Cubist Data Exercise

Xinlong Li

1 Problem Description

The aim of this data exercise is to test a data source, `df['Signal']`, which claims to be predictive of future returns of the SP500 index (use SPY as a proxy). There will be two main parts in the following text. The first part performs a data cleaning on the given data-frame *df*, identifying any errors in the data. flagging them , and suggest a corrected value or if advisable, There are also illegal values which are directly deleted. The second part presents a time series analysis to check if `df['Signal']` could be used to predict future values of `df['ClosePrice']` from SP500.

There are three columns in the data-frame `df`. (1)`df['Date']`. (2)`df['Signal']`, which may be predictive of future returns of the SP500 index (use SPY as a proxy). (3) `df['ClosePrice']`, which is the SPY price.

2 Data Cleaning

Firstly, we should check if there are missing values. There are 6 missing dates which are actually trading days but are absent in the data-frame. They are

```
2013-01-14  2013-01-15  2013-01-16  2013-01-17
2014-01-06  2014-02-11
```

The first 4 dates are continuous and we fill them with interpolation. The last 2 dates are single dates and we impute with previous value. We also check if the dates are unique.

Then we check if all the dates are trading days. We find that there are 4 days which are illegal.

```
2013-12-25  is  Christmas Day
2014-01-01  is  New Year's Day
2014-02-08  and  2014-02-09  are  weekends
```

So we drope them with their corresponding rows.

Next, we check if there are outliers in `df['Signal']`. From Figure 1, we can see there are 2 data points which are significantly larger than others. They are

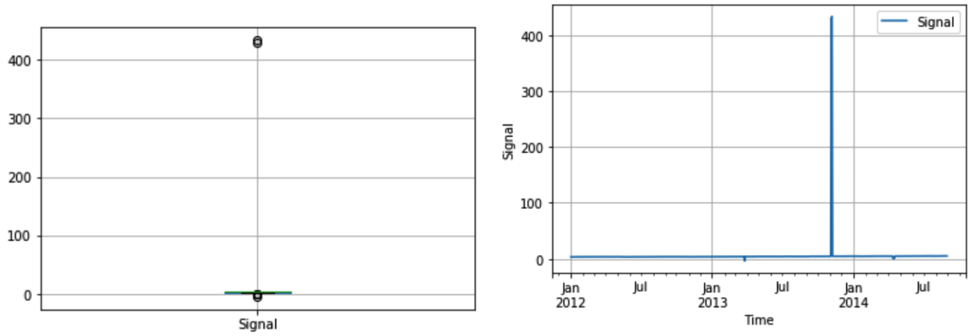

Figure 1: Show outliers at large values. Left: Box-plot of `df['Signal']`. Right: `df['Signal']` vs `df['Date']`

$$2013-11-05,\ \ 2013-11-06$$

we replace them with interpolated values and check again. We can see from Figure 2 that

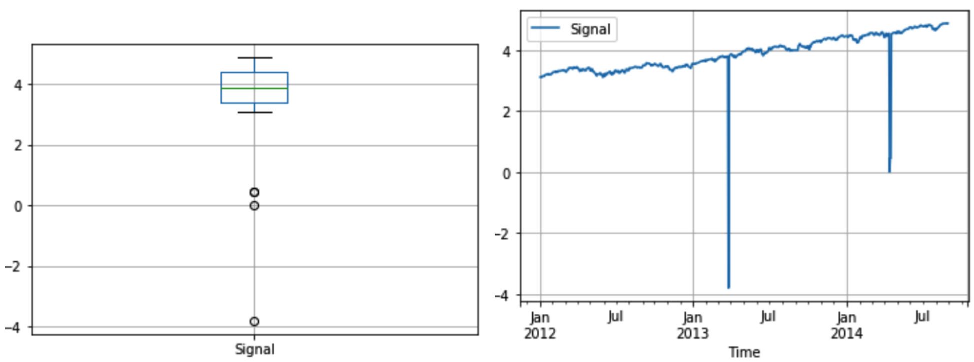

Figure 2: After fixing the outliers at large values. Show outliers at small values. Left: Box-plot of `df['Signal']`. Right: `df['Signal']` vs `df['Date']`

there are also a few data points which are significantly smaller than others. They are

$$\begin{array}{l} 2013-03-26 \\ 2014-04-14\ \ 2014-04-15\ \ 2014-04-16. \end{array}$$

After filling modifying these data points, we perform the same action on `df['ClosePrice']` and find 3 outliners

$$2013-09-12\ \ 2013-09-13\ \ 2013-09-16$$

Finally, we can plot `df['Signal']` and `df['ClosePrice']` on top of each other (shown in Figure 3). `df['ClosePrice']` is approximately 40 times bigger than `df['Signal']`.

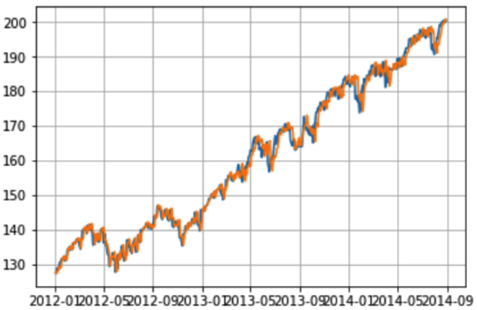

Figure 3: 41×`df['Signal']` (blue) and `df['ClosePrice']` (orange) vs `df['Dates']` after data cleaning.

3 Time Series Analysis

Firstly, we have a glance on the decomposition plot of `df['Signal']`(shown in Figure 4).

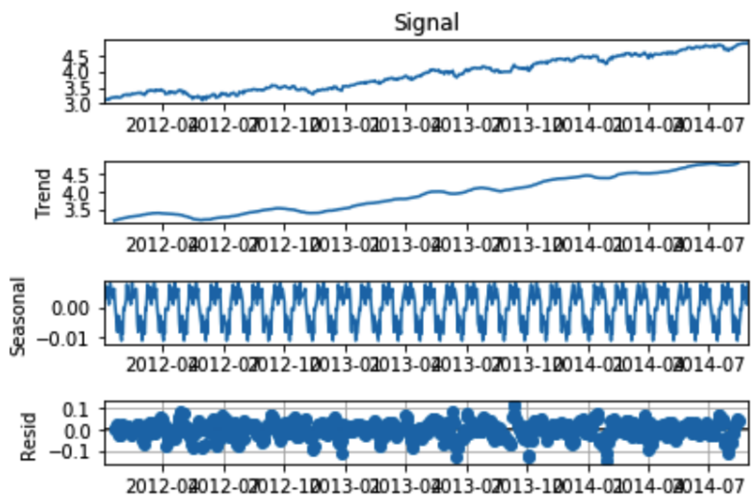

Figure 4: Decomposition plot of `df['Signal']`.

The amplitude of seasonal curve is quite small. So we ignore the seasonality of time series `df['Signal']`.

3.1 Check Stationary

From Fgure 4 above we can see that the time series is not stationary. We confirm this using Augmented Dickey Fuller (ADF) Test.

```
Results of ADF Test:
                   Test Statistic                   -0.306330
                   p-value                           0.924603
                   #Lags Used                        2.000000
                   Number of Observations Used     666.000000
                   Critical Value (1%)              -3.440207
                   Critical Value (5%)              -2.865889
                   Critical Value (10%)             -2.569086
```

The test statistic is bigger than the critical values and the p-value is big, which cannot reject the null hypothesis. This implies that the time series `df['Signal']` is not stationary.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test also gives the same answer.

```
Results of KPSS Test:
                   Test Statistic              3.241585
                   p-value                     0.010000
                   Lags Used                  20.000000
                   Critical Value (10%)        0.347000
                   Critical Value (5%)         0.463000
                   Critical Value (2.5%)       0.574000
                   Critical Value (1%)         0.739000
```

With the test statistic bigger than the critical values and p-value smaller than 0.05, we reject the null hypothesis, which confirms again that the time series `df['Signal']` is not stationary.

We do a quick differencing on the time series `df['Signal']` to make it stationary.

```
df['Signal_diff'] = df['Signal'] - df['Signal'].shift(1)
df['Signal_diff'].dropna().plot()
adf_test(df['Signal_diff'].dropna())
kpss_test(df['Signal_diff'].dropna())
```

```
Results of ADF Test:
Test Statistic                  -20.202023
p-value                           0.000000
#Lags Used                        1.000000
Number of Observations Used     666.000000
Critical Value (1%)              -3.440207
Critical Value (5%)              -2.865889
Critical Value (10%)             -2.569086
```

```
Results of KPSS Test:
Test Statistic              0.050405
p-value                     0.100000
Lags Used                  20.000000
Critical Value (10%)        0.347000
Critical Value (5%)         0.463000
Critical Value (2.5%)       0.574000
Critical Value (1%)         0.739000
```

3.2 ARIMA Model

Let's have a look at the Auto Correlation Function (ACF) figure and Partial Auto Correlation Function (PACF) figure of `df['Signal_diff']` (shown in Figure).

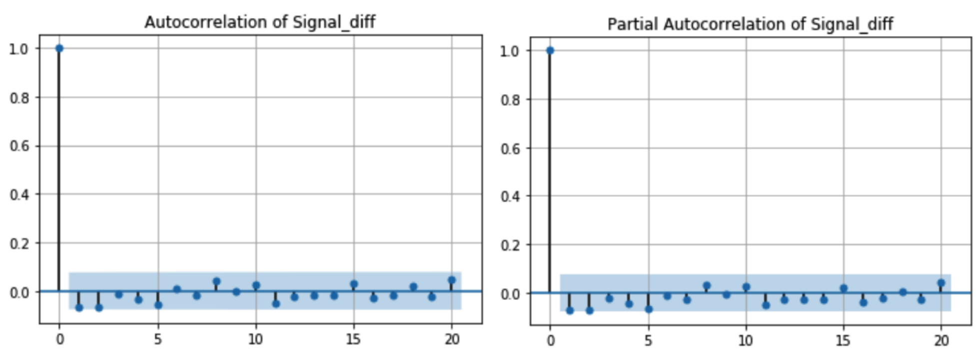

Figure 5: ACF and PACF of `df['Signal_diff']`

We can conclude that p and q should be smaller than 2 in ARIMA(p,d,q) Model. We set d = 1 since we did a differencing on the column `df['Signal']`. In the end, we find that p = q = 1 has lower AIC. So we try (p,d,q) = (1,1,1)

The result of ARIMA(1,1,1) model is shown in Figure 6. The autoregressive term has a p-value that is less than the significance level of 0.05. So I can conclude that the coefficient for the autoregressive term is statistically significant. The same with moving average term.We also plot the residue in Figure 7. the residues follow the normal distribution. And no significant correlation is present, we can conclude that the residuals are independent. Overall, it seems to be a good fit.

In addition, we can run a Ljung Box Test on the residues and the square value of residues in ARIMA(1,1,1) model, which gives a p-value of 0.18 and 0.74, respectively. They are both larger than 0.05, which implies our model does not show lack of fit.

We show the figure of ARIMA(1,1,1) model fitting `df['Signal']`. Figure 8 shows that the ARIMA(1,1,1) model can fit `df['Signal']` very well. so let's use it to forecast

```
                              ARIMA Model Results
==============================================================================
Dep. Variable:               D.Signal   No. Observations:                  668
Model:                 ARIMA(1, 1, 1)   Log Likelihood                1393.087
Method:                       css-mle   S.D. of innovations              0.030
Date:                Mon, 20 Apr 2020   AIC                          -2778.175
Time:                        05:36:13   BIC                          -2760.158
Sample:                    01-04-2012   HQIC                         -2771.195
                         - 08-29-2014
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const              0.0026      0.000     18.386      0.000       0.002       0.003
ar.L1.D.Signal     0.9627      0.011     87.747      0.000       0.941       0.984
ma.L1.D.Signal    -1.0000      0.008   -120.710      0.000      -1.016      -0.984
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            1.0388           +0.0000j            1.0388            0.0000
MA.1            1.0000           +0.0000j            1.0000            0.0000
-----------------------------------------------------------------------------
```

Figure 6: Result of ARIMA(1,1,1) training on `df['Signal']`

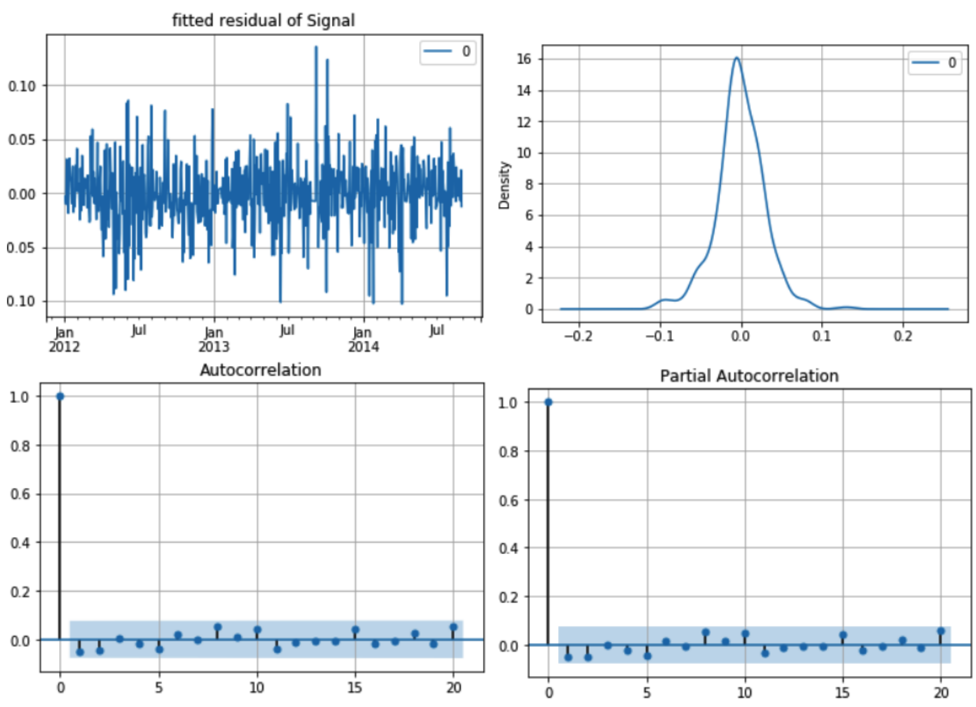

Figure 7: Top left: Residues vs `df['Date']`. Top right: distribution of residues. Bottom left: ACF of residues. Bottom right: PACF of residues

Figure 8: `df['Signal']` curve and its fitting curve.

`df['ClosePrice']` of SP500 to see if `df['Signal']` could be predictive of future returns of the SP500 index.

We use the last 100 rows of data-frame `df` as test set and the rest of the rows as training set. We use `df['Signal']` of the training set to train the ARIMA(1,1,1) model. Then we forecast the `df['ClosePrice']` in the test set using the model. Note that we divided the `df['ClosePrice']` by 38 to make `df['ClosePrice']` and `df['Signal']` equal at the starting date of the test set. From figure 9, the ARIMA(1,1,1) model seems to give a

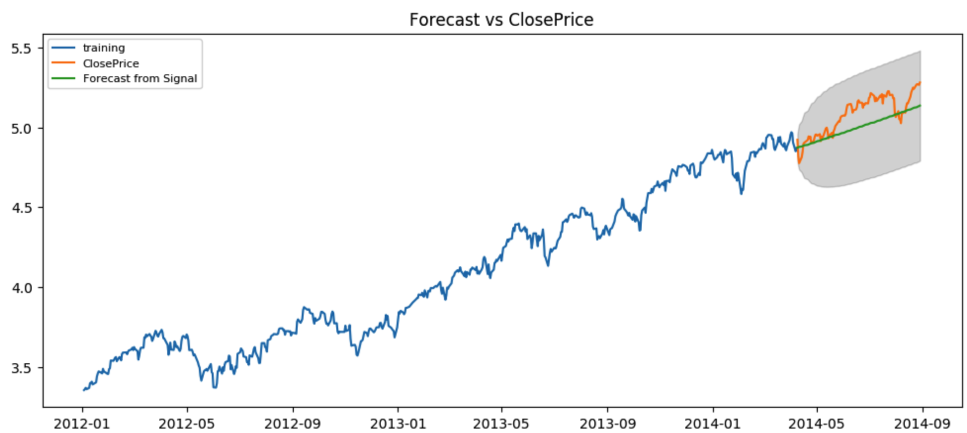

Figure 9: Forecasting of future `df['ClosePrice']` data by training the past `df['Signal']` data using ARIMA(1,1,1) model. The grey area means 95% confidence.

directionally correct forecast. And the actual observed `df['ClosePrice']` lies within the 95% confidence band ($\alpha = 0.05$).

In the end, we calculate accuracy metrics in Figure 10. The MAPE, Correlation and Min-Max Error are used to evaluate the forecast. Around 1.7% MAPE implies the model is about 98.3% accurate in predicting the next 100 trading days. So I think this forecast is quite good.

```
In [34]: 1 arima111.forecast_accuracy()

Out[34]: {'mape': 0.017322513026562892,
          'corr': 0.8557742214558138,
          'minmax': 0.017311234266007558}
```

Figure 10: Calculating the accuracy matrices

4 Conclusion

From this data exercise, we conduct a time series analysis to test the predictive power of `df['Signal']`. In the data cleaning part, we find 4 illegal dates which are not trading days, 6 missing dates which are actually trading days but are absent in the data-frame, 6 outliers belonging to `df['Signal']` and 3 outliers belonging to `df['ClosePrice']`.

In the time series analysis part. We check the stationary, perform a ARIMA model fitting, analyze the residues and forecast the close price. All of the above shows that `df['Signal']` can be predictive of future returns of the SP500 index (use SPY as a proxy).

However, we should tell the Portfolio Manager that `df['ClosePrice']`. is APPROXIMATELY 40 times bigger than `df['Signal']`. We should calculate this factor today before we forecast the close price tomorrow.