Autism spectrum disorder (ASD) classification from resting state fmri data

course project for CG501 Computation and Cognition

Ashish V

May 1, 2024

Contents

1 Introduction

Autism spectrum disorders (ASD) represent a formidable challenge for psychiatry and neuroscience because of their high prevalence, life-long nature, complexity and heterogeneity. Autism Brain Imaging Data Exchange (ABIDE)(Di Martino, 2014) – a grassroots consortium aggregating and openly sharing existing resting-state functional magnetic resonance imaging (R-fMRI) datasets with corresponding structural MRI and phenotypic information from 487 individuals with ASD and 557 age-matched typical controls. ABIDE II has 1004 samples and each sample has 1446 features.
Here in this project, I use different machine learning models to correctly classify ASD from controls.

In the dataset, all the feature columns's title starts with 'fs' which denotes 'fat suppression'. Adipose tissue is predominantly composed of long-chain triglycerides and free fatty acids, whose prinicipal resonances lie between $\delta = 0.9$ and 1.4 ppm. Signal from extraneous fat can "bleed into" and contaminate the spectra from within the defined volume of interest. This technique uses the difference in resonance frequency between fat and water by means of frequency selective pulses (CHESS).

2 Literature review

Most machine learning practitioners do not use deep learning or neural networks on tabular data. It's not clear why deep learning models do not do well on tabular data while it has enabled tremendous progress for learning on image, language, or even audio datasets (Grinsztajn, 2022). The literature review is on the machine learning models and algorithms that I have used in my classification problem, along with other considerations in a machine learning model, like overfitting and underfitting.

`Naive Bayes` is a straightforward method for building classifiers: these are tools that assign categories to instances, which has a collection of feature values, with class labels attached to it. It is a supervised learning algorithm. supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset's features. This group of algorithms, known as naive Bayes classifiers, operates under a shared principle: they all assume that a specific feature's value is not influenced by the value of any other feature, that is all features are independent of each other given the classification variable (Russell, 2003). Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n ;

$$P\left(y \mid x_1, \ldots, x_n\right)=\frac{P(y) P\left(x_1, \ldots, x_n \mid y\right)}{P\left(x_1, \ldots, x_n\right)}$$

Using the naive conditional independence assumption that

$$P\left(x_i \mid y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\right)=P\left(x_i \mid y\right),$$

for all i, this relationship is simplified to

$$P\left(y \mid x_1, \ldots, x_n\right)=\frac{P(y) \prod_{i=1}^n P\left(x_i \mid y\right)}{P\left(x_1, \ldots, x_n\right)}$$

Since $P(x_1.....x_n)$ is constant given the input, we can use the following classification rule:

$$\begin{gathered}P\left(y \mid x_1, \ldots, x_n\right) \propto P(y) \prod_{i=1}^n P\left(x_i \mid y\right) \\ \Downarrow \\ \hat{y}=\arg \max_y P(y) \prod_{i=1}^n P\left(x_i \mid y\right),\end{gathered}$$

GaussianNB implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian distribution.

`Random forests classifier` is an ensemble learning method for classification, that operates by constructing a lot of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees (Ho, Tin Kam 1995). It is supervised learning algorithm.

The process of decision trees begins with an initial query, for instance, "Shall I go surfing?". Further questions like "Is it a strong wave?" or "Is the wind moving offshore?" are then posed, forming the decision nodes that divide the data. These questions guide an individual towards a final decision, represented by the end point or leaf node. Data that meets the criteria moves along the "Yes" path, while the rest follows the alternative route. Decision trees aim to find the optimal way to categorize data, often utilizing the Classification and Regression Tree (CART) algorithm for training. The quality of the division can be assessed using measures such as Gini impurity, information gain, or mean square error (MSE).

`XGBoost`, which stands for Extreme Gradient Boosting, is a machine learning method that falls under the category of ensemble learning. It's widely used for supervised tasks like regression and classification. This technique constructs a predictive model by sequentially combining the results of multiple individual models, typically decision trees.

XGBoost operates by progressively incorporating weak learners into the ensemble, with each subsequent learner concentrating on rectifying the mistakes of its predecessors. It employs a gradient descent optimization approach to minimize a specified loss function during the training phase. Notable aspects of the XGBoost algorithm include its capacity to manage intricate data relationships, the use of regularization techniques to avoid overfitting, and the integration of parallel processing for effective computation. XGBoost is extensively utilized across various fields because of its impressive predictive capabilities and adaptability across diverse datasets.

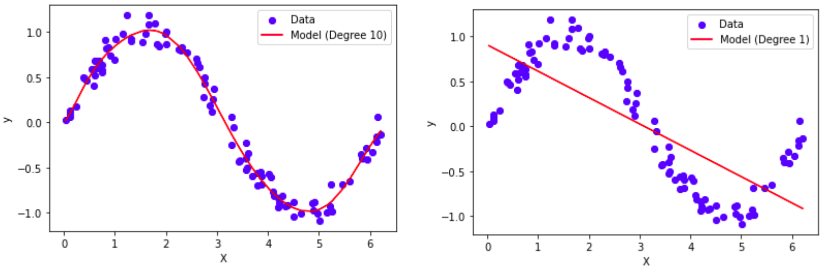
`Lasso` model is a linear method that estimates sparse values for coefficients. It's beneficial in certain situations because it often favors solutions with fewer non-zero coefficients, thereby decreasing the number of features the solution relies on. This feature makes Lasso and its variations essential in the field of compressed sensing. When certain conditions are met, it can accurately identify

the set of non-zero coefficients.
Mathematically, it consists of a linear model with an added regularization term. The objective function to minimize is:

$$\min_{w} \frac{1}{2n_{\text{samples}}} \|Xw - y\|_2^2 + \alpha\|w\|_1$$

The lasso estimate thus solves the minimization of the least-squares penalty with $\alpha\|w\|_1$ added, where α is a constant and $\|w\|_1$ is the l_1-norm of the coefficient vector.

`Overfitting` happens when a model fits the training data too closely, capturing not just the patterns, but also the random fluctuations within the data. This leads to the model performing well on the training data, but poorly on unseen data or new data, as it's focused on memorizing details rather than grasping the core concepts. Essentially, the model has adapted excessively to the training data, including its noise and outliers, which hinders its ability to generalize to new, unseen data.`Underfitting`, on the other hand, happens when a model is too simple to capture the underlying structure of the data. It fails to learn from the training data effectively and performs poorly on both the training and test data. Underfitting can occur when the model is too basic or when the training process is stopped prematurely. Essentially, the model is not complex enough to capture the patterns in the data.



(a) overfitting

(b) underfitting

Figure 1: visual representation of overfitting and underfitting

3 Methodology

ABIDE II data is imported from the pacakge called `ndslib`. There are 1004 rows and 1446 columns in the data. Rows correspond to a person (sample) and each column corresponds attributes which includes, a unique id, site where data is collected, age, gender, ASD diagnosis status (which we copied to another pandas dataframe called 'phenotype'), and the 1440 are the features or the attributes from the fmri; and we copied all 1440 to a pandas dataframe called 'features'.

The following python packages or libraries are used for this project; scikit learn, numpy, pandas, matplotlib and seaborn. further, each machine learning used in this project uses various machine learning algorithms which are imported from the scikit learn library (Pedregosa, F, 2011).

The dataset did not contain any empty cells. Hence, there was no need to insert dummy values like mean. All the target values were encoded in the data as 2 (ASD) and 1 (non-ASD). It is converted to 1 and 0 respectively and initialized as the y vector.

Since there are 1440 features, throwing everything at the models has a lot of computational cost and also, all 1440 may not be important in classifying. A PCA (principal component analysis) did not show any sort data clustering, indicating it's not easy to classify this data. In PCA, PC1 represents the direction of maximum variance in the data. It captures as much variability in the data as possible along this direction. PC2 represents the direction orthogonal (perpendicular) to PC1 that captures the maximum remaining variance after accounting for PC1.

I used Gaussian naive bayes (GNB), Random forest, XGBoost and lasso models to classify the ASD and non-ASD. None of these models gave a satisfactory result. To tackle this problem, I used recursive feature elimination (RFE). It is a feature selection technique that recursively removes features from a model based on their importance, typically using model coefficients or feature importances. It's an iterative approach that starts with all features and gradually eliminates less important ones until the desired number of features is reached. After obtaining the 100 most important features using the above method, I only used these features in my feature vector X.

The dataset was split into training and testing using `train_ test_split` method. The feature matrix X and the label (target) vector y is split in train:test ratio of 0.8:0.2. The model was trained on the training set and further tested on the testing set.

To benchmark the classification models against neural network (NN)models, a very basic NN model was also implemented. The NN model had :

Input Layer: The model begins with a dense layer comprising 64 neurons and ReLU activation function, suitable for processing the input features. Dropout Regularization: To prevent overfitting, dropout layers with a rate of 0.5 are incorporated after each hidden layer. Hidden Layers: Two additional dense layers with 32 neurons each and ReLU activation functions are included to capture complex patterns in the data. Output Layer: The output layer consists of neurons equal to the number of classes (2 in this case), employing the `softmax` activation function for multi-class classification.

Optimization: The model is compiled using the Adam optimizer, known for its efficiency in training neural networks. Loss Function: Categorical cross-entropy is chosen as the loss function, suitable for multi-class classification tasks. Metrics: Accuracy is selected as the evaluation metric to assess the model's performance. Training: The model is trained for 10 epochs with a batch size of 32, ensuring that the weights are updated iteratively to minimize the loss on the training data. Validation: During training, the model's performance is monitored on the validation set (20%) of the data) to prevent overfitting.

4 Results

Since there are 1440 features, throwing everything at the models has a lot of computational cost and also, all 1440 may not be important in classifying. A PCA (principal component analysis) did not show any sort data clustering, which essentially says there is no distinct class based on the data.

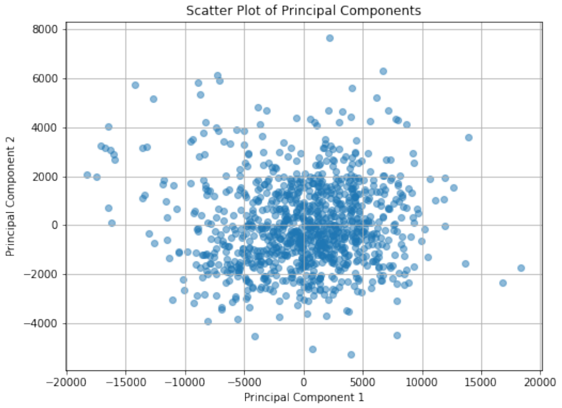
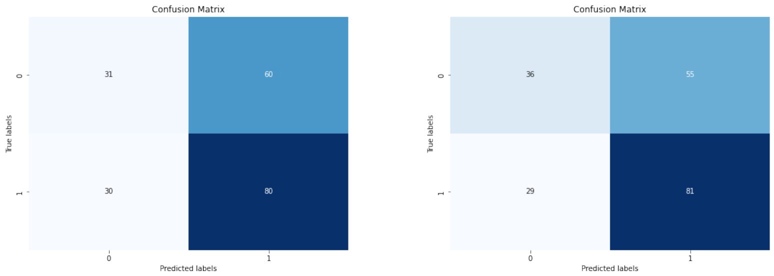


Figure 2: PCA using the 1440 features. No distinct custer is visible.

`Gaussian Naive Bayes (GNB)` model trained and tested on all 1440 fetures did not give a good accuracy, further instead of using 1440 features, the `RFE` most important 100 features was used as the feture matrix X, on which GNB was trained did not give any promising result.

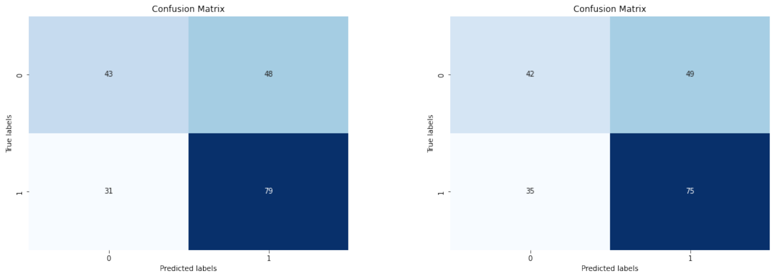
`Random forest` model trained and tested on all 1440 features did not give a good accuracy, further instead of using 1440 features, the `RFE` most important 100 features was used as the feature matrix X, on which Random forest was trained did not give any promising result.



(a) All 1440 features

(b) RFE 100 features

Figure 3: Gaussian Naive Bayes

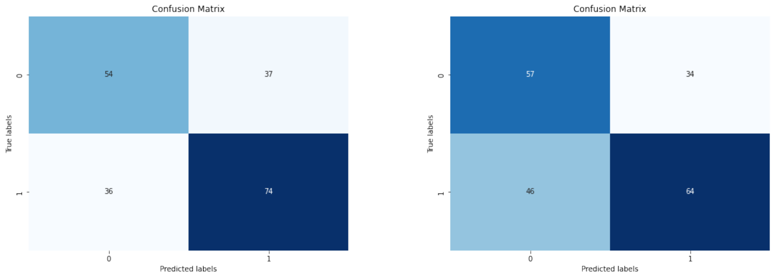


(a) All 1440 features

(b) RFE 100 features

Figure 4: Random forest

`XGBoost` model trained and tested on all 1440 features did not give a good accuracy, further instead of using 1440 features, the `RFE` most important 100 features was used as the feature matrix X, on which XGBoost was trained did not give any promising result.

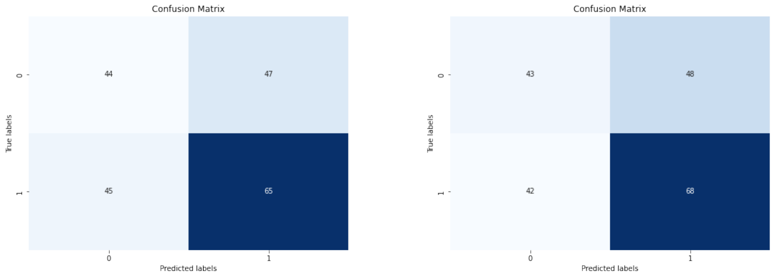


(a) All 1440 features

(b) RFE 100 features

Figure 5: XGBoost

`Lasso` model trained and tested on all 1440 features did not give a good accuracy, further instead of using 1440 features, the `RFE` most important 100 features was used as the feature matrix X, on which Lasso was trained did not give any promising result.



(a) All 1440 features

(b) RFE 100 features

Figure 6: Classification after Lasso regularization

	all features	RFE 100 features
GNB	0.58	0.58
Lasso	0.54	0.55
XGBoost	0.63	0.60
Random forest	0.60	0.58

Above given is a table with accuracy for all the classification models trained on both all features and RFE 100 features.

A simple feed forward `Neural network model` gave an accuracy of 0.547

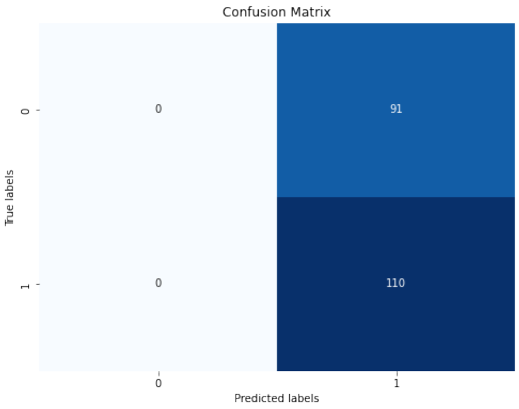


Figure 7: Confusion matrix for a output from a simple feedforward NN model.

5 Discussion

The classification using 1440 feature is daunting task. Narrowing down to the important feature is tricky task to begin with. Throwing everything ie all 1440 models at these models did not yield a promising result, nor did the RFE top 100 most important feature. A neural network model, which according to literature, is not a great tool for working on tabular data, which was evident in the one NN model I implemented.

Overall one thing which is important here is, event thought there distinct class present, what makes them distinct is yet to be found. A lot of feature engineering work is needed to optimize this task.

6 Reference

Di Martino, A et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. Mol Psychiatry. 2014 Jun;19(6):659-67. doi: 10.1038/mp.2013.78. Epub 2013 Jun 18.

Léo Grinsztajn, Edouard Oyallon, Gaël Varoquaux, Why do tree-based models still outperform deep learning on tabular data? 2022

Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

Ho, Tin Kam (1995). Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.