BLG 454E Learning From Data (2019)

Term Project Report

Mehmet Berkan ÜNAL Muhammed Enes KUYCAK Damla Nisa ÇEVİK

A python project for predicting autism spectrum disorder (ASD) classification from given data. Program preprocess the the features before training. It can select from five variants of machine learning models.

I. INTRODUCTION

This project's aim was to predict which person has ASD with using the given data by classification.

First, we implemented simple neural network for the model using keras, however; we realized our model generally overfits the data because the data set is quite small for a neural network [1].

Then, we implemented 4 different variatons of support vector machine with using *sklearn.svm* library. These variations were:

- C-Support Vector Classifier
- Svm with poly kernel
- Svm with gaussian kernel
- Svm with rbf kernel

Kaggle names: Berkan, Enes Kuycak, Damla Nisa

Team name: 150130013_150140056_150170712

Score: 62.5 on Public leaderboard, 32.5 on Private leaderboard

Rank: 48 on Public, 62 on private

II. DATA SET USED

When the dataset was explored, the training data had 596 columns (1 of them is the label column rest of them were feature columns), and 120 samples. This was problematic since the features were more than the samples.

To prepare the data; first, we considered the potential noises or outliers. So as the first step of preprocessing the data we checked for noisy/outlier data with the function our written, remove_noise_and_outliers. Thankfully there was none.

As the second step, we split the data into x and y sets. To use the PCA module of sklearn.decomposition we need to scale the data. Which we used StandardScaler module of sklearn.preprocessing. After that, for better performance (computational and prediction) of the data, we used feature extraction. We gave the pca_variance_percentage variable as 0.95 to get 95% minimum of the variation of the data. This resulted with 50 principle components which we stored as the new features.

We thought about removing the features with low variance with the function "remove_columns_with_low_variance" because they cannot be used for finding patterns [2]. In this function we used VarianceThreshold function from sklearn feature_selection library [3]. But then we scraped this function call as it was not really necessary since we are already selecting best features.

Finally, for selecting the best features out of the remaining features we thought using a Minimum Redundancy and Maximum Relevance algorithm, but couldn't find a fitting algorithm. We decided on using SelectKBest function from sklearn feature_selection library [3] with score_func=chi2. But chi2 algorithm was not usable with negative values, instead we used default score_func which didn't give errors.

III. METHODS USED

create_model.py: We used SVC function from sklearn library to create the support vector machine model for classification. SVC function [4], that we used, takes two arguments. First argument allows us to select kernel type, we selected linear kernel type since the data set is linearly separable [5]. Second argument is the penalty parameter of the error term, by default it is 1 but with trial and error we concluded at 1200.

train_model.py: While training the model we used fit function from Keras library by giving the training data as parameter.

predict_model.py: For predicting the test data we used predict function from Keras library. We only give the test data as parameters, other arguments of the predict function, such as batch_size, set as default.

IV. RESULTS

We tried lots of models by changing the kernel type (poly, linear etc.). We expected the RBF kernel support vector machine the perform best but default svc svm model performed the best (We are not really sure since the scoring of test data was not that useful). At the end we decided to use linear kernel since it was giving the highest accuracy in the public leader

board. We get 62.5 score in public leader board and placed at 40th, however; we placed at last in the private leader board with score 32.5. Previously our submission was placed at 18th with score 52.5 in the private leader board. One of our submission has 70.0 score in public and 52.5 score in private, we did not select specific submission to be shown in the final leader boards, so we do not know why the submission with lowest score is selected.

V. CONCLUSIONS

We had a chance to implement the theoretical knowledge that we learnt in class in this project. While doing this project we had some problems with understanding what is wrong with the model and we did not really have a way to test the parts that we implemented, such as is the problem in the preprocessing part or creating the model and these kinds of issues lost us time.

REFERENCES

[1] https://www.researchgate.net/post/Is_deep_learning_not_useful_if_there_is_no_large_dataset_available

[2] https://riptutorial.com/scikit-learn/example/17328/low-variance-feature-removal

[3] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[5] https://www.geeksforgeeks.org/creating-linear-kernel-svm-in-python/