BLG 454E Learning From Data (2019) Term Project Report

SİNEM ELİF HASEKİ, HAKAN SARAÇ, ECE ÇINAR

Abstract— **Autism Spectrum Disorder(ASD) is a disorder which affects the learning ability, social communication skills and behaviours of the patients. This study aimed to explore a pattern between the brain regions in order to differentiate patients with ASD from healthy individuals. Dimensionality reduction was necessary since the given dataset had 595 features. Using Principal Component Analysis(PCA), dimensionality was reduced to 24 features. A classification model was trained using the AdaBoost classifier with Gaussian Naive Bayes base estimator. Overall classification accuracy of this model was %60 in the public leaderboard and %52.5 in the private leaderboard using the test set provided for the competition. Although the classification accuracy was not significantly high when compared to the other competition participants' scores, building up a robust algorithm was an important benchmark which we managed to provide.**

I. INTRODUCTION

The task was to train a model using the necessary machine learning tools and testing this model using the already-given test set. The main criteria which were kept in sight were robustness and accuracy. These two criteria have to be balanced in an optimal solution for a machine learning problem. High classification accuracy can be the result of coincidence, so robustness should not be ignored. This is why the methods which were used during preprocessing and classification were chosen to obtain robustness and a necessary accuracy of classification.

Several different preprocessing and classification algorithms were combined based on a logical background. Some of the methods used throughout the search for an optimal solution resulted in low robustness rate and some resulted in overfitting. The final methods combined were "AdaBoost", "Gaussian Naive Bayes" and "Principal Component Analysis". These three methods worked fine in integrity so that this combination was determined to be the final solution by the project group.

II. DATA SET USED

In the given dataset, instances have 595 features and there are 2 separate datasets which are training and test data respectively. In the training dataset, there is also a class label column which is used in training the models and predicting the labels of test instances based on the training results. Training dataset contains 120 samples, whereas there are 80 test samples to be predicted. In order to prepare the dataset, dimensionality reduction was crucial in order to eliminate curse of dimensionality. For mapping the features to a lower dimensional space, Principle Component Analysis (PCA) was implemented with the goal of having features maximum variance of 0.85. Principle Component Analysis has the following five steps:

- Standardization: in order to eliminate the chance of having dominant values, standardization of the given data is essential to make the values equally contributing to the analysis.
- Covariance Matrix: covariance matrix is essential for determining the variance of the data and relationships between values, which creates the opportunity to eliminate redundant information and to determine the significance of each component.
- Eigenvalues and Eigenvectors: for organizing the principle components in an uncorrelated way so that it does not have redundant data, eigenvalues and vectors of covariance matrix must be processed. In such manner, dimensionality reduction is accomplished without losing important information.
- Feature vectors: After step 3, decision to keep which feature vectors is made by checking the eigenvalues and significances. Following that, the feature columns which are kept form the feature vector.
- Reorientation of the data: Final dataset is formed by taking the transpose of the feature vector formed in step 4 multiplied with the standardized version of the given dataset formed in step 1. Most powerful changes in the original dataset are made in step 1 and step 5, which construct the final form of the dimensionally-reduced data. After employment of PCA, features were reduced to the amount of 24, which originally had the amount of 595. [1]

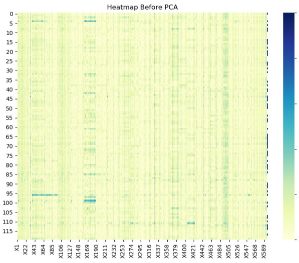


Fig. 1. Heat Map Before PCA

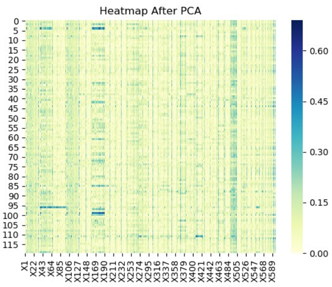


Fig. 2. Heat Map After PCA

III. METHODS USED

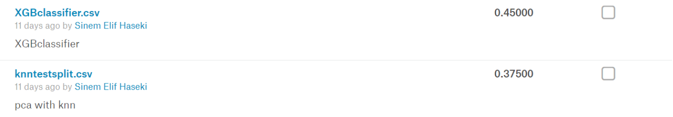
Before moving on to explain the methods used, a brief explanation of the software and the imported libraries would make it easier to understand the flow. First of all, Python programming language was used since it has a vast amount of libraries including the necessary methods and implementation is easier. The libraries which were used and their intended purposes are explained below (Note that some of the libraries and methods below were not used in the final version but they gave us some intuition):

1) Scikit-Learn: One of the most commonly used machine learning libraries. It contains several supervised and unsupervised learning algorithms. Bulding models using statistical data becomes easier with this library.
   - Model Selection
     - GridSearchCV: Used for finding the best parameters among a set of different values.
   - Decomposition
     - PCA: Used for preprocessing with Principal Component Analysis (PCA). It is explained in detail in dataset part.
   - Ensemble
     - AdaBoostClassifier: Boosting algorithms are used to track the accuracy prediction of the model sequentially, which makes the algorithm more robust and reliable. For boosting, AdaBoost (Adaptive Boosting) was employed. It combines multiple classifiers iteratively and creates a highly performing classifier. Basically, AdaBoost selects random training subsets and iteratively trains models on these while assigning weights to poorly performing classifiers so that they will predict more accurate in the next iteration.

       When using AdaBoostClassifier() to boost other models, the boosted model is given with parameter ”base_estimator” and an additional parameter ”algorithm='SAMME'” is added with its n_estimators.
     - RandomForestClassifier: Used for ensemble bagging with Random Forest classifier.
   - Naive Bayes
     - GaussianNB: Creates new x value probabilities using Gaussian Probability Density Function and then implements Naïve Bayes classification method. BernoulliNB() Bernoulli Naïve Bayes is also Naïve Bayes classifier but it is used for only binary features.
     - BernoulliNB: Bernoulli Naïve Bayes is also Naïve Bayes classifier but it is used for only binary features.
   - SVM
     - SVC: Support Vector Classifiers are based on finding the widest margin between two support vectors and usually used with kernels. It generally has 3 different kernel parameters which are ”rbf”: radial basis function, ”linear”: linear kernel, ”poly”: polynomial kernel. It also has ”gamma” parameter which is for justifying the kernel coefficients for rbf, poly and sigmoid kernels.
   - Neighbours
     - KNeighborsClassifier: K Nearest Neighbour algorithm performs quite well with large amount of samples and lower amount of features since the computations of N distances become more complex. It checks the N distances and classifies based on the majority of the results.
2) Numpy: Operations on arrays and matrices become easier using this library. Also scientific and mathematical calculations can be completed with less effort with the help of Numpy.
3) Pandas: Pandas is a Numpy based library. It is generally used for manipulation of data and it is used concurrently with Numpy library. In the project pandas library was used to read files with .csv extension.
4) CSV: CSV library helps reading and writing files with .csv extension. Read and write operations are gathered in objects. This library was used in order to write prediction data into csv files at the end of our project.
5) XGBoost: XGBoost is an ensemble used for boosting with extreme gradient boost and classification.
6) Matplotlib: Matplotlib is a plotting library which supports different plotting methods. In the project, matplotlib was used to visualize the train data, test data and intermediate data but it was only for analyzing how the current algorithm we are implementing works.

IV. RESULTS

First of all, KNN(K-Nearest Neighnors) and XGB classifier algorithms were tried on datasets and very inadequate results were obtained from these trials. KNN performs not quite well with large amount of features and lower amount of samples since the computations of N distances become more complex.

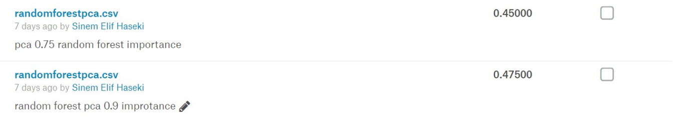


XGBclassifier.csv 0.45000 □
11 days ago by Sinem Elif Haseki
XGBclassifier

knntestsplit.csv 0.37500 □
11 days ago by Sinem Elif Haseki
pca with knn

Fig. 3.

In addition, random forest algorithm was implemented but it cannot create the optimum effect on small samples. Therefore, the use of the random forest algorithm has been abandoned due to poor performance. Random Forest algorithm was also combined with feature selection methods it provides, such as finding feature importances. Based on the obtained importance values, the features with importance values lower than the mean of importances were eliminated. This method is generally powerful with large datasets, but since the dataset in the term project was not as large, it did not perform quite well.

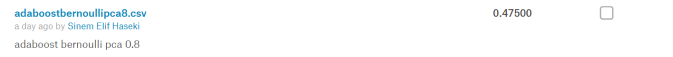


randomforestpca.csv 0.45000 □
7 days ago by Sinem Elif Haseki
pca 0.75 random forest importance

randomforestpca.csv 0.47500 □
7 days ago by Sinem Elif Haseki
random forest pca 0.9 improtance

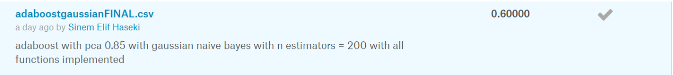
Fig. 4.

After that, AdaBoost which is a boosting algorithm to create a strong classifier from a number of weak classifiers was tested. At first, bernoulli was performed on datasets with pca. After that gaussian naive bayes is used on datasets and these results were better than their predecessors.



adaboostbernoullipca8.csv 0.47500 □
a day ago by Sinem Elif Haseki
adaboost bernoulli pca 0.8

Fig. 5.

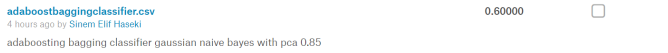


adaboostgaussianFINAL.csv 0.60000 ✓
a day ago by Sinem Elif Haseki
adaboost with pca 0.85 with gaussian naive bayes with n estimators = 200 with all functions implemented

Fig. 6.

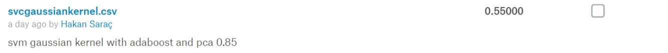
Also, svm with linear kernel trick and bagging are incorprated into AdaBoost algorithm respectively. However, the desired results could not be reached.

Lastly, random forest with feature importance selection and adaboosting was employed, however random forest perfoms well with bagging algorithms, not with boosting algorithms. That's the reason why random forest with adaboosting got lower results, which was expected.



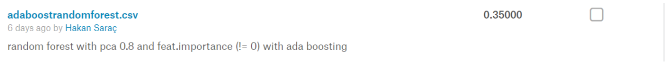
adaboostbaggingclassifier.csv 0.60000 □
4 hours ago by Sinem Elif Haseki
adaboosting bagging classifier gaussian naive bayes with pca 0.85

Fig. 7.



svcgaussiankernel.csv 0.55000 □
a day ago by Hakan Saraç
svm gaussian kernel with adaboost and pca 0.85

Fig. 8.



adaboostrandomforest.csv 0.35000 □
6 days ago by Hakan Saraç
random forest with pca 0.8 and feat.importance (!= 0) with ada boosting

Fig. 9.

V. CONCLUSIONS

In the project, the most efficient code is AdaBoost with Gaussian Naive Nayes n estimators equal to 200 with 0.85 pca due to high and stable results.
In addition to our valid code, we tried both bagging and boosting algorithms together but our result has deteriorated. The reason that we have chosen our valid code is that our valid code is a robust algorithm. Our public rank and score is given below respectively:
19, 0.6750.
Our private rank and score is given below respectively:
22, 0.5250.

VI. REFERENCES

[1] Holland, S. M. (2008). Principal components analysis (PCA). *Department of Geology, University of Georgia.* 249-251. http://strata.uga.edu/8370/handouts/pcaTutorial.pdf.