Holt Winter's Exponential Smoothing (HWES)

1.

The whole algorithms are divided into the following parts.

(1) Setting up the python environment and loading the dataset;

(2) Understanding the structure and content of the dataset;

Specifically speaking, based on the dataset we used, we used "print(train.columns)" and "print(test.columns)" to figure out the variables, and then we tried to understand their meaning. We also attempted to find out other features of the dataset like its data type, its concise information and so on.

(3) We initially plotted tendency of the dataset to figure out whether it meets the requirements of time series.

(4) Based on the hypotheses, we preliminarily graphed the dataset and test the hypotheses.

(5) We imported Exponential Smoothing Model, using fit to create the model and predicting the results.

From our assumption, we expected that the dataset would show a tendency with time passed by, and it would fit seasonal trend. Holt Winter's Exponential Smoothing Model can set up satisfying model when the dataset will change in a regularity with time and it demonstrates a seasonal pattern. That is to say, the change of the dataset may display a recycled pattern, repeating periodically.

2.

The Holt-Winter's Exponential Smoothing Model contains three variables, which are level component, trend component and seasonal component, the formulas are displayed as follows:

$$\text{Level component: } a_t = \alpha(A_t - S_{t-s}) + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$\text{Trend component: } b_t = \beta(a_t - a_{t-1}) + (1 - \beta)\, b_{t-1}$$

$$\text{Seasonal component: } S_t = \gamma(A_t - a_t) + (1 - \gamma)S_{t-s}$$

Here, α, β, and γ are the smoothing parameters for level, trend, and seasonal components, respectively. The forecast for period $t + 1$ is:

$$F_{t+1} = a_t + b_t + S_{t-s+1}$$

The forecast for k periods beyond the last period of observed data (period T) is

$$F_{T+k} = a_T + b_T k + S_{T-s+k}$$

3.

As is indicated by the formulas above, there are three components in Holt-Winters' Exponential Smoothing Model, the dataset ought to show these three components.

(1) Level

Expected value has another name, which varies depending on baseline, intercept, or level.

(2) Trend

Trend is the difference in the y coordinates.

(3) Season

If a series appears to be respective at regular intervals, such an interval is referred to as a season, and the series is said to be seasonal.

Seasonality is required for the Holt-Winters method to work, non-seasonal series (e.g. price stock) cannot be forecasted using this method.

4.

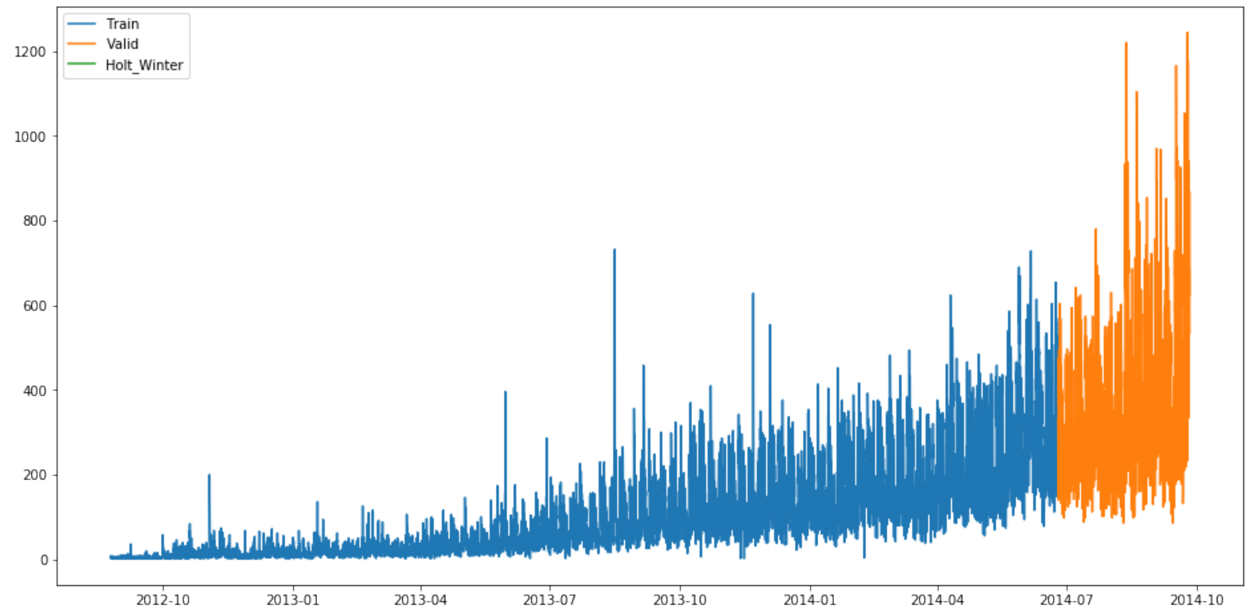
First and foremost, we have concerned about how to determine the figure of α, β, γ in the model. After reading a massive of python coding from other coders, we noticed that they frequently use "fitting" for more than once to find out the best figures for these components. In our model, we merely used "fitting" for once, and when we revised the figures, the graph seemed to be unchanged.

Subsequently, we also met a great number of issues that enabled the coding to work inefficiently. For instance, we encountered attribute error when running the code, also encountered convergence warnings.

By adapting the code shown as follows:

```
y_hat_avg = valid.copy()
fit2 = ExponentialSmoothing(np.asarray(Train['Count']) ,seasonal_periods=30 ,trend='add',
seasonal='add',).fit()
y_hat_avg['Holt_Winter'] = fit1.forecast(len(valid))
plt.figure(figsize=(16,8))
plt.plot( Train['Count'], label='Train')
plt.plot(valid['Count'], label='Valid')
plt.plot(y_hat_avg['Holt_Winter'], label='Holt_Winter')
plt.legend(loc='best')
plt.show()
```

We did not gain the trend we were expecting, on the contrary, we did not succeed in attaining the prediction, with the graph showing as follows.



5.
To run the algorithm, we adapted the following Python library:

- Pandas
- Numpy
- Matplotlib.pyplot
- Datetime
- Warnings
- Statesmodel.api

We also adapted the following functions/methods

- Print
- Figure
- Plot
- Groupby
- Temp
- Fit
- Count
- Length
- Index

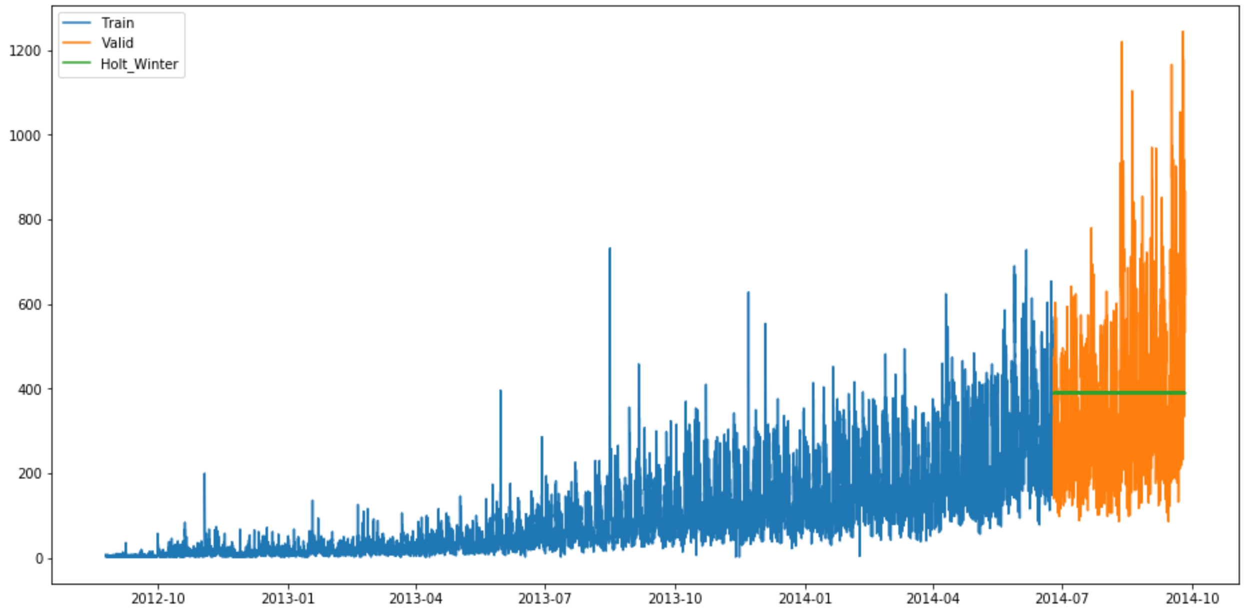
6.
a.
Library: Panda PIL
Function: eval() Method

b.

“The eval() method parses the expression passed to it and runs python expression(code) within the program.
The syntax of eval is:
eval(expression, globals=None, locals=None)

- expression: this string is parsed and evaluated as a Python expression
- globals (optional): a dictionary to specify the available global methods and variables.
- locals (optional): another dictionary to specify the available local methods and variables.”

c.



As was illustrated in the above pictures, the dataset had a trend of time series. The orange part was the actual data, the green part was the predict data by adapting Holt-Winters’ Exponential Smoothing Model. We initially assumed that the prediction result would have the similar tendency of the actual results, thereby we considered that the code did not work effectively and needed to be improved.

7.

Section	Links	Notes
1	https://www.statsmodels.org/stable/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html	
2	https://medium.com/datadriveninvestor/how-to-build-exponential-smoothing-models-using-python-simple-exponential-smoothing-holt-and-da371189e1a1	
3	https://machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/	
4	https://otexts.com/fpp2/holt-winters.html	
5	https://www.geeksforgeeks.org/eval-in-python/	Quata in Note 6(b)
6	https://medium.com/datadriveninvestor/holt-winters-exponential-smoothing-for-time-series-forecasting-5905fad390e5	
7	https://grisha.org/blog/2016/02/17/triple-exponential-smoothing-forecasting-part-iii/	

8	https://www.kaggle.com/kashnitsky/topic-9-part-1-time-series-analysis-in-python	
9	https://biologyforfun.wordpress.com/2018/04/09/help-i-have-convergence-warnings/	Solving convergence warning issues
10	https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Exponential_Smoothing-Trend_and_Seasonal.pdf	
11	https://docs.rapidminer.com/9.3/studio/operators/modeling/time_series/forecasting/holt-winters_trainer.html	
12	https://github.com/nchandra/ExponentialSmoothing	
13	https://github.com/cmdrkeene/holt_winters	
14	https://github.com/coder17173/TripleExponentialSmoothing	
15	https://datahack.analyticsvidhya.com/contest/practice-problem-time-series-2/	Online website of the data