BLG 454E Learning from Data - Spring 2022 Term Project

Ahmet Ramazan Çapoğlu, Zahid Çakıcı, Abdullah Asım Emül, Umut Evren, Yusuf Seven

1 Introduction

The project is consisting of two different datasets, respectively low resolution (LR) matrices which are comprised of 189 samples and 35578 features and high resolution (HR) matrices which are comprised 189 samples and 12720 features. The motivation is to establish a learning model predicting high resolution (HR) connectivity matrix $X^{HR} \in R^{268\times268}$ over the low resolution (LR) matrix of the brain. The process is undergoing through feature vectors and and data matrix representing the corresponding test samples.

Our team of 150190016_150190080_150190006_070190403_070140430 takes place in the competition as 15th with the score of 0.02399.

2 Datasets

Since the dataset does not encompass any missing value or anything unwanted, data cleaning is no required. In order to handle with the massive size of features, principal component analysis (PCA) is used as the method of dimensionality reduction. With PCA, there is at least an order of magnitude of difference in number of features, bringing about a rapid calculation on the resolution processing function which has one-tenth of the calculation duration in comparison to the previous state without PCA.

3 Methods

As shown in figure 1, we start processing our data by extracting features using PCA [2]. By looking at combinations of features with the highest variance over our dataset, we can produce a much simpler set of features representing the essence of what distinguishes samples in that dataset apart.

After this, we perform supervised learning using the provided ground truth. We make use of the LinearRegression [1] function in the Scikit-learn Python library. Finally, to test our findings, we calculate the mean-square-error, mean-absolute-deviation, and Pearson correlation coefficients by comparing our findings to the ground truth of the testing sample.

There are two key algorithms in our pipeline: PCA and linear regression. The PCA algorithm works by finding the eigenvalues of its input matrix and selecting the first n of these values. Eigenvalues can be used as representations of the parts of a matrix that have the greatest extremities, meaning that by performing this operation we are effectively losing a neglible amount of information while greatly reducing the amount of data that we need to work with.

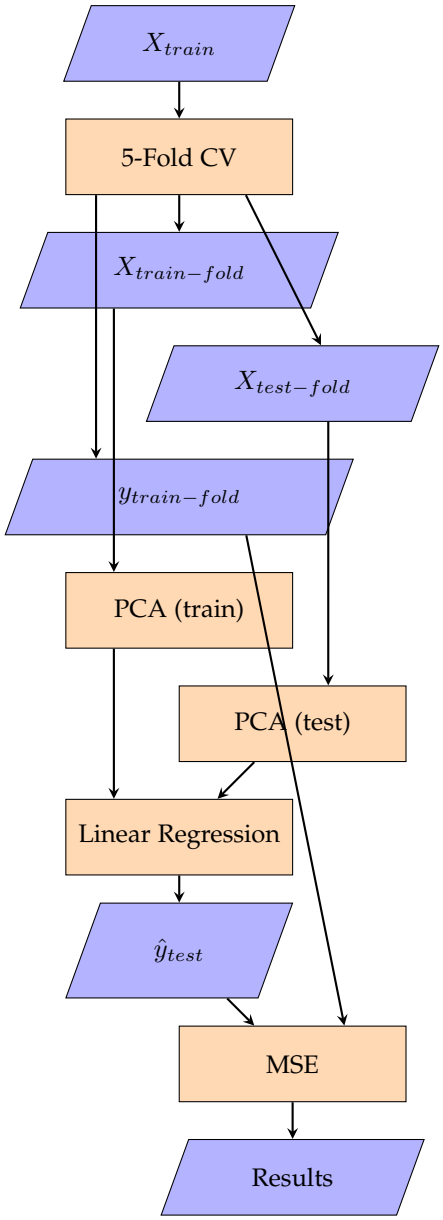

Fig. 1: Our Learning Pipeline

Linear regression as an algorithm, finds a linear model which can most accurately predict the right output from a given input. To achieve this, the algorithm finds the direction in which the parameters of the linear model can move to minimise error the most. By doing this until no more improvements can be made to the model, the algorithm finds the closest fit. We chose this algorithm because we recognised that we were working with a system which was close to linear.

4 Result and Conclusions

First, report your 5-fold cross-validation results on the initial set comprising 189 samples. Explain your results. Second, give your Kaggle score and ranking.

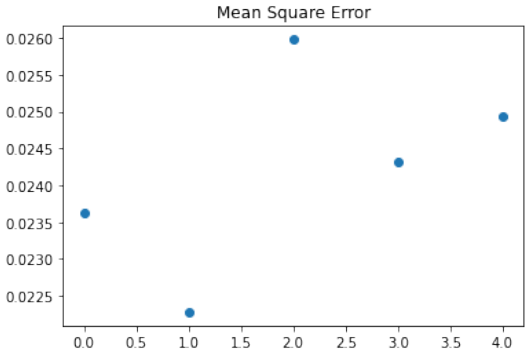

Fig. 2: MSE of Folds

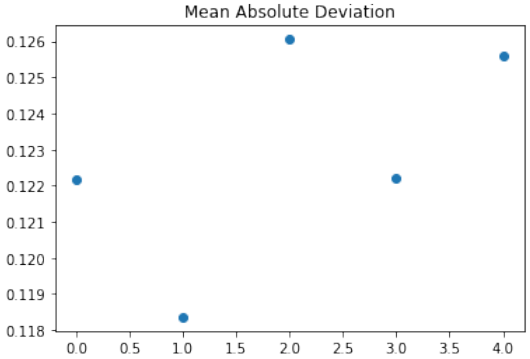

Fig. 3: MAD of Folds

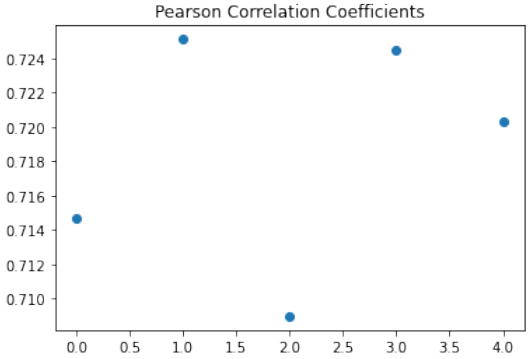

Fig. 4: Pearson Corr. of Folds

Out of the 5 folds we trained and tested, we had mean square errors around 0.024. The highest mean square error was 0.026 while the lowest was 0.022. The mean absolute deviation followed the same pattern as the MSE. The average was around 0.123 with a high of 0.126 and low of 0.118. The Pearson correlation coefficients showed and inverted trend compared to the error as expected. The highest correlation being 0.724 and the lowest 0.709 with an average of 0.719.

Our learning model trained with the full training set and tested on the standalone testing set gave us an MSE score of 0.02399 and a Kaggle leaderboard placement of 15th.

References

[1] Scikit-Learn, LinearRegression: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
[2] Scikit-Learn, PCA: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html