Prediction of the evolution of brain connectivity over time.

1st Joshgun Rzabayli
Istanbul Technical University
Computer Engineering
Istanbul, Turkey
150160901

2nd Farid Huseynov
Istanbul Technical University
Computer Engineering
Istanbul, Turkey
150160904

3rd Sercan Aydın
Istanbul Technical University
Computer Engineering
Istanbul, Turkey
150170707

4th Halis İbrahim Aydın
Istanbul Technical University
Computer Engineering
Istanbul, Turkey
150170721

***Abstract*—The document refers to Learning from Data 2020/2021 Fall term project. Various regression models and dimensional reduction techniques are tested on predicting brain connectivity at t$_1$ from time t$_0$. Support Vector Regression pipelined with selecting 130 highest scored features before training, yielded the least mean squared error on 5-fold cross-validation.**

I. Introduction

In this final project it is required to predict next time point from brain connectivity between two brain parts at time t$_1$ from the given data of time t$_0$, by applying the tools of machine learning. This prediction aims earlier diagnosis of diseases that might affect brain connectivity from time t$_0$ until time t$_1$. We were able to minimize our error rate down to 0.00203, and became 7^{th} place on the Kaggle competition.

II. Data-sets

The brain connectivity data-set has dimensions of R^{35,35}. The matrix is symmetric therefore, only the upper triangular part is used in learning. Every sample is vectorized into 595 features R^{1,595} and with 150 samples the training matrix is in the shape of R^{150,595}.

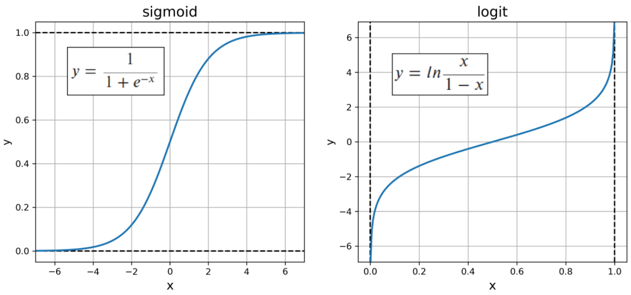

Fig. 1. Sigmoid and logit(inverse sigmoid) functions [1]

Since we believe that the data lies on a sigmoid like space, we applied the inverse sigmoid function to every target value and handled with overflow issues. This allowed us to test various regression models.

III. Methods

Our program needs to map a vector R^{1,d} to another vector R^{1,d}. Instead of using multi-output learning libraries, we trained a different learning and feature selection model for every dimension of the output vector.

A. Model Selection

Since the output data has continuous values, we decided on regression models. In the beginning, we used SelectKBest and set the number of features 80 as constant to evaluate different learning methods. The Regression methods tested for choosing the best model that has the most explanatory power on the dataset are as follows:

· Linear Regression
· Decision Tree Regression
· Random Forest Regression
· Support Vector Regression
· Adaboost Regression

- ***Linear Regression***:
Linear Regression tries to map input samples to continuous target values using a line. To find this line, it chooses the line parameters that will minimize the following loss function.

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + w_0))^2$$

- ***Decision Tree Regression***:

As described in [2], Decision Tree is another supervised learning algorithm that can be used with continuous data. Sayad explains that, by separating the dataset into minor subsets, it creates a tree structure and the best predictor would be the root node [3].

- ***Random Forest Regression***:

Random Forest Regression is a product of ensemble learning. It trains n different, random trees. When a new sample is encountered, every tree makes a prediction, and the average would be the final decision. This improves accuracy and prevents over-fitting.

- ***Support Vector Regression***:

Support Vector Regression applies the principle of SVM(Support Vector Machine) on regression problems. SVMs represent optimal hyperplane with support vectors. Support vector regression, on the other hand, tries to find a plane that will contain the maximum number of data points inside the margin with a fixed distance. Awad and Khanna emphasize that SVR adopts a loss function, in order to penalize predictions that are farther from the desired output [4].

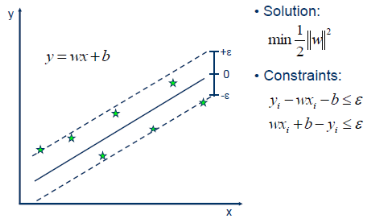
y
$y = wx + b$
+ε
0
-ε
x
• Solution:
$\min \frac{1}{2}\|w\|^2$
• Constraints:
$y_i - wx_i - b \le \varepsilon$
$wx_i + b - y_i \le \varepsilon$

Fig. 2. Support Vector Regression [5]

- ***Adaboost Regression***:

At first, AdaBoost regressor trains on the whole dataset and then repeats this learning process by choosing samples in a non-uniform manner. It gives higher weights to samples that previously caused errors.

In order to test the different models, we compared models' mean squared errors trained on 80 features.

B. Feature Selection and Extraction

Since on 80 best features Support Vector Regression has been tested to have the least mean squared error in 5-fold cross-validation, we decided to continue with that. After choosing the model, we tested PCA to extract and SelectKBest to select different numbers of features varying from 10 to 140 in order to find the one with the lowest error.

- ***PCA***:

PCA(Principal Component Analysis) is an unsupervised method that tries to find an orthogonal basis that minimizes the average distance between data points and the basis lines. This results in uncorrelation.

- ***SelectKBest***:

SelectKBest is a supervised method that learns to select k highest scored features by looking at both sample and target values.

We compared PCA and SelectKBest on the mean squared errors by changing the number of features used in the model.

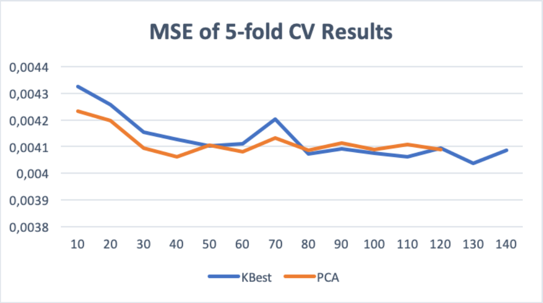

Fig. 3. MSE comparison of different number of features

From figure 3, it is observed that the number of features does not have a specific trend. For the sake of neither underestimating nor overestimating, we have chosen 130 as an optimal number of features.

IV. Results and Conclusion

In a nutshell, Support Vector Regression model with K-best feature selection technique by selecting 130 features was tested to have the least error on this particular dataset. After we trained and tested this pipeline using 5-fold cross-validation, predicted results' mean squared error was $0,004072235$ and Pearson correlation between predicted values and ground truth was $0,676034$. Moreover, in Kaggle competition, we became the 7^{th} team among 18 teams and our best public score was 0.00203.

References

[1] H. Hentschke, "Sigmoid Activation and Binary Crossentropy- A Less Than Perfect Match?," Medium, 21-Feb-2019. [Online]. Available: https://towardsdatascience.com/sigmoid-activation-and-binary-crossentropy-a-less-than-perfect-match-b801e130e31. [Accessed: 13-Jun-2020].

[2] "Python: Decision Tree Regression using sklearn," GeeksforGeeks, 04-Oct-2018. [Online]. Available: https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/. [Accessed: 12-Jun-2020].

[3] S. Sayad, "Decision Tree - Regression," Decision Tree Regression. [Online]. Available: https://www.saedsayad.com/decision_tree_reg.htm. [Accessed: 12-Jun-2020].

[4] Awad M., Khanna R. Support Vector Regression. In: Efficient Learning Machines. Apress, Berkeley, CA, 2015

[5] S. Sayad, "Support Vector Machine - Regression (SVR)." [Online]. Available: http://www.saedsayad.com/support_vector_machine_reg.htm. [Accessed: 12-Jun-2020].