In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from pandas.plotting import lag_plot
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.api import VAR
import statsmodels
import statsmodels.api as sm
from sktime.forecasting.arima import AutoARIMA
import seaborn as sns
import math
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from numpy import mean
from numpy import absolute
from numpy import sqrt
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor
```

In [2]:

```
df = pd.read_csv('./PRSA_data_2010.1.1-2014.12.31.csv')
df
```

Out[2]:

	No	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
0	1	2010	1	1	0	NaN	-21	-11.0	1021.0	NW	1.79	0	0
1	2	2010	1	1	1	NaN	-21	-12.0	1020.0	NW	4.92	0	0
2	3	2010	1	1	2	NaN	-21	-11.0	1019.0	NW	6.71	0	0
3	4	2010	1	1	3	NaN	-21	-14.0	1019.0	NW	9.84	0	0
4	5	2010	1	1	4	NaN	-20	-12.0	1018.0	NW	12.97	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
43819	43820	2014	12	31	19	8.0	-23	-2.0	1034.0	NW	231.97	0	0
43820	43821	2014	12	31	20	10.0	-22	-3.0	1034.0	NW	237.78	0	0
43821	43822	2014	12	31	21	10.0	-22	-3.0	1034.0	NW	242.70	0	0
43822	43823	2014	12	31	22	8.0	-22	-4.0	1034.0	NW	246.72	0	0
43823	43824	2014	12	31	23	12.0	-21	-3.0	1034.0	NW	249.85	0	0

43824 rows × 13 columns

In [3]:

```
df.isna().sum()
```

Out[3]:

```
No          0
year        0
month       0
day         0
hour        0
pm2.5    2067
DEWP        0
TEMP        0
```

```
PRES        0
cbwd        0
Iws         0
Is          0
Ir          0
dtype: int64
```

In [4]:

```
df = df.dropna()
df.dtypes
```

Out[4]:

```
No         int64
year       int64
month      int64
day        int64
hour       int64
pm2.5    float64
DEWP       int64
TEMP     float64
PRES     float64
cbwd      object
Iws      float64
Is         int64
Ir         int64
dtype: object
```

In [5]:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['cbwd'] = le.fit_transform(df['cbwd'])

df = df.iloc[:,1:] # remove row number column
df
```

```
/tmp/ipykernel_96/2253441171.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
ng.html#returning-a-view-versus-a-copy
  df['cbwd'] = le.fit_transform(df['cbwd'])
```

Out[5]:

	year	month	day	hour	pm2.5	DEWP	TEMP	PRES	cbwd	Iws	Is	Ir
24	2010	1	2	0	129.0	-16	-4.0	1020.0	2	1.79	0	0
25	2010	1	2	1	148.0	-15	-4.0	1020.0	2	2.68	0	0
26	2010	1	2	2	159.0	-11	-5.0	1021.0	2	3.57	0	0
27	2010	1	2	3	181.0	-7	-5.0	1022.0	2	5.36	1	0
28	2010	1	2	4	138.0	-7	-5.0	1022.0	2	6.25	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...
43819	2014	12	31	19	8.0	-23	-2.0	1034.0	1	231.97	0	0
43820	2014	12	31	20	10.0	-22	-3.0	1034.0	1	237.78	0	0
43821	2014	12	31	21	10.0	-22	-3.0	1034.0	1	242.70	0	0
43822	2014	12	31	22	8.0	-22	-4.0	1034.0	1	246.72	0	0
43823	2014	12	31	23	12.0	-21	-3.0	1034.0	1	249.85	0	0

41757 rows × 12 columns

In [6]:

```
# convert to datetime
df['datetime'] = pd.to_datetime(df[['year','month','day','hour']], format='%Y/%m/%d, %H')
```

In [7]:

```
df = df.drop(['year','month','day','hour'], axis = 1)
df = df.reset_index()
df = df[['datetime','pm2.5','DEWP','PRES','cbwd','Iws','Is','Ir']]
df
```

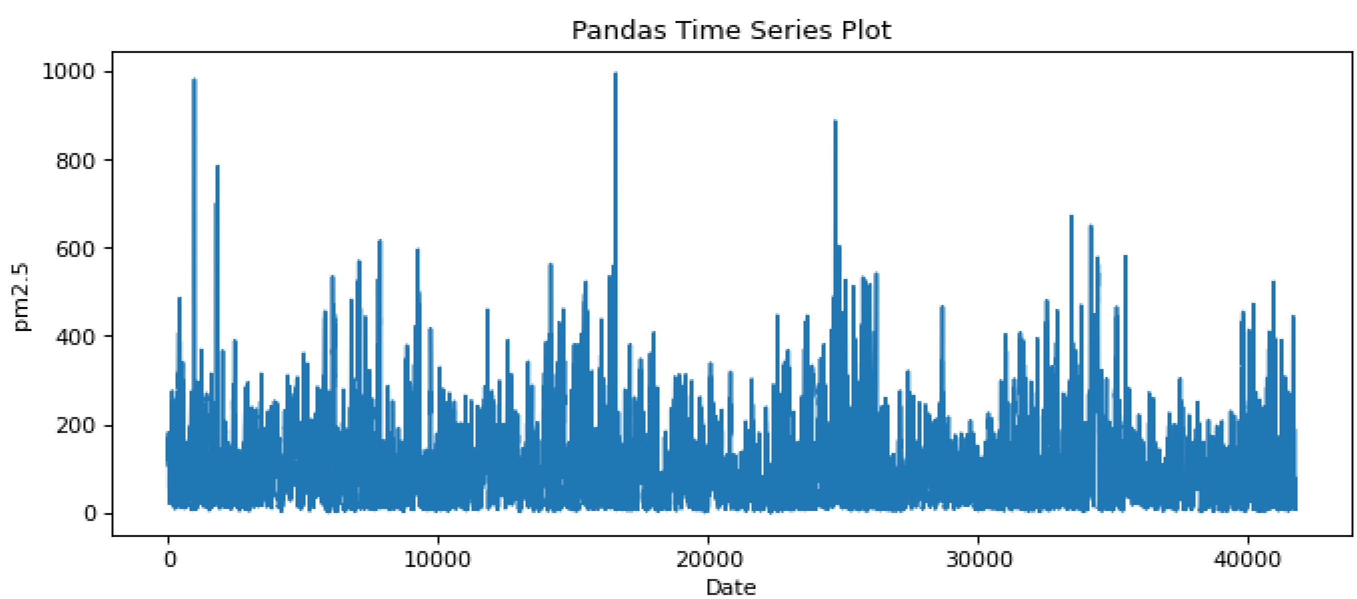
Out[7]:

	datetime	pm2.5	DEWP	PRES	cbwd	Iws	Is	Ir
0	2010-01-02 00:00:00	129.0	-16	1020.0	2	1.79	0	0
1	2010-01-02 01:00:00	148.0	-15	1020.0	2	2.68	0	0
2	2010-01-02 02:00:00	159.0	-11	1021.0	2	3.57	0	0
3	2010-01-02 03:00:00	181.0	-7	1022.0	2	5.36	1	0
4	2010-01-02 04:00:00	138.0	-7	1022.0	2	6.25	2	0
...	...	...	...	...	...	...	...	...
41752	2014-12-31 19:00:00	8.0	-23	1034.0	1	231.97	0	0
41753	2014-12-31 20:00:00	10.0	-22	1034.0	1	237.78	0	0
41754	2014-12-31 21:00:00	10.0	-22	1034.0	1	242.70	0	0
41755	2014-12-31 22:00:00	8.0	-22	1034.0	1	246.72	0	0
41756	2014-12-31 23:00:00	12.0	-21	1034.0	1	249.85	0	0

41757 rows × 8 columns

In [8]:

```
x = df['datetime']
y = df['pm2.5']
plt.figure(figsize = (10, 4), dpi = 80)
plt.xlabel("Date")
plt.ylabel("pm2.5")
plt.title("Pandas Time Series Plot")
plt.plot(range(len(x)), y)
plt.show()
```



Implement Augmented Dickey-Fuller Test

Check if the data is stationary or non-stationary

In [9]:

```
def adf_test(series,title=''):
    """
    Pass in a time series and an optional title, returns an ADF report
    """
    print('Augmented Dickey-Fuller Test: {}'.format(title))
    result = adfuller(series.dropna(),autolag='AIC')
    labels = ['ADF test statistic','p-value','# lags used','# observations']
    out = pd.Series(result[0:4],index=labels)
    for key,val in result[4].items():
        out['critical value ({})'.format(key)]=val
    print(out.to_string())
    if result[1] <= 0.05:
        print("Strong evidence against the null hypothesis")
        print("Reject the null hypothesis")
        print("Data has no unit root and is stationary")
    else:
        print("Weak evidence against the null hypothesis")
        print("Fail to reject the null hypothesis")
        print("Data has a unit root and is non-stationary")
```

In [10]:

```
tmp_df = df.drop('datetime',axis=1)
for item in tmp_df:
    print('ADF test result of {}:'.format(item))
    adf_test(df[item])
    print('---------------------------------------------')
```

```
ADF test result of pm2.5:
Augmented Dickey-Fuller Test:
ADF test statistic        -20.606825
p-value                     0.000000
# lags used                54.000000
# observations          41702.000000
critical value (1%)        -3.430507
critical value (5%)        -2.861609
critical value (10%)       -2.566807
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
---------------------------------------------
ADF test result of DEWP:
Augmented Dickey-Fuller Test:
ADF test statistic     -5.963564e+00
p-value                 2.012344e-07
# lags used             5.500000e+01
# observations          4.170100e+04
critical value (1%)    -3.430507e+00
critical value (5%)    -2.861609e+00
critical value (10%)   -2.566807e+00
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
---------------------------------------------
ADF test result of PRES:
Augmented Dickey-Fuller Test:
ADF test statistic     -7.689118e+00
```

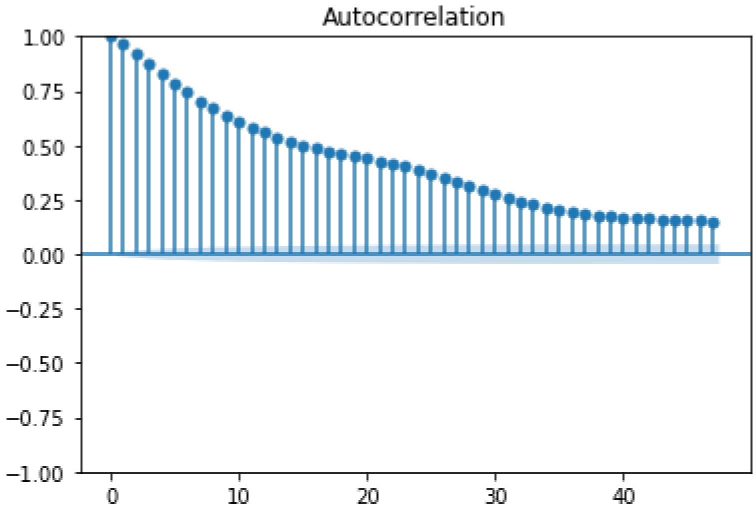
```
p-value                 1.434950e-11
# lags used             5.500000e+01
# observations          4.170100e+04
critical value (1%)    -3.430507e+00
critical value (5%)    -2.861609e+00
critical value (10%)   -2.566807e+00
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
----------------------------------------------
ADF test result of cbwd:
Augmented Dickey-Fuller Test:
ADF test statistic        -23.709927
p-value                     0.000000
# lags used                46.000000
# observations          41710.000000
critical value (1%)        -3.430507
critical value (5%)        -2.861609
critical value (10%)       -2.566807
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
----------------------------------------------
ADF test result of Iws:
Augmented Dickey-Fuller Test:
ADF test statistic        -33.260049
p-value                     0.000000
# lags used                 6.000000
# observations          41750.000000
critical value (1%)        -3.430507
critical value (5%)        -2.861609
critical value (10%)       -2.566807
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
----------------------------------------------
ADF test result of Is:
Augmented Dickey-Fuller Test:
ADF test statistic        -26.706719
p-value                     0.000000
# lags used                55.000000
# observations          41701.000000
critical value (1%)        -3.430507
critical value (5%)        -2.861609
critical value (10%)       -2.566807
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
----------------------------------------------
ADF test result of Ir:
Augmented Dickey-Fuller Test:
ADF test statistic        -33.147777
p-value                     0.000000
# lags used                22.000000
# observations          41734.000000
critical value (1%)        -3.430507
critical value (5%)        -2.861609
critical value (10%)       -2.566807
Strong evidence against the null hypothesis
Reject the null hypothesis
Data has no unit root and is stationary
----------------------------------------------
```

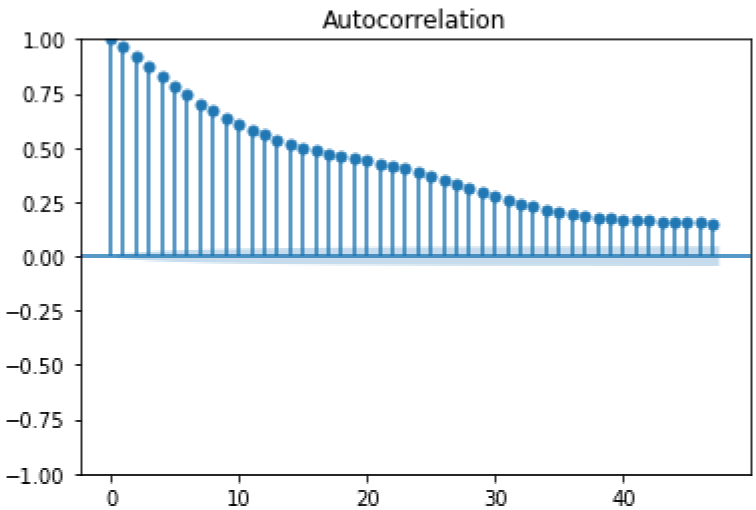
In [11]:

```
from statsmodels.graphics.tsaplots import plot_acf
```

```
plot_acf(df['pm2.5'])
```

Out[11]:





Split the data into train and test

In [12]:

```
feature_col = ['DEWP','PRES','cbwd','Iws','Is','Ir']
target_col = ['pm2.5']

train_size = int(0.70 * len(df))

multivariate_df = df[['datetime'] + target_col + feature_col].copy()
multivariate_df.columns = ['ds', 'y'] + feature_col

train = multivariate_df.iloc[:train_size, :]
X_train, y_train = pd.DataFrame(multivariate_df.iloc[:train_size, [0,2,3,4,5]]), pd.DataFrame(multi
X_test, y_test = pd.DataFrame(multivariate_df.iloc[train_size:, [0,2,3,4,5]]), pd.DataFrame(multiva
```

Modeling

1. VAR

In [13]:

```
df_diff = df.diff().dropna()

X = df_diff[['pm2.5','DEWP','PRES','cbwd','Iws','Is','Ir']]

thresh = int(len(X)*0.3)
```

```
var_train = X[:thresh]
var_test = X[-thresh:]
```

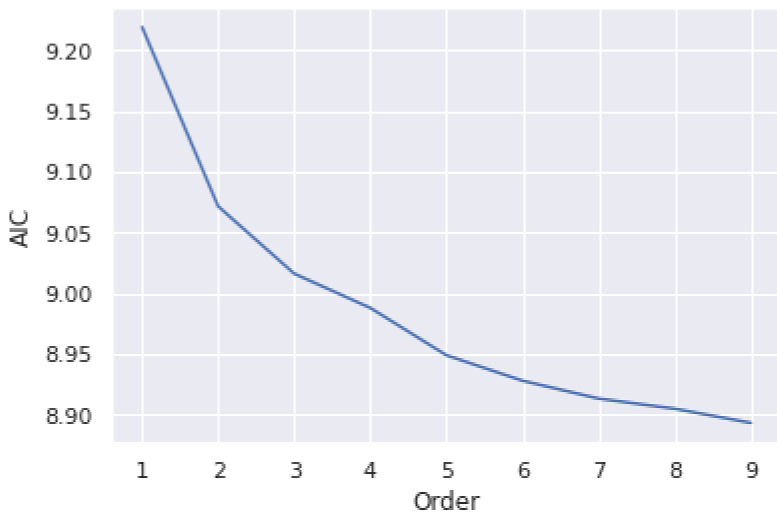
In [14]:

```
var_model = VAR(var_train)
results_aic = []
for p in range(1,10):
    results = var_model.fit(p)
    results_aic.append(results.aic)
```

```
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/statsmodels/tsa/base/tsa_mode
l.py:471: ValueWarning: An unsupported index was provided and will be ignored when e.g. forecasting.
  self._init_dates(dates, freq)
```

In [15]:

```
sns.set()
plt.plot(list(np.arange(1,10,1)), results_aic)
plt.xlabel("Order")
plt.ylabel("AIC")
plt.show()
```



In [16]:

```
result = var_model.fit(10)
result.summary()
```

Out[16]:

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Sun, 17, Apr, 2022
Time:                     07:11:32
--------------------------------------------------------------------
No. of Equations:         7.00000    BIC:                    9.17298
Nobs:                     12516.0    HQIC:                   8.97658
Log likelihood:          -179376.    FPE:                    7170.63
AIC:                      8.87775    Det(Omega_mle):         6892.25
--------------------------------------------------------------------
Results for equation pm2.5
============================================================================
               coefficient       std. error           t-stat            prob
----------------------------------------------------------------------------
const            -0.024082         0.226675           -0.106           0.915
L1.pm2.5          0.103602         0.009028           11.475           0.000
L1.DEWP           1.906238         0.177276           10.753           0.000
L1.PRES          -0.160868         0.329288           -0.489           0.625
L1.cbwd           1.117794         0.287961            3.882           0.000
L1.Iws           -0.009082         0.012823           -0.708           0.479
```

L1.ls	-0.107670	0.555182	-0.194	0.846
L1.lr	-1.315910	0.353962	-3.718	0.000
L2.pm2.5	-0.125506	0.009070	-13.838	0.000
L2.DEWP	1.368481	0.178158	7.681	0.000
L2.PRES	-0.287665	0.332627	-0.865	0.387
L2.cbwd	2.320549	0.350306	6.624	0.000
L2.lws	-0.006908	0.012824	-0.539	0.590
L2.ls	-0.586846	0.554643	-1.058	0.290
L2.lr	-0.398501	0.354849	-1.123	0.261
L3.pm2.5	-0.050508	0.009129	-5.533	0.000
L3.DEWP	0.626113	0.178472	3.508	0.000
L3.PRES	0.162767	0.340949	0.477	0.633
L3.cbwd	2.738358	0.381008	7.187	0.000
L3.lws	-0.009334	0.012820	-0.728	0.467
L3.ls	0.306077	0.554581	0.552	0.581
L3.lr	-0.566542	0.355573	-1.593	0.111
L4.pm2.5	-0.014064	0.009135	-1.540	0.124
L4.DEWP	0.300641	0.178479	1.684	0.092
L4.PRES	-0.508037	0.343555	-1.479	0.139
L4.cbwd	2.598601	0.398817	6.516	0.000
L4.lws	-0.009218	0.012815	-0.719	0.472
L4.ls	0.084089	0.552297	0.152	0.879
L4.lr	-0.054802	0.356186	-0.154	0.878
L5.pm2.5	-0.039359	0.009120	-4.316	0.000
L5.DEWP	0.309611	0.178468	1.735	0.083
L5.PRES	-0.829469	0.342125	-2.424	0.015
L5.cbwd	2.181855	0.406295	5.370	0.000
L5.lws	0.007093	0.012824	0.553	0.580
L5.ls	-0.165686	0.551825	-0.300	0.764
L5.lr	-0.217290	0.356359	-0.610	0.542
L6.pm2.5	-0.059999	0.009115	-6.583	0.000
L6.DEWP	0.426445	0.178209	2.393	0.017
L6.PRES	-0.483912	0.342262	-1.414	0.157
L6.cbwd	1.408980	0.406460	3.466	0.001
L6.lws	-0.002138	0.012816	-0.167	0.868
L6.ls	-0.478160	0.551775	-0.867	0.386
L6.lr	-0.437690	0.356319	-1.228	0.219
L7.pm2.5	-0.038758	0.009127	-4.246	0.000
L7.DEWP	0.349688	0.178004	1.964	0.049
L7.PRES	-0.841832	0.343627	-2.450	0.014
L7.cbwd	1.147445	0.398598	2.879	0.004
L7.lws	0.007531	0.012812	0.588	0.557
L7.ls	-0.378948	0.551673	-0.687	0.492
L7.lr	-0.211165	0.356379	-0.593	0.553
L8.pm2.5	-0.040196	0.009115	-4.410	0.000
L8.DEWP	-0.191893	0.177890	-1.079	0.281
L8.PRES	-0.978205	0.340796	-2.870	0.004
L8.cbwd	0.916606	0.380914	2.406	0.016
L8.lws	-0.021614	0.012808	-1.688	0.091
L8.ls	-0.059416	0.551817	-0.108	0.914
L8.lr	-0.066517	0.355864	-0.187	0.852
L9.pm2.5	-0.054053	0.009042	-5.978	0.000
L9.DEWP	0.216497	0.177604	1.219	0.223
L9.PRES	-0.511924	0.332890	-1.538	0.124
L9.cbwd	0.199000	0.350161	0.568	0.570
L9.lws	-0.004074	0.012817	-0.318	0.751
L9.ls	-0.604236	0.551959	-1.095	0.274
L9.lr	-0.338015	0.355114	-0.952	0.341
L10.pm2.5	-0.045566	0.008954	-5.089	0.000
L10.DEWP	0.256940	0.177269	1.449	0.147
L10.PRES	-0.164221	0.330214	-0.497	0.619
L10.cbwd	0.500988	0.287135	1.745	0.081
L10.lws	-0.004102	0.012809	-0.320	0.749
L10.ls	0.841159	0.552525	1.522	0.128

L10.Ir	-0.272561	0.354510	-0.769	0.442

Results for equation DEWP

	coefficient	std. error	t-stat	prob
const	0.002395	0.011633	0.206	0.837
L1.pm2.5	0.002786	0.000463	6.013	0.000
L1.DEWP	-0.018977	0.009098	-2.086	0.037
L1.PRES	0.015408	0.016899	0.912	0.362
L1.cbwd	0.053523	0.014778	3.622	0.000
L1.Iws	-0.001192	0.000658	-1.811	0.070
L1.Is	0.041654	0.028491	1.462	0.144
L1.Ir	-0.024714	0.018165	-1.361	0.174
L2.pm2.5	0.000286	0.000465	0.615	0.538
L2.DEWP	0.022813	0.009143	2.495	0.013
L2.PRES	-0.038217	0.017070	-2.239	0.025
L2.cbwd	0.064478	0.017977	3.587	0.000
L2.Iws	0.000467	0.000658	0.709	0.478
L2.Is	0.024872	0.028464	0.874	0.382
L2.Ir	0.019165	0.018211	1.052	0.293
L3.pm2.5	0.000723	0.000468	1.544	0.123
L3.DEWP	-0.005496	0.009159	-0.600	0.548
L3.PRES	-0.044352	0.017497	-2.535	0.011
L3.cbwd	0.058995	0.019553	3.017	0.003
L3.Iws	-0.000463	0.000658	-0.703	0.482
L3.Is	0.037586	0.028461	1.321	0.187
L3.Ir	0.024685	0.018248	1.353	0.176
L4.pm2.5	0.000442	0.000469	0.942	0.346
L4.DEWP	-0.024947	0.009159	-2.724	0.006
L4.PRES	-0.055190	0.017631	-3.130	0.002
L4.cbwd	0.101832	0.020467	4.975	0.000
L4.Iws	-0.000610	0.000658	-0.927	0.354
L4.Is	0.014059	0.028343	0.496	0.620
L4.Ir	0.024298	0.018279	1.329	0.184
L5.pm2.5	0.000544	0.000468	1.161	0.246
L5.DEWP	-0.022396	0.009159	-2.445	0.014
L5.PRES	-0.094225	0.017558	-5.367	0.000
L5.cbwd	0.099790	0.020851	4.786	0.000
L5.Iws	-0.000865	0.000658	-1.314	0.189
L5.Is	0.031342	0.028319	1.107	0.268
L5.Ir	-0.009628	0.018288	-0.526	0.599
L6.pm2.5	0.000323	0.000468	0.691	0.490
L6.DEWP	-0.020320	0.009146	-2.222	0.026
L6.PRES	-0.048084	0.017565	-2.738	0.006
L6.cbwd	0.088667	0.020859	4.251	0.000
L6.Iws	0.000262	0.000658	0.398	0.691
L6.Is	0.007593	0.028317	0.268	0.789
L6.Ir	-0.007062	0.018286	-0.386	0.699
L7.pm2.5	0.000748	0.000468	1.596	0.110
L7.DEWP	-0.032781	0.009135	-3.588	0.000
L7.PRES	-0.026008	0.017635	-1.475	0.140
L7.cbwd	0.077752	0.020456	3.801	0.000
L7.Iws	0.000721	0.000658	1.097	0.273
L7.Is	0.005988	0.028311	0.212	0.832
L7.Ir	-0.021477	0.018289	-1.174	0.240
L8.pm2.5	-0.000119	0.000468	-0.255	0.799
L8.DEWP	-0.034094	0.009129	-3.735	0.000
L8.PRES	-0.039699	0.017489	-2.270	0.023
L8.cbwd	0.062279	0.019548	3.186	0.001
L8.Iws	0.000223	0.000657	0.339	0.734
L8.Is	0.031738	0.028319	1.121	0.262
L8.Ir	0.031738	0.018263	1.738	0.082

L9.pm2.5	0.000519	0.000464	1.117	0.264
L9.DEWP	-0.044196	0.009114	-4.849	0.000
L9.PRES	-0.031780	0.017084	-1.860	0.063
L9.cbwd	0.061890	0.017970	3.444	0.001
L9.lws	0.000769	0.000658	1.169	0.242
L9.ls	0.005577	0.028326	0.197	0.844
L9.lr	0.032110	0.018224	1.762	0.078
L10.pm2.5	0.000083	0.000459	0.181	0.856
L10.DEWP	-0.023709	0.009097	-2.606	0.009
L10.PRES	-0.010449	0.016946	-0.617	0.537
L10.cbwd	0.039987	0.014736	2.714	0.007
L10.lws	0.000815	0.000657	1.239	0.215
L10.ls	0.077655	0.028355	2.739	0.006
L10.lr	0.006210	0.018193	0.341	0.733

Results for equation PRES

	coefficient	std. error	t-stat	prob
const	-0.001134	0.006164	-0.184	0.854
L1.pm2.5	0.000169	0.000246	0.687	0.492
L1.DEWP	0.009286	0.004821	1.926	0.054
L1.PRES	0.147288	0.008955	16.447	0.000
L1.cbwd	-0.029895	0.007831	-3.817	0.000
L1.lws	0.000592	0.000349	1.699	0.089
L1.ls	0.006618	0.015098	0.438	0.661
L1.lr	-0.012847	0.009626	-1.335	0.182
L2.pm2.5	-0.000302	0.000247	-1.226	0.220
L2.DEWP	0.010012	0.004845	2.066	0.039
L2.PRES	0.227597	0.009046	25.160	0.000
L2.cbwd	-0.048774	0.009527	-5.120	0.000
L2.lws	0.000852	0.000349	2.444	0.015
L2.ls	-0.023784	0.015084	-1.577	0.115
L2.lr	-0.001266	0.009650	-0.131	0.896
L3.pm2.5	-0.000940	0.000248	-3.785	0.000
L3.DEWP	0.009943	0.004854	2.049	0.040
L3.PRES	0.129774	0.009272	13.996	0.000
L3.cbwd	-0.044869	0.010362	-4.330	0.000
L3.lws	0.000410	0.000349	1.177	0.239
L3.ls	-0.004080	0.015082	-0.271	0.787
L3.lr	0.011782	0.009670	1.218	0.223
L4.pm2.5	-0.000632	0.000248	-2.543	0.011
L4.DEWP	-0.005378	0.004854	-1.108	0.268
L4.PRES	0.008427	0.009343	0.902	0.367
L4.cbwd	-0.038995	0.010846	-3.595	0.000
L4.lws	0.000460	0.000348	1.320	0.187
L4.ls	-0.012208	0.015020	-0.813	0.416
L4.lr	-0.009691	0.009687	-1.000	0.317
L5.pm2.5	-0.000476	0.000248	-1.919	0.055
L5.DEWP	-0.005502	0.004853	-1.134	0.257
L5.PRES	-0.102368	0.009304	-11.002	0.000
L5.cbwd	-0.040067	0.011049	-3.626	0.000
L5.lws	0.000065	0.000349	0.187	0.852
L5.ls	-0.003201	0.015007	-0.213	0.831
L5.lr	-0.012581	0.009691	-1.298	0.194
L6.pm2.5	-0.000591	0.000248	-2.385	0.017
L6.DEWP	-0.004091	0.004846	-0.844	0.399
L6.PRES	-0.095528	0.009308	-10.263	0.000
L6.cbwd	-0.018506	0.011054	-1.674	0.094
L6.lws	0.000048	0.000349	0.137	0.891
L6.ls	-0.019581	0.015006	-1.305	0.192
L6.lr	-0.015441	0.009690	-1.593	0.111
L7.pm2.5	0.000134	0.000248	0.540	0.589

L7.DEWP	-0.007230	0.004841	-1.494	0.135
L7.PRES	-0.024318	0.009345	-2.602	0.009
L7.cbwd	-0.007926	0.010840	-0.731	0.465
L7.Iws	0.000555	0.000348	1.593	0.111
L7.Is	-0.031108	0.015003	-2.073	0.038
L7.Ir	-0.006144	0.009692	-0.634	0.526
L8.pm2.5	0.000024	0.000248	0.096	0.924
L8.DEWP	-0.004717	0.004838	-0.975	0.330
L8.PRES	0.005920	0.009268	0.639	0.523
L8.cbwd	-0.005614	0.010359	-0.542	0.588
L8.Iws	0.001519	0.000348	4.361	0.000
L8.Is	0.003410	0.015007	0.227	0.820
L8.Ir	-0.002347	0.009678	-0.242	0.808
L9.pm2.5	0.000139	0.000246	0.567	0.571
L9.DEWP	0.003628	0.004830	0.751	0.453
L9.PRES	0.060071	0.009053	6.635	0.000
L9.cbwd	-0.004461	0.009523	-0.468	0.639
L9.Iws	0.000599	0.000349	1.718	0.086
L9.Is	0.009227	0.015011	0.615	0.539
L9.Ir	-0.003575	0.009657	-0.370	0.711
L10.pm2.5	-0.000160	0.000243	-0.657	0.511
L10.DEWP	-0.007174	0.004821	-1.488	0.137
L10.PRES	0.064319	0.008980	7.162	0.000
L10.cbwd	-0.001359	0.007809	-0.174	0.862
L10.Iws	0.000505	0.000348	1.450	0.147
L10.Is	-0.000558	0.015026	-0.037	0.970
L10.Ir	-0.010155	0.009641	-1.053	0.292

Results for equation cbwd

	coefficient	std. error	t-stat	prob
const	-0.000188	0.007112	-0.026	0.979
L1.pm2.5	0.000163	0.000283	0.575	0.565
L1.DEWP	0.031327	0.005562	5.632	0.000
L1.PRES	-0.063989	0.010332	-6.194	0.000
L1.cbwd	-0.694013	0.009035	-76.815	0.000
L1.Iws	-0.000880	0.000402	-2.189	0.029
L1.Is	-0.004554	0.017419	-0.261	0.794
L1.Ir	-0.010832	0.011106	-0.975	0.329
L2.pm2.5	-0.000034	0.000285	-0.121	0.904
L2.DEWP	0.014001	0.005590	2.505	0.012
L2.PRES	-0.035769	0.010436	-3.427	0.001
L2.cbwd	-0.522603	0.010991	-47.548	0.000
L2.Iws	-0.000335	0.000402	-0.832	0.406
L2.Is	0.011159	0.017402	0.641	0.521
L2.Ir	-0.017787	0.011134	-1.598	0.110
L3.pm2.5	0.000084	0.000286	0.292	0.770
L3.DEWP	0.009581	0.005600	1.711	0.087
L3.PRES	-0.014258	0.010697	-1.333	0.183
L3.cbwd	-0.428132	0.011954	-35.814	0.000
L3.Iws	-0.000432	0.000402	-1.075	0.282
L3.Is	-0.003529	0.017400	-0.203	0.839
L3.Ir	-0.009470	0.011156	-0.849	0.396
L4.pm2.5	-0.000543	0.000287	-1.894	0.058
L4.DEWP	0.011961	0.005600	2.136	0.033
L4.PRES	0.024875	0.010779	2.308	0.021
L4.cbwd	-0.339156	0.012513	-27.104	0.000
L4.Iws	0.000259	0.000402	0.644	0.519
L4.Is	0.022275	0.017329	1.285	0.199
L4.Ir	-0.021127	0.011175	-1.890	0.059
L5.pm2.5	-0.000426	0.000286	-1.488	0.137
L5.DEWP	0.001267	0.005599	0.226	0.821

```
L5.PRES             0.017716         0.010734            1.650           0.099
L5.cbwd            -0.281147         0.012748          -22.055           0.000
L5.lws              0.000100         0.000402            0.248           0.804
L5.ls               0.035503         0.017314            2.051           0.040
L5.lr              -0.003502         0.011181           -0.313           0.754
L6.pm2.5           -0.000634         0.000286           -2.216           0.027
L6.DEWP            -0.000316         0.005591           -0.056           0.955
L6.PRES             0.005816         0.010739            0.542           0.588
L6.cbwd            -0.224975         0.012753          -17.641           0.000
L6.lws              0.000320         0.000402            0.796           0.426
L6.ls              -0.026913         0.017312           -1.555           0.120
L6.lr              -0.040129         0.011180           -3.589           0.000
L7.pm2.5           -0.000104         0.000286           -0.363           0.717
L7.DEWP            -0.002697         0.005585           -0.483           0.629
L7.PRES            -0.004392         0.010781           -0.407           0.684
L7.cbwd            -0.170432         0.012506          -13.628           0.000
L7.lws              0.000441         0.000402            1.097           0.273
L7.ls               0.003622         0.017309            0.209           0.834
L7.lr              -0.017636         0.011182           -1.577           0.115
L8.pm2.5           -0.000459         0.000286           -1.607           0.108
L8.DEWP             0.007934         0.005581            1.422           0.155
L8.PRES            -0.009176         0.010693           -0.858           0.391
L8.cbwd            -0.132291         0.011951          -11.069           0.000
L8.lws              0.000489         0.000402            1.216           0.224
L8.ls              -0.004985         0.017314           -0.288           0.773
L8.lr              -0.002083         0.011165           -0.187           0.852
L9.pm2.5           -0.000690         0.000284           -2.433           0.015
L9.DEWP            -0.010832         0.005572           -1.944           0.052
L9.PRES            -0.003011         0.010445           -0.288           0.773
L9.cbwd            -0.097122         0.010986           -8.840           0.000
L9.lws             -0.000146         0.000402           -0.363           0.717
L9.ls              -0.001037         0.017318           -0.060           0.952
L9.lr              -0.031402         0.011142           -2.818           0.005
L10.pm2.5          -0.000859         0.000281           -3.059           0.002
L10.DEWP           -0.010646         0.005562           -1.914           0.056
L10.PRES            0.045246         0.010361            4.367           0.000
L10.cbwd           -0.059031         0.009009           -6.552           0.000
L10.lws            -0.000104         0.000402           -0.258           0.796
L10.ls              0.020001         0.017336            1.154           0.249
L10.lr             -0.029262         0.011123           -2.631           0.009
==============================================================================

Results for equation lws
==============================================================================
               coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------
const               0.008689         0.160284            0.054           0.957
L1.pm2.5           -0.008786         0.006384           -1.376           0.169
L1.DEWP            -0.439830         0.125354           -3.509           0.000
L1.PRES             0.032723         0.232844            0.141           0.888
L1.cbwd             0.018631         0.203621            0.091           0.927
L1.lws             -0.001589         0.009067           -0.175           0.861
L1.ls              -0.065718         0.392576           -0.167           0.867
L1.lr              -0.020010         0.250291           -0.080           0.936
L2.pm2.5            0.000579         0.006413            0.090           0.928
L2.DEWP            -0.239086         0.125977           -1.898           0.058
L2.PRES            -0.172198         0.235204           -0.732           0.464
L2.cbwd            -0.014270         0.247706           -0.058           0.954
L2.lws             -0.017425         0.009068           -1.922           0.055
L2.ls              -0.080970         0.392195           -0.206           0.836
L2.lr               0.290149         0.250918            1.156           0.248
L3.pm2.5           -0.005246         0.006455           -0.813           0.416
L3.DEWP            -0.400441         0.126200           -3.173           0.002
L3.PRES             0.304685         0.241089            1.264           0.206
```

```
L3.cbwd           0.048105         0.269415            0.179            0.858
L3.lws           -0.020814         0.009065           -2.296            0.022
L3.ls            -0.092109         0.392151           -0.235            0.814
L3.lr             0.137431         0.251430            0.547            0.585
L4.pm2.5         -0.001618         0.006459           -0.250            0.802
L4.DEWP          -0.313302         0.126205           -2.482            0.013
L4.PRES           0.010500         0.242932            0.043            0.966
L4.cbwd           0.049061         0.282008            0.174            0.862
L4.lws           -0.023244         0.009061           -2.565            0.010
L4.ls            -0.116875         0.390536           -0.299            0.765
L4.lr            -0.024383         0.251863           -0.097            0.923
L5.pm2.5         -0.001627         0.006449           -0.252            0.801
L5.DEWP          -0.492635         0.126197           -3.904            0.000
L5.PRES          -0.116390         0.241921           -0.481            0.630
L5.cbwd           0.086546         0.287296            0.301            0.763
L5.lws           -0.028521         0.009068           -3.145            0.002
L5.ls            -0.214814         0.390202           -0.551            0.582
L5.lr             0.136846         0.251986            0.543            0.587
L6.pm2.5          0.005531         0.006445            0.858            0.391
L6.DEWP          -0.113692         0.126014           -0.902            0.367
L6.PRES          -0.033227         0.242018           -0.137            0.891
L6.cbwd           0.069267         0.287413            0.241            0.810
L6.lws           -0.022144         0.009062           -2.444            0.015
L6.ls            -0.115409         0.390167           -0.296            0.767
L6.lr            -0.130277         0.251958           -0.517            0.605
L7.pm2.5         -0.007261         0.006454           -1.125            0.261
L7.DEWP          -0.295108         0.125869           -2.345            0.019
L7.PRES           0.158800         0.242983            0.654            0.513
L7.cbwd           0.104435         0.281854            0.371            0.711
L7.lws           -0.027873         0.009060           -3.077            0.002
L7.ls            -0.254379         0.390094           -0.652            0.514
L7.lr            -0.074773         0.252000           -0.297            0.767
L8.pm2.5         -0.004520         0.006445           -0.701            0.483
L8.DEWP          -0.236128         0.125789           -1.877            0.060
L8.PRES          -0.289403         0.240981           -1.201            0.230
L8.cbwd           0.095218         0.269349            0.354            0.724
L8.lws           -0.026851         0.009056           -2.965            0.003
L8.ls            -0.298525         0.390197           -0.765            0.444
L8.lr            -0.190435         0.251636           -0.757            0.449
L9.pm2.5          0.004116         0.006394            0.644            0.520
L9.DEWP          -0.117892         0.125586           -0.939            0.348
L9.PRES           0.087272         0.235390            0.371            0.711
L9.cbwd           0.171398         0.247603            0.692            0.489
L9.lws           -0.023524         0.009063           -2.596            0.009
L9.ls             0.275466         0.390297            0.706            0.480
L9.lr             0.039310         0.251106            0.157            0.876
L10.pm2.5        -0.006334         0.006331           -1.000            0.317
L10.DEWP         -0.040537         0.125349           -0.323            0.746
L10.PRES         -0.289801         0.233498           -1.241            0.215
L10.cbwd          0.028743         0.203037            0.142            0.887
L10.lws          -0.031142         0.009058           -3.438            0.001
L10.ls           -0.253037         0.390697           -0.648            0.517
L10.lr            0.069300         0.250679            0.276            0.782
================================================================================

Results for equation ls
================================================================================
               coefficient       std. error           t-stat             prob
--------------------------------------------------------------------------------
const            -0.000162         0.003658           -0.044            0.965
L1.pm2.5         -0.000042         0.000146           -0.287            0.774
L1.DEWP           0.003182         0.002861            1.112            0.266
L1.PRES           0.003440         0.005314            0.647            0.517
L1.cbwd          -0.002955         0.004647           -0.636            0.525
```

L1.lws	0.000117	0.000207	0.567	0.571
L1.ls	-0.014230	0.008959	-1.588	0.112
L1.lr	-0.000901	0.005712	-0.158	0.875
L2.pm2.5	-0.000153	0.000146	-1.043	0.297
L2.DEWP	0.001201	0.002875	0.418	0.676
L2.PRES	-0.006892	0.005368	-1.284	0.199
L2.cbwd	-0.004304	0.005653	-0.761	0.446
L2.lws	0.000000	0.000207	0.002	0.998
L2.ls	-0.026784	0.008950	-2.993	0.003
L2.lr	-0.000134	0.005726	-0.023	0.981
L3.pm2.5	0.000066	0.000147	0.449	0.654
L3.DEWP	0.004265	0.002880	1.481	0.139
L3.PRES	-0.004105	0.005502	-0.746	0.456
L3.cbwd	-0.007419	0.006148	-1.207	0.228
L3.lws	0.000087	0.000207	0.422	0.673
L3.ls	-0.020366	0.008949	-2.276	0.023
L3.lr	-0.000466	0.005738	-0.081	0.935
L4.pm2.5	-0.000112	0.000147	-0.759	0.448
L4.DEWP	0.006400	0.002880	2.222	0.026
L4.PRES	-0.002041	0.005544	-0.368	0.713
L4.cbwd	-0.007163	0.006436	-1.113	0.266
L4.lws	0.000866	0.000207	4.187	0.000
L4.ls	-0.016547	0.008912	-1.857	0.063
L4.lr	-0.000822	0.005748	-0.143	0.886
L5.pm2.5	-0.000071	0.000147	-0.483	0.629
L5.DEWP	0.003794	0.002880	1.317	0.188
L5.PRES	-0.001276	0.005521	-0.231	0.817
L5.cbwd	-0.006887	0.006556	-1.050	0.293
L5.lws	0.000092	0.000207	0.446	0.655
L5.ls	-0.019177	0.008905	-2.154	0.031
L5.lr	-0.000596	0.005750	-0.104	0.917
L6.pm2.5	0.000011	0.000147	0.075	0.940
L6.DEWP	0.003017	0.002876	1.049	0.294
L6.PRES	0.003811	0.005523	0.690	0.490
L6.cbwd	-0.002559	0.006559	-0.390	0.696
L6.lws	0.000053	0.000207	0.257	0.797
L6.ls	-0.040056	0.008904	-4.499	0.000
L6.lr	-0.000351	0.005750	-0.061	0.951
L7.pm2.5	0.000017	0.000147	0.118	0.906
L7.DEWP	0.002395	0.002872	0.834	0.404
L7.PRES	0.000296	0.005545	0.053	0.957
L7.cbwd	-0.001896	0.006432	-0.295	0.768
L7.lws	0.000197	0.000207	0.952	0.341
L7.ls	-0.036219	0.008902	-4.069	0.000
L7.lr	-0.000283	0.005751	-0.049	0.961
L8.pm2.5	-0.000036	0.000147	-0.242	0.809
L8.DEWP	0.001380	0.002871	0.481	0.631
L8.PRES	-0.000102	0.005499	-0.019	0.985
L8.cbwd	-0.001145	0.006147	-0.186	0.852
L8.lws	0.000140	0.000207	0.677	0.498
L8.ls	-0.035389	0.008905	-3.974	0.000
L8.lr	-0.000308	0.005743	-0.054	0.957
L9.pm2.5	-0.000062	0.000146	-0.422	0.673
L9.DEWP	-0.001544	0.002866	-0.539	0.590
L9.PRES	-0.000599	0.005372	-0.112	0.911
L9.cbwd	-0.001462	0.005650	-0.259	0.796
L9.lws	0.000041	0.000207	0.197	0.844
L9.ls	-0.047158	0.008907	-5.295	0.000
L9.lr	-0.001157	0.005730	-0.202	0.840
L10.pm2.5	-0.000068	0.000144	-0.467	0.640
L10.DEWP	-0.000227	0.002861	-0.079	0.937
L10.PRES	0.006013	0.005329	1.128	0.259
L10.cbwd	-0.001136	0.004633	-0.245	0.806
L10.lws	0.000060	0.000207	0.291	0.771

L10.ls	-0.035473	0.008916	-3.979	0.000
L10.lr	-0.000308	0.005721	-0.054	0.957

Results for equation lr

	coefficient	std. error	t-stat	prob
const	-0.000035	0.005733	-0.006	0.995
L1.pm2.5	-0.000318	0.000228	-1.393	0.164
L1.DEWP	-0.002352	0.004483	-0.525	0.600
L1.PRES	0.012300	0.008328	1.477	0.140
L1.cbwd	-0.010578	0.007282	-1.453	0.146
L1.lws	-0.000258	0.000324	-0.796	0.426
L1.ls	-0.000097	0.014040	-0.007	0.994
L1.lr	-0.063841	0.008952	-7.132	0.000
L2.pm2.5	-0.000267	0.000229	-1.163	0.245
L2.DEWP	0.004007	0.004506	0.889	0.374
L2.PRES	0.003567	0.008412	0.424	0.672
L2.cbwd	-0.005349	0.008859	-0.604	0.546
L2.lws	-0.000189	0.000324	-0.582	0.560
L2.ls	-0.001410	0.014027	-0.101	0.920
L2.lr	-0.076131	0.008974	-8.484	0.000
L3.pm2.5	-0.000092	0.000231	-0.399	0.690
L3.DEWP	-0.001942	0.004513	-0.430	0.667
L3.PRES	-0.009311	0.008622	-1.080	0.280
L3.cbwd	-0.001486	0.009636	-0.154	0.877
L3.lws	-0.000059	0.000324	-0.183	0.855
L3.ls	-0.000751	0.014025	-0.054	0.957
L3.lr	-0.066001	0.008992	-7.340	0.000
L4.pm2.5	0.000068	0.000231	0.294	0.769
L4.DEWP	0.002500	0.004514	0.554	0.580
L4.PRES	-0.012540	0.008688	-1.443	0.149
L4.cbwd	0.001675	0.010086	0.166	0.868
L4.lws	-0.000069	0.000324	-0.214	0.830
L4.ls	-0.000495	0.013967	-0.035	0.972
L4.lr	-0.028918	0.009008	-3.210	0.001
L5.pm2.5	0.000301	0.000231	1.303	0.193
L5.DEWP	0.003326	0.004513	0.737	0.461
L5.PRES	-0.000679	0.008652	-0.079	0.937
L5.cbwd	0.000519	0.010275	0.050	0.960
L5.lws	-0.000264	0.000324	-0.813	0.416
L5.ls	-0.000146	0.013955	-0.010	0.992
L5.lr	-0.062109	0.009012	-6.892	0.000
L6.pm2.5	0.000211	0.000231	0.917	0.359
L6.DEWP	0.002078	0.004507	0.461	0.645
L6.PRES	-0.003321	0.008656	-0.384	0.701
L6.cbwd	-0.003255	0.010279	-0.317	0.751
L6.lws	0.000511	0.000324	1.576	0.115
L6.ls	-0.000865	0.013954	-0.062	0.951
L6.lr	-0.004194	0.009011	-0.465	0.642
L7.pm2.5	0.000191	0.000231	0.826	0.409
L7.DEWP	0.002418	0.004502	0.537	0.591
L7.PRES	0.008254	0.008690	0.950	0.342
L7.cbwd	-0.022970	0.010080	-2.279	0.023
L7.lws	0.000023	0.000324	0.072	0.943
L7.ls	-0.000323	0.013952	-0.023	0.982
L7.lr	-0.040104	0.009013	-4.450	0.000
L8.pm2.5	-0.000066	0.000231	-0.288	0.774
L8.DEWP	-0.002931	0.004499	-0.652	0.515
L8.PRES	-0.010426	0.008619	-1.210	0.226
L8.cbwd	-0.024021	0.009633	-2.494	0.013
L8.lws	-0.000179	0.000324	-0.552	0.581
L8.ls	-0.000417	0.013955	-0.030	0.976

```
L8.lr          -0.042658        0.009000        -4.740          0.000
L9.pm2.5       -0.000185        0.000229        -0.807          0.420
L9.DEWP         0.004115        0.004492         0.916          0.360
L9.PRES         0.003542        0.008419         0.421          0.674
L9.cbwd        -0.011686        0.008855        -1.320          0.187
L9.lws          0.000067        0.000324         0.206          0.837
L9.ls          -0.000593        0.013959        -0.042          0.966
L9.lr          -0.019075        0.008981        -2.124          0.034
L10.pm2.5      -0.000179        0.000226        -0.793          0.428
L10.DEWP        0.003645        0.004483         0.813          0.416
L10.PRES        0.002983        0.008351         0.357          0.721
L10.cbwd       -0.006528        0.007262        -0.899          0.369
L10.lws        -0.000127        0.000324        -0.392          0.695
L10.ls         -0.001548        0.013973        -0.111          0.912
L10.lr         -0.065063        0.008965        -7.257          0.000
==========================================================================

Correlation matrix of residuals
            pm2.5      DEWP      PRES      cbwd       lws        ls        lr
pm2.5    1.000000  0.120746 -0.020025  0.050182 -0.025030 -0.006426 -0.007368
DEWP     0.120746  1.000000  0.001463  0.086646 -0.105692  0.011297  0.031616
PRES    -0.020025  0.001463  1.000000 -0.045467 -0.012430 -0.001993  0.016474
cbwd     0.050182  0.086646 -0.045467  1.000000 -0.116305  0.001547 -0.004841
lws     -0.025030 -0.105692 -0.012430 -0.116305  1.000000  0.022205 -0.008694
ls      -0.006426  0.011297 -0.001993  0.001547  0.022205  1.000000 -0.001201
lr      -0.007368  0.031616  0.016474 -0.004841 -0.008694 -0.001201  1.000000
```

In [17]:

```
lagged_values = var_train.values[:]

forecast = pd.DataFrame(results.forecast(y=lagged_values, steps=len(var_test)), index = var_test.ir
forecast
```

Out[17]:

	pm2.5	DEWP	PRES	cbwd	lws	ls	lr
29231	-4.676569	-0.036504	-0.088650	0.006963	-0.441167	0.005068	0.046999
29232	-2.619357	-0.071549	-0.112074	0.015349	-0.844026	0.008171	-0.021045
29233	-0.957402	0.037856	0.029645	-0.088028	-0.774844	0.005257	0.098523
29234	-1.869542	-0.036252	0.049094	0.076275	0.261161	0.002334	-0.052198
29235	-1.748297	0.073814	0.050329	-0.050980	-1.070132	0.002484	0.038854
...	...	...	...	...	...	...	...
41752	-0.000350	0.002572	-0.001822	0.000002	0.002337	-0.000010	0.000014
41753	-0.000350	0.002572	-0.001822	0.000002	0.002337	-0.000010	0.000014
41754	-0.000350	0.002572	-0.001822	0.000002	0.002337	-0.000010	0.000014
41755	-0.000350	0.002572	-0.001822	0.000002	0.002337	-0.000010	0.000014
41756	-0.000350	0.002572	-0.001822	0.000002	0.002337	-0.000010	0.000014

12526 rows × 7 columns

In [18]:

```
forecast['pm2.5'] = df['pm2.5'].iloc[-thresh-1] + forecast['pm2.5'].cumsum()
forecast['DEWP'] = df['DEWP'].iloc[-thresh-1] + forecast['DEWP'].cumsum()
forecast['PRES'] = df['PRES'].iloc[-thresh-1] + forecast['PRES'].cumsum()
forecast['cbwd'] = df['cbwd'].iloc[-thresh-1] + forecast['cbwd'].cumsum()
forecast['lws'] = df['lws'].iloc[-thresh-1] + forecast['lws'].cumsum()
forecast['ls'] = df['ls'].iloc[-thresh-1] + forecast['ls'].cumsum()
```

```
forecast['lr'] = df['lr'].iloc[-thresh-1] + forecast['lr'].cumsum()
forecast
```

Out[18]:

	pm2.5	DEWP	PRES	cbwd	Iws	Is	Ir
29231	56.323431	20.963496	1004.911350	2.006963	22.788833	0.005068	0.046999
29232	53.704073	20.891947	1004.799276	2.022312	21.944808	0.013239	0.025954
29233	52.746671	20.929803	1004.828921	1.934284	21.169964	0.018496	0.124477
29234	50.877129	20.893551	1004.878015	2.010560	21.431125	0.020830	0.072278
29235	49.128832	20.967364	1004.928344	1.959580	20.360993	0.023314	0.111133
...	...	...	...	...	...	...	...
41752	47.329327	53.133435	982.267947	2.024708	49.916958	-0.108059	0.295314
41753	47.328977	53.136007	982.266125	2.024710	49.919295	-0.108069	0.295328
41754	47.328627	53.138578	982.264303	2.024712	49.921633	-0.108079	0.295342
41755	47.328277	53.141150	982.262481	2.024715	49.923970	-0.108089	0.295356
41756	47.327927	53.143722	982.260659	2.024717	49.926307	-0.108099	0.295370

12526 rows × 7 columns

In [19]:

```
score_rmse = mean_squared_error(var_test, forecast, squared = False)
print('RMSE: {}'.format(score_rmse))
```

```
RMSE: 161.1733022466239
```

2. ARIMA

In [20]:

```
from statsmodels.tsa.arima.model import ARIMA
arima_model = ARIMA(y_train, order=(1,1,1))
arima_model = arima_model.fit()
print(arima_model.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                29229
Model:                 ARIMA(1, 1, 1)   Log Likelihood             -135779.968
Date:                Sun, 17 Apr 2022   AIC                         271565.936
Time:                        07:11:34   BIC                         271590.785
Sample:                             0   HQIC                        271573.919
                              - 29229
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.1424      0.008    -17.505      0.000      -0.158      -0.126
ma.L1          0.3256      0.008     43.138      0.000       0.311       0.340
sigma2       634.9759      0.844    752.725      0.000     633.322     636.629
===================================================================================
Ljung-Box (L1) (Q):                   0.09   Jarque-Bera (JB):           9695479.57
Prob(Q):                              0.77   Prob(JB):                         0.00
Heteroskedasticity (H):               0.77   Skew:                            -0.39
Prob(H) (two-sided):                  0.00   Kurtosis:                        92.22
===================================================================================
```

```
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

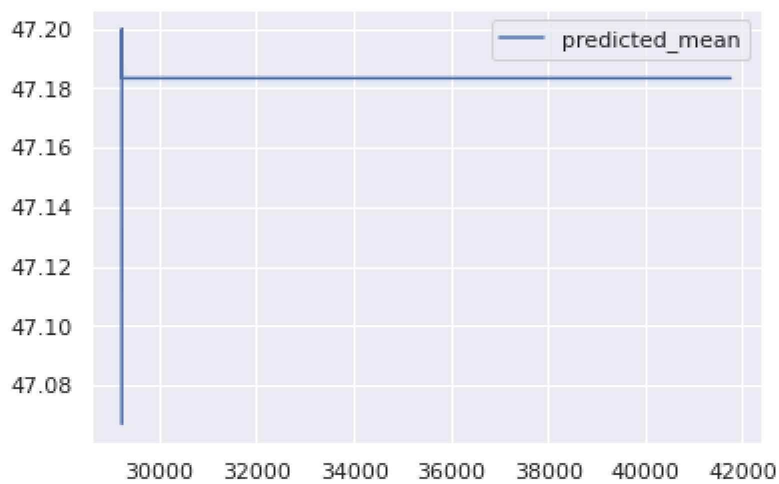
In [21]:

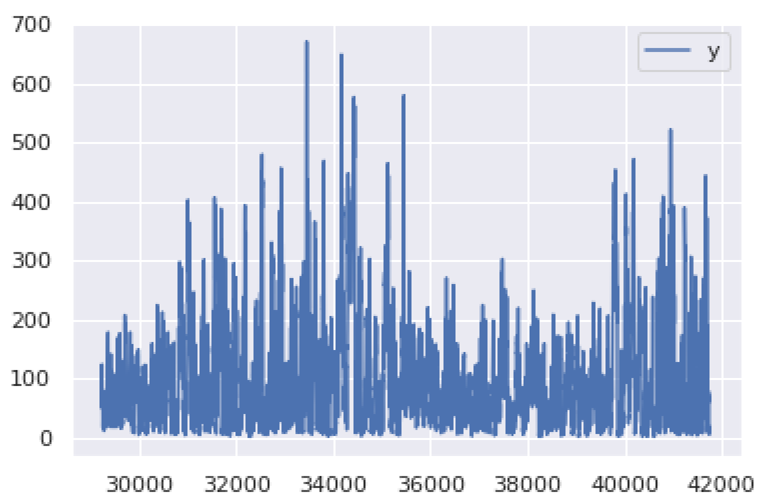
```python
# Predict on test set
# arima_y_pred = arima_model.predict(start=X_train.shape[0],end=(X_train.shape[0]+X_test.shape[0]-1)
arima_y_pred = arima_model.forecast(12528)
# Evaludate the model
score_mae = mean_absolute_error(y_test, arima_y_pred)
score_rmse = mean_squared_error(y_test, arima_y_pred, squared = False)

print('RMSE: {}'.format(score_rmse))
```

```
RMSE: 102.2829744174831
```

In [22]:

```python
arima_y_pred.plot(legend=True)
y_test.plot(legend=True)
plt.show()
```





3. SARIMAX

In [23]:

```python
sari_model = sm.tsa.statespace.SARIMAX(y_train, trend='c', order=(1,1,(1,0,0,1)))
sari_model = sari_model.fit(disp=False)
print(sari_model.summary())
```

```
                               SARIMAX Results
==================================================================================
Dep. Variable:                        y   No. Observations:               29229
Model:                SARIMAX(1, 1, [1, 4])   Log Likelihood             -135779.830
Date:                     Sun, 17 Apr 2022   AIC                         271569.659
```

```
Time:                            07:11:52   BIC                      271611.074
Sample:                                 0   HQIC                     271582.964
                                  - 29229
Covariance Type:                      opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -0.0029      0.205     -0.014      0.989      -0.405       0.399
ar.L1         -0.1440      0.008    -17.105      0.000      -0.160      -0.127
ma.L1          0.3272      0.008     42.001      0.000       0.312       0.342
ma.L4          0.0030      0.004      0.762      0.446      -0.005       0.011
sigma2       634.7316      0.877    724.127      0.000     633.014     636.450
===================================================================================
Ljung-Box (L1) (Q):                   0.08   Jarque-Bera (JB):          9680687.03
Prob(Q):                              0.78   Prob(JB):                        0.00
Heteroskedasticity (H):               0.78   Skew:                           -0.39
Prob(H) (two-sided):                  0.00   Kurtosis:                       92.15
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

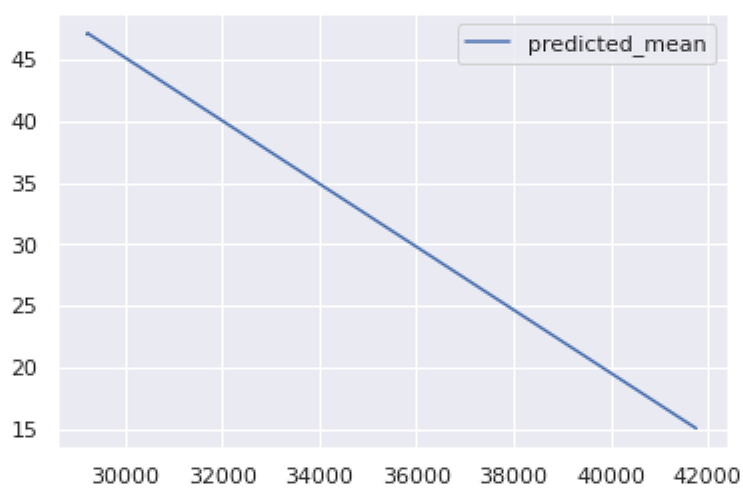
In [24]:

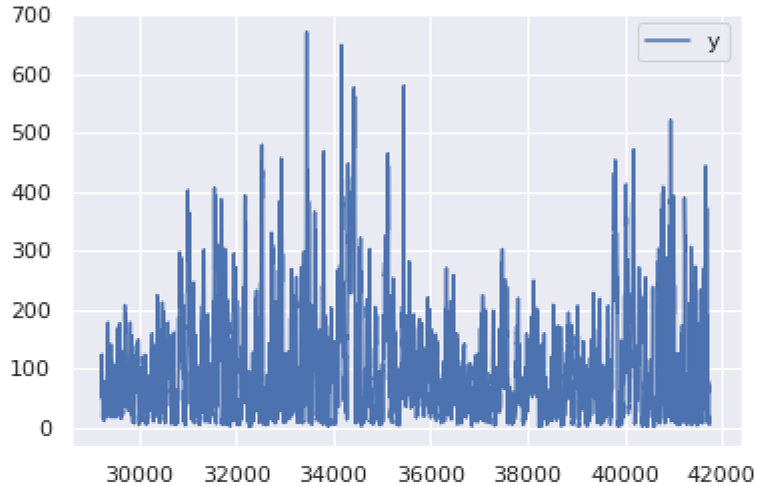
```python
sari_y_pred = sari_model.forecast(12528)

# Evaludate the model
score_mae = mean_absolute_error(y_test, sari_y_pred)
score_rmse = mean_squared_error(y_test, sari_y_pred, squared = False)

print('RMSE: {}'.format(score_rmse))
```

```
RMSE: 111.05757190966203
```

In [25]:

```python
sari_y_pred.plot(legend=True)
y_test.plot(legend=True)
plt.show()
```





4. PROPHET

In [26]:

```
from fbprophet import Prophet
from sklearn.metrics import mean_absolute_error
import math

# Train the model
pro_model = Prophet()
pro_model.add_regressor('DEWP')
pro_model.add_regressor('PRES')
pro_model.add_regressor('cbwd')
pro_model.add_regressor('Iws')
# model.add_regressor('Is')
# model.add_regressor('Ir')

# Fit the model with train set
pro_model.fit(train)

# Predict on valid set
pro_y_pred = pro_model.predict(X_test)

# Calcuate metrics
score_mae = mean_absolute_error(y_test, pro_y_pred['yhat'])
score_rmse = mean_squared_error(y_test, pro_y_pred['yhat'], squared = False)

print('RMSE: {}'.format(score_rmse))
```

```
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/fbprophet/forecaster.py:891: F
utureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
ion. Use pandas.concat instead.
  components = components.append(new_comp)
Initial log joint probability = -139.853
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
      99       63176.4   0.000573654     1489.53           1           1      115
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     199       63247.8   0.000266897     1115.86      0.2338      0.2338      225
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     299       63266.8    0.00696251     1294.64      0.8086      0.8086      333
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     399       63284.2   0.000386751     409.506      0.9821      0.9821      443
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     499       63294.9   0.000726232       599.3      0.3741           1      551
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     599       63297.6    0.00057162     333.159           1           1      656
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes
     699         63304    0.00310836     1136.03      0.3585           1      765
```

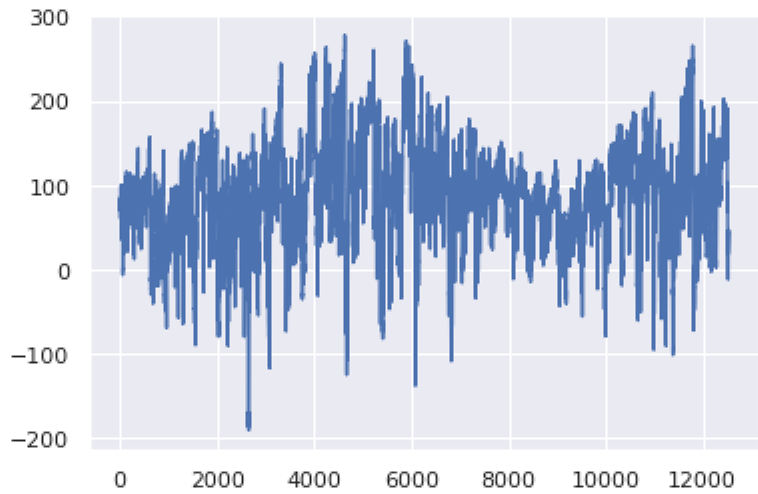
```
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
     799       63308.5    0.00125871       259.461           1           1      874   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
     899       63311.4    0.00030203       403.178           1           1      983   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
     999       63313.2     0.0121493       1806.13           1           1     1088   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1099       63316.9    0.00163247       479.042           1           1     1202   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1199       63317.5   5.04815e-05       133.657      0.8456      0.8456     1324   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1299         63319    0.00173273         645.9           1           1     1430   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1399       63321.8    0.00324804       249.814           1           1     1542   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1499       63323.5   0.000201835       201.136           1           1     1654   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1599       63324.2   0.000560045        120.16      0.9515      0.9515     1771   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1699       63324.4    0.00306173       182.914           1           1     1888   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1799       63325.1    0.00192423       494.422      0.9955      0.9955     1996   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1899       63325.8   6.90815e-05       103.781           1           1     2104   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    1935       63325.9   2.93311e-05        144.13   3.276e-07       0.001     2192  LS failed, Hess
ian reset 
    1999       63325.9   1.41007e-05        79.934      0.5859      0.5859     2273   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    2099       63326.1   1.54181e-05       70.8893      0.3372      0.3372     2386   
    Iter      log prob        ||dx||      ||grad||       alpha      alpha0  # evals  Notes 
    2104       63326.1   3.67136e-06       66.1971       0.339       0.339     2393   
Optimization terminated normally: 
  Convergence detected: relative gradient magnitude is below tolerance
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/fbprophet/forecaster.py:891: F
utureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
ion. Use pandas.concat instead.
  components = components.append(new_comp)
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/fbprophet/forecaster.py:891: F
utureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
ion. Use pandas.concat instead.
  components = components.append(new_comp)
RMSE: 68.81045142066871
```

In [27]:

```
pro_y_pred['yhat'].plot()
plt.show
```

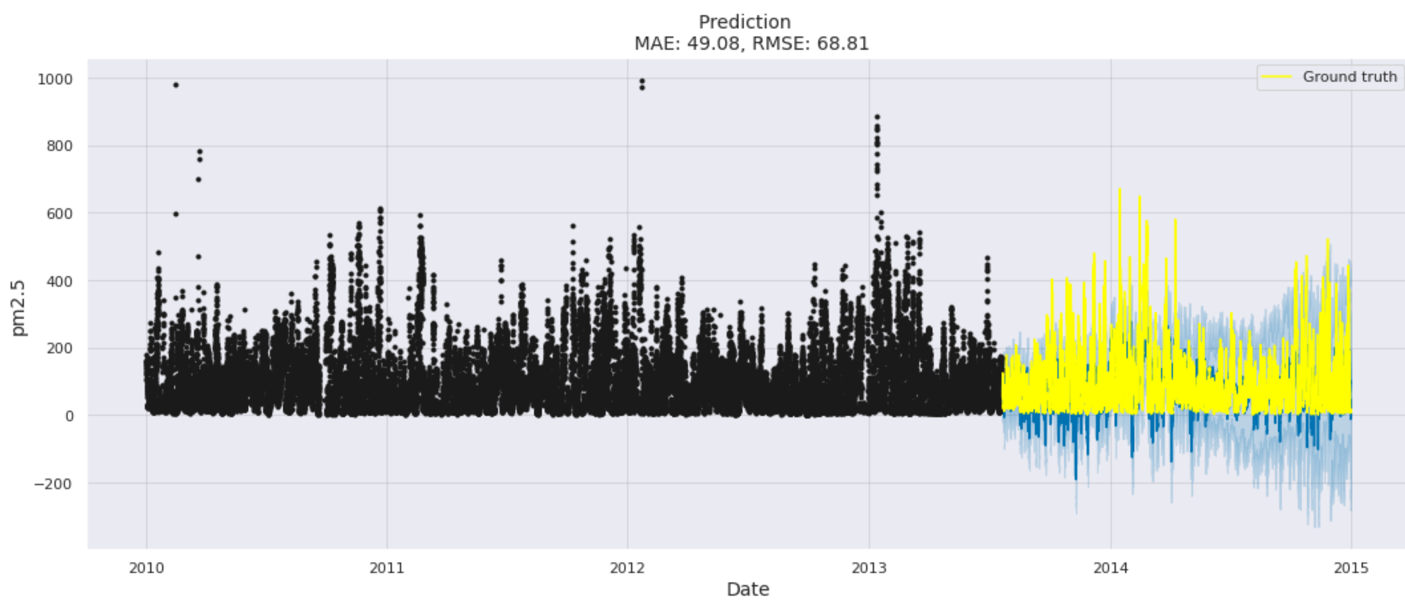
Out[27]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [28]:

```
f, ax = plt.subplots(1)
f.set_figheight(6)
f.set_figwidth(15)

pro_model.plot(pro_y_pred, ax=ax)
sns.lineplot(x=X_test['ds'], y=y_test['y'], ax=ax, color='yellow', label='Ground truth') #navajowhi

ax.set_title(f'Prediction ₩n MAE: {score_mae:.2f}, RMSE: {score_rmse:.2f}', fontsize=14)
ax.set_xlabel(xlabel='Date', fontsize=14)
ax.set_ylabel(ylabel='pm2.5', fontsize=14)

plt.show()
```



5. DECISION TREE

In [29]:

```
# fit final model

X_train['ds'] = pd.to_numeric(pd.to_datetime(X_train['ds']))
X_test['ds'] = pd.to_numeric(pd.to_datetime(X_test['ds']))

tree_model = DecisionTreeClassifier(criterion="entropy", max_depth=100000)
tree_model.fit(X_train, y_train)
tree_y_pred = tree_model.predict(X_test)


cv = KFold(n_splits=10, random_state=1, shuffle=True)
```

```
scores = cross_val_score(estimator = tree_model, X = X_train, y = y_train, scoring='neg_mean_absolu

print('RMSE:',sqrt(mean_squared_error(y_test, tree_y_pred)))
print('RMSE(10-fold cross-validation):',sqrt(mean(absolute(scores))))
```

```
RMSE: 106.35206989707385
RMSE(10-fold cross-validation): 6.138303679002624
```

6. Random Forest Regression

In [30]:

```
rf_model = RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=100,
            max_features='sqrt', min_samples_leaf=4,
            min_samples_split=6, n_estimators=100)
rf_model.fit(X_train, y_train['y'])
rf_y_pred = rf_model.predict(X_test)

cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(estimator = rf_model, X = X_train, y = y_train['y'], scoring='neg_mean_abs

print('RMSE:',sqrt(mean_squared_error(y_test, rf_y_pred)))
print('RMSE(10-fold cross-validation):',sqrt(mean(absolute(scores))))
```

```
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
/home/studio-lab-user/.conda/envs/default/lib/python3.9/site-packages/sklearn/ensemble/_forest.py:39
```

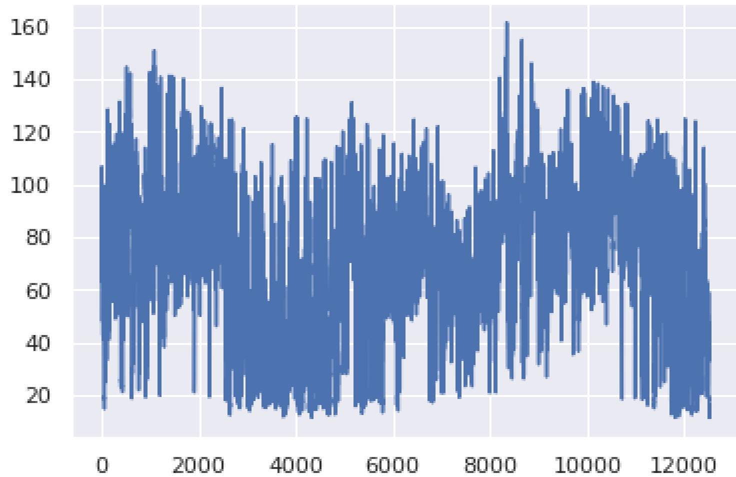
```
6: FutureWarning: Criterion 'mse' was deprecated in v1.0 and will be removed in version 1.2. Use `cr
iterion='squared_error'` which is equivalent.
  warn(
RMSE: 88.8193409104224
RMSE(10-fold cross-validation): 5.702259084476705
```

In [31]:

```python
plt.plot(rf_y_pred)
plt.show
```

Out[31]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [32]:

```python
from sklearn.datasets import make_regression
# define dataset
# define the model
rf_model = RandomForestRegressor()
# fit the model
rf_model.fit(X_train, y_train['y'])
# get importance
importance = rf_model.feature_importances_
# summarize feature importance
for i,v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
plt.bar([x for x in ['datetime','DEWP','PRES','cbwd','lws']], importance)
plt.show()
```

```
Feature: 0, Score: 0.51077
Feature: 1, Score: 0.24630
Feature: 2, Score: 0.10825
Feature: 3, Score: 0.03398
Feature: 4, Score: 0.10070
```