Error Analysis

1.K-fold. K-Fold Cross-Validation divides the original data into K-groups, makes one validation set for each subset, and uses the remaining K-1 sets as training sets, resulting in K models. This K model evaluates the results in the validation set separately, and the final error MSE (Mean Squared Error) adds up to average the cross-validation errors. Cross-validation effectively utilizes limited data, and the evaluation results are as close as possible to the model's performance on the test set, and can be used as an indicator of model optimization.

However, only when the total amount of data is small can other methods continue to improve performance, try K-Fold. If there is a large amount of data, there is no need for more training data, and the training cost - time - also needs to be increased by K times. The total amount of data is large in this project. So maybe we can try The Validation Set Approach, which is the easiest and easy to think of, and because of the large amount of data, the final trained model will work very well. Why not use LOOCV because it has the obvious disadvantage of too much computation.

2.LightGBM

LightGBM uses the histogram algorithm. The basic idea is to first discretize the continuous floating-point eigenvalues into k integers and construct a histogram with a width of K. When traversing the data, the histogram accumulates the required statistics based on the discretized values as indexes in the histogram. After traversing the data once, the histogram traverses to find the optimal split point based on the discrete values of the histogram. Advantages are: faster training speed, lower memory consumption, better accuracy, distributed support, fast processing of large amounts of data, as the current hottest model, there is no room for improvement

3.LSTM, It successfully solves the drawbacks of the original recurrent neural network, becomes the most popular RNN at present, and is successfully applied in many fields such as speech recognition, picture description, natural language processing, etc. There is no doubt that LSTM and GRU and their derivatives can remember a lot of longer-term information! But they can only remember 100 orders of magnitude, not 1000 or longer. Our sequence magnitude is obviously very large in this project, which is not appropriate from this point of view and training them for hardware is very demanding. Perhaps we can think of a different way, abandoning LSTM and using the Attention model because it runs faster and can handle longer sequences.