Using Machine Learning to Classify Autism Spectrum Disorder.

1. Introduction

Autism Spectrum Disorder (ASD) is a neurodevelopmental condition associated with difficulty in social interaction and communication. Individuals suffering from this condition can be treated successfully if the condition is detected early in childhood. However, diagnosing ASD early in the childhood is very difficult and in many low-income countries, lack of proper equipment & trained physicians make diagnosis even more challenging. Machine learning could be used to diagnose ASD quickly in a more efficient and cost-effective way.

In this report, I am comparing two different machine learning methods for autism spectrum disorder diagnosis. Section two discusses problem formulation. Section three discusses the methods used. Section 4 discusses the results obtained and finally section 5 discusses the conclusion drawn from the results. Section 6 lists the references.

2. Problem Formulation

The data used for training the machine learning classifier is obtained from UCI Machine Learning Repository Autistic Spectrum Disorder Screening Data for Children Data Set [1]. The data was collected by the ASD Test app[2]. Each data point is an individual's answers to the ASD test. The question I try to solve is, given the answers of the ASD test for a child, how accurately will a machine learning classifier be able to diagnose if the child has ASD or not.

The ASD test requires an individual to answer the 10 questions on the presence of ASD-related behavioral traits in the AQ-10-Child[3] questionnaire using binary numbers (0,1). The answers to the questions are added to obtain the ASD Test Score, which is an integer between 0 and 10, in each data point. The ASD test also records the individual's gender, age, and asks, "whether the child was born with jaundice" and "whether the child has a family member with PDD (Pervasive Developmental Disorder)". The age is recorded numerically as floats while the gender is recorded as a binary variable (Male = 1, Female = 0) in each data point. Answers to "whether the child was born with jaundice" and "whether the child has a family member with PDD" are also recorded as binary variable (Yes = 1, No = 0) in each data point.

The features for the machine learning problem are the ASD Test Score, gender, age, "whether the child was born with jaundice" and "whether the child has a family member with PDD" in each data point. The label of each data point will be whether the child has been clinically diagnosed with ASD or not, which will be represented by a binary value (Yes = 1, No = 0).

3. Methods

The data used for training the machine learning classifier is obtained from UCI Machine Learning Repository Autistic Spectrum Disorder Screening Data for Children Data Set[1]. It has 292 data points. Among all the columns I will be using, only the age column has 4 missing values. Since 4 is a small

number in contrast to 292, I decided to drop the 4 rows containing the missing values and ended up with 288 data points. The features and labels, that are recorded as 'yes' or 'no' are converted to a numerical value of either 1 or 0 (Yes = 1, No = 0). The gender column records gender either as 'm' (representing male) or as 'f' (representing female), which are also converted to numerical values of 1 or 0 (m = 1, f = 0). The 'Class/ASD' column is the label, and the remaining columns are candidate features.

The features for the machine learning problem are the ASD Test Score, gender, age, "whether the child was born with jaundice" and "whether the child has a family member with PDD" in each data point. The features are represented by the 'result', 'gender', 'age', 'jaundice' and the 'autism' columns of the dataset respectively. ASD Test Score is often used by physicians to clinically diagnose ASD for which it is chosen as a feature. The rest of the features were chosen because they have proved to be effective in detecting the ASD cases from controls in behavior science. The columns representing the scores in each question (i.e. A1_Score, A2_Score, A10_Score) in the dataset were not chosen because the 'result' column, representing the ASD Test score, is the sum of those scores. The remaining columns in the dataset were not chosen because they were hard to quantify.

For model validation, I decided to use K-fold cross validation. My data set has only 288 data points which is a small number in the machine learning context. If I use a single split of this dataset into training, validation, and test, I might, unfortunately, end up choosing data points that are either outliers or not representative of my dataset's statistical properties. As a result, my model won't be trained properly. So, I used K-fold cross validation where the data set is randomly divided into K subsets, using K-1 subsets for training and the remaining subset for validation. I split my dataset into 20% test set and 80% remaining set[4]. I used the remaining set for the K-fold cross validation with 100 folds.

Support Vector Classifier.

Before choosing a suitable machine learning model, I decided to understand the trends in my dataset beforehand. So, I did a seaborn plot (can be found in the code file) of the dataset. From the seaborn plot, the set of scatterplots in the lowest row clearly show that there exists a linear hyperplane that divides the data between the two distinct labels. Thus, I decided to use support vector classifiers (SVC) (Ch. 3.7 of mlbook.cs.aalto.fi)[5]. SVC work very well with linear classification problems. The algorithm works by creating a line or hyperplane that separates the data into classes, making SVC a perfect fit for my problem.

For the SVC, I used hinge loss (Eq 2.11 of mlbook.cs.aalto.fi)[5] because this function is always used to optimize support vector machines (Ch. 3.7 of mlbook.cs.aalto.fi)[5] in machine learning. The hinge loss conceptually puts the emphasis on the boundary points. Anything farther than the closest points contributes nothing to the loss because of the "hinge" (the max) in the function.

Logistic Regression

For my second method, I decided to use Logistic Regression (Ch. 3.6 of mlbook.cs.aalto.fi)[5], because of two main reasons. Firstly, Logistic Regression is widely applied to solved classification problems related to medical diagnosis[6]. Secondly Logistic Regression uses linear maps to come up with a hyperplane or decision boundary that separates observations that belong to a class from all the other observations that do not belong to that class, which is what my classifier should do, based on the observation of the seaborn plot.

For the Logistics Regression method, I decided to use Logistic Loss (Eq 2.12 of mlbook.cs.aalto.fi)[5] because this loss function is always used for Logistic Regression. For a wrong classification, the logistic loss increases monotonically with increasing confidence and for a right classification the loss decreases with increasing confidence.

4. Results

To evaluate the performance of my classifier, I used the F1-score (Eq 2.21 of mlbook.cs.aalto.fi)[5]. I chose F1-score over accuracy score because it minimizes false negatives and false positives better. I don't want my model to fail to diagnose ASD, nor do I want it to give a false ASD diagnosis to someone who doesn't have ASD. F1-score is between 1 and 0 where 1 represents 100% accuracy.

For my first method, the SVC, I obtained the following results during cross validation with all five features

Validation F1-Score	Training F1-Score
1.0	1.0

For my second method, the Logistic Regression, I obtained the following results during cross validation with all five features

Validation F1-Score	Training F1-Score
1.0	1.0

The results show that both Logistic Regression classifier and SVM classifier can predict if a patient has ASD or not with 100% confidence. This might have happened because the ASD test score is strongly correlated with the label. From the seaborn plot it can be seen that, above a certain value of the ASD score, the label is 1, no matter what the other feature values are. Below that value of the ASD score, the label is 0, no matter what the other feature values are. The same F1-scores also make it difficult to choose the final model. So, I decided to drop ASD test score feature, and retrain my models with four remaining features. More business value comes out of a classifier that is able to work without the test score because if you have to do the test anyway, you probably don't need the classifier and the test score is definitely the most expensive feature to obtain.

So, after training the SVC with the remaining four features, the following results were obtained

Validation F1-Score	Training F1-Score
0.4129999999999999	0.5470319862042549

After training the Logistic Regression model with the remaining four features, the following results were obtained

Validation F1-Score	Training F1-Score
0.4383333333333333	0.5689010330056967

The Training & Validation F-1 score is higher for the Logistic Regression model when four features are used. This means that the Logistic Regression is a better model in contrast to SVM for this machine learning problem. So, I fitted the Logistic Regression model with the remaining data set and used the fitted model to obtain the test F1-score. With 5 features the F1- score for test data set was 1.0. With 4 features F1- score for test data set was 0.4067796610169491.

5. Conclusion

From the results, it can be seen that when all of the features are used, the classifier works with 100% accuracy. However, as mentioned before, a classifier that doesn't need the ASD test score will be more worthwhile, because in many underdeveloped countries it might be very difficult to obtain the ASD test score. When the models are trained and validated without the ASD test score, the F1 scores are around 0.5, which is pretty low and there is definitely room for improvement.

Firstly, the validation F1-score is always lower than the training F1-score which indicates that overfitting is happening. The main reason of the overfitting is the small number of data points in my data set. So, to prevent this, a larger data set is need. Also, without the ASD score, the four remaining features that I used at the end are not enough to predict if a child has ASD or not. Thus, more features that are correlated with ASD and can be collected easily should also be used to train the models. With a higher number of features and data points a complicated model like artificial neural networks could be used to diagnose ASD with higher accuracy.

6. References

1. UCI Machine Learning Repository: Autistic Spectrum Disorder Screening Data for Children Data Set. https://archive.ics.uci.edu/ml/datasets/Autistic+Spectrum+Disorder+Screening+Data+for+Children++ .
2. ASD. https://www.asdtests.com/.
3. AQ10-Child.pdf.
4. Split Training and Testing Data Sets in Python - AskPython. https://www.askpython.com/python/examples/split-data-training-and-testing-set (2020).
5. Jung, A. *Working Draft for the Textbook 'Machine Learning. The Basics'*. (2022).
6. Logistic regression. *Wikipedia* (2022).