**Case 1: Feature Engineering**

**Predicting Central Neuropathic Pain**

**Lin Chengli 2529061L**

**Tang Mingkai 2550578T**

# Introduction

## Introduction to case study

It is important to design a model to determine whether a patient has Central Neuropathic Pain (CNP), as some patients with Spinal Cord Injury (SCI) suffer from CNP. There is currently no medical technique for completely curing this pain, and it can only be prevented before the pain occurs, but preventive drugs have serious side effects on patients. Therefore, preventive drugs cannot be used until it is confirmed that the patient is at risk of suffering from CNP. Currently, it is inaccurate for doctors to determine whether a patient has CNP, with academic reports claiming that Brain Electrocephalenogram (EEG) can be used as a characteristic of judgment, while there is a problem that EEG's results contain so many fields that it is difficult for researchers to judge what is useful. The experimental data below, the data came from 18 participants, including 9 features with 48 electrodes.

## Clear the statement of predictive/exploratory question.

There are three questions need to be proved and explored. Firstly, which is the best Feature Selection among the four methods in lesson. The below part demonstrates three of them: wrapper, filtering, and Embedded, comparing with their accuracy. Next, finding the most useful features and remove useless ones, then return a matrix with features and participants. Third question is whether the result is good or not, from the aspect of accuracy, specificity and sensitivity, which can be judged from the numerical value, because they represent the performance of the model.

# Methods

The data is constructed by 9 feature and repeated 48 times, and 18 subjects with 10 repeated. Our method is wrapped out the feature which may decrease the accuracy.

The main idea is that too many features would affect model doing the prediction and so as the accuracy.

First part, we do the aggregation by its feature, for example, we collapse all the 48 data and store its average to the new variables, in the end, we will have 180\*9 matrix and each column represent the average of the feature. For the following, we did wrapper, filter, and embedded with both aggregation data and raw data(non-aggregation)

For the wrapper, by using package called Sklearn (see as Fig 1.), we wrap out the features which lessen the observation to the model.

Graphical user interface, text

Description automatically generatedFig1.

Text

Description automatically generatedAt the begging, we store the accuracy and other judgments from the original data and aggregation data.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedWe then wrap out one feature of 432 with non-aggregation (Fig2.b) and one of 9 with aggregation data (Fig2.a). Calculate the accuracy by cross-validation (cross\_val\_score) and put back the feature if the accuracy is not increase. Repeat the method until go through all the data with both non-aggregation and aggregation.

Graphical user interface, text, application

Description automatically generatedFig2.a above is aggregation with wrapper

Fig2.b below is non-aggregation with wrapper

Then, for the filter part, we do the filter to find out the best n features for model to predict by applying SelectKBest with chi2.

Aggregation non-aggregation

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

For the embedding, we choose t-sne from sklearn.

Aggregation non-aggregation

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

We then adjust the probability which affects computer to predict the answer. For example, algorithm will regard the test is class a while its probability is higher than 0.5, but the standard it takes may not be suitable for this data. We change the probability it would consider as the class from 0.1 to 0.9 and find out the best solution of this data

Text

Description automatically generated with medium confidence

# Result

Firstly, we did the aggregation by taking the average data from each feature. The accuracy shows that this method improves the performance of the model. Aggregation methods reduce the duplicate data which would affects model to predict well for the test data.

Text

Description automatically generatednon-aggregation Aggregation

Chart

Description automatically generatedText

Description automatically generatedwrapper shows the accuracy along while we do it literately. Here is the plot: non-aggregation Aggregation

Chart, line chart

Description automatically generated

The graph obviously shows that the accuracy is upward after we wrap out the noise data from the original one. Moreover, we calculate the number of the feature, which had been eliminated, by its electrodes and numbers. In the figure3., no. 8 has the highest elimination which

Chart, bar chart, histogram

Description automatically generatedFig3

result that this would affect the accuracy badly.

Chart, bar chart

Description automatically generatedAnother figure, ratio has almost 90 eliminations from the wrap out function.

Then the filter we apply is chi-squared, which will output non-negative value. By doing this and the combination of SelectKBest function to pick up the top k number feature from the data to reduce the cost and time.

Non-aggregation Aggregation

Text

Description automatically generated

Text

Description automatically generated

Last, we apply embedding function. We choose t-sne as our embedding function. T-sne is a function that reduce dimension and visualise data for human easily to observe.

Here is the result of the embedding function

Text

Description automatically generatedText

Description automatically generated Non-aggregation Aggregation

# Discussion

In conclusion, doing pre-processing cost lots of time in the whole processing. However, it directly affects the prediction and so as the accuracy. Before doing this task, we thought that the performance of data without aggregation would better than the aggregation one. We now realise that data have been classify or clustery help model to do a better prediction. For judging this model or method good or not, we consider both mse and accuracy. By judging the accuracy and mse we found that embedding function did a better job than wrapper and filter. We thought that embedding function is more suitable for these data than the others.

The idea of aggregation and non-aggregation is that we thought data with the same feature would have similar data. Also, too many duplicate data would unbalance the train data for model to train. In conclude, we found the best accuracy and so as the mse is aggregation with embedding function. However, we doubt that our model is under fitting or overfitting it would be better to use model to predict other data in the future.