**PREDICTING AUTISM IN CHILDREN**

**A Minor Project Report**

**Submitted in Partial Fulfillment of the**

**Requirement for the award of the degree**

**Of**

**BACHELOR OF TECHNOLOGY (B.Tech)**

In

**Data Science**

By

**Shuchi Rawat 159102129**

**Twinkle Jaswani 159102148**

**Vaishnavee Singh 159102153**

**Vaishnavi Singh 159102154**

Under the Guidance of

**Mr. Venkatesh Gauri Shankar**

**(Assistant Professor)**

**Department of Information Technology**



**School of Computing and Information Technology**

**MANIPAL UNIVERSITY JAIPUR**

**JAIPUR-303007**

**RAJASTHAN, INDIA**

**May/2018**

**CERTIFICATE**

Date-03/05/2018

This is to certify that the project titled **PREDICTING AUTISM IN CHILDREN** is a record of the bonafide work done by **SHUCHI RAWAT** (159102129) submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Information Technology** of Manipal University Jaipur, during the academic year 2018-19.

**Mr. Venkatesh Gauri Shankar**

*Project Guide, Dept. of Information Technology*

*Manipal University Jaipur*

**Dr. Sumit Srivastava**

*HOD, Dept. of Information Technology*

*Manipal University Jaipur*

**CERTIFICATE**

Date-03/05/2018

This is to certify that the project titled **PREDICTING AUTISM IN CHILDREN** is a record of the bonafide work done by **TWINKLE JASWANI** (159102148) submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Information Technology** of Manipal University Jaipur, during the academic year 2018-19.

**Mr. Venkatesh Gauri Shankar**

*Project Guide, Dept. of Information Technology*

*Manipal University Jaipur*

**Dr. Sumit Srivastava**

*HOD, Dept. of Information Technology*

*Manipal University Jaipur*

**CERTIFICATE**

Date-03/05/2018

This is to certify that the project titled **PREDICTING AUTISM IN CHILDREN** is a record of the bonafide work done by **VAISHNAVEE SINGH** (159102153) submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Information Technology** of Manipal University Jaipur, during the academic year 2018-19.

**Mr. Venkatesh Gauri Shankar**

*Project Guide, Dept. of Information Technology*

*Manipal University Jaipur*

**Dr. Sumit Srivastava**

*HOD, Dept. of Information Technology*

*Manipal University Jaipur*

**CERTIFICATE**

Date-03/05/2018

This is to certify that the project titled **PREDICTING AUTISM IN CHILDREN** is a record of the bonafide work done by **VAISHNAVI SINGH** (159102154) submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **Information Technology** of Manipal University Jaipur, during the academic year 2018-19.

**Mr. Venkatesh Gauri Shankar**

*Project Guide, Dept. of Information Technology*

*Manipal University Jaipur*

**Dr. Sumit Srivastava**

*HOD, Dept. of Information Technology*

*Manipal University Jaipur*

**ABSTRACT**

Data Analytics and Machine learning in healthcare are one of the most emerging and needed fields in current time. This project aims at performing research and implementation of big data and machine learning techniques on the data related to the patients suffering from the disease called Autism. Autism is a neural disorder disease characterized by impaired social communication, verbal and non-verbal interaction, restrictive and repetitive behavior.

Autism is majorly noticed in children under or about the age of two years. Waiting times for an ASD(Autism Spectrum disorder) diagnosis are lengthy and procedures are not cost effective. This reveals an urgent need for the development of easily implemented and effective screening methods. Therefore, a time-efficient and accessible ASD screening is imminent to help health professionals and inform individuals whether they should pursue formal clinical diagnosis. . Hence, it is very important to have such data which contains details of patients including their symptoms, lab test data, history, vaccination details etc. which gives specific details of patients and their history.

The project firstly aims at training the data model with the set of training data and then testing and evaluating the data model using the test data for the diagnoses of autism. In this way, it should be a research and solution for implementing machine learning to detect and diagnose autism.

Once diagnosed, this project also aims on the proper analysis of the Autistic traits using behavioral tests. A new dataset related to autism screening of children to be utilized for further analysis especially in determining influential autistic traits by calculating a score and improving the classification of ASD cases. In this behavioral test, we record ten behavioral features (ex- how the child responds when you call his/her name) plus ten individuals characteristics (ex- if the child was born with jaundice) that have proved to be effective in detecting the ASD cases from controls in behavior science.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
|  |  |  |
| 3.1 | Architecture of proposed solution | 12 |
| 4.1 | A subset of the Raw Dataset used | 13 |
| 4.2 | Data after importing into weka | 15 |
| 4.3 | Words after applying AttributeSelectionFilter | 16 |
| 4.4 | (Value, Frequency) distribution of some words | 17 |
| 4.5 | StringToWordVector Settings | 18 |
| 4.6 | AttributeSelectionFilter Settings | 19 |
| 4.7 | Description of the second dataset used | 20 |
| 4.8  4.9  4.10  4.11  4.12  4.13  4.14  4.15  5.1  5.2  5.3 | (Value, Frequency) Distribution of features in second dataset  Working Of Support Vector Machines  Loading Necessary Libraries  Loading the Pre-Processed Data  Code Snippet for Label Encoding  Code Snippet for Dividing the data into training and testing set  Code Snippet for Execution of the SVM classifier  Code Snippet for visualization of algorithm efficiency  Results of the SVM classifier  Visualization of the Confusion Matrix  Results of the Autism Screening Score Calculator | 20  22  23  23  23  24  24  25  26  27  28 |

**CONTENTS** **Page no.**

Abstract 6

List of Figures 7

**Chapter 1 INTRODUCTION 9**

**1.1** Motivation 9

**1.2** Problem Statement 9

**Chapter 2 BACKGROUND OVERVIEW 11**

**2.1** Conceptual Overview 11

**2.2** Technologies Used 11

**Chapter 3 METHODOLOGY 12**

**Chapter 4 IMPLEMENTATION 13**

**4.1**  Implementation Using Weka 13

**4.1.1** Data Preprocessing 13

**4.1.1.1** Data Cleaning 13

**4.1.1.2** Data Sampling 14

**4.1.2** Feature Extraction 14

**4.1.2.1** StringToWordVector 14

**4.1.2.1** AttributeSelectionFilter 14

**4.2** Identification of Autistic Traits 18

**4.3** Executing the Classifier Model 21

**4.3.1** About the Classifier 21

**4.3.2** Implementation of the Classifier 23

**Chapter 5 RESULT AND CONCLUSION 26**

**5.1** Results of the Classifier Model 26 **5.2** Results of the Autism Screening Score Calculator: 27

**5.3** Conclusion 28

**REFERENCES 29**

**1. INTRODUCTION**

With the emergence of technology and science in every possible field, there has been a need for automating the processes so as to make them fast and efficient. Health care is one of those industries which are very complex in terms of diagnosis and processes involved with human health issues.

This project focuses on predicting Autism, the data sets are related to the patients details in the form of text containing the detailed explanation of the symptoms from which they are suffering. The data sets contain details of both, patients with Autism and without Autism so as to train with both kinds of instances

***1.1 Motivation***

The motivation behind the project is to have a rapid detection of risk of autism. There are a lot of approaches which are used for diagnosing autism which have a high validity of diagnosing the disease but the disadvantage is that it is very time consuming. This can result in high delays in reaching to a decision.

The downside of the existing exams is their length or the complexity because of which they take a lot of time. They also require clinical facility administered by the trained professionals. These factors cause a lot of delays in diagnosis process. Also because of such lengthy and complex procedure, the diagnosis cannot be provided to all the population which needs treatment. Hence this results in unequal and inconsistent distribution or coverage. The clinical facilities and trained professionals tend to be available more in major cities. They are overall quite less than the population which needs treatment. Due to lack of resources and time constraints, initial diagnostic screenings do not get conducted consistently. It can be so severe that families might have to wait as long as 13 months from initial screening to clinical diagnosis. There is estimation that 27% of the cases remain undiagnosed until the age of 8 years. In the US, the average age of autism diagnosis is above 4 years.

***1.2 Problem Statement***

While doing study about the diagnosis of autism, it was clear that there is a need of something fast and accurate so as to deliver the results of diagnosis on time and then the related therapies and treatment can begin efficiently. If the diagnosis is delayed, it is quite obvious that the treatment therapies will also get delayed. The treatment therapies consist of speech delivery and behavioral therapies which are quite significant for improvements if delivered under the right age, earlier in life. If delivered later than the particular age, its impact does not remain as beneficial as before.

To contribute towards the problem, search for the reasonable datasets is a very important step. After searching and finalizing the dataset, text classification was chosen to perform machine learning techniques and steps. Text classification is a kind of problem in which sentences are processed and then classified under the labels or classes.

The dataset used for this project is the collection of data from “Vaccine Adverse Event Reporting System” (VAERS) . The raw dataset had been downloaded from their government website [6]. Every data instance had to be provided with certain label or class (Positive/Negative) so as to achieve text classification on the data. The main attraction towards the dataset was that it contained the description of the symptoms (symptom\_text) each patient was suffering from. Also it contained other useful attributes such as lab\_data, other\_medications, condition\_history, prior\_vaccination which seemed to be useful in performing text classification towards prediction of autism. The dataset and its processing are explained in the further sections.

**2. BACKGROUND OVERVIEW**

Some research papers suggested that the current approaches which are used for diagnosing autism have a high validity of diagnosing the disease but the disadvantage is that it is very time consuming. This can result in high delays in reaching to a decision. They focused on a relatively small set of children with and without autism. The algorithm or method which is used currently to diagnose autism is “gold-standard Autism Diagnostic Observation Schedule-Generic” (ADOS-G). By using machine learning to derive a classifier, they were able to reduce the length by 72% compared to ADOS-G.

Because of the nature of the disease, Autism is primarily diagnosed through behavioral evaluations and to achieve the measure of impairments three core developmental domains have been designed:

1. Language and Communication

2. Reciprocal Social Interactions

3. Restricted and Repetitive behaviors

The downside of ADOS exams is their length or the complexity because of which they take a lot of time. They also require clinical facility administered by the trained professionals. These factors cause a lot of delays in diagnosis process. Also because of such lengthy and complex procedure, the diagnosis cannot be provided to all the population which needs treatment. Hence this results in unequal and inconsistent distribution or coverage. The clinical facilities and trained professionals tend to be available more in major cities. They are overall 10 quite less than the population which needs treatment.

***2.1 Conceptual Overview***

This project focuses on predicting Autism in children, the dataset used in this project is related to the children, details in the form of text containing the explanation of the symptoms from which they are suffering. The data sets contain details of both, children with Autism and without Autism so as to train with both kinds of instances. There are various stages in the project such as data pre-processing, data filtering, feature extraction, prediction (evaluation), testing the accuracy.

***2.2 Technologies Involved***

In this project, two open source frameworks were used: Weka and Apache Spark. Weka is good to understand the concepts of machine learning as it is simpler than Spark. But when it comes to complex, large-scale data processing and predictive modeling, Spark is faster. Also for cluster computing, Spark becomes quite compatible and scalable.

**3. METHODOLOGY**

The architecture of the project can be described by considering the following process flow:

1. Collecting the raw datasets related to both Autism and Non-Autism cases
2. Performing the data pre-processing on the raw datasets. Preprocessing includes data cleaning and data sampling.
3. Loading the data over the usable storage.
4. Performing data operations towards feature extraction on the preprocessed data. Feature extraction includes several steps.
5. Training and tuning the classifier to achieve high accuracy.
6. Testing or evaluating the classifier or predictive model.

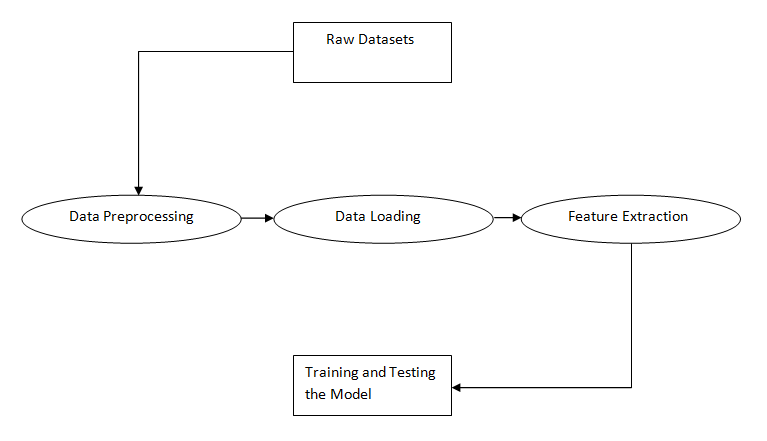


Figure 3.1: Architecture of the proposed solution

**4. IMPLENTATION**

***4.1. Implementation using Weka:***

Weka is open source software. It is a collection of machine learning algorithms used for data mining. The machine learning algorithms can be applied on the datasets directly. Weka contains tools and methods for data preprocessing and predictive modeling. New machine learning schemes can also be developed using Weka .

***4.1.1 Data Preprocessing***

Data preprocessing is one of the most important steps. When we have the raw data, we have to transform it in such a way that it can be used constructively towards accomplishing our task. Often the raw data is not ready to use directly towards the application or towards the sub-processes we want to design. For every machine learning API, platform or framework, the preprocessing of data can vary. The libraries and their usage are different for different frameworks. Data preprocessing includes data cleaning and data sampling.

***4.1.1.1 Data Cleaning***

There were a lot of attributes in the data set and not all were important for training the classifier. For text classification, attribute “symptom\_text” has been considered. There is one more attribute called “vaers\_id” which is a 16 unique id provided to every patient. “symptom\_text” contains description of the state of patients in the form of sentences.

A subset of the raw dataset we obtained can be seen in the Figure 4.1.

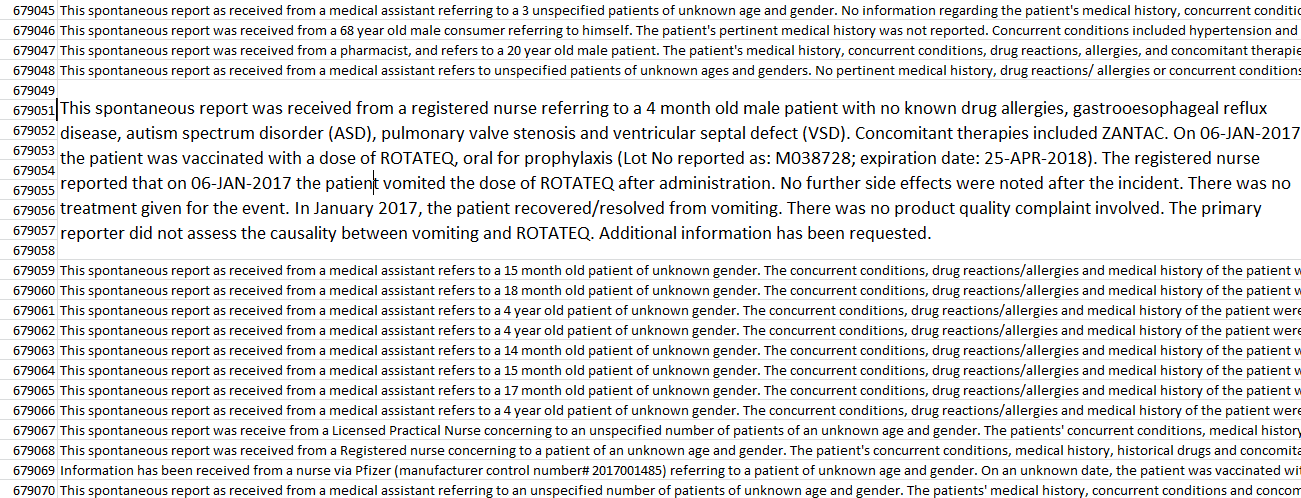


Figure 4.1: A subset of the Raw Dataset used

There are several csv files for different years from 2000 to 2017. First of all, we separated the dataset into two parts, one with positive (with autism) instances and other with negative (without autism) instances. For Weka, the instances of “symptom\_text” had to be converted into separate text files such that each text file contains the symptom\_text of one patient. To execute this, a python script was written to automate the process. For example, the vaers\_id of a particular instance is 22543. Python script created text file with the name as “pos22543” if it was positive instance and “neg22543” if it was negative instance. The text files with positive instances were stored in directory named “Pos” and text files with negative instances were stored in directory named “Neg”.

After getting two separate directories for positive and negative text files, next aim was to create an arff (Attribute-Relation File Format) file which is efficiently supported by Weka. Arff file is less memory intensive and faster. For converting text files into arff file, TextDirectoryLoader class is used. TextDirectoryLoader loads the directory into an arff file. The exact command used on Weka command line interface is:

*> java weka.core.converters.TextDirectoryLoader –dir “Path of source directory” > “Path of destination directory where final arff file is to be kept”*

In this way, arff file containing the data was produced. Weka automatically adds an attribute called “class” and provides a value of class to each instance based on the directory in which they have been stored. For example, names of directories here are Pos and Neg, so the values of class attribute given to positive instances are “Pos” and to negative instances are “Neg”.

***4.1.1.2 Data Sampling***

To ultimately achieve predictive modeling, the data must have to be divided into training set and test set. Ideally, the training set consists of 66% of the whole data and test set consists of 34%. This is done because model should get enough data (two-third of data) to get trained and after that there should be completely new data (Rest one third of data) with which model must be tested for calculating its accuracy towards prediction. But first, For the sake of convenience , we are first implementing feature extraction and further implementation of the algorithms on a small subset of our data. Data Sampling in Weka is quite straightforward because when the classifier is executed, it gives an option of splitting the data in training and test set. So 66% can be chosen for training data.

***4.1.2 Feature Extraction***

After importing data (arff) in Weka, the data can be visualized using Figure 4.2.

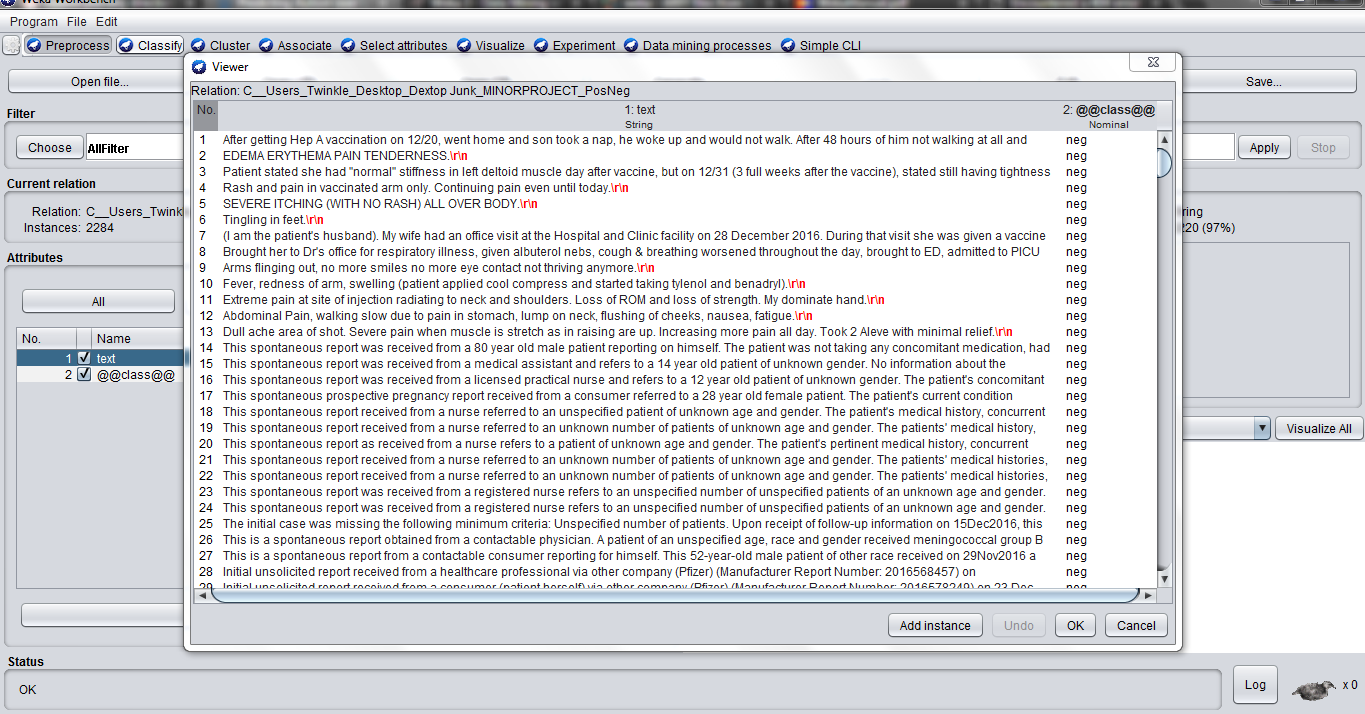


Figure 4.2: Data after importing into Weka

Feature extraction is first step towards the data modeling. Features can be understood as the most important words in the data set which can be considered as deciding factor behind the predictions, For example, features in this project can be the most prominent words in the “symptom\_text”. Features play a significant and vital role in further classification algorithms. There are several ranking algorithms which can be used to select features out of the text.

***4.1.2.1 StringToWordVector:***

String is a datatype which machine learning classifiers usually cannot process. Hence using StringToWordVector, random text has to be transformed into document vectors. Table is formed in which Text Documents are rows (Document vectors), words as columns and values as numbers. In this way, numbers can represent text. StringToWordVector converts String data into numeric or nominal data which learning algorithms can process. There are several settings or parameters which are needed to be set according to the requirements of the problem. Parameters are explained as follows:

***wordsToKeep***This is the total number of words desired to be considered in the data. ***outputWordsCount*** If this setting is put true, then values in the table will be the number of times that word occurs in that document.

***doNotOperateOnPerClassBasis*** If this setting is put true, then the number of “wordsToKeep” is considered in total irrespective of class (Pos/Neg) otherwise the number of “wordsToKeep” is considered on per class basis.

***IDFTransform*** *and* ***TFTransform*** TF-IDF is a way to find words and documents that are strongly related. Higher value of TF-IDF score means the word is important for that document.

***normalizeDocLength*** Normalization refers to measurements taken on different scales and re-measuring them on a common scale.

***Stemmer*** Stemming tries to use words better by breaking them into a smaller form called stem. To perform stemming, it can inspect both sides of a word to try to remove letters from either of side and typically it removes from the suffix side. In Weka, default is NullStemmer which doesn‟t do any stemming. Lovins Stemmers and Porter Stemmers are two popular types. The stem is not necessarily a linguistically valid word. E.g. the word “have” may lose “e” and become “hav”.

***Stop words*** Stop words may tend to be irrelevant for classification. “the”, “is”, “at”, 23 “on” etc. are stop words. StringToWordVector by default uses an English language stop words list in Weka.

***Tokenizer*** Tokenizer algorithms have different ways of splitting up the text. They split into tokens. Alphabetic Tokenizer With the default word tokenizer, some words contain signs like @, &, ~, -, --, whereas using Alphabetic Tokenizer, all of those get eliminated and every token is 100% letters in the alphabet. Alphabetic Tokenizer would have unigrams only, unlike the Ngram Tokenizer where there are words and phrases both.

***minTermFrequency*** minTermFrequency is the minimum number for which any word has to appear to be considered as an attribute. This can be set as per the requirement of the data. ***lowerCaseTokens***If lowerCaseTokens setting is off, both lower case and upper case words are considered different attributes.

***4.1.2.2 AttributeSelectionFilter***

The AttributeSelectionFilter often compliments the StringToWordVector as high quality input data is created. StringToWordVector changed all the symptom\_text and their words into document vectors. AttributeSelection is different. It does not change characters into different numbers. It ranks the attributes and further improves the input data. Under the settings of AttributeSelectionFilter, Evaluator and Search can be chosen which are explained as follows:

***Evaluator – InfoGainAttribueEval*** :Evaluator is the judge for judging the predictive quality of the attribute.

***Search – Ranker*** :Search consults the Judge (Evaluator) to make the final decision to accept or reject the attribute.

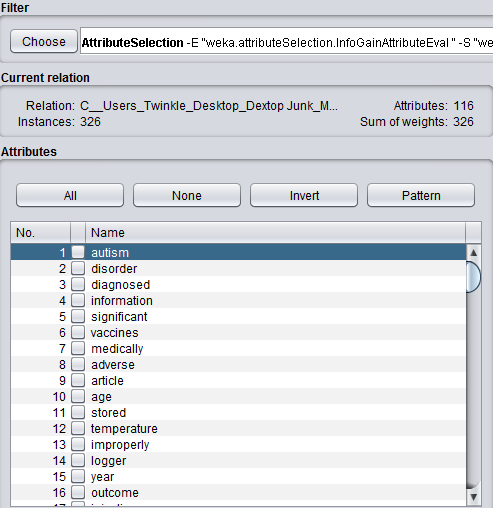
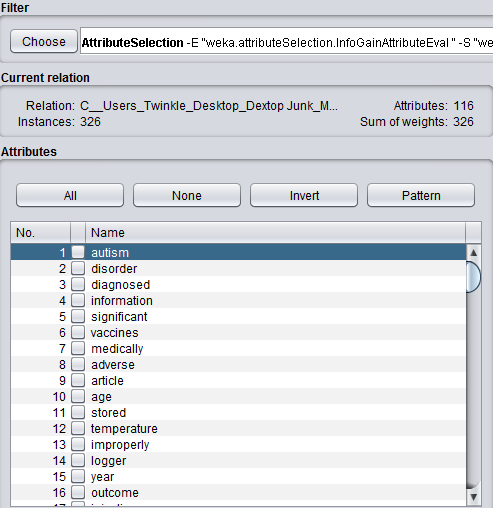


Figure 4.3: Words after applying AttributeSelectionFilter

From figure 4.3, we can see that “autism” is ranked # 1. “disorder” is ranked # 2 and so on.



Figure 4.4: (Value, Frequency) distribution of some words

The settings applied to the data in this project are as shown below:

The settings for StringToWordVector applied to the data in this project are as shown in Figure 4.5.

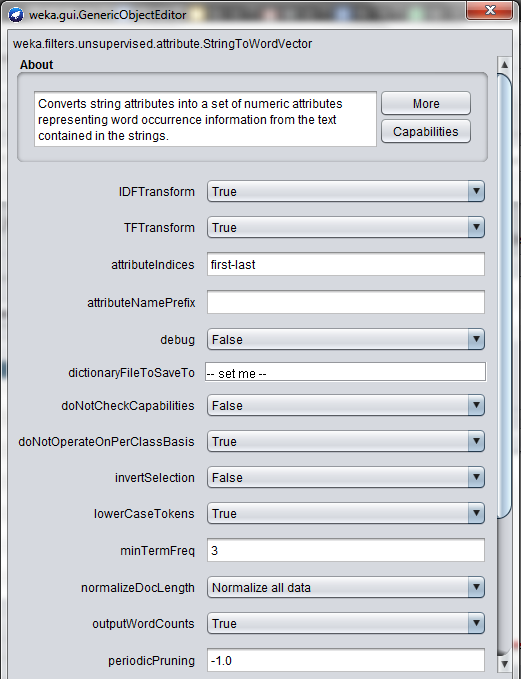


Figure 4.5: (a) :StringToWordVector Settings

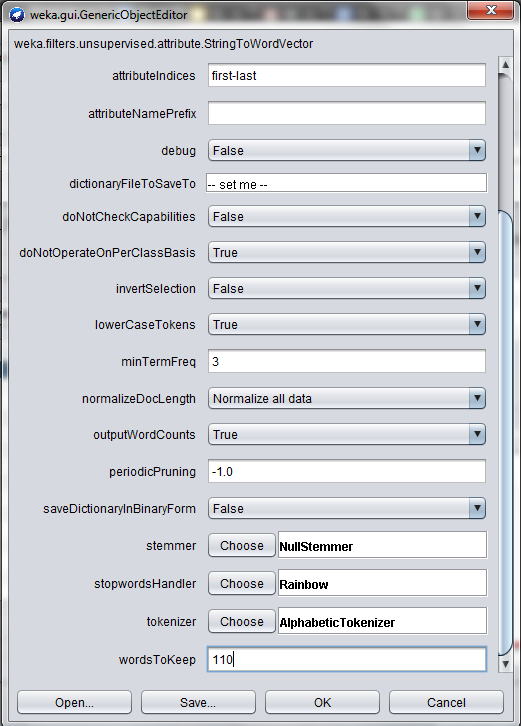


Figure 4.5: (b) StringToWordVector Settings

The settings for AttributeSelectionFilter applied to the data in this project are as shown in Figure 4.6.

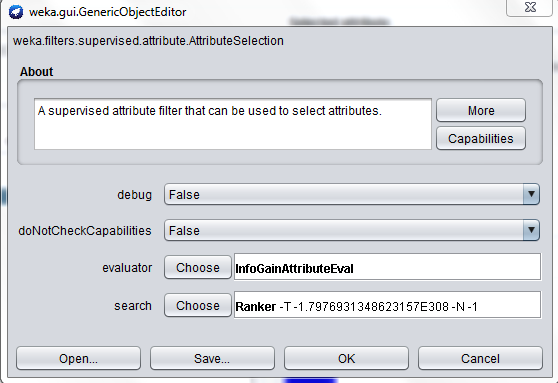


Figure 4.6: AttributeSelectionFilter Settings

***4.2 Identification of Autistic Traits***

Once diagnosed, this project also aims on the proper analysis of the Autistic traits using behavioral tests. A new dataset related to autism screening of children to be utilized for further analysis especially in determining influential autistic traits by calculating a score and improving the classification of ASD cases. In this behavioral test, we record ten behavioral features (ex- how the child responds when you call his/her name) plus ten individuals characteristics (ex- if the child was born with jaundice) that have proved to be effective in detecting the ASD cases from controls in behavior science.

The dataset use for the calculation of the screening source and the identification of the autistic traits is described in Figure 4.7.

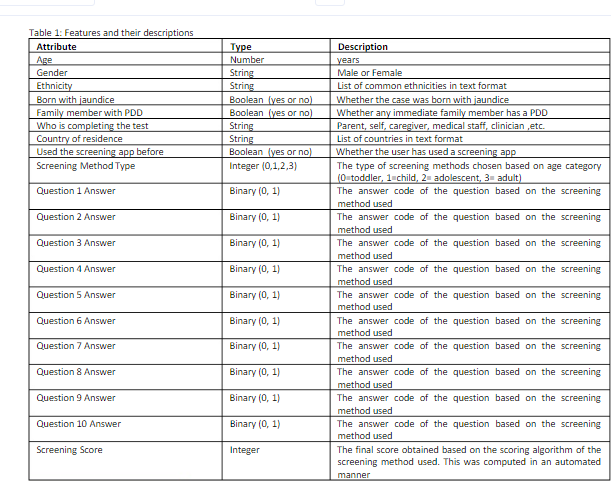


Figure 4.7: Description of the second dataset used

Number of Instances in the second dataset (records in your data set): 292

Number of Attributes in the second dataset(fields within each record): 21

The Questions mentioned in the description above would somewhat be like:

1) If the child looks up when called his/her name,

2) How easy it is to get eye contact with the child, etc.

The Value Frequency Charts of the features in this dataset can be visualised using Figure 4.8.

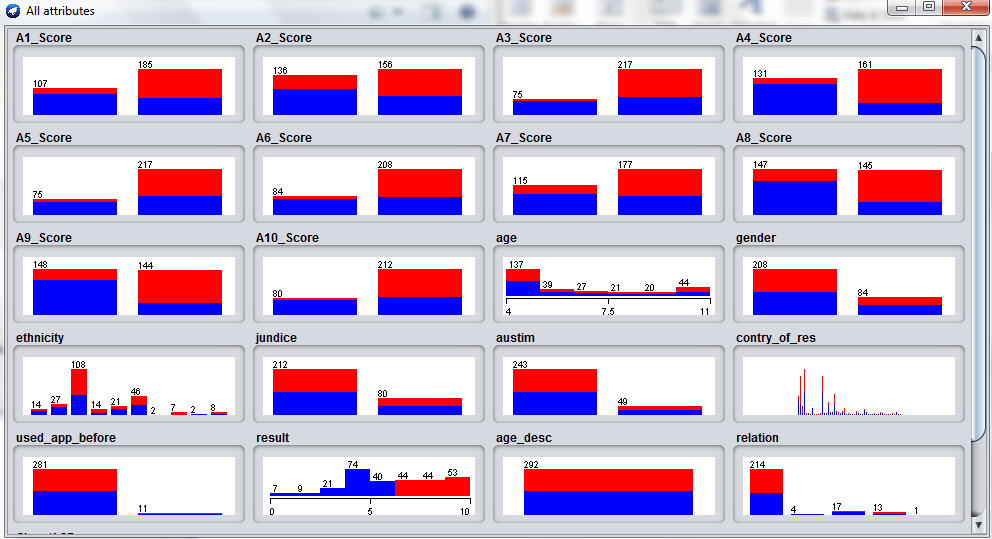


Figure 4.8: (Value, Frequency) Distribution of features in the second dataset

***4.3 Executing the classifier model***

After performing data preprocessing and feature extraction, final set of features have been retrieved. Now in this stage we can execute the classifier of our wish. For this project, we chose Support Vector Machine (SVM).

***4.3.1 About the Classifier***

Specific advantage of the SVM method is its ability to handle high dimensional datasets because of the kernel trick. Support vector machines focus only on the points that are the most difficult to tell apart, whereas other classifiers pay attention to all of the points. The intuition behind the support vector machine approach is that if a classifier is good at the most challenging comparisons (the points in the classes that are closest to each other) then the classifier will be even better at the easy comparisons (comparing points in the classes that are far away from each other).

Unlike other classifiers, the support vector machine is explicitly told to find the best separating line. The support vector machine searches for the closest points (Figure 4.9), which it calls the "support vectors".

Once it has found the closest points, the SVM draws a line connecting them (see the line labeled 'w' in Figure 4.9). It draws this connecting line by doing vector subtraction (point A - point B). The support vector machine then declares the best separating line to be the line that bisects -- and is perpendicular to -- the connecting line.

The support vector machine is better because when you get a new sample (new points), you will have already made a line that keeps B and A as far away from each other as possible, and so it is less likely that one will spillover across the line into the other's territory.

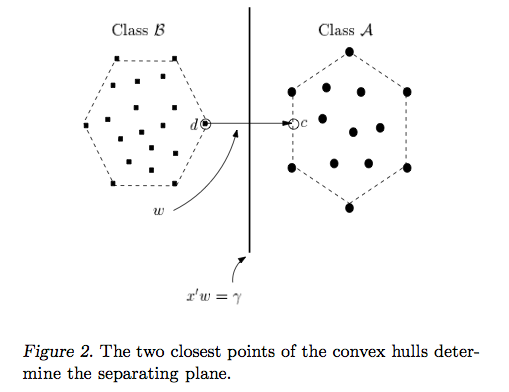


Figure 4.9: Working of Support Vector Machines

***4.3.2 Implementation of the classifier***

First we import all the necessary libraries required to implement our machine learning model and also to visualize the results. The import is shown in Figure 4.10

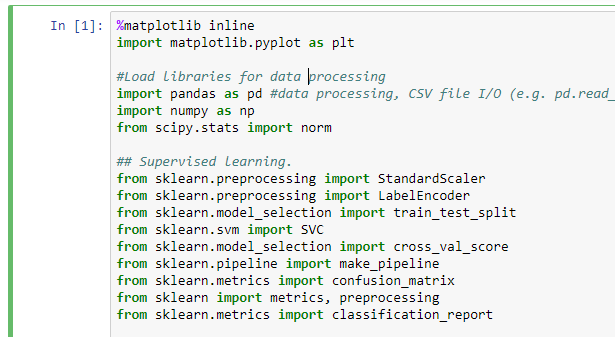


Figure 4.10: Loading Necessary Libraries

Next, we load the data which we obtained after the pre-processing on weka. We changed the pre-processed data into a CSV (Comma Separated Values) file and then used the read\_csv function of pandas to load our data into our ipython notebook. This can be seen in Figure 4.11

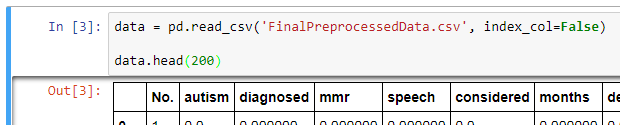


Figure 4.11: Loading the pre-processed data

Next, we encode our class labels into integers in order to implement the classifier. The encoding is done using a Label Encoder from sklearn. The implementation of the Label encoder can be seen in Figure 4.12.

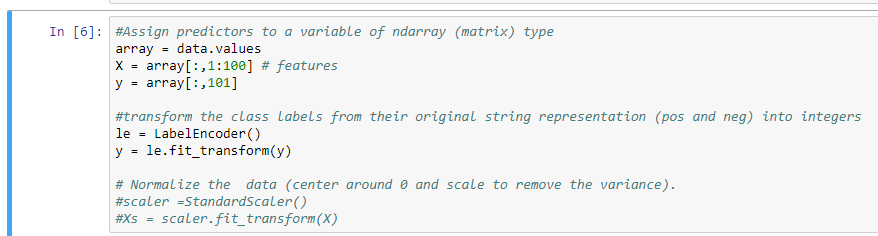


Figure 4.12: Code Snippet for Label Encoding

Our next step would be to divide our data into training and testing sets. Here, we have used 33% of the data as the test set and 67% of the data as the training set. We have used a random variable as our preprocessed data was ordered according to the class labels, i.e., all the tuples with the negative classes were followed by all the tuples with positive classes. The division can be seen in Figure 4.13.

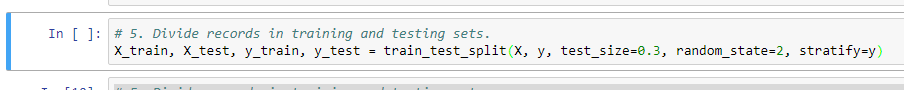


Figure 4.13: Code Snippet for dividing the data into training and testing set

After this step, we are ready to actually execute the Support Vector Machine on the pre-processed data and get the Accuracy. The efficiency of the algorithm is also determined by factors other than accuracy. These factors include Precision, Recall and the Confusion Matrix which tells us about the false positives and false negatives. The implementation of the classifier and the code for the visualization can be seen in Figure 4.14 and Figure 4.15.

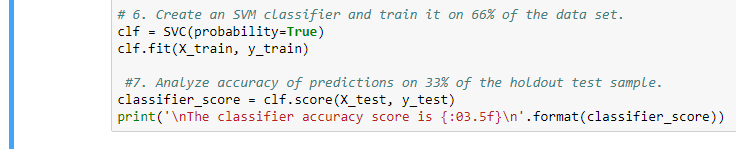


Figure 4.14:Code Snippet for execution of SVM classifier

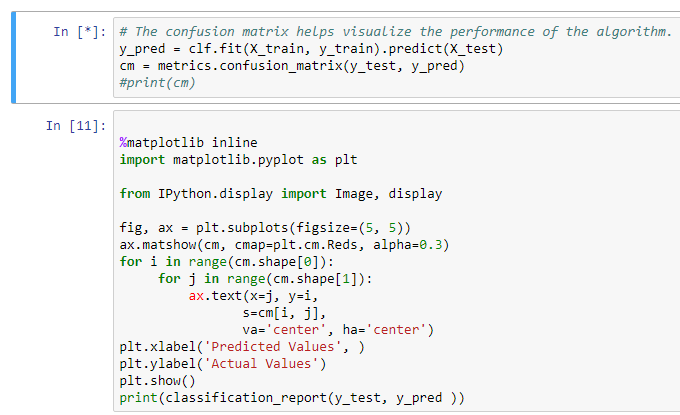


Figure 4.15: Code Snippet for the visualization of algorithm efficiency

**5. RESULTS**

***5.1 Result of the classifier model***

The Result of the classifier model can be visualized in figure 5.1 and figure 5.2.

Figure 5.2 shows the Confusion Matrix generated and also the precision and recall after the execution of the Support Vector Machine.

As shown in Figure 5.1, the accuracy of the classifier is 93.135%.

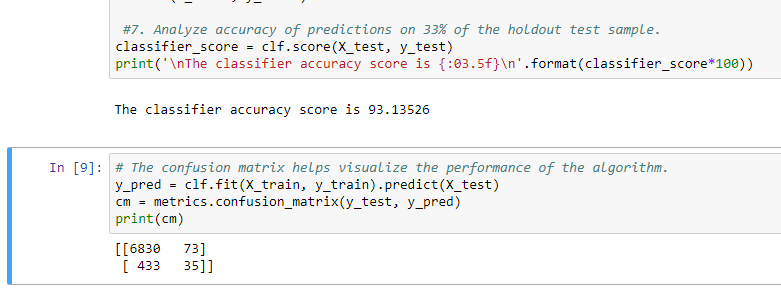


Figure 5.1: Results of the SVM classifier

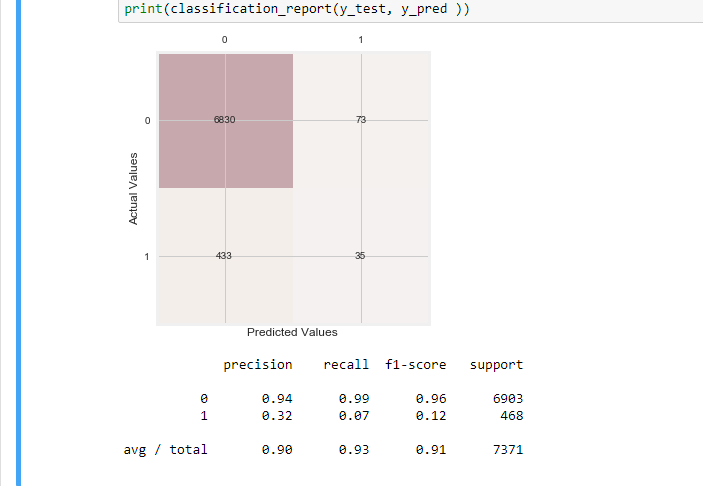


Figure 5.2: Visualization of the Confusion Matrix

After the predictive model is complete, a screening score is to be calculated based on a behavioral test and individual characteristics described in the second dataset. This screening score would help pin-point the autistic traits and will therefore help the patient get a proper set of therapies and treatments.

***5.2 Results of the Autism Screening Score Calculator:***

A program was implemented for the calculation of the Autism Screening Score.

This program can be visualized using Figure 5.3. This part of the projects asks a few questions about the child and generates a score out of 10 based on the answers of those questions. If the Score calculated is less than 5 then that would mean no significant autistic traits were diagnosed in the child but if the score is greater than or equal to 5 then that would mean the a multi-disciplinary assessment of the child is required.

The score calculated also signifies how severe the autistic traits are and what kind of treatment would be best suitable for the child.

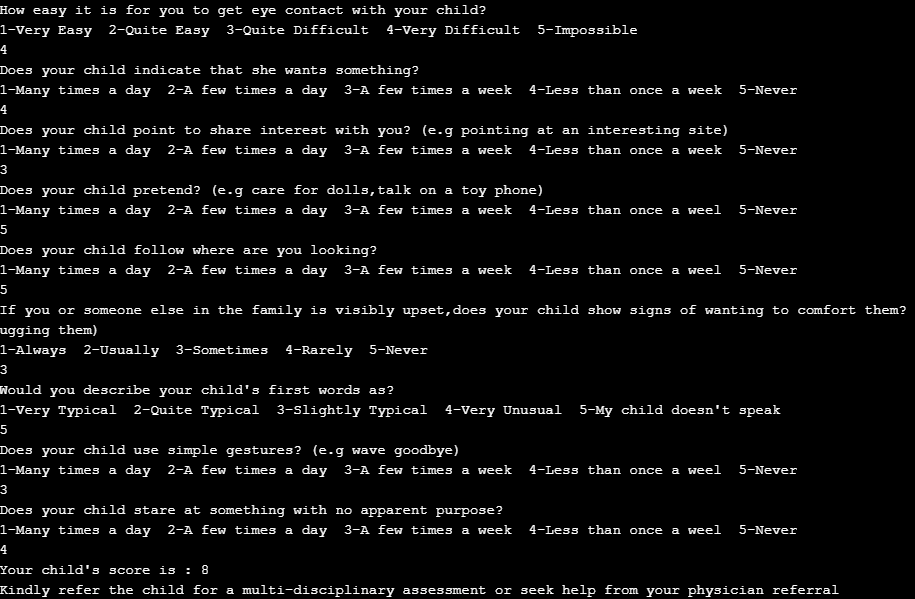


Figure 5.3: Results of the Autism Screening Score Calculator

***5.3 Conclusion***

The project work proposes a new aspect and approach to fasten the procedure of detecting and diagnosing a complex disease called Autism. This approach uses Machine Learning or Predictive Modeling techniques. If this approach gets to be adapted in real life scenario, then doctors can get immensely assisted by technology towards diagnosing this disease efficiently, accurately and in lesser time. In future, several different kinds of datasets can be consumed and utilized towards building data models and perform predictive modeling. The major advantage of such technique is that it has immense potential to get utilized in other areas of medical science or any other field.

**REFERENCES**

*Journal / Conference Papers*

1. Arya, Arpit, "Predicting Autism over Large-Scale Child Dataset" (2015). Master's Projects. 452. http://scholarworks.sjsu.edu/etd\_projects/452.
2. Tabtah, F. “Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment.” (2017). Proceedings of the 1st International Conference on Medical and Health Informatics 2017, pp.1-6. Taichung City, Taiwan, ACM.
3. M Duda,J A Kosmicki, and D P Wall . “Testing the accuracy of an observation-based classifier for rapid detection of autism risk”, Nature Publishing Group .
4. Chaturvedi, V. “The potential of accelerating early detection of Autism through content analysis of YouTube Videos”, Public Library of Science
5. Thabtah, F. (2017). ASDTests. A mobile app for ASD screening. www.asdtests.com [accessed December 20th,2017]

*Web*

[1] Vaccine Adverse Event Reporting System(VAERS), Vaers government website.

[2] Autism Spectrum Disorder, wikipedia

[3] Weka Text Classification, Youtube