In today's first session, we learned several ways to evaluate models. Before learning the evaluation method in earnest, we learned the pipeline function. Pipeline allows you to load and execute multiple functions at once. To solve the overfitting and underfitting problems,

By applying a model learned with a training dataset to the test dataset by splitting the training test, we learned how to finally check the performance. This method was found to ensure that the distribution is uniform for data sets with unbalanced class distribution. The reason for using the validation dataset was also found. This is because of the reason to increase the performance of the model by finding the optimal hyper parameter. In addition, various contents were learned. With k-fold cross-validation, a part of the training dataset is used as a validation dataset. The validation dataset is changed in turns. It helps to improve performance. Parallel processing is possible with multiple CPUs, but in the case of time series data, parallel processing may not be possible. Through the comparison between validation account and test accounting, we learned the strategies to take in the case of high bias and high variance. Increasing C lowers regularization, and increasing C too much leads to a difference between training account and validation account. High regularization is a robust model. The lower the regularization, the more favorable it is for learning, but overfitting can occur.

I didn't do feature engineering, but I moved on to the text classification chapter instead. I learned Naive bays, but it was not easy because there were many equations even though I learned it once in Korea. After that, I learned the full-fledged text classification process. The text of the dataset was vectorized, the training test split, the model was constructed, fit, and the model was evaluated in order. Unlike numerical data, text data was a little difficult to handle.