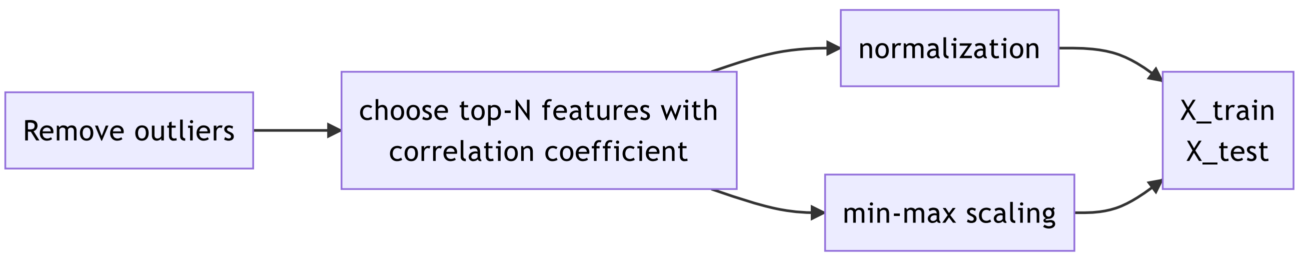
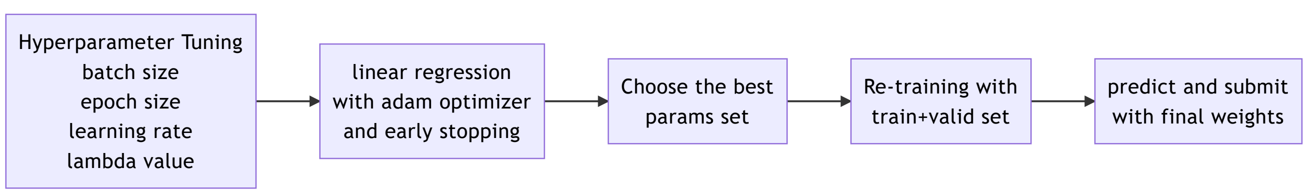
EEML HW1 - Programming Report

B06702064 會計五 林聖硯

Data preprocessing pipeline



Modeling pipeline



Tasks' details

任務	說明
Remove outliers	使用outliers的定義 $[\mu - 1.5IQR, \mu + 1.5IQR]$ 移除y值(第十小時PM2.5)過高之observation
choose top-N features with correlation coefficient (N = **15** by try and error)	使用Pearson correlation coefficient 選出與y最高的前N個features
Hyperparameter tuning	Grid search 以下參數組合 batch size= [64, 128, 256, 512, 1024] epoch size = [20, 30, 50, 100] learning rate = [0.002, 0.005, 0.01, 0.025, 0.05] lambda= [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5]

請實作以下兩種不同feature的模型，回答第 1 ～ 2 題：

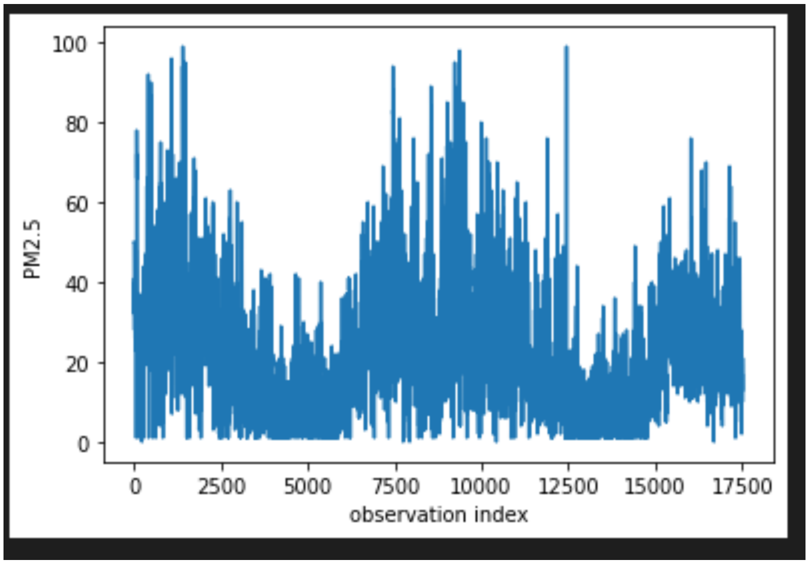
- **Dataset1**: 抽全部9小時內的污染源feature當作一次項(加bias)
- **Dataset2**: 抽全部9小時內pm2.5的一次項當作feature(加bias)
- 以上兩個dataset，除了**choose top-N features with correlation coefficient**此流程外，都有經過**其他所有**的process(包含調整參數等等)，以下問題將由模型調參後的結果回答

1.記錄誤差值 (RMSE)(根據kaggle public+private分數)，討論兩種feature的影響

dataset	Tuning result	validation RMSE	public RMSE	private RMSE
Dataset1	Batch Size=128 Epoch Size=100 (running 93 epoch) learning rate=0.005 lambda=0.001	0.655284	16.20125	17.11370
Dataset2	Batch Size=256 Epoch Size=100 (running 52 epoch) learning rate=0.025 lambda=0.001	4.735835	5.23247	4.87396

Observation:
從第一個Dataset上可發現，使用全部的features當做資料進行訓練，雖然在validation set上表現得很好，但卻沒有辦法generalize到public test/private test set上，顯示模型因為過多的feature而有**overfitting**的情況出現。而在Dataset2上，雖然validation RMSE的結果沒有那麼好，但模型generalize的能力相較Dataset1就強上許多。可能的原因是PM2.5本來就是一個有週期性的時間序列資料，所以只要模型有學出這件事情，用前九個小時的PM2.5來預估第十小時的PM2.5自然準確度就會高，加上其他features對模型來說就是增加額外的noise，反而會導致模型學不好這件事。

移除outlier後，PM2.5在不同小時畫出來之折線圖



2. 解釋什麼樣的data preprocessing可以improve你的training/testing accuracy，e.g., 你怎麼挑掉你覺得不適合的data points。請提供數據(RMSE)以佐證你的想法。

- 首先，我對y值進行EDA，並且發現了在PM2.5=1000時以及PM2.5=100時有兩個斷層。故以下的資料處理步驟，我有針對這兩點特別進行處理。

```python
print(data_train[data_train['PM2.5'] > 5000].shape)
print(data_train[data_train['PM2.5'] > 4000].shape)
print(data_train[data_train['PM2.5'] > 3000].shape)
print(data_train[data_train['PM2.5'] > 2000].shape)
print(data_train[data_train['PM2.5'] > 1000].shape)
#斷層出現在這裡
print(data_train[data_train['PM2.5'] > 800].shape)
print(data_train[data_train['PM2.5'] > 600].shape)
print(data_train[data_train['PM2.5'] > 400].shape)
print(data_train[data_train['PM2.5'] > 200].shape)
print(data_train[data_train['PM2.5'] > 100].shape)
#第二個斷層
print(data_train[data_train['PM2.5'] > 50].shape)
```

✓ 0.5s

```
(3, 18)
(3, 18)
(3, 18)
(5, 18)
(5, 18)
(32, 18)
(86, 18)
(86, 18)
(87, 18)
(98, 18)
(967, 18)
```

- 接著我分別對模型進行以下的資料前處理之組合

資料處理之不同組合	**validation RMSE**	**public RMSE**	**private RMSE**
不進行任何處理	0.655284	16.20125	17.11370
移除y>1000的observation	1.322826	16.06233	16.80802
移除y為outlier之observation	1.422711	16.30997	17.18779
移除y為outlier之observation + normalization	0.655284	16.20125	17.11370
移除y為outlier之observation + min-max scaling	0.683863	16.12268	16.98989
移除y為outlier之observation + choose top 15 features with correlation coefficient	4.94098	5.42117	4.91746
移除y為outlier之observation + choose top 15 features with correlation coefficient + normalization	4.581977	5.48004	4.78525
移除y為outlier之observation + choose top 15 features with correlation coefficient + min-max scaling	4.657251	4.97667	4.76087

Observation

- 從上述圖表中可以看出即使移除outliers加上normalization或min-max scaling，都會讓整個模型overfir validation dataset
- 真正能讓模型在out-sample中大幅進步的關鍵因素為**去掉不重要的features**，做了這步之後，我的模型基本上在training、validation、public/private的資料集上表現不會差太多。原因如上所述，有一些不重要的features加進來訓練可能會增加整個任務的noise，導致模型沒辦法正確的學習

3.一些模型訓練的心得

其實我是public dataset的第一名，我最終選了兩個public score最高的submission繳交成private，結果在private名次竟然掉到了十一名。其實public score最高的兩個model都是我在意外之中(模型還沒完全建立的過程中)做出來的，我還很懊悔沒辦法reproduce他們。結果我明明

有private score更好的模型(也是在模型架好之後做的)，只是因為他們的public score不夠高，我就沒有選擇他們。真的不能完全依賴public score來判斷模型好壞，也要考慮你在model training的時候多了或少了什麼步驟。

Submission and Description	Private Score	Public Score	Use for Final Score
submission_1013_2.csv 9 days ago by b06702064_Martin 1013_2	5.12013	4.64726	☑
submission_1014_4.csv 8 days ago by b06702064_Martin 1014 remove outliers, top 15 corr features / tuning batch size, epoch size, learning rate / L2 normalization	4.84971	4.84708	☑
submission_1020_3.csv 2 days ago by b06702064_Martin 1020 remove outliers, top 15 corr features, normalization / tuning batch size, epoch size, learning rate / randomize weight and bias, early stopping, do not train with all data finally / L2 normalization	4.70913	4.89540	☐
submission_1020_5.csv 2 days ago by b06702064_Martin 1020 remove outliers, top 15 corr features, min-max scaling / tuning batch size, epoch size, learning rate / randomize weight and bias, early stopping, do not train with all data finally / L2 normalization	4.73896	4.91783	☐
submission_1016_1.csv 6 days ago by b06702064_Martin 1016 remove outliers, top 15 corr features / tuning batch size, epoch size, learning rate / do not traing with all data / L2 normalization	4.70995	4.95081	☐

Problem 1 - Logistic Regression

(a)

$$
\begin{aligned}
P_{w,b}(C_1|x) &= \sigma(\sum_{1}^{4} w_i x_i + b) \\
&= \sigma(-7 + 0 - 3 + 50 + 1) \\
&= \frac{1}{1 + \exp(-41)} \\
&\approx 1
\end{aligned}
$$

(b)

$$
\begin{aligned}
w^*, b^* = \arg\max_{w,b} L(w,b) &\iff w^*, b^* = \arg\min_{w,b} -L(w,b) \\
-\ln L(w,b) &= -\ln f_{w,b}(x^1) - \ln f_{w,b}(x^2) - \ln(1 - f_{w,b}(x^3)) \cdots - \ln f_{w,b}(x^N) \\
&= -[1 \times \ln f_{w,b}(x^1) + 0 \times \ln(1 - f_{w,b}(x^1))] - [1 \times \ln f_{w,b}(x^2) + 0 \times \ln(1 - f_{w,b}(x^2))] \\
\ldots&[1 \times \ln f_{w,b}(x^N) + 0 \times \ln(1 - f_{w,b}(x^N))] \\
&= \sum_{n=1}^{N} -[\hat{y}^n \ln f_{w,b}(x^n) + (1 - \hat{y}^n) \ln(1 - f_{w,b}(x^n))]
\end{aligned}
$$

The above equation can be viewed as calculating cross entropy between two Bernoulli distribution P and Q, where $Prob(x^n \in class1|P) = \hat{y}^n$, $Prob(x^n \in class2|P) = 1 - \hat{y}^n$ and $Prob(x^n \in class1|Q) = f(x^n)$, $Prob(x^n \in class2|Q) = 1 - f(x^n)$

(c)

$$
\begin{aligned}
\frac{\partial \ln f_{w,b}(x)}{\partial w_i} &= \frac{\partial \ln f_{w,b}(x)}{\partial z} \times \frac{\partial z}{\partial w_i}, \text{ where } f_{w,b}(x) = \sigma(z), z = \sum_{i=1}^{4} x_i w_i + b \\
&= \frac{1}{\sigma(z)} \times \frac{\partial \sigma(z)}{\partial z} \times x_i \\
&= \frac{1}{\sigma(z)} \times \sigma(z)[1 - \sigma(z)] \times x_i \\
&= [1 - \sigma(z)] x_i \\
\frac{\partial \ln(1 - f_{w,b}(x))}{\partial w_i} &= \frac{\partial \ln(1 - \sigma(z))}{\partial z} \times \frac{\partial z}{\partial w_i} \\
&= -\frac{1}{1 - \sigma(z)} \times \sigma(z)[1 - \sigma(z)] \times x_i \\
&= -\sigma(z) x_i
\end{aligned}
$$

$$
\begin{aligned}
\implies -\frac{\partial \ln L(w,b)}{\partial w_i} &= \sum_{n=1}^{N} -[\hat{y}^n \frac{\partial \ln f_{w,b}(x^n)}{\partial w_i} + (1-\hat{y}^n)\frac{\partial \ln (1-f_{w,b}(x^n))}{\partial w_i}] \\
&= \sum_{n=1}^{N} -[\hat{y}^n [1-f_{w,b}(x^n)]x_i^n - (1-\hat{y}^n) f_{w,b}(x^n) x^n] \\
&= \sum_{n=1}^{N} -[\hat{y}^n - f_{w,b}(x^n)]x_i^n
\end{aligned}
$$

Thus, the update rule is $w_i^{t+1} \leftarrow w_i^t - \eta_t \sum_{n=1}^{N} -[\hat{y}^n - f_{w,b}(x^n)]x_i^n$

Problem 2 - Closed-Form Linear Regression Solution

(a) Since $w \in \mathbb{R}^1$ in this example, denote $w^T = w$

$$
\begin{aligned}
L_{ssq}(w,b) &= \frac{1}{2\times 5} = \sum_{i=1}^{5}(y_i - wx_i - b)^2 \\
\frac{\partial L_{ssq}(w,b)}{\partial w} &= \frac{1}{5}\sum_{i=1}^{5}(y_i - wx_i - b)(-x_i) = 0 \\
&\implies w\sum_{i=1}^{5} x_i^2 + b\sum_{i=1}^{5} x_i = \sum_{i=1}^{5} x_i y_i \\
\frac{\partial L_{ssq}(w,b)}{\partial b} &= \frac{1}{5}\sum_{i=1}^{5}(y_i - wx_i - b)(-1) = 0 \\
&\implies w\sum_{i=1}^{5} x_i + 5b = \sum_{y=i}^{5} y_i
\end{aligned}
$$

From the two equations above, we can get

$$
\begin{cases}
55w + 15b & = 59.7 \\
15w + 5b & = 16.8
\end{cases}
$$

$\implies w = 0.93, b = 0.57$

(b) Let $\beta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} \in \mathbb{R}^{k+1}, \mathbf{X} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathbb{R}^{N\times(k+1)}$, where $x_n = \begin{bmatrix} 1 \\ x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,k} \end{bmatrix} \in \mathbb{R}^{k+1}$

and $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N$

The linear regression model can be represented as $\mathbf{X}\beta = \mathbf{Y}$

$$\begin{aligned}
L(\beta) &= \frac{1}{2N}(\mathbf{X}\beta - \mathbf{Y})^T(\mathbf{X}\beta - \mathbf{Y}) \\
&= \frac{1}{2N}((\mathbf{X}\beta)^T - \mathbf{Y}^T)(\mathbf{X}\beta - \mathbf{Y}) \\
&= \frac{1}{2N}\beta^T\mathbf{X}^T\mathbf{X}\beta - 2(\mathbf{X}\beta)^T\mathbf{Y} - \mathbf{Y}^T\mathbf{Y} \\
\frac{\partial L(\beta)}{\partial \beta} &= \frac{1}{2N}[\mathbf{X}^T\mathbf{X}\frac{\partial \beta^T\beta}{\partial \beta} - 2\mathbf{X}^T\mathbf{Y}\frac{\partial \beta^T}{\partial \beta}] = 0 \\
&\implies (\mathbf{X}^T\mathbf{X})\beta = \mathbf{X}^T\mathbf{Y}
\end{aligned}$$

Assume $\mathbf{X}^T\mathbf{X}$ is invertible,

$$\implies \beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

(c) Following the notation of 2(b)

$$\begin{aligned}
L_{reg}(w,b) &= \frac{1}{2N}\sum_{i=1}^{N}(y_i - (\mathbf{w}^T\mathbf{x}_i + b))^2 + \frac{\lambda}{2}||w||^2 \\
&= \frac{1}{2N}(\mathbf{X}\beta - \mathbf{Y})^T(\mathbf{X}\beta - \mathbf{Y}) + \frac{\lambda}{2}\beta_{-1}^T\beta_{-1}, \text{where } \beta_{-1} = \begin{bmatrix} 0 \\ w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} \in \mathbb{R}^{k+1}
\end{aligned}$$

Compute $\dfrac{\partial \beta_{-1}^T \beta_{-1}}{\partial \beta}$

$$\begin{aligned}\frac{\partial \beta_{-1}^T \beta_{-1}}{\partial \beta} &= \frac{\partial \beta_{-1}}{\partial \beta}\beta_{-1} + \frac{\partial \beta_{-1}}{\partial \beta}\beta_{-1} \\ &= 2 \times \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \beta_{-1} \\ &\stackrel{\text{def}}{=} 2\mathbf{E}^*\beta_{-1}, \text{ where } \mathbf{E}^* \in \mathbb{R}^{(k+1)\times(k+1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial L_{reg}(w,b)}{\partial \beta} &= \frac{1}{2N}[2\mathbf{X}^T\mathbf{X}\beta - 2\mathbf{X}^T\mathbf{Y}] + \lambda \mathbf{E}^*\beta = 0 \\ &\implies \beta(\frac{1}{N}\mathbf{X}^T\mathbf{X} + \lambda\mathbf{E}^*) = \frac{1}{N}\mathbf{X}^T\mathbf{Y}\end{aligned}$$

Assume $\dfrac{1}{N}\mathbf{X}^T\mathbf{X} + \lambda\mathbf{E}^*$ is invertible,

$$\implies \beta = (\frac{1}{N}\mathbf{X}^T\mathbf{X} + \lambda\mathbf{E}^*)^{-1}\frac{1}{N}\mathbf{X}^T\mathbf{Y}$$

Problem3 - Noise and regularization

$$\begin{aligned}\tilde{L}_{ssq}(w,b) &= \mathbb{E}\Big[\frac{1}{2N}\sum_{i=1}^{N}[(\mathbf{w}^T(\mathbf{x}_i + \eta_i)) + b - y_i]^2\Big] \\ &= \frac{1}{2N}\mathbb{E}\Big[\sum_{i=1}^{N}[(\mathbf{w}^T\mathbf{x}_i + b - y_i) + \mathbf{w}^T\eta_i]^2\Big] \\ &= \frac{1}{2N}\sum_{i=1}^{N}\left(\mathbb{E}\Big[(\mathbf{w}^T\mathbf{x}_i + b - y_i)^2\Big] + 2(\mathbf{w}^T\mathbf{x}_i + b - y_i)\mathbb{E}\Big[\mathbf{w}^T\eta_i\Big] + \mathbb{E}\Big[(\mathbf{w}^T\eta_i)^2\Big]\right)\end{aligned}$$

Compute $\mathbb{E}\Big[\mathbf{w}^T\eta_i\Big]$

$$\mathbb{E}\Big[\mathbf{w}^T\eta_i\Big] = \sum_{j=1}^{k} w_j \mathbb{E}[\eta_{i,j}] = 0$$

Compute $\mathbb{E}\Big[(\mathbf{w}^T\eta_i)^2\Big]$

$$\begin{aligned}
\mathbb{E}\Big[(\mathbf{w}^T\eta_i)^2\Big] &= Var[\mathbf{w}^T\eta_i] - (\mathbb{E}[\mathbf{w}^T\eta_i])^2 \\
&= w_1^2 Var[\eta_{i,1}] + w_2^2 Var[\eta_{i,2}] + \cdots + w_k^2 Var[\eta_{i,k}] \\
&= \sum_{j=1}^{k} w_j^2 Var[\eta_{i,j}] \\
&= \sum_{j=1}^{k} w_j^2 \sigma^2 \\
&= \sigma^2 ||w||^2
\end{aligned}$$

$$\begin{aligned}
\tilde{L}_{ssq}(w,b) &= \frac{1}{2N}\sum_{i=1}^{N}\left(\mathbb{E}\Big[(\mathbf{w}^T\mathbf{x}_i + b - y_i)^2\Big] + 2(\mathbf{w}^T\mathbf{x}_i + b - y_i)\mathbb{E}\Big[\mathbf{w}^T\eta_i\Big] + \mathbb{E}\Big[(\mathbf{w}^T\eta_i)^2\Big]\right) \\
&= \frac{1}{2N}\sum_{i=1}^{N}\Big[(f_{w,b}(x_i) - y_i)^2 + \sigma^2||w||^2\Big] \\
&= \frac{1}{2N}\Big[\sum_{i=1}^{N}(f_{w,b}(x_i) - y_i)^2\Big] + \frac{\sigma^2}{2}||w||^2
\end{aligned}$$

Thus, minimizing the expected sum-of-squares loss in the presence of input noise is equivalent to minimizing noise-free sum-of-squares loss with the addition of a L^2-regularization term on the weights.

Problem4 - Kaggle Hacker

(a)

$$\begin{aligned}
e_k &= \frac{1}{N}\sum_{i=1}^{N}(g_k(x_i) - y_i)^2 \\
&= \frac{1}{N}\Big[\sum_{i=1}^{N}(g_k(x_i))^2 - 2\sum_{i=1}^{N} g_k(x_i)y_i + \sum_{i=1}^{N} y_i^2\Big] \\
&= s_k - \frac{2}{N}\sum_{i=1}^{N} g_k(x_i)y_i + e_0 \\
&\implies \frac{2}{N}\sum_{i=1}^{N} g_k(x_i)y_i = s_k + e_0 - e_k \\
&\implies \sum_{i=1}^{N} g_k(x_i)y_i = \frac{N}{2}(s_k + e_0 - e_k)
\end{aligned}$$

(b) Let $\mathbf{G} = \begin{bmatrix} g_1(x_1) & g_2(x_1) & \cdots & g_K(x_1) \\ g_1(x_2) & g_2(x_2) & \cdots & g_K(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(x_N) & g_2(x_N) & \cdots & g_K(x_N) \end{bmatrix} \in \mathbb{R}^{N\times K}$, $\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix} \in \mathbb{R}^K$, $\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N$

$$\min_{\alpha_1,\cdots,\alpha_K} L_{test}\Big(\sum_{k=1}^{K}\alpha_k g_k\Big) = \min_{\alpha_1,\cdots,\alpha_K} \frac{1}{N}(\mathbf{G}\alpha - \mathbf{Y})^T(\mathbf{G}\alpha - \mathbf{Y})$$

$$\frac{\partial L_{test}}{\partial \alpha} = \frac{1}{N}\big[\mathbf{G}^T\mathbf{G}2\alpha - 2\mathbf{G}^T\mathbf{Y}\big] = 0$$

Assume $\mathbf{G}^T\mathbf{G}$ is invertible,

$$\implies \alpha = (\mathbf{G}^T\mathbf{G}^{-1})\mathbf{G}^T\mathbf{Y}$$