**Machine Learning Fall 2020 ——— Homework 1**

學號：B07902037 系級： 資工三 姓名：蔡沛勳

請實做以下兩種不同feature的模型，回答第 (1) ~ (2) 題：

1. 抽全部9小時內的污染源feature當作一次項(加bias)
2. 抽全部9小時內pm2.5的一次項當作feature(加bias)

備註 :   
 a. NR請皆設為0，其他的非數值(特殊字元)可以自己判斷

b. 所有 advanced 的 gradient descent 技術(如: adam, adagrad 等) 都是可以用的

c. 第1-2題請都以題目給訂的兩種model來回答

d. 同學可以先把model訓練好，kaggle死線之後便可以無限上傳。

e. 根據助教時間的公式表示，(1) 代表 p = 9x18+1 而(2) 代表 p = 9\*1+1

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

Preprocess方式：將 data 中壞掉的值轉換成該 label 的平均值，一筆feature中若超過1/5的資料壞掉則刪除該筆資料。

RMSE：

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Train** | **Public Test** | **Private Test** |
| (1) 9 \* 15 + 1 | 5.009809534625709 | 4.87402 | 5.51152 |
| (2) 9 \* 1 + 1 | 6.057614270524903 | 5.43509 | 6.05531 |

可以觀察到兩者皆無法通過Strong baseline，可能原因為preprocess不夠好或feature選擇的問題。兩者比較更可觀察出9 \* 1 + 1個feature的結果又比9 \* 15 + 1個feature的結果還差，可能是feature太少導致model太簡單。而9 \* 15 + 1個feature可能是feature太多導致overfitting (Train 的結果與 Private的結果有段差距)，也可能是feature選得不夠好。

**2. (1%)解釋什麼樣的data preprocessing 可以improve你的training/testing accuracy，ex. 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。**

1. 考慮到每筆feature的值應該有合理的範圍，若超過該範圍則將其約束在其區間中，大於則賦予區間值上限，小於則反之。我對每筆feature取mean及std，並將feature以mean +- k \* std及必須大於0作為區間，測試k在不同值時的表現(以0.5為單位增加)。

(1) Feature：9 \* 15 + 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Range** | **Train** | **Public Test** | **Private Test** |
| mean +- 1 \* std | 5.246385993366273 | 4.82965 | 5.25553 |
| mean +- 1.5 \* std | 4.971691247826385 | 4.74604 | 5.23991 |
| mean +- 2 \* std | 4.952214980016929 | 4.77487 | 5.32179 |
| mean +- 2.5 \* std | 4.978044979068628 | 4.81110 | 5.37307 |
| mean +- 3 \* std | 4.996430850820437 | 4.80838 | 5.39630 |
| No range | 5.009809534625709 | 4.87402 | 5.51152 |

(2) Feature：9 \* 1 + 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Range** | **Train** | **Public Test** | **Private Test** |
| mean +- 1 \* std | 5.550180822382528 | 5.02015 | 5.01809 |
| mean +- 1.5 \* std | 5.308284140484773 | 4.86646 | 4.99397 |
| mean +- 2 \* std | 5.335043224977279 | 4.86088 | 5.01990 |
| mean +- 2.5 \* std | 5.395406882300837 | 4.86714 | 5.03519 |
| mean +- 3 \* std | 5.437311164267323 | 4.87043 | 5.03911 |
| No range | 6.057614270524903 | 5.43509 | 6.05531 |

可以觀察到在規定區間後兩者的表現皆有進步，可見其是有幫助的。可以觀察出對(1)、(2)來說範圍限制在1.5或2個標準差內的表現皆最好，因此以2個標準差為範圍進行下一步preprocess。

2. 接下來可以發現有幾筆資料有許多feature為亂碼或超出範圍，可以懷疑該筆資料可信度較低。因此對每筆資料若包含亂碼或超出範圍就記為一筆error，若error的feature占整筆資料的r%則刪除該筆資料(r以10為單位增加)。此外若label的值是亂碼則直接刪除該筆資料。

(1) Feature：9 \* 15 + 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Error** | **Train** | **Public Test** | **Private Test** |
| Error = 0 | 4.460729620663759 | 4.84918 | 5.30321 |
| Error < =10% | 4.937677492749627 | 4.81639 | 5.37559 |
| Error < =20% | 4.952023018622182 | 4.78377 | 5.32794 |
| Error < =30%v | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =40% | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =50% | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =60% | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =70% | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =80% | 4.952214980016929 | 4.77487 | 5.32179 |
| Error < =90% | 4.952214980016929 | 4.77487 | 5.32179 |
| All | 4.952214980016929 | 4.77487 | 5.32179 |

(2) Feature：9 \* 1 + 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Error rate** | **Train** | **Public Test** | **Private Test** |
| Error = 0 | 4.808685480434029 | 4.91524 | 5.01704 |
| Error < =10% | 4.808685480434029 | 4.91524 | 5.01704 |
| Error < =20% | 4.980062172785758 | 4.90218 | 5.02180 |
| Error < =30%v | 5.041607231690822 | 4.89868 | 5.02396 |
| Error < =40% | 5.082852795921859 | 4.89768 | 5.01517 |
| Error < =50% | 5.121801533441179 | 4.90516 | 5.01099 |
| Error < =60% | 5.188771270025488 | 4.91572 | 4.99808 |
| Error < =70% | 5.259200692583537 | 4.93174 | 4.99727 |
| Error < =80% | 5.334095676114590 | 4.94806 | 5.00054 |
| Error < =90% | 5.3897492106181675 | 4.95714 | 5.00132 |
| All | 5.5501808223825275 | 5.02015 | 5.01809 |

對(1)來說Error<=30% ~90%結果相同，原因為很少有壞的這麼徹底的資料。而Error容忍比率低時在training出的結果有進步，但在test上表現卻更差。對(2)來說刪除Error一定比率上的資料對Test出的結果是有幫助的，差異可能是來自於兩者模型的複雜度不同。可看出Error <= 40%的結果表現較好，推測為一定數量的極端值對training是有幫助的。以上為我所做的preprocess。

**3.(3%) Refer to math problem**

1 - (a).

依題目得到算式：

在 最小值時，對 偏微分等於 0。帶入 得：

由 聯立得到 。

1 - (b).

已知 ，依題目得到算式：

在 最小值時，對 偏微分等於 0，推導出：

將代入得到：

將代回得到：

1 - (c).

依題目得到算式：

在 最小值時，對 偏微分等於 0，對偏微得到上題，對偏微得到：

將代入得到：

將代回得到：

2.

依題目得到算式：

代入後推倒：

第一部分：

第二部分：

第三部分：

因為，所以

三部分總和：

3 - (a).

依題目得到算式：

平方拆開得到：

代入：

得：

3 - (b).

依題目得到算式：

設

得到：

在為最小值時，對偏微分等於0，推得：

在 3 – (a) 推得：

代入得：

設

若存在反矩陣，可推得：