Big Data Analytics Final Report

Stock prediction based on momentum and correlation

Di Zuo (dz2357), Jiayu Ni (jn2585)
Columbia University
dz2357@columbia.edu,

Abstract

We use 5-miniute stock data from stooq.com/db/h/ to analyze and compute the correlation between each stock pair, and apply such correlation to momentum trading strategy and predict the actual price and rise/drop of the stocks. We found that the influence from other stocks can be large or small, and correlation is one of the factors that explain the fluctuation of stock price. The overall accuracy of rise\drop prediction is around 60%, and we also visualize the price prediction result and provide a GUI. The machine learning method we adopt are Support Vector Machine (SVM) for binary (rise\drop) prediction, and Kernel Regression for continuous (actual prices) prediciotn.

Keywords-component; stock prediction; momentum; correlation; machine learning.

I. Introduction

We assume that stocks can impose influence, minor or major, positive or negative, on each other. By analyzing the pattern of fluctuation, we can find such correlation, and apply it to do prediction.

Momentum trading strategy is one kind of trading strategy that "buy stocks with high returns over the previous 3 to 12 months and sell stocks with poor returns over the same time period earn profits of about one percent per month for the following year." Though the results are verified and generally accepted, people are still debating the source of profits. [1]

What our project focuses is adopting concepts in momentum trading strategy and combine with correlation to do stock prediction.

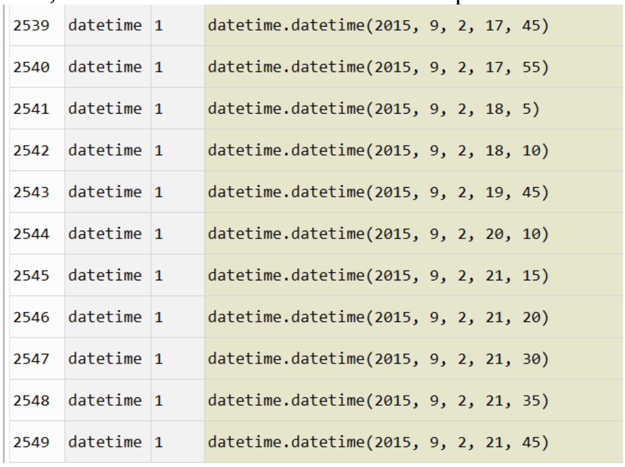
II. Related Works

Our idea and algorithms are mainly based on and inspired by Hong, K. J., and Stephen Satchell's work [2]. "The result shows that the Moving Average rule is popular because it can identify price momentum and is a simple way of tracing and exploiting price autocorrelation structure without necessarily knowing its precise structure." We adopted the concept of Moving Average and combine it with correlation.

III. System Overview

The dataset we obtained from stooq is in txt format and covers stocks prices in U.S. in recent years. The file *pre_5min - nysemkt stocks.py* will write the paths of all stock files to a txt file. The main program *correlation_5min.py* will show the load the dataset, build timeline, transform the dataset to lists in Python, and show GUI that can draw graphs.

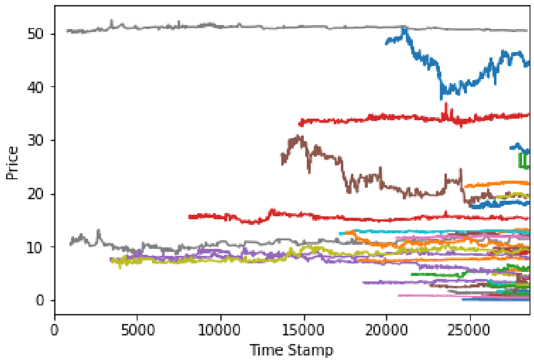
The raw dataset has stock prices every 5 minutes, and starts from 15:35 to 22:00 every week day. There are lots of missing data and we handled that by linearly interpolating[3] the missing ones. To deal with the actual time, we transformed the time to timestamp:



2539	datetime	1	datetime.datetime(2015, 9, 2, 17, 45)
2540	datetime	1	datetime.datetime(2015, 9, 2, 17, 55)
2541	datetime	1	datetime.datetime(2015, 9, 2, 18, 5)
2542	datetime	1	datetime.datetime(2015, 9, 2, 18, 10)
2543	datetime	1	datetime.datetime(2015, 9, 2, 19, 45)
2544	datetime	1	datetime.datetime(2015, 9, 2, 20, 10)
2545	datetime	1	datetime.datetime(2015, 9, 2, 21, 15)
2546	datetime	1	datetime.datetime(2015, 9, 2, 21, 20)
2547	datetime	1	datetime.datetime(2015, 9, 2, 21, 30)
2548	datetime	1	datetime.datetime(2015, 9, 2, 21, 35)
2549	datetime	1	datetime.datetime(2015, 9, 2, 21, 45)

Usually each timestamp is for 5 minutes.

Original graph of some of the stocks:



Since the data is not always complete for all stocks, we will not deal with the stock with insufficient data.

After transforming the raw dataset to lists that python can understand, we compute the correlations and apply Machine Learning algorithms and do the prediction.

IV. Algorithm

1. Momentum + Correlation

The difference between long term moving average (LMA) and short term moving average (SMA) is what momentum trading strategy focuses on. With the correlation of each stock pair, we have furthermore information: the influence from other stocks. The feature that can be extracted from the dataset is:

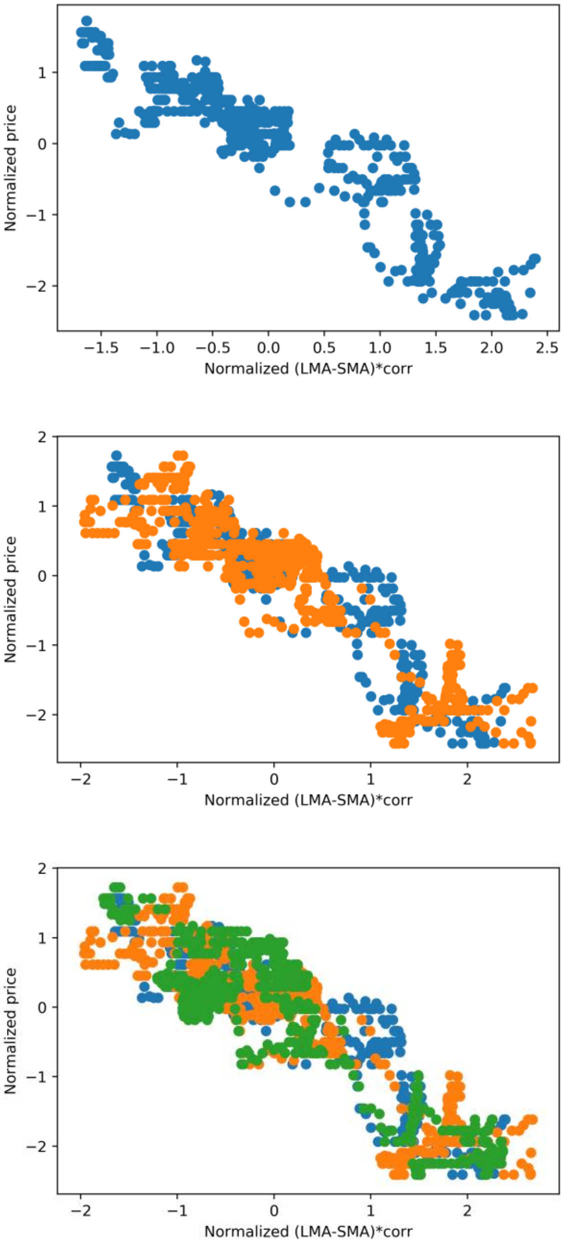
(LMA - SMA) * correlation

We adopt such algorithm: for a target stock, compute the correlation with all other stocks. Filter them and only take those high correlation. E.g. correlation > 0.9 or correlation < -0.9. Then we can form a dataset: the features of one stock are (LMA - SMA) * correlation of all other highly correlated stocks, and the label is rise\drop in the next time period. We also consider the price in the next time period and do regression to predict the actual price.

We will call (LMA – SMA) * correlation *"the difference"* for simplicity.

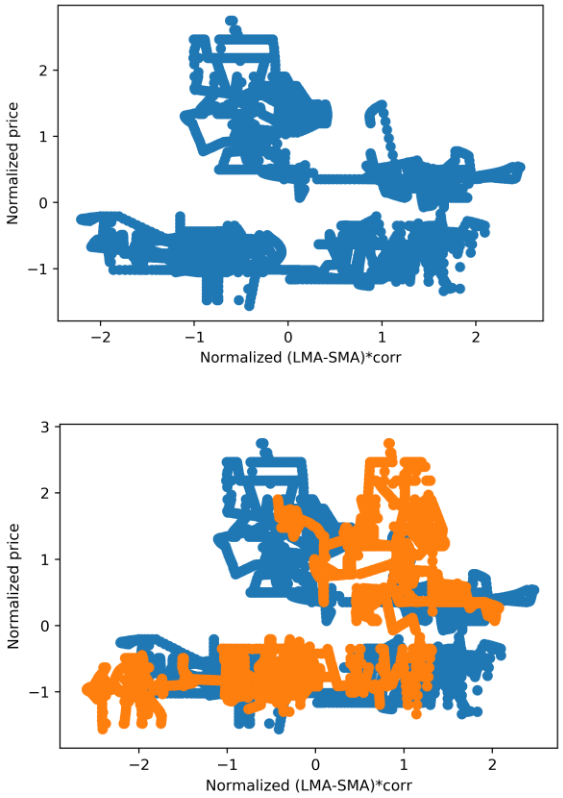
2. Machine Learning Algorithms

Before using choosing machine learning algorithms, we first visualize the dataset. This can help us pick suitable methods. Here is one example of the dataset for regression:



(Stock: ssy. Only showing three of the correlated stocks.)

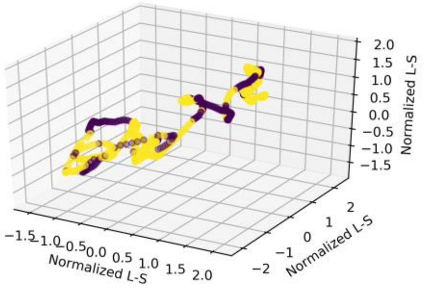
We can see from the graphs that there is some kind of relationship between the price and the "difference". But this is not always the case:



(Stock: ainc. Only showing two of the correlated stocks.) There is barely any relationship between the price and the “difference”. Hardly can we figure any method to capture the relationship and do regression.

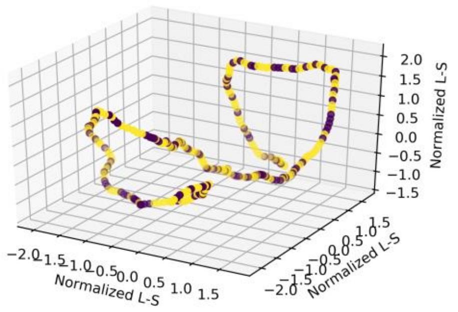
We tried linear regression and kernel regression, and the latter one outperforms the former one.

As for binary (rise\drop) dataset, such is the same situation:



(Stock: visi. Only showing randomly three of the correlated stocks.)

Boundaries between the rise and drop can be found.



(Stock: trxc. Only showing randomly three of the correlated stocks)
It is in a mess: whatever the differences are, the stock can rise or drop regardless.

We tried Decision Tree, Naïve Bayes (with Gaussian or multinomial model), and SVM. For such non-linear boundary, SVM works well.

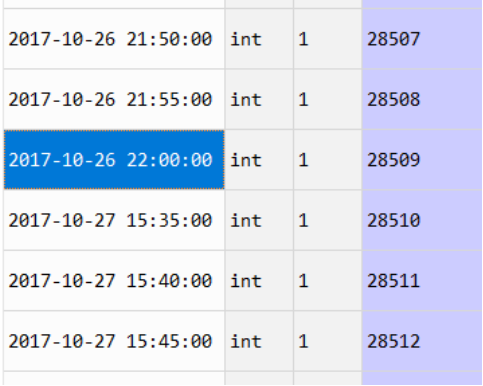
3. Parameters Picking

There are two important parameters that influence the algorithms: l and s in LMA and SMA. l is the number of points Long-term Moving Average looks at, and s is the number of points Short-term Moving Average looks at. For and instance, if l=100 and s=5, the LMA of time t is the average of data points from t-99 to t, the SMA is average of data points from t-4 to t. This strongly influence the accuracy of the models.

Our method of picking these two parameters is dynamic: we consider the valid length of data of given stock, and then determine l and s:

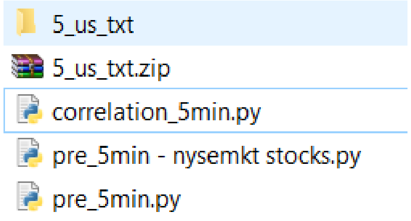
```
l = max(500, validLength/6)
s = max(30, l/100)
if s>50:
    s=50
```

In this way, l is at least 500 and is greater if there is much valid data; s is at least 30 and at most 50, or is l/100 if between 30 to 50. Note that the dataset is by 5 minutes, 12 timestamps is one hour, and data of one day is from 15:35 to 22:00, so 77 timestamps is one day, 500 timestamps is 6.5 days, or one week plus 1.5 days, since market closes on weekends.



2017-10-26 21:50:00	int	1	28507
2017-10-26 21:55:00	int	1	28508
2017-10-26 22:00:00	int	1	28509
2017-10-27 15:35:00	int	1	28510
2017-10-27 15:40:00	int	1	28511
2017-10-27 15:45:00	int	1	28512

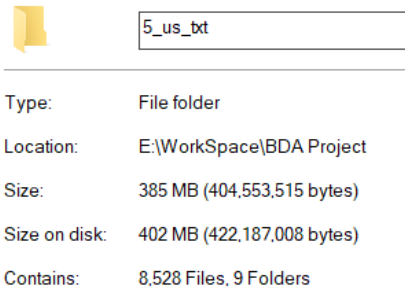
V. Software Package Description



5_us_txt
5_us_txt.zip
correlation_5min.py
pre_5min - nysemkt stocks.py
pre_5min.py

Preparation:

Download 5 minutes dataset of stocks in U.S. from stoop and unzip. Run *pre_5min.py* (or *pre_5min – nysemkt stocks.py* to do test only on 369 stocks to save time)



5_us_txt

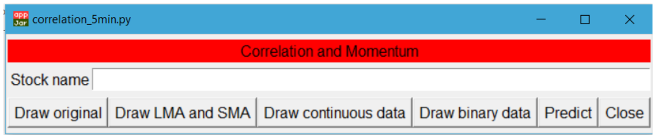
Type:	File folder
Location:	E:\WorkSpace\BDA Project
Size:	385 MB (404,553,515 bytes)
Size on disk:	402 MB (422,187,008 bytes)
Contains:	8,528 Files, 9 Folders

Dataset is 385 MB.

This will use pickle to generate a file that contains paths of all stock data.

Then run *correlation_5min.py* to do the analysis.

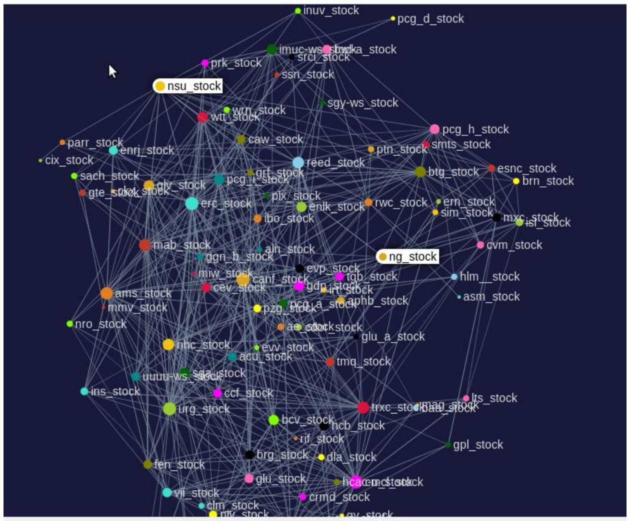
GUI



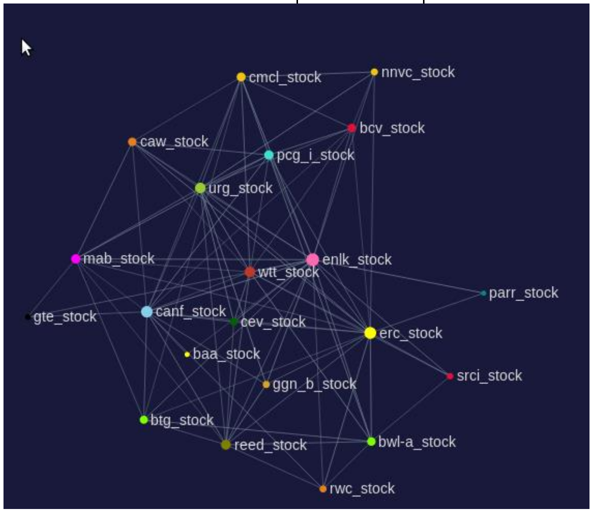
Type a stock name and click the graph to show. Note that if there is not enough data (raw data file size <10K), exception will be raised.

VI. Experiment Results

1. Visualization of Correlation



Center: enlk
Depth: 2
Condition: |Correlation| >0.9



Center: enlk
Depth: 1
Condition: |Correlation| >0.9

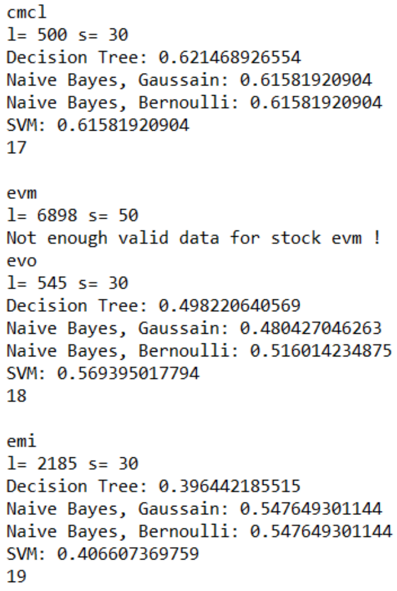
With what we learned in class we wrote the correlation data to csv file and load it as graph, and visualized it.

2. Rise\drop Prediction

In the first place, we choose cross validation to verify our methods. We split the data into two, 70% for training and 30% for testing.

For rise\drop prediction, which is essentially binary classification, we tried Decision Tree, Naïve Bayes with Gaussian and multinomial model, and Support Vector Machine (SVM). We also used methods like K-Nearest

Neighbors with Ball-Tree on a few stocks, but did not work well.

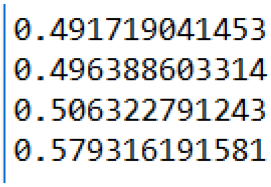


```
cmcl
l= 500 s= 30
Decision Tree: 0.621468926554
Naive Bayes, Gaussain: 0.61581920904
Naive Bayes, Bernoulli: 0.61581920904
SVM: 0.61581920904
17

evm
l= 6898 s= 50
Not enough valid data for stock evm !
evo
l= 545 s= 30
Decision Tree: 0.498220640569
Naive Bayes, Gaussain: 0.480427046263
Naive Bayes, Bernoulli: 0.516014234875
SVM: 0.569395017794
18

emi
l= 2185 s= 30
Decision Tree: 0.396442185515
Naive Bayes, Gaussain: 0.547649301144
Naive Bayes, Bernoulli: 0.547649301144
SVM: 0.406607369759
19
```

Nice accuracy and Poor accuracy



```
0.491719041453
0.496388603314
0.506322791243
0.579316191581
```

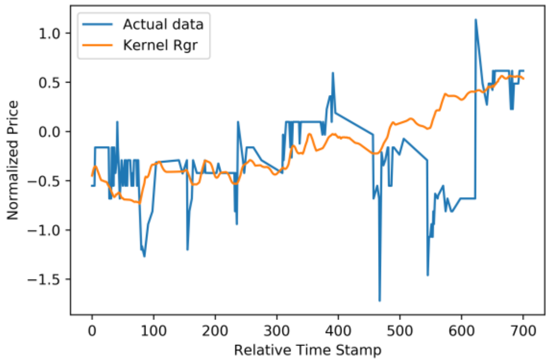
Accuracy of DecisionTree, Naïve Bayes with Gaussian, Naïve Bayes with Multinomial, SVM.

Though accuracy over 50% means potentially profitable, we need to be aware that the output is binary, 50% is equal to wild guess.

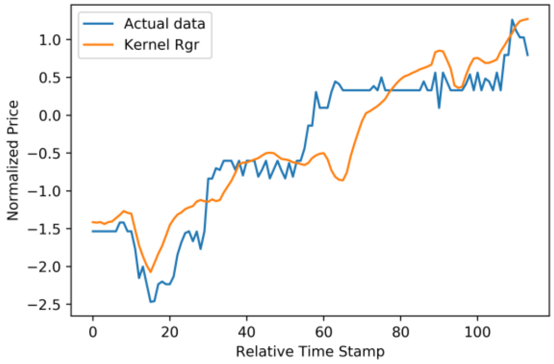
As I have mentioned in **IV-2. Machine Learning Algorithms**, some stocks have obvious relationship with “difference” of other correlated stocks, while some stocks do not. Such is especially true if we look at the accuracy of some certain stocks:

3. Price Prediction

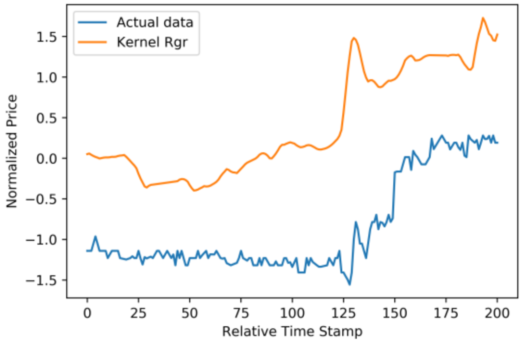
For price prediction, which is essentially regression, we tried Linear Regression and Kernel Regression. As such relationship is not necessarily linear, Kernel Regression worked better.



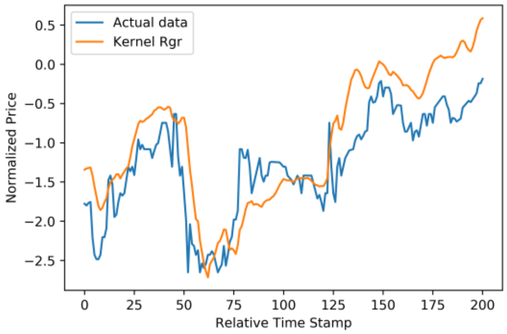
(stock: caw)
Cannot predict sharp rises or drops.



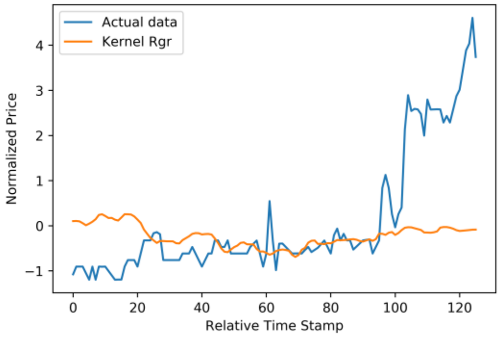
(stock: ae)
Well predicted stock



(Stock: aau)
Relatively close.



(Stock: lglr)
Quite accurate.



(Stock: mlss)
Again, failed to see sharp changes.

The regression can predict the general inclination, but it did not see the all the sharp rises and drops. My explanation is that, the drastic changes might be caused by other things, e.g. big scandal, new policy, etc. Such cannot be foreseen from simply the data of the correlated stocks.

VII. Conclusion

We used numpy to find the correlations between stocks and combined them with the concept of Moving Average from momentum trading strategy and used sklearn to develop and predict the price movement. We found that for some stocks, they are indeed influenced by other stocks, and such pattern can be utilized to predict the stock price movement. But it also happens that some stocks are not easily influenced by others; sometimes stock prices change because of other factors, and therefore these cases cannot be accurately predicted. The results show that sharp movements of stock prices tend not to be foreseen, while for some stocks, the general inclination and fluctuation can, to some extent be predicted.

Acknowledgment

We appreciate what we have learned from course Big Data Analytics and Professor Lin. In our project, we visualized the correlation we found using the tools we previously learned.

Appendix

Youtube link: https://youtu.be/58WDIJlTx1Y

References

[1] Jegadeesh, N. and Titman, S. (2001), Profitability of Momentum Strategies: An Evaluation of Alternative Explanations. The Journal of Finance, 56: 699–720. doi: 10.1111/0022-1082.00342J.

[2] Hong, K. J., and Stephen Satchell. "Time series momentum trading strategy and autocorrelation amplification." Quantitative Finance 15.9 (2015): 1471-1487.

[3] StackOverflow: How to interpolate NaN values in numpy array
https://stackoverflow.com/questions/6518811/interpolate-nan-values-in-a-numpy-array