智能电网：企业电力大数据预测

石恩名

广州优亿信息科技有限公司

CONTENTS

优亿科技 USEEASE

研究背景

智能电网：企业电力大数据预测

背景¶

精细化电量预测是中长期电量预测发展的一个重要趋势。针对目前我国的中长期电量预测的预测对象主要是总电量，预测结果准确度难以进一步提高，且提供的信息十分有限的问题，本文开展了基于企业用电行业的中长期电量预测研究。

思路

（1）精细化电量预测是中长期电量预测的重要指标，特别是对核心用电大户的电量预测可以对长期的电力发展规划起到十分重要的作用。本研究主要侧重于“有色金属”和“橡胶”两大用点巨头的用电预测。

（2）用电预测本质上是一种时间序列的预测，类似的时间序列例子包括：股票债券等金融产品的价格趋势预测、网络流量预测、人口和经济增长预测等。

（3）企业用电预测和股票预测类似，主要会受到大环境和企业自身发展的影响，在时间上具有周期性、波动性。

探索性分析

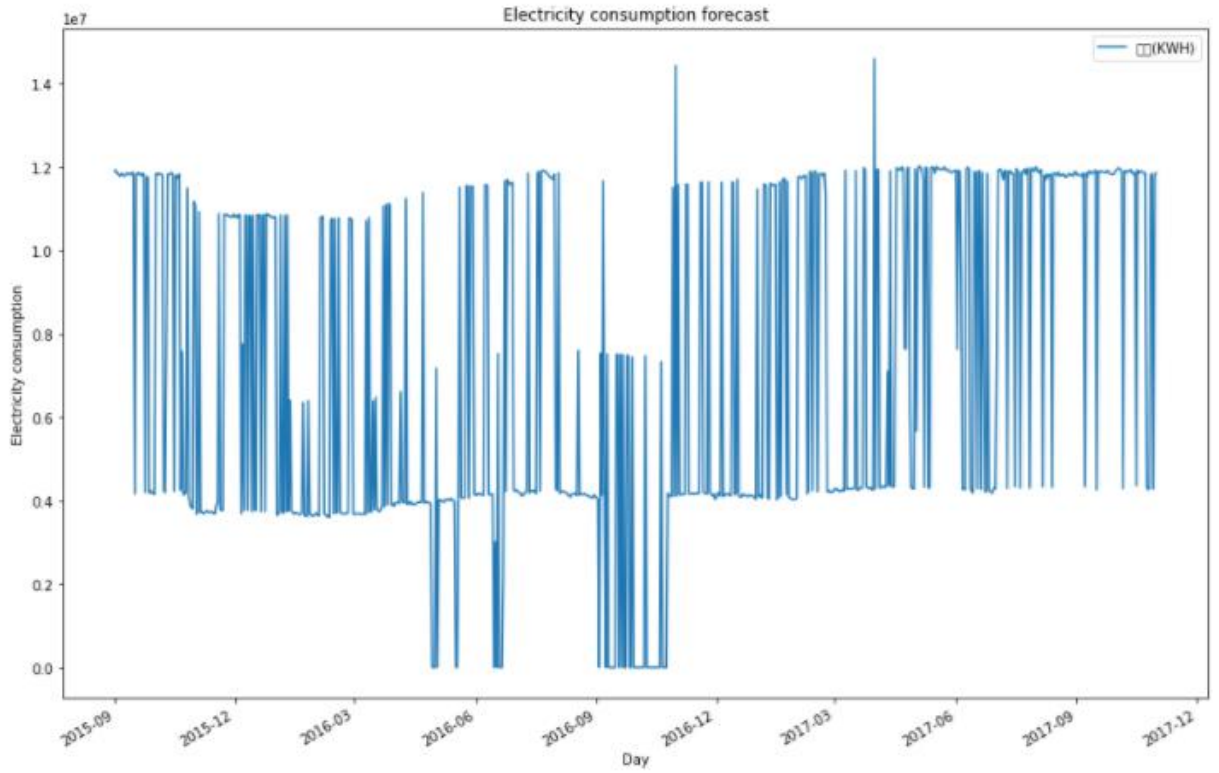
云南铝业：数据加载与探索

云南铝业（000807）

```
# 加载数据
consumption = pd.read_excel('data/Daliy_consumption.xlsx')
consumption.head()
```

	客户名称	时间	电量(KWH)
0	昆明市官渡区荣宏塑料厂	2015-09-01	1596.98
1	昆明市官渡区荣宏塑料厂	2015-09-02	1658.23
2	昆明市官渡区荣宏塑料厂	2015-09-03	1634.56
3	昆明市官渡区荣宏塑料厂	2015-09-04	1586.91
4	昆明市官渡区荣宏塑料厂	2015-09-05	1543.75

```
# 画图
fig = plt.figure(1,figsize=[15,10])
#画图
aluminum['电量(KWH)'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



优亿科技 USEEASE

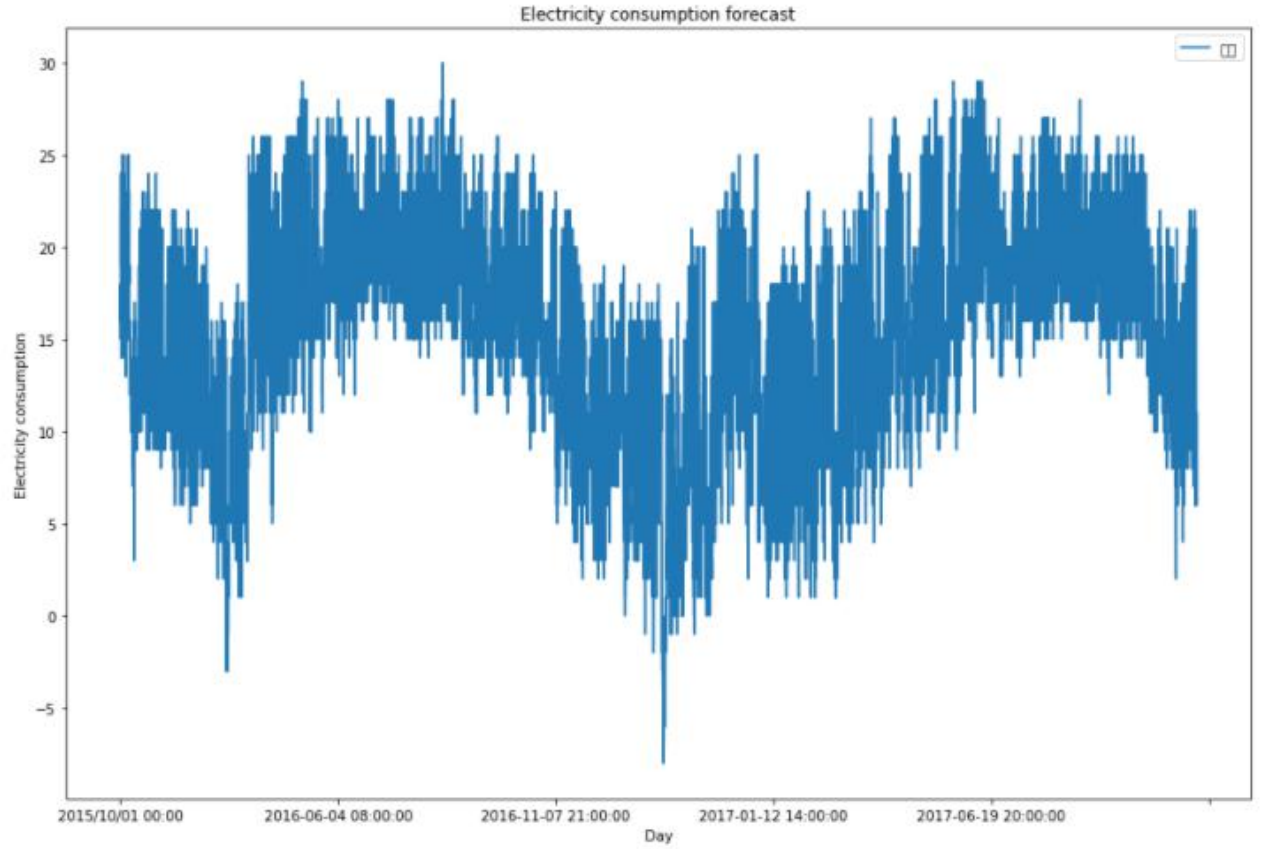
云南铝业：数据加载与探索

天气数据

```
weather.head()
```

	时间	温度	湿度
0	2015/10/01 00:00	16.0	88%
1	2015/10/01 01:00	16.0	94%
2	2015/10/01 02:00	16.0	94%
3	2015/10/01 02:00	17.0	95%
4	2015/10/01 03:00	16.0	94%

```
# 画图
fig = plt.figure(1,figsize=[15,10])
#画图
weather['温度'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



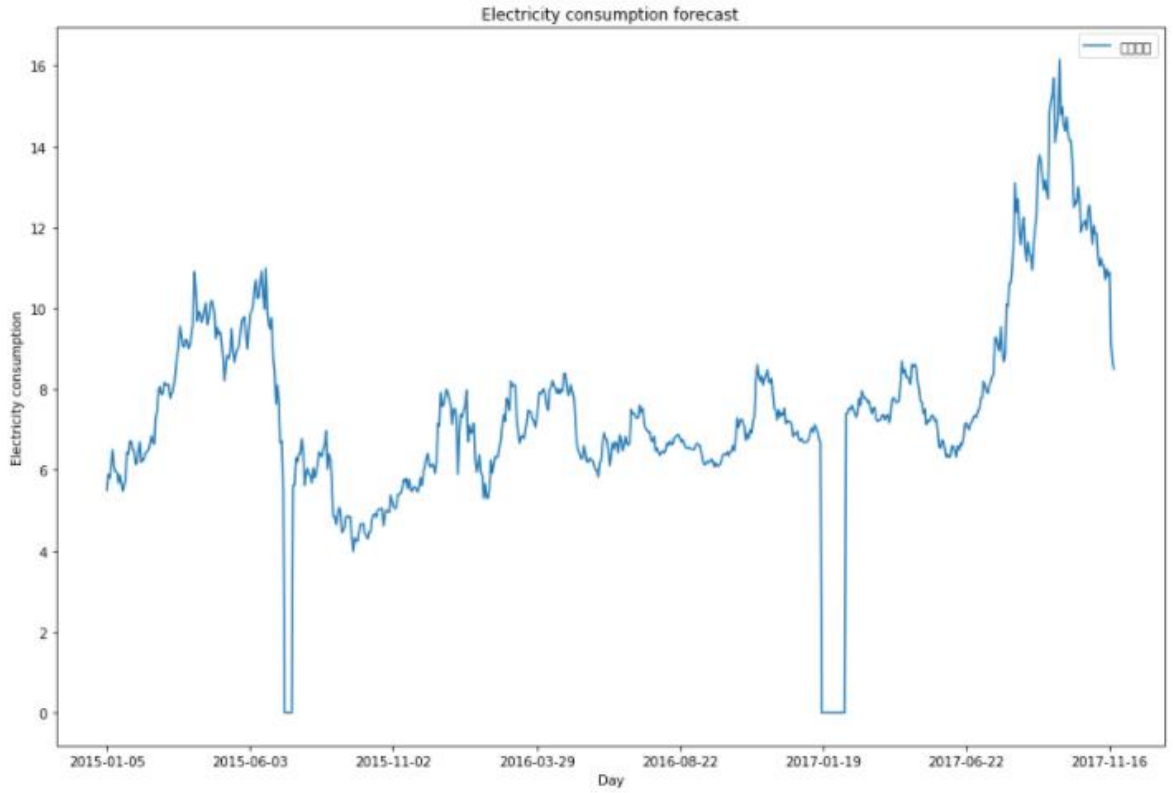
云南铝业：数据加载与探索

股票数据

```
stock.head()
```

交易日期	证券代码	证券简称	交易日期	交易所	昨日收盘	今日开盘	成交数量	最高成交	最低成交	最近成交	总笔数	涨跌幅	成交金额
2015-01-05	807	云铝股份	2015-01-05	深交所	5.52	5.50	50792270	5.90	5.48	5.90	18257	6.8841	2.927935e+08
2015-01-06	807	云铝股份	2015-01-06	深交所	5.90	5.89	57370713	6.16	5.80	5.86	18195	-0.6780	3.437802e+08
2015-01-07	807	云铝股份	2015-01-07	深交所	5.86	5.80	66484885	6.36	5.76	6.26	23552	6.8259	4.058442e+08
2015-01-08	807	云铝股份	2015-01-08	深交所	6.26	6.20	57324552	6.46	6.01	6.26	19410	0.0000	3.570538e+08
2015-01-09	807	云铝股份	2015-01-09	深交所	6.26	6.50	73726132	6.89	6.21	6.21	28906	-0.7987	4.799892e+08

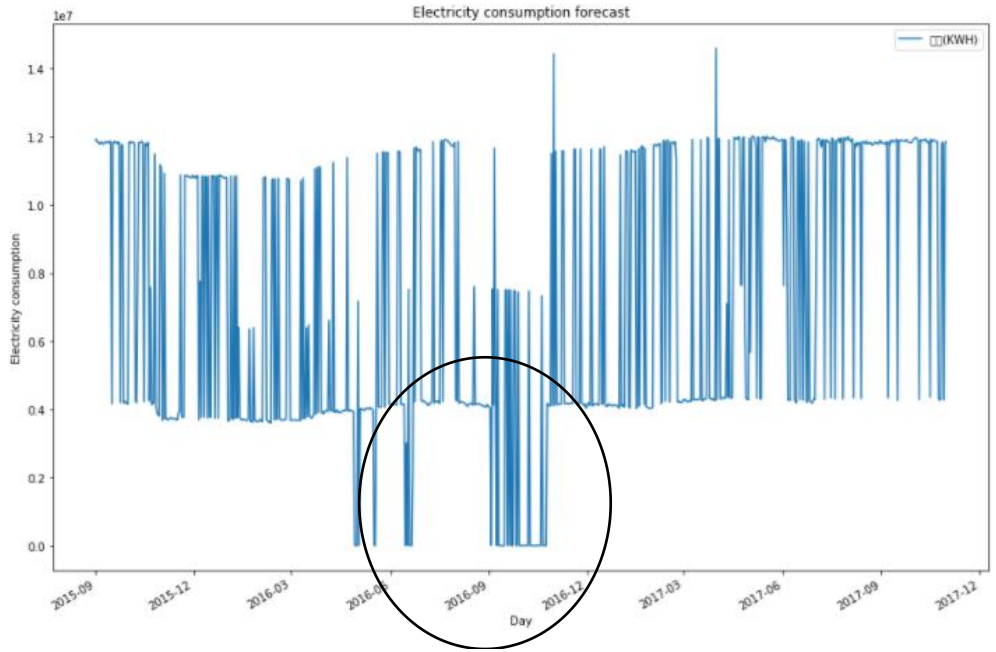
```
# 画图
fig = plt.figure(1,figsize=[15,10])
#画图
stock['今日开盘'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



优亿科技 USEEASE

云南铝业：特征探索

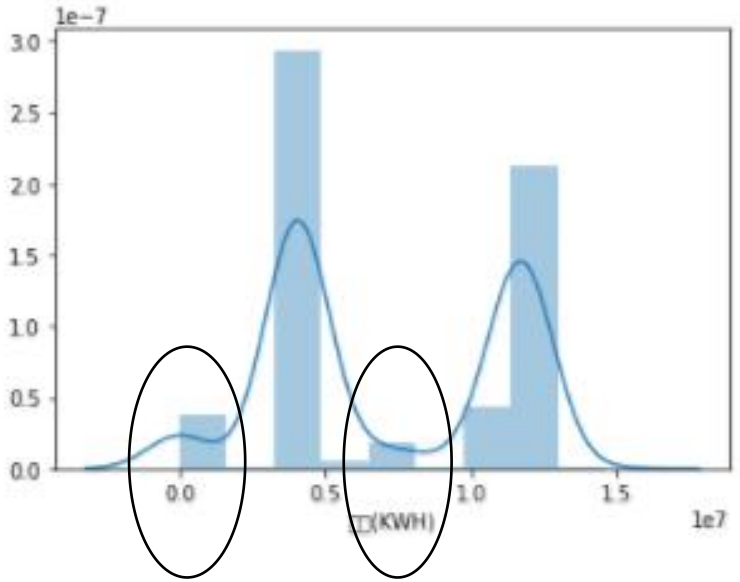
```
# 画图
fig = plt.figure(1,figsize=[15,10])
#画图
aluminum['电量(KWH)'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



用电量时序折线图

```
sns.distplot(aluminum['电量(KWH)'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bc2d1d6d8>
```



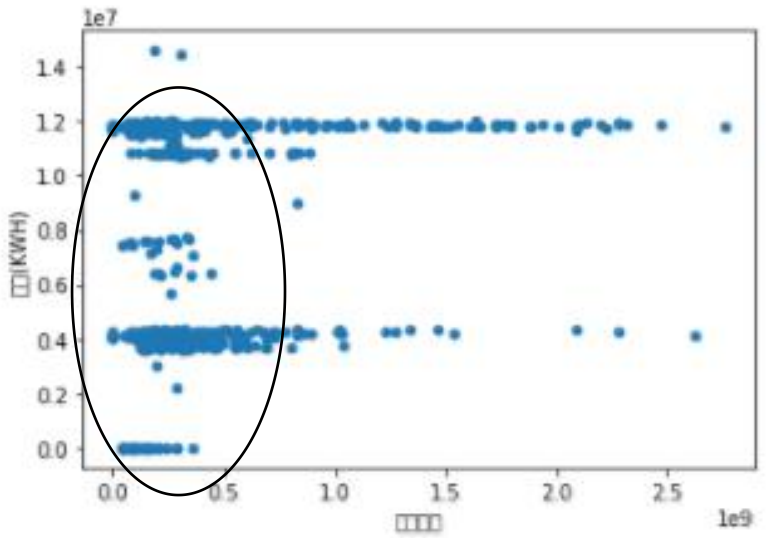
用电量直方图

优亿科技 USEEASE

云南铝业：特征探索

```
aluminum.plot.scatter(x='成交金额', y='电量(KWH)')
```

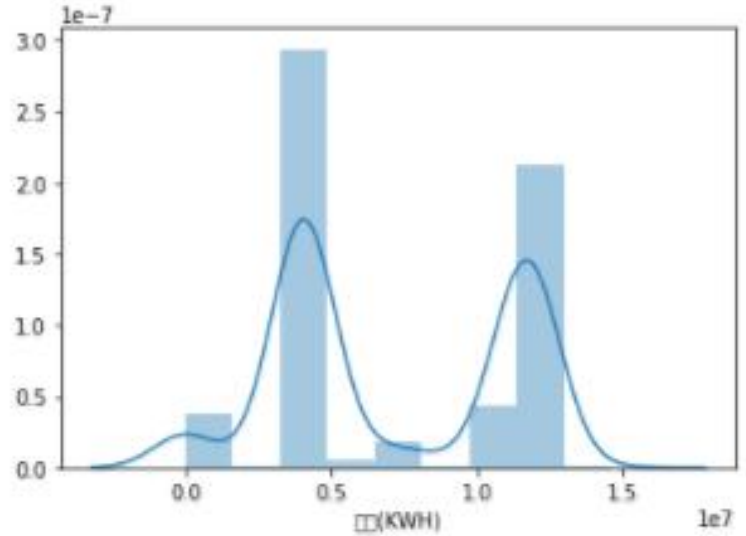
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9c2014ea20>
```



股票成交金额与用电量散点图

```
sns.distplot(aluminum['电量(KWH)'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bc2d1d6d8>
```



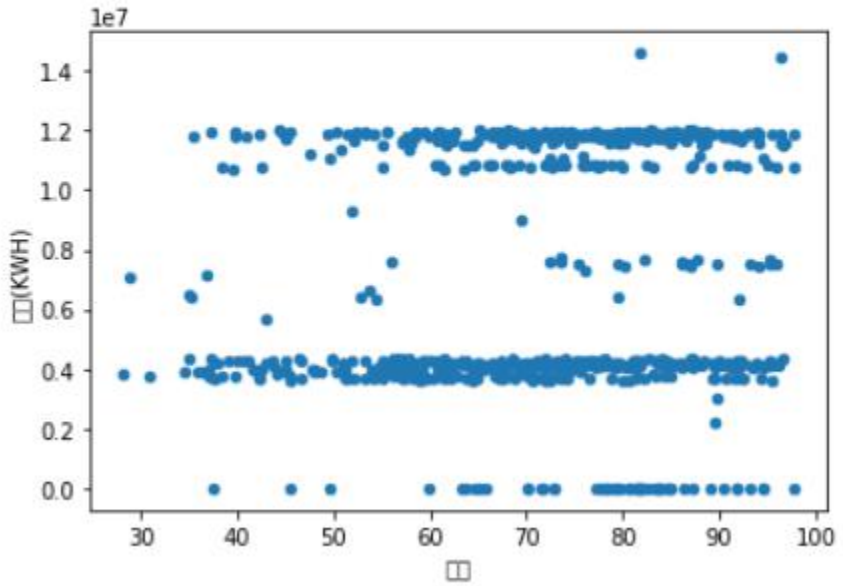
用电量直方图

优亿科技 USEEASE

云南铝业：特征探索

```
aluminum.plot.scatter(x='湿度', y='电量(KWH)')
```

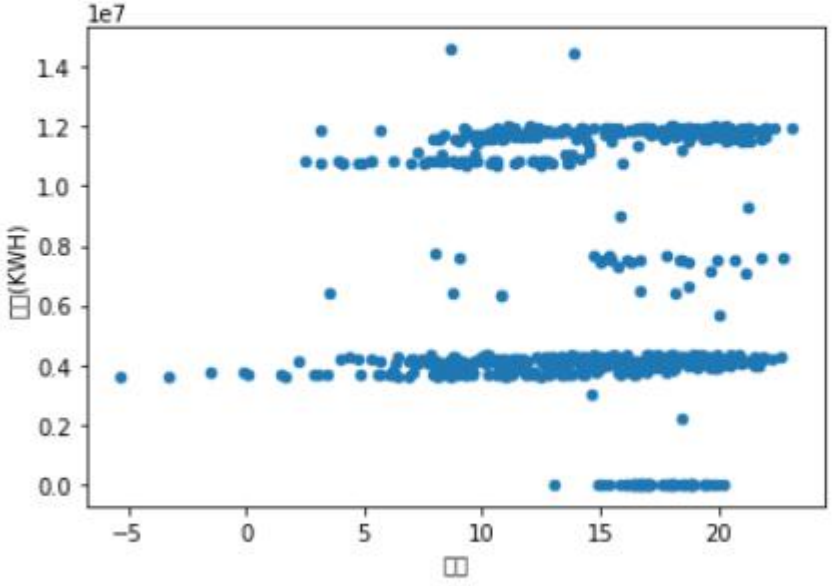
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ca01a9e8>
```



湿度与用电量散点图

```
aluminum.plot.scatter(x='温度', y='电量(KWH)')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14ca337e10>
```



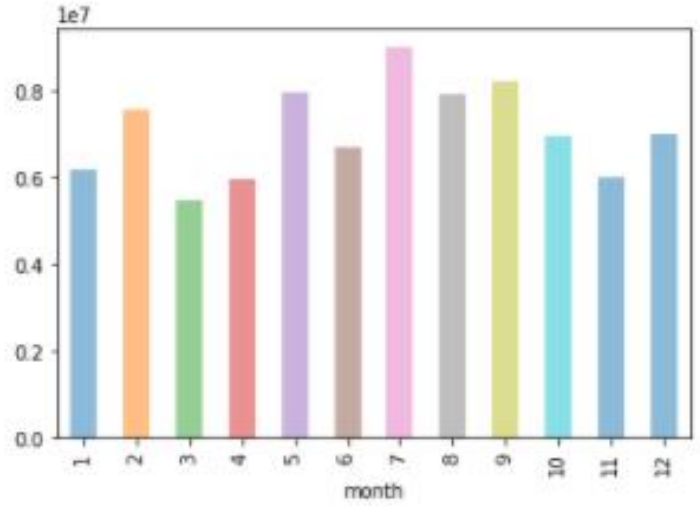
温度与用电量散点图

优亿科技 USEEASE

云南铝业：特征探索

```
aluminum_month = aluminum.groupby('month').mean()
aluminum_month['电量(KWH)'].plot(kind='bar', alpha=0.5)
```

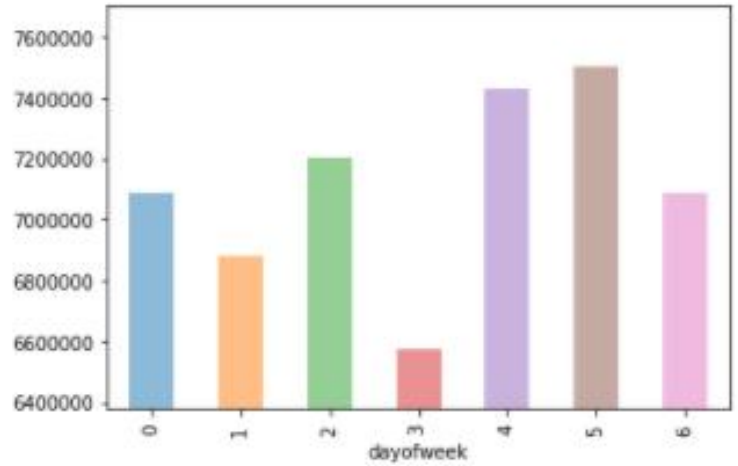
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bc2a0ec88>
```



用电量各月份均值图

```
aluminum_dayofweek = aluminum.groupby('dayofweek').mean()
min_aluminum_dayofweek = min(aluminum_dayofweek['电量(KWH)'])
max_aluminum_dayofweek = max(aluminum_dayofweek['电量(KWH)'])
aluminum_dayofweek['电量(KWH)'].plot(kind='bar', alpha=0.5, ylim=
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bc29013c8>
```



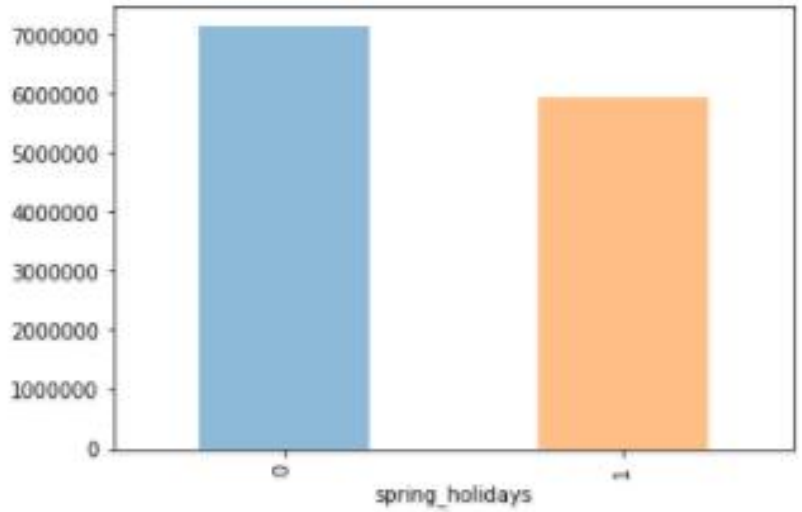
用电量各星期均值图

优亿科技 USEEASE

云南铝业：特征探索

```
aluminum_holiday = aluminum.groupby('spring_holidays').mean()
aluminum_holiday['电量(KWH)'].plot(kind='bar', alpha=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9bc2b10710>
```



春节期间用电量均值图

荣宏塑料：数据加载与探索

荣宏塑料

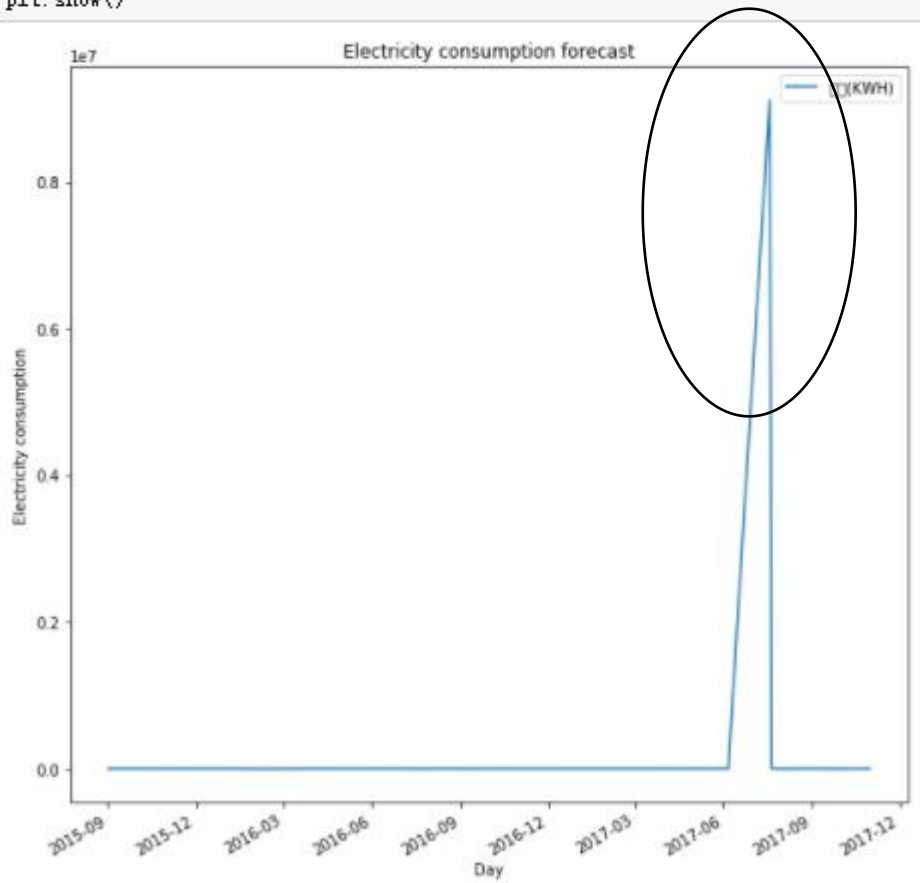
```
]: # 加载数据
consumption = pd.read_excel('data/Daliy_consumption.xlsx')
consumption.head()
```

]:

	客户名称	时间	电量(KWH)
0	昆明市官渡区荣宏塑料厂	2015-09-01	1596.98
1	昆明市官渡区荣宏塑料厂	2015-09-02	1658.23
2	昆明市官渡区荣宏塑料厂	2015-09-03	1634.56
3	昆明市官渡区荣宏塑料厂	2015-09-04	1586.91
4	昆明市官渡区荣宏塑料厂	2015-09-05	1543.75

```
# 画图
fig = plt.figure(1,figsize=[10,10])
#画图
cons_plastics['电量(KWH)'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```

异常值？？



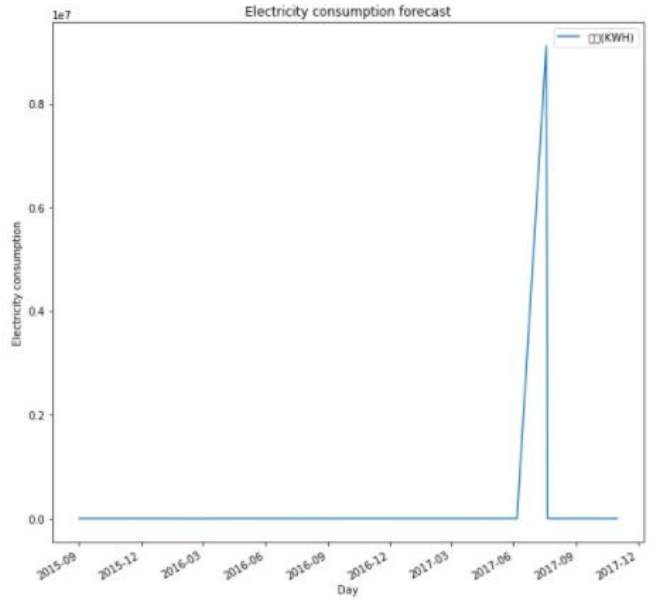
优亿科技 USEEASE

荣宏塑料：删除异常值

```
cons_plastics[cons_plastics['电量(KWH)'] == max(cons_plastics['电量(KWH)'])]
```

时间	客户名称	时间	电量(KWH)	month	dayofweek	day	holiday	holiday_without_weekly	spring_holidays
2017-07-19	昆明市官渡区荣宏塑料厂	2017-07-19	9127297.15	7	2	19	0	0	0

```
# 画图
fig = plt.figure(1,figsize=[10,10])
#画图
cons_plastics['电量(KWH)'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



发现7月19日数据比平时数据搞了几十倍，为了更好的预测将其作为异常值进行剔除。

```
consumption[consumption['电量(KWH)']==min(consumption['电量(KWH)'])]
```

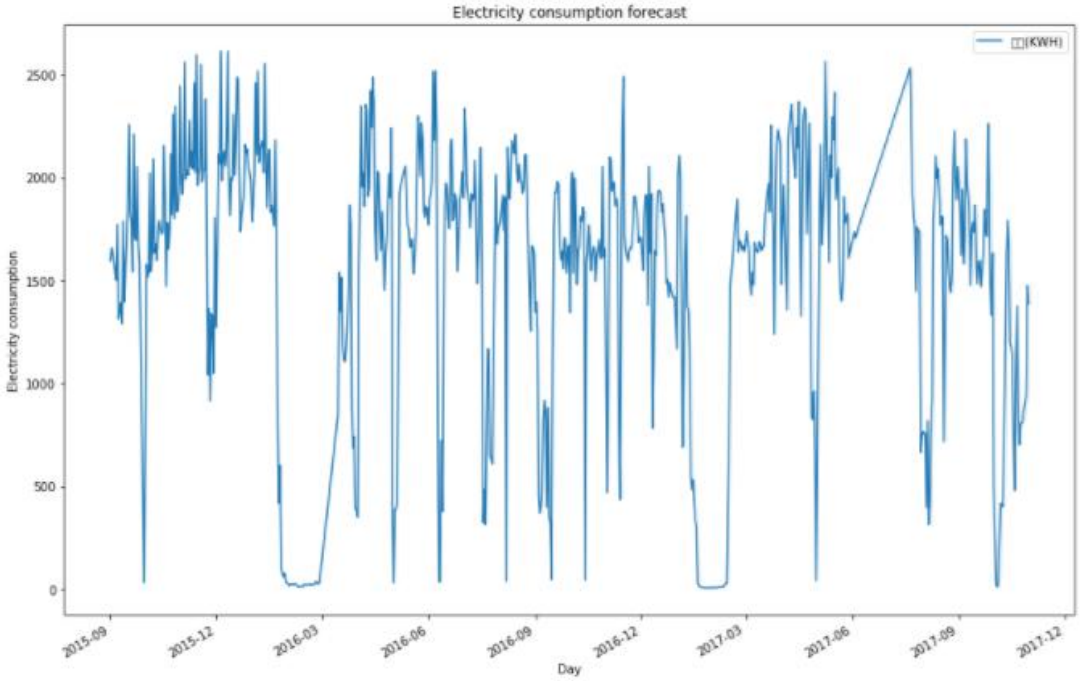
	客户名称	时间	电量(KWH)
1081	云南铝业股份有限公司	2016-09-15	-8.0

发现出现负用电量，应该为异常值，全部设置为0。

优亿科技 USEEASE

荣宏塑料：数据加载与探索

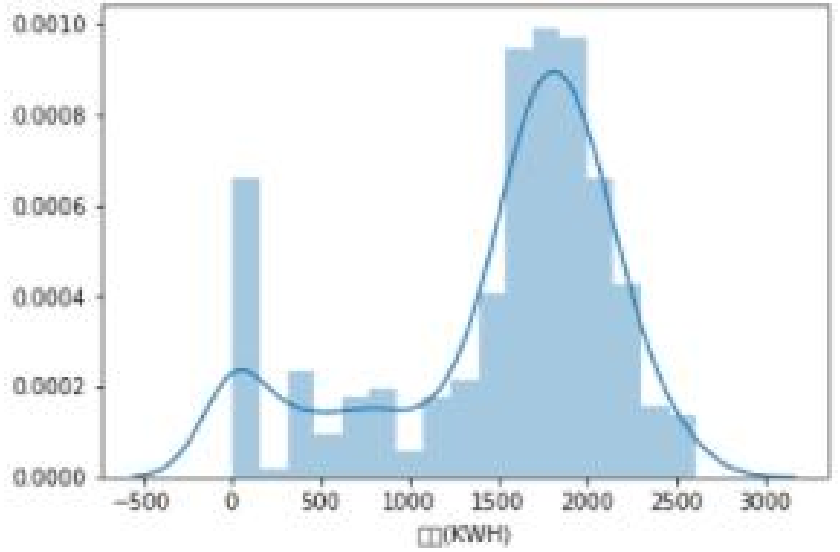
```
# 画图
fig = plt.figure(1,figsize=[15,10])
#画图
cons_plastics['电量(KWH)'].plot()
#图标签
plt.ylabel('Electricity consumption')
plt.xlabel('Day')
plt.title('Electricity consumption forecast')
plt.legend()
plt.show()
```



剔除异常值后的时序图

```
sns.distplot(cons_plastics['电量(KWH)'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f423e71d0>
```



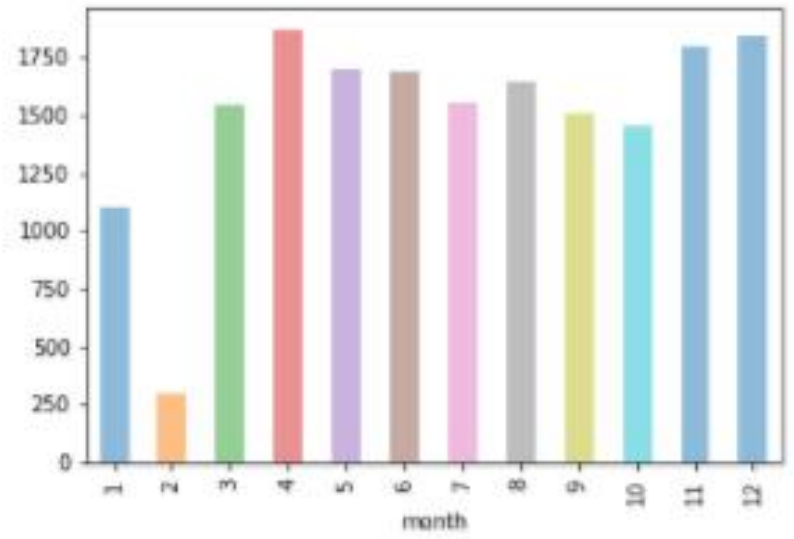
用电量直方图

优亿科技 USEEASE

荣宏塑料：数据加载与探索

```
cons_plastics_month = cons_plastics.groupby('month').mean()
cons_plastics_month['电量(KWH)'].plot(kind='bar', alpha=0.5)
```

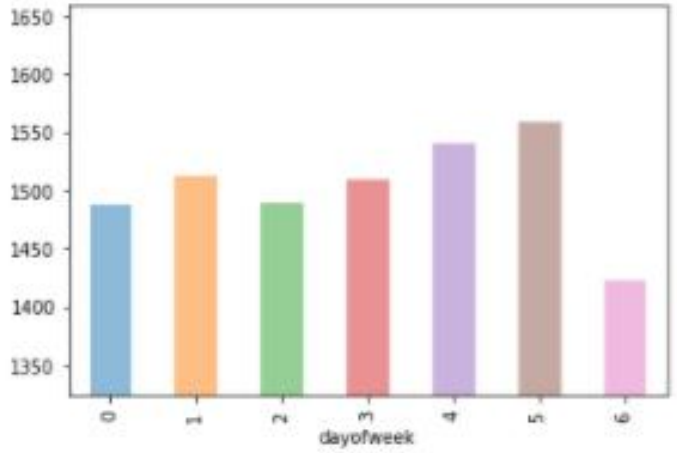
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f669a6e45c0>
```



各月份平均用电量

```
cons_plastics_dayofweek = cons_plastics.groupby('dayofweek').mean()
min_cons_plastics_dayofweek = min(cons_plastics_dayofweek['电量(KWH)'])
max_cons_plastics_dayofweek = max(cons_plastics_dayofweek['电量(KWH)'])
cons_plastics_dayofweek['电量(KWH)'].plot(kind='bar', alpha=0.5, ylim=(mi
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f669a5bcbe0>
```



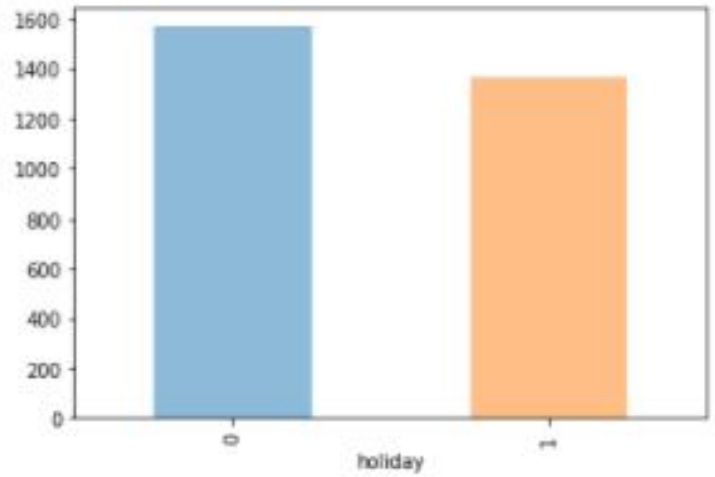
各星期平均用电量

优亿科技 USEEASE

荣宏塑料：数据加载与探索

```
cons_plastics_holiday = cons_plastics.groupby('holiday').mean()
cons_plastics_holiday['电量(KWH)'].plot(kind='bar', alpha=0.5)
```

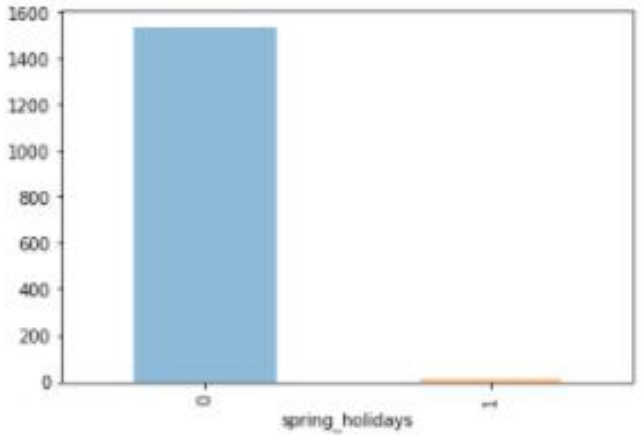
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f669a4b2ba8>
```



节假日平均用电量

```
cons_plastics_holiday = cons_plastics.groupby('spring_holidays').mean()
cons_plastics_holiday['电量(KWH)'].plot(kind='bar', alpha=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f669a439668>
```



春节平均用电量

优亿科技 USEEASE

特征处理：以云南铝业为例

构建移动平滑窗口

根据对云南铝业的探索性分析我们发现，其时间变化特征具有一定的周期性，因此我们可以构建一个移动平滑算法构建均值线作为其中的特征。

移动平均给固定跨越期限内的每个变量值以不相等的权重。其原理是：历史各期产品需求的数据信息对预测未来期内的需求量的作用是不一样的。除了以n为周期的周期性变化外，远离目标期的变量值的影响力相对较低，故应给予较低的权重。

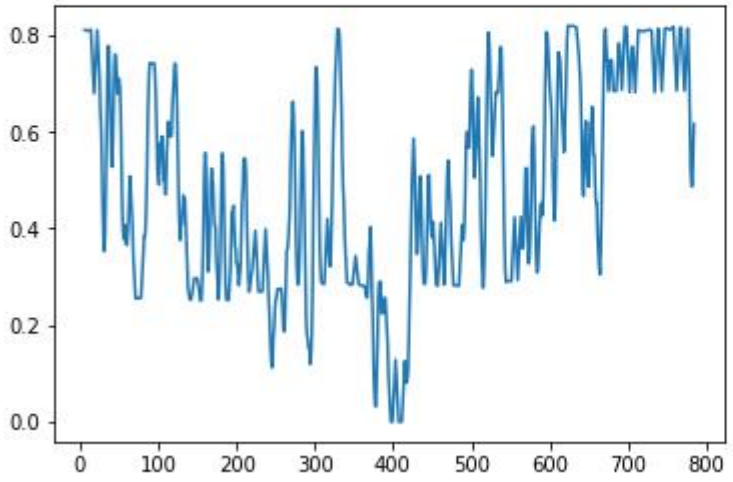
加权移动平均法的计算公式如下：

$$F_t = w_1 A_{t-1} + w_2 A_{t-2} + \ldots + w_n A_{t-n}$$

构建移动平滑窗口

```
aluminum_windows7 = aluminum['电量(KWH)'].rolling(window=7, win_type='triang').mean()
aluminum_windows7 = aluminum_windows7.dropna()
aluminum_windows7.plot()
```

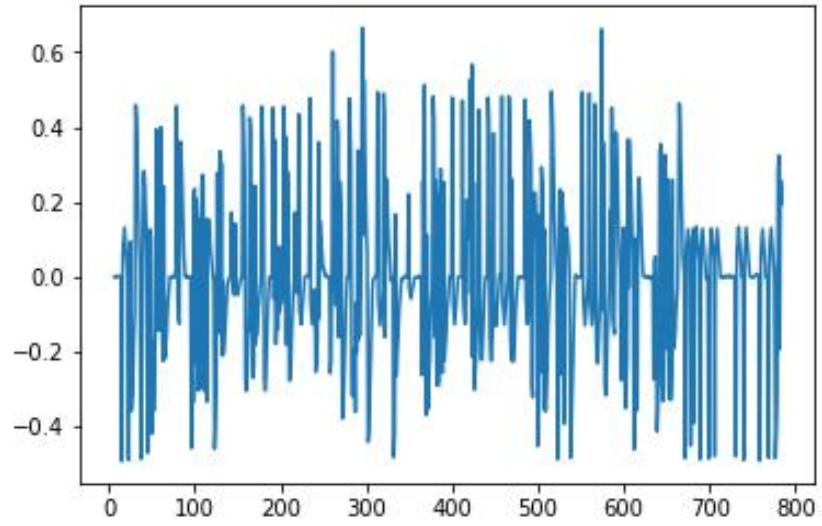
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14c60a7ef0>
```



七天移动平滑趋势图

```
moving_avg_diff = aluminum['电量(KWH)']-aluminum_windows7
moving_avg_diff.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14c4bed7b8>
```

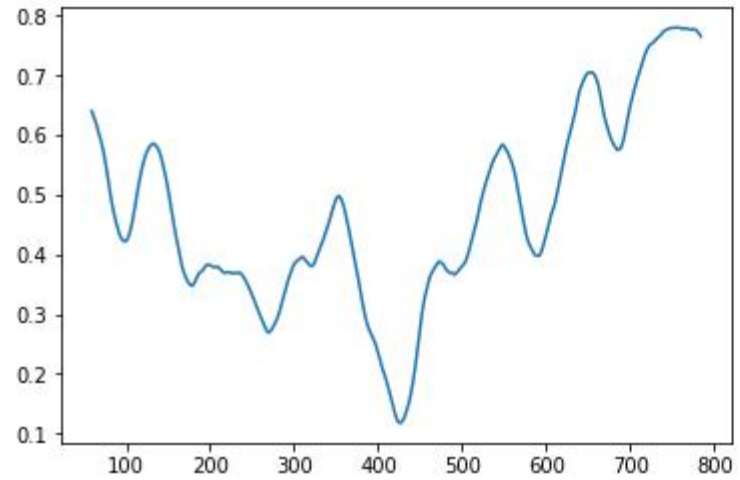


七天差分周期图

构建移动平滑窗口

```
aluminum_windows60= aluminum['电量(KWH)'].rolling(window=60, win_type='triang').mean()
#aluminum_windows14 = aluminum_windows14.dropna()
aluminum_windows60.plot()
```

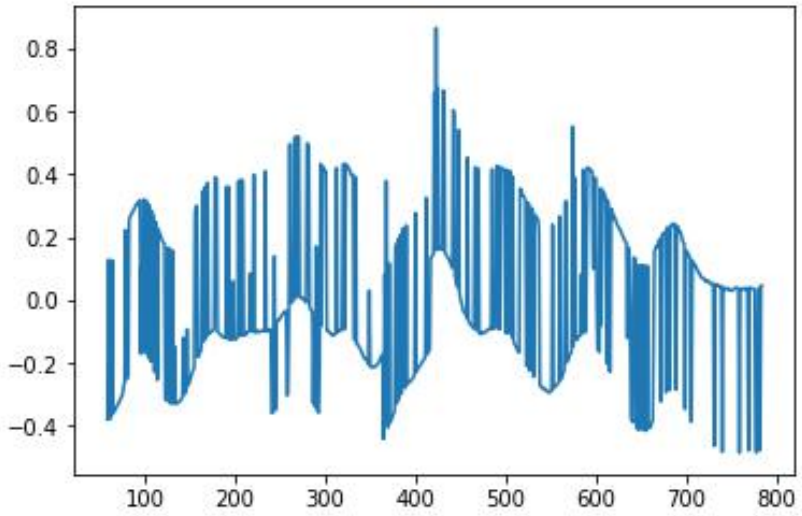
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14c4952898>
```



三十天移动平滑趋势图

```
moving_avg_diff = aluminum['电量(KWH)']-aluminum_windows60
moving_avg_diff.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14c4906438>
```



三十天差分图

优亿科技 USEEASE

数据归一化

数据的标准化（normalization）是将数据按比例缩放，使之落入一个小的特定区间。在某些比较和评价的指标处理中经常会用到，去除数据的单位限制，将其转化为无量纲的纯数值，便于不同单位或量级的指标能够进行比较和加权。其中最典型的就是数据的归一化处理，即将数据统一映射到[0,1]区间上。

归一化的数据使得数据在各个维度伸缩均匀化，模型更快速的拟合，同时保证最优解的等价性。

影响因素

通过上面的探索性分析，我们发现云南铝业其作为上市公司，其用电量和其股票的交易价格、交易金额等存在着一定的相关性，同时其星期、月份、节假日有一定的相关性。因此，我们选择“股票交易价格”、“月份”、“春节”等因素作为变量。

数据建模：以云南铝业为例

评估指标选择

由于这次的问题是一个回归问题，因此评估指标采用最常用的均方误差(mean-square error,MSE)作为评估指标。

均方误差（mean-square error, MSE）是反映估计量与被估计量之间差异程度的一种度量。设t是根据子样确定的总体参数θ的一个估计量，$(θ-t)^2$的数学期望，称为估计量t的均方误差。它等于σ2+b2，其中σ2与b分别是t的方差与偏倚。

一般地，在样本量一定时，评价一个点估计的好坏标准使用的指标总是点估计 $\hat{\theta}$ 与参数真值 θ 的距离的函数，最常用的函数是距离的平方，由于估计量 $\hat{\theta}$ 具有随机性，可以对该函数求期望，这就是下式给出的均方误差：

$$MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2.$$

实验

本次实验采用了三个模型，并进行比较：

（1）用于预测时间序列的ARIMA模型。

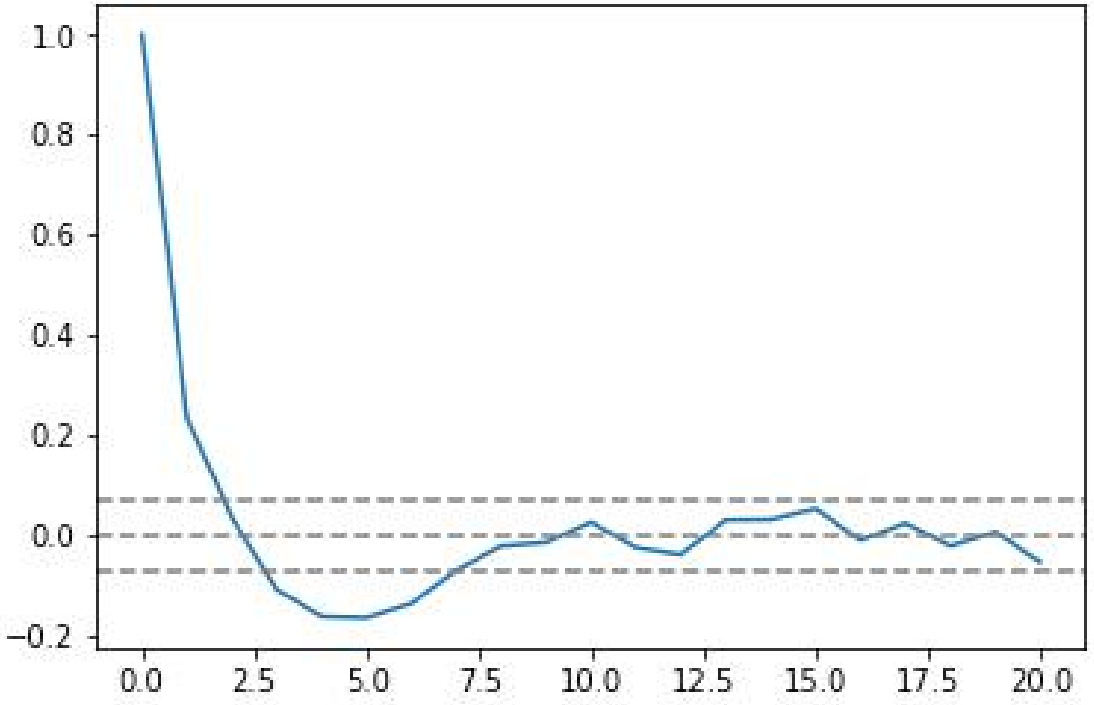
（2）xgboost+移动平滑窗口模型。

（3）LSTM循环神经网络模型。

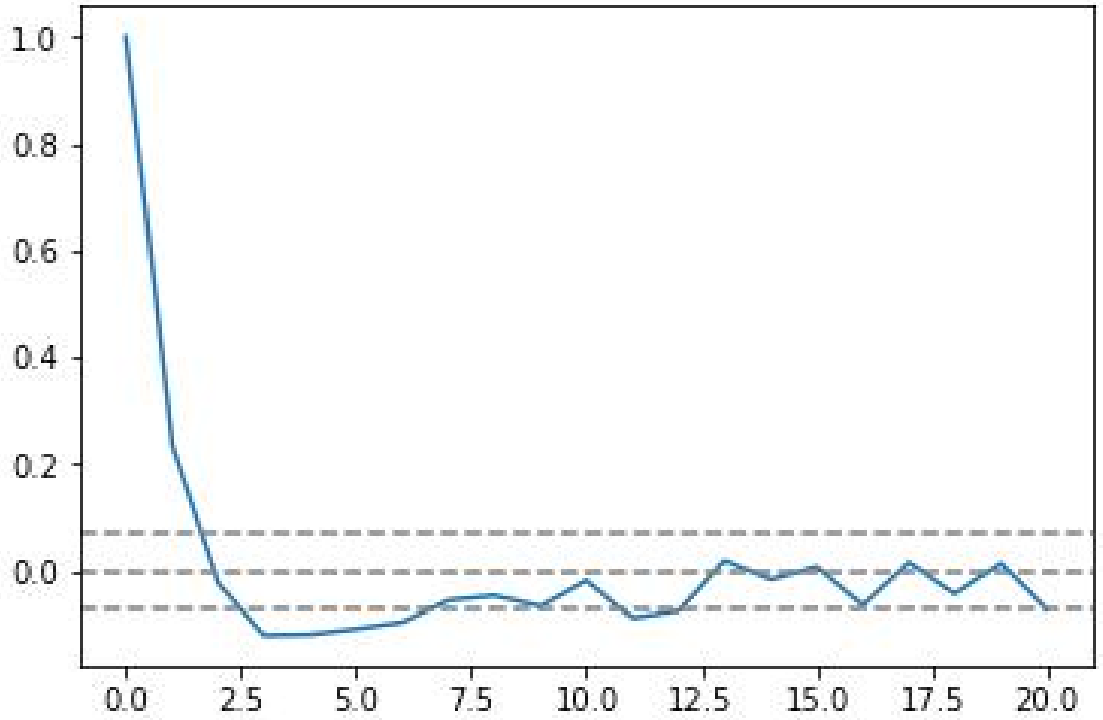
基准模型：ARIMA模型

ARIMA模型全称为自回归积分滑动平均模型(Autoregressive Integrated Moving Average Model,简记ARIMA)。ARIMA（p，d，q）称为差分自回归移动平均模型，AR是自回归， p为自回归项； MA为移动平均，q为移动平均项数，d为时间序列成为平稳时所做的差分次数。所谓ARIMA模型，是指将非平稳时间序列转化为平稳时间序列，然后将因变量仅对它的滞后值以及随机误差项的现值和滞后值进行回归所建立的模型。ARIMA模型根据原序列是否平稳以及回归中所含部分的不同，包括移动平均过程（MA）、自回归过程（AR）、自回归移动平均过程（ARMA）以及ARIMA过程。

基准模型：ARIMA模型



自相关图（Autocorrelation，acf）



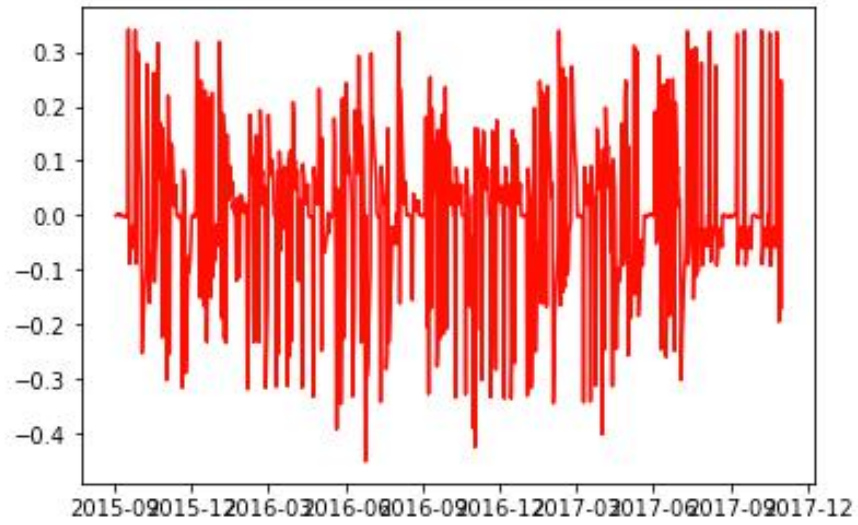
偏自相关图(Partial Autocorrelation,pacf)

基准模型：ARIMA模型

拟合训练的差分图

```
#AR model
model=ARIMA(aluminum['电量(KWH)'],order=(7,1,0))
result_AR=model.fit(disp=-1)
plt.plot(result_AR.fittedvalues,color='red')
```

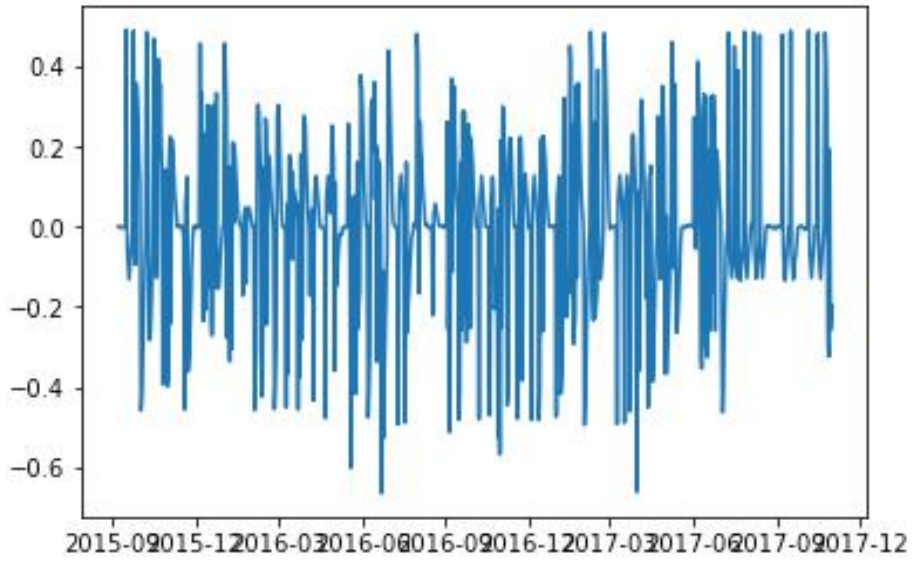
```
[<matplotlib.lines.Line2D at 0x7f14bf522470>]
```



真实的差分图

```
plt.plot( - moving_avg_diff)
```

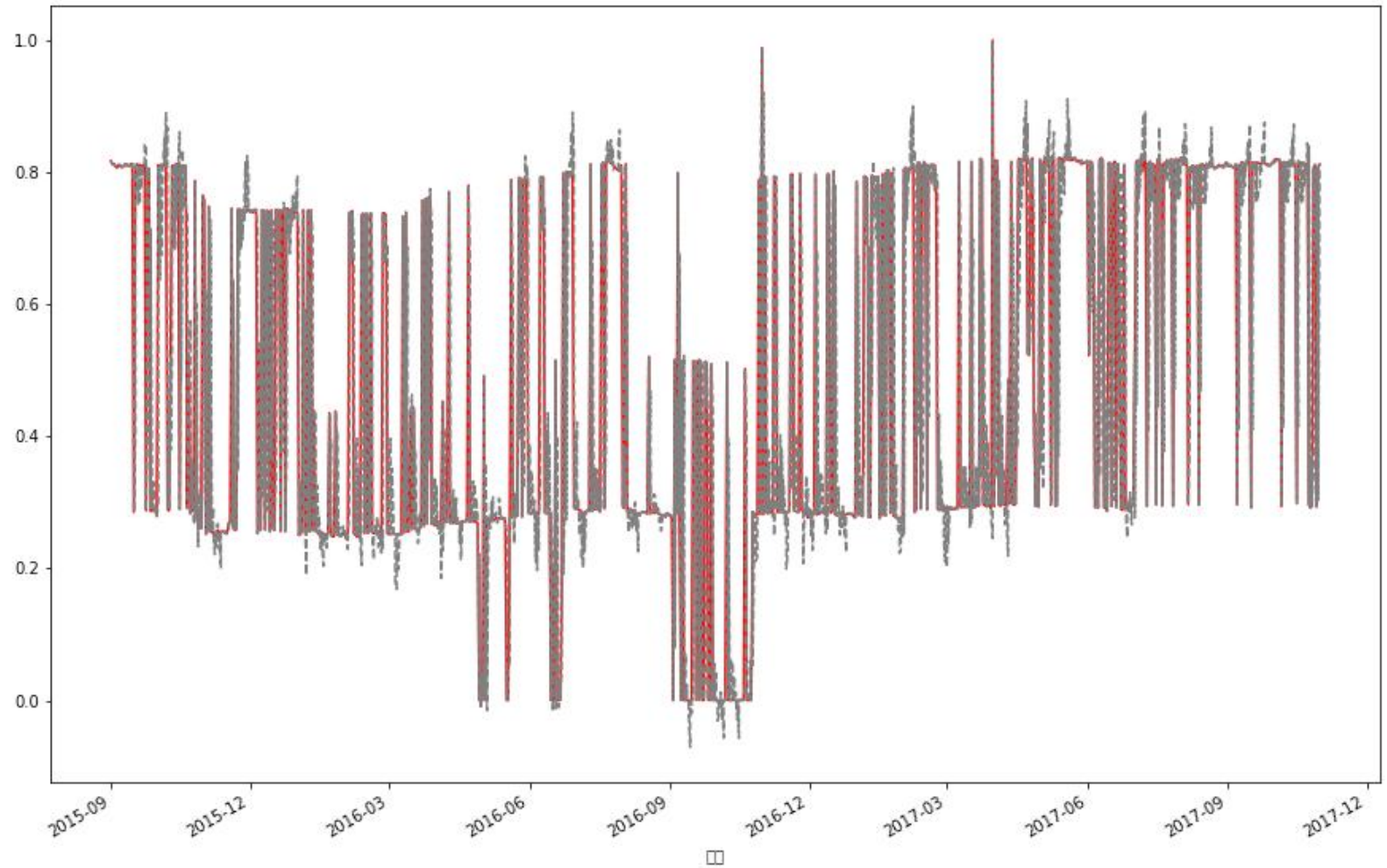
```
[<matplotlib.lines.Line2D at 0x7f14bf4ea7b8>]
```



基准模型：ARIMA模型

```
fig = plt.figure(1,figsize=[15,10])
result['电量(KWH)'].plot(color='red')
result[0].plot(linestyle='—',color='gray')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14bf6227b8>
```

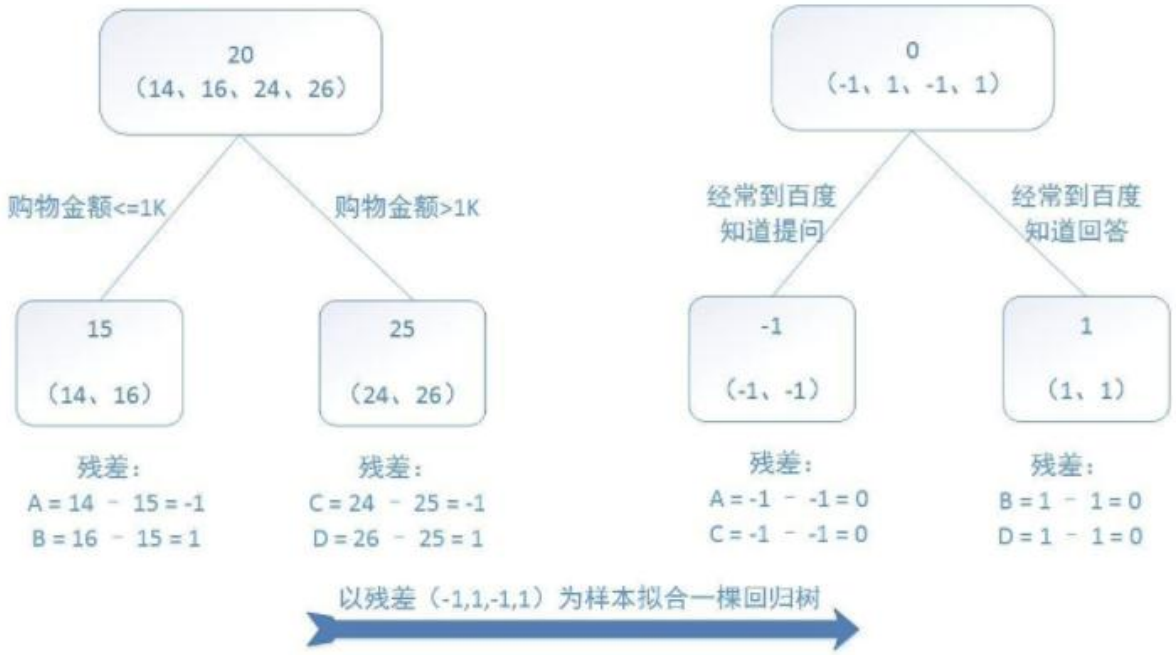


数据回测：MSE为0.027835

xgboost+移动平滑窗口

xgboost是一种提升树算法，其作为大规模并行boosted tree的工具，是目前最快最好的开源boosted tree工具包，是一种迭代的决策树算法，该算法由多棵决策树组成，所有树的结论累加起来做最终答案。

提升树是迭代多棵回归树来共同决策。当采用平方误差损失函数时，每一棵回归树学习的是之前所有树的结论和残差，拟合得到一个当前的残差回归树，残差的意义如公式：残差 = 真实值 - 预测值 。提升树即是整个迭代过程生成的回归树的累加。



xgboost+移动平滑窗口

训练效果

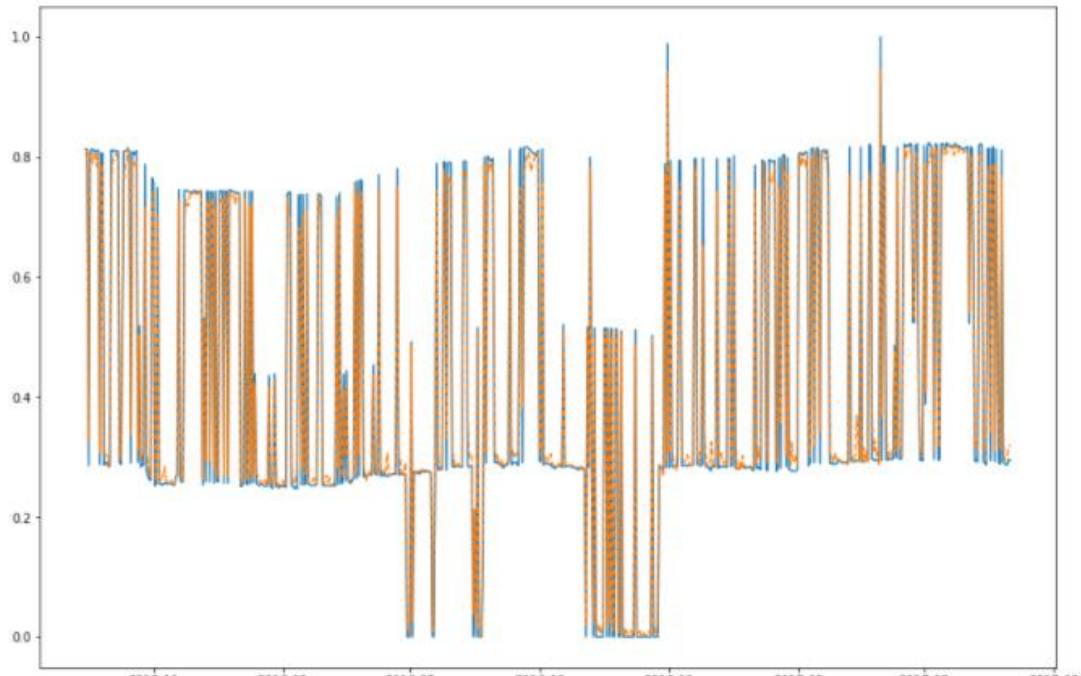
```
eval_set=[(train_x, train_y), (test_x, test_y)]
clf_xgb_without_weight.fit(train_x, train_y,
                           eval_set=eval_set,
                           early_stopping_rounds=100,
                           eval_metric='rmse',
                           verbose=20
                          )
```

```
[0]    validation_0-rmse:0.271691      validation_1-rmse:0.299444
[20]   validation_0-rmse:0.237208      validation_1-rmse:0.270882
[40]   validation_0-rmse:0.208949      validation_1-rmse:0.249755
[60]   validation_0-rmse:0.185714      validation_1-rmse:0.236353
[80]   validation_0-rmse:0.164424      validation_1-rmse:0.226363
[100]  validation_0-rmse:0.146348      validation_1-rmse:0.219598
[120]  validation_0-rmse:0.132187      validation_1-rmse:0.214978
[140]  validation_0-rmse:0.120617      validation_1-rmse:0.21203
[160]  validation_0-rmse:0.110168      validation_1-rmse:0.209903
[180]  validation_0-rmse:0.100872      validation_1-rmse:0.207859
[200]  validation_0-rmse:0.093613      validation_1-rmse:0.206098
[220]  validation_0-rmse:0.087463      validation_1-rmse:0.204431
[240]  validation_0-rmse:0.082164      validation_1-rmse:0.20382
[260]  validation_0-rmse:0.077094      validation_1-rmse:0.203467
[280]  validation_0-rmse:0.07199       validation_1-rmse:0.203096
[300]  validation_0-rmse:0.066726      validation_1-rmse:0.202193
[320]  validation_0-rmse:0.061831      validation_1-rmse:0.201671
[340]  validation_0-rmse:0.058042      validation_1-rmse:0.201152
[360]  validation_0-rmse:0.054634      validation_1-rmse:0.201004
[380]  validation_0-rmse:0.0516        validation_1-rmse:0.200149
[400]  validation_0-rmse:0.048651      validation_1-rmse:0.199964
[420]  validation_0-rmse:0.046078      validation_1-rmse:0.200016
[440]  validation_0-rmse:0.043935      validation_1-rmse:0.200035
[460]  validation_0-rmse:0.041968      validation_1-rmse:0.200134
[480]  validation_0-rmse:0.040051      validation_1-rmse:0.199889
```

数据回测：MSE为0.00340203

```
pred_y = clf_xgb_without_weight.predict(train_x)
```

```
# 画图
fig = plt.figure(1,figsize=[15,10])
plt.plot(train.index, train_y)
plt.plot(train.index, pred_y, '—')
plt.show()
```

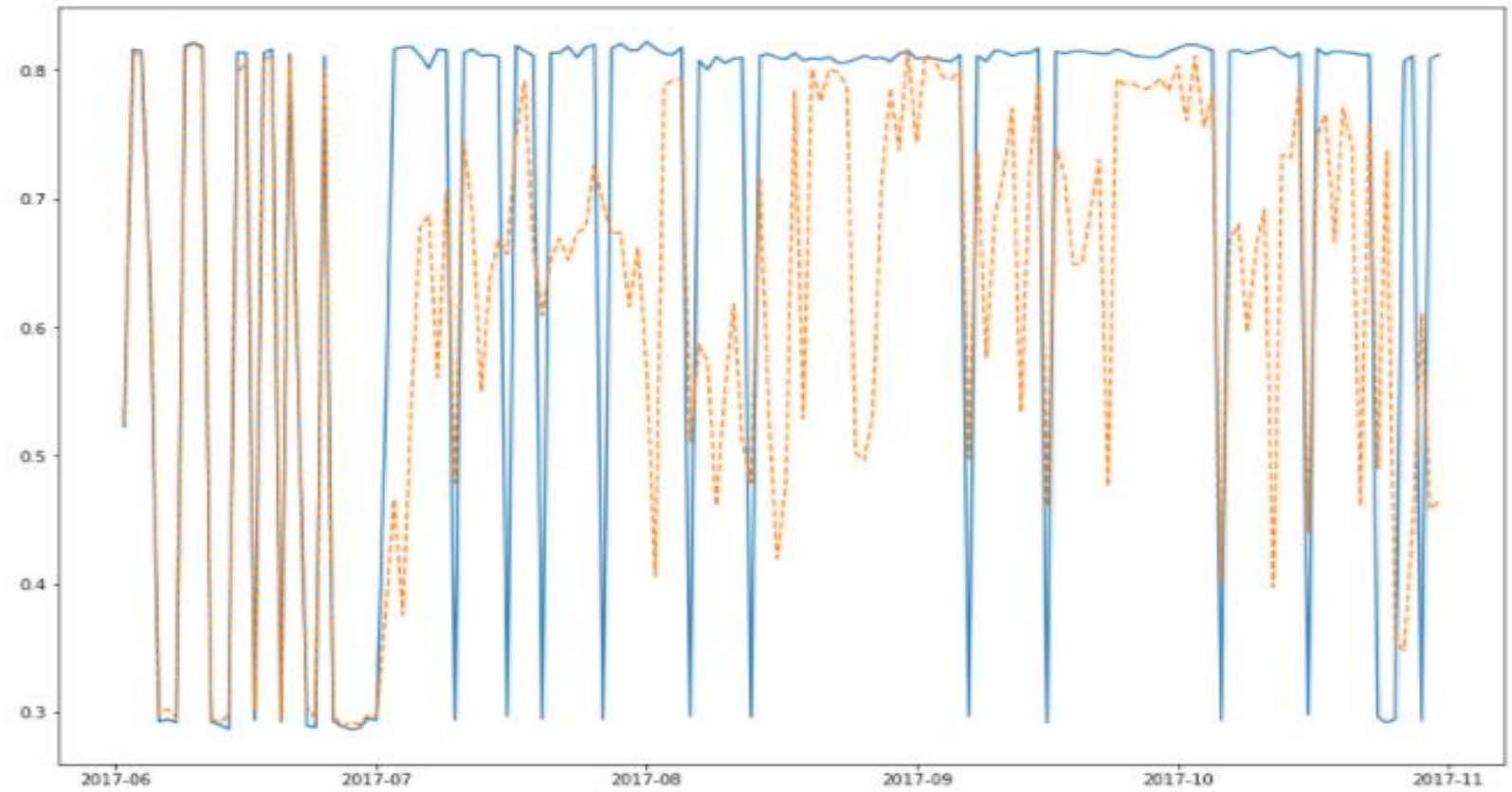


优亿科技 USEEASE

xgboost+移动平滑窗口

实际预测效果：MAE为0.124379

```
# 画图
fig = plt.figure(1,figsize=[15,10])
plt.plot(test.index, test_y)
plt.plot(test.index, pred_y, '--')
plt.show()
```

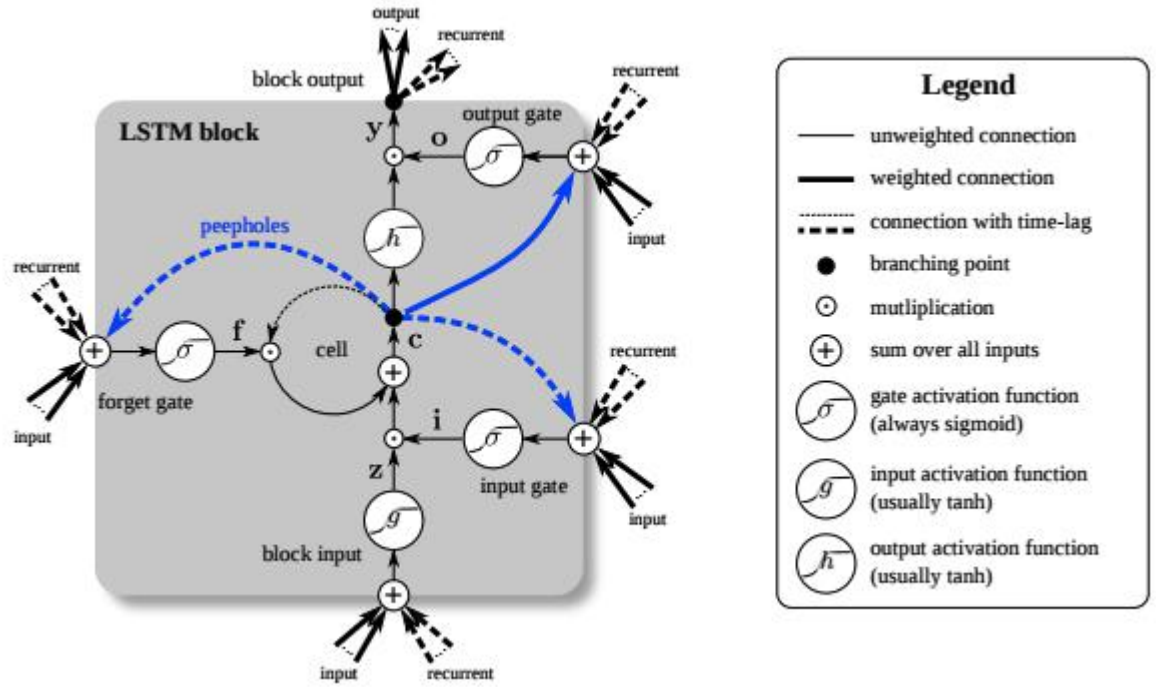


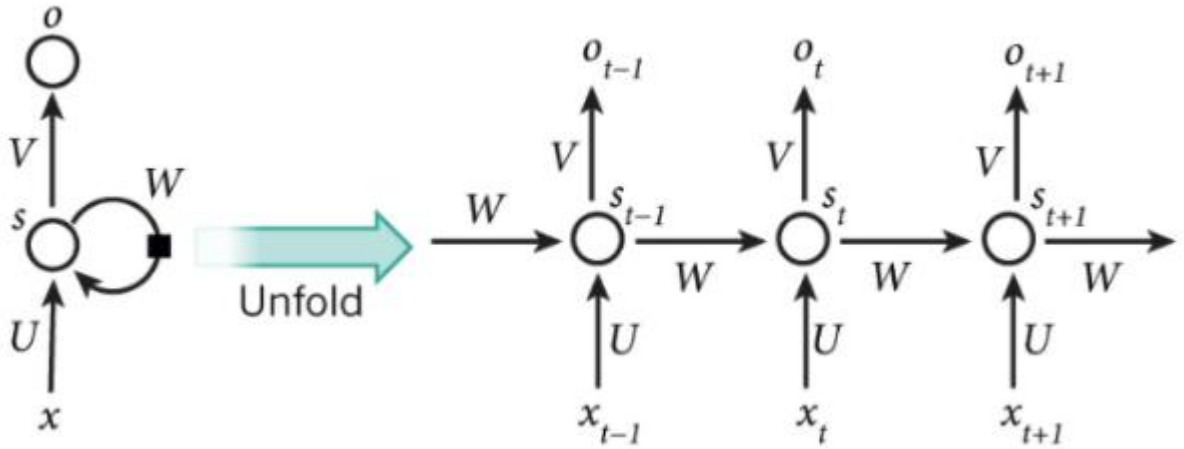
循环神经网络模型：LSTM时序算法预测

时间序列模型最常用的深度学习算法是递归神经网络（recurrent neural network, RNN）。相比与普通神经网络的各计算结果之间相互独立的特点，RNN的每一次隐含层的计算结果都与当前输入以及上一次的隐含层结果相关。

通过这种方法，RNN的计算结果便具备了记忆之前几次结果的特点。

LSTM模型是一种RNN的变型，最早由Juergen Schmidhuber提出的。

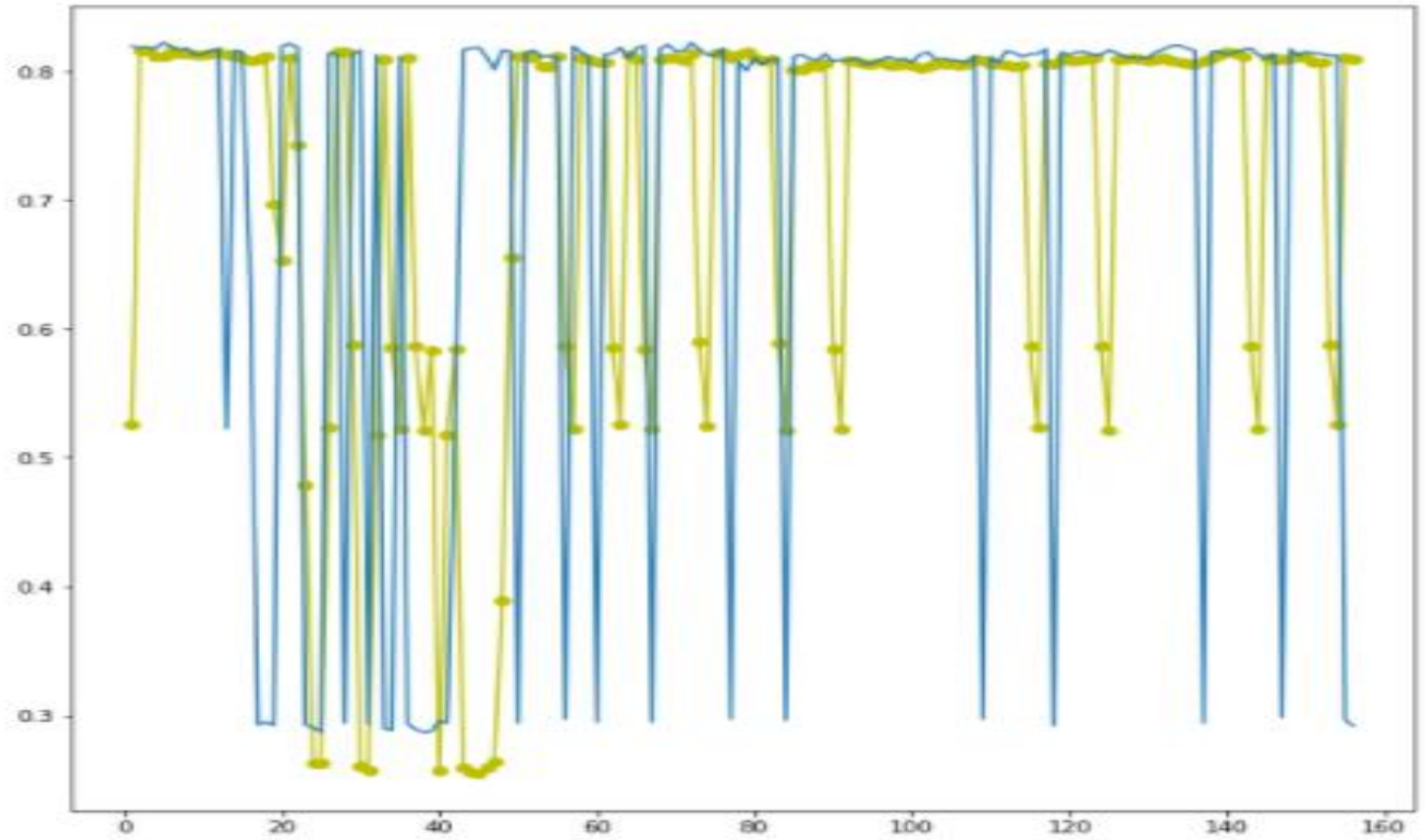




循环神经网络模型：LSTM时序算法预测

实际预测效果：MAE为0.104879

```
# 画图
fig = plt.figure(1,figsize=[10,10])
plt.plot(x, predict_y, 'yo-')
plt.plot(x, test_y)
plt.show()
```



总结

总结与改进

（1）云铝的用电量和其股价、月份、春节具有一定的相关性。

（2）三个模型各有优缺点，相对来说基于深度学习的循环神经网络效果最好。

（3）由于特定企业数据存在着企业内部特定的影响因素，因此比行业预测的难度要更大，需要企业内部信息的补充。

后续改进

（1）寻找云铝周期变化的因素，补充新的数据。

（2）用多时序模型进行组合预测。


Thanks for
your attention!