Data Mining-HW1 report

109350008　張詠哲

(Every experiment result is trained by same hyperparameter.　And the metrics is MSE on validation data instead of RMSE)

1. **How do you select features for your model input, and what preprocessing did you perform?**

   - **preprocessing**

   I implement 3 preprocessing method: turn the data into input form, filled nan value by specific way and normalize the data.

   For turn data into input form

   As the prediction require (use the previous 9 hours data to predict the next hour PM2.5 value). I turn the data into such input form. Every feature divide into 0~8 hours and the 10th hour value of PM2.5 as the ground truth (Picture 1). And I also create a data about per feature every day every hour (Picture 2). Schematic diagram as following picture.

	AMB_TEMP0	AMB_TEMP1	AMB_TEMP2	AMB_TEMP3	AMB_TEMP4	AMB_TEMP5	\
0	11.1	11.2	11.4	11.5	11.6	11.7	
1	11.2	11.4	11.5	11.6	11.7	11.9	
2	11.4	11.5	11.6	11.7	11.9	12.1	
3	11.5	11.6	11.7	11.9	12.1	12.7	
4	11.6	11.7	11.9	12.1	12.7	13.9	

	AMB_TEMP6	AMB_TEMP7	AMB_TEMP8	CH40	...	WS_HR0	WS_HR1	WS_HR2	WS_HR3	\
0	11.9	12.1	12.7	2.01	...	2.6	2.4	2.5	2.5	
1	12.1	12.7	13.9	1.99	...	2.4	2.5	2.5	2.1	
2	12.7	13.9	14.9	2.00	...	2.5	2.5	2.1	2.1	
3	13.9	14.9	15.7	2.02	...	2.5	2.1	2.1	2.1	
4	14.9	15.7	16.1	2.03	...	2.1	2.1	2.1	2.6	

	WS_HR4	WS_HR5	WS_HR6	WS_HR7	WS_HR8	ground_truth
0	2.1	2.1	2.1	2.6	2.6	11.0
1	2.1	2.1	2.6	2.6	3.1	10.0
2	2.1	2.6	2.6	3.1	3.0	16.0
3	2.6	2.6	3.1	3.0	2.9	13.0
4	2.6	3.1	3.0	2.9	3.1	15.0

Picture 1. Input data form

	AMB_TEMP	CH4	CO	NMHC	NO	NO2	NOx	O3	PM10	PM2.5	RAINFALL	\
0	11.1	2.01	0.31	0.10	1.5	11.9	13.5	21.6	38.0	25.0	0.0	
1	11.2	1.99	0.28	0.10	1.4	10.4	11.9	25.1	29.0	24.0	0.0	
2	11.4	2.00	0.28	0.08	1.4	9.8	11.2	25.6	27.0	13.0	0.0	
3	11.5	2.02	0.33	0.09	1.5	12.1	13.7	22.4	24.0	14.0	0.0	
4	11.6	2.03	0.32	0.10	1.4	12.4	13.9	21.1	29.0	15.0	0.0	

	RH	SO2	THC	WD_HR	WIND_DIREC	WIND_SPEED	WS_HR
0	64.0	1.327273	2.11	38.0	53.0	3.0	2.6
1	65.0	2.100000	2.09	41.0	46.0	3.4	2.4
2	63.0	2.100000	2.08	49.0	43.0	2.7	2.5
3	63.0	1.800000	2.11	54.0	54.0	3.0	2.5
4	63.0	1.100000	2.13	50.0	50.0	2.6	2.1

Picture 2. Per feature data

For the filled nan values

Consider the nature of data. Because it's about air quality of per day per hour. Therefore, I don't choose to fill the nan value by average value of per feature (for example. average of "AMB_TEMP" every day every hours). Because it may various in different day due to some weather factor, such as monsoons and raining. If I consider other day's value in average may cause bigger bias. Therefore, I choose to filled the nan values by average of other hours in that nan value day. I've also experiment to fill the nan values by average of 1 hour before and 1 hour after value, but the result is not better that previous method. Table 1 is the experiment result.

Filled nan method	MSE
By every day every hour	16.73
By that day other hour (better result)	15.28
By that day and nan value 1 hour before and 1 hour after	15.71

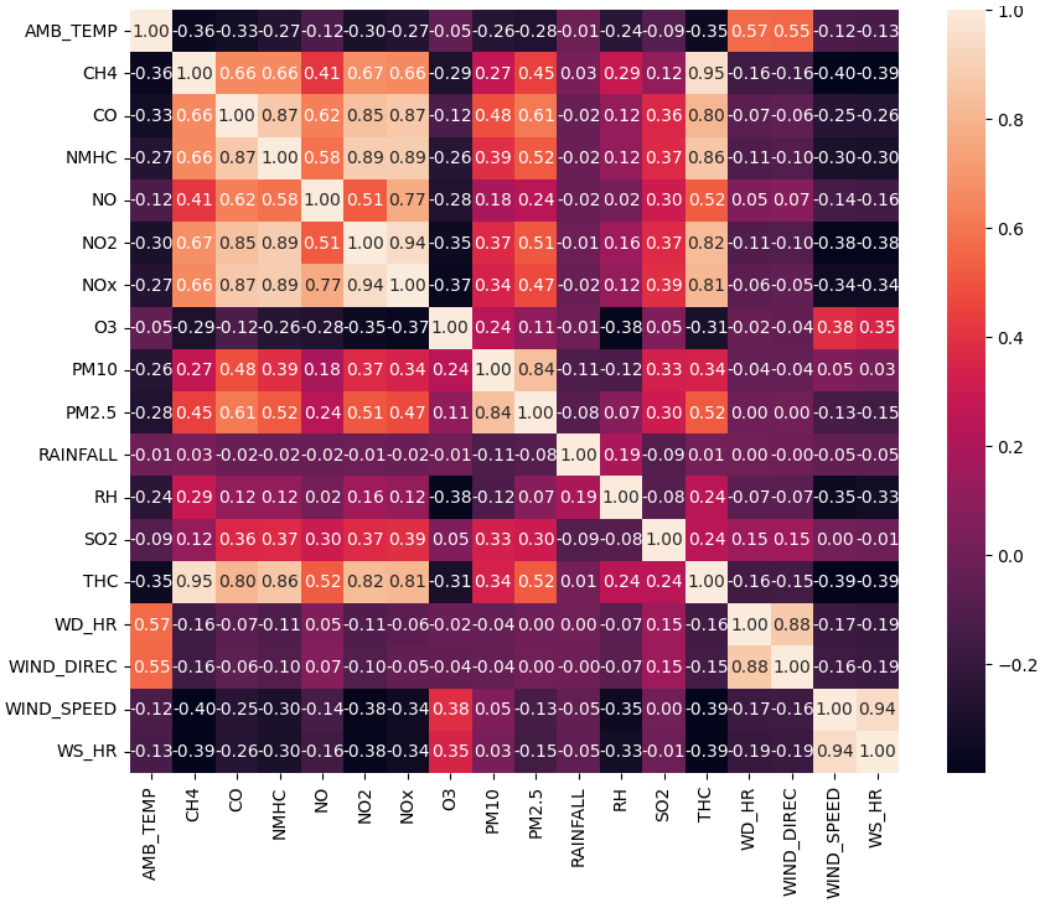
Table 1. result of different complement method

For normalization

I implement the standard scaler, which means that standardizes features by centering them around zero (removing the mean) and scaling them to have unit variance (dividing by the standard deviation). To ensures that all features are on the same scale

- **Feature selection**

I first calculate the Pearson correlation between every feature by per feature data (Picture 3). Then choose the absolute correlation between every feature and PM2.5 (because I have to predict the PM2.5) >= 0.4 (|r| >= 0.4) as the selected feature , then use the same threshold (|r| >= 0.4) to select the hours between every selected feature and the ground truth (use the input form data). And the final feature I choose are ['CH46', 'CH47', 'CH48', 'CO0', 'CO1', 'CO2', 'CO3', 'CO4', 'CO5', 'CO6', 'CO7', 'CO8', 'NMHC5', 'NMHC6', 'NMHC7', 'NMHC8', 'NO24', 'NO25', 'NO26', 'NO27', 'NO28', 'NOx5', 'NOx6', 'NOx7', 'NOx8', 'PM100', 'PM101', 'PM102', 'PM103', 'PM104', 'PM105', 'PM106', 'PM107', 'PM108', 'PM2.50', 'PM2.51', 'PM2.52', 'PM2.53', 'PM2.54', 'PM2.55', 'PM2.56', 'PM2.57', 'PM2.58', 'THC4', 'THC5', 'THC6', 'THC7', 'THC8']. As for the threshold, I also experiment other threshold value, and the 0.4 results in better result. Experiment result in table 2



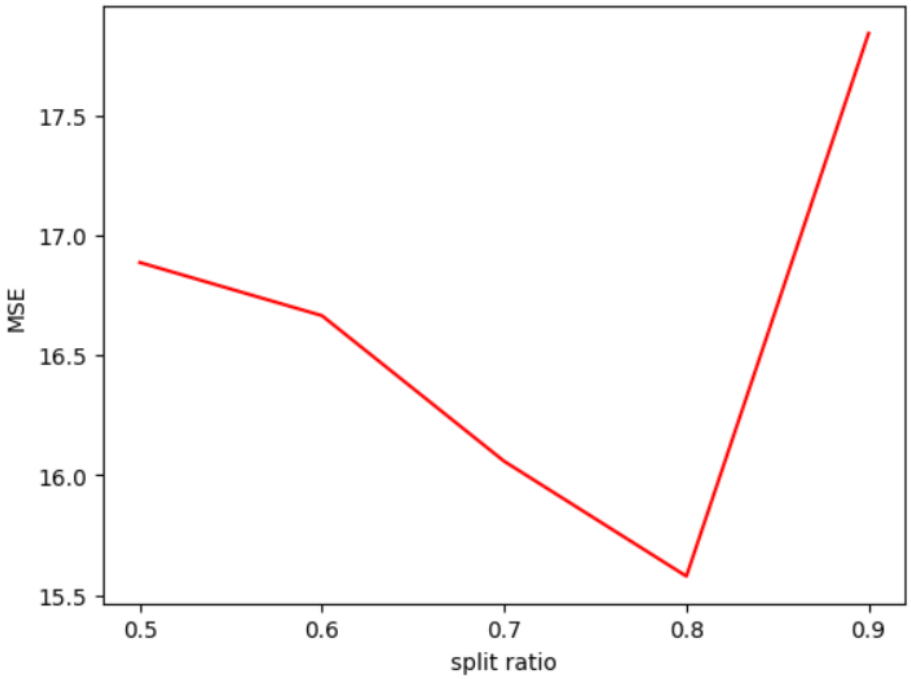
Picture 3. Pearson correlation for different feature

Threshold	MSE
0.3	15.56
0.4	15.28
0.5	15.63

Table 2. Different correlation threshold result

2. Compare the impact of different amounts of training data on the PM2.5 prediction accuracy. Visualize the results and explain them.

In this part. I experiment 5 amounts of training data: 0.5, 0.6, 0.7, 0.8, 0.9 ratio of total training data. And Picture 4 are the experiment result.



Picture 4. Different split ratio MSE result in training data

As the above result picture shows. Split ratio in 0.7 gets the best result. I think the reason is that the more data (0.5~0.8), the model can learn a more representative set of patterns and features from the data. Therefore, results in better result. But if it takes too much data (0.8~0.9) may leads to overfitting, results in a worse performance.

3. **Discuss the impact of regularization on PM2.5 prediction accuracy.**

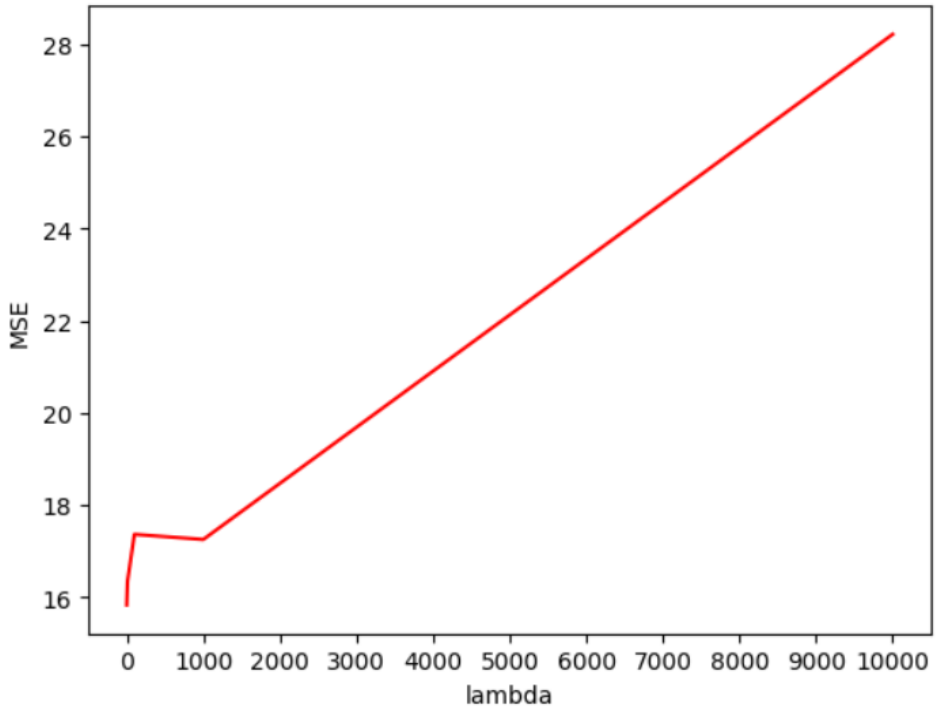
I implement the regularization of linear regression as the teacher's slide.

$$y = b + \sum w_i x_i$$

The functions with smaller w_i are better

$$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i \right) \right)^2 \boxed{+\lambda \sum (w_i)^2}$$

And in this part. I experiment 5 regularization term (lambda) 0, 10, 100, 1000, 10000 And Picture 5 are the visualization of experiment result.



Picture 5. Different lambda result

As the above result picture shows. When lambda = 0 gets the best result and the MSE gets higher when lambda increase. This trend indicates that stronger regularization leads to higher prediction errors in this case. Lower lambda values such as lambda = 0, result in less regularization, allowing the model to fit the training data closely but increasing the risk of overfitting.

And for generally the choice of lambda influences the model's ability to generalize to new data. Too much regularization (high lambda) may lead to underfitting, while too little regularization (low lambda) may result in overfitting.