BLG 454E Learning from Data
Fall – 2020

-Term Project-

Problem: Predict the evolution of brain connectivity over time.

The term project has 3 deliverables: The Kaggle competition results, the source code, and a report explaining the designed framework. You will be provided with **templates to use** for the report.

1. Kaggle Competition (20 points)

Description

We have created a private class competition on Kaggle. Please click the following link for the term project competition https://www.kaggle.com/t/c682f52a20d44e0d9d1709964318efc3

Dataset

Given an elderly population, ach brain is encoded in a symmetric connectivity matrix $\mathbf{X} \in \mathbb{R}^{35 \times 35}$, where an element $\mathbf{X}(i, j)$ denotes the strength of the connectivity between two brain regions i and j. By vectorizing the off-diagonal upper triangular part of $\mathbf{X}$, we generate a feature vector $\mathbf{x} \in \mathbb{R}^{1 \times d}$ $(d = 595)$ representing a single sample (i.e., brain). By stacking the samples vectors vertically across N=150 subjects, we construct the data matrix $\mathbf{D} \in \mathbb{R}^{N \times d}$.

By measuring the brain connectivity at two different timepoints t_0 and t_1, spaced out by 6 months, we can track the changes in the brain as a network. The goal of this project is to predict brain connectivity features at t_1 from brain connectivity measured at a previous timepoint t_0.

If we formalize this mathematically, we aim to learn a mapping f that maps each feature vector $\mathbf{x}(t_0)$ measured at initial timepoint t_0 to $\mathbf{x}(t_1)$ at a follow-up timepoint t_1:

$$f: \mathbb{R}^d \rightarrow \mathbb{R}^d$$

$$f(\mathbf{x}(t_0)) = \hat{\mathbf{x}}(t_1) \approx \mathbf{x}(t_1)$$

$\hat{\mathbf{x}}(t_1)$ denotes the predicted feature vector by the model f.

Goal

In this challenge, we ask you to apply the tools of machine learning to predict the brain connectivity for the next time point.

Submission Process

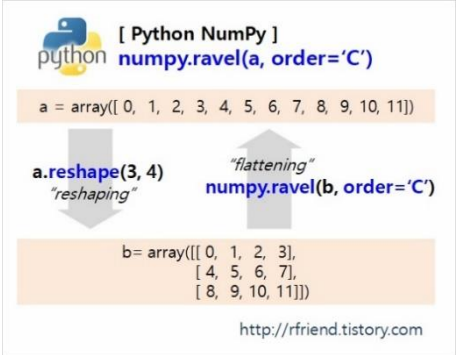
To see the performance of your model on test data, submit your predictions of test data to Kaggle in the defined format. Kaggle will calculate and rank the submission scores using the public test data throughout the competition. These scores are publicly visible on public leaderboard. After the competition end, a *private* test data is used to calculate final model performance. Private leaderboard is not released to users until the competition has been closed. Public leaderboard is calculated with 50% of the test data. The final results will be based on the other 50%, so the final standings may be different. Therefore, train your model as general as possible to avoid overfitting on train and public part of the test data.

Scoring Metric

In Kaggle, your submission is evaluated by the Mean Squared Error (MSE).

Submission File Format

Since you are requested to submit a solution dataframe (a matrix) with the size of 80 x 595 and Kaggle doesn't allow that kind of submissions, you **have to vectorize** (it is called **melting** in data science) your dataframe. Once you created your pandas dataframe (say **`df`**), do the following:




```
meltedDF = df.to_numpy().flatten()
```

Those who doesn't want to convert his/her dataframe to numpy before melting it, they can directly use pandas' **`melt()`** method: (But using the above line is the simplest and the safest way)

https://pandas.pydata.org/docs/reference/api/pandas.melt.html

You should submit a csv file with exactly 80x595 entries plus a header row. The file should have exactly 2 columns:

1. ID: [1,…,47600]
2. Predicted (contains your real-valued values)

```
ID,Predicted
1,0.22343
2,0.0244
3,0.59028
...
47598,0.2655
47599,0.04379
47600,0.02373
```

Submission CSVs must have a header row consisting of ID and Predicted as in the sample submission. Using different column names causes a fail in submission process. ID column must include all ID values between [1, 47600].

PS: Your submission will raise an error in cases: you have extra columns (beyond ID and Predicted), extra rows, ID column doesn't consist of integers between [1,47600], Predicted column includes value other than real-values.

You can download the sample submission file (sampleSubmission.csv) on the Data page.

Rules

- Every student has to create a Kaggle account
- Form a team of **3 or 4 students (The "team" tab on the competition)**
- Individual submissions are **not allowed**. In such a case, send us an email so that we assign a random teammate.
- Team members **must** be students **officially registered** to the LfD class
- Team names should be in the following format: StudentID1 _StudentID2_StudentID3
- Submission format is explained and a sampleSubmission file (sampleSubmission.csv) is given in the competition webpage.
- You are allowed to use only **Python** programming languages (with jupyter) for the implementation.
- Academic dishonesty including cheating, plagiarism, and direct copying is unacceptable. Note that your codes and reports will be checked using plagiarism tools!

2. Report (40 points)

Prepare a report in Latex/Word using provided IEEE Conference Paper template. Your report must **not exceed** 2 pages (**one extra page** can be allowed for the **main Figure** illustrating the learning pipeline)!

The report should consist of the following sections:

1. **Introduction:** Mention about what and why you did in this project briefly. Give your final score and rank in the competition with your Kaggle name and team name.

2. **Datasets:** Explain your methods for data preprocessing in detail.

3. **Methods:** **The how?** Describe each component of your brain network evolution prediction framework. Include a **main figure** illustrating the key steps of the proposed solution (learning pipeline). Explain how you train and test your model in

general. **The why?** Explain **why** you have made selected such components. Give all details about the methods like the algorithms used, parameter tuning, etc.

4. **Results and Conclusions: First**, report your **5-fold cross-validation** results on the initial set comprising 140 samples. You can also provide the scores you measured with other evaluation metrics such as the *MAD (mean absolute distance) and Pearson correlation* between the predicted and ground truth feature vectors and plots of the related performance. Explain your results. To test your model on the test_t0 of the Kaggle competition, you will retrain your model on the whole train_t0 dataset, then test it on the test_t0 to predict test_t1 (as explained in the Kaggle competition). **Second**, give your Kaggle score and ranking.

5. **References:** The list of references cited in the report. Don't forget the citation to the related reference in the report.

3. Code with 5-fold CV (30 points)

The version of your code that you will upload should have 5-fold cross-validation implemented. The code should take as input two datasets (at timepoint t0 and t1), performs 5-fold cross-validation for training and testing the designed framework model. The code will have two outputs: (1) the predicted samples at timepoint t1 saved in a predictions.cvs file (you can use the same Kaggle format to same them), and (2) the MSE between the ground truth and predicted samples.

Important note 1: the code will take in to 150 samples and perform 5-fold CV on this set. At this stage, you don't need to use the extra test set that was provided to you.

Tidy up your code as to

- run simply,
- get all necessary inputs as function parameters (train and test data, model parameters),
- produce output, i.e. the submission file (test predictions)
- have explanatory comments

Important note 2: Use the following random anchorization seed when applying 5-fold CV:

```
- import random as r
- r.seed(1)
```

Important note 3: For computing the MSE, once you complete your 5-fold CV, you will end up with predicted vectors and ground truth (actual) vectors for the 150 samples. You can compute the MSE between their vectorized versions as follows:

```
- from sklearn.metrics import mean_squared_error as mse
- actual = actual.to_numpy().flatten() #melt t1 dataset (ground truth)
- predicted = predicted.to_numpy().flatten()#melt your prediction
- mse(predicted,actual)#returns mse result for two melted matrices
```

Above, actual and predicted are pandas dataframe, and at line 2 and 3, we convert them into numpy's ndarray before melting (vectorize) it.

4. Project Overall Evaluation

For the project, you will provide a final report in IEEE conference paper format (that is given to you in both Word and Latex format). Total score of your project will be calculated as follows:

- The Kaggle competition (50 points)
  - 30 points: 5-fold CV
  - 20 points: Your Kaggle rank
- Report: 40 points
- Code: 10 points

- Your code should be clean and readable. Implement your code as powerful as possible befitting for a 4th grade student. Weak coding might cause losing 5 to 10 points.

Bonus Marks

Top five team will be rewarded with bonus marks, respectively, 30pts, 25pts, 20pts, 15pts and 5pts., according to the average of the public and private leaderboard scores.

Ninova Submission Policy

- Submit your PDF report, PPT, and code in a zip/rar file through Ninova on time.
  - **Unnecessary uploadings (files, pictures, etc.) will be penalized!**
  - **Only put things in your zip file that you are asked to.**
- No late submissions will be accepted

Become a co-author?

The codes of the top 10-15 best performing teams will be included in a research paper to submit to a high-impact journal. If you are interested in becoming a co-author, please drop an email to islem.rekik@gmail.com and CC both research assistants (kamard@itu.edu.tr and akti15@itu.edu.tr). You can help in the paper writing or related tasks (e.g., double-checking the methods, codes). We will work on the paper in February.

For more, you can check our previously published Kaggle competition from the LfD class at:

https://github.com/basiralab/BrainNet-ML-ToolBox

References

To learn more about Kaggle Competitions, https://www.kaggle.com/docs/competitions

---

Res. Asst. Şeymanur AKTI, akti15@itu.edu.tr

Res. Asst. Doğay KAMAR, kamard@itu.edu.tr