
|(asd_data['ethnicity'].isnull())
```

```
|(asd_data['jundice'].isnull())|(asd_data['austim'].isnull())
```

```
|(asd_data['contry_of_res'].isnull())
```

```
|(asd_data['used_app_before'].isnull())|(asd_data['result'].isnull())|(asd_data['age_de  
s c'].isnull())
```

```
|(asd_data['relation'].isnull())]
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
sns.set(style="whitegrid", color_codes=True)
```

```
sns.violinplot(x="jundice", y="result", hue="austim", data=asd_data,  
split=True, inner="quart", palette={'yes': "r", 'no': "b"})
```

```
sns.despine(left=True)
```

```
sns.violinplot(x="jundice", y="result", hue="Class/ASD", data=asd_data,  
split=True, inner="quart", palette={'YES': "r", 'NO': "b"})
```

```
sns.despine(left=True)
```

```
sns.violinplot(x="gender", y="result", hue="Class/ASD", data=asd_data,  
split=True, inner="quart", palette={'YES': "r", 'NO': "b"})
```

```
sns.despine(left=True)
```

```
sns.factorplot(x="jundice", y="result", hue="Class/ASD", col="gender",  
data=asd_data, kind="swarm");
```

```
sns.factorplot(x="gender", y="result", hue="Class/ASD",  
col="relation", data=asd_data, kind="box", size=4, aspect=.5,  
palette={'YES': "r", 'NO': "b"});
```

```
g = sns.factorplot(x="result", y="jundice",  
hue="gender", row="relation",  
data=asd_data,  
orient="h", size=2, aspect=3.5, palette={'f': "r", 'm': "b"},  
kind="violin", dodge=True, cut=0, bw=.2)
```

```
plt.hist(asd_classes, bins=10)
```

```
plt.xlim(0,1)
```



```

plt.title('Histogram of Class/ASD')
plt.xlabel('Class/ASD from processed data')
plt.ylabel('Frequency')

from sklearn.model_selection import train_test_split

np.random.seed(1234)

X_train, X_test, y_train, y_test = train_test_split(features_final, asd_classes,
train_size=0.80, random_state=1)
print "Training set has {} samples.".format(X_train.shape[0])
print "Testing set has {} samples.".format(X_test.shape[0])
from sklearn.model_selection import train_test_split

np.random.seed(1234)

X_train, X_test, y_train, y_test = train_test_split(features_final, asd_classes,
train_size=0.80, random_state=1)

print "Training set has {} samples.".format(X_train.shape[0])
print "Testing set has {} samples.".format(X_test.shape[0])

from sklearn import neighbors

knn = neighbors.KNeighborsClassifier(n_neighbors=10)
cv_scores = cross_val_score(knn, features_final, asd_classes, cv=10)

cv_scores.mean()
from sklearn.cross_validation import cross_val_score
cross_val_score(knn, features_final, asd_classes, cv=10,

scoring='roc_auc').mean() from sklearn.ensemble import AdaBoostClassifier

52
model = AdaBoostClassifier(random_state=0)
model.fit(X_train, y_train)

```

```
importances = model.feature_importances_  
vs.feature_plot(importances, X_train, y_train)  
  
sns.stripplot(x="result", y="Class/ASD", data = asd_data,  
jitter=True); import pydotplus  
dot_data = tree.export_graphviz(dectree,  
out_file=None,  
filled=True,  
rounded=True,  
special_characters=True)  
graph = pydotplus.graph_from_dot_data(dot_data)  
from IPython.display import Image  
Image(graph.create_png())
```


AUTISM SPECTRUM DISORDER DETECTION USING MACHINE LEARNING TECHNIQUES.

Sagam Nikhil Reddy,
Student, Computer Science and
Engineering Department,
Sathyabama Institute of Science and
Technology,
Rajiv Gandhi Salai, Chennai –

600119 nikhilreddy0118@gmail.com
Dr. R. Aroul Canessane, ME., Ph.D.,
Assistant Professor, Computer Science
and Engineering Department,
Sathyabama Institute of Science and
Technology,
Rajiv Gandhi Salai, Chennai –
600119 aroul.cse@sathyabama.ac.in
Kalyanapu sridhar

Student, Computer Science and
Engineering Department,
Sathyabama Institute of Science and
Technology,
Rajiv Gandhi Salai, Chennai –
600119
Kalyanapusridhar247@gmail.com

ABSTRACT

Autism Spectrum Disorder (ASD) is a neurodevelopmental ailment related to large healthcare expenses, and early analysis can significantly lessen expenses. Unfortunately, inside the outdoor, the time and fee of the product isn't always well worth it. With the financial impact of autism and the prevalence of ASD international growing daily, there's an pressing need to expand strategies that are easy to put into effect and powerful for screening. Thus, speedy and low cost screening for ASD is necessary to help health experts and tell humans whether or not they need to have a proper clinical diagnosis. The speedy increase inside the quantity of ASD instances global has brought about demands for applicable behavioral statistics. However, such records are rare, making it hard to carry out a detailed analysis to improve the performance, sensitivity, specificity, and accuracy of the predictive procedure of ASD screening. There are presently very confined facts available on autism for clinical trials or screening, and maximum of them are genetic in nature. We consequently advise a new dataset relevant to adult screening for autism, which includes 20 characteristics that can be used for similarly analysis, specially for figuring out influential autistic traits and improving the classification of ASD instances. In this dataset, we document ten behavioral traits (AQ-10-Adult) plus ten man or woman traits that have been

shown to be powerful in evaluating cases of ASD among technological know-how controls.

Keywords – Autism, Machine learning, CNN.

INTRODUCTION

Autism Spectrum Disorder (ASD) is the name of a collection of developmental problems that have an effect on the anxious device. Symptoms of ASD range from mild to excessive and most generally consist of speech issues, troubles with communication, and repetitive behaviors. Many different signs and symptoms may additionally encompass anxiety, mood problems, and attention deficit/hyperactivity disorder (ADHD).

ASD has a great economic effect on the healthcare industry, both due to the boom in ASD instances and the time and price associated with diagnosing a patient. Early detection of ASD can help both sufferers and the health care sector via prescribing cures and/or medicines that

sufferers need, thereby reducing the lengthy-time period fees associated with delayed prognosis. Thus, health specialists around the world are in need of simple, price powerful, dependable and lower priced ASD screening techniques that can as it should be are expecting if a patient has sure measured characteristics of ASD and inform human beings whether or not they need to have a formal

clinical analysis.

But problems stay. Conducting such behavioral research requires running with datasets that capture information on behavioral styles and different factors such as gender, age, ethnicity, and many others. Such information are rare, making it difficult to conduct analyzes to enhance the efficiency, sensitivity, specificity, and accuracy of predictions. Screening method for ASD. Very confined scientific studies or statistics associated with autism screening is to be had, and most of it's miles genetic in nature. This information is extremely sensitive and hard to acquire on social and personal subjects.

Single nucleotide polymorphisms (SNPs) are genetic sequence versions that arise at a function in which one DNA nucleotide is exchanged for another. SNPs in genes related to autism spectrum disease (ASD) may be beneficial biomarkers for the prognosis of ASD and assist to identify the genetic causes of ASD.

The purpose of this task is to perceive SNPs associated with ASD based on genomic DNA genotyping of SNPs in a huge cohort of ASD patients and healthy relatives. The

dataset from the Omnibus Gene Expression database (GSE6754) carries over 6000 samples from 1400 families. SNPs are calculated inside the three-dimensional genotype area, the space among all affected and unaffected topics in the cohort. The consequences suggest that SNPs with a excessive diploma of distance might be associated with ASD. High-ranking SNPs that presently don't have any known affiliation with ASD have the ability to end up new biomarkers. The involvement of genes containing these SNPs in biological pathways that may be applicable to ASD is discovered the usage of the pathway database.

LITERATURE SURVEY

In 2018, the CDC's Autism and Developmental Disabilities Monitoring Network file that about 1 in 59 kids inside the United States, and the range keeps to upward push, are diagnosed with ASD. As machine studying plays a role in reworking many sectors of the

58

enterprise, most executives trust that artificial intelligence will maximize productiveness inside the exertions financial system.

Similarly, information technological know-how can clear up actual issues in the natural sciences, global improvement, the arts, and lots of other disciplines that deal with big quantities of facts. Because medical studies on autism has been taking place for years, machine getting to know.

Machine mastering is emerging from emerging technologies that rely upon repeated calculations to make selections to adapt to new data. Companies that generate large quantities of data are commonly interested in records mining, set of rules development, and inexpensive records garage and processing. When device studying is used in bioinformatics, scientists are maximum likely to use neural networks, genetic algorithms, and fuzzy good judgment. One of the maximum vital troubles that this undertaking enables to resolve is the classification of genes which might be stricken by a disease or disorder and which can be prominent from ordinary genes.

Researchers cannot inform actually how autism is established, however linkage studies and variety versions (CNV) in more than a thousand households, with at the least two affected individuals can provide an explanation for the possibility that chromosome 11p12-p13 is an autism chance locus in linkage and evaluation of neurexin in CNV. . Clearly, linkage screening may be a tool for diagnosing the origin of autism.

IN P.P. In the research paper, Sans also proposed the combination of a deep neural network with a help vector gadget (SVM) classifier within the ultimate step. Another earlier paintings is primarily based on an intra-problem approach. Some move-concern techniques have also been proposed by way of gathering EEG samples from all subjects and randomly setting apart them into training and testing, as H. Zeng et al. This method is, of course, random, and therefore effects in a mixture of a few training models with experiments, which isn't to be skipped.

In a studies paper by using Y. Liuet, all authors do area version, a department of studying transfer, to tailor the source and target data distributions for a extra green distribution in a cross-concern scenario.

Dr. Yusuf Hossain et al [3] proposed a non-intrusive device the usage of the eye closure device as an enter parameter. In the paper through Y. Liuet, all EEG functions, statistics, higher order intersection, fractured size, eigenvalue and spectral power were extracted and blended with several classifiers which include logistic regression, 1-nearest linear discriminant analysis, linear SVM. And rural Bayes.

Mika Sunagawa et al [4] proposed a model that correctly captures the entire hole range, from mild to excessive.

Monagi H. Alkinani et al. [5] He posted an intensive evaluation of comparisons of numerous Internet-based research to identify driver distraction, drowsiness, fatigue, and alertness. With João Ruivo Paulo et al [7] he investigated the distracted remark method using EEG alerts inside the remedy enterprise and the consequences showed a extra uniform accuracy of seventy five.87% and above for move-validation with one exception. Wang et al., N. Hatami et al., Z. Zhao and all methodologies (recurring plots and Gramian perspective fields), have been efficaciously utilized in pc vision algorithms in combination with deep learning, had been used greater recently. Work within the area of EEG research, but nonetheless noticeably unexplored.

OPEN PROBLEMS IN EXISTING SYSTEM

I changed into able to discover the open supply statistics to be had in the UCI system studying repository. The records changed into these days posted on December 24, 2017. The dataset, which I will talk to because the ASD dataset, comes with a csv report that includes 704 times, 21 descriptive attributes, a mixture of numerical and express variables. A short description of the asd facts set may be observed in this web page. This dataset become designed through Prof. Fadi Fayez Tabtah Department of Digital Technology at the Massachusetts Institute of Technology, Auckland, New Zealand. With data to be had on humans with ASD, the intention is to predict new patients and classify them into one in all styles of "sufferers". Has ASD" or "the patient does now not have ASD". In other phrases, we are working on a binary classification trouble with the remaining goal of being able to insert new cases, i.E. While we've got a new person patient with some traits, we would really like to predict if he has a high possibility of getting ASD.

SOFTWARE-REQUIREMENTS SPECIFICATION DOCUMENT

We will use the term machine gaining knowledge of to refer to the creation and use of models which are educated from statistics, i.E., it's miles a categorised dataset with the suitable response from which the version can analyze. I will even upload a diffusion set of rules characteristic to locate which of the 20 variables are maximum essential in determining whether or not someone has ASD or now not. This paintings targets to explore several aspects of classification opposition for machine getting to know supervisors

Namely;

- Decision Trees

- Random woods
- Support Vector Machine (SVM)
- k-nearest (KNN)
- SIMPLE Bayes
- Logistic regression
- Linear Discriminant Analysis (LDA)
- Multilayer perceptron (MLP)

Machine studying is emerging from emerging technologies that depend on repeated calculations to make decisions to evolve to new records. Companies that generate big amounts of records are commonly interested in information mining, set of rules improvement, and cheaper statistics garage and processing. When machine mastering is used in bioinformatics, scientists are most probable to use

fashions. Next, we method the information by using pre processing it with appropriate algorithms. Using classification algorithms consisting of decision tree, random wooded area, guide vector device (SVM), okay nearest (KNN), simplex, logistic regression, linear discriminant evaluation (LDA), multilevel perceptron (MLP).

Using class algorithms, this dataset is split into datasets (1) human beings with ASD (2) people with out ASD. And after identifying people with ASD in the statistics set, we will evaluate the accuracy of the applied type algorithms. Due to its type, logistic regression has high accuracy compared to other algorithms. After evaluating the accuracy, if the accuracy of the test exceeds the accuracy of the schooling, if so, the process could be repeated from the dataset category with the algorithms, if now not, we can go to the visualization component. There isn't any need to visualize this dataset in plot models. Once the version is skilled, it'll have statistics approximately humans with ASD primarily based at the attributes of the given dataset.

SELECTED METHODOLOGY OR PROCESS MODEL

Data Preprocessing

Unfortunately, there are numerous invalid or missing records in this dataset, which can be represented by using query records. So we want to pre-method our statistics so that it is ready to be utilized by machine mastering algorithms. Here we begin with the aid of changing entries with "?" and convert to NAN (not a number). We conclude that there are quite some lacking values inside the statistics set, however the missing values are random (proved with Python code, see Figure thirteen). In other words, it seems suitable to really eliminate all traces containing lacking records and no longer threat corrupting our facts.

A
view at the locations of the missing values in ASD data.

Implementation

Assuming that the facts to be had for analytics represents the very procedure we need to version, and the procedure is predicted to remain solid, then the information we've now have to be an inexpensive illustration of the data we count on inside the destiny. . For this reason, as a test of a

neural networks, genetic algorithms, and fuzzy good judgment. One of the maximum important problems that this task contributes to is the type of genes which can be stricken by a disorder or disorder and which may be distinguished from ordinary genes.

DESCRIPTION OF PROPOSED SYSTEM

Taking the dataset, we go to previous datasets to locate lacking values, cast off noise and normalize to keep away from raw records and numerous dataset troubles with the aid of developing new values and retaining a higher distribution in addition to a ratio inside the data. We introduce statistics transformation feature improvement to enhance the predictive performance of system studying

59

few present records, it is a honest and probably manner to make a accurate estimate of our model. So we divide the information into two elements. Eighty% of the information will be used to teach the model and this facts might be referred to as education facts and 20% of the facts can be used to check the accuracy and performance of the model on information that the version has by no

means visible before. And will be called the check

statistics set. So there are 487 samples in our schooling pattern, and 122 samples in the check sample. Below is a flowchart that represents the raw data processing

Cross Validation

Cross validation is a version validation approach for evaluating how well the results of a machine getting to know algorithm will generalize to an unseen records set. The reason of pass-validation is to "check" the version dataset throughout the schooling phase (i.E. Usually with an unknown independent dataset.

k-Fold Cross Validation

One of the negative aspects of doing cross-validation is that we lose quite a number of the data that become used to educate the version, and thus the predictions may be advanced. In a conventional split educate, the metric errors can range significantly, i.E., the mistake relies upon greatly on which points fall within the schooling set and which fall inside the test, and for that reason the estimate can vary drastically. In line with how the division is made. To overcome those issues, a famous version estimation method is used, of the equal kind as move-validation, but with mild variations, called okay folder-validation.

k-fold Cross Validation

The gain of okay-cross-valuation is twofold as it does no longer depend how the records is cut up. Each given item hits the desired test once and arrives at the schooling set (k-1) time. The ensuing errors variance decreases because the estimate will increase. When we preprocessed the ASD dataset, we located no difficulties or troubles.

ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Note that the class selection tree will talk over with our version. In this manner, all destiny fashions that we study could be as compared to the choice tree of the classifier and it is able to be as accurate as, but nothing higher, than the choice tree of the classifier. In different phrases, no destiny version may be able to bias the decision of the classifier tree primarily based on its overall performance records.

installation, and prediction, meter fabrication instances and expenses are expected to lower. From the visualization shown in Figure eight, we will see that the most crucial function "outcome" (in step with the burden aspect) contributes plenty in comparison to different features. Here we list the primary 5 important functions so as of importance;

1. "Events";
2. "closer to himself";
3. "united states of america of house";
4. "yundis-no"
5. "Yundis-da"

This shows that we're trying to reduce the function area and simplify the statistics needed to construct the model. Thus we are able to install the model regardless of huge data sets in much less time.

A gini diagram of

PROJECT MANAGEMENT PLAN

Refinement

Here we discuss viable approaches to in addition improve the above fashions.

SVM: Finding the kernel offers accurate nonlinear validation outcomes.

KNN: Choosing K is continually difficult, and we determined to run the KNN algorithm the usage of specific values of K, via loops of values from K = eleven to K = ninety nine. The effects show that there's no great development in the move-validation ratings. For extraordinary values of K, in fact the first-class result changed into K = 94 with a cross-validation rating of 0.9606. An important feature. A principal mission in getting to know from written information, including the autism facts we study here, is figuring out which capabilities provide the maximum predictive strength. If the relationship among some key functions and the title is placed on the goal, we can simplify our expertise of the phenomenon, which is almost constantly beneficial. In this task, which means that we need to discover a small quantity of traits that most appropriately expect whether or not an person has ASD. Choose a scikit-study classifier (eg boosting, adaboost, random woodland) that has the feature_importances characteristic, that's the feature that has the most essential feature according to the selected classifier. Figure 17 indicates the 5 foremost functions for the ASD dataset the usage of special classifiers that attribute this selection as critical. We ought to ask how the version will perform if we simply use the whole thing.

Feature importance's with two Classifiers
Since fewer capabilities are required for formation,

Comparism of metrices using different learning algorithms.

TRANSITION/ SOFTWARE TO OPERATIONS PLAN

Mathematical Modeling:

The first step is to become aware of SNPs with the highest possibility of inclusive of one or greater ASD biomarkers,

using the array matrix in Excel GSE6754, insert each sickness or wholesome man or woman and genotype both AA, AB and BB diseased and healthful people. Populace

A -dimensional graphical representation calculates the space between an affected and an unaffected problem in genotype space. The AB axis isn't blanketed.

basic and superior statistical approaches from example and attribute calculations to figuring out confusion matrices and receiver running traits (ROC) curves.

A sample of 10 pinnacle-ranked SNPs turned into drawn from the 61 SNPs with the best distance between ASD patients and healthy people within the three-dimensional scoring space of the genotype. The distances of each SNP, whilst affected (A) and the population are not balanced, tend to be larger than for the A/U balanced population. Unequal and balanced cohorts are not matched among SNPs. The more the space, the more the likelihood of finding a SNP within the gene related to autism. In addition, all genes at every SNP have been detected and retrieved from all GPL2641 gene expression databases. Evidence of autism changed into accumulated to decide if every gene is an ASD biomarker. Of the 61 targeted SNPs inside the excessive-order mismatch cohort, 45 are associated with ASD. Other genes inside the 51 pinnacle-ranked SNPs are not listed in this desk PIK3C3, DLGAP1, RNF180, SPATA5L1, C13orf25, GLI3, C7orf25, CRP, FAM46D, TBX22, FANCL, ICOS, ALS2CR19, NAPE-PLD, SEL1L, FLRT2, RTTN, EIF3S3, TRPS1, ADARB2, SPRY2, DSCAM, EPM2A, UTRN, MAGEE2, CXorf26, RAG2, NGL-1, RAG1, TMOD1, NR5A2, PTPRC, RCHY1, SMARCC2, RNASEH1, FLT1, TPH1, ASCL1, CDH9, KHDRBS2, OPHN1, AR, SH3BGR1, ABCC12, NLGN4X, VCX3A, CA10, and KIF2B.

practical representation of X-Y-Z axis in EEG data The first five SNP distance sequences in the unbalanced (red) and balanced (orange) affected and healthful affected person cohorts are proven within the 3-d Cartesian

Gene Location in Biological Pathways:

For each of the 61 collection SNPs discovered in GSE6754, genes had been analyzed in saved biological tracts GPL2641.

The detection of the NRXN1 gene within the equal organic pathway is an example of gene identification. Machine learning evaluation:

Weka 3.8, evolved by using the University of Waikato in New Zealand, is a famous Java-based machine learning tool for data mining and knowledge retrieval. Excel accepts documents stored as CSV files and performs

61

aircraft. The distances highlighted in orange have become notably shorter after random deletion of the genotyping information for 1014 healthful topics.

visualization of EEG data The receiver running function (ROC) place underneath the curve turned into measured for genotyping facts with 10, 15, and 20 essential SNPs in step with thousand patients. Subfigures A-C are the facts collected when the population is affected and now not integrated, subfigures D-F are the information accumulated while the populace balances. Weka 3.8 uses the following tool mastering techniques: 10-fold cross-validation (CV) (A and D), N fold CV (B and E), and break up percent (C and F). The pink bars constitute the J48 type (decision tree), the gold bars are random wild, and the blue bars are simple.

REFERENCES:

- Ching, M. S. L., et al. Deletions of NRXN1 (neurexin-1) predispose to a wide spectrum of developmental disorders. American Journal of Medical Genetics. Part B, Neuropsychiatric Genetics: The Official Publication of the International Society of Psychiatric Genetics, 153B (4), 937–947. (2010).
- Chini, V., Shalaby, K., Al-Sarraj, Y., Taha, R., Alshaban, F., Kambouris, M., & El-Shanti, H. X linked Genes with Novel Rare Variants Identified by WGS in ASD Patients are Involved in Neurodevelopment. Qatar Foundation Annual Research Conference Proceedings, HBPP1350. (2016).
- Frank, E., Hall, M.A., Witten, I.H. The WEKA workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques." Retrieved August 2, 2018.
- Glatt, S. J., Tsuang, M. T., Winn, M., Chandler, S. D., Collins, M., Lopez, L., ...Courchesne, E. Blood Based Gene Expression Signatures of Autistic Infants and Toddlers. Journal of the American Academy of Child and Adolescent Psychiatry, 51(9), 934-44. e2. (2012).
- Kim, H.-G., Kishikawa, S., Higgins, A. W., Seong, I.-S., Donovan, D. J., Shen, Y., ...Gusella, J. F. Disruption of neurexin 1 associated with autism spectrum disorder. American Journal of Human Genetics, 82(1),

199-207. (2008).