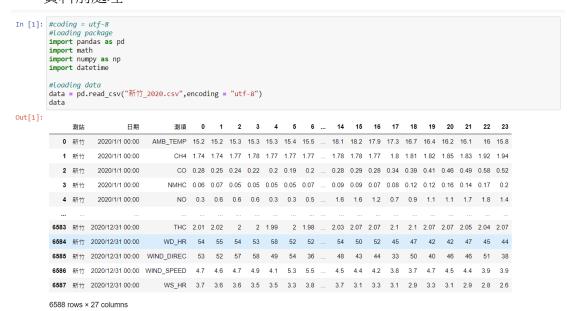
● 資料前處理

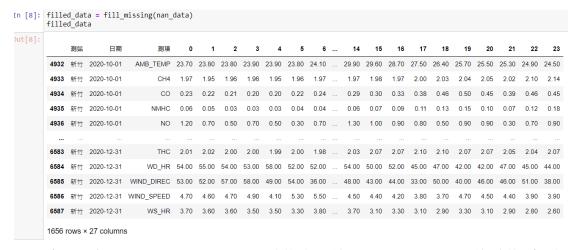


把 csv 檔用 utf-8 編碼後便可以進到 DataFrame 裡面且將 column name 保持中文不亂碼。

```
In [3]: data["日期"]= pd.to_datetime(data["日期"],format ="%Y/%m/%d %H:%M")
        data["日期"]
Out[3]: 0
               2020-01-01
               2020-01-01
        1
               2020-01-01
        2
        3
               2020-01-01
               2020-01-01
        4
              2020-12-31
        6583
        6584
               2020-12-31
               2020-12-31
        6585
        6586
               2020-12-31
               2020-12-31
        6587
        Name: 日期, Length: 6588, dtype: datetime64[ns]
```

將日期照著格式轉換後並透過篩選(10/1-12/31)把需要的一千多筆資料轉存。

	測站	日期	測項	0	1	2	3	4	5	6	 14	15	16	17	18	19	20	21	22	23
4932	新竹	2020-10-01	AMB_TEMP	23.7	23.8	23.8	23.9	23.9	23.8	24.1	29.9	29.6	28.7	27.5	26.4	25.7	25.5	25.3	24.9	24.5
4933	新竹	2020-10-01	CH4	1.97	1.95	1.96	1.96	1.95	1.96	1.97	1.97	1.98	1.97	2	2.03	2.04	2.05	2.02	2.1	2.14
4934	新竹	2020-10-01	CO	0.23	0.22	0.21	0.2	0.2	0.22	0.24	0.29	0.3	0.33	0.38	0.46	0.5	0.45	0.39	0.46	0.45
4935	新竹	2020-10-01	NMHC	0.06	0.05	0.03	0.03	0.03	0.04	0.04	0.06	0.07	0.09	0.11	0.13	0.15	0.1	0.07	0.12	0.18
4936	新竹	2020-10-01	NO	1.2	0.7	0.5	0.7	0.5	0.3	0.7	1.3	1	0.9	8.0	0.5	0.9	0.9	0.3	0.7	0.9
6583	新竹	2020-12-31	THC	2.01	2.02	2	2	1.99	2	1.98	2.03	2.07	2.07	2.1	2.1	2.07	2.07	2.05	2.04	2.07
6584	新竹	2020-12-31	WD_HR	54	55	54	53	58	52	52	54	50	52	45	47	42	42	47	45	44
6585	新竹	2020-12-31	WIND_DIREC	53	52	57	58	49	54	36	48	43	44	33	50	40	46	46	51	38
6586	新竹	2020-12-31	WIND_SPEED	4.7	4.6	4.7	4.9	4.1	5.3	5.5	4.5	4.4	4.2	3.8	3.7	4.7	4.5	4.4	3.9	3.9
0507		0000 40 04	140 110	^ 7	^ ^	^ ^	0.5	0.5	^ ^	^ ^	^ 7	^ 4	^ ^	^ 4	~ ^	^ ^	^ 4	00	^ ^	0.0



寫了一個 function(fill_missing),功能大致就是取 missing value 的前後(直到有值)做平均。但在執行的過程中發現後面某一項會往後找到沒有值(到 12/31 23:00),因此多設了一個限制,超過限定範圍則取前或後有值的為其值。

● 時間序列 比較下列 MAE

```
In [15]: from sklearn.linear_model import LinearRegression from sklearn.metrics import mean_absolute_error regressor = LinearRegression() regressor.fit(x_train, y_train) y_pred = regressor.predict(x_test) mae = mean_absolute_error(y_test,y_pred) #regression,預測一小時後,只看pm2.5特徵 mae
```

Out[15]: 2.5223536456517683

特徵採用僅 pm2.5,使用 Linear Regression,預測一小時後的 pm2.5 值的 MAE。

```
In [16]: from xgboost import XGBClassifier
xgbc = XGBClassifier()
xgbc.fit(x_train,y_train)
y_pred = xgbc.predict(x_test)
#xgboost,預測一小時後,只看pm2.5
mae = mean_absolute_error(y_test,y_pred)
mae

C:\Users\Home\anaconda3\lib\site-packages\xgboosrecated and will be removed in a future release.
se when constructing XGBClassifier object; and 2 lass - 1].
warnings.warn(label_encoder_deprecation_msg, Users)
```

[07:22:14] WARNING: ..\src\learner.cc:1061: Starulti:softprob' was changed from 'merror' to 'mlog

Out[16]: 3.176151761517615

特徵採用僅 pm2.5,使用 XGBoost,預測一小時後的 pm2.5 值的 MAE。

(不知道為甚麼好像 xgboost 在 jupyter notebook 上說即將被刪除)

```
In [19]: regressor = LinearRegression()
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
mae = mean_absolute_error(y_test,y_pred)
#regression,預測六小時後,只看pm2.5特徵
mae
```

Out[19]: 4.579414220758536

特徵採用僅 pm2.5,使用 Linear Regression,預測六小時後的 pm2.5 值的 MAE。

```
In [20]: xgbc = XGBClassifier()
xgbc.fit(x_train,y_train)
y_pred = xgbc.predict(x_test)
#xgboost,預測六小時後,只看pm2.5
mae = mean_absolute_error(y_test,y_pred)
mae
```

[07:22:18] WARNING: ..\src\learner.cc:1061: S ulti:softprob' was changed from 'merror' to '

C:\Users\Home\anaconda3\lib\site-packages\xgb
recated and will be removed in a future relea
se when constructing XGBClassifier object; an
lass - 1].
warnings.warn(label_encoder_deprecation_msg

Out[20]: 5.34174624829468

特徵採用僅 pm2.5,使用 XGBoost,預測六小時後的 pm2.5 值的 MAE。

```
In [23]: regressor = LinearRegression()
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
mae = mean_absolute_error(y_test,y_pred)
#regression,預測一小時後,看全部特徵
mae
```

Out[23]: 2.718420294073232

全部特徵皆採用,使用 Linear Regression,預測一小時後的 pm2.5 值的 MAE。

```
In [24]: xgbc = XGBClassifier()
xgbc.fit(x_train,y_train)
y_pred = xgbc.predict(x_test)
#xgboost,預測一小時後,看全部
mae = mean_absolute_error(y_test,y_pred)
mae
```

[07:22:38] WARNING: ..\src\learner.cc:1061: Star ulti:softprob' was changed from 'merror' to 'mlo

C:\Users\Home\anaconda3\lib\site-packages\xgboos
recated and will be removed in a future release.
se when constructing XGBClassifier object; and 2
lass - 1].
 warnings.warn(label_encoder_deprecation_msg, U

Out[24]: 3.292682926829268

全部特徵皆採用,使用 XGBoost,預測一小時後的 pm2.5 值的 MAE。

```
In [27]: regressor = LinearRegression()
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
mae = mean_absolute_error(y_test,y_pred)
#regression,預測六小時後,看全部特徵
mae
```

Out[27]: 5.749477319528851

全部特徵皆採用,使用 Linear Regression,預測六小時後的 pm2.5 值的 MAE。

```
mae = mean_absolute_error(y_test,y_pred)
mae
```

[07:23:00] WARNING: ..\src\learner.cc:1061: Starting ulti:softprob' was changed from 'merror' to 'mloglos

C:\Users\Home\anaconda3\lib\site-packages\xgboost\sk
recated and will be removed in a future release. To
se when constructing XGBClassifier object; and 2) En
lass - 1].
warnings.warn(label_encoder_deprecation_msg, UserW

Out[28]: 5.066848567530696

全部特徵皆採用,使用 XGBoost,預測六小時後的 pm2.5 值的 MAE。

當預測時間拉長,MAE 皆會變大;但在 XGBoost 跟 Linear Regression 的比較上可能參數調整或資料內容有關,Linear Regression 大致表現的都比較好, XGBoost 只有在預測六小時、採用全特徵時贏過 Linear Regression。