University of East London

Msc. In Data Science

ADVANCED DECISION: PREDICTIVE ANALYTICS AND MACHINE LEARNING

(DS7003)

By

Motunrayo Aderemi (2328860)

Screening and Analysis of Autism Spectrum Disorder in Adult Using Machine Learning Techniques.

Course coordinator: Dr. Yang Li

Table of Contents

Abstract

Autistic Spectrum Disorder(ASD) is a neurodevelopmental disorder that incurs significant healthcare costs. Early diagnosis of ASD can help reduce these costs, but unfortunately, the diagnosis process is time-consuming and expensive. Given the economic impact of autism and the increasing prevalence of ASD worldwide, there is a pressing need for the development of efficient and effective screening methods.

This project aims to predict the presence of ASD in adults using the dataset available at https://archive.ics.uci.edu/ml/datasets/Autism+Screening+ Adult. The dependent variable in this dataset is the class, which indicates whether a patient is positive or negative for ASD. To achieve this goal, I will employ machine learning techniques such as decision tree, logistic regression, and KNN. By comparing the performance of these methods, I hope to identify the most effective approach for predicting ASD in adults.

1.0 Introduction

Autism is a neurodevelopmental disorder that is characterized by deficits in social interaction, communication, and restricted and repetitive behaviors. The American Psychiatric Association (2000) defines Autism Spectrum Disorder as a developmental disorder that typically appears in childhood, ideally within the first three years of life. The symptoms of ASD can vary widely, which is why it is referred to as a "spectrum" disorder. (Anibal *et al*.2018) Diagnosis of ASD can be challenging as no specific medical test can confirm it. While some people are diagnosed in childhood, others may not receive a diagnosis until adolescence or adulthood. There is no standardized treatment for ASD, but early detection and intervention can lead to improved outcomes. Scientists believe that genetic factors and environmental influences during development may contribute to the disorder. (Thabtah, 2018) Risk factors for ASD include low birth weight, having a sibling with ASD, and having elderly parents. Overall, individuals with ASD experience issues with social interaction and communication which includes

- Insensitivity to pain
- Difficulty maintaining proper eye contact.
- Limited response to sounds
- Lack of desire for cuddling
- Challenges expressing themselves through gestures.
- Limited social interaction with others
- Inappropriate attachment to objects
- Preference for solitary living
- Echolalia (repetition of words or phrases)
- Inappropriate laughter or giggling

Furthermore, individuals with ASD may display limited interests and repetitive behaviours, such as:

- Repeating words or phrases excessively
- Becoming upset when routines are disrupted.
- Developing a passing interest in specific topics (e.g., numbers, facts)
- Being less sensitive to light, noise, and other stimuli than others.

Detecting autism spectrum disorder early and providing appropriate treatment is essential for managing symptoms and improving quality of life for those affected. However, currently, there is no specific medical test available for detecting autism. (Vaishali *et al.* 2018)Observing a child's behavior is often the primary method of identifying ASD symptoms, and this is typically done by parents and teachers in older children and adolescents attending school. The school's special education team can evaluate the symptoms and recommend further testing if necessary. However, identifying ASD symptoms in adults is more challenging due to the overlapping symptoms with other mental health disorders. Although Autism-specific brain imaging can only be performed after 2 years of age, observing a child's behavioral changes can aid in identifying ASD symptoms as early as 6 months of age.(Mythili *et al.* 2014)

1.1 Related Works

M.S. Mythili, A.R. Mohamed Shanavas(2014) and their team conducted a study on ASD that used classification techniques to detect the severity and presence of autism. The researchers employed SVM and Fuzzy techniques along with WEKA tools to analyze students' behavior and social interaction. In a separate study, Mariam(2019) and her colleagues used the decision tree algorithm to examine data from the National Database for Autism Research (NDAR).

Fadi Thabtah et al(2017) proposed an ASD screening model based on Machine Learning Adaptation that is designed to align with the DSM-5. This screening tool's purpose is to identify individuals with ASD and achieve one or more screening goals. The authors of this paper investigated the benefits and drawbacks of Machine Learning classification in ASD screening, as well as issues with existing screening tools and their dependability. They did this by contrasting the use of the DSM-IV manual with the DSM-5 manual.

J.A. Kosmicki and colleagues (2015) proposed a machine learning method to detect autism by identifying the fewest number of traits possible in their study. They evaluated a subset of autism-related behaviours in children using the Autism Diagnostic Observation Schedule (ADOS), specifically clinical assessments of ASD. The ADOS is made up of four modules. To identify stepwise backward features, the researchers used eight different machine learning algorithms on score sheets from 4540 people. The study discovered that ASD risk could be identified using nine of the 28 Module 2 behaviours and twelve of the 28 Module 3 behaviours, with overall accuracies of 98.27% and 97.66%, respectively. The study's goal was to find the most accurate and efficient method of detecting ASD.

Li B and colleagues (2017) conducted a study to explore the use of machine learning classifiers for detecting autism in adults using imitation methods. The study aimed to identify discriminative test conditions and kinematic parameters. Sixteen individuals with autism spectrum disorder participated in the study and performed a series of hand movements. Machine learning methods were employed to extract 40 kinematic constraints from 8 imitation conditions. The study found that machine learning methods can be used to analyze high-dimensional data and achieve diagnostic classification of autism with a small sample size. The RIPPER algorithm achieved sensitivity rates of 87.30% for Va, 80.95% for CHI and IG, 84.13% for Correlation and CFS, and 80.00% for no feature selection on the AQ-Adolescent dataset.

2.0 Exploratory Data Analysis

Data exploration analysis (EDA) is the study and evaluation of data to discover patterns, relationships, and trends. Using summary statistics and graphical representations, this process involves conducting preliminary investigations on data to identify anomalies, detect patterns, validate assumptions, and test hypotheses. EDA is important in data science and machine learning because it allows researchers to gain insights into the underlying structure and distribution of data.

2.1 Dataset Description

The dataset for this study was obtained from the UCI Repository, which is available publicly on https://archive.ics.uci.edu/ml/datasets/Autism+Screening+Adult. The datasets used are the Adult Autism Screening Datasets. Each dataset contains 20 attributes with continuous, categorical, and binary values. Class is the dependent attribute that determines whether an individual has ASD (1) or not (0).

Serial No	Attributes	Description	Data Type
1-10	A1_Score to A_10 score	Questions based on the screening method	Binary
11	Age	Age of patients	Integer
12	Gender	Gender of the patients	String
13	Ethnicity	Patient's ethnic group	String
14	Jaundice	If the patient had jaundice at the time of birth	String
15	Autism	If any family member has Autism	String
16	Country of residence	Country of the patient	String
17	Used app before	If the screening app has been used by the patient before or not	Strings
18	Score	Screening score based on the 10 questions.	Integer
19	Relation	Who is answering the questions	Strings
20	Class Asd	Diagnosis result	Strings

Fig 2.1; A table showing the description of the dataset.

2.2 Data Pre-processing

Data pre-processing is the process of converting raw data into an analysis-ready format. It consists of a set of techniques and methods used to clean, transform, and prepare data for analysis. Data pre-processing's primary goal is to ensure that the data is accurate, complete, consistent, and relevant for the analysis task at hand. Data cleaning was done on the dataset before the analysis.

Data cleaning which is the process of detecting and correcting errors or inconsistencies in data, such as missing values, outliers, and duplicates. During the processing and cleaning of this data which was done Excel before loading on R for analysis, some missing values were encountered which could lead to errors in the analysis 96 missing data were removed from 704, which leaves the analysis with 608 observations. The missing data were removed to avoid error in the analysis. An outlier was also encountered in the age attributes which was removed during the cause of data cleaning. Also, during this process, some observations were spelt incorrectly which could lead to error in the analysis.

2.3 Data Transformation

The process of converting data from one format to another is known as data transformation. Data transformation is a way of converting data from one structure, or representation to another in order to make it more suitable for analysis or other applications. Depending on the nature of the data and the specific goals of the analysis, data transformation can involve a wide range of techniques**.** The technique used in this work is data normalization. The dataset was normalized using the SoftMax function. The SoftMax function takes a vector of numerical values as input and converts them so that their sum is between 0 and 1. SoftMax normalisation was used to transform variables with different value ranges so that they could be compared meaningfully.

Also, Attributes like Gender, Ethnicity, Jaundice, Autism, Country, Used app, Relation, and Class were converted from strings into Binary and Integers using 1 and 0 for easy implementation of the algorithms. This was done on Excel before loading the dataset on R for analysis.

Class	Yes- 1 No- 0
Gender	F- 1 M- 0
Jaundice	Yes- 1 No- 0
Autism	Yes- 1 No- 0
Used app	Yes- 1 No- 0

Fig 2.2; A table showing transformed data.

2.4 Data Visualization

2.4.1 Missing map Observation

The data was imported into R after it was collected from the UCI site. R creates a working directory and reads the file to look for missing data. A library called Amelia was used to look for missing data. This assists in visualising the entire dataset and provides information on available and missing data. The data was already cleaned on excel during data pre-processing before it was loaded into R so there was no missing data in this case. R read the data and found no missing data, as shown in the figure below.

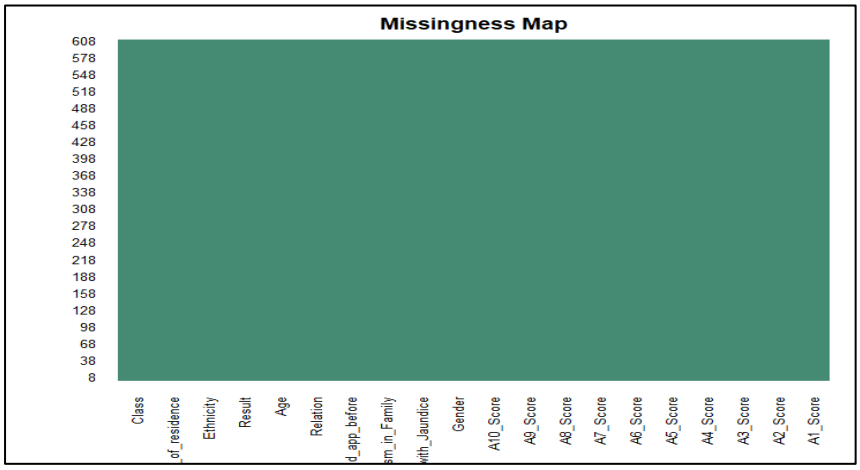


Fig 2.3; A plot to show missing data.

2.4.2 Bar plots representing Ethnicity and Class

A bar plot is a common graphical representation for categorical data in R. The frequency or proportion of observations in each category of a categorical variable can be visualised using bar plots.

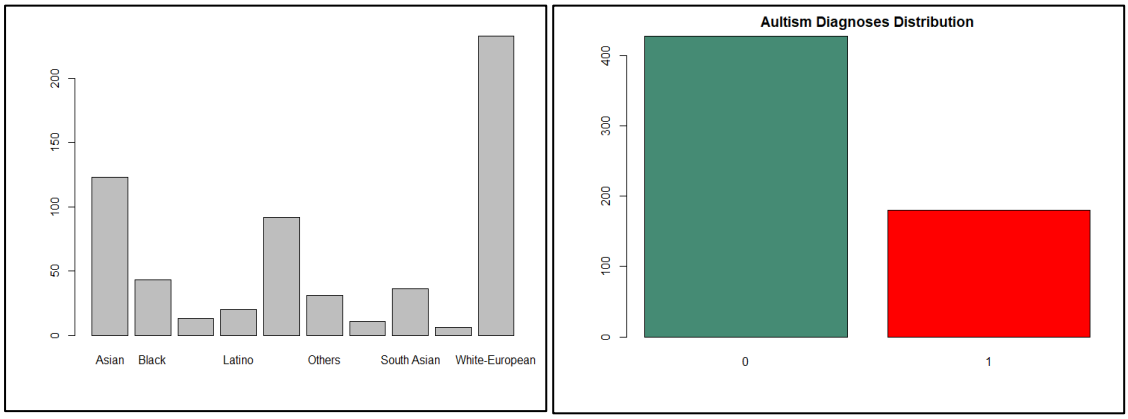


Fig 2.4; Bar plots showing Ethnicity and class.

The above plots shows for Ethnicity, which White-European tops the chat with 233 autism patients from white-European and also the plot showing the class which shows that results with No which is represented by 0 is more than results with Yes which is represented with 1. There is an higher number of people without Autism.

2.4.3 Visualizing The class distribution by independent variables using Barplots

When we use barplots to visualise the class distribution by independent variables, we are essentially displaying the frequency or proportion of observations in each category of the independent variable(s) and how it relates to the dependent variable (i.e., class).

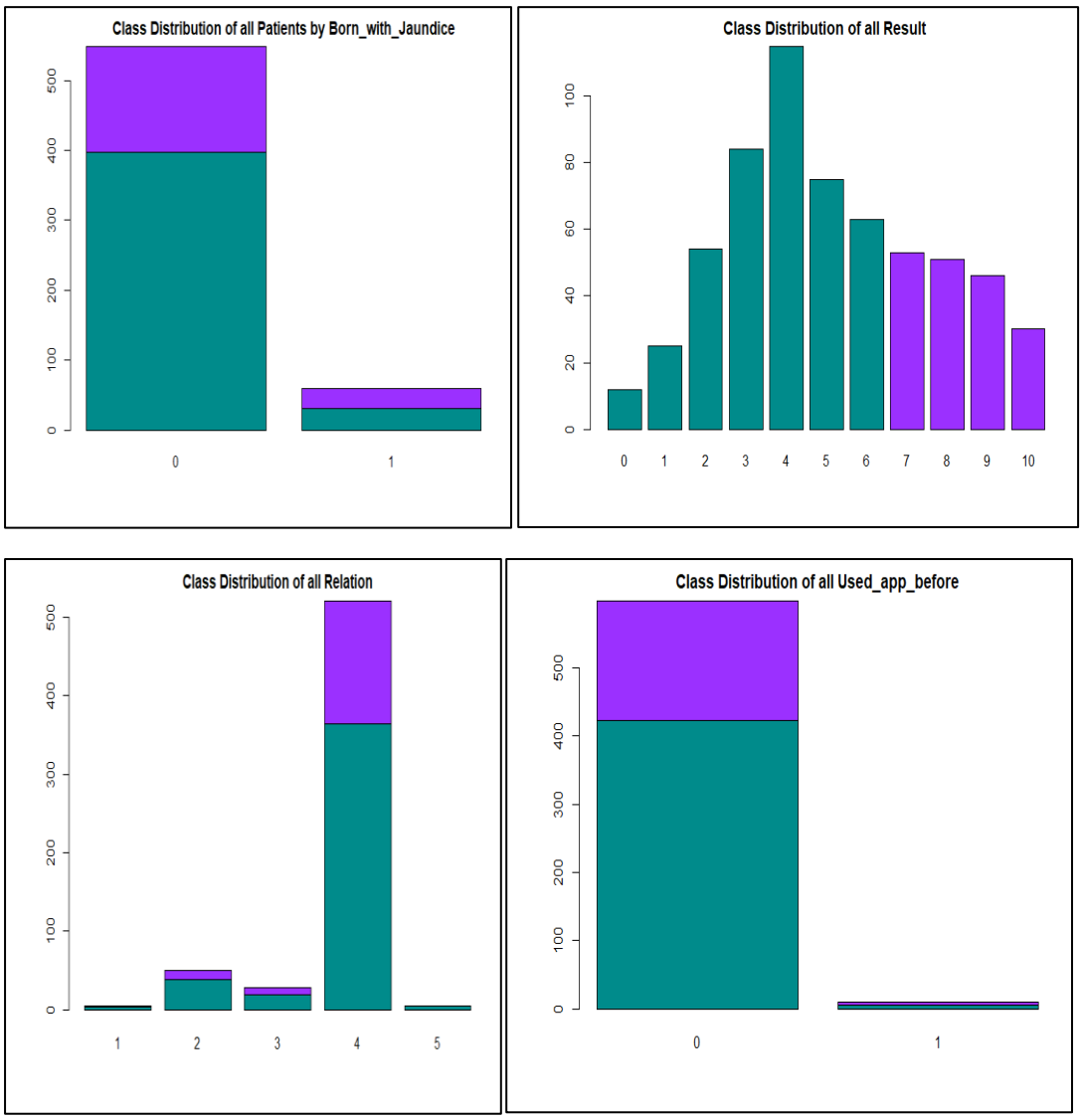


Fig 2.5 Bar plots showing the class distribution by independent variables.

Above are the bar plots showing the barplot of class distribution of all the patients by some of their independent variables. This shows a higher proportion of people born with jaundice don't have autism and the lower proportion of people born with jaundice have the Autism. The second plot shows that there are more patients with score of 4 after taking the question test. The third plot shows that the highest proportion were the people who answered the questions themselves. The fourth plot shows that there is highest proportion of people who have used the app before than people who haven't.

2.4.4 Visualizing The percentage distribution using Pie chart

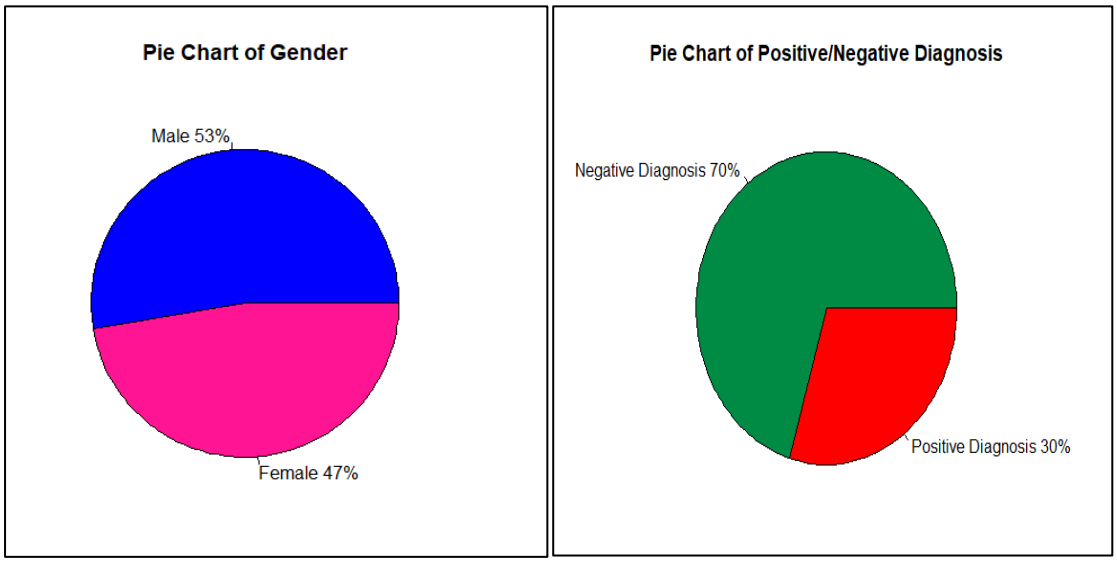


Fig 2.6; Pie charts showing percentage distribution.

Above are the pie charts showing the percentage distribution of Gender which shows that there is a higher percentage of Males with 53% compared to Females with 47% and the Class of ASD which shows a higher percentage of Negative diagnosis(as represented by 0) with 70% as compared with Positive diagnosis(as represented by 1) with 30%.

2.4.7 Creating a boxplot to show the distribution of Autism Data

Boxplots are graphical representations of numerical data distributions through their quartiles that can be used to summarise the distribution of a variable and identify outliers. In data analysis and statistics, they are commonly used to display and compare the distribution of one or more variables across different groups or categories.

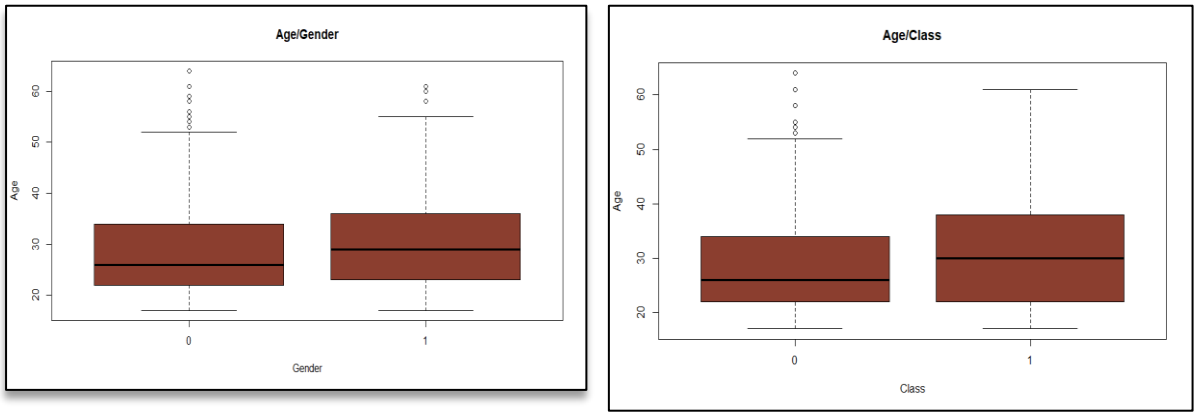


Fig 2.7; Boxplot showing the distribution of age against gender and Age against class.

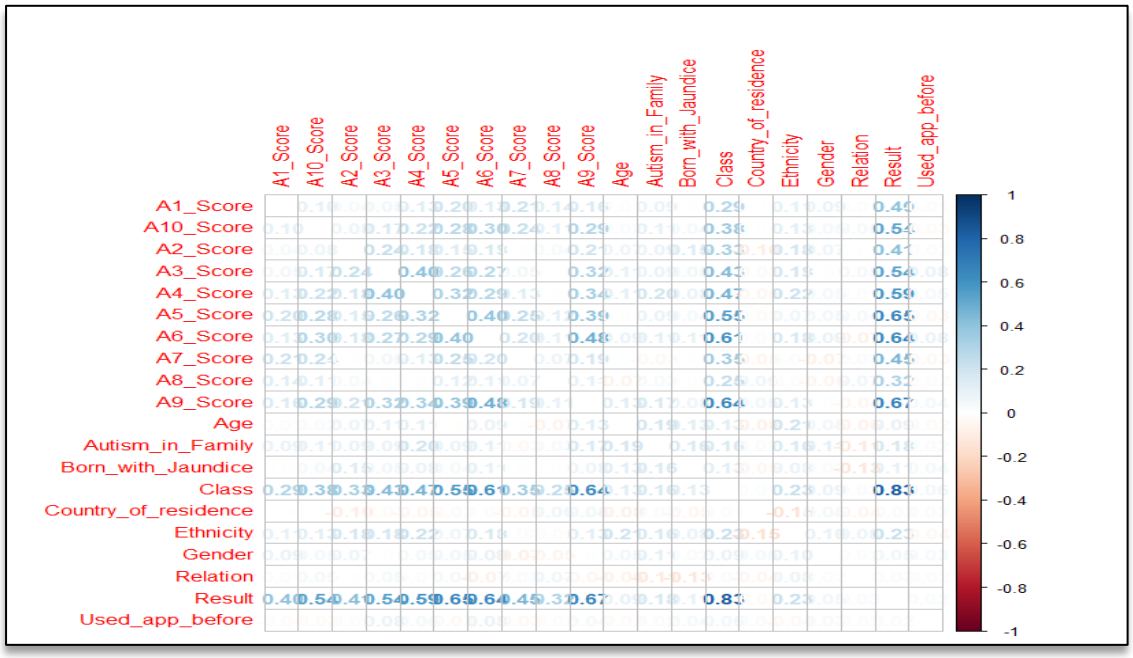
The boxplot above depicts the distribution of the Age variable for each level of the Gender variable. And for each level if class variable. The boxplot is divided into two sections, one for each level of the Gender variable, and each box represents the Age variable distribution for the gender and class. The height of each box represents the data's interquartile range (IQR), which is the range between the 25th

and 75th percentiles. The line inside the box represents the data's median, and the whiskers extend from the box to show the data's range, excluding any outliers. The boxplot can provide a quick visual overview of the Age variable by Gender and class.

2.5 Correlation Analysis

2.5.1 Test for correlation and multicollinearity using matrix.

A test for correlation which is used to measure the relationship between two variables was conducted on the dataset. The correlation matrix shown aids in identifying strong positive or negative correlations between pairs of variables, as well as any groups of variables that are highly correlated with one another. The plot's colour scale helps in highlighting any particularly strong correlations. The correlation coefficients displayed in each cell also helps to quantify the degree of correlation between variables. In this dataset correlation matrix, there is a strong positive correlation between the dependent variable(class) and an independent variable (Result). There is also a positive correlation between the A1_score to A10_score and Result and some other variables.



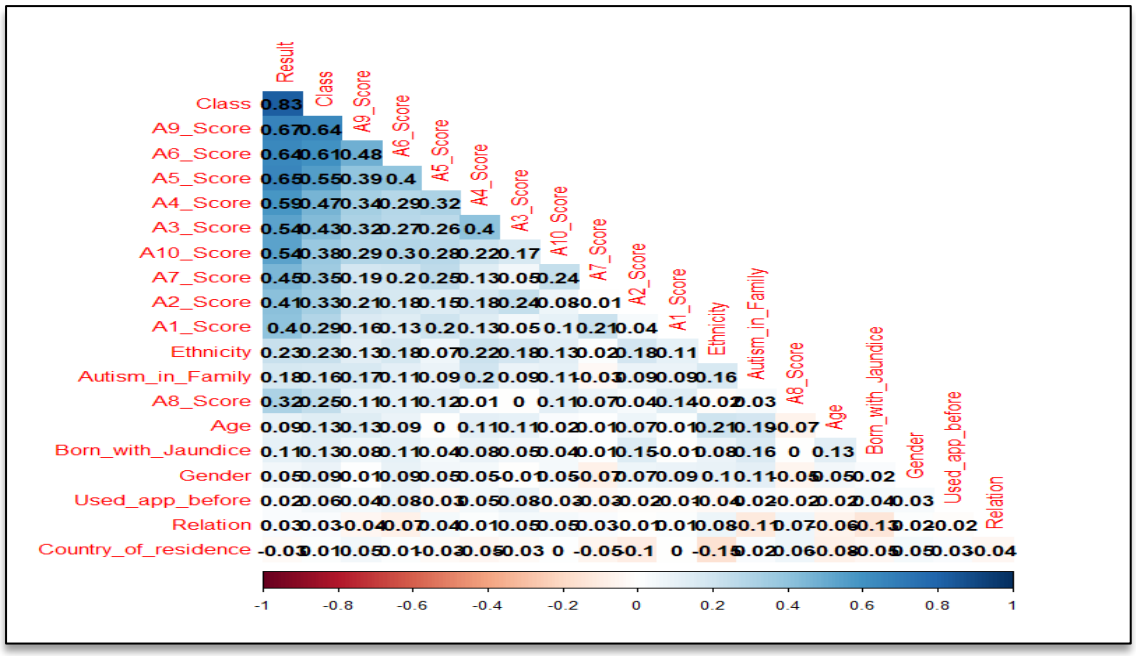


Fig 2.8; Correlation matrix plots for all attributes in the data set to demonstrate the relationship between dependent and independent attributes.

2.5.2 Test for Partial Correlation

The process of partial correlation involves analyzing the relationship between two variables while taking into consideration the impact of a third variable. If the resulting p-value is below the chosen level of significance, typically 0.05, the null hypothesis is rejected, and it is concluded that there is a significant partial correlation between the two variables after controlling for the third variable. Conversely, if the p-value exceeds the level of significance, the null hypothesis is accepted, and it is concluded that there is insufficient evidence to support the existence of a significant partial correlation. The following tests display the partial correlation between the dependent variable (Class) and several independent variables that may exhibit partial correlation.

- **Test for Class, Ethnicity and Country of residence**

```
> pcor.test(Class, Ethnicity, Country_of_residence) #Geographical factors
   estimate      p.value statistic   n gp  Method
1 0.2323985 6.897948e-09  5.877161 608  1 pearson
```

This test assesses the relationship between "Class" and "Ethnicity," while controlling for the effect of "Country_of_residence." The p-value of the test is 0.000000000689, which is very small and well below the significance level of 0.05. Based on the p-value being less than the chosen significance level (usually 0.05), The null hypothesis is rejected and it is concluded that is a significant partial correlation between the variables Class, Ethnicity, and Country_of_residence.

- **Test for Class, Age and Gender**

```
> pcor.test(Class, Age, Gender) #Base level factors
   estimate     p.value statistic   n gp  Method
1 0.1282674 0.001541539  3.181239 608  1 pearson
>
```

The p-value of the test 0.00154 which is below 0.05. we can reject the null hypothesis and conclude that there us a significant correlation between Class, Age and Gender.

- **Test for Class, A10_Score and Result.**

```
> pcor.test(Class, A10_Score, Result) #Lab results factors
    estimate      p.value statistic   n gp  Method
1 -0.1415221 0.0004702346 -3.516376 608  1 pearson
>
```

The p-value of the test 0.00047which is below 0.05. The null hypothesis is rejected and it is concluded that there us a significant correlation between Class, A10_Score and Result.

Pearson correlation test was also carried out to check for the correlation between dependent variables and all the independent variables which shows positive correlations between the dependent variable and independent variables, but the strongest positive correlation happens to the between Class and Result.

2.5.3 Test for Correlation using Pearson's Test

Pearson's correlation test is a technique in statistics that measures the direction and strength of a linear relationship between two variables that are continuous. This method uses Pearson's correlation coefficient (r) to evaluate the degree of association between the two variables. The coefficient value ranges from -1 to +1, where -1 indicates a perfect negative correlation, +1 indicates a perfect positive correlation, and 0 represents no correlation.

- **Test for Class and Age**

```
> cor.test(Class, Age)

        Pearson's product-moment correlation

data:  Class and Age
t = 3.2861, df = 606, p-value = 0.001075
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.05335966 0.20962409
sample estimates:
      cor
0.1323139
```

The p-value of the test 0.001075 which is below 0.05. The null hypothesis is rejected, and it is concluded that there is a significant correlation between Class and Age

- **Test for Class and Jaundice**

```
> cor.test(Class, Born_with_Jaundice)

        Pearson's product-moment correlation

data:  Class and Born_with_Jaundice
t = 3.1821, df = 606, p-value = 0.001537
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.04918365 0.20561756
sample estimates:
      cor
0.1281979
```

The p-value of the test 0.001537 which is below 0.05. The null hypothesis is rejected, and it is concluded that there is a significant correlation between Class and Jaundice

- **Test for Class and Result**

```
> cor.test(Class, Result)

        Pearson's product-moment correlation

data:  Class and Result
t = 36.227, df = 606, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8002241 0.8506785
sample estimates:
      cor
0.8271104
```

The p-value of the test 0.000000000000000022 which is below 0.05. The null hypothesis is rejected and it is concluded that there is a significant correlation between Class and result. This appears to be strongest positive correlation.

3.0 Methodology

3.1 Factor Analysis

Factor analysis is a statistical technique used to discover the underlying factors or dimensions that account for the correlations between a collection of observed variables. The observed variables are classified into factors in a factor analysis based on the common patterns of variation they share. These variables stand for underlying dimensions or constructs that are inferred from observed variables but cannot be seen directly.

Different techniques, such as principal component analysis (PCA) can be used to conduct factor analysis. A method called PCA finds the factors that contribute the most variance to the data, whereas a more sophisticated technique called MLE calculates the likelihood of the observed data given a set of fictitious factor loadings. A set of factor loadings, which represent the direction and strength of the relationship between each observed variable and each factor, is typically included in the output of a factor analysis. The factor loadings can be used to analyse each factor's significance as well as judge the validity and dependability of the factor analysis.

3.1.1 KMO Test

Kaiser Meyer Olkin (KMO) test was conducted to measure the suitable attributes for my analysis. The KMO statistic has a range of 0 to 1, with values closer to 1 indicating that factor analysis would be more appropriate for the observed variables. For factor analysis, a KMO value of 0.6 or higher is typically acceptable.

A.

```
library(psych)
# The Kaiser-Meyer-Olkin (KMO) test to measure suitable attributes for my analysis
KMO(Aultism.mat)
```

B.

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = Aultism.mat)
Overall MSA =  0.5
MSA for each item =
            A1_Score             A2_Score             A3_Score             A4_Score             A5_Score
                 0.5                  0.5                  0.5                  0.5                  0.5
            A6_Score             A7_Score             A8_Score             A9_Score            A10_Score
                 0.5                  0.5                  0.5                  0.5                  0.5
              Gender   Born_with_Jaundice     Autism_in_Family      Used_app_before             Relation
                 0.5                  0.5                  0.5                  0.5                  0.5
                 Age               Result            Ethnicity Country_of_residence                Class
                 0.5                  0.5                  0.5                  0.5                  0.5
```

Fig 3.1; (A) R code of KMO test, (B) KMO adequacy result for all the attributes

Above is the result from the result from the KMO test which shows the overall MSA as 0.5. This suggest that the dataset is good for the analysis and there will not be a need for factor analysis.

KMO measure	Interpretation
KMO ≥ 0.90	Marvelous
0.80 ≤ KMO < 0.90	Meritorious
0.70 ≤ KMO < 0.80	Average
0.60 ≤ KMO < 0.70	Mediocre
0.50 ≤ KMO < 0.60	Terrible
KMO < 0.50	Unacceptable

Fig 3.2; Kaiser Meyer Olkin (KMO) level of acceptance by the adequacy value.

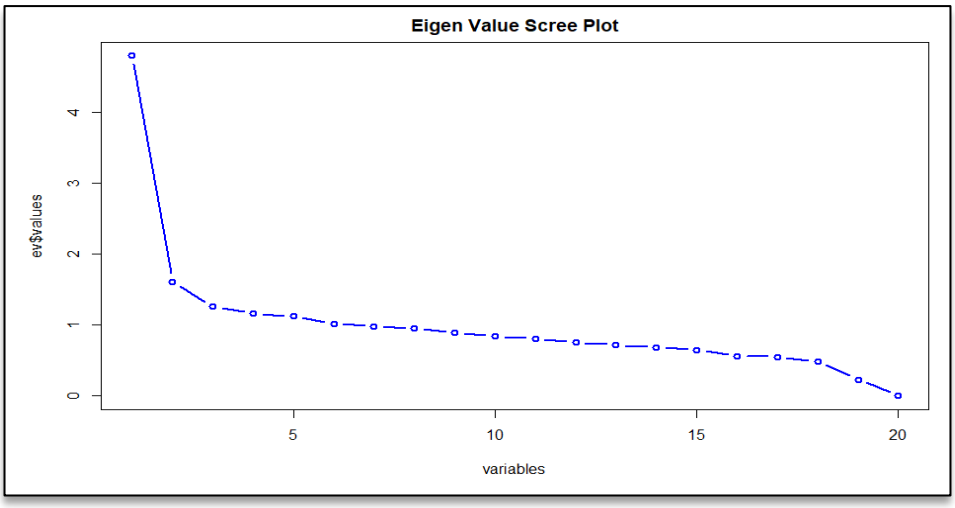
3.1.2 Eigen Value analysis

The number of factors to keep in the analysis is determined using the eigenvalue analysis technique in factor analysis. The objective of factor analysis is to isolate a small set of latent factors that can account for the covariance between observed variables in a dataset.

A.

```
> ev$values
 [1] 4.790874e+00 1.605637e+00 1.252520e+00 1.163258e+00 1.117082e+00 1.014550e+00 9.765437e-01
 [8] 9.458846e-01 8.926679e-01 8.325180e-01 8.013179e-01 7.519975e-01 7.136689e-01 6.843052e-01
[15] 6.453593e-01 5.563176e-01 5.488175e-01 4.785752e-01 2.281058e-01 4.797543e-16
> sum(ev$values)
[1] 20
```

B.



C.

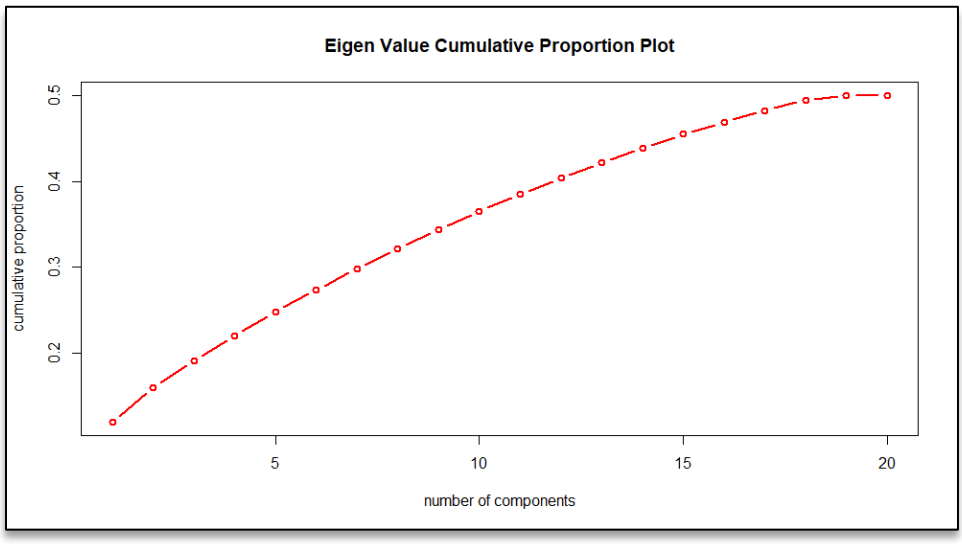


Fig 3.3; (A) Eigen Value (B) Eigen Value Scree plot (c) Eigen Value cumulative proportion plot

An Eigen value analysis and a scree plot of the eigen values was conducted to support the KMO result. As a result of factor analysis, no variable was removed.

3.1.3 Principal Component Analysis (PCA)

PCA was the technique used for the factor analysis. This Principal Components Analysis was carried out on the dataset. PCA was designed to reduce data dimensionality by extracting latent variables, known as principal components, that account for the greatest amount of variation in the original variables.

```
> pca
Principal Components Analysis
Call: principal(r = Aultism.mat[-120], nfactors = 5, rotate = "varimax",
    method = "maximum likelihood")
Standardized loadings (pattern matrix) based upon correlation matrix
                     RC1   RC4   RC3   RC2   RC5   h2    u2 com
A1_Score            0.15  0.59  0.05 -0.03  0.25 0.43 0.565 1.5
A2_Score            0.43 -0.16  0.27  0.05  0.08 0.30 0.705 2.1
A3_Score            0.68 -0.28  0.12 -0.10 -0.02 0.57 0.427 1.5
A4_Score            0.64 -0.06  0.17  0.02  0.10 0.45 0.546 1.2
A5_Score            0.60  0.32 -0.04 -0.05 -0.04 0.47 0.527 1.6
A6_Score            0.66  0.17 -0.10  0.16  0.05 0.51 0.493 1.3
A7_Score            0.25  0.57  0.08  0.02 -0.33 0.51 0.490 2.1
A8_Score            0.13  0.45 -0.21 -0.08  0.04 0.27 0.730 1.7
A9_Score            0.70  0.15 -0.09  0.17  0.01 0.55 0.448 1.3
A10_Score           0.42  0.36  0.02  0.00  0.02 0.31 0.691 2.0
Gender              0.02  0.03 -0.04 -0.07  0.76 0.58 0.417 1.0
Born_with_Jaundice  0.12 -0.07  0.13  0.59 -0.01 0.39 0.612 1.2
Autism_in_Family    0.14  0.07  0.08  0.46  0.47 0.46 0.536 2.3
Used_app_before     0.21 -0.39 -0.35  0.01  0.08 0.32 0.676 2.6
Relation            0.05  0.03  0.21 -0.72  0.13 0.59 0.414 1.2
Age                 0.09 -0.04  0.42  0.40  0.21 0.39 0.607 2.6
Result              0.91  0.39  0.06  0.03  0.02 0.98 0.023 1.4
Ethnicity           0.23  0.01  0.59  0.00  0.35 0.52 0.479 2.0
Country_of_residence 0.00  0.01 -0.67  0.03  0.26 0.52 0.476 1.3
Class               0.83  0.30  0.01  0.07  0.07 0.79 0.209 1.3

                       RC1  RC4  RC3  RC2  RC5
SS loadings           4.30 1.67 1.36 1.35 1.25
Proportion Var        0.21 0.08 0.07 0.07 0.06
Cumulative Var        0.21 0.30 0.37 0.43 0.50
Proportion Explained  0.43 0.17 0.14 0.14 0.13
Cumulative Proportion 0.43 0.60 0.74 0.87 1.00

Mean item complexity =  1.7
Test of the hypothesis that 5 components are sufficient.

The root mean square of the residuals (RMSR) is  0.08
 with the empirical chi square  1374.31  with prob <  6.9e-223

Fit based upon off diagonal values = 0.87
```

Fig 4.5; Result of the factor analysis using Principal Component Analysis.

3.2 Machine Learning

Machine learning is a field of artificial intelligence that involves the development of algorithms and statistical models that enable computers to learn from data and make predictions or decisions without being explicitly programmed for every scenario. Machine learning algorithms can recognise relationships and patterns in complex datasets that are large and use these patterns to predict or classify new data. The three types of machine learning include:

In **Supervised learning**, the machine is trained on a labelled dataset that contains both input and output data. Based on the patterns and relationships discovered during the training process, the machine learns to make predictions on new data.

Unsupervised learning requires the machine to identify patterns or relationships in unlabelled data without any guidance.

Reinforcement learning involves a machine learning approach that relies on receiving feedback in the form of rewards or punishments based on its actions and learning through trial and error. The goal is for the machine to make decisions that maximize the rewards it receives.

3.3 Classification Modelling

Five classification models were used in the study which includes Logistic Regression, Decision tree and K-Nearest Neighbors. The performance of each model was compared based on Accuracy, F1 score and precision. Below, you will find a succinct summary of the classification models utilized in our analysis.

3.3.1 Logistic Regression

Logistic Regression is a statistical method used for performing binary classification tasks, which involves predicting a binary outcome variable based on one or more independent variables. The technique models the relationship between the binary outcome variable and the independent variables using the logistic function, also known as the sigmoid function. The logistic function transforms the output of the linear regression into a probability value ranging from 0 to 1. The predicted binary outcome variable is then determined based on this probability. Logistic regression aims to identify the best model that describes the relationship between the binary outcome variable of interest and the independent variables, by fitting a logistic function to the data.

3.3.2 K-Nearest Neighbors (KNN)

K-Nearest Neighbours (KNN) is a technique of classification that identifies the k-nearest data points in the training dataset to a new data point and assigns the most of the class of those k-nearest neighbours as the new data point's class. KNN is a non-parametric technique, implying it doesn't make any assumptions about the data's underlying distribution. It is a simple and user-friendly algorithm suitable for both classification and regression problems.

3.3.3 Decision Tree

The Decision Tree algorithm is widely used for classification and involves partitioning the input space into subsets based on the values of input features. At each step, the algorithm selects the best feature to divide the data into the most homogeneous subsets with respect to the target variable. This recursive process continues until a stopping criterion is met, such as reaching the maximum tree depth or the minimum number of samples per leaf. Decision Trees are known for their simplicity and interpretability, and they can be used for both classification and regression tasks. However, they may overfit and be affected by minor changes in the data.

3.4 Confusion Matrix

A confusion matrix is a table used to assess the effectiveness of a classification model. It compares a model's predicted labels to the data's true labels and summarises the results in a table. The table is divided into two rows and two columns to represent the binary classification problem's two classes (positive and negative). The number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) predicted by the model is represented by the four cells in the table.

True Positives: These are positive cases that were predicted correctly to be positive.

False Positives: These are negative cases that were predicted incorrectly to be positive.

True Negatives: These are negative cases that were predicted correctly to be negative.

False Negatives: These are positive cases that were predicted incorrectly to be negative.

The confusion matrix is a useful tool for assessing a classification model's accuracy, precision, recall, and F1 score.

3.5 Performance Metrics

Precision is a performance metric that assesses a model's ability to correctly identify positive cases, or the accuracy of its positive predictions. It calculates the percentage of positive predictions that are true positives while ignoring false positives. A high precision score indicates that the model has a low proclivity to misclassify negative cases as positive and vice versa. A low precision score, on the other hand, indicates that the model is more likely to misclassify negative cases as positive, resulting in a high false positive rate.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Accuracy is a metric that measures the proportion of correctly classified instances regardless of their actual class to evaluate a classification model's overall performance. In simple terms, it quantifies the model's frequency of correct predictions over the total number of predictions made.

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

Recall is used to evaluate the ability of a classification model to identify positive cases correctly out of all the actual positive cases. The calculation involves dividing the number of true positive cases by the total number of positive cases. A high recall score means that the model correctly identifies a significant proportion of positive cases while producing a low false negative rate. Conversely, a low recall score implies that the model misses many positive cases and has a high false negative rate.

$$\text{Recall} = \frac{TP}{TP+FN}$$

Specificity is a performance metric that assesses the model's ability to identify negative cases correctly out of all actual negative cases. It denotes the proportion of true negative cases in relation to the total number of negative cases.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

F1 score is a performance metric that assesses precision and recall in a balanced manner. It is calculated as the harmonic mean of precision and recall and yields a single score that considers both metrics. The F1 score is a number between 0 and 1, with 1 indicating excellent precision and recall and 0 indicating poor performance.

$$\text{F1-Score} = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$$

3.6 Testing and Training of Model

The testing and training process helps prevent issues of model overfitting or underfitting. Overfitting occurs when a model is too complex and captures noise from the training set, leading to poor performance on the testing set. Underfitting, on the other hand, happens when a model is too simple and fails to capture the underlying patterns in the data, resulting in poor performance on both the training and testing sets.

For this analysis, package called **"**caret" was used to test and train the model. The seed was set to "3690" and the data was split into testing and training datasets by ratio 70:30 respectively.

```
set.seed(3690)
library(caret)
Train <- Aultism.nm[1:426, ]
Test <- Aultism.nm[427:608, ]

prop.table(table(Train$Class)) #70
table(Train$Class)
prop.table(table(Test$Class)) #30
table(Test$Class)
```

Because of the imbalanced data, over sampling was done to make the train and test data balanced.

```
> prop.table(table(Train.dat$Class))

  0   1
0.5 0.5
> table(Train.dat$Class)

  0   1
311 311
```

4.0 Results and Discussion

4.1 Results

Three different algorithms were used in this study to evaluate the performance of our machine Learning models. These include Logistic Regression (LR), K-Nearest Neighbors (KNN), and Decision Tree. In order to assess the effectiveness of these models, we utilized the confusion matrix and F1 measure.

4.1.1 Logistic Regression

A logistic regression was carried out after setting another seed to "12480". The dependent attribute (class) was factored against independent attributes in the data set. The Model was initially built with the dependent variable against all the independent variables. This Model showed no correlation between dependent and independent variable as there was no significance. Some variables were removed with the guidance correlation plot and factor analysis.

The Regression model was built using the Generalised Linear Model (glm)with a binomial family and a logit link function. "Class," a binary response variable that denotes the existence or absence of an autism spectrum disorder, is the measure of response. The predictor variables include different autism-related scores, demographic data (such as gender, age, ethnicity, and country of residence), and additional elements (such as whether the individual had jaundice at birth or has a family history of autism, as well as whether they have ever used an app related to autism).Two algorithms, "**LR.Model1**" and "**LR.Model2**," were modelled, and their accuracies were 93.4% and 94.5%, respectively. As compared to my LR.Model1 , LR.Model2 has a higher value and performed better.

LR Model 1

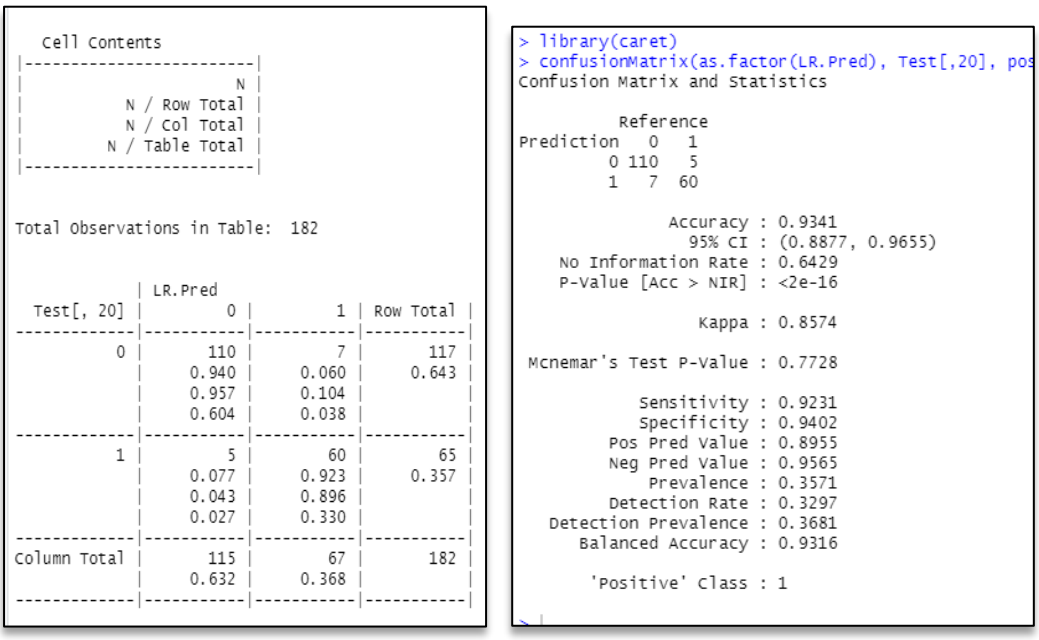
In Model1, it was observed that eleven variables were statistically significant, with Age having the lowest level of significance. To check for multicollinearity in the model, Variance Inflation Factor (VIF) was used. VIF measures the degree of correlation between independent variables in a regression model, and higher values indicate a higher degree of multicollinearity. The VIF values for all the variables were found to be within acceptable limits, indicating that there were no internal correlations among the independent variables, and hence no multicollinearity in the model. Therefore, the model can be considered reliable.

The summary of the Model

```
Call:
glm(formula = Class ~ A1_Score + A2_Score + A3_Score + A4_Score +
    A5_Score + A6_Score + A7_Score + A8_Score + Gender + Born_with_Jaundice +
    Autism_in_Family + Used_app_before + Age + Ethnicity + Country_of_residence +
    Relation, family = "binomial", data = Train.dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9434  -0.0053   0.0001   0.0365   2.4094

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -28.8839     4.6105  -6.265 3.73e-10 ***
A1_Score               4.3706     1.0827   4.037 5.42e-05 ***
A2_Score               4.2754     0.8594   4.975 6.53e-07 ***
A3_Score               3.3868     0.7282   4.651 3.30e-06 ***
A4_Score               5.5882     1.0351   5.399 6.71e-08 ***
A5_Score               6.4589     1.0750   6.008 1.87e-09 ***
A6_Score               5.7715     0.9432   6.119 9.41e-10 ***
A7_Score               5.1083     0.9567   5.340 9.32e-08 ***
A8_Score               3.9191     0.9615   4.076 4.58e-05 ***
Gender                 1.9227     0.6810   2.823  0.00475 **
Born_with_Jaundice     1.2535     1.6413   0.764  0.44505
Autism_in_Family       0.4820     1.0741   0.449  0.65362
Used_app_before        8.0030     6.8523   1.168  0.24284
Age                    2.7569     1.4771   1.866  0.06197 .
Ethnicity             -0.5535     0.7395  -0.749  0.45414
Country_of_residence   2.4124     1.6986   1.420  0.15555
Relation               1.9188     2.0410   0.940  0.34715
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 862.28  on 621  degrees of freedom
Residual deviance: 107.44  on 605  degrees of freedom
AIC: 141.44

Number of Fisher Scoring iterations: 9
```

Fig 4.1; Logistic model result for LR.Model1

Accuracy of the Result



```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  182


             | LR.Pred
  Test[, 20] |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       110 |         7 |       117 |
             |     0.940 |     0.060 |     0.643 |
             |     0.957 |     0.104 |           |
             |     0.604 |     0.038 |           |
-------------|-----------|-----------|-----------|
           1 |         5 |        60 |        65 |
             |     0.077 |     0.923 |     0.357 |
             |     0.043 |     0.896 |           |
             |     0.027 |     0.330 |           |
-------------|-----------|-----------|-----------|
Column Total |       115 |        67 |       182 |
             |     0.632 |     0.368 |           |
-------------|-----------|-----------|-----------|
```

```
> library(caret)
> confusionMatrix(as.factor(LR.Pred), Test[,20], pos
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 110   5
         1   7  60

               Accuracy : 0.9341
                 95% CI : (0.8877, 0.9655)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8574

 Mcnemar's Test P-Value : 0.7728

            Sensitivity : 0.9231
            Specificity : 0.9402
         Pos Pred Value : 0.8955
         Neg Pred Value : 0.9565
             Prevalence : 0.3571
         Detection Rate : 0.3297
   Detection Prevalence : 0.3681
      Balanced Accuracy : 0.9316

       'Positive' Class : 1
```

Fig 4.2; (a) Cross table analysis and, (b) Confusion matrix summary of LR.Model1

The cross table shows the predictions made using a test set and a logistic regression model. The number of observations that belong to each predicted and actual class combination is shown in the table. According to the model, 115 cases would fall into class 0 and 67 cases would fall into class 1. Actually, there were 65 cases in class 1 and 117 cases in class 0, respectively.The table also includes a number of ratios. The number of observations in each row is indicated by the row total, and the number of observations in each column is indicated by the column total. For each actual class (0 or 1), N / Row Total displays the percentage of predicted class (0 or 1). N/Col Total displays the percentage of the real class (0 or 1). Also, the confusion matrix summary shows the percentage of accurate predictions among all predictions, or accuracy, in this case is 0.9341(93% accurate). Sensitivity, which is 0.9231, is the percentage of true positives among all positive cases. Specificity, which equals 0.9402, is the percentage of true negatives among all negative cases. PPV, which is 0.8955, is the percentage of actual positive results versus all predicted positive cases. NPV, which is 0.9565, is the percentage of actual negative outcomes versus all predicted negative outcomes. The percentage of positive cases in the dataset is called prevalence and it is 0.3571. The detection rate, or 0.3297, is the percentage of true positive cases among all cases. The average of sensitivity and specificity, or 0.9316, is considered to be balanced accuracy.

With high accuracy, balanced accuracy, and high values for sensitivity and specificity, the model appears to perform well . The fact that the PPV is a little lower than the NPV, however, may point to class imbalance or other problems with the dataset. The agreement between the model's predictions and the actual labels is measured using the kappa statistic, which accounts for the possibility of agreement by chance. Significant agreement is indicated by a kappa value of 0.8574.

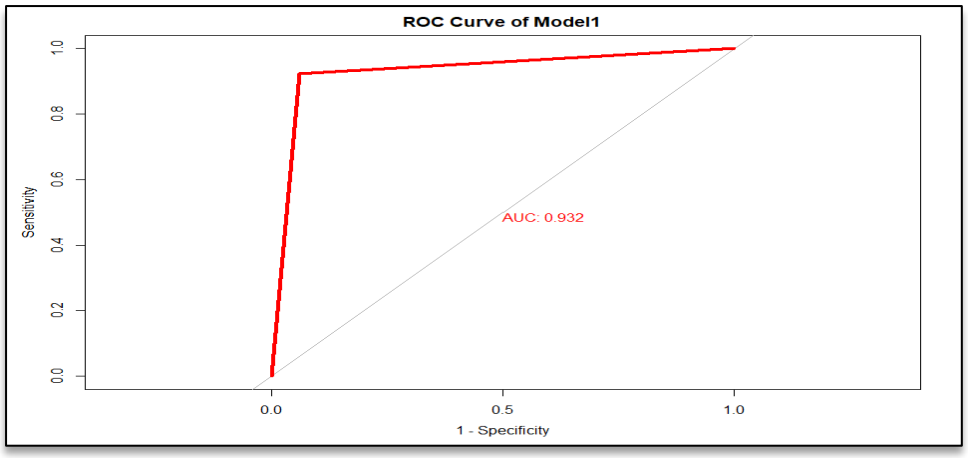


Fig 4.3; Receiver Operating Characteristic Curve of Model 1

The ROC curve which is a graphical representation sensitivity and 1-specificity for various classification thresholds shows that the AUC Value is 0.932. This indicates a good model.

LR Model 2

Model2 shows that all the variable used are statistically significant. A test for multicollinearity was also carried out using VIF which shows no multicollinearity in the model. It indicates it is also a good model.

The summary of the Model

```
Call:
glm(formula = Class ~ A1_Score + A2_Score + A3_Score + A4_Score +
    A5_Score + A6_Score + A7_Score + A8_Score + Gender + Age,
    family = "binomial", data = Train.dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5826  -0.0072   0.0005   0.0519   2.6937

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -25.6020     3.7172  -6.887 5.68e-12 ***
A1_Score      4.1970     0.9241   4.542 5.58e-06 ***
A2_Score      4.1024     0.7722   5.313 1.08e-07 ***
A3_Score      3.3063     0.6546   5.051 4.40e-07 ***
A4_Score      4.8778     0.8216   5.937 2.90e-09 ***
A5_Score      6.1982     0.9728   6.372 1.87e-10 ***
A6_Score      5.3026     0.8007   6.622 3.54e-11 ***
A7_Score      4.3503     0.7910   5.500 3.80e-08 ***
A8_Score      4.1042     0.8759   4.686 2.79e-06 ***
Gender        2.0269     0.6071   3.339 0.000842 ***
Age           2.8073     1.3219   2.124 0.033698 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 862.28  on 621  degrees of freedom
Residual deviance: 119.38  on 611  degrees of freedom
AIC: 141.38

Number of Fisher Scoring iterations: 9
```

Fig 4.4; Logistic model result for LR.Model2

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  182


             | LR.Pred2
  Test[, 20] |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       112 |         5 |       117 |
             |     0.957 |     0.043 |     0.643 |
             |     0.957 |     0.077 |           |
             |     0.615 |     0.027 |           |
-------------|-----------|-----------|-----------|
           1 |         5 |        60 |        65 |
             |     0.077 |     0.923 |     0.357 |
             |     0.043 |     0.923 |           |
             |     0.027 |     0.330 |           |
-------------|-----------|-----------|-----------|
Column Total |       117 |        65 |       182 |
             |     0.643 |     0.357 |           |
-------------|-----------|-----------|-----------|
```

```
> library(caret)
> confusionMatrix(as.factor(LR.Pred2), Test[,20], posit
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 112   5
         1   5  60

               Accuracy : 0.9451
                 95% CI : (0.9013, 0.9733)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8803

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9231
            Specificity : 0.9573
         Pos Pred Value : 0.9231
         Neg Pred Value : 0.9573
             Prevalence : 0.3571
         Detection Rate : 0.3297
   Detection Prevalence : 0.3571
      Balanced Accuracy : 0.9402

       'Positive' Class : 1
```

Fig 4.5; (a) Cross table analysis and, (b) Confusion matrix summary of LR.Model2

The table shows the number of observations for each predicted and actual class in Model 2. The model predicted that 117 cases belong to class 0 and 65 cases belong to class 1. However, the actual number of cases in class 1 and class 0 were 65 and 117, respectively. The table also includes ratios such as accuracy, sensitivity, specificity, PPV, and NPV. The accuracy of the model is 0.9451, which means that it accurately predicted 95% of the cases. The sensitivity of the model is 0.9231, which is the percentage of true positive cases among all positive cases, and the specificity is 0.9573, which is the percentage of true negative cases among all negative cases. The PPV, which is the percentage of actual positive results versus all predicted positive cases, is 0.9231, and the NPV, which is the percentage of actual negative outcomes versus all predicted negative outcomes, is 0.9573. The prevalence of positive cases in the dataset is 0.3571, and the detection rate, which is the percentage of true positive cases among all cases, is 0.3297. The balanced accuracy, which is the average of sensitivity and specificity, is 0.9402.

The model appears to perform better than Model1 with higher accuracy, balanced accuracy, and high values for sensitivity and specificity. However, the lower PPV than NPV may suggest class imbalance or other issues with the dataset. The kappa statistic measures the agreement between the model's predictions and actual labels, accounting for the possibility of chance agreement. The kappa value of 0.8803 indicates significant agreement between the model's predictions and the actual labels. This Model is a better model is a better model as compared to model 1.

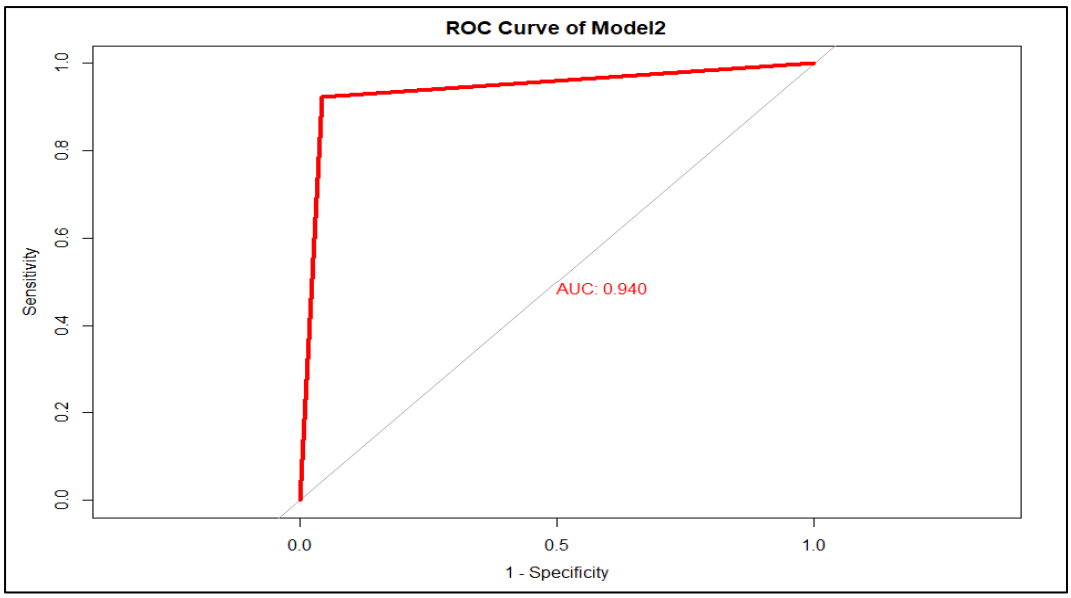


Fig 4.6; Receiver Operating Characteristic Curve of Model 2

The AUC Value is 0.940. This indicates a better model as compared to model 1.

4.1.2 K-NEAREST NEIGBOUR

For KNN model, a seed of "12480" was set. Several k indexes were implemented for the KNN model using random selections of odd numbers from 1 to 21, as well as the rounded square root of all the observations in the dataset. The rounded square root of the data set was 25 (Square root of 622). K=3 and K=25 gave better results. K=25 gave an accuracy of 91.2% while K=3 gave an accuracy of 90.6%.

K=25

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  182


             | knn25
  Test[, 20] |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       101 |        16 |       117 |
             |     0.863 |     0.137 |     0.643 |
             |     1.000 |     0.198 |           |
             |     0.555 |     0.088 |           |
-------------|-----------|-----------|-----------|
           1 |         0 |        65 |        65 |
             |     0.000 |     1.000 |     0.357 |
             |     0.000 |     0.802 |           |
             |     0.000 |     0.357 |           |
-------------|-----------|-----------|-----------|
Column Total |       101 |        81 |       182 |
             |     0.555 |     0.445 |           |
-------------|-----------|-----------|-----------|
```

```
> confusionMatrix(as.factor(knn25), Test[,20
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 101   0
         1  16  65

               Accuracy : 0.9121
                 95% CI : (0.8612, 0.9489)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8185

 Mcnemar's Test P-Value : 0.0001768

            Sensitivity : 1.0000
            Specificity : 0.8632
         Pos Pred Value : 0.8025
         Neg Pred Value : 1.0000
              Precision : 0.8025
                 Recall : 1.0000
                     F1 : 0.8904
             Prevalence : 0.3571
         Detection Rate : 0.3571
   Detection Prevalence : 0.4451
      Balanced Accuracy : 0.9316

       'Positive' Class : 1

>
```

Fig 4.7; (a) Cross table analysis and, (b) Confusion matrix summary of k=25 for KNN algorithm.

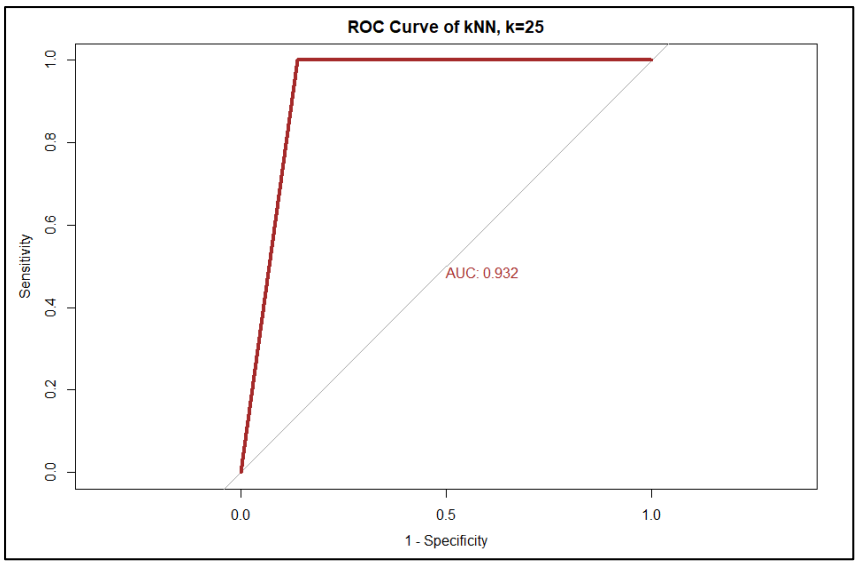


Fig 4.8; Receiver Operating Characteristic Curve for KNN, K= 25

The "0" diagnosis indicates a positive diagnosis, while the "1" diagnosis indicates a negative diagnosis in class diagnosis. The accuracy of the k-NN model is 0.9121 (91.2%), and the true positive rate or sensitivity is 1.0000, while the true negative rate or specificity is 0.8632. The precision or positive predictive value is 0.8025, and the predictive value of the negative is 1.0000. The area under the curve (AUC) is 0.9316. The confusion matrix for the k-NN model includes the following terms: True Positives (TP), with 65 observations correctly identified as positive (class 1) observations, and True Negatives (TN), with 101 observations correctly classified as negative (class 0). False Positives (FP), in this case, are the 16 observations falsely predicted as positive (class 1) but were negative (class 0). There were no False Negatives (FN) in this case, which means that none of the observations were falsely predicted to be positive (class 1) instead of negative (class 0). The AUC value for the k-NN model is 0.932.

K=3

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  182


             | knn3
  Test[, 20] |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       105 |        12 |       117 |
             |     0.897 |     0.103 |     0.643 |
             |     0.955 |     0.167 |           |
             |     0.577 |     0.066 |           |
-------------|-----------|-----------|-----------|
           1 |         5 |        60 |        65 |
             |     0.077 |     0.923 |     0.357 |
             |     0.045 |     0.833 |           |
             |     0.027 |     0.330 |           |
-------------|-----------|-----------|-----------|
Column Total |       110 |        72 |       182 |
             |     0.604 |     0.396 |           |
-------------|-----------|-----------|-----------|
```

```
> confusionMatrix(as.factor(knn3), Test[,20], mod
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 105   5
         1  12  60

               Accuracy : 0.9066
                 95% CI : (0.8547, 0.9446)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : 2.289e-16

                  Kappa : 0.8013

 Mcnemar's Test P-Value : 0.1456

            Sensitivity : 0.9231
            Specificity : 0.8974
         Pos Pred Value : 0.8333
         Neg Pred Value : 0.9545
              Precision : 0.8333
                 Recall : 0.9231
                     F1 : 0.8759
             Prevalence : 0.3571
         Detection Rate : 0.3297
   Detection Prevalence : 0.3956
      Balanced Accuracy : 0.9103

       'Positive' Class : 1

>
```

Fig 4.9; (a) Cross table analysis and, (b) Confusion matrix summary of k=3 for KNN algorithm

The accuracy of the k-NN model was 0.9066 (90.1%). The model's sensitivity was 0.9231, indicating a high true positive rate. The specificity was 0.8974. The precision, or positive predictive value, was

0.8333, while the negative predictive value was 0.9545. The value for the area under the curve was 0.9103. The number of true positives in the confusion matrix was 60, and the number of true negatives was 105. There were 12 false positives (observations predicted as positive when they were actually negative) and 5 false negatives (observations predicted as negative when they were actually positive).

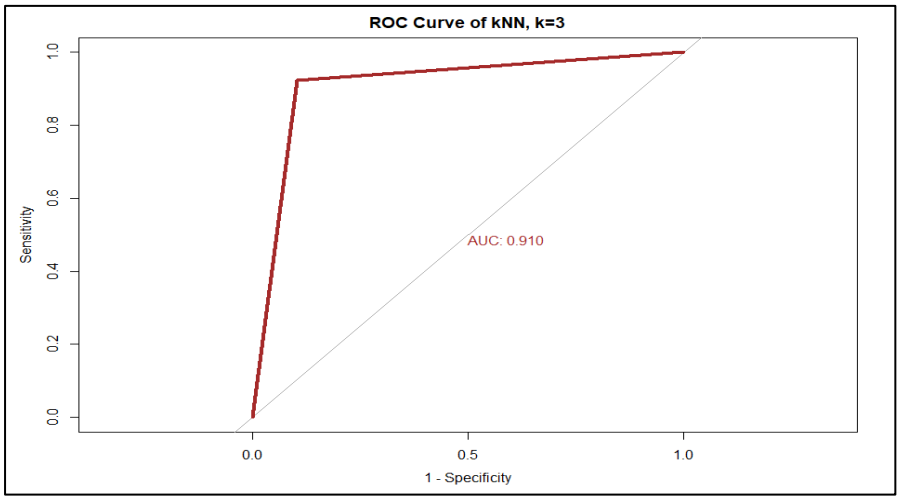


Fig 4.8; Receiver Operating Characteristic Curve for KNN, K= 25

4.1.3 Decision tree

Decision trees are nonlinear classifiers that are evaluated based on their prediction accuracy and sensitivity scores.

Model 1

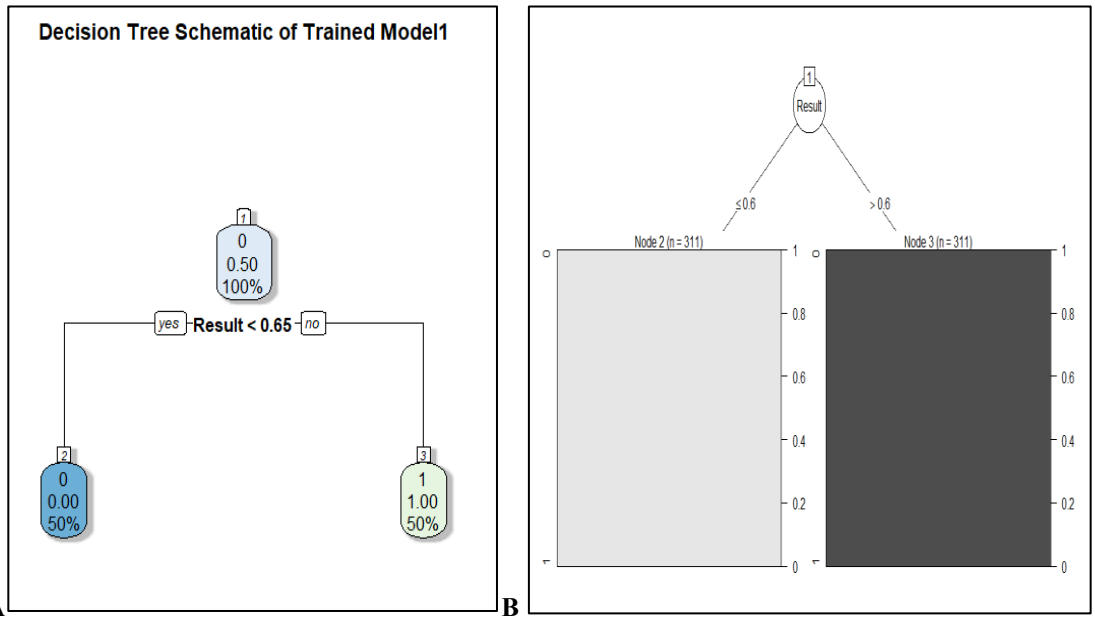
```
Call:
C5.0.default(x = Train.dat[-20], y = Train.dat[, 20])

Classification Tree
Number of samples: 622
Number of predictors: 19

Tree size: 2

Non-standard options: attempt to group attributes
```

```
Call:
C5.0.default(x = Train.dat[-20], y = Train.dat[, 20])


C5.0 [Release 2.07 GPL Edition]         Fri Apr 21 15:20:56 2023
-------------------------------

Class specified by attribute `outcome'

Read 622 cases (20 attributes) from undefined.data

Decision tree:

Result <= 0.6: 0 (311)
Result > 0.6: 1 (311)


Evaluation on training data (622 cases):

          Decision Tree
        ----------------
        Size      Errors

           2    0( 0.0%)   <<


         (a)   (b)    <-classified as
        ----  ----
         311          (a): class 0
               311    (b): class 1


       Attribute usage:

       100.00% Result


Time: 0.0 secs
```

Fig 4.9; Decision tree model

The output shows displays the outcome of fitting a decision tree to the Train.dat data using R's C5.0 algorithm, where the response variable occupies the 20th column of the data. The results indicate that

there are 622 samples and 19 predictors in this classification tree. The root node and a single leaf node are the only two nodes in the tree because its size is 2. This implies that the model might be overly straightforward and might profit from being more intricate.



A B

Fig 4.10; (A) decision tree model training (B)a plot of the boosting model from the decision tree with the corresponding 7 nodes.

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  182


              | DT_pred
  Test[, 20] |         0 |         1 | Row Total |
-------------|-----------|-----------|-----------|
           0 |       117 |         0 |       117 |
             |     1.000 |     0.000 |     0.643 |
             |     0.643 |     0.000 |           |
-------------|-----------|-----------|-----------|
           1 |         0 |        65 |        65 |
             |     0.000 |     1.000 |     0.357 |
             |     0.000 |     0.357 |           |
-------------|-----------|-----------|-----------|
Column Total |       117 |        65 |       182 |
-------------|-----------|-----------|-----------|
```

```
> confusionMatrix(DT_pred, Test[,20], posit
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 117   0
         1   0  65

               Accuracy : 1
                 95% CI : (0.9799, 1)
    No Information Rate : 0.6429
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.3571
         Detection Rate : 0.3571
   Detection Prevalence : 0.3571
      Balanced Accuracy : 1.0000

       'Positive' Class : 1
```

Fig 4.11; (a) Cross table analysis and, (b) Confusion matrix summary of Decision Tree Model.

According to the matrix, there are 65 true positives, 117 true negatives, and no false positives or negatives. This indicates that the model's accuracy, sensitivity, and specificity are all perfect at 1.0. The model's predictions are 100% accurate since both the positive predictive value (PPV) and negative predictive value (NPV) are 1.0. This could be as a result of overfitting of the model. Additional statistics provided below the confusion matrix provide more information regarding the model's performance. The 95% confidence interval for accuracy is (0.9799, 1), which shows the confidence that the model's true accuracy falls within this range. The "No Information Rate" is 0.6429, which is the accuracy that would be obtained by always predicting the most common class (in this case, 0). The kappa statistic calculates the degree of agreement between predicted and true classes, with a value of 1 indicating perfect

agreement. The "Detection Rate" and "Detection Prevalence" values indicate how frequently the model correctly detects the positive class (1).

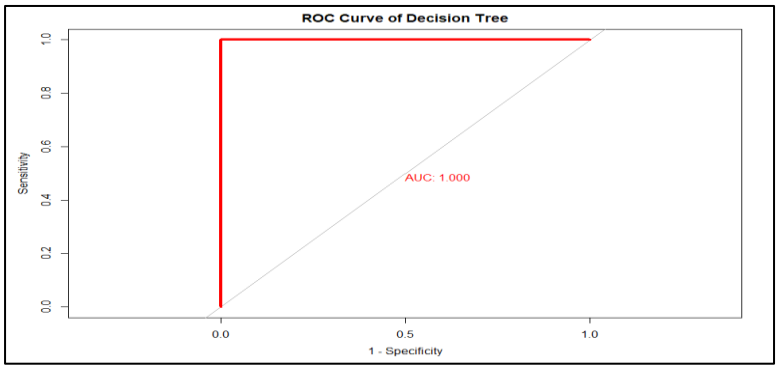


Fig 4.12; Receiver Operating Characteristic Curve of Decision tree.

The above graph shows the evaluation of the performance of a decision tree model on a test dataset using ROC analysis. The AUC is 1 which shows the model is perfect.

Model 2

Model 2 focuses with improving model performance by pruning the tree in order to simplify and/or avoid over-fitting.

```
Call:
C5.0.default(x = Train.dat[-20], y = Train.dat$Class, control = C5.0Control(minCases = 9))

Classification Tree
Number of samples: 622
Number of predictors: 19

Tree size: 2

Non-standard options: attempt to group attributes, minimum number of cases: 9
```

Fig 4.13; Pruned model for decision tree model 2

After building a model to prune Model 1 which seem to overfit, the accuracy and results was the same as Model1.

4.2 DISCUSSION

Decision Tree, Logistic Regression, and K-Nearest Neighbors (KNN) were the classification models used for the project.

Model	TP	FP	TN	FN	Accuracy	Recall	Specificity	F1-Score	Precision	AUC
Logistic Regression	112	5	60	5	94.5	92.3	95.7	93.969255	95.7	94
Decision Tree	117	0	65	0	100	100	100	100	100	100
KNN	101	16	65	0	91.2	100	86.3	92.646269	86.3	93

Table 4.1; summary of all the classification models applied in this study in order of their accuracy

The table presented summarizes the performance metrics of the supervised machine learning algorithms used in this study. Notably, the Decision Tree model achieved exceptional results, with 100% accuracy, recall, precision, specificity, and F1 score, and without any false classifications. Logistic Regression gave the accuracy of 94.5% and recall of 92.3 with the precision of 95.7 while KNN gave an accuracy of 91.2% and recall of 100 with precision of 86.3.

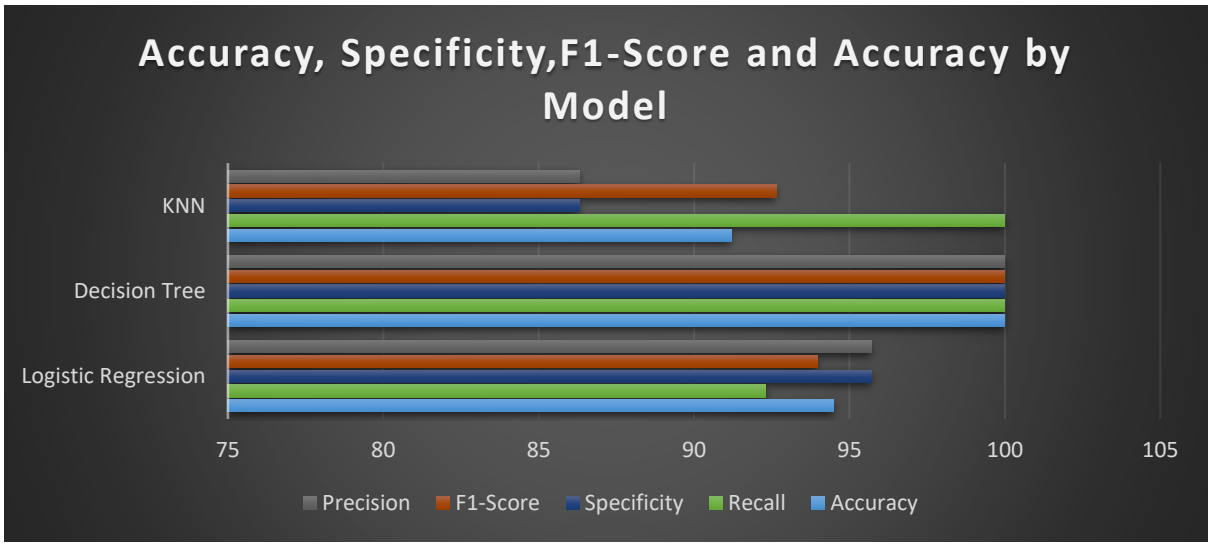


Fig 4.15; Bar chart of all models by their Recall, Accuracy,Specifity and F1-score metrics.

The Decision Tree model achieved perfect accuracy (100%), recall (100%), specificity (100%), and F1-score (100%), implying that it correctly classified all instances. However, the model may have overfitted the training data and may not generalise well to new data. The model was pruned to get a better model but it ended up with the same result which shows it be might actually overfit the training data.

The KNN model had the highest recall (100%) but the lowest accuracy (91.2%) of the models. It correctly identified all true positive cases but had a lower precision of 86.3%, indicating that some false positives were identified.

The Logistic Regression model achieves a high accuracy (94.5%) and AUC (94%). It correctly predicted 112 true positive cases but misclassified 5 negative cases as positive (false positives), yielding a precision of 95.7% and a recall of 92.3%. As evidenced by its high specificity of 95.7%, the model performed well in identifying both positive and negative cases. This shows that this might be the best model for the study.

5.0 CONCLUSION

Conducting autism screening is a crucial initial step in comprehending autistic characteristics and expediting referrals for further assessment in a clinical environment. Assessing behavioural traits associated with Autism Spectrum Disorder (ASD) can be time-taking, especially when symptoms overlap. Regrettably, no rapid and precise diagnostic test to identify ASD has been found and there is also no optimised and comprehensive tool for screening that is specifically developed to detect the early onset of ASD. To take care of this, an ASD prediction models were created which makes use of a small set of behavioural features drawn from diagnosis datasets. We tested three different models and discovered that Logistic Regression had the best accuracy for our dataset.

Thabtah (2017) conducted a study utilizing an ASD screening dataset to perform predictive analysis. The study evaluated Naive Bayes (NB) and Logistic Regression (LR) classifiers in terms of accuracy, sensitivity, and specificity. The results showed that LR had the highest performance with accuracy rates of 99.85%, sensitivity rates of 99.90%, and specificity rates of 99.70% for adults. This study highlights that a machine learning model, specifically Logistic Regression, can effectively assist medical professionals in detecting autism in adults with high accuracy and clinical acceptability

However, the study acknowledges the need for a bigger data to generalize more and incorporate deep learning techniques in future studies. The results obtained from this study can serve as a starting point for other researchers to investigate ASD detection using machine learning models on this or other ASD datasets.

References

American Psychiatric Association (2000). Diagnostic and Statistical Manual of Mental Disorders DSM-IV-TR (Text Revision) (Washington, D.C.: American Psychiatric Association).

Anibal Sólon Heinsfeld, Alexandre Rosa Franco, R. Cameron Craddock, Augusto Buchweitz and Felipe Meneguzzi,(2018) "Identification of autism spectrum disorder using deep learning and the ABIDE dataset," NeuroImage: Clinical, vol. 17, pp. 16-23.

Thabtah Fadi.(2018) "Machine learning in autistic spectrum disorder behavioral research: A review and ways forward". Informatics for Health and Social Care : 1-20

Vaishali, R., and R. Sasikala.(2018) "A machine learning based approach to classify Autism with optimum behaviour sets ". International Journal of Engineering & Technology 7(4): 18.

M. S. Mythili, and AR Mohamed Shanavas. (2014) "A study on Autism spectrum disorders using classification techniques". International Journal of Soft Computing and Engineering (IJSCE), 4: 88-91

Mariam M. Hassan and Hoda M. O. Mokhtar.(2019) "Investigating autism etiology and heterogeneity by decision tree algorithm," Informatics in Medicine Unlocked, vol. 16, 100215.

Fadi Thabtah. (2017). "Autism spectrum disorder screening: machine learning adaptation and DSM-5 fulfillment." In Proceedings of the 1st International Conference on Medical and Health Informatics, pp. 1-6. ACM.

J. A. Kosmicki, V. Sochat, M. Duda, and D. P. Wall. (2015) "Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning." Translational psychiatry, 5(2): e514.

Baihua Li, Arjun Sharma, James Meng, Senthil Purushwalkam, and Emma Gowen. (2017) "Applying machine learning to identify autistic adults using imitation: An exploratory study." PloS one, 12(8): e0182652.