

Temperature Models

June 6, 2021

```
[69]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import AdaBoostRegressor
from sklearn import svm
from sklearn.metrics import confusion_matrix, mean_absolute_error
from sklearn.metrics import r2_score
```

```
[35]: data = pd.read_csv('..\Dataset\data_trial.csv')
```

```
[36]: df = pd.DataFrame(data)
df_temp = df.date.str.split(" ", expand = True)
df['date'] = df_temp
df['date'] = pd.to_datetime(df['date'])
```

```
[37]: df.set_index('date', inplace = True)
```

```
[38]: df.drop(columns=['blizzard'], inplace=True)
```

```
[39]: df
```

```
[39]:
```

	tempr	ap	ws	rh
date				
2016-01-01	1.16	980.83	8.27	67.60
2016-01-01	1.35	980.95	10.91	65.30
2016-01-01	1.53	981.07	8.59	64.70
2016-01-01	1.43	981.37	8.18	66.10
2016-01-01	1.24	981.66	9.19	73.86
...
2021-02-04	-4.58	975.76	3.17	60.35
2021-02-04	-3.88	975.98	5.95	51.05
2021-02-04	-4.50	976.21	7.17	49.89
2021-02-04	-4.98	976.44	7.22	50.56
2021-02-04	-4.93	976.67	7.07	51.44

```
[44688 rows x 4 columns]
```

```
[40]: def series_to_supervised(df, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    #df = pd.DataFrame(data)
    cols, names = list(), list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j+1, i)) for j in range(n_vars)]
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j+1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j+1, i)) for j in
→range(n_vars)]
    # put it all together
    agg = pd.concat(cols, axis=1)
    # drop rows with NaN values
    if dropnan:
        agg.dropna(inplace=True)
    return agg
```

```
[41]: n_in = 5
sup_data = series_to_supervised(df, n_in, 1, dropnan= True)
```

```
[42]: sup_data
```

```
[42]:
```

	tempr	ap	ws	rh	tempr	ap	ws	rh	tempr \
date									
2016-01-01	1.16	980.83	8.27	67.60	1.35	980.95	10.91	65.30	1.53
2016-01-01	1.35	980.95	10.91	65.30	1.53	981.07	8.59	64.70	1.43
2016-01-01	1.53	981.07	8.59	64.70	1.43	981.37	8.18	66.10	1.24
2016-01-01	1.43	981.37	8.18	66.10	1.24	981.66	9.19	73.86	1.57
2016-01-01	1.24	981.66	9.19	73.86	1.57	981.96	10.45	68.28	1.48
...
2021-02-04	-1.61	974.81	3.26	75.71	-2.30	974.86	4.50	79.85	-2.79
2021-02-04	-2.30	974.86	4.50	79.85	-2.79	975.13	1.45	67.20	-3.14
2021-02-04	-2.79	975.13	1.45	67.20	-3.14	975.36	3.11	61.67	-4.33
2021-02-04	-3.14	975.36	3.11	61.67	-4.33	975.59	1.94	65.52	-4.58
2021-02-04	-4.33	975.59	1.94	65.52	-4.58	975.76	3.17	60.35	-3.88

	ap	...	ws	rh	tempr	ap	ws	rh	tempr \
date									
2016-01-01	981.07	...	8.18	66.10	1.24	981.66	9.19	73.86	1.57
2016-01-01	981.37	...	9.19	73.86	1.57	981.96	10.45	68.28	1.48
2016-01-01	981.66	...	10.45	68.28	1.48	982.45	8.72	69.09	1.51

2016-01-01	981.96	...	8.72	69.09	1.51	982.82	9.14	69.54	1.55
2016-01-01	982.45	...	9.14	69.54	1.55	983.05	8.44	68.19	1.45
...
2021-02-04	975.13	...	3.11	61.67	-4.33	975.59	1.94	65.52	-4.58
2021-02-04	975.36	...	1.94	65.52	-4.58	975.76	3.17	60.35	-3.88
2021-02-04	975.59	...	3.17	60.35	-3.88	975.98	5.95	51.05	-4.50
2021-02-04	975.76	...	5.95	51.05	-4.50	976.21	7.17	49.89	-4.98
2021-02-04	975.98	...	7.17	49.89	-4.98	976.44	7.22	50.56	-4.93

	ap	ws	rh
date			
2016-01-01	981.96	10.45	68.28
2016-01-01	982.45	8.72	69.09
2016-01-01	982.82	9.14	69.54
2016-01-01	983.05	8.44	68.19
2016-01-01	983.14	7.42	67.67
...
2021-02-04	975.76	3.17	60.35
2021-02-04	975.98	5.95	51.05
2021-02-04	976.21	7.17	49.89
2021-02-04	976.44	7.22	50.56
2021-02-04	976.67	7.07	51.44

[44683 rows x 24 columns]

```
[43]: cols = ['tempr_t', 'ap_t', 'ws_t', 'rh_t']
for j in range(1, n_in+1):
    for i in range(1, 5):
        cols.insert(0, cols[-i] + '-' + str(j))
print(cols)
sup_data.columns = cols
```

```
['tempr_t-5', 'ap_t-5', 'ws_t-5', 'rh_t-5', 'tempr_t-4', 'ap_t-4', 'ws_t-4',
'rh_t-4', 'tempr_t-3', 'ap_t-3', 'ws_t-3', 'rh_t-3', 'tempr_t-2', 'ap_t-2',
'ws_t-2', 'rh_t-2', 'tempr_t-1', 'ap_t-1', 'ws_t-1', 'rh_t-1', 'tempr_t',
'ap_t', 'ws_t', 'rh_t']
```

```
[44]: sup_data.loc['2019-05-01':'2019-10-01']
```

```
[44]:      tempr_t-5  ap_t-5  ws_t-5  rh_t-5  tempr_t-4  ap_t-4  ws_t-4  \
date
2019-05-01    -10.60  968.53   12.52   43.83    -10.32  968.48   17.77
2019-05-01    -10.32  968.48   17.77   63.75    -10.70  968.66   19.94
2019-05-01    -10.70  968.66   19.94   64.68    -10.95  969.13   16.75
2019-05-01    -10.95  969.13   16.75   66.13    -10.73  969.34   18.12
2019-05-01    -10.73  969.34   18.12   60.55    -11.22  969.56   16.13
...          ...    ...    ...    ...    ...    ...    ...
```

2019-10-01	-15.99	987.22	7.41	39.90	-16.17	986.87	12.46
2019-10-01	-16.17	986.87	12.46	40.60	-16.27	986.77	13.78
2019-10-01	-16.27	986.77	13.78	40.79	-16.47	986.85	13.74
2019-10-01	-16.47	986.85	13.74	41.65	-16.58	986.78	15.36
2019-10-01	-16.58	986.78	15.36	41.26	-17.28	986.42	19.19

	rh_t-4	tempr_t-3	ap_t-3	...	ws_t-2	rh_t-2	tempr_t-1	ap_t-1	\
date				...					
2019-05-01	63.75	-10.70	968.66	...	16.75	66.13	-10.73	969.34	
2019-05-01	64.68	-10.95	969.13	...	18.12	60.55	-11.22	969.56	
2019-05-01	66.13	-10.73	969.34	...	16.13	62.14	-11.52	969.51	
2019-05-01	60.55	-11.22	969.56	...	17.04	63.62	-11.42	969.64	
2019-05-01	62.14	-11.52	969.51	...	17.07	58.95	-11.43	970.04	
...	
2019-10-01	40.60	-16.27	986.77	...	13.74	41.65	-16.58	986.78	
2019-10-01	40.79	-16.47	986.85	...	15.36	41.26	-17.28	986.42	
2019-10-01	41.65	-16.58	986.78	...	19.19	47.25	-17.72	986.09	
2019-10-01	41.26	-17.28	986.42	...	20.28	52.68	-17.71	985.59	
2019-10-01	47.25	-17.72	986.09	...	19.38	53.24	-17.72	984.92	

	ws_t-1	rh_t-1	tempr_t	ap_t	ws_t	rh_t
date						
2019-05-01	18.12	60.55	-11.22	969.56	16.13	62.14
2019-05-01	16.13	62.14	-11.52	969.51	17.04	63.62
2019-05-01	17.04	63.62	-11.42	969.64	17.07	58.95
2019-05-01	17.07	58.95	-11.43	970.04	16.51	55.37
2019-05-01	16.51	55.37	-11.16	970.67	16.27	50.33
...
2019-10-01	15.36	41.26	-17.28	986.42	19.19	47.25
2019-10-01	19.19	47.25	-17.72	986.09	20.28	52.68
2019-10-01	20.28	52.68	-17.71	985.59	19.38	53.24
2019-10-01	19.38	53.24	-17.72	984.92	20.61	54.13
2019-10-01	20.61	54.13	-17.96	984.40	20.16	56.39

[3696 rows x 24 columns]

```
[45]: sup_data = sup_data.resample('D').mean()
```

```
[46]: sup_data
```

```
[46]:
```

	tempr_t-5	ap_t-5	ws_t-5	rh_t-5	tempr_t-4	\
date						
2016-01-01	1.353158	981.775263	8.076316	65.385789	1.291053	
2016-01-02	1.504167	975.300000	9.081250	52.919167	1.509167	
2016-01-03	1.472083	968.008750	10.731667	59.789167	1.461250	
2016-01-04	-0.210417	979.956667	14.297083	83.582500	-0.261250	
2016-01-05	1.492083	975.076667	7.490833	52.660417	1.595417	

...
2021-01-31	-1.075000	971.448750	9.830417	49.120833	-1.013750
2021-02-01	-1.145000	975.370833	8.505833	54.052500	-1.229167
2021-02-02	-1.036250	978.087500	7.933750	54.898333	-0.969583
2021-02-03	-1.789167	966.490417	16.091250	43.295417	-1.975833
2021-02-04	-2.407500	971.908750	9.431667	60.444167	-2.385000

	ap_t-4	ws_t-4	rh_t-4	tempr_t-3	ap_t-3	...	\
date						...	
2016-01-01	981.732105	8.307895	64.510526	1.179474	981.663158	...	
2016-01-02	974.883750	8.749167	53.077500	1.544167	974.461250	...	
2016-01-03	968.161667	11.156667	61.366667	1.407500	968.370833	...	
2016-01-04	980.296667	13.915417	82.547917	-0.290417	980.590417	...	
2016-01-05	974.450833	7.712083	52.314167	1.716667	973.822917	...	
...	
2021-01-31	971.350833	9.552917	49.371250	-0.948333	971.290417	...	
2021-02-01	975.621667	8.413750	53.932083	-1.315833	975.878750	...	
2021-02-02	977.856667	8.397083	54.750000	-0.915417	977.572083	...	
2021-02-03	966.201250	16.145833	42.958333	-2.151667	965.975417	...	
2021-02-04	972.329583	8.887083	61.190833	-2.325000	972.732500	...	

	ws_t-2	rh_t-2	tempr_t-1	ap_t-1	ws_t-1	rh_t-1	\
date							
2016-01-01	9.130000	63.246316	0.897368	981.475263	9.452632	62.479474	
2016-01-02	7.590417	52.956667	1.717917	973.563750	7.120417	52.743750	
2016-01-03	12.598333	65.002500	1.185417	968.970833	13.306667	66.924583	
2016-01-04	12.573750	80.030833	-0.279583	981.029583	12.053333	78.617500	
2016-01-05	8.480000	52.035833	1.930000	972.566667	8.731250	52.108333	
...	
2021-01-31	8.688750	49.940417	-0.738750	971.287083	8.267083	50.008333	
2021-02-01	8.402917	53.530833	-1.491250	976.393333	8.317500	53.552917	
2021-02-02	9.087500	54.547917	-0.792500	976.875000	9.351667	54.230000	
2021-02-03	16.274583	42.893333	-2.470833	965.706250	16.276667	43.289583	
2021-02-04	8.106250	61.439167	-2.294583	973.474167	7.803750	61.277500	

	tempr_t	ap_t	ws_t	rh_t
date				
2016-01-01	0.746316	981.335789	9.763684	61.320526
2016-01-02	1.877083	973.100417	6.620833	52.495000
2016-01-03	1.025000	969.352500	14.052083	68.708333
2016-01-04	-0.247917	981.176667	11.511250	77.202917
2016-01-05	2.013333	971.953333	9.087500	52.332917
...
2021-01-31	-0.651250	971.340000	8.035417	50.252917
2021-02-01	-1.575833	976.644167	8.139583	53.721667
2021-02-02	-0.737083	976.452917	9.656667	53.725833
2021-02-03	-2.584583	965.677083	16.209583	43.877500

2021-02-04 -2.308750 973.810417 7.512083 60.979167

[1862 rows x 24 columns]

```
[47]: df1 = sup_data.loc['2019-05-01':'2019-10-01']
df1
```

```
[47]:
```

	tempr_t-5	ap_t-5	ws_t-5	rh_t-5	tempr_t-4	\
date						
2019-05-01	-10.352083	972.363750	14.874167	55.267917	-10.292083	
2019-05-02	-12.940417	989.585833	11.286250	50.522083	-12.994583	
2019-05-03	-12.291250	990.073333	16.870000	64.536250	-12.484167	
2019-05-04	-16.948750	987.153333	12.591667	79.383750	-17.050000	
2019-05-05	-18.032917	988.842917	12.604583	68.871250	-18.028750	
...	
2019-09-27	-12.364167	980.049583	18.650417	52.376250	-12.505417	
2019-09-28	-13.627083	983.465833	10.802500	46.164167	-13.612083	
2019-09-29	-14.977917	985.869583	4.370833	50.815833	-15.071667	
2019-09-30	-16.531667	986.101667	7.485417	41.111250	-16.622500	
2019-10-01	-16.595000	987.189167	9.325833	39.579583	-16.545833	

	ap_t-4	ws_t-4	rh_t-4	tempr_t-3	ap_t-3	...	\
date						...	
2019-05-01	972.791667	14.572500	55.677083	-10.248333	973.264167	...	
2019-05-02	990.194167	11.696250	50.981250	-13.047083	990.740833	...	
2019-05-03	989.844167	16.832917	65.173750	-12.665833	989.628333	...	
2019-05-04	987.121667	12.732500	78.367083	-17.175417	987.106667	...	
2019-05-05	988.872500	12.017917	69.342500	-17.997083	988.882083	...	
...	
2019-09-27	980.258750	18.891250	52.399583	-12.650000	980.447083	...	
2019-09-28	983.617917	10.253750	45.787500	-13.606250	983.780417	...	
2019-09-29	985.848333	4.296250	50.876250	-15.156667	985.825417	...	
2019-09-30	986.166667	7.764167	40.871250	-16.721667	986.229583	...	
2019-10-01	987.183750	9.599583	39.923333	-16.505833	987.160833	...	

	ws_t-2	rh_t-2	tempr_t-1	ap_t-1	ws_t-1	rh_t-1	\
date							
2019-05-01	13.310000	54.630833	-10.277917	974.373333	12.810417	53.650417	
2019-05-02	12.695417	52.098750	-13.035000	991.590000	13.142500	53.207917	
2019-05-03	16.482917	66.440000	-12.952083	989.205833	16.287500	66.543333	
2019-05-04	13.052917	76.542917	-17.477083	987.139167	13.238750	76.124167	
2019-05-05	11.331250	70.664167	-17.913750	988.837500	10.910833	71.854583	
...	
2019-09-27	19.272500	52.420417	-12.890833	980.786667	19.357917	52.291250	
2019-09-28	8.980417	45.526667	-13.610833	984.104583	8.284583	45.467500	
2019-09-29	4.780833	50.535833	-15.337917	985.770833	5.311250	50.342083	
2019-09-30	7.900417	40.507083	-16.864583	986.367917	7.802083	40.295000	

2019-10-01	10.216250	41.057500	-16.440417	987.018750	10.529167	41.710417
------------	-----------	-----------	------------	------------	-----------	-----------

	tempr_t	ap_t	ws_t	rh_t
date				
2019-05-01	-11.103333	973.687500	12.503750	52.651250
2019-05-02	-12.236667	993.220000	13.423333	54.440417
2019-05-03	-13.077917	989.006667	15.997500	66.973750
2019-05-04	-17.633333	987.174167	13.387083	75.445833
2019-05-05	-17.883333	988.776250	10.543333	72.696250
...
2019-09-27	-12.973750	980.947500	19.315417	52.125417
2019-09-28	-13.616250	984.262500	7.774167	45.377083
2019-09-29	-15.438750	985.740000	5.775417	50.155833
2019-09-30	-16.923333	986.428750	7.711250	40.115417
2019-10-01	-16.415000	986.910000	10.779583	42.470417

[154 rows x 24 columns]

```
[48]: from sklearn.ensemble import RandomForestRegressor
```

```
[49]: model = RandomForestRegressor(n_estimators=1000)
```

```
[50]: df1
```

```
[50]:
```

	tempr_t-5	ap_t-5	ws_t-5	rh_t-5	tempr_t-4	\
date						
2019-05-01	-10.352083	972.363750	14.874167	55.267917	-10.292083	
2019-05-02	-12.940417	989.585833	11.286250	50.522083	-12.994583	
2019-05-03	-12.291250	990.073333	16.870000	64.536250	-12.484167	
2019-05-04	-16.948750	987.153333	12.591667	79.383750	-17.050000	
2019-05-05	-18.032917	988.842917	12.604583	68.871250	-18.028750	
...	
2019-09-27	-12.364167	980.049583	18.650417	52.376250	-12.505417	
2019-09-28	-13.627083	983.465833	10.802500	46.164167	-13.612083	
2019-09-29	-14.977917	985.869583	4.370833	50.815833	-15.071667	
2019-09-30	-16.531667	986.101667	7.485417	41.111250	-16.622500	
2019-10-01	-16.595000	987.189167	9.325833	39.579583	-16.545833	

	ap_t-4	ws_t-4	rh_t-4	tempr_t-3	ap_t-3	...	\
date							
2019-05-01	972.791667	14.572500	55.677083	-10.248333	973.264167	...	
2019-05-02	990.194167	11.696250	50.981250	-13.047083	990.740833	...	
2019-05-03	989.844167	16.832917	65.173750	-12.665833	989.628333	...	
2019-05-04	987.121667	12.732500	78.367083	-17.175417	987.106667	...	
2019-05-05	988.872500	12.017917	69.342500	-17.997083	988.882083	...	
...	
2019-09-27	980.258750	18.891250	52.399583	-12.650000	980.447083	...	

2019-09-28	983.617917	10.253750	45.787500	-13.606250	983.780417	...
2019-09-29	985.848333	4.296250	50.876250	-15.156667	985.825417	...
2019-09-30	986.166667	7.764167	40.871250	-16.721667	986.229583	...
2019-10-01	987.183750	9.599583	39.923333	-16.505833	987.160833	...

	ws_t-2	rh_t-2	tempr_t-1	ap_t-1	ws_t-1	rh_t-1 \
date						
2019-05-01	13.310000	54.630833	-10.277917	974.373333	12.810417	53.650417
2019-05-02	12.695417	52.098750	-13.035000	991.590000	13.142500	53.207917
2019-05-03	16.482917	66.440000	-12.952083	989.205833	16.287500	66.543333
2019-05-04	13.052917	76.542917	-17.477083	987.139167	13.238750	76.124167
2019-05-05	11.331250	70.664167	-17.913750	988.837500	10.910833	71.854583
...
2019-09-27	19.272500	52.420417	-12.890833	980.786667	19.357917	52.291250
2019-09-28	8.980417	45.526667	-13.610833	984.104583	8.284583	45.467500
2019-09-29	4.780833	50.535833	-15.337917	985.770833	5.311250	50.342083
2019-09-30	7.900417	40.507083	-16.864583	986.367917	7.802083	40.295000
2019-10-01	10.216250	41.057500	-16.440417	987.018750	10.529167	41.710417

	tempr_t	ap_t	ws_t	rh_t
date				
2019-05-01	-11.103333	973.687500	12.503750	52.651250
2019-05-02	-12.236667	993.220000	13.423333	54.440417
2019-05-03	-13.077917	989.006667	15.997500	66.973750
2019-05-04	-17.633333	987.174167	13.387083	75.445833
2019-05-05	-17.883333	988.776250	10.543333	72.696250
...
2019-09-27	-12.973750	980.947500	19.315417	52.125417
2019-09-28	-13.616250	984.262500	7.774167	45.377083
2019-09-29	-15.438750	985.740000	5.775417	50.155833
2019-09-30	-16.923333	986.428750	7.711250	40.115417
2019-10-01	-16.415000	986.910000	10.779583	42.470417

[154 rows x 24 columns]

```
[51]: labels = np.array(df1['tempr_t'])
train_set = df1.iloc[:, 0:(n_in*4)]
train_set
```

```
[51]:      tempr_t-5      ap_t-5      ws_t-5      rh_t-5      tempr_t-4 \
date
2019-05-01 -10.352083  972.363750  14.874167  55.267917 -10.292083
2019-05-02 -12.940417  989.585833  11.286250  50.522083 -12.994583
2019-05-03 -12.291250  990.073333  16.870000  64.536250 -12.484167
2019-05-04 -16.948750  987.153333  12.591667  79.383750 -17.050000
2019-05-05 -18.032917  988.842917  12.604583  68.871250 -18.028750
...          ...          ...          ...          ...          ...
```


2019-09-27	-12.364167	980.049583	18.650417	52.376250	-12.505417
2019-09-28	-13.627083	983.465833	10.802500	46.164167	-13.612083
2019-09-29	-14.977917	985.869583	4.370833	50.815833	-15.071667
2019-09-30	-16.531667	986.101667	7.485417	41.111250	-16.622500
2019-10-01	-16.595000	987.189167	9.325833	39.579583	-16.545833

	ap_t-4	ws_t-4	rh_t-4	tempr_t-3	ap_t-3 \
date					
2019-05-01	972.791667	14.572500	55.677083	-10.248333	973.264167
2019-05-02	990.194167	11.696250	50.981250	-13.047083	990.740833
2019-05-03	989.844167	16.832917	65.173750	-12.665833	989.628333
2019-05-04	987.121667	12.732500	78.367083	-17.175417	987.106667
2019-05-05	988.872500	12.017917	69.342500	-17.997083	988.882083
...
2019-09-27	980.258750	18.891250	52.399583	-12.650000	980.447083
2019-09-28	983.617917	10.253750	45.787500	-13.606250	983.780417
2019-09-29	985.848333	4.296250	50.876250	-15.156667	985.825417
2019-09-30	986.166667	7.764167	40.871250	-16.721667	986.229583
2019-10-01	987.183750	9.599583	39.923333	-16.505833	987.160833

	ws_t-3	rh_t-3	tempr_t-2	ap_t-2	ws_t-2	rh_t-2 \
date						
2019-05-01	13.987917	55.299167	-10.232083	973.786667	13.310000	54.630833
2019-05-02	12.180417	51.301667	-13.072500	991.210417	12.695417	52.098750
2019-05-03	16.712917	65.991667	-12.824167	989.418750	16.482917	66.440000
2019-05-04	12.887083	77.274167	-17.313750	987.115417	13.052917	76.542917
2019-05-05	11.592083	69.777917	-17.951250	988.869167	11.331250	70.664167
...
2019-09-27	19.115000	52.461667	-12.780000	980.620417	19.272500	52.420417
2019-09-28	9.629583	45.551250	-13.614167	983.945833	8.980417	45.526667
2019-09-29	4.405417	50.752917	-15.246250	985.797083	4.780833	50.535833
2019-09-30	7.929583	40.769167	-16.792500	986.298750	7.900417	40.507083
2019-10-01	9.917917	40.428750	-16.475833	987.107500	10.216250	41.057500

	tempr_t-1	ap_t-1	ws_t-1	rh_t-1
date				
2019-05-01	-10.277917	974.373333	12.810417	53.650417
2019-05-02	-13.035000	991.590000	13.142500	53.207917
2019-05-03	-12.952083	989.205833	16.287500	66.543333
2019-05-04	-17.477083	987.139167	13.238750	76.124167
2019-05-05	-17.913750	988.837500	10.910833	71.854583
...
2019-09-27	-12.890833	980.786667	19.357917	52.291250
2019-09-28	-13.610833	984.104583	8.284583	45.467500
2019-09-29	-15.337917	985.770833	5.311250	50.342083
2019-09-30	-16.864583	986.367917	7.802083	40.295000
2019-10-01	-16.440417	987.018750	10.529167	41.710417

[154 rows x 20 columns]

```
[52]: from sklearn.model_selection import train_test_split
      tx_train, tx_test, ty_train, ty_test = train_test_split(train_set, labels,
      ↪ test_size = 0.2, shuffle = False)
```

```
[53]: model.fit(tx_train, ty_train)
```

```
[53]: RandomForestRegressor(n_estimators=1000)
```

```
[54]: ty_pred = model.predict(tx_test)
```

```
[55]: tx_train
```

```
[55]:      tempr_t-5      ap_t-5      ws_t-5      rh_t-5      tempr_t-4 \
date
2019-05-01 -10.352083    972.363750    14.874167    55.267917 -10.292083
2019-05-02 -12.940417    989.585833    11.286250    50.522083 -12.994583
2019-05-03 -12.291250    990.073333    16.870000    64.536250 -12.484167
2019-05-04 -16.948750    987.153333    12.591667    79.383750 -17.050000
2019-05-05 -18.032917    988.842917    12.604583    68.871250 -18.028750
...
2019-08-27 -18.304583    977.288333     8.815417    43.658750 -18.301667
2019-08-28 -18.033750    986.090833    11.196667    46.077500 -17.807917
2019-08-29 -11.428750    992.631667     6.111250    81.760000 -11.344583
2019-08-30 -14.317083   1001.310417     6.370833    62.330417 -14.440000
2019-08-31 -14.557917    994.171250     9.222917    49.395417 -14.618333

      ap_t-4      ws_t-4      rh_t-4      tempr_t-3      ap_t-3 \
date
2019-05-01    972.791667    14.572500    55.677083 -10.248333    973.264167
2019-05-02    990.194167    11.696250    50.981250 -13.047083    990.740833
2019-05-03    989.844167    16.832917    65.173750 -12.665833    989.628333
2019-05-04    987.121667    12.732500    78.367083 -17.175417    987.106667
2019-05-05    988.872500    12.017917    69.342500 -17.997083    988.882083
...
2019-08-27    977.552083     8.848333    44.466250 -18.303750    977.858750
2019-08-28    986.297083    10.847083    46.435833 -17.575000    986.482500
2019-08-29    993.128333     6.187917    82.770417 -11.273333    993.636250
2019-08-30   1001.240417     6.411250    60.928333 -14.560417   1001.140000
2019-08-31    993.797083     9.750833    49.285000 -14.656250    993.427917

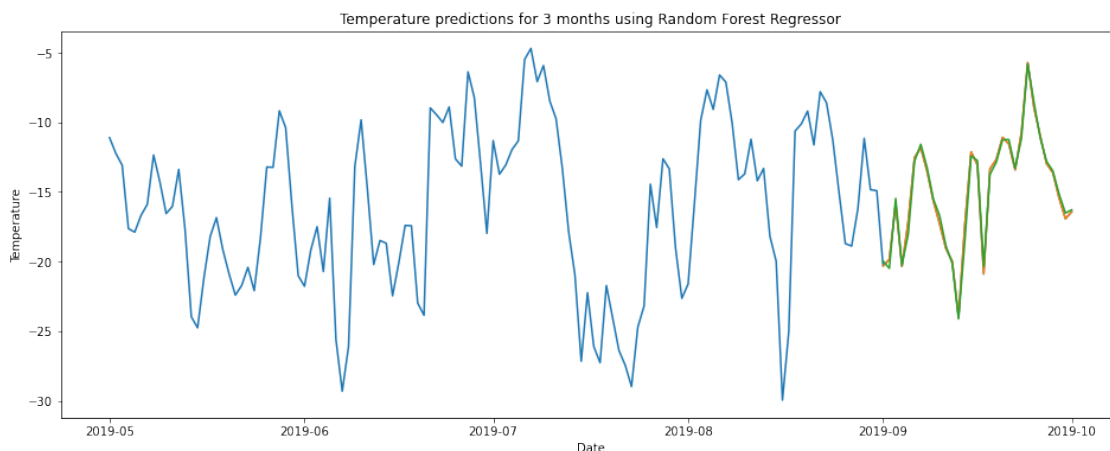
      ws_t-3      rh_t-3      tempr_t-2      ap_t-2      ws_t-2 \
date
2019-05-01    13.987917    55.299167 -10.232083    973.786667    13.310000
2019-05-02    12.180417    51.301667 -13.072500    991.210417    12.695417
```

2019-05-03	16.712917	65.991667	-12.824167	989.418750	16.482917
2019-05-04	12.887083	77.274167	-17.313750	987.115417	13.052917
2019-05-05	11.592083	69.777917	-17.951250	988.869167	11.331250
...
2019-08-27	8.951667	45.320417	-18.284583	978.215417	9.075417
2019-08-28	10.420000	47.210833	-17.341667	986.647083	10.000417
2019-08-29	6.249167	83.284167	-11.215417	994.158333	6.246667
2019-08-30	6.480833	59.601250	-14.659167	1001.004167	6.632917
2019-08-31	10.192917	48.927500	-14.730000	993.060833	10.692500

	rh_t-2	tempr_t-1	ap_t-1	ws_t-1	rh_t-1
date					
2019-05-01	54.630833	-10.277917	974.373333	12.810417	53.650417
2019-05-02	52.098750	-13.035000	991.590000	13.142500	53.207917
2019-05-03	66.440000	-12.952083	989.205833	16.287500	66.543333
2019-05-04	76.542917	-17.477083	987.139167	13.238750	76.124167
2019-05-05	70.664167	-17.913750	988.837500	10.910833	71.854583
...
2019-08-27	45.945417	-18.229583	978.604583	9.387083	46.341250
2019-08-28	48.203333	-17.128750	986.802500	9.620833	49.494583
2019-08-29	83.597083	-11.159583	994.689167	6.242500	83.799583
2019-08-30	58.300000	-14.760417	1000.846250	6.717500	56.944583
2019-08-31	48.728750	-14.811250	992.707500	11.104167	48.447500

[123 rows x 20 columns]

```
[56]: plt.figure(figsize=(16,6))
plt.plot(df1.index, labels, label = 'original data')
plt.plot(tx_test.index, ty_test, label = 'expected')
plt.plot(tx_test.index, ty_pred, label = 'predicted')
plt.title("Temperature predictions for 3 months using Random Forest Regressor")
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.show()
```



```
[57]: mae = mean_absolute_error(ty_test, ty_pred)
      print(mae)
```

0.3251237768817122

```
[58]: print(r2_score(ty_test, ty_pred_2))
```

0.9907653912592276

1 Extra Trees Regressor

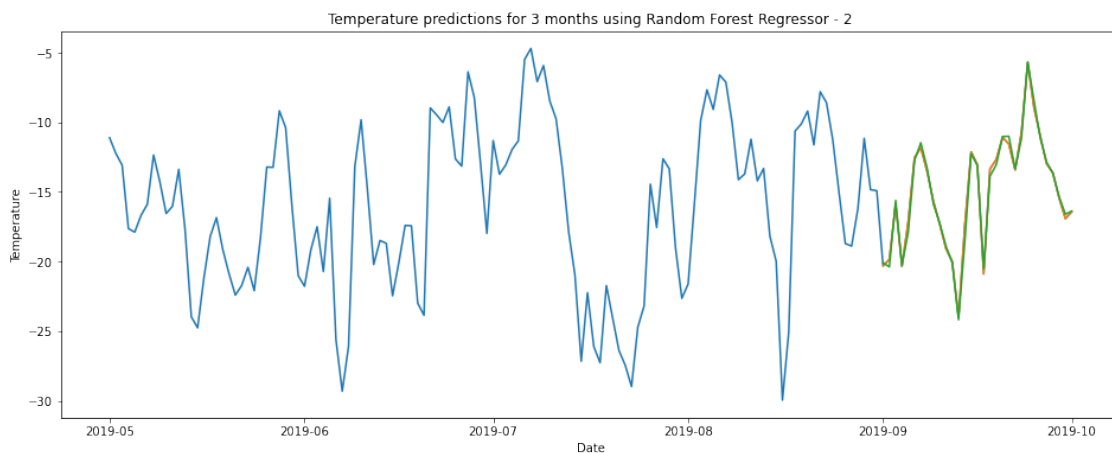
```
[59]: from sklearn.ensemble import ExtraTreesRegressor
      model_2 = ExtraTreesRegressor()
```

```
[60]: model_2.fit(tx_train, ty_train)
```

```
[60]: ExtraTreesRegressor()
```

```
[61]: ty_pred_2 = model_2.predict(tx_test)
```

```
[62]: plt.figure(figsize=(16,6))
      plt.plot(df1.index, labels, label = 'original data')
      plt.plot(tx_test.index, ty_test, label = 'expected')
      plt.plot(tx_test.index, ty_pred_2, label = 'predicted')
      plt.title("Temperature predictions for 3 months using Random Forest Regressor - 2")
      plt.xlabel("Date")
      plt.ylabel("Temperature")
      plt.show()
```



```
[63]: mae = mean_absolute_error(ty_test, ty_pred_2)
print(mae)
```

0.27575846774193735

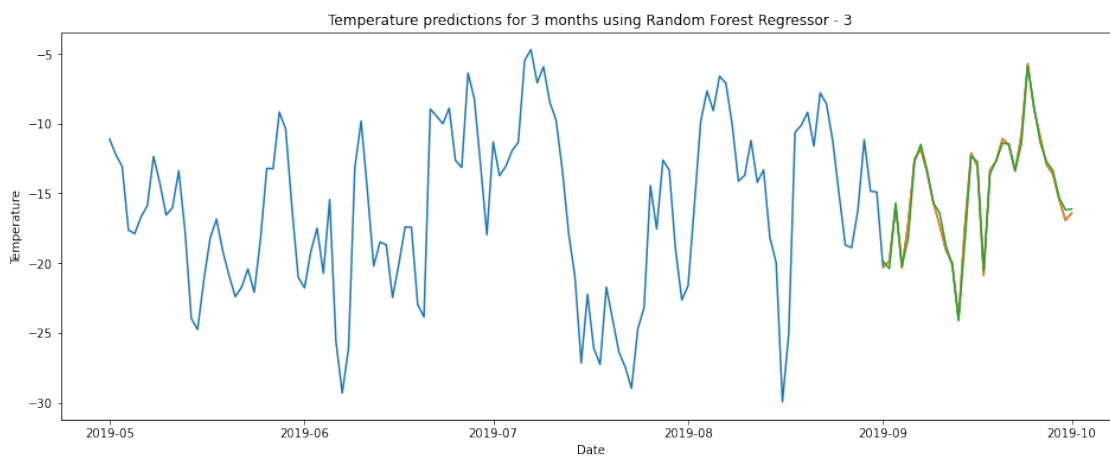
```
[64]: print(r2_score(ty_test, ty_pred_2))
```

0.991377082164826

2 Adaboost Regressor

```
[65]: model_3 = AdaBoostRegressor()
model_3.fit(tx_train,ty_train)
ty_pred_3 = model_3.predict(tx_test)
```

```
[67]: plt.figure(figsize=(16,6))
plt.plot(df1.index, labels, label = 'original data')
plt.plot(tx_test.index, ty_test, label = 'expected')
plt.plot(tx_test.index, ty_pred_3, label = 'predicted')
plt.title("Temperature predictions for 3 months using Random Forest Regressor - 3")
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.show()
```



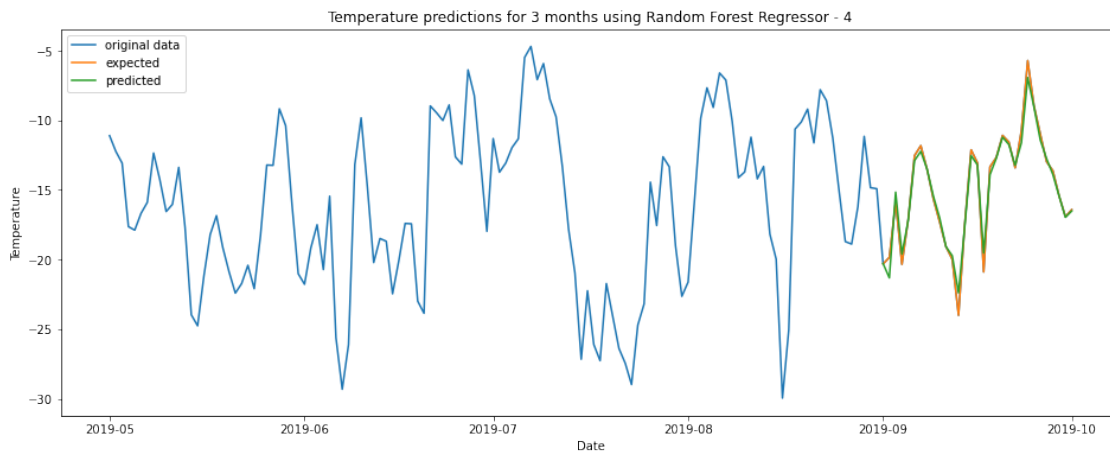
```
[317]: print(r2_score(ty_test, ty_pred_4))
```

0.9868241762456882

3 Support Vector Machines models

```
[123]: model_4 = svm.SVR(kernel = 'poly', C = 200, degree = 10)
model_4.fit(tx_train, ty_train)
ty_pred_4 = model_4.predict(tx_test)
```

```
[124]: plt.figure(figsize=(16,6))
plt.plot(df1.index, labels, label = 'original data')
plt.plot(tx_test.index, ty_test, label = 'expected')
plt.plot(tx_test.index, ty_pred_4, label = 'predicted')
plt.title("Temperature predictions for 3 months using Random Forest Regressor - 4")
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.legend()
plt.show()
```



```
[ ]:
```